

《数据库应用技术》

总题量 (941): 单选(538) 简答(52) 判断(286) 综合题(65)

单选(538)--

- 1、“FROM 表 1LEFTJOIN 表 2ON 表 1. 列=表 2. 列”，该子句表达的是 ()。-->**B.该子句表达是左外连接，连接结果中将包含表 1 中的全部数据**
- 2、“SELECTNULL+100”的返回结果是 ()。-->**A.NULL**
- 3、NULL+100 的执行结果是-->**NULL**
- 4、SQLServer 的系统主数据库是 ()。-->**A.master**
- 5、SQLserver 的账号管理功能中，账号可分为 ()。-->**C.登录账号和用户账号**
- 6、SQLServer 默认的登陆账号为 ()。-->**A.sa**
- 7、SQLSERVER 数据库的备份不支持 ()。-->**D.I/O 接口设备**
- 8、SQLServer 数据库一般包括两大类数据库类型 ()。-->**D.系统数据库和用户数据库**
- 9、SQLServer 数据库中，系统表名的前缀通常是 ()。-->**B.sys**
- 10、SQLServer 数据库中的 NULL 值 (空值) ()。-->**D.表示没有值、不确定的值**
- 11、SQLServer 数据库中的系统表的表名通常以()为前缀。-->**B.sys**
- 12、SQLServer 数据库中的一个完整的备份通常要包括 ()。-->**A.系统数据库、用户数据库和事务日志**
- 13、SQLServer 提供的 Nchar、Nvarchar、Ntext 等几种双字节数据类型实际上是一种 ()。-->**A.Unicode 数据类型**
- 14、SQLServer 系统中的所有系统级信息存储于 () 数据库中。-->**A.master**
- 15、SQLServer 中进行数据库恢复的 SQL 语句是 ()。-->**C.RESTOREDATABASE**

- 16、SQLSE 设 RVER 数据库的备份不支持 ()。磁带设备-->**D.I/O 接口设备**
- 17、TOP5percent 的作用是 ()。-->**D.返回查询结果的前 5%行数据**
- 18、_SQLServer 数据库是基于 () 数据模型的。-->**C.关系模型**
- 19、() 操作属于数据控制。-->**A.REVOKE**
- 20、() 是对数据库进行插入、删除、修改和查询等基本操作。-->**B.数据操纵**
- 21、() 是用户看到的数据视图。-->**A.外模式**
- 22、() 数据库包含了所有系统级信息，对 SQLServer 系统来说至关重要，一旦受到损坏，有可能导致 SQLServer 系统的彻底瘫痪。-->**A.master 数据库**
- 23、() 属于数据的动态特征。-->**B.插入数据**
- 24、按照账号管理的方式，SQLServer 用户分为 ()。-->**C.登录账号和用户账号**
- 25、把第一个关系和第二个关系中都出现的记录提取出来，形成新的关系称为 ()。-->**C.两个关系的交**
- 26、把一个关系 R 中按所需顺序选取若干个属性构成新的关系称为 ()。-->**C.投影运算**
- 27、本教材介绍的数据库管理系统基于 () 数据库。-->**C.MicrosoftSQLServer**
- 28、本教材介绍的数据库管理系统是 ()。-->**C.Microsoft SQL.Server**
- 29、不属于 VB 中 ADO 对象模型中的对象的是 ()。-->**D.Update**
- 30、不属于事务特征的是 ()。-->**完整性**
- 31、不属于数据库管理数据的特点的是 ()。-->**B.数据冗余不可避免**
- 32、不属于数据库管理系统功能的是 ()。-->**B.数据库文件读写**
- 33、不属于数据模型三要素的是 ()。-->**数据库管理系统**
- 34、采用事务控制机制对材料管理数据库进行操作，利用 UPDATE 语句将材料号为“ADOS”的材料改为“BOO5”，如果对入库材料表的更新操作结束后，还没来得及对出库材料表进行更新操作，突然停电了，SQL 的事务控制功能将 ()。-->**C.清除对入库材料表的修改**
- 35、查询考试成绩大学 75 分的学生的学号和姓名，正确的语句是 ()。-->**C.SELECT Sno,SnameFROM Student WHERE Sno IN (SELECT Sno FROM SC WHERE Grade>75)。**
- 36、查询考试成绩大于 75 分的学生的学号和姓名，正确的语句是 ()。-->**D.20**
- 37、查询课程名中带有“Windows”的课程，则条件语句应包含 ()。-->**B.WHERE 课程名 LIKE '%Windows%'**
- 38、查询年龄最大的前 3 位学生的姓名和年龄，下列语句正确的是 ()。-->**C.SELECT TOP 3 姓名, 年龄 FROM 学生表 ORDER BY 年龄 DESC**
- 39、查询年龄最大的前 5 位学生的姓名和年龄的语句，正确的是 ()。-->**C.SELECT TOP 5 姓名, 年龄 FROM 学生表 ORDER BY 年龄 DESC**
- 40、查询年龄最大的职工姓名，正确的语句是 ()。-->**D.实现复杂的约束**

- A.SELECT 姓名 FROM 职工表 WHERE 年龄= ()。SELECTMAX (年龄) FROM 职工表)。
- 41、查询年龄最小的职工姓名，正确的语句是 ()。-->**C.SELECT 姓名 FROM 职工表 WHERE 年龄=(SELECT TOP 1 年龄 FROM 职工表 ORDERBY 年龄 ASC)**
- 42、查询所有学生及其选课情况的详细信息的语句，正确的是()。-->**C.SELECT * FROM Student INNER JOIN SC ON Student. Sno=SC. Sno**
- 43、查询姓名中带有“芳”字的学生，则 SQL 条件语句应包含 ()。-->**B.Where 姓名 LIKE '%芳%'**
- 44、查询姓名中带有“玲”字的学生，则条件语句应包含 ()。-->**B.WHERE 姓名 LIKE '%玲%'**
- 45、查询语句中，错误的是 ()。-->**B. SELECT Sno FROM SC GROUP BY Sno WHERE COUNT(*) > 3**
- 46、触发器的主要作用是 ()。-->**D.实现复杂的约束**
- 47、创建存储过程的用处主要是 ()。-->**提高数据操作效率**
- 48、创建视图的主要作用是 ()。-->**D.提供用户视角的数据**
- 49、创建数据库的 SQL 语句是 ()。-->**C.CREATEDATABASE**
- 50、从广义上讲，数据库系统由 () 组成。-->**数据库，数据库管理系统，用户和应用程序**
- 51、从逻辑层次上看，数据库不包括 ()。-->**D.内模式**
- 52、从最终用户应用程序的视角看，数据库系统的三级模式结构是 ()。-->**C.外模式、模式和内模式**
- 53、当关系 R 和 S 进行连接操作时，如果 R 中的元组不满足连接条件，在连接结果中也会将这些记录保留下来的操作是 ()。-->**左外连接**
- 54、当执行由 UPDATE 语句引发的触发器时，下列关于该触发器临时工作表的说法，正确的是 ()。-->**C.系统会自动产生 INSERTED 表和 DELETED 表，用 INSERTED 表存放更改后的数据，用 DELETED 表存放更改前的数据**
- 55、第二范式是指 ()。-->**B.设一个关系为 R，它是满足第一范式的，若 R 中不存在非主属性对主键的部分函数依赖，则称该关系符合第二范式**
- 56、对 char (10) 类型的列，若存储“中国”，其占用的字节数是 ()。-->**C.10**
- 57、对 nchar (10) 类型的列，若存储“数据库”，其占用的字节数是 ()。-->**D.20**
- 58、对 NCHAR(10)类型的数据，系统为其分配的存储空间是()。-->**B.固定的 20 个字节**
- 59、对 SQLServer 数据库，下列属于系统推荐的次要数据文件的扩展名是 ()。-->**C.. ndf**
- 60、对 SQLServer 数据库，下列属于系统推荐的日志文件扩展名的是 ()。-->**D.. ldf**
- 61、对 SQLServer 数据库，下列属于系统推荐的主要数据文件的扩展名是 ()。-->**B.. mdf**
- 62、对 Student (Sno,Sname) 和 SC (Sno,Cno,Grade) 表，查询考试成绩大于 75 分的学生的学号和姓名，正确的语句是 ()。-->**C.SELECT Sno, Sname FROM Student WHERE Sno IN (SELECT Sno FROM SC WHERE Grade>75)**

63、对 Student (Sno,Sname) 和 SC (Sno,Cno,Grade) 表, 查询考试成绩小于 60 分的学生的学号和姓名, 正确的语句是 ()。
-->C.SELECT Sno, Sname FROM Student WHERE Sno IN (SELECT Sno FROM SC WHERE Grade<60)

64、对关系模式进行规范化的主要目的是 ()。-->B.维护数据的一致性

65、对于病房管理数据库, 当采用事务控制机制时, 利用 UPDATE 语句将病人表和病房情况表中病人编号“A1001”改为“AB1001”, 如果对病人表的更新操作结束后, 还没来得及对病房情况表中对应的病人编号字段进行更新操作, 突然停电了, SQL 的事务控制功能将 ()。-->C.清除对病人表的修改

66、对于教学管理数据库, 当采用事务控制机制时, 利用 UPDATE 语句将课程表中课程号为“A002”的课程改为“A2002”后, 还没来得及对成绩表进行更新操作, 突然停电了, SQL 的小事务控制功能将 ()。-->C.清除对课程表的修改

67、对于教学管理数据库, 当采用事务控制机制时, 利用 UPDATE 语句将学生表中学号为“002”的学生的学号改为“2002”, 还没来得及对成绩表进行更新操作, 突然停电了, SQL 的事务控制功能将 ()。-->C.清除对学生表的修改

68、对于教学管理数据库, 当采用事务控制机制时, 利用 UPDATE 语句将学生表中学号为“002”的学生学号改为“2002”后, 如果对学生表的更新操作结束后, 还没来得及对成绩表中对应的学号字段进行更新操作, 突然停电了, SQL 的事务控制功能将 ()。-->C.清除对学生表的修改

69、给 Course (Cno,Cname,Credit) 表插入一条新记录, 其中 Credit 为整型类型, 其他为字符串类型。课程号为“C205”, 课程名是“高等数学”, 学分是 4。正确的语句是 ()。-->B.INSERT INTO Course VALUES('C205','高等数学',4)

70、给 SC (sno,cno,grade) 表插入一条新记录, 学号为“20160105”, 选修的课程号是“002”, 成绩空缺。下列语句正确的是 ()。
-->B.INSERT INTO SC VALUES('20160105','002', NULL)

71、给 SC (Sno,Cno,Grade) 表插入一条新记录, 学号为“20160506”, 选修的课程号是“306”, 成绩空缺。下列语句正确的是 ()。
-->B.INSERT INTO SC VALUES('20160506','306', null)

72、给 SC 表插入一条新记录, 学号为“20160105”, 选修的课程号是“002”, 成绩空缺。下列语句正确的是 ()。-->B.INSERT INTO SC VALUES ('20160105','002',null)

73、给 SC 表插入一条新记录, 学号为“20160105”, 选修的课程号是“002”, 成绩空缺。下列语句正确的是 ()。
B. INSERT INTO SC VALUES('20160105', '002', null)

74、给 Student (Sno, Sname, Ssex, Sage) 表 (Sage 为整型, 其他均为字符型) 插入一条新记录, 学号为“20160203”, 姓名: 李小红, 性别: 女, 年龄: 18。正确的语句是 ()。
C.INSERT INTO Student VALUES('20160203', '李小红', '女', 18)

75、给 T (c1, c2, c3) 表插入一条新记录, c1 列为“111”, c2 列为“112”, c3 列空缺。下列语句正确的是 ()。
C.INSERT INTO T VALUES('111', '112', null)

76、给教师(工号, 姓名, 工资)表插入一条新记录, 工号为“1111”, 姓名为“张三”, 工资空缺。正确的语句是 ()。
A.INSERT INTO 教师 VALUES ('111', '张 E', NULL)

77、关系数据库规范化理论是为了解决关系数据库中 () 问题而引入的。-->A.插入、删除和数据冗余

78、关系数据库系统采用关系模型作为数据的组织方式。关系模型通常由三部分组成 ()。-->A.数据结构、关系操作集合和关系的完整性

79、关于 DEFAULT 约束的说法, 错误的是 ()。
d. 在列级完整性约束和表级完整性约束处都可以定义 DEFAULT 约束

80、关于 soLServer 数据库文件的说法, 错误的是 ()。
D.日志文件必须与数据文件存放在不同的磁盘上

81、关于 SQLServer 备份设备的说法, 正确的是 ()。-->备份设备是一个逻辑设备, 可以建立在磁盘上

82、关于 SQLServer 差异备份的说法, 正确的是 ()。
B.通常情况下差异备份比完整备份花费的时间少

83、关于 SQLServer 差异备份的说法, 正确的是 ()。-->B.通常情况下差异备份比完整备份花费的时间少

84、关于 SQLServer 身份验证模式的说法, 正确的是 ()。
在安装过程中可以设置身份验证模式, 安装完成之后还可以再对其进行修改

85、关于 SQLServer 身份验证模式的说法, 正确的是 ()。-->C.在安装过程中可以设置身份验证模式, 安装完成之后还可以再对其进行修改

86、关于 SQLServer 数据库的说法, 正确的是 ()。
一个数据库可由多个数据文件和多个日志文件组成

87、关于 SQLServer 数据库的说法, 正确的是 ()。-->一个数据库可由多个数据文件和多个日志文件组成

88、关于 SQLServer 数据库服务器登录账户的说法, 错误的是 ()。
所有的 Windows 用户都自动是 SQL Server 的合法账户

89、关于 SQLServer 数据库文件的说法, 错误的是 ()。-->D.日志文件必须与数据文件存放在不同的磁盘上

90、关于 TOP 选项的作用, 说法正确的是 ()。
A.TOP10 返回的结果只能是前 10 行数据

91、关于“FROM 表 1 LEFT JOIN 表 2 ON 表 1.列一表 2.列”的说法, 正确的是 ()。
B.该子句表达是左外连接, 连接结果中将包含表 1 中的全部数据

92、关于差异备份的说法, 正确的是 ()。-->B.差异备份备份的是从上次完整备份到当前时间数据库变化的内容

93、关于传统关系代数的说法, 错误的是 ()。-->B.有 m 目 x 行数据的关系 R 和有 n 目 y 行数据的关系 S 进行笛卡尔积运算, 其结果是有 (m+n) 目和 (x+y) 行数据的关系

94、关于存储过程的说法, 正确的是 ()。-->A.在定义存储过程的代码中可以包含数据的增、删、改、查语句

95、关于第三范式的说法, 错误的是 ()。-->第三范式的关系模式一定不包含传递函数依赖, 但可以包含部分函数依赖

96、关于非聚集索引的说法, 正确的是 ()。-->A.在一个表上可以建立多个非聚集索引

97、关于关系代数中并运算的说法, 正确的是 ()。-->C.并运算的结果是两个关系中全部数据的集合, 并且去掉重复的数据

98、关于关系代数中选择运算的说法, 正确的是 ()。-->A.选择运算是从行的方向选择集合中的数据, 选择运算后的行数有可能减少

99、关于关系数据模型的说法, 正确的是 ()。-->B.关系数据模型采用的数据结构是二维表

100、关于关系运算中选择运算的说法, 正确的是 ()。-->B.选择运算是从水平方向选取表中满足条件的记录

101、关于集合并运算的说法, 正确的是 ()。-->进行并运算的两个关系的列个数必须相同, 对应列的语义相同

102、关于聚集索引的说法, 正确的是 ()。-->B.在一个表上最多只能建立一个聚集索引

103、关于扩大 SQLServer 数据库空间的说法, 错误的是 ()。-->D.只能通过添加新文件的方法扩大日志的空间

104、关于命名实例和默认实例, 正确的说法是-->一台计算机上可以有多个命名实例

105、关于模式的正确描述是 ()。-->D.模式只有一个

106、关于实体联系模型中联系的说法, 错误的是 ()。-->D.一个联系可以与任何实体有关

107、关于视图的说法, 正确的是 ()。-->B.视图提供了数据的逻辑独立性

108、关于收缩 SQLServer 数据库空间的说法, 错误的是 ()。-->C.收缩数据文件空间时不能小于该文件的初始大小

109、关于数据库管理系统的说法, 错误的是 ()。-->C.数据库应用程序可以不经过数据库管理系统而直接读取数据库文件

110、关于数据库管理系统的说法, 错误的是 ()。-->C.数据库管理系统不支持多用户同时访问相同数据库

111、关于数据库三级模式中“模式”的说法, 错误的是 ()。
C.模式也称为存储模式。

112、关于数据库设计中新奥尔良方法的说法, 错误的是 ()。-->逻辑结构设计是在需求分析之后、概念结构设计之前进行的

113、关于索引的说法, 错误的是 ()。-->A.索引由索引项组成, 索引项只能由表中的一个列组成

114、关于索引的说法, 正确的是 ()。-->C.在一个表上可以建立多个唯一的非聚集索引

115、关于用文件管理数据的说法, 错误的是 ()。-->D.将相关的数据存储在一个文件中, 有利于用户对数据进行分类, 因此也可以加快用户操作数据的效率

116、关于自连接的说法, 错误的是 ()。-->A.自连接属于外连接的一种, 必须为进行自连接的两个表起别名

117、关于自然连接的说法, 错误的是 ()。-->进行自然连接的两个关系可以没有列名相同的列

118、货币类型数据类型属于 ()。-->A.数值型

119、将第一个关系的记录加到第二个关系中, 生成的新关系称为 ()。-->B.两个关系的并

120、仅允许 Windows 操作系统用户连接到 SQLServer 数据库服务器的验证模式是 ()。-->B.Windows 身份验证模式

121、聚合函数中, 不忽略空值的是 ()。-->COUNT (*)

122、利用 Recordset 对象可以得到查询结果集。将 SC 表中的所有数据放置到 Recordset 对象 ADOrs 中的语句, 正确的是 ()。
A. ADOrs.Open "Select * From SC"

123、利用文件方式管理数据, 存在的缺点是 ()。-->**B.数据冗余不可避免**

124、利用文件管理数据库, 存在 () 缺点。-->**B.数据冗余不可避免**

125、某数据表中有如下列定义语句: Credit NUMERIC(3,1) CHECK (Credit>0), 下列关于该语句功能的描述, 正确的是 () -->**C.Credit 的取值范围要大于 0**

126、某数据表中有如下列定义语句: Sno CHAR(9) PRIMARY KEY, 下列关于该语句功能的描述, 正确的是 () -->**A.Sno 为主键**

127、如 A 果要存储的数据是图形图像等, 应使用 () 做为数据类型。-->**D.位型**

128、如果某数据库的日志文件空间已用完, 则用户对该数据库 ()。-->**D.可以查询数据**

129、如果日志文件的存储空间不够, 则 ()。-->**D.只可以查询数据库中数据**

130、如果事务 T 获得了数据项 A 上的共享锁, 则其他事务对 A ()。-->**A.只能读不能写**

131、如果事务 T 获得了数据项 A 上的排他锁, 则其他事务对 A ()。-->**D.不能读也不能写**

132、如果希望从成绩表中查询出所有成绩大于等于 90 分的记录, 那么 SQL 条件语句应该是 ()。-->**A.Where 成绩 >= 90**

133、如果希望从成绩表中查询出所有成绩小于等于 60 分的记录, 那么条件语句应该是 ()。-->**D.Where 成绩 <= 60**

134、如果希望从学生表中查询出所有姓“张”的同学, 那么条件语句应该是 ()。-->**B.WHERE 姓名 LIKE '张%'**

135、如果希望从学生表中查询出同学的平均年龄, 那么语句中应该包含 ()。-->**C.SELECT AVG (年龄) AS 平均 FROM 学生表**

136、如果希望从学生表中查询出同学的平均年龄, 下列语句中正确的是 ()。
C.SELECT AVG (年龄) AS 平均 FROM 学生表

137、如果希望所有新创建的数据库自动具有某些对象或授权, 可以将这些对象和授权事先放在 ()。-->**C.model 数据原**

138、如果希望统计学生表中选修网页设计课程的学生的人数, 那么语句中应该包含 ()。-->**D.SELECT COUNT (*) AS 专业人数 FROM 学生表 WHERE 专业='网页设计'**

139、如果要存储的数据是带有小数位的数据, 应使用 () 做为数据类型。-->**A.数值数据类型**

140、如果要存储的数据是图形图像等, 应使用 () 作为数据类型。-->**D.二进制数据类型**

141、如果要存储的数据是图形图像等, 应使用 () 做为数据类型。-->**D.位型**

142、如果要设计一个用于记录“真/假”、“开/关”等状态信息的数值, 最佳的数据类型是 ()。-->**B.bit**

143、如果有函数依赖 X→Y, 并且对 X 的某个真子集 X', 有 X'→Y 成立, 则称 ()。-->**Y 部分函数依赖于 X**

144、若关系模式中不存在非主属性对主键的部分函数依赖, 则此关系模式至少属于 ()。-->**B.第二范式**

145、若关系模式中不存在非主属性对主键的传递函数依赖和部分函数依赖, 则此关系模式至少属于 ()。-->**C.第三范式**

146、若关系模式中不存在复合数据项, 则此关系模式一定属于 ()。-->**A.第一范式**

147、若事务 T 对数据项 D 已加了 S 锁, 则其他事务对数据项 D ()。-->**A.可以加 S 锁, 但不能加 X 锁**

148、若希望利用 CASE 表达式显示不同课程, 下列正确的是 ()。-->**D. Case 课程 When '英语' THEN 'ENG' When '物理' THEN 'PHY' When '数学' THEN 'MAT' End**

149、若希望利用 CASE 表达式显示部门, 下列正确的是 ()。-->**D. Case 部门 When '计算机系' THEN 'CS' When '数学系' THEN 'MA' Else '其他部门' End**

150、若要定义“价格”列的数据类型, 假设价格用定点小数类型保存, 整数部分 6 位, 小数点后 2 位。最合适的类型是 ()。-->**C.numeric(8,2)**

151、若要收回用户 U 对 T 表的查询权, 下列语句正确的是 ()。C.REVOKE SELECT ON T FROM U

152、若要授予用户 U 对 T 表具有查询权限, 下列语句正确的是 ()。-->**B.GRANT SELECT ON T TO U**

153、删除存储过程 P1 的语句, 正确的是 ()。D. DROP PROC P1

154、删除计算机系学生 (在 student 表中) 的选课记录 (在 SC 表中) 的语句, 正确的是 ()。B.DELETE SC FROM SC JOIN Student b ON SC.Sno=b.Sno WHERE Sdept='计算机系'

155、设 ADOcn 是一个 Connection 对象, 与数据库的连接字符串存放在 cnStr 变量中。下列关于使用 ADOcn 建立与数据源连接的语句, 正确的是 ()。B. ADOcn. Open cn Str

156、设 SC 表中记录成绩的列为: Grade, 类型为 int。若在查询成绩时, 希望将成绩按“优”、“良”、“中”、“及格”和“不及格”形式显示, 正确的 Case 函数是 ()。C.Case
When Grade between 90 and 100 THEN '优'
When Grade between 80 and 89 THEN '良'
When Grade between 70 and 79 THEN '中'
When Grade between 60 and 69 THEN '及格'
Else '不及格'
End

157、设关系 R 是满足第一范式的, 若 R 中不存在非主属性对主键的部分函数依赖, 则 R 符合 ()。-->**A.第二范式**

158、设计性能优良的关系模式称为规范化, 规范化的主要理论依据是 ()。-->**C.关系规范化理论**

159、设课程成绩表的结构为 (学号, 课程号, 成绩)。如果希望查询出“成绩大于 90 分且课程号首字母为 A 的所有记录”则对应的 SQL 语句是 ()。-->**C.SELECT * FROM 课程成绩表 WHERE 成绩 > 90 AND 课程号 LIKE 'A%'**

160、设某职工表中有用于存放年龄 (整数) 的列, 下列类型中最适合年龄列的是 ()。C.tinyint

161、设某职工表中有用于存放年龄 (整数) 的列, 最适合年龄列的类型是 ()。-->**C.tinyint**

162、设实体 A 与实体 B 之间是一对多联系。如下逻辑结构设计方法中, 最合适的是 ()。-->**A.实体 A 和实体 B 分别对应一个关系模式, 且外键放在实体 B 的关系模式中**

163、设实体 A 与实体 B 之间是一对多联系。下列进行的逻辑结构设计方法中, 最合适的是 ()。-->**A.为实体 A 和实体 B 分别设计一个关系模式, 且外键放在实体 B 的关系模式中**

164、设数据库中已有表 5-1 至 5-3 所示的 Student、Course 和 SC 表。现要查询学生选的第 2 学期开设课程的情况, 只需列出学号、姓名、所在系和所选的课程号。该查询涉及到的表是 ()。D. Student、SC 和 Course 表

165、设学生表和成绩表的结构分别为 (学号, 姓名, 所在系) 制 (学号, 课程名, 成绩), 如果希望按分数降序查询出“英语系中选修了计算机课程的学生姓名和成绩”, 则对应的 SQL 语句是 ()。-->**D.SELECT 姓名, 成绩 FROM 学生表, 成绩表 WHERE 所在系='英语系' AND 课程名='计算机' AND 学生表.学号=课程表.学号 ORDER BY 成绩 DESC**

166、设学生表和课程表的结构分别为 (学号, 姓名) 和 (学号, 课程名, 成绩), 如果希望查询出“选修了, 数据库应用技术”课程的学生姓名和成绩”, 则对应的 SQL 语句是 ()。D.SELECT 姓名, 成绩 FROM 学生表 JOIN 课程表 ON 学生表.学号=课程表.学号 WHERE 课程名='数据库应用技术'

167、设学生表和选课表的结构分别为 (学号, 姓名) 和 (学号, 课程号, 成绩), 如果希望查询出“所有姓‘王’的学生且选修了‘A001’课程的学生”, 则对应的 SQL 语句是 ()。-->**C.SELECT 姓名 FROM 学生表, 选课表 WHERE 学生表.学号=选课表.学号 AND 学生表.姓名 LIKE '王%' AND 选课表.课程号='A001'**

168、若要查询不姓“张”的学生, 下列条件子句中正确的是 ()。-->**C.Where 姓名 NOT LIKE '张%'**

169、若要查询姓“张”且名字是 2 个汉字的学生, 下列条件子句中正确的是 ()。B.WHERE 姓名 LIKE '张_'

170、若要查询姓名中第二个字是“红”的学生, 下列条件子句中正确的是 ()。-->**A.Where 姓名 LIKE ' _ 红%'**

171、若要查询姓名中第二个字是“萍”的学生, 下列条件子句中正确的是 ()。-->**B.where 姓名 LIKE ' _ 萍%'**

172、设一个关系为 R, X 和 Y 是它的两个属性集。若对于 X 上的每个值都有 Y 上的一个惟一值与之对应, 则称 X 和 Y ()。-->**C.具有函数依赖关系**

173、设一个关系为 R, 如果它的每个属性都是不可再分的, 则称这个关系是符合 ()。-->**A.第一范式**

174、设有 Student (Sno, Sname, Ssex, Sage), 给所有学生年龄增加 1 岁, 正确的语句是 ()。C.UPDATE Student SET Sage=Sage+1

175、设有 Teachers 表, 该表的定义如下:
CREATE TABLE Teachers(
Tno CHAR(8) PRIMARY KEY,
Tname VARCHAR(10) NOT NULL,
Age TINYINT CHECK(Age BETWEEN 25 AND 65))
插入语句中, 不能正确执行的是 (D. INSERT INTO Teachers VALUES('T100','张宏'))。

176、设有表 M, 将其中的 Type 列的数据类型改为 NCHAR(4)。能实现该功能的语句是 ()。B. ALTER TABLE M ALTER COLUMN Type NCHAR(4)

177、设有表 T，现要为该表新增加一个列，列名为：c1，类型为 int。能正确实现该功能的语句是（）。

a. ALTER TABLE T ADD c1 int

178、设有定义索引的语句：

CREATE UNIQUE CLUSTERED INDEX IdxonT (C1) 该语句的作用是（）。

D.在 C1 列上建立了一个唯一的聚集索引

179、设有定义索引的语句：CREATE UNIQUE INDEX IdxonT (C1) 该语句的作用是（）。-->C.在 C1 列上建立了一个唯一的非聚集索引

180、设有关系模式：图书销售（书号，销售时间，销售数量，销售总价），设同一时间可销售多种图书，但同一图书同一时间只能销售一次。其中适合作为主键的是（）。-->C.（书号，销售时间）

181、设有关系模式：学生（学号，姓名，所在系，身份证号）、系（系名，系办公地点）。关于两个关系模式中属性引用关系的描述，正确的是（）。-->A.设置学生关系模式中的“所在系”为外键

182、设有关系模式：学生（学号，姓名，所在系，专业，身份证号），其中能够作为候选键的是（）。-->B.学号、身份证号

183、设有关系模式：学生（学号，姓名，身份证号），其中“学号”是主键。若要限制“身份证号”取值不重，下列约束最合适的是（）。-->B.UNIQUE

184、设有描述商品销售情况的关系模式：销售（商品号，顾客号，销售时间，销售数量），设一个商品可销售给多名顾客，一个顾客可以多次购买同一个商品，每次购买或销售都要记录下时间和数量。该关系模式的主键是（）。-->C.（商品号，顾客号，销售时间）

185、设有描述图书出版情况的关系模式：出版（书号，出版日期，印刷数量），设一本书可以被出版多次，每次出版都有一个印刷数量。该关系模式的主键是（）。-->（书号，出版日期）

186、设有描述图书印刷情况的关系模式：印刷（书号，印刷日期，印刷数量），设一本书可在不同日期被印刷多次，每次印刷都有一个印刷数量。该关系模式的主键是（）。-->B.（书号，印刷日期）

187、设有描述鲜花进货情况的关系模式：鲜花（鲜花编号，进货日期，进货价格，进货数量），设可在不同日期多次进货同一种鲜花，每次进货的价格和数量可以不同。该关系模式的主键是（）。-->B.（鲜花编号，进货日期）

188、设有如下备份操作：现从备份中对数据库进行恢复，正确的恢复顺序为（）。-->B.完整备份 1，差异备份 1，日志备份 3，日志备份 4

189、设有商品表（商品号，商品名，类别，单价），现要将 S011 和 S012 商品的单价加 10 元，下列语句中正确的是（）。-->A.UPDATE 商品表 SET 单价=单价+10 WHERE 商品号='S011' OR 商品号='S012'

190、设有实体：班、教室，一个班可在多个教室上课，一个教室可用于多个班使用，则班和教室之间联系的类型是（）。-->D.多对多

191、设有实体：图书、作者，假设一本书可以有多个作者，一个作者可以编写多本图书，则图书和作者直接的联系类型是（）。

-->D.多对多

192、设有数据表：Student (Sno,Sname,Ssex,Sage)，给所有学生年龄增加 1 岁，正确的语句是（）。-->C.UPDATE Student SET Sage = Sage + 1

193、设有数据表：教师（教师号，教师名，职称，所在部门）；授课（教师号，课程号，授课学年）。现要删除“讲师”职称的授课记录。正确的语句是（）。-->B.DELETE 授课 FROM 教师 a JOIN 授课 b ON a. 教师号 = b. 教师号 WHERE 职称 = '讲师'

194、设有数据表：商品（商品号，商品名，类别，单价）；生产（商品号，厂家编号，生产日期，生产数量）。现要删除 2014 年 12 月 31 日之前生产的全部商品。正确的语句是（）。-->B.DELETE 商品 WHERE 商品号 IN (SELECT 商品号 FROM 生产 WHERE 生产日期 < '2014/12/31')

195、设有数据表：图书（书号，书名，类别，出版日期，单价），现将“量子力学”类图书的单价降低 10。正确的语句是（）。-->B.UPDATE 图书 SET 单价 = 单价 - 10 WHERE 类别 = '量子力学'

196、设有数据表：选课（学号，课程号，成绩），现要将学号为 S001 和 S002 学生的成绩加 15 分，正确的语句是（）。-->A.UPDATE 选课 SET 成绩 = 成绩 + 15 WHERE 学号 = 'S001' OR 学号 = 'S002'

197、设有数据表：选课（学号，课程号，成绩），现要删除成绩为空的所有选课记录，正确的语句是（）。-->C.DELETE 选课 WHERE 成绩 IS NULL

198、设有图书（书号，书名，类别，出版日期，单价），现将“量子力学”类图书的单价降低 10。正确的语句是（）。-->B.UPDATE 图书 SET 单价=单价-10 WHERE 类别='量子力学'

199、设有图书（书号，书名，类别，价格），现要将查询结果按类别降序、价格升序排序。下列排序子句正确的是（）。

B.ORDER BY 类别 DESC, 价格 ASC

200、设有图书（书号：书名，价格），现要查询价格最低的图书价格，正确的是（）

C.SELECT MIN (价格) FROM 图书

201、设有图书销售表（书号，销售时间，销售数量，销售总价），设同一时间可销售多种图书，但同一图书同一时间只能销售一次。其中适合作为主键的是（）。-->D.（书号，销售时间）

202、设有下列定义索引的语句：

CREATE UNIQUE INDEX Idxon T (C1)

该语句的作用是（在 C1 列上建立了一个唯一的非聚集索引）

203、设有销售表（经销商号，汽车型号，销售时间，销售价格），现要将 S011 和 S012 汽车的单价减 1000 元，下列语句中正确的是（）。

A. UPDATE 销售表 SET 销售价格=销售价格 - 1000 WHERE 汽车型号='S011' OR 汽车型号='S012'

204、设有销售表（经销商号，汽车型号，销售时间，销售价格），现要删除销售价格为空的所有记录，下列语句中正确的是（）。

-->C.DELETE 销售表 WHERE 销售价格 IS NULL

205、设有选课（学号，课程号，成绩），现要将 S001 和 S002 学生的成绩加 10 分，正确的语句是（）。

A.UPDATE 选课 SET 成绩=成绩+10 WHERE 学号='S001' OR 学号='S002'

206、设有选课（学号，课程号，成绩），现要将 S001 和 S002 学生的成绩加 15 分，正确的语句是（）。

A.UPDATE 选课 SET 成绩=成绩+15 WHERE 学号='S001' OR 学号='S002'

207、设有选课（学号，课程号，成绩），现要删除“S001”和“S002”两个学生的选课记录，正确的语句是（）。

D.DELETE 选课 WHERE 学号=' S001 ' OR 学号=' S002 '

208、设有选课表（学号，课程号，成绩），现要查询没考试的学生学号和课程号，下列语句中正确的是（）。

C.SELECT 学号，课程号 FROM 选课表 WHERE 成绩 IS NULL

209、设有选课表（学号，课程号，成绩），现要统计选了课程的学生人数，正确的语句是（）。-->D.SELECT COUNT(DISTINCT 学号) FROM 选课表

210、设有学生表（学号，姓名，所在系），查询姓“刘”的学生人数的语句，正确的是（）。-->B.SELECT COUNT(*) FROM 学生表 WHERE 姓名 LIKE '刘%'

211、设有学生表（学号，姓名，所在系）。下列建立统计每个学生人数的视图语句，正确的是（）。-->D.CREATE VIEW v1(系名,人数) AS SELECT 所在系,COUNT(*) FROM 学生表 GROUP BY 所在系

212、设有学生表（学号，姓名，所在系，身份证号）和系表（系名，系办公地点）。关于两个表的引用关系的描述，正确的是（）。-->A.设置学生表中的“所在系”为外键

213、设有学生表（学号，姓名，所在系，专业，身份证号），其中能够作为候选键的是（）。-->B.学号、身份证号

214、设在 SC (Sno,Cno,Grade) 表上定义了如下触发器：CREATE TRIGGER tri1 ON SC AFTER INSERT...当执行语句：

INSERT INTO SC VALUES ('s001','c01',90) 会引发该触发器执行。下列关于触发器执行时表中数据的说法，正确的是（）。-->A.SC 表和 INSERTED 表中均包含新插入的数据

215、声明并创建 Connection 对象的语句，正确的是（）。

A. Dim ADONcn As New Connection

216、实体 A 与实体 B 之间是一对多联系，表示（）。-->B.实体 A 中的每个实例在实体 B 中可以有多个实例与其对应

217、事务的一致性是指（）。-->C.事务可以使数据库从一个一致性状态到另一个一致性状态

218、事务所包含的操作作为一个整体，要么都做，要么都不做，这个特性称为事务的（）。-->D.原子性

219、事务一旦提交，其对数据的更改就是永久的，这个特性称为事务的（）。-->B.持久性

220、事务一旦提交，其对数据库中数据的修改就是永久的，以后的操作或故障不会对事务的操作结果产生任何影响。这个特性是事务的（）。-->D.持久性

221、视图对应关系数据库三级模式中的（）。-->A.外模式

222、数据处理技术的发展不包括（）。-->B.信息管理阶段

223、数据库的备份不包括（）。-->A.操作系统

224、数据库管理系统常见的数据模型有（）。-->B.层次、关系和网状

225、数据库管理系统中的（）功能实现对数据库的插入、删除、修改。-->**B.数据操纵**

226、数据库管理系统中的（）功能实现对数据库的转储、恢复。-->**D.数据库的建立与维护**

227、数据库三级模式结构的划分，有利于（）。-->**A.数据的独立性**

228、数据库物理设计的主要内容包括（）。-->**A.存储记录的格式设计**

229、数据库系统的物理独立性是指（）。-->**B.不会因为数据存储结构的变化而影响应用程序**

230、数据库系统是由若干部分组成的。下列不属于数据库系统组成部分的是（）。-->**B.操作系统**

231、数据库系统中的三级模式以及模式间的映像提供了数据的独立性。下列关于两级映像的说法，正确的是（）。-->**C.外模式到模式的映像以及模式到内模式的映像都是由 DBMS 实现的**

232、数据库应用程序开发中，（）属于物理设计。-->**B.存储方法设计**

233、数据库应用程序开发中，概念设计阶段的主要目的是（）。-->**D.建立概念模型**

234、数据库应用程序开发中，下列哪一个环节属于物理设计（）。-->**B.存储方法设计**

235、数据库应用程序开发中，需求分析阶段的主要目的是（）。-->**A.回答“干什么”的问题**

236、数据库中一个事务的执行不能被其他事务干扰，这个特性称为事务的（）。-->**C.隔离性**

237、数据模型三要素是指（）。-->**数据结构、数据操作和数据完整性约束**

238、所述功能中，不属于 SQL 语言功能的是（）。-->**D.提供方便的用户操作界面功能**

239、讨论函数依赖主要是为了解决（）问题。-->**C.数据更新异常**

240、通常情况下，“数据库管理系统”的英文缩写是（）。-->**A.DBMS**

241、通常情况下，“数据库管理员”的英文缩写是（）。-->**C.DBA**

242、通过账号管理，SQLServer 用户分为（）。-->**C.登录账号和用户账号**

243、为变量赋值的语句中，错误的是（）。
D. SET @X = 10 + 'A'

244、为数据库系统的临时表提供工作空间的系统数据库是（）。-->**C.tempdb**

245、系模式：学生（学号,姓名,身份证号,性别,所在系),下列适合做候选键的是（）。-->**C.身份证号、学号**

246、下列 SQL 对语句中，字段的值具有唯一性的约束是（）。-->**A.学号 char (8) NOTNULLCONSTRAINTun_noUNIQUE**

247、下列 SQL 语句中，对字段创建唯一的聚集索引的约束是（）。-->**B.学号 Char (8) NOTNULLCONSTRAINTPk 学生表 PRIMARYKEY**

248、下列 SQL 语句中，进行默认约束的是（）。-->**D.成绩 numeric (9,2) CONSTRAINTdf_scoreDEFAULT**

249、下列 SQL 语句中，能够查询每个系教师人数的是（）。-->**C.SELECT 系编号, COUNT (*) FROM 教师表 GROUPBY 系编号**

250、下列 SQL 语句中，能够对查询结果进行排序的是（）。-->**A.SELECT*FROM 学生表 WHERE 专业='软件'ORDERBY 年龄**

251、下列 SQL 语句中，能够统计每个专业的人数的是（）。-->**A.SELECT 专业, COUNT (*) AS 学生人数 FROM 学生表 GROUPBY 专业**

252、下列 SQL 语句中，能够完成并运算是（）。-->**A.SELECT*FROM 篮球爱好者 UNIONSELECT*FROM 足球爱好者**

253、下列 SQL 语句中，能够完成插入功能的是（）。-->**A.INSERTINTO 学生表 VAUES ('2006001', '王小明','男')。**

254、下列 SQL 语句中，能够完成笛卡尔积运算是（）。-->**D.SELECT*FROM 素描画作者, 动漫作者**

255、下列 SQL 语句中，能够完成笛卡尔积运算是（）。-->**D.SELECT*FROM 篮球爱好者, 足球爱好者**

256、下列 SQL 语句中，能够完成交运算是（）。-->**B.SELECT*FROM 刘德华出演的电影 INTERSECTSELECT*FROM 梅艳芳出演的电影**

257、下列 SQL 语句中，能够完成交运算是（）。-->**D.SELECT*FROM 刘德华出演的电影, 梅艳芳出演的电影**

258、下列 SQL 语句中，能够完成求某列最大值的是（）。
B.SELECT MAX (Age) FROM Student

259、下列 SQL 语句中，能够完成删除功能的是（）。-->**D.DELETETFROM 学生表 WHERE 性别='男'**

260、下列 SQL 语句中，能够完成修改功能的是（）。-->**B.UPDATE 教师表 SET 姓名='王振东'WHERE 学号='053006'**

261、下列 SQL 语句中，实现唯一约束的是（）。-->**A.学号 char (8) NOTNULLCONSTRAINTun_noUNIQUE**

262、下列 SQL 语句中，实现主键约束的是（）。-->**B.学号 char (8) NOTNULLCONSTRAINTPK 学生表 PRIMARYKEY**

263、下列 SQL 语句中，限制字段取值范围的约束是（）。-->**C.性别 char (2) NOTNULLcheck (性别='男'or 性别='女')。**

264、下列（）操作属于数据操作。-->**B.UPDATE**

265、下列（）操作属于数据控制。-->**A.REVOKE**

266、下列（）定义外键约束。-->**D.FOREIGNKEY**

267、下列（）可以删除视图 CSSStudent。-->**B.DROPVIEWCSSStudent**

268、下列（）情况，适于建立索引。-->**A.基本表的某列中记录数量特别多**

269、下列（）情况，一般不建立索引。-->**B.基本表的某列中记录数量特别少**

270、下列（）是指数据库管理系统。-->**D.DBMS**

271、下列（）是指数据库管理系统的英文缩写。-->**D.DBMS**

272、下列（）约束定义列的取值不能重复。-->**C.UNIQUE**

273、下列表达式中，与“年龄 NOTBETWEEN20AND30”等价的是（）。-->**D.年龄 20 OR 年龄 30**

274、下列表达式中，与“所在部门 NOTIN ('教务', '房管’)”等价的是（）。-->**A.所在部门!='教务'AND 所在部门!='房管'**

275、下列表达式中，与“所在部门 NOTIN ('教务','房管’)”等价的是（）。-->**A.所在部门 !='教务' AND 所在部门 !='房管'**

276、下列表达式中，与“所在部门 NOTIN ('财务对后勤’)”等价的是（）。-->**A.所在部门!='财务'AND 所在部门!='后勤'**

277、下列表达式中，与“专业 NOTIN ('网络', '通信’)”等价的是（）。
A.专业 !='网络'AND 专业 !='通信'

278、下列不属于 ADO 对象的是（）。-->**A.OLEDB**

279、下列不属于 VB 中 ADO 对象模型中的对象的是-->**Update**

280、下列不属于事务特性的是（）。-->**B.完整性**

281、下列不属于数据库管理数据特点的是-->**应用程序与数据的物理存储紧密相关**

282、下列不属于数据库管理数据特点的是（）。-->**D.程序紧密依赖于数据**

283、下列不属于数据库管理数据特点的是（）。-->**C.应用程序与数据的物理存储紧密相关**

284、下列不属于数据库管理系统功能的是（）。-->**C.磁盘文件管理功能**

285、下列不属于数据库系统组成部分的是（）。-->**B.操作系统**

286、下列不属于索引类型的选项是（）。-->**D.事务日志索引**

287、下列不属于字符串类型的是（）。-->**D.float**

288、下列不属于组织层数据模型的是（）。-->**D.概念模型**

289、下列操作中，用于更改数指的是（）。-->**D.UPDATE**

290、下列操作中，用于删除数据的是（）。-->**B.DELETE**

291、下列定义视图的语句，正确的是（）。
D.CREATE VIEW V1(Sno,max_g)AS SELECT Sno,MAX(Grade) FROM SC GROUP BY Sno

292、下列定义外键约束的子句，正确的是（）。-->**B.FOREIGN KEY (C1) REFERENCES T(C1)**

293、下列对相应关系数据库中视图的是（）。-->**B.外模式**

294、下列符合第三范式的关系模式是（）。-->**D.教师(教师编号, 教师姓名, 教师性别)。**

295、下列符合第三范式的关系是（）。-->**B.零件(零件编号, 零件名称, 零件重量, 零件储存的仓库编号)。**

296、下列符合第三范式的关系是（）。-->**D.职工(职工号, 职工姓名, 性别, 籍贯, 所属部门编号)。**

297、下列符合第三范式的关系是（）。-->**D.学生(学号, 姓名, 性别)。**

298、下列符合第三范式的关系是（）。-->**D.职工(职工号, 职工姓名, 性别, 籍贯, 所属部门编号)。**

299、下列功能中，不属于数据库管理系统功能的是（）。-->**B.数据库文件读写**

300、下列关系代数运算中，进行列选择的运算是（）。-->**B.投影运算**

301、下列关系中，符合第 2 范式但不符合第 3 范式的是（）。-->**B.学号姓名性别籍贯系号系名系电话**

302、下列关系中，符合第 2 范式但不符合第 3 范式的是（）。-->**B.商品编号**

303、下列关系中，符合第 2 范式但不符合第 3 范式的是（）。-->**B.学号: 0101002; 姓名: 张华; 性别: 男; 专业: 电子; 课程号: A02...**

304、下列关于 DBMS 的说法，错误的是（）。-->**D.用户可以不通过 DBMS 操作数据库文件以提高数据操作效率**

305、下列关于 E-R 模型中联系的说法,正确的是（）。-->**B.联系是描述实体和实体之间的关联关系的**

306、下列关于 SELECT 语句中 HAVING 子句的说法，错误的是（）。-->C.在一个 SLEECT 语句中，HAVING 子句的执行先于 WHERE 子句

307、下列关于 SQLServer 命名实例和默认实例的说法，错误的是（）。-->D.一台计算机上可以有多个默认实例

308、下列关于 SQLServer 日志备份的说法，正确的是（）。-->B.日志备份仅备份数据库的日志内容，不备份数据

309、下列关于 SQLServer 日志文件的说法，错误的是（）。-->D.不同的日志文件必须存储在不同的位置

310、下列关于 SQLServer 数据库服务器登录账户的说法，错误的是（）。-->所有的 Windows 用户都自动是 SQLServer 的合法账户

311、下列关于 SQLServer 数据库文件的说法，错误的是（）。-->D.一个数据库的主要数据文件和日志文件不能存放在相同文件夹中

312、下列关于 SQLServer 数据库组成的说法，正确的是（）。-->A.一个数据库可由多个数据文件和多个日志文件组成

313、下列关于 SQLServer 数据库组成的说法，正确的是（）。-->A.一个数据库可由多个数据文件和多个日志文件组成

314、下列关于 SQLServer 完整备份的说法，错误的是（）。-->D.完整备份之后必须进行一次差异备份

315、下列关于 SQLServer 中扩大数据库空间的说法，正确的是（）。-->C.日志文件和数据文件的空间都可以扩大

316、下列关于 SQLServer 中扩大数据库空间的说法，正确的是（）。-->C.日志文件和数据文件的空间都可以扩大

317、下列关于 SQLServer 主要数据文件的说法，正确的是（）。-->C.主要数据文件的推荐扩展名是.mdf

318、下列关于 TOP 的说法，正确的是（）。-->B.TOP10 返回的结果一定是 10 行数据

319、下列关于差异备份的说法，正确的是（）。-->差异备份备份的是从上次完整备份到当前时间数据库变化的内容

320、下列关于触发器的说法，正确的是（）。-->B.触发器主要用于实现复杂的数据约束

321、下列关于触发器的说法，正确的是（）。-->C.触发器是由数据更改操作引发的自动执行的代码

322、下列关于存储过程优点的说法，正确的是（）。-->允许模块化程序设计

323、下列关于非聚集索引的说法，错误的是（）。-->C.非聚集索引必须先于聚集索引建立

324、下列关于非聚集索引的说法，正确的是（）。-->A.在一个表上可以建立多个非聚集索引

325、下列关于概念层数据模型的说法，正确的是（）。-->A.概念层数据模型要真实地模拟现实世界

326、下列关于关系代数中投影运算的说法，正确的是（）。-->C.投影运算是从列的方向选择关系中的若干列，投影运算后的行数有可能减少

327、下列关于关系代数中选择运算的说法，正确的是（）。-->A.选择运算是从行的方向选择关系中的数据，选择运算后的行数有可能减少

328、下列关于关系代数中选择运算的说法，正确的是（）。-->B.选择运算是从水平方向选取关系中满足条件的记录

329、下列关于关系数据模型的说法，正确的是（）。-->C.关系数据模型采用的是简单二维表结构

330、下列关于关系运算中选择运算的说法，正确的是（）。-->B.选择运算是从水平方向选取表中满足条件的记录

331、下列关于集合并运算的说法，正确的是（）。-->C.进行并运算的两个关系的列个数必须相同，对应列的语义相同

332、下列关于聚集索引的说法，正确的是（）。-->B.在一个表上最多只能建立一个聚集索引

333、下列关于聚集索引与非聚集索引的说法，正确的是（）。-->C.在一个表上可以同时存在聚集索引和非聚集索引

334、下列关于嵌套子查询的说法，正确的是（）。-->B.嵌套子查询是先执行子查询，然后再根据子查询返回结果执行外层查询

335、下列关于日志备份的说法，错误的是-->第一次对数据库进行的备份可以是日志备份

336、下列关于实体联系模型中联系的说法，错误的是（）。-->D.一个联系可以不与任何实体有关

337、下列关于视图的描述不正确的是（）。-->C.视图与数据表是一样的

338、下列关于视图的描述不正确的是（）。-->C.视图与数据表一样要物理存储数据

339、下列关于视图的说法，错误的是（）。-->A.只能在基本表上建立视图，不能在视图上再建立视图

340、下列关于视图的说法，正确的是（）。-->D.视图是面向特定用户的信息需求设计的

341、下列关于视图的说法，正确的是（）。-->B.对视图的操作最终都转换为对基本表的操作

342、下列关于视图的说法，正确的是（）。-->B.视图提供了数据的逻辑独立性

343、下列关于视图优点的说法，错误的是（）。-->C.破坏了数据的一致性

344、下列关于数据库概念结构设计的描述，正确的是（）。-->D.概念结构设计与要使用的数据库管理系统无关

345、下列关于数据库管理系统的说法，错误的是（）。-->数据库管理系统不支持多用户同时访问相同数据库

346、下列关于数据库管理系统的说法，错误的是（）。-->C.数据库应用程序可以不经过数据库管理系统而直接读取数据库文件

347、下列关于数据库结构设计的说法，错误的是（）。-->D.逻辑结构设计是在概念结构设计之前进行的

348、下列关于数据库逻辑结构设计的描述，正确的是（）。-->A.逻辑结构设计是在概念结构设计之后进行的

349、下列关于数据库三级模式中“模式”的说法，错误的是（）。-->D.模式是对整个数据库的底层表示

350、下列关于数据库设计中新奥尔良方法的说法，错误的是（）。-->逻辑结构设计是在需求分析之后、概念结构设计之前进行的

351、下列关于数据库中事务的说法，正确的是（）。-->B.一个事务应该包含的操作是由用户的业务决定的

352、下列关于数据库中事务原子性的说法，正确的是（）。-->B.事务的原子性是指一个事务中的操作要么全做，要么全不做

353、下列关于数据模型的说法，正确的是-->关系数据模型采用的是简单二维表结构

354、下列关于索引的说法，错误的是（）。-->A.索引由索引项组成，索引项只能由表中的一个列组成

355、下列关于索引的说法，正确的是（）。-->C.在一个表上可以建立多个唯一的非聚集索引

356、下列关于索引的说法不正确的是（）。-->B.索引一经建立就需要人工在手动的方式进有维护

357、下列关于索引的说法错误的是（）。-->A.索引一经建立就需要人工进行维护

358、下列关于外键的说法，正确的是（）。-->D.一个表可以有多个外键

359、下列关于用文件管理数据的说法，错误的是（）。-->D.将相关的数据存储在一个文件中，有利于用户对数据进行分类，因此也可以加快用户操作数据的效率

360、下列关于主键的说法，正确的是（）。-->D.主键可以由多个列共同组成

361、下列关于自连接的说法，错误的是（）。-->A.自连接属于外连接的一种，必须为进行自连接的两个表起别名

362、下列聚合函数中，不忽略空值的是（）。-->D.COUNT(*)

363、下列扩展名中，属于 SQLServer 推荐的日志文件扩展名的是（）。-->D.ldf

364、下列类型中，属于定点小数类型的是（）。-->A.numeric

365、下列类型中，属于浮点类型的是（）。-->D.real

366、下列类型中，属于普通编码可变量字符串类型的是（）。-->D.varchar

367、下列模式中，用于表达数据库底层的是（）。-->B.内模式

368、下列哪个系统数据库包含了复制到每个用户数据库中的表（）。-->C.model 数据库

369、下列哪些不属于索引的类型（）。-->D.事务日志索引

370、下列哪种情况比较适于建立索引（）。-->A.基本表的记录数量特别多

371、下列筛选成绩在 80-90 之间（包括边界值）的子句，正确的是（）。C.WHERE 成绩 BETWEEN 80 AND 90

372、下列筛选平均成绩大于等于 80 的子句，正确的是（）
B.HAVING AVG(成绩) >=80

373、下列删除表的语句，正确的是（）。-->A.DROPTABLE

374、下列删除存储过程 P1 的语句，正确的是（）
DROP PROC P1

375、下列声明并创建 Connection 对象的语句，正确的是（）。Dim ADON As New Connection

376、下列声明并创建 Recordset 对象的语句，正确的是（）。Dim ADON As New Recordset

377、下列数据库中，应禁止用户修改删除，并应经常对其进行备份的是（）。-->A.master 数据库

378、下列数据类型中，属于统一编码可变量字符串类型的是（）。-->B.nvarchar

379、下列说法中，（）属于表设计原则。-->A.遵守第三范式标准的数据库设计

380、下列说法中，（）属于字段设计原则。-->B.字段长度尽量充足

381、下列条件子句中，能够筛选出 Col 列中以“a”开始的所有数据的是（）。-->**B.Where Col LIKE 'a%'**

382、下列条件子句中，能够筛选出不姓“张”的是（）。-->**C.Where 姓名 NOT LIKE '张%'**

383、下列条件子句中，能够筛选出不姓“王”的数据的是（）。A.WHERE 姓名 NOT LIKE '王%'

384、下列条件子句中，能够筛选出不姓“张”也不姓“刘”的是（）C.WHERE 姓名 LIKE '['张刘]%

385、下列条件子句中，能够筛选出成绩在 60~80（包括边界值）的是（）。-->**C.Where 成绩 >=60 AND 成绩 <=80**

386、下列条件子句中，能够筛选出价格不在 100~200（不包括边界值）的是（）。-->**A.Where 价格 NOT BETWEEN 100 AND 200**

387、下列条件子句中，能够筛选出价格在 100~200（包括边界值）的是（）。-->**D.Where 价格 >=100 AND 价格 <=200**

388、下列条件子句中，能够筛选出姓名中第二个字是“玲”的是（）。-->**B.Where 姓名 LIKE ' %玲 %'**

389、下列条件子句中，能够选出不姓“王”的数据的是（）。A.WHERE 姓名 NOT LIKE '王%'

390、下列选项中不属于数据库概念设计的是（）。-->**A.回答“干什么”的问题**

391、下列一段代码的功能是（）。
B.修改姓名为“张明”的学生类型为“本科生”

392、下列一段代码的功能是（）。

```
$mysqli = mysqli_connect ('localhost', 'root', '', 'students')
$mysqli->query ("set names utf8");
$sql = 'update tb_stu set type="本科生" where name="张明";
$res = $mysqli->query ($sql);
```


答案：B.修改姓名为“张明”的学生类型为“本科生”

393、下列用于表达关系代数中投影运算的运算符是（）。-->**B.Π**

394、下列用于定义列的取值范围的约束是（）。-->**C.CHECK**

395、下列用于定义主键约束的是（）。-->**C.PRIMARY KEY**

396、下列用于描述数据的物理存储的是（）。-->**B.内模式**

397、下列用于实现修改存储过程的语句是-->**ALTER PROC....**

398、下列用于收回用户权限的语句是（）。-->**C.REVOKE**

399、下列语句运行的结果是（）。-->**C.30ABC**

400、下列语句运行的结果是（）。DECARE@xint=30,@yint=20, @zintSET@z=x+yPrint@z-->**B.50**

401、下列约束中，用于实现参照完整性的是（）。
B.FOREIGN KEY

402、下列约束中用于限制列取值范围的是（）。
C.CHECK

403、下列运算中，属于传统的集合运算的是（）。-->**D.并运算**

404、下列运算中，属于专门的关系运算的是（）。-->**C.集合的连接运算**

405、下列运算属于关系代数中传统的集合运算的是（）。-->**D.并运算**

406、下列运算属于关系代数中专门的关系运算的是（）。-->**C.集合的连接运算**

407、下列在表达式中，与“所在部门 NOTIN ('财务','后勤)’”等价的是（）。-->**A.所在部门 != '财务' AND 所在部门 != '后勤'**

408、下列属于 SQLServer 推荐的日志文件扩展名的是（）。-->**D.ldf**

409、下列属于定点小数类型的是（）。-->**A.numeric**

410、下列属于浮点类型的是（）。-->**D.real**

411、下列属于面向用户信息需求设计的是（）。-->**A.外模式**

412、下列属于普通编码可变长字符串类型的是（）。-->**D.varchar**

413、下列属于数据的动态特征的是（）。-->**B.插入数据**

414、下列属于数据的静态特征的是（）。-->**C.建立约束**

415、下列属于数据定义操作的是-->**CREATE**

416、下列属于数据定义操作的是（）。-->**C.CREATE**

417、下列属于数据访问权限控制操作的是（）。-->**A.REVOKE**

418、下列属于数据更新操作的是-->**UPDATE**

419、下列属于数据更新操作的是（）。-->**B.UPDATE**

420、下列属于数据库逻辑设计内容的是（）。-->**B.将 ER 图转换为特定 DBMS 所支持的数据模型**

421、下列属于数据库逻辑设计内容的是（）。-->**B.将概念设计的结果转换为特定 DBMS 所支持的数据模型**

422、下列属于数据库模式范畴的是（）。-->**B.用户看到的一张数据表**

423、下列属于数据库模式范畴的是（）。-->**C.描述数据库中数据的逻辑结构**

424、下列属于数据库模式前范畴是（）。-->**B.用户看到的一张数据表**

425、下列属于数据库内模式的范畴是（）。-->**A.数据存储在否进行加密**

426、下列属于数据库外模式的范畴是（）。-->**D.不同的用户可以看到不同的数据**

427、下列属于数据库系统组成部分的是（）。-->**B.数据库、数据库管理系统、用户和应用程序**

428、下列属于数据模型三要素的是（）。-->**D.数据结构、数据操作和数据完整性约束**

429、下列属于数据删除操作的是（）。-->**C.DELETE**

430、下列属于数据物理独立性的是（）。-->**A.当内模式发生变化时，保持模式不变**

431、下列属于数值类型的是（）。-->**A.int**

432、下列属于统一编码可变长字符串类型的是（）。-->**B.nvarchar**

433、下列属于整型类型的是（）。-->**B.int**

434、现要定义“工作年限”列的数据类型，假设用整型类型，且其值不会超过 100，则最合适的数据类型是（）。-->**C.tinyint**

435、现要利用 Student 表查询年龄最小的学生姓名和年龄。下列实现此功能的查询语句中，正确的是（）。
SELECT TOP 1 Sname, Sage FROM Student ORDER BY Sage

436、现要统计选课表中 C01 课程的平均成绩，正确的语句是（）。
D.SELECTAVG (成绩) FROM 选课表 WHERE 课程号=' C01'

437、现要统计选课表中 C01 课程的总成绩，正确的语句是（）。-->**C.SELECT SUM(成绩) FROM 选课表 WHERE 课程号 = 'C01'**

438、现要统计选课表中 C01 课程的最低成绩，正确的语句是（）。-->**C.SELECT MIN(成绩) FROM 选课表 WHERE 课程号 = 'C01'**

439、现要统计选课表中 C01 课程的最高成绩，正确的语句是（）。-->**A.SELECT MAX(成绩) FROM 选课表 WHERE 课程号 = 'C01'**

440、现要统计选课表中 COI 课程的总成绩，正确的语句是（）。
C.SELECTSUM(成绩 FROM 选课表 WHERE 课程号='COI'

441、现有学生表和选课表，其结构为：学生（学号，姓名，入学日期，毕业日期）选课（学号，课程号，考试日期，成绩）现要求选课表中的考试日期必须在学生表中相应学生的入学日期和毕业日期之间。下列实现方法中，正确的是（）。-->**B.在选课表上建立一个插入和更新操作的触发器**

442、修改存储过程 P1 的语句，正确的是（）。
B.ALTER PROC P1

443、选择与投影运算，属于关系模型中（）的功能。-->**B.关系代数**

444、选择与投影运算，属于关系模型中（）的功能。系代数-->**D.数据独立性**

445、一般不需要建立索引的情况是（）。-->**B.基本表的某列中记录数量特别少**

446、一个病房只有一个负责人，一个人只能担任一个病房的负责人，则病房与负责人之间的联系是（）。-->**B.一对一**

447、一个关系中每条记录和第二个关系的每条记录进行连接，所组成的新关系称为（）。-->**C.两个关系的积**

448、一个商品表有商品编号、商品名称、商品单价、库存数量共 4 个属性，下列哪条语句能够建立一个视图，该视图包含单价高于 1 万元的的商品的商品编号和商品名称（）。
-->**C.CREATEVIEWCSSStudentASSELECT 商品编号, 商品名称 FROM 商品表 WHERE 商品单价 > 10000**

449、一个系有多名学生，一个学生只能属于一个系，则系和学生之间的联系是（）。-->**A.一对一**

450、一个学生表有学生学号、学生姓名、年龄、系共 4 个属性，下列语句（）能够建立一个视图，该视图包含计算机系学生的学号和姓名。-->**C.CREATEVIEWCSSStudentASSELECT 学生学号, 学生姓名 FROM 学生表 WHERE 系='计算机系'**

451、一个银行营业所可以有多个客户，一个客户也可以在多个营业所进行存取款业务，则客户和银行营业所之间的联系是（）。-->**D.多对多**

452、一个用户访问 SQLServer 通常要依次经过 4 个层次的安全控制，分别是（）。-->**A.操作系统、服务器、数据库、表和列**

453、一名教师可以讲授多门课程，一门课程可以被多个教师讲授，则教师与课程之间的联系是（）。-->**D.多对多**

454、一名教师可以教多门课程，一门课程可以被多个教师讲授，这是（）联系。
D.多对多

455、已知关系 R1（课程号，课程名，学分，系号），关系 R2（系号，系名，系主任，办公电话），系号在 R2 中为主键，在 R1 中为外键。系号的值包括 202,203,204,下面记录不应出现的是（）。-->**B.R1 (1021, 材料力学, 5,208)。**

456、以二维表的形式表示关系之间的联系，属于关系模型中（）的功能。-->**A.数据结构**

457、以下不属于 SQLServer 数据库特点的是（）。-->**C.基于面向对象的数据模型**

458、以下关于 TOP 的说法，说法正确的是（）。-->**A.TOP 10 返回的结果只能是前 10 行数据**

459、用户使用 SQLServer 时, 通常需要依次经过两个安全性阶段 ()。-->**C. 身份验证、权限认证**

460、用于表达关系代数中投影运算的运算符是 ()。
b. []

461、用于定义主键约束的是 ()。-->**C.PRIMARYKEY**

462、用于实现实体完整性约束的是 ()。
a. PRIMARY KEY

463、用于限制列的取值不能重复的约束是 ()。-->**D.UNIQUE**

464、用于限制列的取值范围的约束是 ()。-->**BCHECK**

465、有关系表:选课(学号,课号,成绩),下列统计每个学生的选课门数的语句,正确的 ()。
A.SELECT 学号, COUNT(*)FROM 选课 GROUP BY 学号

466、有关系表:选课(学号,课程号,成绩),下列聚合函数中,统计最高成绩的是 ()。-->**A.MAX(成绩)**

467、有关系表:选课(学号,课程号,成绩),下列统计每个学生的选课门数的语句,正确的是 ()。
A.SELECT 学号, COUNT(*) FROM 选课 GROUP BY 学号

468、有关系模式:写书(书号,作者号,作者名,完成日期),主键为(书号,作者号)。此关系模式至少属于 ()。-->**A.第一范式**

469、有关系模式:写书(书号,作者号,作者名,完成日期),主键为(书号,作者号)。此关系模式至少属于 ()。-->**A.第一范式**

470、有关系模式:选课(学号,姓名,课程号,成绩),主键为(学号,课程号)。此关系模式至少属于 ()。-->**A.第一范式**

471、有关系模式:学生(学号,姓名,身份证号,性别,所在系),下列适合做候选键的是 ()。-->**C.身份证号、学号**

472、有关系模式:学生(学号,姓名,所在系,系主任),设一个系只有一个系主任,则该关系模式至少属于 ()。-->**B.第二范式**

473、有关系模式选课(学号,姓名,课程号,成绩),主键为(学号,课程号)。此关系模式至少属于 ()。
A.第一范式

474、有三个表,供应商表(供应商编号,供应商名称,供应商所在城市),零件表(零件编号,零件名称,零件颜色,零件重量),工程表(工程编号,工程名称,工程所在城市),供应关系表(供应商编号,零件编号,工程编号,零件数量),供应关系表用于处理为指定的工程提供零件的供应商编号、零件编号等信息。下列查询的含义是 ()。SELECT 供应商编号 FROM 供应关系表 WHERE 工程编号='K101'-->**B.查找为 K101 工程提供零件的供应商编号**

475、有严格的数学基础并在此基础上发展起来的数据库模型是 ()。-->**C.关系模型**

476、有严格的数学基础的数据库模型是 ()。-->**A.关系模型**

477、在 E-R 图中,实体使用 () 图形表示。-->**D.矩形框**

478、在 E-R 图中,实体用 () 描述。-->**C.矩形框**

479、在 E-R 图中,下列图形中用于描述联系的是 ()。-->**B.菱形框**

480、在 E-R 图中,下列图形中用于描述实体的是 ()。-->**C.矩形框**

481、在 E-R 图中,下列图形中用于描述属性的是 ()。-->**A.圆角矩形**

482、在 E/R 图中,联系用 () 描述。-->**B.菱形框**

483、在 E/R 图中,属性用 () 描述。-->**圆角矩形**

484、在 E 也 R 图中,实体用 () 描述。
C.矩形框

485、在 SQL 语句中,用于更新表数据的语句是 ()。-->**UPDATE**

486、在 VB 中,可以利用 Connection 对象的 Execute 方法实现对数据库的更改操作。关于删除 SC 表中学号为 S01 的选课记录的语句,正确的是 ()。
B. ADOcn.Execute "DELETE FROM SC WHERE 学号 = 'S01'"

487、在 VB 中,可以利用 Connection 对象的 Execute 方法实现对数据库的更改操作。下列删除 SC 表中学号为 S01 的选课记录的语句,正确的是 ()。-->**B.ADOcn.Execute"DELETEFROMSCWHERE 学号='S01'"**

488、在 VB 中可以利用 Connection 对象的 Execute 方法实现对数据库的更改操作。下列将 SC 表中学号为 S01 学生的成绩增加 10 分的语句,正确的是 ()。
-->**C.ADOcn.Execute"UPDATESCSETGrade=Grade+10WHERE 学号='S01'"**

489、在表达式中,与“所在部门 NOTIN ('财务','后勤)“等价的 ()。-->**A.所在部门!=财务'AND 所在部门!=后勤'**

490、在成绩数据表中,要计算某课程的平均成绩,可采用 ()。-->**D.Avg 函数**

491、在成绩数据表中,要计算选择某课程的学生人数总和,可采用 ()。-->**C.Count 函数**

492、在第一个关系中出现,同时在第二个关系也出现的记录组成的新关系称为 ()。-->**C.两个关系的交**

493、在关系数据库中,将 2NF 的关系模式规范化到 3NF,通常采用的方法是 ()。-->**C.分解关系模式 D.合并关系模式**

494、在关系数据库中,解决数据冗余和操作异常通常采用的方法是 ()。-->**分解关系模式**

495、在关系数据库中,解决一个表中的数据冗余和操作异常通常采用的方法是 ()。-->**C.分解关系模式**

496、在简单恢复模式下,可以进行的备份是 ()。-->**C.仅完整备份和差异备份**

497、在将 E-R 图转换为关系模式时,一般将 m:n 联系转换成一个独立的数据库。下列关于这种联系产生的数据库的主键的说法,正确的是 ()。-->**C.至少包含 m 端和 n 端关系模式的主键**

498、在将局部 E-R 图合并为全局 E-R 图时,可能会产生一些冲突。下列冲突中不属于合并 E-R 图冲突的是 ()。-->**B.语法冲突**

499、在教师关系中,能够成为主关键字(或称主码)的属性是 ()。-->**B.教师编号**

500、在生产商关系中,能够成为主关键字(或称主码)的属性是 ()。-->**B.生产商编号**

501、在视图的定义语句中,只能包含 ()。-->**A.数据查询语句**

502、在数据库管理技术发展过程的几个阶段中,数据独立性最高的阶段是 ()。-->**D.数据库管理阶段**

503、在数据库的三级模式中,描述数据库中全体数据的逻辑结构和特征的是 ()。-->**B.模式**

504、在数据库管理系统的三级封锁协议中,一级封锁协议能够解决的问题是 ()。-->**A.不丢失修改**

505、在数据库三级模式结构中,用户所看到的数据视图是 ()。-->**A.外模式**

506、在数据库设计中,表设计原则遵守 () 标准。-->**第三范式**

507、在数据库设计中,将 E-R 图转换为关系数据模型是 () 完成的工作。-->**逻辑设计阶段**

508、在数据库设计中,将 E-R 图转换为关系数据模型是下述哪个阶段完成的工作 ()。-->**C.逻辑设计阶段**

509、在数据库设计中,进行用户子模式设计是 () 要完成的工作。-->**逻辑结构设计阶段**

510、在数据库设计中,进行用户子模式设计是下述哪个阶段要完成的工作 ()。-->**C.逻辑结构设计阶段**

511、在条件子句中,能够筛选出 Col 列中以“a”开始的所有数据的是 ()
Where Col LIKE 'a%'

512、在条件子句中,能够筛选出价格在 100~200 (包括边界值) 的是 ()。
D.Where 价格 >=100AND 价格 <=200

513、在图书借阅关系中,能够成为主关键字(或称主码)的属性是 ()。-->**C.图书编号**

514、在物理层面,SQLServer 数据库是由数据文件和事务日志文件两个操作系统文件组成的,它们的后缀分别是 ()。-->**A.MDF 和 LDF**

515、在系统数据库中, () 包含了复制到每个用户数据库中的表。-->**C.model 数据库**

516、在系统数据库中, () 包含了系统级信息。-->**A.master 数据库**

517、在系统数据库中,包含了系统级信息的数据库是 ()。-->**A.master 数据库**

518、在销售商品关系中,能够成为主关键字(或称主码)的属性是 ()。-->**B.商品编号**

519、在一个课程关系中,最可能成为主关键字(或称主码)的属性是 ()。-->**C.课程编号**

520、在一个物资台账数据表中,能够成为主关键字(或称主码)的字段是 ()。-->**B.物资编码**

521、在一个学生关系中,能够成为主关键字(或称主码)的属性是 ()。-->**C.学号**

522、在用数据模型描述数据时,一般要求数据模型要满足三个要求。不属于数据模型应满足要求的是 ()。-->**A.能够描述并发数据**

523、在用数据模型描述数据时,一般要求数据模型要满足三个要求。下列不属于数据模型应满足要求的是 ()。-->**A.能够描述并发数据**

524、在用数据模型描述数据时,一般要求数据模型要满足三个要求。下列描述中,不属于数据模型应满足的要求的是 ()。-->**A.能够描述并发数据**

525、针对 PHP 语句“\$mysqli==mysqli->connect('localhost','root','@abc','lib_user');”下列解释错误的是 ()。
B.以 lib_user 账户登陆

- 526、属于 SQLServer 推荐的日志文件扩展名的是 ()。-->D.ldf
- 527、属于 SQLServer 最核心服务的是 ()。
数据库引擎服务
- 528、属于 SQLServer 最核心服务的是 ()。-->A 数据库引擎服务
- 529、属于传统的集合运算的是 ()。-->D 并运算
- 530、属于浮点类型的是 ()。-->D.real
- 531、属于关系模型中关系操作的范畴是 ()。-->A.关系代数中的并运算
- 532、属于普通编码可变量字符串类型的是 ()。-->D.varchar
- 533、属于日期时间型的数据类型是 ()。-->C.Datetime
- 534、属于数据的动态特征的是 ()。-->B.插入数据
- 535、属于数据定义操作的是 ()。
C.CREATE
- 536、属于数据控制操作的是 ()。-->A.REVOKE
- 537、属于数据库逻辑设计内容的是 ()。-->将概念设计的结果转换为特定 DBMS 所支持的数据模型
- 538、属于数据库系统组成部分的是 ()。-->B.数据库、数据库管理系统、用户和应用程序

简答(52)--

- 1、SQLServer 的安全验证过程是什么? ...
- 2、SQLServer 数据库由几类文件组成? 这些文件的...
- 3、SQLServer 最核心的服务是什么?
- 4、Transact-SQL 支持的主要数据类型有哪些? ...
- 5、WHEREAgeBETWEEN20AND30 子句, 查找的 Age 范围...
- 6、“WHEREsdeptNOTIN('CS','IS','MA')”子句, ...
- 7、把 E-R 模型转换为关系模式的转换规则有哪些? ...
- 8、触发器的作用是什么? 前触发和后触发的主要区...
- 9、存储过程的作用是什么? 为什么利用存储过程可...
- 10、登录账号和用户账号的含义是什么? 它们有什么...
- 11、关系数据库的三个完整性约束是什么? 各是什么...
- 12、关系数据模型的优化包含哪些方法? ...
- 13、简述 SQLServer 数据库的存储结构。...
- 14、简述 SQL 语言的功能, 每个功能的作用是什么? ...
- 15、简述数据库设计分为哪几个主要的阶段? ...
- 16、简述数据库设计中的字段设计原则的主要内容...
- 17、简述数据库设计中物理设计阶段的主要内容有...
- 18、启动服务使用的是 SQLServer 的哪个工具? ...
- 19、请简单说明数据库系统的组成。...
- 20、请简述 C/S 和 B/S 的主要原理, 并说明它们各自的...
- 21、请简要说明 SQLServer 提供的主要备份种类, 每...
- 22、请简要说明 SQLServer 中使用视图的优点? ...
- 23、请简要说明关系规范化的必要性。...
- 24、请简要说明开发数据库应用系统的一般步骤。...
- 25、请简要说明视图的作用。...
- 26、请简要说明数据库逻辑设计所包含的工作。...
- 27、请简要说明数据库应用系统设计的步骤。...
- 28、请需要说明 SQLServer 时中使用存储过程的优...
- 29、权限的管理包含哪些操作?
- 30、日志备份备份的是哪段时间的哪些内容? 差异备...
- 31、若对某数据库进行了完整备份、差异备份和日...

- 32、设有关系模式: 学生选课 (学号, 姓名, 所在系, 性别, 课...
- 33、什么是数据和数据处理? 数据与信息的关系联系...
- 34、什么是数据库的逻辑结构设计? 简述其设计步骤...
- 35、事务日志文件有什么用途? 是否也需要备份? 为什...
- 36、试说明使用视图的好处。
- 37、试说明事务的概念及四个特征。...
- 38、数据独立性指的是什么? 它能带来哪些好处? ...
- 39、数据独立性指的是什么? 它能带来哪些好处? ...
- 40、数据库设计的目的是什么, 请简要说明。...
- 41、数据库设计分为哪几个阶段? 每个阶段的主要工...
- 42、数据库设计分为哪几个阶段? 每个阶段的主要工...
- 43、数据库设计有什么原则, 用这些原则有什么好处...
- 44、说明实体-联系模型中的实体、属性和联系的...
- 45、索引的作用是什么?
- 46、索引分为哪几种类型? 分别是什么? 它们的主要区...
- 47、外连接与内连接的主要区别是什么? ...
- 48、为什么要进行规范化? 规范化的相关理论是什么...
- 49、相关子查询与嵌套子查询在执行方面的主要区...
- 50、需求分析阶段的任务是什么? 其中发现事实的方...
- 51、与文件管理相比, 用数据库管理数据有哪些优点...
- 52、自连接与普通内连接的主要区别是什么? ...

1、SQLServer 的安全验证过程是什么?

答: 在 SQLServer 中, 用户要访问数据库中的数据, 必须经过三个认证过程。第一个是身份认证, 这通过登录账户 (SQLServer 称之为登录名) 来标识用户, 身份认证只验证用户连接到 SQLServer 数据库服务器的资格, 即验证该用户是否具有连接到数据库服务器的“连接权”; 第二个是访问权限认证, 当用户访问某数据库时, 必须具有该数据库的访问权, 即验证用户是否是该数据库的合法用户。第三个是操作权限认证, 当用户访问数据库中的数据或对象时, 必须具有合适的操作权限。

2、SQLServer 数据库由几类文件组成? 这些文件的推荐扩展名分别是什么?

答: 由 2 类文件组成: 数据文件和日志文件。主要数据文件的推荐扩展名为 mdf, 次要数据文件的推荐扩展名为 ndf, 日志文件的推荐扩展名为 ldf。

3、SQLServer 最核心的服务是什么?

答: 数据库引擎服务。

4、Transact-SQL 支持的主要数据类型有哪些?

答: 数值类型 (包括准确数值类型和浮点数值类型)、字符串类型 (包括统一字符编码的字符串类型和普通字符编码的字符串类型)、日期时间类型以及二进制数据类型。

5、WHEREAgeBETWEEN20AND30 子句, 查找的 Age 范围是多少?

答: Age 大于等于 20 并且小于等于 30

6、“WHEREsdeptNOTIN('CS','IS','MA')”子句, 查找的数据是什么?

答: 查找'CS','IS','MA'三个系之外的其他系。

7、把 E-R 模型转换为关系模式的转换规则有哪些?

答: 1) 一个实体转换为一个关系模式。实体的属性就是关系模式的属性, 实体的码就是关系模式的主键。

2) 对于实体间的联系有以下不同的情况:

①一个 1:1 联系通常是与任意一端所对应的关系模式合并, 并将在该关系模式的属性中加入另一个实体的码和联系本身的属性。

②一个 1:n 联系通常是与 n 端所对应的关系模式合并, 并将在该关系模式中加入 1 端实体的码以及联系本身的属性。

③一个 m:n 联系必须转换为一个独立的关系模式。与该联系相连的各实体的码以及联系本身的属性均转换为该关系模式的属性, 且关系模式的主键包含各实体的码。

3) 具有相同主键的关系模式可以合并。

8、触发器的作用是什么? 前触发和后触发的主要区别是什么?

答: 触发器可以实现复杂的完整性约束, 同时还可以实现一些商业规则。

前触发器并不真正执行引发触发器执行才数据操作语句; 后触发器是在引发触发器执行的数据操作语句执行完后, 再引发触发器执行。

9、存储过程的作用是什么? 为什么利用存储过程可以提高数据的操作效率?

答: 存储过程可以提供模块化程序设计, 提高数据操作效率, 简化客户端编程。

因为数据库管理系统是在创建存储过程时对 SQL 代码进行分析和优化, 并在第一次执行时进行语法检查和编译, 将编译好的可执行代码存储在内存的一个专门缓冲区中, 以后再执行此存储过程时, 只需直接执行内存中的可执行代码即可。

10、登录账号和用户账号的含义是什么? 它们有什么区别联系?

答案: 参考答案及评分要点: 登录账号用于身份验证, 使得用户到达登录到 SQLServer 服务器, 属于服务器层面, 相当于大门的钥匙; 用户账号用于权限验证, 属于数据库层面, 相当于自己房间的钥匙。用户账号在定义时便与一个登录账号相关联, 即任何一个用户账号背后都有一个登录账号。

11、关系数据库的三个完整性约束是什么? 各是什么含义?

答: 实体完整性、参照完整性和用户定义的完整性。实体完整性指关系数据库中所有的表都必须有主键。参照完整性用于描述和约束实体之间的关联关系。用户定义的完整性是针对某一具体应用领域定义的数据约束条件, 它反映某一具体应用所涉及的数据必须满足应用语义的要求。

12、关系数据模型的优化包含哪些方法?

答: 关系数据模型的优化通常以关系规范化理论为指导, 并考虑系统的性能。具体方法为:

1) 确定各属性间的函数依赖关系。根据需求分析阶段得出的语义, 分别写出每个关系模式的各属性之间的函数依赖以及不同关系模式中各属性之间的数据依赖关系。

2) 对各个关系模式之间的数据依赖进行极小化处理, 消除冗余的联系。

3) 判断每个关系模式的范式, 根据实际需要确定最合适的范式。

4) 根据需求分析阶段得到的处理要求, 分析这些模式对于这样的应用环境是否合适, 确定是否要对某些模式进行分解或合并。

13、简述 SQLServer 数据库的存储结构。

答案: SQLServer 数据库的存储结构包括物理存储结构和逻辑存储结构两个层面。在物理层面上, SQLServer 数据库是由两个操作系统文件组成的, 即数据文件和事务日志文件, 它们的后缀分别是 MDF 和 LDF。数据文件主要用于存储数据, 事务日志文件主要用于存储数据库的更新情况等事务日志信息。在逻辑层次上,

数据库是由表、视图、存储过程等一系列数据对象组成的。当采用 SQLServer 企业管理器将上述数据库文件“打开”后，就能看到数据库的逻辑结构了。数据库中的数据分别存储在几个不同的对象中，而这些对象是用户在操作数据库时，实际能够看到和接触到的，属于逻辑存储结构。常用的数据库对象主要包括：表 (Table)、索引 (Index)、视图 (View)、触发器 (Triggers)、存储过程 (StoreProcedure)、缺省 (Default)、约束 (Constraint)、用户 (User) 以及图表 (Diagram) 和用户自定义的数据类型等。

14、简述 SQL 语言的功能，每个功能的作用是什么？

答：SQL 按其功能可分为四大部分：数据定义、数据控制、数据查询和数据操作。数据定义功能用于定义、删除和修改数据库中的对象；数据查询功能用于实现查询数据的功能，查询数据是数据库中使用最多的操作；数据操作功能用于增加、删除和修改数据；数据控制功能用于控制用户对数据库的操作权限。

15、简述数据库设计分为哪几个主要的阶段？

答案：数据库设计分为以下阶段：需求分析、概念设计、逻辑设计、物理设计、数据库实施和运行维护。

16、简述数据库设计中的字段设计原则的主要内容？

答案：首先，长度应尽量充足。其次，小心保留词。最后，慎用使用触发器。

17、简述数据库设计中物理设计阶段的主要内容有哪些？

答案：存储记录的格式设计。存储方法设计。访问方法设计。完整性和安全性考虑。

18、启动服务使用的是 SQLServer 的哪个工具？

答：SQLServer 配置管理器。

19、请简单说明数据库系统的组成。

答案：广义上讲，数据库系统由数据库、数据库管理系统（及其开发工具）、应用系统和用户组成，其中用户可分为数据库管理员、应用程序开发人员和最终用户三类。

20、请简述 C/S 和 B/S 的主要原理，并说明它们各自的优缺点。

答：传统的 C/S 模式是一种两层结构的系统，第一层是在客户机系统上结合了表示与业务逻辑；第二层是通过网络结合了数据库服务器。C/S 模式主要由客户应用程序和数据库两部分组成。C/S 结构软件（即客户机/服务器模式）分为客户机和服务器两层。B/S 模式，即浏览器/服务器模式，是一种从传统的二层 C/S 模式发展起来的新的网络结构模式，其本质是三层结构 C/S 模式。C/S 和 B/S 的优缺点。

(1) 系统的性能

在系统的性能方面，B/S 占有优势的是其异地浏览和信息采集的灵活性。

采用 C/S 结构时，客户端和服务端都能够处理任务，这虽然对客户机的要求较高，但因此可以减轻服务器的压力。

(2) 系统的开发

C/S 结构是建立在中间件产品基础之上的，对应用开发者提出了较高的要求。应用程序的维护、移植和互操作变得复杂。如果客户端是在不同的操作系统上，C/S 结构的软件需要开发不同版本的客户端软件。

与 B/S 结构相比，C/S 技术发展历史更为“悠久”。从技术成熟度及软件设计、开发人员的掌握水平来看，C/S 技术应是更成熟、更可靠的。

(3) 系统的升级维护

C/S 系统升级成本比较大。B/S 与 C/S 处理模式相比，减轻了异地用户系统维护与升级的成本。所有的升级操作只需要针对服务器进行。

21、请简要说明 SQLServer 提供的主要备份种类，每种类型的备份所备份的主要内容是什么？

答案：三种备份类型：完整备份；或完整数据库备份、差异备份（或差异数据库备份）和事务日志备份。完整备份：备份数据库的全部内容。差异备份：备份自最近的完整备份之后变化的内容。事务日志备份：只备份自上次备份后到当前备份时间新增的日志内容。

22、请简要说明 SQLServer 中使用视图的优点？

答案：视图是经过预编译的 SELECT 语句，存储在数据库服务器端，因此执行视图比从客户端直接执行 SELECT 语句速度更快、效率更高一些。视图属于用户模式范畴，在实际中一般的用户不一定具有 SELECT 语句方面的专门知识，从用户友好性角度来说，视图更便于用户使用。利用视图可以简化的形式表达复杂的 SELECT 语句组，如嵌套查询等。

23、请简要说明关系规范化的必要性。

答案：关系规范化能够减少数据冗余、更新异常、插入异常、删除异常

24、请简要说明开发数据库应用系统的一般步骤。

答：1). 通常要声明一个全局数据库连接对象
2). 一般要创建 Main 子过程，并将它设置为启动对象。在其中顺序完成以下操作：

(1) 以非模态方式显示飞溅窗。
(2) 创建全局数据库连接对象，并连接 SQLServer 或 Access 等数据库。

(3) 以模态方式显示登录窗，等待用户登录。
(4) 登录成功后，以非模态方式显示主窗体，否则退出程序。

3). 主窗体一般均提供增加、删除、修改、查询等对数据库的管理功能，所有操作均使用全局数据库连接对象作为与数据库交互的通道，通过 SQL 语句对数据库进行处理，其中：

(1) 增加功能。使用 Insert 语句。在新增一条记录时，一般应先执行一 Select 语句，以防止记录的主关键词重复。

(2) 删除功能。使用 Delete 语句。在执行删除操作之前，一般会使用 MsgBox 函数进行提示。

(3) 修改功能。使用 Update 语句。

(4) 查询功能。使用 Select 语句。一般会根据用户的需求设置查询条件。

4). 其他功能。比如数据库备份和恢复功能、用户权限管理功能等。

25、请简要说明视图的作用。

答案：一是简化数据查询语句；二是使用户能从多角度看待同一数据；三是提高对数据的安全性；四是提供对一定程度的逻辑独立性。

26、请简要说明数据库逻辑设计所包含的工作。

答案：一是将概念结构转换为关系数据模型。二是对关系数据模型进行优化。三是设计面向用户的外模式。

27、请简要说明数据库应用系统设计的步骤。

答案：数据库应用系统设计的步骤包括需求分析、概念设计、逻辑设计和物理设计。

28、请简要说明 SQLServer 时中使用存储过程的优点？

答案：执行速度快；模块化的程序设计；减少网络流量；保重系统的安全性。

29、权限的管理包含哪些操作？

答：权限的管理包括授予权限、收回权限和拒绝访问三种。

30、日志备份备份的是哪段时间的哪些内容？差异备份备份的是哪段时间的哪些内容？

答：日志备份备份的是从上次备份到当前时刻新增的数据库日志内容。

差异备份备份的是从上次完整备份到当前时刻数据库的变化部分。

31、若对某数据库进行了完整备份、差异备份和日志备份，则当恢复该数据库时，正确的恢复顺序是什么？

答：要求先恢复完整备份，然后恢复完整备份之后最近的差异备份，最后再按备份顺序逐个恢复差异备份后的全部日志备份。

32、设有关系模式：学生选课（学号，姓名，所在系，性别，课程号，课程名，学分，成绩）。

设一个学生可选多门课程，一门课程可以被多个学生选。每个学生由学号唯一标识，一个学生只在一个系学习；每门课程由课程号唯一标识。每个学生选的每门课程有唯一的成绩。

(1) 请指出此关系模式的候选键。

(2) 写出该关系模式的函数依赖集。

(3) 该关系模式属于第几范式？请简单说明理由。

(4) 若不是第三范式的，请将其规范化为第三范式关系模式，并指出分解后每个关系模式的主键和外键。

(本题 3 分)

答：(1) 候选键：(学号，课程号)

(2) 学号→姓名，学号→所在系，学号→性别，课程号→课程名，课程号→学分，

(学号，课程号)→成绩

(3) 属于第一范式，因为存在部分函数依赖：学号→姓名。

(4) 第三范式关系模式：

学生(学号，姓名，所在系，性别)

课程(课程号，课程名，学分)

考试(学号，课程号，成绩)，学号为引用学生的外键，课程号为引用课程的外键。

33、什么是数据库和数据处理的？数据与信息区别联系是什么？

答案：数据 (Data) 是一组表示数量、行动和目标的非随机的可鉴别的符号。数据处理通常包括查找、统计、分类、修改、计算、汇总等各种方法。信息 (Information) 是经过加工的，能对接收者的行为和决策产生影响的、有价值的信息。如果将数据比喻为原料，那么信息就是数据经过加工而生产的产品，是有价值的信息。与原料和产品的概念相似，一个系统的产品可能是另一个系统的原料，那么一个系统的信息可能成为另一个系统的数据。

34、什么是数据库的逻辑结构设计？简述其设计步骤。

答：数据库逻辑设计的任务是把概念设计阶段产生的 E-R 图转换为具体 DBMS 支持的组织层数据模型，这些模型在功能、性能、完整性和一致性约束方面满足应用要求。

关系数据库的逻辑设计一般包含三项工作：

- 1) 将概念结构转换为关系数据模型。
- 2) 对关系数据模型进行优化。
- 3) 设计面向用户的外模式。

35、事务日志文件有什么用途？是否也需要备份？为什么？

答案：SQLServer 数据库是由两个操作系统文件组成的，即数据文件和事务日志文件。事务日志文件主要用于存储数据库的更新情况等事务日志信息。事务日志文件非常重要，所有对数据库的更改操作都会记录于此，当数据库损坏时，数据库管理员可使用事务日志恢复数据库。从事务日志文件的作用上可以看出，它是非常有用和重要的，因此也需要备份。

36、试说明使用视图的好处。

答：利用视图可以简化客户端的数据查询语句，使用户能从多角度看待同一数据，可以提高数据的安全性，视图对应数据库三级模式中的外模式，因此提供了一定程度的逻辑独立性。

37、试说明事务的概念及四个特征。

答：事务是用户定义的数据操作系列，这些操作作为一个完整的工作单元执行。一个事务内的所有语句作为一个整体，要么全部执行，要么全部不执行。

事务的特征包括：原子性、一致性、隔离性和持久性。

38、数据独立性指的是什么？它能带来哪些好处？

答：数据独立性包括逻辑独立性和物理独立性两部分。物理独立性是指当数据的存储结构发生变化时，不影响数据库模式；逻辑独立性是指当模式发生变化时，不影响外模式，从而不影响应用程序的特性。这两个独立性使用户只需关心逻辑层即可，同时增强了应用程序的可维护性。

39、数据独立性指的是什么？它能带来哪些好处？

答：数据独立性包括逻辑独立性和物理独立性两部分。物理独立性是指当数据的存储结构发生变化时，不影响数据库模式；逻辑独立性是指当模式发生变化时，不影响外模式，从而不影响应用程序的特性。这两个独立性使用户只需关心逻辑层即可，同时增强了应用程序的可维护性。

40、数据库设计的目的是什么，请简要说明。

答案：数据库设计指的是设计数据库及其应用系统的技术，是指对于一个给定的应用环境，包括硬件环境和操作系统、数据库管理系统（DBMS）等软件环境，如何使用这个环境来表达用户的要求，构造最优的数据库模式，建立数据库及围绕数据库展开的应用系统，使之能够有效地收集、存储、操作和管理数据，满足企业组织中各类用户的应用需求（信息需求和处理需求）。

41、数据库设计分为哪几个阶段？每个阶段的主要工作是什么？

答案：数据库设计一般包含如下过程：一是需求分析。对解用户系统需求，解决“干什么”的问题。二是结构设计，包括概念设计、逻辑设计和物理设计。概念设计的目标是产生反映全体用户信息需求的整体数据库概念结构；数据库逻辑设计的任务是把概念设计阶段产生的 E-R 图转换为具体 DBMS 支持的组织层数据模型，也就是转换为 DBMS 可以处理的数据库逻辑结构；物理设计的任务和目标是为逻辑数据模型选取一个最适合应用环境的物理结构。三是数据库实施，主要任务和目标就是在实际的计算机系统中建立数据库应用系统。四是数据库运行和维护阶段。主要工作包括：数据库的备份和恢复、数据库的安全性和完整性控制、监视、分析、调整数据库性能、数据库的重组。

42、数据库设计分为哪几个阶段？每个阶段的主要工作是什么？

答：数据库设计一般包含如下过程：

- 1) 需求分析。了解用户系统需求，解决“干什么”的问题？
- 2) 结构设计，包括概念设计、逻辑设计和物理设计。概念设计的目标是产生反映全体用户信息需求的整体数据库概念结构；数据库逻辑设计的任务是把概念设计阶段产生的 E-R 图转换为具体 DBMS 支持的组织层数据模型，也就是转换为 DBMS 可以处理的数据库逻辑结构；物理设计的任务和目标是为逻辑数据模型选取一个最适合应用环境的物理结构。
- 3) 数据库实施，主要任务和目标就是在实际的计算机系统中建立数据库应用系统。
- 4) 数据库运行和维护阶段。主要工作包括：数据库的备份和恢复、数据库的安全性和完整性控制、监视、分析、调整数据库性能、数据库的重组。

43、数据库设计有什么原则，用这些原则有什么好处？

答案：数据库设计的原则包括表设计原则、字段设计原则和其他原则。表设计原则的核心内容是遵守第三范式标准。字段设计原则包括字段长度应尽量充足，小心保留字的使用，慎用触发器，其他原则主要指适当使用视图。

44、说明实体-联系模型中的实体、属性和联系的概念。

答：实体是具有公共性质的并可相互区分的现实世界对象的集合。属性是实体所具有的特征或性质。联系是实体之间的关联关系。

45、索引的作用是什么？

答：索引可以加快数据的查询效率。

46、索引分为哪几种类型？分别是什么？它们的主要区别是什么？

答：分为聚集索引和非聚集索引两种。聚集索引会对数据进行物理排序，非聚集索引不对数据进行物理排序。

47、外连接与内连接的主要区别是什么？

答：进行外连接操作时，其连接结果中将包含一张表中的所有数据，不管该数据是否满足连接条件；而进行内连接操作时，其连接结果必须是两个表中都满足连接条件的数据。

48、为什么要进行规范化？规范化的相关理论是什么？请简要说明。

答案：参考答案及评分要点：为确保数据库设计的正确性，在数据库逻辑设计阶段，常使用关系规范化理论来指导关系型数据库的设计。其基本思想是，每个关系都应满足一定的规范，才能使关系模式设计合理，达到减少冗余，提高查询效率的目的。一个好的数据库关系，应该没有数据冗余、更新异常、插入异常、删除异常问题的存在，这就必须进行规范化处理。规范化使用的理论是函数依赖。

49、相关子查询与嵌套子查询在执行方面的主要区别是什么？

答：相关子查询的执行过程是先外后内，而嵌套子查询的执行过程是先内后外。而且相关子查询中必须有与外层查询的关联，而嵌套子查询中，内、外层查询不需要有关联关系。

50、需求分析阶段的任务是什么？其中发现事实的方法有哪些？

答：需求分析阶段的主要工作就是，通过调查从用户处获得对数据库的下列需求。

- 1) 信息需求：用户将从数据库中获得的信息的内容和性质，对信息的要求、导出数据的要求，即确定在数据库中存储哪些数据。
- 2) 处理需求：用户要完成什么处理功能，对某种处理的执行频度、

用户要求的响应时间，处理的方式是联机处理还是批处理等。

3) 安全性和完整性的约束。

51、与文件管理相比，用数据库管理数据有哪些优点？

答：(1) 相互关联的数据集合；(2) 较少的数据冗余；(3) 程序与数据相互独立；(4) 保证数据的安全可靠；(5) 最大限度地保证数据的正确性；(6) 数据可以共享并能保证数据的一致性。

52、自连接与普通内连接的主要区别是什么？

答：自连接中进行连接操作的表在物理上是一张表，而普通内连接进行连接的表在物理上是两张表。

判断(286)--

1、'tell'+236 的结果为：tell236。

错

2、“WHERE grade!-NULL”子的作用是筛选出 grade 不为空的数据。

-->错

3、/**/之间的内容用于表示注释，对程序无影响。

对

4、B/S 模式的开发、维护工作主要集中在服务器端，可减轻异地用户的维护成本。

错

5、B/S 模式的开发、维护工作主要集中在数据库服务器端，可减轻异地用户的维护成本。-->错

6、B/S 模式实际上就是两层结构的 c/s 模式。-->错

7、Bit 数据类型可以存储小数-->错

8、C/S 模式主要由客户应用程序和数据库两部分组成。

对

9、CHAR (n) 数据类型中，n 的值不能超过 4000。-->错

10、CHECK 约束用于提供列的默认值。-->错

11、CREATE CLUSTERED INDEX idx...的作用是定义的一个非聚集索引。

错

12、CREATE CLUSTERED INDEX 错误 id 错误...的作用是定义的一个非聚集索引。-->错

13、date 数据类型用于定义字符-->错

14、decimal (p,s) 数据类型属于字符型。-->错

15、DEFAULT 约束的作用是限制列的取值不重。-->错

16、DEFAULT 约束用于限制列的值在指定的范围内。-->错

17、DEFAULT 只能在“列级完整性约束定义”处定义。

对

18、DROP INDEX studentNo_index2 的功能是删除名为 studentNo_index2 的视图。(错)

19、DROPPROC 的功能是删除表结构。

错

20、DROP VIEW 的功能是删除表结构。-->错

21、DROP VIEW 的功能是删除表结构。

错

22、DROP 语句不允许在触发器中使用-->对

23、E-R 模型的联系只能关联两个实体。-->错

24、E-R 模型设计属于数据库概念结构设计。-->对

25、E-R 模型是一种用于描述数据的组织形式的模型。-->错

26、E-R 模型中的“E”代表联系。-->错
27、E-R 图中的“E”表示的是实体。-->对
28、FOREIGNKEY 约束能够自动建立索引。
错
29、HAVING 子句用于对分组后的统计结果再进行过滤，它用于组而不是对单个记录。-->对
30、Msdb 数据库是用户必须建立的，用作 SQLServer 实例上创建的所有数据库的模板。-->错
31、NOTNULL 用于限制列的取值不能为空。-->对
32、Recordset 对象是数据表的查询结果，通常可使用它进行数据的插入操作。-->错
33、RETURN 的作用是按照条件退出。-->错
34、ROLLBACK 表示事务正常结束，COMMIT 表示事务中的全部操作被撤销。-->错
35、soLServer 允许将一行数据存储在多个数据页上。-->错
36、SQLServer 允许将一行数据存储在多个数据页上。
X
37、SQLServer 中的 tempdb 数据库是一个临时数据库，属用户数据库。-->错
38、SQLServer2008 的 SSMS 工具只支持图形化方法，不支持编写 SQL 语句。-->错
39、SQLServer2008 是一个集中式结构的数据库管理系统。-->错
40、SQLServer2008 中，当登录账户成为某数据库的合法用户之后，其对该数据库中的用户数据和对象自动拥有各种操作权限。-->错
41、SQLServer2008 中，用户创建的数据库的大小不能小于 9M-->错
42、SQLServer2008 最核心的服务是 SSM-->错
43、SQLServer2008 最核心的服务是 SSMS。
错
44、SQLServer2008 最核心的服务是 SSMSO。-->错
45、SQLServer 不允许将同一个数据库的数据文件和日志文件存储在相同的位置。-->错
46、SQLServer 不允许将同一个数据库的数据文件和日志文件存储在相同的位置。-->错
47、SQLServer 的 SSMS 工具只支持图形化操作方法，不支持编写 SQL 语句。-->错
48、SQLServer 的 SSMS 工具只支持图形化操作方法，不支持编写 SQL 语句。-->错
49、SQLServer 的日志文件可以与次要数据文件存放在相同文件夹中。-->对
50、SQLServer 的日志文件可以与次要数据文件存放在相同文件夹中。
对
51、SQLServer 的系统数据库是由 DBMS 自动创建和维护的。-->对
52、SQLServer 的一个数据库只能有一个日志文件。
错
53、SQLServer 的一个数据库只能有一个日志文件。-->错

54、SQLServer 数据库可以自动通过后台实现并发控制，不需要用户干涉。-->对
55、SQLServer 数据库中的 NULL（空值）表示的是“0”值。-->错
56、SQLServer 数据库中的 NULL 值（空值）表示的是“0”值-->错
57、SQLServer 数据库中的 NULL 值（空值）表示的是“空格”或“0”值。-->错
58、SQLServer 一个数据库不能有多个日志文件。-->错
59、SQLServer 一个数据库可以有多个次要数据文件，这些次要数据文件必须在同一个磁盘上。-->错
60、SQLServer 一个数据库可以有多个次要数据文件。-->对
61、SQLServer 一个数据库可以有多个主要数据文件。-->错
62、SQLServer 一个用户账号一般都对应一个登录账号。-->对
63、SQLServer 一个用户账号一般都对应一个登录账号。
对
64、SQLServer 允许将同一个数据库的数据文件和日志文件存储在相同的位置。-->对
65、SQLServer 允许将一行数据存储在多个数据页上
错
66、SQLServer 允许将一行数据存储在多个数据页上。-->错
67、SQLServer 中，一个数据库必须包含次要数据文件，而且可以包含多个次要数据文件。-->错
68、SQLServer 中，一个数据库必须有一个日志文件，也可以有多个日志文件。-->对
69、SQLServer 中的 tempdb 数据库是一个临时数据库，属用户数据库-->错
70、SQLServer 中的 tempdb 数据库是一个临时数据库，属用户数据库
错
71、SQLServer 中的“简单恢复模式”不支持对数据库进行日志备份。-->对
72、SQLServer 中的索引一般采用 B 树结构。-->对
73、SQLServer 中的字符串常量只能用双引号括起来，例如“软件工程”。-->错
74、SQLServer 最核心的服务是 SSMS。-->错
75、SQLServer 最核心的服务是 SSMS。-->错
76、SQLServer 最核心的服务是数据库引擎服务。-->对
77、SQL 结构化查询语言是一种过程化语言。（错）
78、SQL 虽被称为“查询语言”，其功能却不仅仅限于查询（对）
79、SQL 语句“BEGINTRAN[SACTION]事务名”的主要作用是：开始一个事务。-->对
80、SQL 语句“BEGINTRAN[SACTION]事务名”的作用是：回滚一个事务。-->错
81、SQL 语句“COMMITTRAN[SACTION]事务名”的作用是：回滚一个事务。-->错
82、SQL 语言的特点是过程化。-->错
83、SSMS 工具提供了图形化的操作界面来创建和维护对象，下拉列表框列出了当前查询编辑器所连接的数据库服务器上的所有数据库，master 表示当前正在操作的数据库是 master。-->对
84、SSMS 只支持用图形化方法访问数据库，不支持用编写语句的方法访问数据库。

X
85、Tempdb 数据库是系统自动生成的临时数据库，用于保存临时对象或中间结果集。-->对
86、UNIQUE 用于限制列的值在指定范围内。-->错
87、Unique 约束用于提供列的默认值。-->错
88、UNIQUE 约束用于限制列的值在指定范围内。-->错
89、UNIQUE 约束只能限制一个列的取值不重复。-->错
90、Unique 约束属于用户定义的完整性约束。-->对
91、UPDATE、INSERT 和 DELETE 都可引发触发器代码的执行。-->对
92、VB 提供了三种数据库访问引擎，分别是 Jet 引擎、ODBC 和 OLEDB，目前主要使用的是 OLEDB。-->对
93、“SageTINYINTCHECK (Sage>22ANDSage<-60)”“限制 Sage 列的取值只能在 22 到 60 之间。
✓
94、
“SELECT*FROMStudentINNERJOINSConStudent.Sno=SC.Sno”语句可将 Student 与 SC 连接起来。
对
95、“SELECT*FROMStudentWHERENameLIKE'[张李刘]%'”语句可以查找姓张、李、刘的学生。
对
96、“SELECT*FROMStudentWHERENameLIKE'张_’”语句中的‘张_’用于查找姓张的学生。
错
97、“SELECT*FROMStudentWHERENameLIKE'张_’”语句中的‘张_’用于查找姓张的学生。
错
98、“SELECTCOUNT (DISTINCTSno) FROMSC”语句用于统计学生的总人数。
错
99、“SsexNCHAR (1) DEFAULT'女’”语句中的 DEFAULT 约束表示 Ssex 变量的默认值为‘女’。
对
100、按指针链接方向查找数据是关系数据库的特点。-->错
101、把实验室中机器上的数据库备份到自己的移动设备上，只需要备份主数据文件（MDF）即可。-->错
102、备份数据库的主要作用是保证数据的完整性。-->错
103、不管对表进行什么类型的操作，在表上建立的索引越多越能提高操作效率。-->错
104、不能将文本文件中的数据导出到数据库中-->错
105、参照完整性规则通常是指两个参照和被参照关系中，参照关系中每条记录的外键或者为空，或者等于被参照关系中某条记录的主键。-->对
106、操作系统也是数据库系统的组成部分。-->错
107、差异备份是备份从最近的完整备份之后的日志文件。-->错
108、触发器可以实现复杂的数据完整性约束。-->对
109、触发器可由对数据的增、删、改、查操作引发。-->错
110、触发器名在数据库中可以不唯一的。-->错

111、触发器是一种特殊的存储过程，可以把它看作是表定义的一部分，主要用来保证数据的完整性。（对）
112、触发器是一种特殊的存储过程，它基于一个表创建，可以针对多个表进行操作。-->对
113、传统的关系运算包括选择、投影和连接等。-->错
114、存储过程的参数有输入参数和输出参数两种。。-->对
115、存储在客户端的代码并可以调用执行的就是存储过程。-->错
116、当登录账户成为某数据库的合法用户之后，其对该数据库中的用户数据和对象自动拥有各种操作权限。-->错
117、当基本表中的数据发生变化时，从视图中查询出的数据并不随之变化。-->错
118、当删除一个视图时，与之相关联的基本表也会自动被删除。（错）
119、当数据库损坏时，数据库管理员可使用辅助文件恢复数据库。-->错
120、登录账号用于身份验证，用户账号用于权限验证。（对）
121、等值连接要求相等的分量必须有共同的属性名。-->错
122、第二范式的关系模式也一定是第三范式的。-->错
123、第三范式的关系模式也一定是第二范式的-->对
124、第三范式的关系模式一定不包含部分函数依赖。-->对
125、定义“开关状态”列的类型为 char (2)，并为该列指定默认值“关”的子句是：开关状态 char (2) default'关'for 开关状态-->错
126、定义触发器语句中不允许使用 DROP 语句。-->对
127、定义视图时，可以指定视图的全部列名。-->对
128、对固定长度的字符串数据，如果空间没有被占满，系统自动用 0 填充。-->错
129、对数据库大小的修改，需要 msdb 数据库维护。-->错
130、对象权限是用户在已经创建好的对象上行使的权限，例如 CREATETABLE。-->错
131、非聚集索引并不改变数据的物理存储顺序，可以在一个表上建立多个非聚集索引。-->对
132、概念层数据模型描述的是数据的组织方式。-->错
133、根据索引的特点，应该对那些数据量大、查询频度较高、实时性要求强的基本表创建索引。。-->对
134、更改数据操作属于数据的静态特征。。-->错
135、关系操作的对象和结果都是记录，而不是集合。。-->错
136、关系代数中，进行交运算的两个关系不需要有相同的结构。-->错
137、关系代数中，两个关系并运算的结果是包含了两个关系中的所有数据，但不包含重复数据。-->对
138、关系代数中的“与”运算是两个条件都为真时，“与”的结果为真。-->对
139、关系代数中的选择运算是从水平方向选取数据，其结果不影响关系的列个数。-->对
140、关系模型三要素包括数据结构、数据操作和数据完整性约束。-->对
141、关系模型由数据结构、数据操作和数据完整性约束三个部分组成-->对
142、关系数据库的组织层数据模型是简单二维表。-->对

143、关系数据库即保存视图的定义，也保存视图中的数据。-->错
144、关系数据库中，用户访问数据时需要知道物理层的数据存储使用的指针。-->错
145、关系数据库中，用户能够看到物理层的数据存储使用的指针。-->错
146、关系数据模型是一种组织层数据模型。-->对
147、关系数据模型的数据结构是简单二维表。。-->对
148、关系数据模型是二维表结构的数据模型。-->对
149、关系数据模型是一种组织层数据模型。-->对
150、关系数据模型为用户提供了导航式的数据访问方式。-->错
151、关系数据模型为用户提供了非导航式的数据访问方式。-->对
152、关系数据模型允许一个属性包含更小的属性。-->错
153、计算机数据处理技术大致经历了人工管理、文件管理和 SQLServer 数据库管理三大阶段（错）。
154、将 E-R 模型转换为关系模型是数据库逻辑结构设计任务。-->对
155、将概念模型转换为关系模型属于概念结构设计任务。-->错
156、将实验室中机器上的数据库备份到自己的移动设备上，只需要备份主数据文件（MDF）即可。。-->错
157、具有 UNIQUE 约束的字段值不能重复。。-->对
158、具有数据更改权限的用户都自动具有该数据的查询权限。。-->错
159、具有一对多联系的两个实体，通常被转换为 2 个关系模式。-->对
160、可以将 Excel 文件中的数据导入到 SQLServer 中。。-->对
161、可以在一个表的一个操作建立多个前触发型触发器。-->错
162、逻辑文件名是在所有 SQL 语句中引用物理文件时所使用的名称。物理文件名包括存储文件的路径以及物理文件名本身。-->对
163、每个用户均需要通过登录账号建立自己的连接能力，以便获得对 SQL 的访问权限。。-->对
164、面向对象数据模型是数据库技术和面向对象程序设计方法相结合。。-->对
165、目前，在数据库系统中，使用最广泛的数据模型是面向对象模型。（错）
166、目前的关系数据库系统一般都支持标准 SQL 语句（对）
167、能够惟一表示数据表中的每条记录的字段或者字段的组合称为主码或主键（对）。
168、任何一个用户账号背后都有一个登录账号 o（对）
169、任何一个用户账号背后都有一个登录账号。。-->对
170、日志备份备份的是从上次备份到当前时刻新增的数据库日志内容。-->对
171、日志文件的推荐扩展名为.ldf，用于存放恢复数据库的所有日志信息。每个数据库只能有一个日志文件。-->错
172、如果 master 数据库出现故障，不影响对 SQLServer 服务的启动。。-->错

173、如果存储过程有输入参数并且没有为输入参数指定默认值，则在调用存储过程时，会自动生成一个常量值。-->错
174、如果存在非主属性对主键的部分函数依赖，则关系模式不是 2NF 的。-->对
175、如果存在非主属性对主键的传递函数依赖，则关系模式不是 3NF 的。-->对
176、如果使用 Connection 对象建立与数据库的连接，在声明对象之后建立数据库连接的方法是 Create。。-->错
177、如果我们想将数据库系统从被安装的系统盘转移到 E 盘上，可以进行数据库分离和附加操作。。-->对
178、若某个关系模式的主键只由一个列组成，则此关系模式一定是第三范式的。-->错
179、若授予某用户对某数据表具有更改权限，则该用户对该表自动具有查询权限。-->错
180、若一个学校有 900 余学生，学号用数字字符表示，考虑到学校未来发展，学生人数有可能过千，学号的长度最好设为 4 位。-->对
181、三级抽象模式的定义是 E-R 图最基本的内容。（错）
182、删除数据库，只能删除数据文件，并不删除日志文件。-->错
183、设 ADOrs 是一个 Recordset 对象，则 ADOrs (“学号”)表示得到“学号”列的值。-->对
184、设计局部 E-R 图属于逻辑设计的任务。-->错
185、实体完整性约束通常用主键实现。-->对
186、使用 FOREIGNKEY 约束可以实现参照完整性约束。-->对
187、使用数据库管理数据可以实现程序与数据的相互独立。-->对
188、使用索引的一个优点是，在对数据进行插入、更改和删除操作时，不需要对索引进行相应维护，就可以使索引与数据保持一致。-->错
189、事务的持久性是指数据库中一个事务的执行不能被其他事务干扰。-->错
190、事务的原子性是指事务对数据的修改是持久的。-->错
191、事务的原子性指事务一旦提交，其对数据库中的数据的改变是永久性的。-->错
192、事务可以保证在一个事务中的全部操作或者全部成功，或者全部失败。-->对
193、事务日志备份既备份数据库的日志记录也备份数据库的数据。。-->错
194、事务一旦提交其对数据的更改就是永久的。-->对
195、事务一致性是指数据库中的一个事务的执行不能被其他事务干扰。-->错
196、事务一致性是指数据库中的一个事务的执行不能被其他事务干扰。-->错
197、视图是一个虚表，数据库中只存储视图的定义，而不存储视图所包含的数据，这些数据仍存放在原来的基本表中。-->对
198、数据操作功能用于控制用户对数据库的操作权限。。-->错
199、数据的插入、删除和修改称为数据模型三要素。-->错
200、数据的特征分为静态特征和动态特征。-->对
201、数据控制功能用于定义、修改数据库中的对象。。-->错
202、数据库的并发控制是数据库管理员的管理任务之一。-->错

203、数据库的概念结构设计通常与具体的数据库管理系统无关。-->对

204、数据库的空间会出现不够用的情况，如果日志空间不够了则意味着不能再向数据库中插入数据；如果数据空间不够了，则意味着不能再对数据库数据进行任何修改操作。-->错

205、数据库概念结构设计的主要工作是将 E-R 模型转换为关系表。-->错

206、数据库管理系统可以脱离操作系统独立运行。-->错

207、数据库管理系统是一个系统软件，这个软件主要负责将磁盘上的数据库文件读入到内存中。-->错

208、数据库管理系统属于应用软件。-->错

209、数据库即保存视图的定义，也保存视图中的数据。。-->错

210、数据库逻辑设计的一个工作是设计面向用户的外模式。-->对

211、数据库三级模式中，内模式到模式的映像提供了数据的物理独立性。-->错

212、数据库三级模式中，外模式到模式的映像提供了数据的逻辑独立性。-->对

213、数据库三级模式中的模式是面向全体用户的数据需求设计的。-->对

214、数据库三级模式中的内模式描述数据的物理存储。-->对

215、数据库三级模式中的内模式又称为存储模式，描述数据的物理存储。。-->对

216、数据库三级模式中的外模式是面向全体用户的数据需求设计的。-->错

217、数据库设计就是建立一个数据库应用系统。-->错

218、数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，有效存储数据，满足用户信息要求和处理要求。（对）

219、数据库设计中的行为设计主要是描述数据库的结构。-->错

220、数据库系统可以脱离操作系统独立运行-->错

221、数据库应用程序开发中，需求分析阶段的主要目的是回答“干什么，，的问题。（对）

222、数据库只需要反映数据本身的内容，数据库系统反映数据之间的联系。-->错

223、数据库中只有一个模式，也只有一个内模式，所以模式 / 内模式映像是惟一的。（对）。

224、数据流图是需求分析的常用工具之一，人们也经常称它为 DFD 图。。-->对

225、数据冗余不仅造成存储空间浪费，还会导致数据的不一致。-->对

226、数据是经过加工的，有价值的数字。（错）

227、数据字典是需求分析的常用工具之一，简称为 DD。。-->对

228、数据字典是需求分析的常用工具之一，人们也经常称它为 DFD 图。。-->错

229、索引的作用是提高数据的查询效率。-->对

230、索引建立的越多越能提高数据查询效率。-->错

231、索引实际上是一张建立在基本表上的虚表。。-->错

232、索引是在基本表的列上建立的一种数据库对象，它和基本表分开存储，它的建立或撤消对数据的内容毫无影响。（对）

233、索引一般来说可以提高数据增、删、改、查的效率。-->错

234、索引与视图具有相同的用途。。-->错

235、通常应该在频繁进行更新操作的列上建立非聚集索引。-->错

236、通过定义主键约束可以实现参照完整性。-->错

237、同一个数据上如果被加了排他锁，则不能再被加任何其他锁。-->对

238、外模式是面向每类用户的信息需求设计的。-->对

239、为某基本表建立索引，如果对基本表中的数据进行增删操作，则需要手工重新建立对应的索引文件。。-->错

240、为某基本表建立索引，如果对索引进行撤销操作则与之对应的的基本表的内容也会被删除。。-->错

241、限制“年龄”列取值范围在 0-160 的约束表达式是：check（年龄 in 0-160）。。-->错

242、选择、投影和连接操作属于传统的关系运算。-->错

243、一个 DEFAULT 约束可以为表中的多个列提供默认值。-->错

244、一个不规范的关系模式通常会引发插入异常、删除异常和更新异常，导致大量的数据冗余。（对）

245、一个关系模式的主键可由多个列共同组成。-->对

246、一个关系模式只能有一个候选键。-->错

247、一个数据库必须包含次要数据文件，可以包含一个或多个次要数据文件。-->错

248、一个数据库必须有一个日志文件，也可以有多个日志文件-->对

249、一个数据库可以有多个外模式。-->对

250、一级封锁协议是对读数据加共享锁，读完即释放。-->错

251、隐含权限是数据库管理系统预先定义好的，不需要再明确地授予的权限。-->对

252、用户创建的数据库的大小不能小于 3M-->对

253、用户可以不通过数据库管理系统直接操作数据库文件。-->错

254、用户可以不通过数据库系统直接操作数据库-->错

255、由于聚集索引项决定了表中数据的物理存储顺序，因此一个表只能包含一个聚集索引。-->错

256、在 E-R 模型中，联系只能且必须关联两个实体。-->错

257、在 E-R 模型中，实体之间的联系有一对-->对

258、在 SQLServer 中，不允许删除正在被用户使用的数据库。-->对

259、在 SQLServer 中，可以手工扩大数据文件和日志文件的空间。-->对

260、在 SQLServer 中，可以手工扩大数据文件和日志文件的空间。。-->对

261、在 SQLServer 中，删除数据库时，只删除了数据库的数据文件，并不删除日志文件。-->错

262、在 SQLServer 中，数据库文件有两个名称，一个是操作系统管理的物理文件名，一个是数据库管理系统管理的逻辑文件名。-->对

263、在 SQLServer 中，数据库文件有两个名称，一个是主文件名，一个是次文件名。-->错

264、在关系代数中，选择、投影和连接操作属于传统的关系运算。-->错

265、在关系模型中，实体之间的联系是通过指针来实现的。-->错

266、在恢复数据库之前，如果数据库的日志文件没有损坏，可在恢复之前对数据库进行一次尾部日志备份，这样可将数据的损失减少到最小。-->对

267、在建立聚集索引时，数据库管理系统首先要将数据按聚集索引列进行物理排序。-->对

268、在教师任课关系（教师编号，教师姓名，教师职称，课程号，课程名，课时费）中，主键是教师编号。（错）

269、在进行数据导入导出时，soLServer 会自动维护目的数据的完整性。-->错

270、在进行数据库概念结构设计时，通常需要考虑所使用的数据库管理系统。-->错

271、在进行数据库概念结构设计时，要充分考虑所使用的数据库管理系统。-->错

272、在进行数据库逻辑结构设计时，通常不需要考虑所使用的数据库管理系统。-->错

273、在设计概念层数据模型时要考虑具体的数据库管理系统。-->错

274、在实际应用中，尽可能不要通过视图对记录进行更新操作。-->对

275、在数据库三级模式中，外模式到模式的映像提供了数据的物理独立性。-->错

276、在数据库设计中，通常是先进行逻辑结构设计，然后再进行概念结构设计。-->错

277、在数据库系统中，应用程序可以直接访问数据库中的所有数据。-->错

278、在一台计算机上安装一次 SQLServer 时，就生成了一个实例。在一台计算机上只能安装一个默认实例，同时也只能安装一个命名实例。-->错

279、在一台计算机上只能安装一个 SQLServer 默认实例。-->对

280、值特别长的列适宜创建索引。。-->错

281、主键可以取空值。（错）

282、主要数据文件的推荐扩展名是 mdf 对

283、足用户信息要求和处理要求。（对）

284、组织层数据模型是一种与具体 DBMS 无关的数据模型。-->错

285、遵守两段锁协议的并发调度一定是正确的调度。-->对

286、做结构设计时，要尽量使设计符合第三范式的要求。（对）

综合题(65)--

- 1、设有描述教师授课情况的如下三张表，各表结构...
- 2、设有描述教师授课情况的如下三张表，各表结构...
- 3、设有描述教师授课情况的如下三张表，各表结构...
- 4、设有描述教师授课情况的如下三张表，各表结构...
- 5、设有描述教师授课情况的如下三张表，各表结构...
- 6、设有描述商品生产的如下三张表，各表结构如下...
- 7、设有描述商品生产的如下三张表，各表结构如下...
- 8、设有描述商品生产的如下三张表，各表结构如下...
- 9、设有描述商品生产的如下三张表，各表结构如下...
- 10、设有描述商品生产的如下三张表，各表结构如下...
- 11、设有描述商品生产的如下三张表，各表结构如下...
- 12、设有描述图书出版情况的如下三张表，各表结构...
- 13、设有描述图书出版情况的如下三张表，各表结构...
- 14、设有描述图书出版情况的如下三张表，各表结构...
- 15、设有描述图书出版情况的如下三张表，各表结构...
- 16、设有描述图书出版情况的如下三张表，各表结构...
- 17、设有图书、书店、销售三张表，各表结构如下：...
- 18、设有图书、书店、销售三张表，各表结构如下：...
- 19、设有图书、书店、销售三张表，各表结构如下：...
- 20、设有图书、书店、销售三张表，各表结构如下：...
- 21、设有图书、书店、销售三张表，各表结构如下：...
- 22、设有图书、书店、销售三张表，各表结构如下：...
- 23、有如下房屋、租客和租赁三张表，各表结构如下...
- 24、有如下房屋、租客和租赁三张表，各表结构如下...
- 25、有如下房屋、租客和租赁三张表，各表结构如下...
- 26、有如下房屋、租客和租赁三张表，各表结构如下...
- 27、有如下房屋、租客和租赁三张表，各表结构如下...
- 28、有如下房屋、租客和租赁三张表，各表结构如下...
- 29、有如下教师、课程和授课三张表，各表结构如下...
- 30、有如下教师、课程和授课三张表，各表结构如下...
- 31、有如下课题、成员和参与三张表，各表结构如下...
- 32、有如下课题、成员和参与三张表，各表结构如下...
- 33、有如下汽车、经销商和销售三张表，各表结构如...
- 34、有如下汽车、经销商和销售三张表，各表结构如...
- 35、有如下汽车、生产厂家和销售三张表，各表结构...
- 36、有如下汽车、生产厂家和销售三张表，各表结构...
- 37、有如下汽车、生产厂家和销售三张表，各表结构...
- 38、有如下汽车、生产厂家和销售三张表，各表结构...
- 39、有如下汽车、生产厂家和销售三张表，各表结构...
- 40、有如下商品、客户和销售三张表，各表结构如下...
- 41、有如下商品、生产厂家和生产三张表，各表结构...
- 42、有如下图书、经销商和销售三张表，各表结构如...
- 43、有如下图书、经销商和销售三张表，各表结构如...
- 44、有如下图书、书店和销售三张表，各表结构如下...
- 45、有如下图书、书店和销售三张表，各表结构如下...
- 46、有如下图书、书店和销售三张表，各表结构如下...
- 47、有如下图书、书店和销售三张表，各表结构如下...
- 48、有如下图书、书店和销售三张表，各表结构如下...
- 49、有如下图书、书店和销售三张表，各表结构如下...

- 50、有如下图书、书店和销售三张表，各表结构如下...
- 51、有如下项目、会员和健身三张表，各表结构如下...
- 52、有如下项目、会员和健身三张表，各表结构如下...
- 53、有如下项目、会员和健身三张表，各表结构如下...
- 54、有如下项目、会员和健身三张表，各表结构如下...
- 55、有如下项目、会员和健身三张表，各表结构如下...
- 56、有如下项目、会员和健身三张表，各表结构如下...
- 57、有如下项目、会员和健身三张表，各表结构如下...
- 58、有如下项目、运动员和比赛三张表，各表结构如...
- 59、有如下项目、运动员和比赛三张表，各表结构如...
- 60、有如下项目、运动员和比赛三张表，各表结构如...
- 61、有如下项目、运动员和比赛三张表，各表结构如...
- 62、有如下项目、运动员和比赛三张表，各表结构如...
- 63、有如下项目、运动员和比赛三张表，各表结构如...
- 64、有如下项目、运动员和比赛三张表，各表结构如...
- 65、有如下项目、运动员和比赛三张表，各表结构如...

1、设有描述教师授课情况的如下三张表，各表结构如下：

“教师”表结构：

教师号：普通编码定长字符类型，长度为 10，主键。

教师名：普通编码可变长字符类型，长度为 10，非空。

职称：统一字符编码定长字符类型，长度为 3，取值为：“教授”、“副教授”、“讲师”。

所在部门：普通编码可变长字符类型，长度为 20。

“课程”表结构：

课程号：普通编码定长字符类型，长度为 20，主键。

课程名：统一字符编码可变长字符类型，长度为 20，非空。

开课学期：小整型。

学分：定点小数类型，小数点前 2 位，小数点后 1 位

“授课”表结构：

教师号：普通编码定长字符类型，长度为 10，引用教师表的外键。

课程号：普通编码定长字符类型，长度为 20，引用课程表的外键。

授课学年：普通编码定长字符类型，长度为 6。

(1) 在课程表中插入一行数据，课程号：K100，课程名：新课程，开课学期：2，学分：待定。下列语句正确的是 ()。

[A.INSERT INTO 课程 VALUES\('K100','新课程',2, NULL\)](#)

[C.INSERT INTO 课程\(课程号,课程名,开课学期\) VALUES\('K100','新课程',2\)](#)

(2) 现要创建满足如下要求的视图：查询“教授”职称的教师的授课情况，列出教师名、讲授的课程名及课程学分。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) [A.教师名](#),[课程名](#),[学分](#) FROM (2) [D.课程](#) aJOIN 授课 b ON a.课程号 = b.课程号 JOIN (3) [E.教师](#) c ON (4) [D.课程](#) WHERE (5) [B.职称](#) = '教授'

(3) 有如下定义课程表和授课表的语句，请补全空白部分。

CREATE TABLE 课程(课程号 char(20) [\(1\)D.primary key](#),课程名 [\(2\)N.nvarchar\(20\)](#) [\(3\)E.not null](#) ,开课学期 [\(4\)A.smallint](#) ,学分 [\(5\)H.numeric\(3,1\)](#))

CREATE TABLE 授课(教师号 [\(6\)B.char\(10\)](#),课程号 char(20),授课学年 [\(7\)F.char\(6\)](#) , [\(8\)K.foreign key](#) (课程号) [\(9\) \[I.references](#) [\(10\) G.课程](#)(课程号))

(4) 删除学分小于 2 的课程记录。下列语句正确的是 ()。

[A.DELETE FROM 课程 WHERE 学分 < 2](#)

[C.DELETE 课程 WHERE 学分 < 2](#)

(5) 查询第 2 学期开设的课程总门数。下列语句正确的是 ()。

[A.SELECT COUNT\(*\) FROM 课程 WHERE 开课学期 = 2](#)

[B.SELECT COUNT\(课程号\) FROM 课程 WHERE 开课学期 = 2](#)

[C.SELECT COUNT\(课程名\) FROM 课程 WHERE 开课学期 = 2](#)

(6) 查询学分为 4 或 5 的课程的信息, 列出课程名、学分和开课学期。下列语句正确的是 ()。

B.SELECT 课程名, 学分, 开课学期 FROM 课程 WHERE 学分 = 4 OR 学分 = 5

C.SELECT 课程名, 学分, 开课学期 FROM 课程 WHERE 学分 IN(4, 5)

2、设有描述教师授课情况的如下三张表, 各表结构如下:

“教师”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 主键。

教师名: 普通编码可变长字符类型, 长度为 10, 非空。

职称: 统一字符编码定长字符类型, 长度为 3, 取值为: “教授”、“副教授”、“讲师”。

所在部门: 普通编码可变长字符类型, 长度为 20。

“课程”表结构:

课程号: 普通编码定长字符类型, 长度为 20, 主键。

课程名: 统一字符编码可变长字符类型, 长度为 20, 非空。

开课学期: 小整型。

学分: 定点小数类型, 小数点前 2 位, 小数点后 1 位

“授课”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 引用教师表的外键。

课程号: 普通编码定长字符类型, 长度为 20, 引用课程表的外键。

授课学年: 普通编码定长字符类型, 长度为 6。

(1)

统计“计算机学院”的教师人数。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 教师 WHERE 所在部门 = '计算机学院'

B.SELECT COUNT(教师号) FROM 教师 WHERE 所在部门 = '计算机学院'

C.SELECT COUNT(教师名) FROM 教师 WHERE 所在部门 = '计算机学院'

(2) 有如下定义教师表和授课表的语句, 请补全空白部分。

CREATE TABLE 教师(教师号 char(10) (1) D.primary key 教师名 varchar(10) (2) E.not null, 职称 (3) A.nchar(3) (4) F.check (职称 IN('教授','副教授','讲师')), 所在部门 (5) N.varchar(20))

CREATE TABLE 授课(教师号 char(10), 课程号 (6) B.char(20), 授课学年 (7) H.char(6), (8) K.foreign key (教师号) (9) I.references (10) C.教师(教师号))

(3) 删除授课学年为“20000101”的授课记录。下列语句正确的是 ()。

A.DELETE FROM 授课 WHERE 授课学年 = '20000101'

C.DELETE FROM 授课 WHERE 授课学年 LIKE '20000101'

(4) 现要创建满足如下要求的视图: 查询“计算机学院”全体教师的授课情况, 列出教师名、讲授的课程名及授课学年。请补全空白部分 (设视图名为: V1)。

CREATE (1) D.VIEW V1 AS SELECT (2) E.教师名, 课程名, 授课学年 FROM 课程 a JOIN (3) A.授课 b ON (4) C.a. 课程号 = b. 课程号 JOIN 教师 c ON c. 教师号 = b. 教师号 WHERE (5) B.所在部门 = '计算机学院'

(5) 在课程表中插入一行数据, 课程号: K100, 课程名: 新课程, 开课学期: 未定, 学分: 3. 5。下列语句正确的是 ()。

A.INSERT INTO 课程 VALUES('K100', '新课程', NULL, 3. 5)

C.INSERT INTO 课程(课程号, 课程名, 学分) VALUES('K100', '新课程', 3. 5)

D.INSERT INTO 课程(课程号, 课程名, 学分, 开课学期) VALUES('K100', '新课程', 3. 5, NULL)

(1) 查询有“教授”职称的教师名和所在部门。下列语句正确的是 ()。

A.SELECT 教师名, 所在部门 FROM 教师 WHERE 职称 = '教授'

B.SELECT 教师名, 所在部门 FROM 教师 WHERE 职称 IN ('教授')

3、设有描述教师授课情况的如下三张表, 各表结构如下:

“教师”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 主键。

教师名: 普通编码可变长字符类型, 长度为 10, 取值不重。

职称: 统一字符编码定长字符类型, 长度为 3, 取值为: “教授”、“副教授”、“讲师”。

所在部门: 普通编码可变长字符类型, 长度为 20。

“课程”表结构:

课程号: 普通编码定长字符类型, 长度为 20, 主键。

课程名: 统一字符编码可变长字符类型, 长度为 20, 非空。

开课学期: 小整型。

学分: 定点小数类型, 小数点前 2 位, 小数点后 1 位

“授课”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 引用教师表的外键。

课程号: 普通编码定长字符类型, 长度为 20, 引用课程表的外键。

授课学年: 普通编码定长字符类型, 长度为 6。

(教师号, 课程号, 授课学年) 为主键。

(1) 现要创建满足如下要求的视图: 查询“202201”授课学年开设的课程名、授课教师名、所在部门。请补全空白部分 (设视图名为: V1)。

CREATE VIEW V1 AS SELECT (1) A.课程名, 教师名, 所在部门 FROM (2) B.教师 a JOIN 授课 b ON a. 教师号 = b. 教师号 JOIN (3) D.课程 c ON (4) C.b. 课程号 = c. 课程号 WHERE (5) E.授课学年 = '202201'

(2) 将“数据库”课程的学分增加 1 分。下列语句错误的是 ()。

A.UPDATE 课程 SET 学分 = 1 WHERE 课程名 = '数据库'

C.UPDATE 学分 SET 学分 = 学分 + 1 WHERE 课程名 = '数据库'

D.UPDATE 学分 SET 学分 = 1 WHERE 课程名 = '数据库'

(3) 查询姓“张”的教师名和所在部门。下列语句错误的是 ()。

B.SELECT 教师名, 所在部门 FROM 教师 WHERE 教师名 = '张%'

C.SELECT 教师名, 所在部门 FROM 教师 WHERE 教师名 LIKE '张'

D.SELECT 教师名, 所在部门 FROM 教师 WHERE 教师名 = '张'

(4) 统计每个授课学年每个教师的授课总门数。下列语句正确的是 ()。

A.SELECT 授课学年, 教师号, COUNT(*) FROM 授课 GROUP BY 授课学年, 教师号

B.SELECT 授课学年, 教师号, COUNT(教师号) FROM 授课 GROUP BY 授课学年, 教师号

C.SELECT 授课学年, 教师号, COUNT(课程号) FROM 授课 GROUP BY 授课学年, 教师号

(5) 有如下定义教师表和授课表的语句, 请补全空白部分。

CREATE TABLE 教师(教师号 char(10) (1) D.primary key 教师名 varchar(10) (2) E.unique, 职称 (3) A.nchar(3) check ((4) F.职称 IN('教授','副教授','讲师')), 所在部门 (5) N.varchar(20))

CREATE TABLE 授课(教师号 char(10), 课程号 (6) B.char(20), 授课学年 (7) H.char(6), (8) K.foreign key (教师号) (9) I.references (10) C.教师(教师号))

(6) 在教师表中插入一行数据, 教师号: T100, 教师名: 新教师, 职称: 未知, 所在部门: 未定。下列语句正确的是 ()。

B.INSERT INTO 教师 VALUES('T100', '新教师', NULL, NULL)

C.INSERT INTO 教师(教师号, 教师名) VALUES('T100', '新教师')

4、设有描述教师授课情况的如下三张表, 各表结构如下:

“教师”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 主键。

教师名: 普通编码可变长字符类型, 长度为 10, 非空。

职称: 统一字符编码定长字符类型, 长度为 3, 取值为: “教授”、“副教授”、“讲师”。

所在部门: 普通编码可变长字符类型, 长度为 20。

“课程”表结构:

课程号: 普通编码定长字符类型, 长度为 20, 主键。

课程名: 统一字符编码可变长字符类型, 长度为 20, 非空。

开课学期: 小整型。

学分: 定点小数类型, 小数点前 2 位, 小数点后 1 位

“授课”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 引用教师表的外键。

课程号: 普通编码定长字符类型, 长度为 20, 引用课程表的外键。

授课学年：普通编码定长字符类型，长度为6。

(1) 现要创建满足如下要求的视图：查询第4、5、6学期开设的课程的课程名、授课教师名和开课学期。请补全空白部分（设视图名为：V1）。
CREATE VIEW V1 AS SELECT (1) A.课程名,教师名,开课学期 FROM (2) D.课程 aJOIN 授课 b ON a. 课程号 = b. 课程号 JOIN (3) B.教师 c ON (4) C.b. 教师号 = c. 教师号 WHERE (5) E.开课学期 BETWEEN 4 AND 6

(2) 有如下定义课程表和教师表的语句，请补全空白部分。
CREATE TABLE 课程(课程号 (1) D.char(20) primary key, 课程名 (2) N.nvarchar(20) not null, 开课学期 (3) A.smallint, 学分 (4) H.numeric(3,1))CREATE TABLE 教师(教师号 (5) B.char(10) primary key, 教师名 (6) F.varchar(10) not null, 职称 (7) G.nchar(3) (8) I.check (职称 (9) K.in ('教授','副教授','讲师')), 所在部门 (10) C.varchar(20))

(3) 将“张三”老师的职称改为“教授”。下列语句正确的是 ()。

A.UPDATE 教师 SET 职称 = '教授' WHERE 教师名 = '张三'

B.UPDATE 教师 SET 职称 = '教授' WHERE 教师名 LIKE '张三'

(4) 查询“教授”和“副教授”职称的教师名、职称和所在部门。下列语句正确的是 ()。

A.SELECT 教师名, 职称, 所在部门 FROM 教师 WHERE 职称 = '教授' OR 职称 = '副教授'

C.SELECT 教师名, 职称, 所在部门 FROM 教师 WHERE 职称 IN ('教授','副教授')

(5) 在教师表中插入一行数据，教师号：T100，教师名：新教师，职称：教授，所在部门：未定。下列语句正确的是 ()。

A.INSERT INTO 教师 VALUES('T100','新教师','教授', NULL)

C.INSERT 教师(教师号, 教师名, 职称) VALUES('T100','新教师','教授')

D.INSERT 教师 VALUES('T100','新教师','教授', NULL)

(6) 统计第2学期开设的课程的总学分。下列语句错误的是 ()。

A.SELECT COUNT(学分) FROM 课程 WHERE 开课学期 = 2

B.SELECT MAX(学分) FROM 课程 WHERE 开课学期 = 2

D.SELECT AVG(学分) FROM 课程 WHERE 开课学期 = 2

5、设有描述教师授课情况的如下三张表，各表结构如下：

“教师”表结构：

教师号：普通编码定长字符类型，长度为10，主键。

教师名：普通编码可变量长字符类型，长度为10，非空。

职称：统一字符编码定长字符类型，长度为3，取值为：“教授”、“副教授”、“讲师”。

所在部门：普通编码可变量长字符类型，长度为20。

“课程”表结构：

课程号：普通编码定长字符类型，长度为20，主键。

课程名：统一字符编码可变量长字符类型，长度为20，非空。

开课学期：小整型，取值大于等于1。

学分：定点小数类型，小数点前2位，小数点后1位

“授课”表结构：

教师号：普通编码定长字符类型，长度为10，引用教师表的外键。

课程号：普通编码定长字符类型，长度为20，引用课程表的外键。

授课学年：普通编码定长字符类型，长度为6。

主键：(教师号，课程号，授课学年)

(1) 统计每个部门的教师人数。下列语句正确的是 ()。

A.SELECT 所在部门, COUNT(*) FROM 教师 GROUP BY 所在部门

B.SELECT 所在部门, COUNT(教师号) FROM 教师 GROUP BY 所在部门

C.SELECT 所在部门, COUNT(教师名) FROM 教师 GROUP BY 所在部门

(2) 有如下定义课程表和授课表的语句，请补全空白部分。

CREATE TABLE 课程(课程号 char(20) (1) D.primary key, 课程名 (2) N.nvarchar(20) (3) E.not null, 开课学期 (4) A.smallint (5) K.check (开课学期 >= 1), 学分 (6) H.numeric(3,1))

CREATE TABLE 授课(教师号 (7) B.char(10), 课程号 char(20), 授课学年 (8) F.char(6), foreign key (课程号 (9) I.references 课程((10) G.课程号))

(3) 删除“VB编程”的课程信息。下列语句正确的是 ()。

C.DELETE FROM 课程 WHERE 课程名 = 'VB编程'

D.DELETE FROM 课程 WHERE 课程名 LIKE 'VB编程'

(4) 查询第3、4学期开设的课程的课程名和学分。下列语句正确的是 ()。

A.SELECT 课程名, 学分 FROM 课程 WHERE 开课学期 = 3 OR 开课学期 = 4

B.SELECT 课程名, 学分 FROM 课程 WHERE 开课学期 IN (3, 4)

(5) 现要创建满足如下要求的视图：查询讲授学分为2-4的课程的教师名、课程名、课程学分。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 (1) A.AS SELECT 课程名, 教师名, 学分 FROM (2) B.教师 aJOIN 授课 b ON a. 教师号 = b. 教师号 JOIN (3) E.课程 c ON (4) C.b. 课程号 = c. 课程号 WHERE (5) D.学分 BETWEEN 2 AND 4

(6) 将“T01”教师的“C01”课程的授课学年改为“202202”。下列语句错误的是 ()。

A.UPDATE 授课学年 SET 授课学年 = '202202' WHERE 教师号 = 'T01' AND 课程号 = 'C01'

B.UPDATE 授课学年 SET 授课学年 = '202202' WHERE 教师号 = 'T01' OR 课程号 = 'C01'

C.UPDATE 授课 SET 授课学年 = '202202' WHERE 教师号 = 'T01' OR 课程号 = 'C01'

6、设有描述商品生产的如下三张表，各表结构如下：

“商品”表结构：

商品号：普通编码定长字符类型，长度为10，主键。

商品名：普通编码可变量长字符类型，长度为20，非空。

类别：统一字符编码定长字符类型，长度为2，取值为：“空调”、“电视”。

单价：定点小数，小数点前4位，小数点后2位。

“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为20，主键。

厂家名：普通编码可变量长字符类型，长度为30，非空。

厂家地址：统一字符编码可变量长字符类型，长度为30。

“生产”表结构：

商品号：普通编码定长字符类型，长度为10，引用商品表的外键。

厂家编号：普通编码定长字符类型，长度为20，引用生产厂家表的外键。

生产日期：日期类型。

生产数量：整型，取值大于等于100。

(1) 现要创建满足如下要求的视图：查询“北京一厂”厂家生产的商品的商品名、单价和生产数量。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) [E] 商品名, 单价, 生产数量 FROM (2) [B] 商品 aJOIN 生产 b ON (3) [A] a. 商品号 = b. 商品号 JOIN 生产厂家 c ON (4) [C] b. 厂家编号 = c. 厂家编号 WHERE (5) [D] 生产厂家 = '北京一厂'

(2) 有如下定义生产厂家表和生产表的语句，请补全空白部分。

CREATE TABLE 生产厂家(厂家编号 char(20) (1) [D] primary key, 厂家名 (2) [C] varchar(30) (3) [M] not null, 厂家地址 (4) [K] nvarchar(30))CREATE TABLE 生产(商品号 (5) [B] char(10), 厂家编号 char(20), 生产日期 (6) [G] date, 生产数量 int (7) [E] check ((8) [A] 生产数量 >= 100), FOREIGN KEY(厂家编号) (9) [H] references (10) [I] 生产厂家(厂家编号))

(3) 在“生产厂家”表中插入一行数据，厂家编号为“P100”，厂家名为“新厂家”，厂家地址未定。下列语句正确的是 ()。

A.INSERT INTO 生产厂家 VALUES('P100','新厂家', NULL)

B.INSERT 生产厂家(厂家编号, 厂家名) VALUES('P100','新厂家')

D.INSERT 生产厂家 VALUES('P100','新厂家', NULL)

(4) 查询厂家地址的前两个汉字是“北京”的厂家编号和厂家名。下列语句错误的是 ()。

A.SELECT 厂家编号, 厂家名 FROM 生产厂家 WHERE 厂家地址 LIKE '北京#'

C.SELECT 厂家编号, 厂家名 FROM 生产厂家 WHERE 厂家地址 LIKE '北京'

D.SELECT 厂家编号, 厂家名 FROM 生产厂家 WHERE 厂家地址 LIKE '北京#'

(5) 删除单价在10-20（包括边界值）的商品。下列语句错误的是 ()。

A.DELETE FROM 商品 WHERE 单价 >= 10 OR 单价 <= 20

[C.DELETE FROM 商品 WHERE 单价 >=10 OR <=20](#)

[D.DELETE FROM 商品 WHERE 单价 >=10 AND <=20](#)

(6) 统计单价高于 4000 的商品的数量。下列语句正确的是 ()。

[A.SELECT COUNT\(*\) FROM 商品 WHERE 单价 > 4000](#)

[B.SELECT COUNT\(商品号\) FROM 商品 WHERE 单价 > 4000](#)

[C.SELECT COUNT\(商品名\) FROM 商品 WHERE 单价 > 4000](#)

7、设有描述商品生产的如下三张表，各表结构如下：

“商品”表结构：

商品号：普通编码定长字符类型，长度为 10，主键。

商品名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 3，取值为：“手机”、“笔记本”。

单价：整型。

“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。

厂家名：普通编码可变长字符类型，长度为 30，非空。

厂家地址：统一字符编码可变长字符类型，长度为 40。

“生产”表结构：

商品号：普通编码定长字符类型，长度为 10，引用商品表的外键。

厂家编号：普通编码定长字符类型，长度为 20，引用生产厂家表的外键。

生产日期：日期类型。

生产数量：整型，取值大于等于 100。

主键：(商品号，厂家编号，生产日期)。

(1) 现要创建满足如下要求的视图：查询 2022 年 1 月 1 日及之后生产的商品名、类别、生产厂家名和生产日期。请补全空白部分（设视图名为：V1）。CREATE (1) [E.VIEW V1 AS SELECT](#) (2) [B.商品名,类别,厂家名,生产日期](#) FROM (3) [C.生产](#) aJOIN 商品 b ON (4) [A.a.商品号 = b.商品号](#) JOIN 生产厂家 c ON c. 厂家编号 = b. 厂家编号 WHERE (5) [D.生产日期 >= '2022-1-1'](#)

(2) 将“手机”类商品的单价提高 5%。下列语句正确的是 ()。

[A.UPDATE 商品 SET 单价 = 单价 * 1.05 WHERE 类别 = '手机'](#)

[B.UPDATE 商品 SET 单价 = 单价 + 单价 * 0.05 WHERE 类别 = '手机'](#)

(3) 查询单价在 1000-2000（包括边界值）的商品的商品名、类别和单价，下列语句正确的是 ()。

[A.SELECT 商品名,类别,单价 FROM 商品 WHERE 单价 BETWEEN 1000 AND 2000](#)

[C.SELECT 商品名,类别,单价 FROM 商品 WHERE 单价 >= 1000 AND 单价 <= 2000](#)

(4) 删除厂家地址的前两个字不是“北京”的生产厂家。下列语句错误的是 ()。

[A.DELETE FROM 生产厂家 WHERE 厂家地址 NOT = '北京'](#)

[B.DELETE FROM 生产厂家 WHERE 厂家地址 NOT LIKE '北京%'](#)

[C.DELETE FROM 生产厂家 WHERE 厂家地址 NOT = '北京%'](#)

(5) 有如下定义生产厂家表和生产表的语句，请补全空白部分。

CREATE TABLE 生产厂家 (厂家编号 char(20) (1) [D.primary key](#), 厂家名 (2) [B.varchar\(30\)](#) (3) [E.not null](#), 厂家地址 (4) [C.nvarchar\(40\)](#))

CREATE TABLE 生产 (商品号 char(10), 厂家编号 char(20), 生产日期 (5) [G.date](#), 生产数量 (6) [N.int](#) (7) [F.check](#) (生产数量 (8) [M.>=100](#)), (9) [I.foreign key](#) (厂家编号) (10) [K.references](#) 生产厂家 (厂家编号))

(6) 统计“手机”类商品数量。下列语句正确的是 ()。

[A.SELECT COUNT\(商品号\) FROM 商品 WHERE 类别 = '手机'](#)

[B.SELECT COUNT\(*\) FROM 商品 WHERE 类别 = '手机'](#)

8、设有描述商品生产的如下三张表，各表结构如下：

“商品”表结构：

商品号：普通编码定长字符类型，长度为 10，主键。

商品名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 2，取值为：“空调”、“手机”。

单价：定点小数，小数点前 4 位，小数点后 2 位。

“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。

厂家名：普通编码可变长字符类型，长度为 30，非空。

厂家地址：统一字符编码可变长字符类型，长度为 30。

“生产”表结构：

商品号：普通编码定长字符类型，长度为 10，引用商品表的外键。

厂家编号：普通编码定长字符类型，长度为 20，引用生产厂家表的外键。

生产日期：日期类型。

生产数量：整型，默认值为 100。

(1) 有如下定义商品表和生产表的语句，请补全空白部分。

CREATE TABLE 商品 (商品号 char(10) (1) [D.primary key](#), 商品名 (2) [C.varchar\(20\)](#) not null, 类别 (3) [K.nchar\(2\)](#) (4) [E.check](#) ((5) [F.类别](#) in ('空调', '手机')), 单价 (6) [M.numeric\(6,2\)](#))

CREATE TABLE 生产 (商品号 char(10), 厂家编号 char(20), 生产日期 (7) [G.date](#), 生产数量 int (8) [B.default](#) 100, (9) [H.foreign key](#) (商品号) references (10) [N.商品\(商品号\)](#))

(2) 统计“P01”号商品的总生产数量。下列语句错误的是 ()。

[A.SELECT COUNT\(*\) FROM 生产 WHERE 商品号 = 'P01'](#)

[C.SELECT COUNT\(生产数量\) FROM 生产 WHERE 商品号 = 'P01'](#)

[D.SELECT TOTAL\(生产数量\) FROM 生产 WHERE 商品号 = 'P01'](#)

(3) 删除生产数量少于 10 的生产记录。下列语句正确的是 ()。

[A.DELETE FROM 生产 WHERE 生产数量 < 10](#)

[C.DELETE 生产 WHERE 生产数量 < 10](#)

(4) 现要创建满足如下要求的视图：查询生产数量在 1000-1500（包括边界值）的商品的商品名、单价和生产数量。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) [E.商品名, 单价, 生产数量](#) FROM (2) [B.生产厂家](#) aJOIN 生产 b ON (3) [C.a. 厂家编号 = b. 厂家编号](#) JOIN 商品 c ON (4) [A.b. 商品号 = c. 商品号](#) WHERE (5) [D.生产数量 >= 1000 AND 生产数量 <= 1500](#)

(5) 在“商品”表中插入一行数据，商品号为“G100”，商品名为“新商品”，类别为“手机”，单价未知。下列语句正确的是 ()。

[C.INSERT INTO 商品 VALUES\('G100','新商品','手机',NULL\)](#)

[D.INSERT 商品 VALUES\('G100','新商品','手机',NULL\)](#)

(6) 查询“空调”类商品的商品名和单价。下列语句正确的是 ()。

[A.SELECT 商品名,单价 FROM 商品 WHERE 类别 = '空调'](#)

[C.SELECT 商品名,单价 FROM 商品 WHERE 类别 in \('空调'\)](#)

9、设有描述商品生产的如下三张表，各表结构如下：

“商品”表结构：

商品号：普通编码定长字符类型，长度为 10，主键。

商品名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 3，取值为：“手机”、“笔记本”。

单价：整型。

“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。

厂家名：普通编码可变长字符类型，长度为 30，非空。

厂家地址：统一字符编码可变长字符类型，长度为 40。

“生产”表结构：

商品号：普通编码定长字符类型，长度为 10，引用商品表的外键。

厂家编号：普通编码定长字符类型，长度为 20，引用生产厂家表的外键。

生产日期：日期类型。

生产数量：整型。

主键：(商品号，厂家编号，生产日期)。

(1) 删除 2022 年 6 月 1 日之前的生产记录。下列语句正确的是 ()。

A.DELETE FROM 生产 WHERE 生产日期 < '2022/6/1'

D.DELETE 生产 WHERE 生产日期 < '2022/6/1'

(2) 有如下定义商品表和生产表的语句, 请补全空白部分。

CREATE TABLE 商品(商品号 (1) C.char(10) (2) D.primary key, 商品名 (3) B.varchar(20) (4) E.not null, 类别 (5) M.nchar(3) (6) F.check (类别 IN('手机', '笔记本')), 单价 int)

CREATE TABLE 生产(商品号 char(10), 厂家编号 (7) N.char(20), 生产日期 (8) G.date, 生产数量 int, (9) K.foreign key (商品号) (10) I.references 商品(商品号))

(3) 查询“手机”类商品的商品名和单价。下列语句正确的是 ()。

A.SELECT 商品名, 单价 FROM 商品 WHERE 类别 = '手机'

B.SELECT 商品名, 单价 FROM 商品 WHERE 类别 LIKE '手机'

(4) 查询单价大于 4000 的“笔记本”类商品的商品名和单价。下列语句正确的是 ()。

B.SELECT 商品名, 单价 FROM 商品 WHERE 单价 > 4000 AND 类别 = '笔记本'

C.SELECT 商品名, 单价 FROM 商品 WHERE 单价 > 4000 AND 类别 IN ('笔记本')

(5) 现要创建满足如下要求的视图: 查询生产“手机”类商品的厂家名和厂家地址。请补全空白部分 (设视图名为: V1)。

CREATE (1) C.VIEW V1 AS SELECT (2) B.厂家名, 厂家地址 FROM 生产厂家 a JOIN (3) E.生产 b ON (4) A.a. 厂家编号 = b. 厂家编号 JOIN 商品 c ON c. 商品号 = b. 商品号 WHERE (5) D.类别 = '手机'

(6) 在商品表中插入一行数据, 商品号: SP100, 商品名: 新商品, 类别: 手机, 单价未知。下列语句正确的是 ()。

A.INSERT INTO 商品(商品号, 商品名, 类别) VALUES('SP100', '新商品', '手机')

B.INSERT INTO 商品 VALUES('SP100', '新商品', '手机', NULL)

C.INSERT INTO 商品(商品号, 商品名, 类别, 单价) VALUES('SP100', '新商品', '手机', NULL)

10、设有描述商品生产的如下三张表, 各表结构如下:

“商品”表结构:

商品号: 普通编码定长字符类型, 长度为 10, 主键。

商品名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 3, 取值为: “手机”、“笔记本”。

单价: 整型。

“生产厂家”表结构:

厂家编号: 普通编码定长字符类型, 长度为 20, 主键。

厂家名: 普通编码可变长字符类型, 长度为 30, 非空。

厂家地址: 统一字符编码可变长字符类型, 长度为 40。

“生产”表结构:

商品号: 普通编码定长字符类型, 长度为 10, 引用商品表的外键。

厂家编号: 普通编码定长字符类型, 长度为 20, 引用生产厂家表的外键。

生产日期: 日期类型。

生产数量: 整型, 取值大于等于 100。

主键: (商品号, 厂家编号, 生产日期)。

(1) 有如下定义生产厂家表和生产表的语句, 请补全空白部分。

CREATE TABLE 生产厂家(

厂家编号 char(20) (1) [D] primary key,

厂家名(2) [B] varchar(30) (3) [E] not null,

厂家地址(4) [C] nvarchar(40)

)

CREATE TABLE 生产(

商品号 char(10),

厂家编号 char(20),

生产日期 (5) [G] date,

生产数量 (6) [N] int (7) [F] check (生产数量(8) [M] >=100),

(9) [I] foreign key (厂家编号) (10) [K] references 生产厂家(厂家编号)

(2) 【多选题】

1) 查询单价在 1000-2000 (包括边界值) 的商品的商品名、类别和单价, 下列语句正确的是 ()。

[A.] SELECT 商品名, 类别, 单价 FROM 商品 WHERE 单价 BETWEEN 1000 AND 2000

[C.] SELECT 商品名, 类别, 单价 FROM 商品 WHERE 单价 >= 1000 AND 单价 <= 2000

2) 统计“手机”类商品数量。下列语句正确的是 ()。

[A.] SELECT COUNT(商品号) FROM 商品 WHERE 类别 = '手机'

[B.] SELECT COUNT(*) FROM 商品 WHERE 类别 = '手机'

3) 将“手机”类商品的单价提高 5%。下列语句正确的是 ()。

[A.] UPDATE 商品 SET 单价 = 单价 * 1.05 WHERE 类别 = '手机'

[B.] UPDATE 商品 SET 单价 = 单价 + 单价 * 0.05 WHERE 类别 = '手机'

4) 删除厂家地址的前两个字不是“北京”的生产厂家。下列语句错误的是 ()。

[A.] DELETE FROM 生产厂家 WHERE 厂家地址 NOT = '北京'

[B.] DELETE FROM 生产厂家 WHERE 厂家地址 NOT LIKE '北京'

[C.] DELETE FROM 生产厂家 WHERE 厂家地址 NOT = '北京%'

(3) 现要创建满足如下要求的视图: 查询 2022 年 1 月 1 日及之后生产的商品名、类别、生产厂家名和生产日期。请补全空白部分 (设视图名为: V1)。

CREATE (1) [E] VIEW V1 AS

SELECT (2) [B] 商品名, 类别, 厂家名, 生产日期 FROM (3) [C] 生产

a JOIN 商品 b ON (4) [A] a. 商品号 = b. 商品号 JOIN 生产厂家 c ON

c. 厂家编号 = b. 厂家编号 WHERE (5) [D] 生产日期 >= '2022-1-1'

11、设有描述商品生产的如下三张表, 各表结构如下:

“商品”表结构:

商品号: 普通编码定长字符类型, 长度为 10, 主键。

商品名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 4, 取值为: “家用电器”、“手机”。

单价: 整型。

“生产厂家”表结构:

厂家编号: 普通编码定长字符类型, 长度为 20, 取值不重。

厂家名: 普通编码可变长字符类型, 长度为 30。

厂家地址: 统一字符编码可变长字符类型, 长度为 30。

“生产”表结构:

商品号: 普通编码定长字符类型, 长度为 10。

厂家编号: 普通编码定长字符类型, 长度为 20。

生产日期: 日期类型。

生产数量: 整型。

(1) 查询生产数量为 100、200、300 的商品号和生产数量, 下列语句正确的是 ()。

B.SELECT 商品号, 生产数量 FROM 生产 WHERE 生产数量 in (100, 200, 300)

C.SELECT 商品号, 生产数量 FROM 生产 WHERE 生产数量 = 100 OR 生产数量 = 200 OR 生产数量 = 300

(2) 删除 2010 年 1 月 1 日之前的生产记录。下列语句正确的是 ()。

A.DELETE FROM 生产 WHERE 生产日期 < '2010-1-1'

B.DELETE 生产 WHERE 生产日期 < '2010-1-1'

(3) 统计每类商品的平均单价。下列语句错误的是 ()。

A.SELECT 类别, AVG(*) FROM 商品 GROUP BY 类别

C.SELECT 类别, AVG(*) FROM 商品 ORDER BY 类别

D.SELECT 类别, AVG(单价) FROM 商品 ORDER BY 类别

(4) 将“手机”类商品的单价降低 100。下列语句错误的是 ()。

B.UPDATE 单价 SET 单价 = 单价 - 100 WHERE 类别 = '手机'

C.ALTER 商品 SET 单价 = 单价 - 100 WHERE 类别 = '手机'

D.ALTER 单价 SET 单价 = 单价 - 100 WHERE 类别 = '手机'

(5) 有如下定义商品表和生产厂家表的语句, 请补全空白部分。CREATE TABLE 商品(商品号 (1) B.char(10) (2) D.primary key, 商品名 (3) C.varchar(20) (4) E.not null, 类别 (5) K.nchar(4) check (类别 (6) Fin ('家用电器', '手机'), 单价 (7) N.int) CREATE TABLE 生产厂家(厂家编号 char(20) (8) M.unique, 厂家名 (9) I.varchar(30), 厂家地址 (10) G.nvarchar)

(6) 现要创建满足如下要求的视图: 查询单价小于 4000 的“手机”类商品的名称、单价和生产厂家。请补全空白部分(设视图名为: V1)。CREATE (1) B.VIEW V1 AS SELECT (2) E.商品名, 单价, 生产厂家 FROM 生产厂家 a JOIN 生产 b ON (3) C.a. 厂家编号 = b. 厂家编号 JOIN 商品 c ON (4) A.b. 商品号 = c. 商品号 WHERE (5) D.类别 = '手机' AND 单价 < 4000

12、设有描述图书出版情况的如下三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 20, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

出版年份: 小整型。

印刷数量: 整型, 默认值为: 1000。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“作者”表结构:

作者号: 普通编码定长字符类型, 长度为 11, 主键。

作者名: 普通编码可变长字符类型, 长度为 20, 非空。

职称: 统一字符编码定长字符类型, 长度为 2, 取值为: “高级”、“中级”、“初级”。

“出版”表结构:

书号: 普通编码定长字符类型, 长度为 20, 引用图书表的外键。

作者号: 普通编码定长字符类型, 长度为 11, 引用作者表的外键。

作者序号: 整型。

主键: (书号, 作者号)

(1) 查询“高级”职称作者人数。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 作者 WHERE 职称 = '高级'

D.SELECT COUNT(作者号) FROM 作者 WHERE 职称 = '高级'

(2) 查询 2020 年—2022 年(包括边界值)出版的图书的书名、出版年份和单价。下列语句正确的是 ()。

A.SELECT 书名, 出版年份, 单价 FROM 图书 WHERE 出版年份 BETWEEN 2020 AND 2022

C.SELECT 书名, 出版年份, 单价 FROM 图书 WHERE 出版年份 IN (2020, 2021, 2022)

(3) 统计每年印刷的图书总数量。下列语句错误的是 ()。

B.SELECT 出版年份, SUM(印刷数量) FROM 图书 ORDER BY 出版年份

C.SELECT 出版年份, COUNT(印刷数量) FROM 图书 GROUP BY 出版年份

D.SELECT 出版年份, COUNT(印刷数量) FROM 图书 ORDER BY 出版年份

(4) 现要创建满足如下要求的视图: 查询单价在 50-80 之间(包括边界值)的图书的书名、作者名、出版年份和单价。请补全空白部分(设视图名为: V1)。CREATE VIEW V1 AS SELECT (1) 【C】 书名, 作者名, 出版年份, 单价 FROM 图书 JOIN 出版 ON (2) 【A】 图书. 书号 = 出版. 书号 JOIN 作者 ON (3) 【E】 作者. 作者号 = 出版. 作者号 WHERE (4) 【D】 单价 BETWEEN 50 (5) 【B】 AND 80

(5) 删除“作者序号”大于 2 的出版记录。下列语句正确的是 ()。

A.DELETE FROM 出版 WHERE 作者序号 > 2

C.DELETE 出版 WHERE 作者序号 > 2

(6) 有如下定义图书表和作者表的语句, 请补全空白部分。CREATE TABLE 图书(书号 (1) 【B】 char(20) primary key, 书名 (2) 【C】 varchar(30) not null, 出版年份 (3) 【E】 smallint, 印刷数量 int (4) 【K】 default 1000, 单价 (5) 【M】 numeric(6, 2))

CREATE TABLE 作者(作者号 (6) 【H】 char(11) primary key, 作者名 (7) 【F】 varchar(20) not null, 职称 (8) 【A】 nchar(2) check ((9) 【I】 职称 (10) 【N】 in ('高级', '中级', '初级')) CREATE TABLE 出版(

13、设有描述图书出版情况的如下三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 20, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

出版年份: 小整型。

印刷数量: 整型, 取值大于等于 1000。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“作者”表结构:

作者号: 普通编码定长字符类型, 长度为 11, 主键。

作者名: 普通编码可变长字符类型, 长度为 30, 非空。

职称: 统一字符编码定长字符类型, 长度为 2, 取值为: “高级”、“中级”、“初级”。

“出版”表结构:

书号: 普通编码定长字符类型, 长度为 20, 引用图书表的外键。

作者号: 普通编码定长字符类型, 长度为 11, 引用作者表的外键。

作者序号: 整型。

主键: (书号, 作者号)

(1) 现要创建满足如下要求的视图: 查询“高级”和“中级”职称作者出版的图书的书名、作者名、出版年份和单价。请补全空白部分(设视图名为: V1)。

CREATE VIEW V1 AS SELECT (1) A. 书名, 作者名, 出版年份, 单价 FROM 图书 JOIN 出版 ON (2) C. 图书. 书号 = 出版. 书号 JOIN 作者 ON (3) B. 作者. 作者号 = 出版. 作者号 WHERE (4) E. 职称 IN ((5) D. '高级', '中级')

(2) 查询“初级”和“中级”职称的作者名和职称。下列语句正确的是 ()。

A.SELECT 作者名, 职称 FROM 作者 WHERE 职称 IN ('初级', '中级')

D.SELECT 作者名, 职称 FROM 作者 WHERE 职称 = '初级' OR 职称 = '中级'

(3) 将 2022 年及之后出版的图书的单价增加 10%。下列语句正确的是 ()。

A.UPDATE 图书 SET 单价 = 单价 * 1.1 WHERE 出版年份 >= 2022

B.UPDATE 图书 SET 单价 = 单价 + 单价 * 0.1 WHERE 出版年份 >= 2022

(4) 删除 2010 年之前出版的图书记录。下列语句正确的是 ()。

A.DELETE FROM 图书 WHERE 出版年份 <= 2010

B.DELETE 图书 WHERE 出版年份 <= 2010

(5) 查询“中级”职称的作者人数。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 作者 WHERE 职称 = '中级'

B.SELECT COUNT(作者号) FROM 作者 WHERE 职称 = '中级'

C.SELECT COUNT(作者名) FROM 作者 WHERE 职称 = '中级'

(6) 有如下定义作者表和出版表的语句, 请补全空白部分。

CREATE TABLE 作者(作者号 (1) H.char(11) (2) D.primary key, 作者名 (3) C.varchar(30) (4) B.char(20), 职称 (5) A.nchar(2) check(职称 (6) N.in ('高级', '中级', '初级'))

CREATE TABLE 出版(书号 (7) B.char(20), 作者号 char(11), 作者序号 int, Primary key (8) G.(书号, 作者号), (9) K.foreign key (书号) (10) I.references 图书(书号))

14、设有描述图书出版情况的如下三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 20, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

出版年份: 小整型。

印刷数量: 整型, 取值大于等于 1000。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“作者”表结构:

作者号: 普通编码定长字符类型, 长度为 11, 主键。

作者名: 普通编码可变长字符类型, 长度为 30, 非空。

职称: 统一字符编码定长字符类型, 长度为 2, 取值为: “高级”、“中级”、“初级”。

“出版”表结构:

书号: 普通编码定长字符类型, 长度为 20, 引用图书表的外键。

作者号: 普通编码定长字符类型, 长度为 11, 引用作者表的外键。

作者序号: 整型。

主键: (书号, 作者号)

(1) 查询单价在 50-80 之间 (包括边界值) 的图书的书名和单价。下列语句正确的是 ()。

[B.SELECT 书名, 单价 FROM 图书 WHERE 单价 BETWEEN 50 AND 80](#)

[D.SELECT 书名, 单价 FROM 图书 WHERE 单价 >= 50 AND 单价 <= 80](#)

(2) 在出版表中插入一行数据: 书号为“B100”, 作者号为“W200”, 作者序号未知。下列语句正确的是 ()。

[B.INSERT INTO 出版 VALUES\('B100', 'W200', NULL\)](#)

[C.INSERT INTO 出版\(书号, 作者号\) VALUES\('B100', 'W200'\)](#)

(3) 现要创建满足如下要求的视图: 查询“张山”和“李思”两个作者出版的图书的书名、作者名、出版年份。请补全空白部分 (设视图名为: V1)。

CREATE VIEW V1 AS SELECT (1) [C.书名, 作者名, 出版年份](#) FROM 图书 JOIN 出版 ON (2) [A.图书. 书号 = 出版. 书号](#) JOIN 作者 ON (3) [B.作者. 作者号 = 出版. 作者号](#) WHERE (4) [E.作者名](#) (5) [D.IN\('张山', '李思'\)](#)

(4) 有如下定义作者表和出版表的语句, 请补全空白部分。

CREATE TABLE 作者(作者号 (1) [H.char\(11\)](#) (2) [D.primary key](#), 作者名 (3) [C.varchar\(30\)](#) (4) [E.not null](#), 职称 (5) [A.nchar\(2\)](#) check((6) [I.职称](#) in ('高级', '中级', '初级'))

CREATE TABLE 出版(书号 (7) [B.char\(20\)](#), 作者号 char(11), 作者序号 int, Primary key (8) [G.\(书号, 作者号\)](#), (9) [K.foreign key](#) (书号) references 图书((10) [N.书号](#)))

(5) 查询 2022 年出版的图书总数量。下列语句正确的是 ()。

[A.SELECT COUNT\(书名\) FROM 图书 WHERE 出版年份 = 2022](#)

[B.SELECT COUNT\(书号\) FROM 图书 WHERE 出版年份 = 2022](#)

(6) 删除“初级”职称的作者记录。下列语句正确的是 ()。

[A.DELETE FROM 作者 WHERE 职称 = '初级'](#)

[C.DELETE FROM 作者 WHERE 职称 LIKE '初级'](#)

15、设有描述图书出版情况的如下三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 20, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

出版年份: 小整型。

印刷数量: 整型, 取值大于等于 1000。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“作者”表结构:

作者号: 普通编码定长字符类型, 长度为 11, 主键。

作者名: 普通编码可变长字符类型, 长度为 30, 非空。

职称: 统一字符编码定长字符类型, 长度为 2, 取值为: “高级”、“中级”、“初级”。

“出版”表结构:

书号: 普通编码定长字符类型, 长度为 20, 引用图书表的外键。

作者号: 普通编码定长字符类型, 长度为 11, 引用作者表的外键。

作者序号: 整型。

主键: (书号, 作者号)

(1) 在作者表中插入一行数据: 作者号为“z350”, 作者名为“张三”, 职称未知。下列语句正确的是 ()。

[B.INSERT INTO 作者 VALUES\('z350', '张三', NULL\)](#)

[D.INSERT INTO 作者\(作者号, 作者名, 职称\) VALUES\('z350', '张三', NULL\)](#)

(2) 统计“初级”职称的作者人数。下列语句正确的是 ()。

[C.SELECT COUNT\(作者号\) FROM 作者 WHERE 职称 = '初级'](#)

[D.SELECT COUNT\(作者名\) FROM 作者 WHERE 职称 = '初级'](#)

(3) 有如下定义图书表和出版表的语句, 请补全空白部分。

CREATE TABLE 图书(书号 (1) [B.char\(20\)](#) (2) [D.primary key](#), 书名 (3) [C.varchar\(30\)](#) (4) [E.not null](#), 出版年份 (5) [A.smallint](#), 印刷数量 int (6) [F.check](#) (印刷数量 >= 1000), 单价 (7) [M.numeric\(6,2\)](#))

CREATE TABLE 出版(书号 char(20), 作者号 char(11), 作者序号 int, Primary key (8) [G.\(书号, 作者号\)](#), (9) [K.foreign key](#) (书号) (10) [I.references](#) 图书(书号))

(4) 查询 2014 和 2015 年出版的图书的书名和单价。下列语句正确的是 ()。

[A.SELECT 书名, 单价 FROM 图书 WHERE 出版年份 = 2014 OR 出版年份 = 2015](#)

[C.SELECT 书名, 单价 FROM 图书 WHERE 出版年份 IN \(2014, 2015\)](#)

(5) 现要创建满足如下要求的视图: 查询单次印刷数量超过 6000 的图书的书名、作者名、出版年份和印刷数量。请补全空白部分 (设视图名为: V1)。

CREATE (1) [E.VIEW V1 AS](#) SELECT (2) [A.书名, 作者名, 出版年份, 印刷数量](#) FROM 图书 JOIN 出版 ON (3) [B.图书. 书号 = 出版. 书号](#) JOIN 作者 ON (4) [C.作者. 作者号 = 出版. 作者号](#) WHERE (5) [D.印刷数量 > 6000](#)

(6) 将 2013 年之前 (不包括 2013 年) 出版的图书的单价降低 10%。下列语句正确的是 ()。

[A.UPDATE 图书 SET 单价 = 单价 * 0.9 WHERE 出版年份 < 2013](#)

[B.UPDATE 图书 SET 单价 = 单价 - 单价 * 0.1 WHERE 出版年份 < 2013](#)

16、设有描述图书出版情况的如下三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 20, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

出版年份: 小整型。

印刷数量: 整型, 取值大于等于 1000。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“作者”表结构:

作者号: 普通编码定长字符类型, 长度为 11, 主键。

作者名: 普通编码可变长字符类型, 长度为 20, 非空。

职称: 统一字符编码定长字符类型, 长度为 2, 取值为: “高级”、“中级”、“初级”。

“出版”表结构:

书号: 普通编码定长字符类型, 长度为 20, 引用图书表的外键。

作者号: 普通编码定长字符类型, 长度为 11, 引用作者表的外键。

作者序号: 整型。

主键: (书号, 作者号)

(1) 有如下定义图书表和出版表的语句, 请补全空白部分。CREATE TABLE 图书(书号 char(20) (1) [A.primary key](#), 书名 (2) [C.varchar\(30\)](#) not null, 出版年份 (3) [E.smallint](#), 印刷数量 int check ((4) [K.印刷数量](#) >= 1000), 单价 (5) [D.numeric\(6,2\)](#))

CREATE TABLE 出版(书号 char(20), 作者号 (6) [H.char\(11\)](#), 作者序号 (7) [F.int](#), (8) [I.foreign key](#) (书号) (9) [G.references](#) (10) [N.图书](#)(书号))

[B.作者 M.numeric\(4,2\)](#)

(2) 现要创建满足如下要求的视图: 查询作者序号为 1 的“高级”职称作者编写的的图书的书名、作者名、出版年份和单价。请补全空白部分 (设视图名为: V1)。

CREATE (1) [D.VIEW V1 AS](#) SELECT (2) [C.书名, 作者名, 出版年份, 单价](#) FROM 图书 a JOIN (3) [A.出版](#) b ON a. 书号 = b. 书号 JOIN 作者 c ON (4) [E.c. 作者号 = b. 作者号](#) WHERE 作者序号 = 1 AND (5) [B.职称 = '高级'](#)

(3) 查询 2022 年出版的印刷数量大于等于 5000 的图书的书名、印刷数量和单价。下列语句正确的是 ()。

[B.SELECT 书名, 印刷数量, 单价 FROM 图书 WHERE 出版年份 = 2022 AND 印刷数量 >= 5000](#)

[D.SELECT 书名, 印刷数量, 单价 FROM 图书 WHERE 印刷数量 >= 5000 AND 出版年份 = 2022](#)

(4) 删除印刷数量等于 1000 的图书记录。下列语句正确的是 ()。

[A.DELETE FROM 图书 WHERE 印刷数量 = 1000](#)

[C.DELETE 图书 WHERE 印刷数量 = 1000](#)

(5) 更改作者“张三”的职称为“高级”。下列语句错误的是 ()。

[B.ALTER 作者 SET 职称 = '高级' WHERE 作者名 = '张三'](#)

C.UPDATE 职称 SET 职称 = '高级' WHERE 作者名 = '张三'
D.ALTER 职称 SET 职称 = '高级' WHERE 作者名 = '张三'

(6) 查询 2022 年出版的图书的最高单价。下列语句错误的是 ()。

B.SELECT MIN(单价) FROM 图书 WHERE 出版年份 = 2022
C.SELECT MAX(单价) FROM 图书 WHERE 出版年份 IS 2022
D.SELECT MIN(单价) FROM 图书 WHERE 出版年份 IS 2022

17、设有图书、书店、销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 10，取值不重。

书名：普通编码可变长字符类型，长度为 30，非空。

类别：统一字符编码定长字符类型，长度为 4，取值范围：{'高等数学','计算机'}。

出版日期：日期。

单价：定点小数类型，整数部分 4 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 10。

书店编号：普通编码定长字符类型，长度为 20，引用“书店”表的外键。

销售时间：日期时间类型。

销售数量：整型。

主键：(书号，书店编号，销售时间)。

(1) 删除 2000 年 1 月 1 日之前的图书销售记录。下列语句正确的是 ()。

A.DELETE 销售 WHERE 销售时间 < '2000-1-1'
B.DELETE 销售 WHERE 销售时间 < '2000/1/1'
C.DELETE 销售 WHERE 销售时间 < '2000-01-01'
D.DELETE 销售 WHERE 销售时间 < '2000/01/01'

(2) 有如下定义图书表和销售表的语句，请补全空白部分。

CREATE TABLE 图书(书号 char(10) (1) I.unique , 书名 (2) C.varchar(30) (3) E.not null , 类别 (4) B.nchar(4) (5) F.check ('高等数学','计算机'), 出版日期 date, 单价 (7) M.numeric(6,2)

CREATE TABLE 销售(书号 char(10), 书店编号 char(20), 销售时间 (8) N.datetime, 销售数量 int (9) H.foreign key (书号) references (10) D.图书(书号))

(3) 向“书店”表插入一行数据，书店编号为：SD100，书店名为：北京新书店，地址未知。下列语句正确的是 ()。

A.INSERT INTO 书店 VALUES('SD100','北京新书店',NULL)
C.INSERT INTO 书店(书店编号,书店名) VALUES('SD100','北京新书店')

(4) 统计每类图书的销售总数量。下列语句错误的是 ()。

A.SELECT 类别, SUM(销售数量) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 ORDER BY 类别
B.SELECT 类别, COUNT(销售数量) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 ORDER BY 类别
D.SELECT 类别, COUNT(销售数量) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 GROUP BY 类别

(5) 现要创建满足如下要求的视图：查询“计算机”类图书的销售情况，列出书名、单价、销售时间和销售数量。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) C.书名, 单价, 销售时间, 销售数量 FROM 书店 JOIN (2) B.销售

ON (3) D.书店. 书店编号 = 销售. 书店编号 JOIN 图书 ON (4) E.图书. 书号 = 销售. 书号 WHERE (5) A.类别 = '计算机'

(6) 查询“计算机”类图书的书名、出版日期和单价。下列语句正确的是 ()。

A.SELECT 书名, 出版日期, 单价 FROM 图书 WHERE 类别 = '计算机'
C.SELECT 书名, 出版日期, 单价 FROM 图书 WHERE 类别 IN ('计算机')

18、设有图书、书店、销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 10，主键。

书名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 4，取值为：“经济管理”、“深度学习”。

出版日期：日期。

单价：定点小数类型，整数部分 3 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

成立日期：日期型。

“销售”表结构：

书号：普通编码定长字符类型，长度为 10，引用图书表的外键。

书店编号：普通编码定长字符类型，长度为 20，引用书店表的外键。

销售时间：日期时间类型。

销售数量：整型，取值大于等于 1。

主键：(书号，书店编号，销售时间)。

(1) 向“图书”表插入一行数据，书号为：S01，书名为：NewBook，单价为：49，出版日期和类别未知。下列语句正确的是 ()。

B.INSERT INTO 图书 VALUES('S01','NewBook',NULL,NULL,49)
C.INSERT INTO 图书(书号,书名,单价) VALUES('S01','NewBook',49)

(2) 查询 2022 年 1 月 1 日之后出版的图书的销售情况，列出书号、书名、出版日期、销售时间和销售数量。下列语句正确的是 ()。

B.SELECT a. 书号, 书名, 出版日期, 销售时间, 销售数量 FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 WHERE 出版日期 >= '2022-1-1'
D.SELECT 销售. 书号, 书名, 出版日期, 销售时间, 销售数量 FROM 图书 JOIN 销售 ON 图书. 书号 = 销售. 书号 WHERE 出版日期 >= '2022-1-1'

(3) 删除图书表中类别为 NULL 的数据。下列语句错误的是 ()。

A.DELETE FROM 图书 WHERE 类别 = NULL
C.DELETE FROM 图书 WHERE 类别 IS 'NULL'
D.DELETE FROM 图书 WHERE 类别 = 'NULL'

(4) 查询“经济管理”类图书的书名和出版日期，下列语句正确的是 ()。

A.SELECT 书名, 出版日期 FROM 图书 WHERE 类别 = '经济管理'
B.SELECT 书名, 出版日期 FROM 图书 WHERE 类别 LIKE '经济管理'

(5) 有如下定义书店表和销售表的语句，请补全空白部分。

CREATE TABLE 书店 (书店编号 (1) K.char(20) (2) D.primary key, 书店名 (3) B.varchar(30) (4) E.not null, 地址 (5) A.nvarchar(30), 成立日期 (6) G.date)
CREATE TABLE 销售(书号 char(10), 书店编号 char(20), 销售时间 datetime, 销售数量 (7) I.int (8) F.check (销售数量 >= 1), (9) M.foreign key (书店编号) (10) H.references 图书(书店编号))

(6) 现要创建满足如下要求的视图：查询销售了“数据库原理”图书的书店名和地址。请补全空白部分（设视图名为：V1）。

CREATE (1) C.VIEW V1 AS SELECT (2) B.书店名, 地址 FROM 书店 a JOIN 销售 b ON (3) A.a. 书店编号 = b. 书店编号 JOIN 图书 c ON (4) E.c. 书号 = b. 书号 WHERE (5) D.书名 = '数据库原理'

19、设有图书、书店、销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 10，取值不重。

书名：普通编码可变长字符类型，长度为 30，非空。

类别：统一字符编码定长字符类型，长度为 4，取值范围：{'高等数学','计算机'}。

出版日期：日期。

单价：定点小数类型，整数部分 4 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 10。

书店编号：普通编码定长字符类型，长度为 20，引用“书店”表的外键。

销售时间：日期时间类型。销售数量：整型，取值大于等于 1。

主键：(书号，书店编号，销售时间)。

(1) 查询“高等数学”类图书的书名和单价。下列语句正确的是 ()。

[A.SELECT 书名,单价 FROM 图书 WHERE 类别 = '高等数学'](#)

[C.SELECT 书名,单价 FROM 图书 WHERE 类别 IN \('高等数学'\)](#)

(2) 有如下定义书店表和和销售表的语句，请补全空白部分。

CREATE TABLE 书店(书店编号 char(20) (1) [H.primary key](#), 书店名 (2) [C.varchar\(30\)](#) (3) [E.not null](#), 地址 (4) [A.nvarchar\(30\)](#))

CREATE TABLE 销售(书号 (5) [B.char\(10\)](#), 书店编号 char(20), 销售时间 (6) [N.datetime](#), 销售数量 (7) [K.int](#) (8) [F.check](#) (9) [G.nchar\(10\)](#) (销售数量 >= 1), foreign key(书店编号) (9) [M.references](#) (10) [D.书店](#) (书店编号))

(3) 删除地址为 NULL 的书店。下列语句错误的是 ()。

[A.DELETE 书店 WHERE 地址 = 'NULL'](#)

[B.DELETE 书店 WHERE 地址 IS 'NULL'](#)

[C.DELETE 书店 WHERE 地址 = NULL](#)

(4) 统计每类图书的平均单价。下列语句错误的是 ()。

[A.SELECT 类别, AVG\(书号\) FROM 图书 GROUP BY 类别](#)

[B.SELECT 类别, AVG\(书名\) FROM 图书 GROUP BY 类别](#)

[D.SELECT 类别, AVG\(类别\) FROM 图书 GROUP BY 类别](#)

(5) 现要创建满足如下要求的视图：查询“C 语言”图书的销售情况，列出销售的书店名、销售时间和销售数量。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 AS SELECT (1) [C.书店名, 销售时间, 销售数量](#) FROM 书店 JOIN (2) [B.销售](#) ON (3) [D.书店. 书店编号 = 销售. 书店编号](#) JOIN 图书 ON (4) [A.图书. 书号 = 销售. 书号](#) WHERE (5) [E. 书名 = 'C 语言'](#)

(6) 在销售表中插入一行数据：书号：S60，书店编号：SD05，销售时间：2022-12-12，销售数量：待定。下列语句正确的是 ()。

[A.INSERT INTO 销售 VALUES\('S60','SD05','2022-12-12', NULL\)](#)

[B.INSERT INTO 销售\(书号,书店编号,销售时间\) VALUES\('S60','SD05','2022-12-12'\)](#)

[C.INSERT INTO 销售\(书号,书店编号,销售时间,销售数量\) VALUES\('S60','SD05','2022-12-12', NULL\)](#)

[D.INSERT INTO 销售\(书店编号,销售时间,书号\)VALUES\('SD05','2022-12-12','S60'\)](#)

20、设有图书、书店、销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 20，主键。

书名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 4，取值为：“高等数学”、“量子力学”。

出版日期：日期。

单价：定点小数类型，整数部分 3 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 20，引用图书表的外键。

书店编号：普通编码定长字符类型，长度为 20，引用书店表的外键。

销售时间：日期时间类型。

销售数量：整型。

主键：(书号，书店编号，销售时间)。

(1) 现要创建满足如下要求的视图：查询销售了“高等数学”类图书的书店名和地址。请补全空白部分 (设视图名为：V1)。CREATE (1) [C.VIEW V1 AS](#) SELECT (2) [B.书店名, 地址](#) FROM 书店 JOIN 销售 ON (3) [A.书店. 书店编号 = 销售. 书店编号](#) JOIN 图书 ON (4) [E.销售. 书号 = 图书. 书号](#) WHERE (5) [D.类别 = '高等数学'](#)

(2) 查询“量子力学”类图书的书名和出版日期。下列语句正确的是 ()。

[A.SELECT 书名,出版日期 FROM 图书 WHERE 类别 = '量子力学'](#)

[B.SELECT 书名,出版日期 FROM 图书 WHERE 类别 LIKE '量子力学'](#)

(3) 查询单价大于等于 60 的图书的销售情况，列出书号、书名、类别、销售时间和销售数量。下列语句正确的是 ()。

[B.SELECT a. 书号,书名,类别,销售时间,销售数量 FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 WHERE 单价 >= 60](#)

[C.SELECT b. 书号,书名,类别,销售时间,销售数量 FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 WHERE 单价 >= 60](#)

(4) 有如下定义图书表和和销售表的语句。请补全空白部分。CREATE TABLE 图书(书号 char(20) (1) [D.primary key](#), 书名 (2) [B.varchar\(20\)](#) (3) [E.not null](#), 类别 (4) [M.nchar\(4\)](#) (5) [F.check](#) (类别 IN ('高等数学','量子力学')), 出版日期 date, 单价 (6) [A.numeric\(5,2\)](#)) CREATE TABLE 销售(书号 char(20), 书店编号 char(20), 销售时间 (7) [G.datetime](#), 销售数量 (8) [H.int](#), (9) [K.foreign key](#) (书号) (10) [I.references](#) 图书(书号))

(5) 将“量子力学”类图书的单价降低 10。下列语句错误的是 ()。

[A.UPDATE 图书 SET 单价 - 10 WHERE 类别 = '量子力学'](#)

[C.UPDATE 单价 SET 单价 - 10 WHERE 类别 = '量子力学'](#)

[D.UPDATE 单价 SET 单价 = 单价 - 10 WHERE 类别 = '量子力学'](#)

(6) 统计每类图书的销售总数量。下列语句错误的是 ()。

[A.SELECT 类别,COUNT\(销售数量\) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 GROUP BY 类别](#)

[C.SELECT 类别,COUNT\(销售数量\) FROM 图书 GROUP BY 类别](#)

[D.SELECT 类别,SUM\(销售数量\) FROM 图书 GROUP BY 类别](#)

21、设有图书、书店、销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 20，主键。

书名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 4，取值为：“高等数学”、“量子力学”。

出版日期：日期。

单价：定点小数类型，整数部分 3 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 20，引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。
销售时间: 日期时间类型。
销售数量: 整型。
主键: (书号, 书店编号, 销售时间)。

(1) 有如下定义图书表和销售表的语句。请补全空白部分。

CREATE TABLE 图书(
书号 char(20)(1) 【D.】 primary key,

书名 (2) 【B.】 varchar(20) (3) 【E.】 not null,
类别 (4) 【M.】 nchar(4) (5) 【F.】 check (类别 IN ('高等数学','量子力学')),

出版日期 date,
单价 (6) 【A.】 numeric(5,2)

)

CREATE TABLE 销售(
书号 char(20),
书店编号 char(20),
销售时间(7) 【G.】 datetime,

销售数量(8) 【H.】 int, (9) 【K.】 foreign key (书号) (10) 【I.】 references 图书(书号)

)

(2) 【多选题】

1) 查询“量子力学”类图书的书名和出版日期。下列语句正确的是 ()。

【A.】 SELECT 书名,出版日期 FROM 图书 WHERE 类别 = '量子力学'

【B.】 SELECT 书名,出版日期 FROM 图书 WHERE 类别 LIKE '量子力学'

2) 查询单价大于等于 60 的图书的销售情况, 列出书号、书名、类别、销售时间和销售数量。下列语句正确的是 ()。

【B.】 SELECT a.书号,书名,类别,销售时间,销售数量 FROM 图书 a JOIN 销售 b ON a.书号 = b.书号 WHERE 单价 >= 60

【C.】 SELECT b.书号,书名,类别,销售时间,销售数量 FROM 图书 a JOIN 销售 b ON a.书号 = b.书号 WHERE 单价 >= 60

3) 统计每类图书的销售总数量。下列语句错误的是 ()。

【A.】 SELECT 类别,COUNT(销售数量) FROM 图书 a JOIN 销售 b ON a.书号 = b.书号 GROUP BY 类别

【C.】 SELECT 类别,COUNT(销售数量) FROM 图书 GROUP BY 类别

【D.】 SELECT 类别,SUM(销售数量) FROM 图书 GROUP BY 类别

4) 将“量子力学”类图书的单价降低 10。下列语句错误的是 ()。

【A.】 UPDATE 图书 SET 单价 - 10 WHERE 类别 = '量子力学'

【C.】 UPDATE 单价 SET 单价 - 10 WHERE 类别 = '量子力学'

【D.】 UPDATE 单价 SET 单价 = 单价 - 10 WHERE 类别 = '量子力学'

(3) 现要创建满足如下要求的视图: 查询销售了“高等数学”类图书的书店名和地址。请补全空白部分 (设视图名为: V1)。

CREATE (1) 【C.】 VIEW V1 AS

SELECT (2) 【B.】 书店名,地址 FROM

书店 JOIN 销售 ON (3) 【A.】 书店.书店编号 = 销售.书店编号 JOIN 图书 ON (4) 【E.】 销售.书号 = 图书.书号 WHERE (5) 【D.】 类别 = '高等数学'

22、设有图书、书店、销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 30, 非空。

类别: 统一字符编码定长字符类型, 长度为 4, 取值为: “经济管理”、“软件工程”。

出版日期: 日期。

单价: 定点小数类型, 整数部分 3 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 取值不重。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10。

书店编号: 普通编码定长字符类型, 长度为 20。

销售时间: 日期时间类型。

销售数量: 整型。

主键: (书号, 书店编号, 销售时间)。

(1) 查询 2021 年 1 月 1 日出版的图书的书名和出版日期。下列语句正确的是 ()。

A.SELECT 书名,出版日期 FROM 图书 WHERE 出版日期 = '2021-1-1'

C.SELECT 书名,出版日期 FROM 图书 WHERE 出版日期 IN ('2021-1-1')

(2) 有如下定义图书表和书店表的语句, 请补全空白部分。CREATE TABLE 图书(书号 char(10)

(1) D.primary key, 书名 (2) C.varchar(30) (3) E.not null, 类别 (4) B.nchar(4) (5) F.check (类别

类别 (6) N.in ('经济管理','软件工程'), 出版日期 date, 单价 (7) M.numeric(5,2)) CREATE TABLE 书店(书店编号 (8) H.char(20) (9) I.unique, 书店名 varchar(30) not null, 地址 (10) A.nvarchar(30))

(3) 统计“经济管理”类图书的销售总数量。下列语句正确的是 ()。

A.SELECT SUM(销售数量) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 WHERE 类别 = '经济管理'

C.SELECT SUM(销售数量) FROM 图书 a JOIN 销售 b ON a. 书号 = b. 书号 WHERE 类别 LIKE '经济管理'

(4) 将“经济管理”类图书的单价降低 10%。下列语句正确的是 ()。

B.UPDATE 图书 SET 单价 = 单价 * 0.9 WHERE 类别 = '经济管理'

C.UPDATE 图书 SET 单价 = 单价 * 0.1 WHERE 类别 = '经济管理'

(5) 现要创建满足如下要求的视图: 查询“北京海淀书店”销售的图书的书名、销售时间和销售数量。请补全空白部分 (设视图名为: V1)。CREATE (1) B.VIEW V1 AS SELECT (2) C.书名,销售时间,销售数量 FROM 书店 a JOIN 销售 b ON (3) D.a. 书店编号 = b. 书店编号 JOIN 图书 c ON (4) E.c. 书号 = b. 书号 WHERE (5) A.地址 = '北京海淀书店'

(6) 查询书店地址前 2 个字是“北京”的书店名、地址。下列语句错误的是 ()。

A.SELECT 书店名,地址 FROM 书店 WHERE 地址 = '北京%'

C.SELECT 书店名,地址 FROM 书店 WHERE 地址 = '北京'

D.SELECT 书店名,地址 FROM 书店 WHERE 地址 LIKE '北京'

23、有如下房屋、租客和租赁三张表, 各表结构如下:

“房屋”表结构:

房屋号: 普通编码定长字符类型, 长度为 10, 主键。

房主姓名: 统一编码可变长字符类型, 长度为 10, 非空。

居室数: 整型, 取值范围为 1-4。

地址: 统一编码可变长字符类型, 长度为 30。

月租金: 整型。

“租客”表结构:

租客号: 普通编码定长字符类型, 长度为 18, 主键。

租客名: 统一编码可变长字符类型, 长度为 10, 非空。

联系电话: 普通编码定长字符类型, 长度为 11。

“租赁”表结构:

房屋号: 普通编码定长字符类型, 长度为 10, 引用“房屋”表的外键 a'

租客号: 普通编码定长字符类型, 长度为 18, 引用“租客”表的外键。

起租日期: 日期型。

租期: 整型, 默认值 12。

主键：(房屋号, 租客号, 起租日期)。

178. 写出创建如下三张数据表的 SQL 语句。

“房屋”表结构：

```
CREATE TABLE 房屋(  
房屋号 char(10) primary key,  
房主姓名 nvarchar(10) not null,  
居室数 int CHECK(居室数 between 1 and 4),  
地址 nvarchar(30),  
月租金 int  
)
```

“租客”表结构：

```
CREATE TABLE 租客(  
租客号 char(18) primary key,  
租客名 nvarchar(10) not null,  
联系电话 char(11)  
)
```

“租赁”表结构：

```
CREATE TABLE 租赁(  
房屋号 char(10),  
租客号 char(18),  
起租日期 date,  
租期 int default 12,  
Primary key(房屋号, 租赁号, 起租日期),  
Foreign key(房屋号) references 房屋 (房屋号),  
Foreign key(租客号) references 租客 (租客号)  
)
```

179. 依据第 31 题所创建的三张表，写出完成下列操作的 SQL 语句。

① 查询二居室和三居室的所有房屋信息，列出房屋地址和月租金。

```
SELECT 房屋地址, 月租金 FROM 房屋 WHERE 居室数=2 OR 居室数=3;
```

② 查询租期超过 24 的房屋情况，列出房屋号、地址和租期。

```
SELECT a.房屋号, 地址, 租期 FROM 房屋 a JOIN 租赁 b ON a.房屋号=b.房屋号 WHERE 租期>24;
```

③ 将居室数大于等于 3 的房屋的月租金降低 200。

```
UPDATE 房屋 SET 月租金=月租金-200 WHERE 居室数>=3
```

④ 删除所有姓“张”的租客信息。

```
DELETE FROM 租客 WHERE 租客名 LIKE '张%';
```

评分要点：DELETE 部分 2 分，WHERE 部分 3 分

180. 依据第 31 题所创建的三张表，写出创建满足如下要求的视图的 SQL 语句：查询起租日期在 2019 年 1 月 1 日之后（包括该日期）的三居室房屋的出租情况，列出房屋号、地址、月租金、租客名和租期。

```
CREATE VIEW V1 AS  
SELECT a.房屋号, 地址, 月租金, 租客名, 租期  
FROM 房屋 a JOIN 租赁 b ON a.房屋号=b.房屋号  
JOIN 租客 c ON c.租客号=b.租客号  
WHERE 居室数=3 AND 起租日期>='2019-1-1'
```

24. 有如下房屋、租客和租赁三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 20，非空。

居室数：整型，取值范围为 1-6。

地址：统一编码可变长字符类型，长度为 30。月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11，默认值“1111”。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型，取值大于等于 3。

主键：(房屋号, 租客号, 起租日期)。

(1) 有如下定义租客表和租赁表的语句，请补全空白部分。

```
CREATE TABLE 租客(  
租客号 char(18) (1) 【D】 primary key ,  
租客名(2) 【G】 nvarchar(10) not null ,  
联系电话(3) 【B】 char(11) (4) 【I】 default '1111'  
)
```

```
CREATE TABLE 租赁(  
房屋号(5) 【A】 char(10) ,  
租客号 char(18),  
起租日期(6) 【C】 date ,  
租期(7) 【M】 int (8) 【K】 check (租期 >= 3), foreign key( (9) 【E】 租客号) references (10) 【H】 租客(租客号)  
)
```

(2) 【多选题】

1) 查询姓“张”的房主拥有的房屋的房屋号、居室数、地址。下列语句错误的是 ()。

A. SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 LIKE '张'

C. SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 = '张'

D. SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 = '张%'

2) 统计月租金低于 4000 的房屋数。下列语句正确的是 ()。

A. SELECT COUNT(*) FROM 房屋 WHERE 月租金 < 4000

B. SELECT COUNT(房屋号) FROM 房屋 WHERE 月租金 < 4000

C. SELECT COUNT(房主姓名) FROM 房屋 WHERE 月租金 < 4000

3) 删除月租金为空的房屋信息。下列语句错误的是 ()。

A. DROP 房屋 WHERE 月租金 IS NULL

B. DROP 房屋 WHERE 月租金 = NULL

D. DELETE 房屋 WHERE 月租金 = NULL

4) 在房屋表中插入一行数据，房屋号“F123”，房主名“张房主”，居室数 3，地址：朝阳区 A 街道，月租金未定。下列语句正确的是 ()。

B. INSERT INTO 房屋 VALUES('F123', '张房主', 3, '朝阳区 A 街道', NULL)

C. INSERT INTO 房屋(房屋号, 房主姓名, 居室数, 地址) VALUES('F123', '张房主', 3, '朝阳区 A 街道')

D. INSERT INTO 房屋(房主姓名, 房屋号, 地址, 居室数) VALUES('张房主', 'F123', '朝阳区 A 街道', 3)

(3) 现要创建满足如下要求的视图：查询 3 居室房屋的出租情况，列出房主姓名、租客名和月租金。请补全空白部分（设视图名为：V1）。

```
CREATE VIEW V1 (1) 【E】 AS
```

```
SELECT 房主姓名, 租客名, 月租金
```

```
FROM (2) 【B】 房屋 a JOIN 租赁 b ON a.房屋号 = b.房屋号 JOIN (3) 【A】 租客 c ON (4) 【C】 c.  
租客号 = b.租客号
```

```
WHERE (5) 【D】 居室数 = 3
```

25. 有如下房屋、租客和租赁三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 10，非空。

居室数：整型，取值范围为 1-4。

地址：统一编码可变长字符类型，长度为 30。

月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型，默认值 12。

主键：(房屋号，租客号，起租日期)。

(1) 统计居室数为 2 和 3 的房屋数量。下列语句正确的是 ()。

[A.SELECT COUNT\(房屋号\) FROM 房屋 WHERE 居室数 = 2 OR 居室数 = 3](#)

[B.SELECT COUNT\(房屋号\) FROM 房屋 WHERE 居室数 IN \(2,3\)](#)

[C.SELECT COUNT\(*\) FROM 房屋 WHERE 居室数 = 2 OR 居室数 = 3](#)

[D.SELECT COUNT\(*\) FROM 房屋 WHERE 居室数 IN \(2,3\)](#)

(2) 查询月租金在 5000-6000 (包括边界值) 房屋号、居室数和地址。下列语句正确的是 ()。

[B.SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 月租金 >= 5000 AND 月租金 <= 6000](#)

[D.SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 月租金 BETWEEN 5000 AND 6000](#)

(3) 删除租期小于 3 的租赁记录。下列语句正确的是 ()。

[A.DELETE FROM 租赁 WHERE 租期 < 3](#)

[B.DELETE 租赁 WHERE 租期 < 3](#)

(4) 将房屋号“F123”、租客号“Z321”的租期改为 18。下列语句错误的是 ()。

[A.UPDATE 租赁 SET 租期 = 18 WHERE 房屋号='F123' OR 租客号='Z321'](#)

[C.UPDATE 租期 SET 租期 = 18 WHERE 房屋号='F123' OR 租客号='Z321'](#)

[D.UPDATE 租期 SET 租期 = 18 WHERE 房屋号='F123' AND 租客号='Z321'](#)

(5) 现要创建满足如下要求的视图：查询租客“张三”的租房情况，列出房屋号、居室数、月租金和租期。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 AS SELECT (1) [E.a. 房屋号, 居室数, 月租金, 租期](#) FROM 房屋 a JOIN (2) [A. 租赁](#) b ON a. 房屋号 = b. 房屋号 JOIN (3) [D. 租客](#) c ON (4) [C.c. 租客号 = b. 租客号](#) WHERE (5) [B. 租客名 = '张三'](#)

(6) 有如下定义租客表和租赁表的语句，请补全空白部分。

CREATE TABLE 租客(租客号 char(18) (1) [D.primary key](#), 租客名(2) [G.nvarchar\(10\)](#) (3) [A.not null](#), 联系电话 (4) [F.char\(11\)](#))

CREATE TABLE 租赁(房屋号(5) [M.char\(10\)](#), 租客号 char(18), 起租日期 (6) [C.date](#), 租期 (7) [H.int](#) (8) [I.default](#) 12, (9) [K.foreign key](#) (房屋号) references 房屋((10) [E.房屋号](#)))

26、有如下房屋、租客和租赁三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 10，非空。

居室数：整型，取值范围为 1-4。

地址：统一编码可变长字符类型，长度为 30。

月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型。

主键：(房屋号，租客号，起租日期)。

(1) 统计 2 居室的房屋数。下列语句正确的是 ()。

[A.SELECT COUNT\(房屋号\) FROM 房屋 WHERE 居室数 = 2](#)

[B.SELECT COUNT\(房主姓名\) FROM 房屋 WHERE 居室数 = 2](#)

(2) 将居室数大于等于 3 的房屋的月租金降低 200。下列语句错误的是 ()。

[B.UPDATE 月租金 SET 月租金 = 月租金 - 200 WHERE 居室数 >= 3](#)

[C.ALTER 房屋 SET 月租金 = 月租金 - 200 WHERE 居室数 >= 3](#)

[D.ALTER 月租金 SET 月租金 = 月租金 - 200 WHERE 居室数 >= 3](#)

(3) 现要创建满足如下要求的视图：查询 3 居室房屋的出租情况，列出房屋号、地址、月租金、租客名和租期。请补全空白部分 (设视图名为：V1)。

CREATE (1) [D.VIEW V1 AS](#)

SELECT (2) [E.a. 房屋号, 地址, 月租金, 租客名, 租期](#) FROM 房屋 a

JOIN (3) [A. 租赁](#) b ON a. 房屋号 = b. 房屋号

JOIN 租客 c ON (4) [C.c. 租客号 = b. 租客号](#)

WHERE (5) [B. 居室数 = 3](#)

(4) 删除所有姓“张”的租客信息。下列语句错误的是 ()。

[B.DELETE FROM 租客 WHERE 租客名 LIKE '张 %'](#)

[C.DELETE FROM 租客 WHERE 租客名 = '张 %'](#)

[D.DELETE FROM 租客 WHERE 租客名 = '张 %'](#)

(5) 查询 1 居室和 2 居室的所有房屋信息，列出房屋地址和月租金。下列语句正确的是 ()。

[A.SELECT 地址, 月租金 FROM 房屋 WHERE 居室数 = 1 OR 居室数 = 2](#)

[C.SELECT 地址, 月租金 FROM 房屋 WHERE 居室数 IN \(1, 2\)](#)

[D.SELECT 地址, 月租金 FROM 房屋 WHERE 居室数 BETWEEN 1 AND 2](#)

(6) 有如下定义房屋表和租赁表的语句，请补全空白部分。

CREATE TABLE 房屋(房屋号 char(10) (1) [D.primary key](#), 房主姓名 (2) [G.nvarchar\(10\)](#) (3) [A.not null](#), 居室数 int (4) [F.check](#) (居室数 between 1 and 4), 地址 (5) [N.nvarchar\(30\)](#), 月租金 int)

CREATE TABLE 租赁 (房屋号 char(10), 租客号 (6) [M.char\(18\)](#), 起租日期 (7) [C.date](#), 租期 int, (8) [K.foreign key](#) ((9) [E.房屋号](#)) (10) [I.references](#) 房屋(房屋号))

27、有如下房屋、租客和租赁三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 10，非空。

居室数：整型，取值范围为 1-6。

地址：统一编码可变长字符类型，长度为 30。

月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11，取值不重。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型，默认值 6。

主键：(房屋号，租客号，起租日期)。

(1) 统计 3 居室的最高租金和最低租金。下列语句错误的是 ()。

[B.SELECT MAX\(月租金\),AVG\(月租金\)FROM 房屋 WHERE 居室数=3](#)

[C.SELECT TOP\(月租金\),MIN\(月租金\)FROM 房屋 WHERE 居室数=3](#)

[D.SELECT MAX\(*\),MIN\(*\)FROM 房屋 WHERE 居室数=3](#)

(2) 现要创建满足如下要求的视图：查询月租金超过 8000 的房屋号、居室数、租客名和租期。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 (1) [D.AS](#) SELECT a. 房屋号,居室数,租客名,租期 FROM (2) [B.房屋](#) a JOIN 租赁 b ON a. 房屋号 = b. 房屋号 JOIN (3) [A.租客](#) c ON (4) [C.c. 租客号 = b. 租客号](#) WHERE (5) [E.月租金 > 8000](#)

(3) 有如下定义房屋表和租赁表的语句，请补全空白部分。

CREATE TABLE 房屋 (房屋号 char(10) (1) [D.primary key](#), 房主姓名 (2) [G.nvarchar\(10\)](#) not null, 居室数 int (3) [K.check](#) ((4) [M.居室数](#) between 1 and 6), 地址 (5) [N.nvarchar\(30\)](#), 月租金 int)

CREATE TABLE 租赁 (房屋号 char(10), 租客号 (6) [B.char\(18\)](#), 起租日期 (7) [C.date](#), 租期 int (8) [I.default](#) 6, foreign key(房屋号) references (9) [H.房屋](#) (10) [E.房屋号](#))

(4) 删除租期为 3 的租赁记录。下列语句错误的是 ()。

[A.DROP 租赁 WHERE 租期 = 3](#)

[B.DROP 租期 WHERE 租期 = 3](#)

[D.DELETE 租期 WHERE 租期 = 3](#)

(5) 在租赁表中插入一行数据，房屋号“F123”，租客号“Z111”，起租日期“2022 年 10 月 1 日”，租期未定。下列语句正确的是 ()。

[A.INSERT INTO 租赁 VALUES\('F123', 'Z111', '2022-10-1', NULL\)](#)

[C.INSERT INTO 租赁\(房租号, 租客号, 起租日期\) VALUES\('F123', 'Z111', '2022-10-1'\)](#)

[D.INSERT INTO 租赁\(起租日期, 租客号, 房租号\) VALUES\('2022-10-1', 'Z111', 'F123'\)](#)

(6) 查询地址的前三个字是“朝阳区”的房屋号、居室数和月租金。下列语句错误的是 ()。

[B.SELECT 房屋号, 居室数, 月租金 FROM 房屋 WHERE 地址 = '朝阳区'](#)

[C.SELECT 房屋号, 居室数, 月租金 FROM 房屋 WHERE 地址 LIKE '朝阳区'](#)

[D.SELECT 房屋号, 居室数, 月租金 FROM 房屋 WHERE 地址 = '朝阳区'](#)

28、有如下房屋、租客和租赁三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 20，非空。

居室数：整型，取值范围为 1-4。

地址：统一编码可变长字符类型，长度为 30。

月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11，取值不重。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型，默认值 12。

主键：(房屋号，租客号，起租日期)。

(1)

查询起租日期在 2022 年 6 月 1 日起及之后的房屋出租情况，列出房屋号、起租日期和租期。下列语句正确的是 ()。

[A.SELECT 房屋号, 起租日期, 租期 FROM 租赁 WHERE 起租日期 >= '2022-6-1'](#)

[B.SELECT 房屋号, 起租日期, 租期 FROM 租赁 WHERE 起租日期 >= '2022-06-01'](#)

[C.SELECT 房屋号, 起租日期, 租期 FROM 租赁 WHERE 起租日期 >= '2022/6/1'](#)

[D.SELECT 房屋号, 起租日期, 租期 FROM 租赁 WHERE 起租日期 >= '2022/06/01'](#)

(2) 在租客表中插入一行数据，租客号“Z111”，租客名“张丽”，联系电话未知。下列语句正确的是 ()。

[A.INSERT INTO 租客 VALUES\('Z111', '张丽', NULL\)](#)

[C.INSERT INTO 租客\(租客号, 租客名\) VALUES\('Z111', '张丽'\)](#)

[D.INSERT INTO 租客\(租客号, 租客名, 联系电话\) VALUES\('Z111', '张丽', NULL\)](#)

(3) 现要创建满足如下要求的视图：查询租期在 12-24 的房屋号、居室数、租客名和租期。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 AS SELECT (1) [D.a. 房屋号,居室数,租客名,租期](#) FROM (2) [B.房屋](#) a JOIN 租赁 b ON a. 房屋号 = b. 房屋号 JOIN (3) [A.租客](#) c ON (4) [C.c. 租客号 = b. 租客号](#) WHERE (5) [E.租期 BETWEEN 12 AND 24](#)

(4) 将房屋号为“F123”的房屋的月租金增加 100。下列语句错误的是 ()。

[B.UPDATE 房屋 SET 月租金 = 100 WHERE 房屋号='F123'](#)

[C.ALTER 房屋 SET 月租金 = 月租金 + 100 WHERE 房屋号='F123'](#)

[D.ALTER 房屋 SET 月租金 = 100 WHERE 房屋号='F123'](#)

(5) 有如下定义房屋表和租客表的语句，请补全空白部分。

CREATE TABLE 房屋(房屋号(1) [M.char\(10\)](#) primary key, 房主姓名(2) [G.nvarchar\(20\)](#) not null, 居室数 int (3) [K.check](#) ((4) [H.居室数](#) between 1 and 4), 地址 (5) [N.nvarchar\(30\)](#), 月租金 (6) [C.int](#))

CREATE TABLE 租客(租客号(7) [B.char\(18\)](#) primary key, 租客名(8) [F.nvarchar\(10\)](#)not null , 联系电话 (9) [A.char\(11\)](#) (10) [I.unique](#))

(6) 统计每种居室数的平均月租金。下列语句错误的是 ()。

[A.SELECT 居室数, COUNT\(月租金\)FROM 房屋 GROUP BY 居室数](#)

[C.SELECT 居室数, COUNT\(月租金\)FROM 房屋 ORDER BY 居室数](#)

[D.SELECT 居室数, AVG\(月租金\)FROM 房屋 ORDER BY 居室数](#)

29、有如下教师、课程和授课三张表，各表结构如下：

“教师”表结构：

教师号：普通编码定长字符类型，长度为 10，主键。

教师名：普通编码可变长字符类型，长度为 20，非空。

所在部门：普通编码可变长字符类型，长度为 30。

职称：普通编码定长字符类型，长度为 6，取值为：“教授”、“副教授”、“其他”。

“课程”表结构：

课程号：普通编码定长字符类型，长度为 20，主键。

课程名：普通编码可变长字符类型，长度为 40，非空。

学时数：微整型。

开课学期：微整型

“授课”表结构：

教师号：普通编码定长字符类型，长度为 10，引用教师表的外键。

课程号：普通编码定长字符类型，长度为 20，引用课程表的外键。

授课学时数：整型。

授课年份：整型。

主键：(教师号，课程号，授课年份)。

(1) 写出创建如下三张数据表的 SQL 语句。

“教师”表结构：

[CREATE TABLE 教师\(](#)

[_教师号 char\(10\) primary key,](#)

[_教师名 varchar\(20\) not null,](#)

[_所在部门 varchar\(30\),](#)

[_职称 char\(6\) CHECK \(职称 IN \('教授', '副教授', '其他'\)\)](#)

[\)](#)

“课程”表结构：

[CREATE TABLE 课程\(](#)

[_课程号 char\(20\) primary key,](#)

课程名 varchar(40) not null.
学时数 tinyint,
开课学期 tinyint
)

“授课”表结构:

```
CREATE TABLE 授课(  
    教师号 char(10),  
    课程号 char(20),  
    授课学时 int,  
    授课年份 int,  
    Primary key(教师号, 课程号, 授课年份),  
    Foreign key(教师号)references 教师 (教师号),  
    Foreign key(课程号)references 课程 (课程号)  
)
```

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

①查询教授职称的教师姓名和所在部门。

```
SELECT 教师名, 所在部门 FROM 教师 WHERE 职称='教授'
```

②查询每学期开设的学时数大于 40 的课程门数。

```
SELECT 开课学期, COUNT(*)课程门数 FROM 课程 WHERE 学时数>40 GROUP BY 开课学期
```

③查询 2019 年全部课程的授课情况, 列出课程名、学时数、教师名和授课学时数。

```
SELECT 课程名, 学时数, 教师名, 授课学时  
FROM 课程 JOIN 授课 ON 课程. 课程号=授课. 课程号  
JOIN 教师 ON 教师. 教师号=授课. 教师号  
WHERE 授课年份=2019
```

④删除没人讲授的课程。

```
DELETE FROM 课程 WHERE 课程号 NOT IN (SELECT 课程号 FROM 授课)
```

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询 2019 授课年份全体授课教师的教师名、所授的课程名和授课学时数。

```
CREATE VIEW VI AS  
SELECT 教师名, 课程名, 授课学时  
FROM 授课 JOIN 教师 ON 授课. 教师号=教师. 教师号  
JOIN 课程 ON 课程. 课程号=授课. 课程号  
WHERE 授课年份=2019
```

30、有如下教师、课程和授课三张表, 各表结构如下:

“教师”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 主键。
教师名: 普通编码可长字符类型, 长度为 10, 非空。
职称: 统一字符编码定长字符类型, 长度为 3, 取值为: “教授”、“副教授”、“讲师”。
所在部门: 普通编码可长字符类型, 长度为 20。

“课程”表结构:

课程号: 普通编码定长字符类型, 长度为 20, 主键。
课程名: 普通编码可长字符类型, 长度为 30, 非空。
开课学期: 微整型。
学时: 定点小数类型, 小数点前 2 位, 小数点后 1 位

“授课”表结构:

教师号: 普通编码定长字符类型, 长度为 10, 引用教师表的外键。

课程号: 普通编码定长字符类型, 长度为 20, 引用课程表的外键。

授课学年: 普通编码定长字符类型, 长度为 6, 非空。

主键: (教师号, 课程号, 授课学年)。

(1) 写出创建如下三张数据表的 SQL 语句。

“教师”表结构:

```
CREATE TABLE 教师(  
    教师号 char(10) primary key,  
    教师名 varchar(10) not null,  
    职称 nchar(3) CHECK (类别 IN('教授','副教授','讲师')),  
    所在部门 varchar(20),  
)
```

“课程”表结构:

```
CREATE TABLE 课程(  
    课程号 char(20) primary key,  
    课程名 varchar(30) not null,  
    开课学期 tinyint,  
    学时 numeric(3,1)  
)
```

“授课”表结构:

```
CREATE TABLE 授课(  
    教师号 char(10),  
    课程号 char(20),  
    授课学年 char(6) not null,  
    Primary key(教师号, 课程号, 授课学年),  
    Foreign key(教师号) references 教师(教师号),  
    Foreign key(课程号) references 课程(课程号)  
)
```

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

①查询全部有“教授”职称的教师名和所在部门。

```
SELECT 教师名, 所在部门 FROM 教师 WHERE 职称='教授'
```

②查询学时大于等于 30 的课程的授课情况, 列出课程名、学时和授课学年。

```
SELECT 课程名, 学时, 授课学年  
FROM 课程 a JOIN 授课 b ON a.课程号 = b.课程号  
WHERE 学时 >= 30
```

③删除“讲师”职称的授课记录

```
DELETE 授课  
FROM 教师 a JOIN 授课 b ON a.教师号 = b.教师号  
WHERE 职称='讲师'
```

④在课程表中插入一行数据, 课程号: K100, 课程名: 新课程, 开课学期: 未定, 学时: 24。

```
INSERT INTO 课程  
VALUES('K100','新课程',NULL,24)
```

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询“计算机学院”全体教师的授课情况, 列出教师名、讲授的课程名及授课学年。

```
CREATE VIEW VI AS  
SELECT 教师名, 课程名, 授课学年 FROM 课程 a  
JOIN 授课 b ON a.课程号 = b.课程号  
JOIN 教师 c ON c.教师号 = b.教师号  
WHERE 所在部门='计算机学院'
```

31、有如下课题、成员和参与三张表, 各表结构如下:

“课题”表结构:

课题号: 普通编码定长字符类型, 长度为 10, 主键。
课题名: 统一编码可变长字符类型, 长度为 30, 非空。
经费: 整型。
年限: 整型, 大于等于 1。

“成员”表结构:

成员号: 普通编码定长字符类型, 长度为 18, 主键。

成员名: 统一编码可变长字符类型, 长度为 10, 非空。

专长: 统一编码定长字符类型, 长度为 4, 取值为 (“网络架构”, “数据分析”)。

“参与”表结构:

课题号: 普通编码定长字符类型, 长度为 10, 引用“课题”表的外键。

成员号: 普通编码定长字符类型, 长度为 18, 引用“成员”表的外键。

参加日期: 日期型。

主键: (课题号, 成员号)。

(1) 写出创建如下三张数据表的 SQL 语句。

“课题”表结构:

CREATETABLE 课题(课题号 char(10)primary key, 课题名 nvarchar (30) not null, 经费 int, 年限 int check(年限>=1))

“成员”表结构:

CREATE TABLE 成员(成员号 char(18)primary key, 成员名 nvarchar(30)not null, 专长 nchar(4) check(专长 in('网络架构','数据分析'))

“参与”表结构:

CREATE TABLE 参与(课题号 char(10), 成员号 char(18), 参加日期 date, Primary key(课题号, 成员号), Foreign key(课题号)references 课题(课题号), Foreign key(成员号)references 成员(成员号))

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

① 查询经费大于等于 100, 年限在 2-4 的课题详细信息。

SELECT * FROM 课题 WHERE 经费 >= 100 AND 年限 BETWEEN 2 AND 4;

② 查询“数据分析”专长的成员参与的课题情况, 列出成员名、课题号和参加日期。

SELECT 成员名, 课题号, 参加日期 FROM 成员 a JOIN 参与 b ON a.成员号=b.成员号 WHERE 专长='数据分析'

③ 删除没有人参加的课题。

DELETE 课题 WHERE 课题号 NOT IN (SELECT 课题号 FROM 参与)

④ 在课题表中插入一新课题, 课题号: 错误 100, 课题名: 新课堂, 年限: 5, 经费未定。

INSERT INTO 课题 VALUES('错误 100','新课堂',5,NULL)

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询“数据分析”专长的成员参加的年限超过 3 年的课题的情况, 列出成员名、课题名、经费、年限、参加日期。

CREAT EVIEWV1 AS SELECT 成员名, 课题名, 经费, 年限, 参加日期 FROM 课题 a JOIN 参与 b ON a. 课题号=b. 课题号 JOIN 成员 c ON c. 成员号=b. 成员号 WHERE 专长='数据分析'

32、有如下课题、成员和参与三张表, 各表结构如下:

“课题”表结构:

课题号: 普通编码定长字符类型, 长度为 10, 主键。

课题名: 统一编码可变长字符类型, 长度为 30, 非空。

经费: 整型。

年限: 整型, 大于等于 1。

“成员”表结构:

成员号: 普通编码定长字符类型, 长度为 18, 主键。

成员名: 统一编码可变长字符类型, 长度为 10, 非空。

专长: 统一编码定长字符类型, 长度为 4, 取值为{“网络架构”, “数据分析”}。

“参与”表结构:

课题号: 普通编码定长字符类型, 长度为 10, 引用“课题”表的外键。

成员号: 普通编码定长字符类型, 长度为 18, 引用“成员”表的外键。

参加日期: 日期型。

主键: (课题号, 成员号)。

(1) 写出创建如下三张数据表的 SQL 语句。

“课题”表结构:

CREATE TABLE 课题(

课题号 char(10) primary key,

课题名 nvarchar(30) not null,

经费 int,

年限 int check(年限 >=1)

)

“成员”表结构:

CREATE TABLE 成员(

成员号 char(18) primary key,

成员名 nvarchar(30) not null,

专长 nchar(4) check(专长 in ('网络架构','数据分析'))

)

“参与”表结构:

CREATE TABLE 参与(

课题号 char(10),

成员号 char(18),

参加日期 date,

Primary key(课题号, 成员号),

Foreign key(课题号) references 课题(课题号),

Foreign key(成员号) references 成员(成员号)

)

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

① 查询经费大于等于 100, 年限在 2-4 的课题详细信息。

SELECT * FROM 课题 WHERE 经费 >= 100 AND 年限 BETWEEN 2 AND 4

② 查询“数据分析”专长的成员参与的课题情况, 列出成员名、课题号和参加日期。

SELECT 成员名, 课题号, 参加日期 FROM 成员 a JOIN 参与 b ON a. 成员号 = b. 成员号 WHERE 专长 = '数据分析'

③ 删除没有人参加的课题。

DELETE 课题 WHERE 课题号 NOT IN (SELECT 课题号 FROM 参与)

④ 在课题表中插入一新课题, 课题号: X100, 课题名: 新课堂, 年限: 5, 经费未定。

INSERT INTO 课题 VALUES('X100','新课堂',5,NULL)

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询“数据分析”专长的成员参加的年限超过 3 年的课题的情况, 列出成员名、课题名、经费、年限、参加日期。

CREATE VIEW V1 AS

SELECT 成员名, 课题名, 经费, 年限, 参加日期

FROM 课题 a JOIN 参与 b ON a. 课题号 = b. 课题号

JOIN 成员 c ON c. 成员号 = b. 成员号

WHERE 专长 = '数据分析'

33、有如下汽车、经销商和销售三张表, 各表结构如下:

“汽车”表结构:

汽车型号: 普通编码定长字符类型, 长度为 20, 主键。

汽车类别: 普通编码定长字符类型, 长度为 10, 取值为: “小轿车”、“商务车”、“SUV”

生产年份: 整型。

“经销商”表结构:

经销商号: 普通编码定长字符类型, 长度为 20, 主键。

经销商名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 普通编码可变长字符类型, 长度为 50。

“销售”表结构:

经销商号: 普通编码定长字符类型, 长度为 20, 引用经销商表的外键。

汽车型号: 普通编码定长字符类型, 长度为 20, 引用汽车表的外键。

销售时间: 日期时间型。

销售价格: 整型

主键: (经销商号, 汽车型号, 销售时间)。

(1) 写出创建如下三张数据表的 SQL 语句。

“汽车”表结构:

CREATE TABLE 汽车(

汽车型号 char(20) primary key,

汽车类别 char(10) CHECK(汽车类别 in('小轿车','商务车','SUV')),

生产年份 int

)

“经销商”表结构:

CREATE TABLE 经销商(

经销商号 char(20) primary key,

经销商名 varchar(30) not null,

地址 varchar(50)

)

“销售”表结构:

CREATE TABLE 销售(

经销商号 char(20),

汽车型号 char(20),

销售时间 datetime,

销售价格 int,

Primary key(经销商号, 汽车型号, 销售时间),

Foreign key(经销商号) references 经销商(经销商号),

Foreign key(汽车型号) references 汽车(汽车型号)

)

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

①查询 2014 年生产的汽车型号和汽车类别。

SELECT 汽车型号,汽车类别 FROM 汽车 WHERE 生产年份 = 2014

②查询 2013 年卖出的销售价格 30 万元以上 (包括 30 万元) 的汽车的总数量。

SELECT COUNT(*) 销售总数量 FROM 销售 WHERE 销售价格 >= 300000

AND 销售时间 BETWEEN '2013/1/1' AND '2013/12/31'

③将销售价格超过 100 万元的汽车降价 5 万。

UPDATE 销售 SET 销售价格 = 销售价格 - 50000 WHERE 销售价格 > 1000000

④在汽车表中插入一行数据: 汽车型号为“Q100”, 汽车类别为“小轿车”, 生产年份为 2014。

INSERT INTO 汽车 VALUES('Q100','小轿车',2014)

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询销售“小轿车”的经销商的经销商名和地址。

CREATE VIEW V1 AS

SELECT 经销商名,地址

FROM 经销商 JOIN 销售 ON 经销商.经销商号 = 销售.经销商号

JOIN 汽车 ON 汽车.汽车型号 = 销售.汽车型号

WHERE 汽车类别 = '小轿车'

34、有如下汽车、经销商和销售三张表, 各表结构如下:

“汽车”表结构:

汽车型号: 普通编码定长字符类型, 长度为 20, 主键。

汽车类别: 普通编码定长字符类型, 长度为 10, 取值为: “轿车”、“跑车”、“SUV”

生产年份: 整型。

“经销商”表结构:

经销商号: 普通编码定长字符类型, 长度为 20, 主键。

经销商名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 普通编码可变长字符类型, 长度为 50。

“销售”表结构:

经销商号: 普通编码定长字符类型, 长度为 20, 引用经销商表的外键。

汽车型号: 普通编码定长字符类型, 长度为 20, 引用汽车表的外键。

销售时间: 日期时间型。

销售价格: 整型

主键: (经销商号, 汽车型号, 销售时间)。

(1) 写出创建如下三张数据表的 SQL 语句。

“汽车”表结构:

CREATE TABLE 汽车(

汽车型号 char(20) primary key,

汽车类别 char(10) CHECK(汽车类别 in('轿车','跑车','SUV')),

生产年份 int

)

“经销商”表结构:

CREATE TABLE 经销商(

经销商号 char(20) primary key,

经销商名 varchar(30) not null,

地址 varchar(50)

)

“销售”表结构:

CREATE TABLE 销售(

经销商号 char(20),

汽车型号 char(20),

销售时间 datetime,

销售价格 int,

Primary key(经销商号, 汽车型号, 销售时间),

Foreign key(经销商号) references 经销商(经销商号),

Foreign key(汽车型号) references 汽车(汽车型号)

)

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

①查询 2015 年生产的汽车型号和汽车类别。

SELECT 汽车型号,汽车类别 FROM 汽车 WHERE 生产年份 = 2015

②查询 2016 年卖出的销售价格 30 万元以上 (包括 30 万元) 的汽车的总数量。

SELECT COUNT(*) 销售总数量 FROM 销售 WHERE 销售价格 >= 300000

AND 销售时间 BETWEEN '2016/1/1' AND '2016/12/31'

③将销售价格超过 100 万元的汽车降价 5 万。

UPDATE 销售 SET 销售价格 = 销售价格 - 50000 WHERE 销售价格 > 1000000

④在汽车表中插入一行数据: 汽车型号为“Z4”, 汽车类别为“跑车”, 生产年份为 2014。

INSERT INTO 汽车 VALUES('Z4','跑车',2014)

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询销售“SUV”的经销商的经销商名和地址。

CREATE VIEW V1 AS

SELECT 经销商名,地址

FROM 经销商 JOIN 销售 ON 经销商.经销商号 = 销售.经销商号

JOIN 汽车 ON 汽车.汽车型号 = 销售.汽车型号

WHERE 汽车类别 = 'SUV'

35、有如下汽车、生产厂家和销售三张表，各表结构如下：

“房屋”表结构：

房屋号：普通编码定长字符类型，长度为 10，主键。

房主姓名：统一编码可变长字符类型，长度为 20，非空。

居室数：整型，取值范围为 1-6。

地址：统一编码可变长字符类型，长度为 30。

月租金：整型。

“租客”表结构：

租客号：普通编码定长字符类型，长度为 18，主键。

租客名：统一编码可变长字符类型，长度为 10，非空。

联系电话：普通编码定长字符类型，长度为 11，默认值“1111”。

“租赁”表结构：

房屋号：普通编码定长字符类型，长度为 10，引用“房屋”表的外键。

租客号：普通编码定长字符类型，长度为 18，引用“租客”表的外键。

起租日期：日期型。

租期：整型，取值大于等于 3。

主键：(房屋号，租客号，起租日期)。

(1) 有如下定义租客表和租赁表的语句，请补全空白部分。

CREATE TABLE 租客 (租客号 char(18)(1) D.primary key, 租客名 (2) G.nvarchar(10) not null, 联系电话 (3) B.char(11) (4) L.default '1111')

CREATE TABLE 租赁 (房屋号 (5) A.char(10), 租客号 char(18), 起租日期 (6) C.date, 租期 (7) M.int (8) K.check (租期 >= 3), foreign key(9) E.租客号) references (10) H.租客(租客号))

(2) 删除月租金为空的房屋信息。下列语句错误的是 ()。

A.DROP 房屋 WHERE 月租金 IS NULL

B.DROP 房屋 WHERE 月租金 = NULL

D.DELETE 房屋 WHERE 月租金 = NULL

(3) 在房屋表中插入一行数据，房屋号“F123”，房主名“张房主”，居室数 3，地址：朝阳区 A 街道，月租金未定。下列语句正确的是 ()。

B.INSERT INTO 房屋 VALUES('F123', '张房主', 3, '朝阳区 A 街道', NULL)

C.INSERT INTO 房屋(房屋号, 房主姓名, 居室数, 地址) VALUES('F123', '张房主', 3, '朝阳区 A 街道')

D.INSERT INTO 房屋(房主姓名, 房屋号, 地址, 居室数) VALUES('张房主', 'F123', '朝阳区 A 街道', 3)

(4) 统计月租金低于 4000 的房屋数。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 房屋 WHERE 月租金 < 4000

B.SELECT COUNT(房屋号) FROM 房屋 WHERE 月租金 < 4000

C.SELECT COUNT(房主姓名) FROM 房屋 WHERE 月租金 < 4000

(5) 查询姓“张”的房主拥有的房屋的房屋号、居室数、地址。下列语句错误的是 ()。

A.SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 LIKE '张'

C.SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 = '张'

D.SELECT 房屋号, 居室数, 地址 FROM 房屋 WHERE 房主姓名 = '张%'

(6) 现要创建满足如下要求的视图：查询 3 居室房屋的出租情况，列出房主姓名、租客名和月租金。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 (1) E.AS SELECT 房主姓名, 租客名, 月租金 FROM (2) B.房屋 a JOIN 租赁 b ON a. 房屋号 = b. 房屋号 JOIN (3) A.租客 c ON (4) C.c. 租客号 = b. 租客号 WHERE (5) D.居室数 = 3

36、有如下汽车、生产厂家和销售三张表，各表结构如下：

“汽车”表结构：

汽车型号：普通编码定长字符类型，长度为 10，主键。

汽车名：统一编码可变长字符类型，长度为 20，非空。

颜色：统一编码定长字符类型，长度为 2，取值为：“白色”、“红色”、“银灰”。

出厂价格：整型，默认值 5。

“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。

厂家名：统一编码可变长字符类型，长度为 10，非空。

厂址：统一编码可变长字符类型，长度为 20。

联系电话：普通编码定长字符类型，长度为 8。

“销售”表结构：

汽车型号：普通编码定长字符类型，长度为 10，引用“汽车”表的外键。

厂家编号：普通编码定长字符类型，长度为 20，引用“生产厂家”的外键。

销售日期：日期型。

销售价格：整型。

(1) 有如下定义汽车表和生产厂家表的语句，请补全空白部分。

CREATE TABLE 汽车(

汽车型号 (1) [C] char(10) primary key,

汽车名 颜色 (2) [N] nvarchar(20) not null,

(3) [A] nchar(2) (4) [D] check (颜色 (5) [F] in ('白色','红色','银灰')),

出厂价格 int (6) [K] default 5

)

CREATE TABLE 生产厂家(

厂家编号 (7) [G] char(20) primary key,

厂家名 (8) [B] nvarchar(10) not null,

厂址 (9) [N] nvarchar(20),

联系电话 (10) [H] char(8)

)

(2) 多选题

1) 查询厂址前 2 个汉字为“长春”的厂家名和厂址。下列语句错误的是 ()。

[B.] SELECT 厂家名, 厂址, 联系电话 FROM 生产厂家 WHERE 厂址 LIKE '长春'

[C.] SELECT 厂家名, 厂址 FROM 生产厂家 WHERE 厂址 = '长春%'

[D.] SELECT 厂家名, 厂址 FROM 生产厂家 WHERE 厂址 = '长春'

2) 统计 2022 年 6 月的汽车销售总数量。下列语句错误的是 ()。

[A.] SELECT COUNT(销售数量) FROM 销售

WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

[C.] SELECT COUNT(*) FROM 销售

WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

[D.] SELECT SUM(*) FROM 销售

WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

3) 将“银灰”色汽车的出厂价格降低 5%。下列语句正确的是 ()。

[A.] UPDATE 汽车 SET 出厂价格 = 出厂价格 * 0.95 WHERE 颜色 = '银灰'

[B.] UPDATE 汽车 SET 出厂价格 = 出厂价格 - 出厂价格 * 0.05 WHERE 颜色 = '银灰'

[D.] INSERT INTO 销售 VALUES('W100', 'Q110', '2022-3-6', NULL)

[C.] INSERT INTO 销售(厂家编号, 汽车型号, 销售日期) VALUES('Q110', 'W100', '2022-3-6')

[D.] INSERT INTO 销售(汽车型号, 厂家编号, 销售日期) VALUES('W100', 'Q110', '2022-3-6')

(3) 现要创建满足如下要求的视图：查询销售价格 在 80-100 万的汽车名、生产厂家、出厂价格。请补全空白部分 (设视图名为：V1)。

CREATE VIEW V1 AS SELECT (1) [E] 汽车名, 生产厂家, 出厂价格 FROM

汽车 a JOIN 销售 b ON (2) [A] a.汽车型号 = b.汽车型号

JOIN (3) [D] 生产厂家 c ON (4) [B] b.厂家编号 = c.厂家编号

WHERE (5) [C] 销售价格 BETWEEN 80 AND 100

37、有如下汽车、生产厂家和销售三张表，各表结构如下：

“汽车”表结构：

汽车型号：普通编码定长字符类型，长度为 10，主键。

汽车名：统一编码可变长字符类型，长度为 20，非空。
颜色：统一编码定长字符类型，长度为 2，取值为：“白色”、“红色”、“银灰”。
出厂价格：整型。
“生产厂家”表结构：
厂家编号：普通编码定长字符类型，长度为 20，主键。
厂家名：统一编码可变长字符类型，长度为 10，非空。
厂址：统一编码可变长字符类型，长度为 20。
联系电话：普通编码定长字符类型，长度为 8。
“销售”表结构：
汽车型号：普通编码定长字符类型，长度为 10，引用“汽车”表的外键。
厂家编号：普通编码定长字符类型，长度为 20，引用“生产厂家”的外键。
销售日期：日期型。
销售价格：整型。

(1) 统计“红色”汽车的平均出厂价格。下列语句错误的是 ()。

B.SELECT MAX(出厂价格) FROM 汽车 WHERE 颜色 = '红色'
C.SELECT MIN(出厂价格) FROM 汽车 WHERE 颜色 = '红色'
D.SELECT SUM(出厂价格) FROM 汽车 WHERE 颜色 = '红色'

(2) 删除 2000 年 1 月 1 日之前的销售记录。下列语句正确的是 ()。

A.DELETE 销售 WHERE 销售日期 < '2000-1-1'
B.DELETE 销售 WHERE 销售日期 < '2000-01-01'
C.DELETE 销售 WHERE 销售日期 < '2000/1/1'
D.DELETE 销售 WHERE 销售日期 < '2000/01/01'

(3) 在生产厂家表中插入一行数据，厂家编号：C100，厂家名：汽车一厂，厂址：长春，联系电话：未知。下列语句正确的是 ()。

A.INSERT INTO 生产厂家 VALUES('C100', '汽车一厂', '长春', NULL)
C.INSERT INTO 生产厂家(厂家编号, 厂家名, 厂址) VALUES('C100', '汽车一厂', '长春')
D.INSERT INTO 生产厂家(厂家编号, 厂家名, 厂址, 联系电话) VALUES('C100', '汽车一厂', '长春', NULL)

(4) 查询联系电话以“123”开始的生产厂家的厂家名、厂址和联系电话。下列语句错误的是 ()。

B.SELECT 厂家名, 厂址, 联系电话 FROM 生产厂家 WHERE 联系电话 LIKE '123'
C.SELECT 厂家名, 厂址, 联系电话 FROM 生产厂家 WHERE 联系电话 = '123%'
D.SELECT 厂家名, 厂址, 联系电话 FROM 生产厂家 WHERE 联系电话 = '123'

(5) 有如下定义汽车表和和销售表的语句，请补全空白部分。

CREATE TABLE 汽车(汽车型号 char(10) (1) E.primary key, 汽车名 (2) N.nvarchar(20) (3) D.not null, 颜色 (4) A.nchar(2) check (颜色) (5) F.in ('白色', '红色', '银灰'), 出厂价格 int)
CREATE TABLE 销售(汽车型号 char(10), 厂家编号 (6) G.char(20), 销售日期 (7) H.date, 销售价格 int, foreign key (8) K.汽车型号 (9) I.references (10) C.汽车 (汽车型号))
B.varchar(20) M.of

(6) 现要创建满足如下要求的视图：查询“汽车一厂”厂家生产的汽车名、汽车颜色、出厂价格和销售价格。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) E.汽车名, 颜色, 出厂价格, 销售价格 FROM 汽车 a JOIN 销售 b ON (2) A.a. 汽车型号 = b. 汽车型号 JOIN (3) D.生产厂家 c ON (4) B.b. 厂家编号 = c. 厂家编号 WHERE (5) C.厂家名 = '汽车一厂'

38、有如下汽车、生产厂家和销售三张表，各表结构如下：

“汽车”表结构：
汽车型号：普通编码定长字符类型，长度为 10，主键。
汽车名：统一编码可变长字符类型，长度为 20，非空。
颜色：统一编码定长字符类型，长度为 2，取值为：“白色”、“红色”、“银灰”。
出厂价格：整型。
“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。
厂家名：统一编码可变长字符类型，长度为 10，非空。
厂址：统一编码可变长字符类型，长度为 20。
联系电话：普通编码定长字符类型，长度为 8。
“销售”表结构：
汽车型号：普通编码定长字符类型，长度为 10，引用“汽车”表的外键。
厂家编号：普通编码定长字符类型，长度为 20，引用“生产厂家”的外键。
销售日期：日期型。
销售价格：整型。

(1) 查询出厂价格在 30 万到 40 万之间（包括边界值）的汽车数量。下列语句正确的是 ()。

B.SELECT COUNT(汽车型号) FROM 汽车 WHERE 出厂价格 BETWEEN 30 AND 40
C.SELECT COUNT(汽车名) FROM 汽车 WHERE 出厂价格 >= 30 AND 出厂价格 <= 40
D.SELECT COUNT(*) FROM 汽车 WHERE 出厂价格 >= 30 AND 出厂价格 <= 40

(2) 将出厂价格低于 20 万的汽车的出厂价格提高 5%。下列语句正确的是 ()。

A.UPDATE 汽车 SET 出厂价格 = 出厂价格 + 出厂价格 * 0.05 WHERE 出厂价格 < 20
B.UPDATE 汽车 SET 出厂价格 = 出厂价格 * 1.05 WHERE 出厂价格 < 20

(3) 有如下定义生产厂家表和和销售表的语句，请补全空白部分。

CREATE TABLE 生产厂家(厂家编号 char(20) (1) D.primary key, 厂家名 (2) B.nvarchar(10) (3) E.not null, 厂址 (4) N.nvarchar(20), 联系电话 (5) M.char(8))

CREATE TABLE 销售(汽车型号 (6) G.char(10), 厂家编号 char(20), 销售日期 (7) H.date, 销售价格 int, (8) K.foreign key (厂家编号) (9) I.references (10) C.生产厂家 (厂家编号))

(4) 查询颜色为“白色”或“红色”的汽车的汽车名和颜色。下列语句正确的是 ()。

B.SELECT 汽车型号, 汽车名 FROM 汽车 WHERE 颜色 = '白色' OR 颜色 = '红色'
C.SELECT 汽车名, 颜色 FROM 汽车 WHERE 颜色 IN ('白色', '红色')

(5) 删除销售价格低于 10 万的汽车销售记录。下列语句正确的是 ()。

A.DELETE 销售 WHERE 销售价格 < 10
C.DELETE FROM 销售 WHERE 销售价格 < 10

(6) 现要创建满足如下要求的视图：查询 2022 年汽车的销售情况，列出汽车名、该车的厂家名和销售价格。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) E.汽车名, 厂家名, 销售日期 FROM (2) D.汽车 a JOIN 销售 b ON (3) A.a. 汽车型号 = b. 汽车型号 JOIN 生产厂家 c ON (4) B.b. 厂家编号 = c. 厂家编号 WHERE (5) C.销售日期 BETWEEN '2022-1-1' AND '2022-12-31'

39、有如下汽车、生产厂家和销售三张表，各表结构如下：

“汽车”表结构：
汽车型号：普通编码定长字符类型，长度为 10，主键。
汽车名：统一编码可变长字符类型，长度为 20，非空。
颜色：统一编码定长字符类型，长度为 2，取值为：“白色”、“红色”、“银灰”。
出厂价格：整型，默认值 5。
“生产厂家”表结构：

厂家编号：普通编码定长字符类型，长度为 20，主键。
厂家名：统一编码可变长字符类型，长度为 10，非空。
厂址：统一编码可变长字符类型，长度为 20。
联系电话：普通编码定长字符类型，长度为 8。
“销售”表结构：

汽车型号：普通编码定长字符类型，长度为 10，引用“汽车”表的外键。
厂家编号：普通编码定长字符类型，长度为 20，引用“生产厂家”的外键。
销售日期：日期型。
销售价格：整型。

(1) 查询厂址前 2 个汉字为“长春”的厂家名和厂址。下列语句错误的是 ()。

B.SELECT 厂家名, 厂址, 联系电话 FROM 生产厂家 WHERE 厂址 LIKE '长春'

C.SELECT 厂家名,厂址 FROM 生产厂家 WHERE 厂址 = '长春'
D.SELECT 厂家名,厂址 FROM 生产厂家 WHERE 厂址 = '长春'

(2) 有如下定义汽车表和生产厂家表的语句,请补全空白部分。CREATE TABLE 汽车(汽车型号 (1) C.char(10) primary key,汽车名 (2) N.nvarchar(20) not null,颜色 (3) A.nchar(2) (4) D.check (颜色 (5) F.in ('白色','红色','银灰')),出厂价格 int (6) K.default 5)CREATE TABLE 生产厂家(厂家编号 (7) G.char(20) primary key,厂家名 (8) B.nvarchar(10) not null,厂址 (9) N.nvarchar(20),联系电话 (10) H.char(8))

(3) 现要创建满足如下要求的视图:查询销售价格在 80-100 万的汽车名、生产厂家、出厂价格。请补全空白部分(设视图名为:V1)。

CREATE VIEW V1 AS SELECT (1) E.汽车名,生产厂家,出厂价格 FROM 汽车 aJOIN 销售 b ON (2) A.a.汽车型号 = b.汽车型号 JOIN (3) D.生产厂家 c ON (4) B.b.厂家编号 = c.厂家编号 WHERE (5) C.销售价格 BETWEEN 80 AND 100

(4) 统计 2022 年 6 月的汽车销售总数量。下列语句错误的是 ()。

A.SELECT COUNT(销售数量) FROM 销售 WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

C.SELECT COUNT(*) FROM 销售 WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

D.SELECT SUM(*) FROM 销售 WHERE 销售日期 >= '2022-6-1' AND 销售日期 <= '2022-6-30'

(5) 在销售表中插入一行数据,汽车型号:W220,厂家编号:Q110,销售日期:2022 年 3 月 6 日,销售价格:待定。下列语句正确的是 ()。

A.INSERT INTO 销售 VALUES('W100','Q110','2022-3-6',NULL)

C.INSERT INTO 销售(厂家编号,汽车型号,销售日期) VALUES('Q110','W100','2022-3-6')

D.INSERT INTO 销售(汽车型号,厂家编号,销售日期) VALUES('W100','Q110','2022-3-6')

(6) 将“银灰”色汽车的出厂价格降低 5%。下列语句正确的是 ()。

A.UPDATE 汽车 SET 出厂价格 = 出厂价格 * 0.95 WHERE 颜色 = '银灰'

B.UPDATE 汽车 SET 出厂价格 = 出厂价格 - 出厂价格 * 0.05 WHERE 颜色 = '银灰'

40、有如下商品、客户和销售三张表,各表结构如下:

“商品”表结构:

商品号:普通编码定长字符类型,长度为 10,主键。

商品名:统一编码可变长字符类型,长度为 10。

类别:统一编码定长字符类型,长度为 2,取值为(百货,电器)。

价格:整型

“客户”表结构:

客户号:普通编码定长字符类型,长度为 18,主键。

客户名:统一编码可变长字符类型,长度为 10,非空。

电话:普通编码定长字符类型,长度为 11,取值不重。

“销售”表结构:

商品号:普通编码定长字符类型,长度为 10,引用“商品”表的外键。

客户号:普通编码定长字符类型,长度为 18,引用“客户”表的外键。

销售日期:日期型。

销售数量:整型。

主键:(商品号,客户号,销售日期)。

(1) 写出创建如下三张数据表的 SQL 语句。

“商品”表结构:

CREATETABLE 商品(
商品号 char(10)PRIMARY KEY,
商品名 nvarchar(10),
类别 nchar(2)check(类别 in('百货','电器')),
价格 int
)

“客户”表结构:

CREATETABLE 客户(
客户号 char(18)primary key,
客户名 nvarchar(10)not null,
电话 char(11)unique
)

“销售”表结构:

CREATETABLE 销售(
商品号 char(10),
客户号 char(18),
销售日期 date,
销售数量 int,
Primary key(商品号,客户号,销售日期),
foreign key(商品号)references 商品(商品号),
foreign key(客户号)references 客户(客户号)
)

(2) 依据第 1 题所创建的三张表,写出完成下列操作的 SQL 语。

① 查询价格小于等于 100 的“百货”类商品的商品名和价格

SELECT 商品名,价格 FROM 商品 WHERE 价格 <= 100 AND 类别 = '百货'

② 查询姓“李”的客户的购买情况,列出客户名、购买的商品号和购买日期(即销售日期)。

SELECT 客户名,商品号,销售日期 FROM 客户 a join 销售 bona.客户号 = b.客户号 WHERE 客户名 LIKE '李%'

③ 删除 2020 年 1 月 1 日之前(不包括该日期)的商品销售记录

DELETEFROM 销售 WHERE 销售日期 < '2020/1/1'

④ 在客户表中插入一行数据,客户号:K100,客户名:新客户,电话:未知

INSERTINTO 客户(客户号,客户名)VALUES('K100','新客户')

(3) 依据第 1 题所创建的三张表,写出创建满足如下要求的视图的 SQL 语句:查询“百货”类商品一次销售数量超过 50 的的商品的销售情况,列出商品名价格客户名销售日期和售数量

CREATEVIEW V1AS

SELECT 商品名,价格,客户名,销售日期,销售数量

FROM 商品 aJOIN 销售 bONa.商品号 = b.商品号

JOIN 客户 cONc.客户号 = b.客户号

WHERE 类别 = '百货' AND 销售数量 > 50

41、有如下商品、生产厂家和生产三张表,各表结构如下:

“商品”表结构:

商品号:普通编码定长字符类型,长度为 10,主键。

商品名:普通编码可变长字符类型,长度为 20,非空。

类别:统一字符编码定长字符类型,长度为 5,取值为:“智能穿戴”、“4G 手机”、“笔记本电脑”。

单价:整型。

“生产厂家”表结构:

厂家编号:普通编码定长字符类型,长度为 20,主键。

厂家名:普通编码可变长字符类型,长度为 30,非空。

厂家地址:统一字符编码可变长字符类型,长度为 40。

“生产”表结构:

商品号:普通编码定长字符类型,长度为 10,引用商品表的外键。

厂家编号:普通编码定长字符类型,长度为 20,引用生产厂家表的外键。

生产日期:日期类型。

生产数量:整型。

主键:(商品号,厂家编号,生产日期)。

(1) 写出创建如下三张数据表的 SQL 语句。

“商品”表结构:

```
CREATETABLE 商品(  
商品号 char(10) primary key,  
商品名 varchar(20) not null,  
类别 nchar(5) CHECK (类别 IN('智能穿戴','4G 手机','笔记本电脑')),  
单价 int  
)
```

“生产厂家”表结构:

```
CREATETABLE 生产厂家(  
厂家编号 char(20) primary key,  
厂家名 varchar(30) not null,  
厂家地址 nvarchar(40)  
)
```

“生产”表结构:

```
CREATETABLE 生产(  
商品号 char(10),  
厂家编号 char(20),  
生产日期 date,  
生产数量 int,  
Primarykey(商品号,厂家编号,生产日期),  
Foreignkey(商品号) references 商品(商品号),  
Foreignkey(厂家编号) references 生产厂家(厂家编号)  
)
```

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

① 查询“智能穿戴”类商品的商品名和单价。

```
SELECT 商品名, 单价 FROM 商品 WHERE 类别 = '智能穿戴'
```

② 查询单价大于 8000 的“笔记本电脑”类商品的生产情况, 列出商品名、单价和生产日期。

```
SELECT 商品名, 单价, 生产日期 FROM 商品 a JOIN 生产 b ON a.商品号 = b.商品号 WHERE 单价 > 8000 AND 类别 = '笔记本电脑'
```

③ 删除 2014 年 1 月 1 日之前生产的全部商品。

```
DELETE 商品 WHERE 商品号 IN (SELECT 商品号 FROM 生产 WHERE 生产日期 < 2014/1/1)
```

④ 在商品表中插入一行数据, 商品号: SP501, 商品名: 智能手表, 类别: 智能穿戴, 单价未知。

```
INSERT INTO 商品(商品号, 商品名, 类别) VALUES('SP501', '智能手表', '智能穿戴')
```

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询生产“4G 手机”类商品的厂家名和厂家地址。

```
CREATE VIEW V1 AS
```

```
SELECT 厂家名, 厂家地址 FROM 生产厂家 a
```

```
JOIN 生产 b ON a.厂家编号 = b.厂家编号
```

```
JOIN 商品 c ON c.商品号 = b.商品号  
WHERE 类别 = '4G 手机'
```

42、有如下图书、经销商和销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 统一编码可变长字符类型, 长度为 20, 非空。

类别: 统一编码定长字符类型, 长度为 4, 取值为: “计算机”、“通信”、“外语”。

出版日期: 日期型。

单价: 整型

“经销商”表结构:

商家编号: 普通编码定长字符类型, 长度为 20, 主键。

商家名: 统一编码可变长字符类型, 长度为 10, 非空。

商家地址: 统一编码可变长字符类型, 长度为 30。

联系电话: 普通编码定长字符类型, 长度为 11, 取值不重。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用“图书”表的外键。

商家编号: 普通编码定长字符类型, 长度为 20, 引用“经销商”的外键。

销售日期: 日期型, 默认值为系统当前日期。

主键: (书号, 商家编号, 销售日期)。

(1) 写出创建如下三张表的 SQL 语句。

“图书”表结构:

```
CREATETABLE 图书(  
书号 char(10) primary key,  
书名 nvarchar(20) not null,  
类别 nchar(4) CHECK (类别 IN('计算机','通信','外语')),  
出版日期 date,  
单价 int  
)
```

“经销商”表结构:

```
CREATETABLE 经销商(  
商家编号 char(20) primary key,  
商家名 nvarchar(10) not null,  
商家地址 nvarchar(30),  
联系电话 char(11) unique  
)
```

“销售”表结构:

```
CREATETABLE 销售(  
书号 char(10),  
商家编号 char(20),  
销售日期 datedefaultgetdate (),  
Primarykey(书号, 商家编号, 销售日期),  
Foreignkey(书号) references 图书(书号),  
Foreignkey(商家编号) references 经销商(商家编号)  
)
```

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

① 查询“计算机”类全部图书的详细信息。

```
SELECT * FROM 图书 WHERE 类别 = '计算机'
```

② 查询单价在 40-60 的图书的销售情况, 列出书名、单价、销售日期。

```
SELECT 书名, 单价, 销售日期  
FROM 图书 a JOIN 销售 b ON a.书号 = b.书号  
WHERE 单价 BETWEEN 40 AND 60
```

③ 将“计算机”类图书的单价降低 10%。

```
UPDATE 图书  
SET 单价 = 单价 - 单价 * 0.1  
WHERE 类别 = '计算机'
```

④ 从销售表中删除 2000 年 1 月 1 日之前出版的图书信息。

```
DELETE 销售  
FROM 图书 a JOIN 销售 b ON a.书号 = b.书号
```

WHERE 出版日期 <'2000/1/1'

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询地址为“长春”的经销商的图书销售情况, 列出所销售的每本图书的书号和销售次数。

```
CREATE VIEW V1(书号,销售数量)AS  
SELECT 书号,COUNT(*)  
FROM 经销商 aJOIN 销售 b ON a.商家编号 = b.商家编号  
WHERE 商家地址 = '长春'  
GROUP BY 书号
```

43、有如下图书、经销商和销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 统一编码可变长字符类型, 长度为 20, 非空。

类别: 统一编码定长字符类型, 长度为 4, 取值为: “计算机”、“通信”、“外语”。

出版日期: 日期型。

单价: 整型

“经销商”表结构:

商家编号: 普通编码定长字符类型, 长度为 20, 主键。

商家名: 统一编码可变长字符类型, 长度为 10, 非空。

商家地址: 统一编码可变长字符类型, 长度为 30。

联系电话: 普通编码定长字符类型, 长度为 11, 取值不重。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用“图书”表的外键。

商家编号: 普通编码定长字符类型, 长度为 20, 引用“经销商”的外键。

销售日期: 日期型, 默认值为系统当前日期。

主键: (书号, 商家编号, 销售日期)。

(1) 写出创建如下三张表的 SQL 语句。

“图书”表结构:

```
CREATE TABLE 图书(  
书号 char(10) primary key,  
书名 nvarchar(20) not null,  
类别 nchar(4) CHECK (类别 IN('计算机','通信','外语')),  
出版日期 date,  
单价 int  
)
```

“经销商”表结构:

```
CREATE TABLE 经销商(  
商家编号 char(20) primary key,  
商家名 nvarchar(10) not null,  
商家地址 nvarchar(30),  
联系电话 char(11) unique  
)
```

“销售”表结构:

```
CREATE TABLE 销售(  
书号 char(10),  
商家编号 char(20),  
销售日期 date default getdate(),  
Primary key(书号,商家编号,销售日期),  
Foreign key(书号) references 图书(书号),  
Foreign key(商家编号) references 经销商(商家编号)  
)
```

]

(2) 依据第 1 题所创建的三张表, 写出完成下列操作的 SQL 语句。

①查询“计算机”类全部图书的详细信息。

```
SELECT * FROM 图书 WHERE 类别 = '计算机'
```

②查询单价在 40-60 的图书的销售情况, 列出书名、单价、销售日期。

```
SELECT 书名, 单价, 销售日期 FROM 图书 a JOIN 销售 b ON a.书号 = b.书号  
WHERE 单价 BETWEEN 40 AND 60
```

③将“计算机”类图书的单价降低 10%。

```
UPDATE 图书 SET 单价 = 单价 - 单价 * 0.1 WHERE 类别 = '计算机'
```

④从销售表中删除 2000 年 1 月 1 日之前出版的图书信息。

```
DELETE 销售 FROM 图书 a JOIN 销售 b ON a.书号 = b.书号 WHERE 出版日期 < '2000/1/1'
```

(3) 依据第 1 题所创建的三张表, 写出创建满足如下要求的视图的 SQL 语句: 查询地址为“长春”的经销商的图书销售情况, 列出所销售的每本图书的书号和销售次数。

```
CREATE VIEW V1(书号,销售数量)AS  
SELECT 书号,COUNT(*)  
FROM 经销商 a JOIN 销售 b ON a.商家编号 = b.商家编号  
WHERE 商家地址 = '长春'  
GROUP BY 书号
```

44、有如下图书、书店和销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 4, 取值为: “科技英语”、“现代汉语”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 3 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。

销售时间: 日期时间类型。

销售数量: 整型, 取值大于等于 1。

(1) 现要创建满足如下要求的视图: 查询“科技英语”类图书的销售情况, 列出书名、书店名、销售时间和销售数量。请补全空白部分 (设视图名为: V1)。CREATE VIEW V1 AS SELECT (1) A.书名, (2) 书店名, (3) 销售时间, (4) 销售数量 FROM 图书 a JOIN (2) B.销售 b ON (3) D.a. 书号 = b. 书号 JOIN 书店 c ON (4) C.c. 书店编号 = b. 书店编号 WHERE (5) E.类别 = '科技英语'

(2) 在销售表中插入一行数据: 书号: S100, 书店编号: D200, 销售时间: 待定, 销售数量: 50。下列语句正确的是 ()。

```
B.INSERT INTO 销售 VALUES('S100', 'D200', NULL, 50)
```

```
C.INSERT INTO 销售(书号,书店编号,销售数量) VALUES('S100', 'D200', 50)
```

(3) 有如下定义图书表和书店表的语句, 请补全空白部分。CREATE TABLE 图书(书号 (1) E.char(10) primary key, 书名 (2) N.varchar(20) not null, 类别 (3) M.nchar(4) (4) L.check (类别 (5) K.in ('科技英语','现代汉语')), 出版日期 (6) C.date, 单价 (7) H.numeric) CREATE TABLE 书店(书店编号 (8) G.char(20) (5,2) primary key, 书店名 (9) B.varchar(30) not null, 地址 (10) F.nvarchar(30))

(4) 查询“科技英语”类图书的平均单价。下列语句错误的是 ()。

```
B.SELECT MAX(单价) FROM 图书 WHERE 类别 = '科技英语'
```

```
C.SELECT MIN(单价) FROM 图书 WHERE 类别 = '科技英语'
```

```
D.SELECT SUM(单价) FROM 图书 WHERE 类别 = '科技英语'
```

(5) 删除 2000 年 1 月 1 日之前的图书销售记录。下列语句正确的是 ()。

[A.DELETE 销售 WHERE 销售时间 <= '2000-1-1'](#)
[B.DELETE 销售 WHERE 销售时间 <= '2000/01/01'](#)

(6) 查询单次销售数量在 400-500 (包括边界值) 的书号、书店编号、销售时间。下列语句正确的是 ()。

[B.SELECT 书号,书店编号,销售时间 FROM 销售 WHERE 销售数量 BETWEEN 400 AND 500](#)
[D.SELECT 书号,书店编号,销售时间 FROM 销售 WHERE 销售数量 >= 400 AND 销售数量 <= 500](#)

45、有如下图书、书店和销售三张表,各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 4, 取值为: “高等数学”、“当代文学”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 3 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。

销售时间: 日期时间类型。

销售数量: 整型。

(1) 有如下定义图书表和销售表的语句, 请补全空白部分。CREATE TABLE 图书(书号 char(10)

(1) [D.primary key](#), 书名 (2) [N.varchar\(20\)](#) not null, 类别 (3) [A.nchar\(4\)](#) (4) [F.check](#) (类别 IN('高等数学', '当代文学')), 出版日期 date, 单价 (5) [E.numeric\(5,2\)](#) CREATE TABLE 销售(书号 char(10), 书店编号 (6) [B.char\(20\)](#), 销售时间 (7) [C.datetime](#), 销售数量 (8) [H.int](#), (9) [K.foreign key](#) (书号) (10) [I.references](#) 图书(书号))

(2) 查询单价在 30-50 (包括边界值) 的图书数量。下列语句正确的是 ()。

[B.SELECT COUNT\(书号\) FROM 图书 WHERE 单价 BETWEEN 30 AND 50](#)
[D.SELECT COUNT\(书号\) FROM 图书 WHERE 单价 >= 30 AND 单价 <= 50](#)

(3) 将“高等数学”类图书的单价降低 10。下列语句错误的是 ()。

[B.UPDATE 单价 SET 单价 = 单价 - 10 WHERE 类别 = '高等数学'](#)
[C.ALTER 图书 SET 单价 = 单价 - 10 WHERE 类别 = '高等数学'](#)
[D.ALTER 单价 SET 单价 = 单价 - 10 WHERE 类别 = '高等数学'](#)

(4) 在书店表中插入一行数据, 书店编号: SD100, 书店名: 新华书店, 地址: 西单。下列语句正确的是 ()。

[A.INSERT INTO 书店 VALUES\('SD100','新华书店','西单'\)](#)
[B.INSERT INTO 书店\(书店编号,书店名,地址\) VALUES\('SD100','新华书店','西单'\)](#)
[C.INSERT INTO 书店 VALUES\('SD100','新华书店','西单'\)](#)
[D.INSERT INTO 书店\(书店编号,书店名,地址\) VALUES\('SD100','新华书店','西单'\)](#)

(5) 现要创建满足如下要求的视图: 查询销售了“高等数学”类图书的书店名和地址。请补全空白部分 (设视图名为: V1)。CREATE (1) [D.VIEW V1 AS SELECT](#) (2) [E.书店名,地址](#) FROM 书店 a JOIN (3) [A.销售 b ON a. 书店编号 = b. 书店编号 JOIN 图书 c ON](#) (4) [C.c. 书号 = b. 书号](#) WHERE (5) [B.类别 = '高等数学'](#)

(6) 查询“高等数学”类图书的书名和出版日期。下列语句正确的是 ()。

[A.SELECT 书名,出版日期 FROM 图书 WHERE 类别 = '高等数学'](#)

[B.SELECT 书名,出版日期 FROM 图书 WHERE 类别 LIKE '高等数学'](#)

46、有如下图书、书店和销售三张表,各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 3, 取值为: “计算机”、“英语”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。

销售时间: 日期时间类型。

销售数量: 整型, 取值大于等于 100。

(1) 查询“计算机”、“英语”两类图书的书名和单价。下列语句错误的是 ()。

[A.SELECT 书名,单价 FROM 图书 WHERE 类别 = '计算机' OR '英语'](#)
[B.SELECT 书名,单价 FROM 图书 WHERE 类别 = '计算机' AND '英语'](#)
[D.SELECT 书名,单价 FROM 图书 WHERE 类别 = '计算机' AND 类别 = '英语'](#)

(2) 将 2020 年之前出版的图书降价 5%。下列语句正确的是 ()。

[C.UPDATE 图书 SET 单价 = 单价 * 0.95 WHERE 出版日期 < '2020-1-1'](#)
[D.UPDATE 图书 SET 单价 = 单价 - 单价 * 0.05 WHERE 出版日期 < '2020-1-1'](#)

(3) 删除单价高于 100 的图书记录。下列语句正确的是 ()。

[A.DELETE FROM 图书 WHERE 单价 > 100](#)
[C.DELETE 图书 WHERE 单价 > 100](#)

(4) 统计每类图书的最高单价和最低单价。下列语句错误的是 ()。

[B.SELECT 类别, MAX\(单价\), AVG\(单价\) FROM 图书 GROUP BY 类别](#)
[C.SELECT 类别, TOP\(单价\), MIN\(单价\) FROM 图书 GROUP BY 类别](#)
[D.SELECT 类别, MAX\(单价\), SUM\(单价\) FROM 图书 GROUP BY 类别](#)

(5) 有如下定义书店表和销售表的语句, 请补全空白部分。

CREATE TABLE 书店(书店编号 char(20) (1) [B.primary key](#), 书店名 (2) [N.varchar\(30\)](#) (3) [C.not null](#), 地址 (4) [M.nvarchar\(30\)](#))

CREATE TABLE 销售(书号 (5) [G.char\(10\)](#), 书店编号 char(20), 销售时间 (6) [D.datetime](#), 销售数量 (7) [F.int](#) (8) [K.check](#) (销售数量 >= 100), (9) [E.foreign key](#) (书号) (10) [I.references](#) 图书(书号))

[A.numeric\(4,2\)](#) [H.numeric\(6,2\)](#)

(6) 现要创建满足如下要求的视图: 查询 2022 年 1 月 1 日及之后销售的图书的书名、书店名、销售时间、销售数量。请补全空白部分 (设视图名为: V1)。

CREATE VIEW V1 AS SELECT 书名,书店名,销售时间,销售数量 (1) [E.FROM](#) 图书 a JOIN (2) [A.销售 b ON](#) (3) [B.a. 书号 = b. 书号](#) JOIN (4) [D.书店 c ON c. 书店编号 = b. 书店编号](#) WHERE (5) [C.销售时间 >= '2022-1-1'](#)

47、有如下图书、书店和销售三张表,各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 4, 取值为: “高等数学”、“当代文学”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 3 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。

销售时间: 日期时间类型。

销售数量: 整型, 取值大于等于 1。

(1) 有如下定义书店表和销售表的语句, 请补全空白部分。

```
CREATE TABLE 书店(
```

```
书店编号 char(20) (1) [D] primary key,
```

```
书店名(2) [N] varchar(30) (3) [A] not null,
```

```
地址 (4) [B] nvarchar(30)
```

```
)
```

```
CREATE TABLE 销售(
```

```
书号 (5) [G] char(10),
```

```
书店编号 char(20),
```

```
销售时间 销售数量(6) [C] datetime,
```

```
(7) [H] int check (8) [M] 销售数量 >= 1,
```

```
(9) (书号) [K] foreign key (10) [I] references 图书(书号)
```

```
)
```

(2) 【多选题】

1) 查询单价在 30-50 (包括边界值) 的图书的书名和单价。下列语句正确的是 ()。

[A.] SELECT 书名, 单价 FROM 图书 WHERE 单价 BETWEEN 30 AND 50

[C.] SELECT 书名, 单价 FROM 图书 WHERE 单价 >= 30 AND 单价 <= 50

2) 统计书店总数量。下列语句正确的是 ()。

[B.] SELECT COUNT(书店号) FROM 书店

[D.] SELECT COUNT(*) FROM 书店

3) 将书店编号为“SD201”的店名改为“王府井新店”。下列语句错误的是 ()。

[B.] UPDATE 书店名 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

[C.] ALTER 书店 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

[D.] ALTER 书店名 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

4) 在图书表中插入一行数据, 书号: S100, 书名: 线性代数, 类别: 高等数学, 出版日期: 2022 年 5 月 10 日, 单价: 待定。下列语句正确的是 ()。

[B.] INSERT INTO 图书 VALUES('S100','线性代数','高等数学','2022-5-10',NULL)

[C.] INSERT INTO 图书(书号, 书名, 类别, 出版日期) VALUES('S100','线性代数','高等数学'

'2022-5-10')

(3) 现要创建满足如下要求的视图: 查询“王府井书店”销售的图书情况, 列出书名、销售时间和销售数量。请补全空白部分 (设视图名为: V1)。

```
CREATE (1) [D] VIEW V1 AS SELECT (2) [A] 书名, 销售时间, 销售数量 FROM
```

```
图书 a JOIN (3) [E] 销售 b ON a.书号 = b.书号 JOIN 书店 c ON (4) [C] c.书店编号 = b.书店编号
```

```
WHERE (5) [B] 书店名 = '王府井书店'
```

48、有如下图书、书店和销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 3, 取值为: “计算机”、“英语”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址: 统一字符编码可变长字符类型, 长度为 30。

“销售”表结构:

书号: 普通编码定长字符类型, 长度为 10, 引用图书表的外键。

书店编号: 普通编码定长字符类型, 长度为 20, 引用书店表的外键。

销售时间: 日期时间类型。

销售数量: 整型, 取值大于等于 100。

(1) 有如下定义书店表和销售表的语句, 请补全空白部分。

```
CREATE TABLE 书店(
```

```
书店编号 char(20) (1) [B] primary key,
```

```
书店名(2) [N] varchar(30) (3) [C] not null,
```

```
地址 (4) [M] nvarchar(30)
```

```
)
```

```
CREATE TABLE 销售(
```

```
书号 (5) [G] char(10),
```

```
书店编号 char(20),
```

```
销售时间(6) [D] datetime,
```

```
销售数量 (7) [F] int (8) [K] check (销售数量 >= 100),
```

```
(9) [E] foreign key (书号) (10) [I] references 图书(书号)
```

```
)
```

(2) 【多选题】

1) 查询“计算机”、“英语”两类图书的书名和单价。下列语句错误的是 ()。

[A.] SELECT 书名, 单价 FROM 图书 WHERE 类别 = '计算机' OR '英语'

[B.] SELECT 书名, 单价 FROM 图书 WHERE 类别 = '计算机' AND '英语'

[D.] SELECT 书名, 单价 FROM 图书 WHERE 类别 = '计算机' AND 类别 = '英语'

2) 统计每类图书的最高单价和最低单价。下列语句错误的是 ()。

[B.] SELECT 类别, MAX(单价), AVG(单价) FROM 图书 GROUP BY 类别

[C.] SELECT 类别, TOP(单价), MIN(单价) FROM 图书 GROUP BY 类别

[D.] SELECT 类别, MAX(单价), SUM(单价) FROM 图书 GROUP BY 类别

3) 将 2020 年之前出版的图书降价 5%。下列语句正确的是 ()。

[C.] UPDATE 图书 SET 单价 = 单价 * 0.95 WHERE 出版日期 < '2020-1-1'

[D.] UPDATE 图书 SET 单价 = 单价 - 单价 * 0.05 WHERE 出版日期 < '2020-1-1'

4) 删除单价高于 100 的图书记录。下列语句正确的是 ()。

[A.] DELETE FROM 图书 WHERE 单价 > 100

[C.] DELETE 图书 WHERE 单价 > 100

(3) 现要创建满足如下要求的视图: 查询 2022 年 1 月 1 日及之后销售的图书的书名、书店名、销售时间、销售数量。请补全空白部分 (设视图名为: V1)。

```
CREATE VIEW V1 AS
```

```
SELECT 书名, 书店名, 销售时间,
```

```
销售数量 (1) [E] FROM
```

```
图书 a JOIN (2) [A] 销售 b ON (3) [B] a.书号 = b.书号 JOIN (4) [D] 书店 c
```

```
ON c.书店编号 = b.书店编号 WHERE (5) [C] 销售时间 >= '2022-1-1'
```

49、有如下图书、书店和销售三张表, 各表结构如下:

“图书”表结构:

书号: 普通编码定长字符类型, 长度为 10, 主键。

书名: 普通编码可变长字符类型, 长度为 20, 非空。

类别: 统一字符编码定长字符类型, 长度为 3, 取值为: “计算机”、“英语”。

出版日期: 日期类型。

单价: 定点小数类型, 小数点前 4 位, 小数点后 2 位。

“书店”表结构:

书店编号: 普通编码定长字符类型, 长度为 20, 主键。

书店名: 普通编码可变长字符类型, 长度为 30, 非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 10，引用图书表的外键。

书店编号：普通编码定长字符类型，长度为 20，引用书店表的外键。

销售时间：日期时间类型。

销售数量：整型。

(1) 有如下定义图书表和销售表的语句，请补全空白部分。

CREATE TABLE 图书(书号 char(10) (1) B.primary key, 书名 (2) N.varchar(20) not null, 类别 (3) M.nchar(3) check ((4) K.类别 in ('计算机','英语')), 出版日期 (5) C.date, 单价 (6) H.numeric(6,2))
CREATE TABLE 销售(书号 char(10), 书店编号 char(20), 销售时间 (7) D.datetime, 销售数量 (8) F.int, (9) E.foreign key (书号) (10) I.references 图书(书号)) A.numeric(4,2) G.char(20)

(2) 将书号为“S100”的图书的单价提高 10 元。下列语句错误的是 ()。

B.UPDATE 图书 SET 单价 = 10 WHERE 书号 = 'S100'

C.ALTER 图书 SET 单价 = 单价 + 10 WHERE 书号 = 'S100'

D.ALTER 图书 SET 单价 = 10 WHERE 书号 = 'S100'

(3) 查询书店地址的前两个字是“北京”的书店名和地址。下列语句错误的是 ()。

A.SELECT 书店名,地址 FROM 书店 WHERE 地址 LIKE '北京'

C.SELECT 书店名,地址 FROM 书店 WHERE 地址 = '北京'

D.SELECT 书店名,地址 FROM 书店 WHERE 地址 = '北京%'

(4) 从书店表中删除店名为“ABC 书店”的记录。下列语句正确的是 ()。

A.DELETE 书店 WHERE 书店名 = 'ABC 书店'

B.DELETE FROM 书店 WHERE 书店名 = 'ABC 书店'

(5) 查询 2022 年 5 月图书的销售总数量。下列语句错误的是 ()。

A.SELECT COUNT(*) FROM 销售 WHERE 销售时间 BETWEEN '2022-5-1' AND '2022-5-31'

B.SELECT COUNT(销售数量) FROM 销售 WHERE 销售时间 BETWEEN '2022-5-1' AND '2022-5-31'

C.SELECT SUM(*) FROM 销售 WHERE 销售时间 BETWEEN '2022-5-1' AND '2022-5-31'

(6) 现要创建满足如下要求的视图：查询单价在 50 以上的图书的销售情况，列出书名、单价、销售时间和销售数量。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) A.书名,单价,销售时间,销售数量 FROM 图书 a JOIN (2) E.销售 b ON (3) B.a. 书号 = b. 书号 JOIN (4) D.书店 c ON c. 书店编号 = b. 书店编号 WHERE (5) C.单价 > 50

50、有如下图书、书店和销售三张表，各表结构如下：

“图书”表结构：

书号：普通编码定长字符类型，长度为 10，主键。

书名：普通编码可变长字符类型，长度为 20，非空。

类别：统一字符编码定长字符类型，长度为 4，取值为：“高等数学”、“当代文学”。

出版日期：日期类型。

单价：定点小数类型，小数点前 3 位，小数点后 2 位。

“书店”表结构：

书店编号：普通编码定长字符类型，长度为 20，主键。

书店名：普通编码可变长字符类型，长度为 30，非空。

地址：统一字符编码可变长字符类型，长度为 30。

“销售”表结构：

书号：普通编码定长字符类型，长度为 10，引用图书表的外键。

书店编号：普通编码定长字符类型，长度为 20，引用书店表的外键。

销售时间：日期时间类型。

销售数量：整型，取值大于等于 1。

(1) 有如下定义书店表和销售表的语句，请补全空白部分。CREATE TABLE 书店(书店编号 char(20) (1) D.primary key, 书店名 (2) N.varchar(30) (3) A.not null, 地址 (4) B.nvarchar(30))
CREATE TABLE 销售(书号 (5) G.char(10), 书店编号 char(20), 销售时间 (6) C.datetime, 销售数量 (7) H.int check ((8) M.销售数量 >= 1), (9) K.foreign key (书号) (10) I.references 图书(书号)) E.nchar(10) F.nchar(30)

(2) 查询单价在 30-50（包括边界值）的图书的书名和单价。下列语句正确的是 ()。

A.SELECT 书名,单价 FROM 图书 WHERE 单价 BETWEEN 30 AND 50

C.SELECT 书名,单价 FROM 图书 WHERE 单价 >= 30 AND 单价 <= 50

(3) 在图书表中插入一行数据，书号：S100，书名：线性代数，类别：高等数学，出版日期：2022 年 5 月 10 日，单价：待定。下列语句正确的是 ()。

B.INSERT INTO 图书 VALUES('S100','线性代数','高等数学','2022-5-10',NULL)

C.INSERT INTO 图书(书号,书名,类别,出版日期) VALUES('S100','线性代数','高等数学','2022-5-10')

(4) 现要创建满足如下要求的视图：查询“王府井书店”销售的图书情况，列出书名、销售时间和销售数量。请补全空白部分（设视图名为：V1）。CREATE (1) D.VIEW V1 AS SELECT (2) A.书名,销售时间,销售数量 FROM 图书 a JOIN (3) E.销售 b ON a. 书号 = b. 书号 JOIN 书店 c ON (4) C.c. 书店编号 = b. 书店编号 WHERE (5) B.书店名 = '王府井书店'

(5) 统计书店总数量。下列语句正确的是 ()。

B.SELECT COUNT(书店号) FROM 书店

D.SELECT COUNT(*) FROM 书店

(6) 将书店编号为“SD201”的店名改为“王府井新店”。下列语句错误的是 ()。

B.UPDATE 书店名 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

C.ALTER 书店 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

D.ALTER 书店名 SET 书店名 = '王府井新店' WHERE 书店编号 = 'SD201'

51、有如下项目、会员和健身三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 10，非空。

类别：普通编码定长字符类型，长度为 6，取值：瑜伽、跑步、健美操。

费用：整型，默认值 100。

“会员”表结构：

会员号：普通编码定长字符类型，长度为 18，主键。

会员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1。

联系电话：普通编码定长字符类型，长度为 11。

“健身”记录表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

会员号：普通编码定长字符类型，长度为 18，引用“会员”表的外键。

健身日期：日期型。

健身时长：整型。（注：以小时为单位）

主键：（项目号，会员号，健身日期）。

(1) 将价格超过 500 的项目费用降低 100。下列语句错误的是 ()。

A.UPDATE 费用 SET 费用 = 费用 - 100 WHERE 费用 > 500

C.ALTER 费用 SET 费用 = 费用 - 100 WHERE 费用 > 500

D.ALTER 项目 SET 费用 = 费用 - 100 WHERE 费用 > 500

(2) 统计每类项目的平均费用。下列语句错误的是 ()。

A.SELECT 类别, COUNT(费用) FROM 项目 GROUP BY 类别

C.SELECT 类别, COUNT(*) FROM 项目 GROUP BY 类别

D.SELECT 类别, AVG(*) FROM 项目 GROUP BY 类别

(3) 查询“2022 年 2 月 1 日”健身的会员号、项目号和健身时长。下列语句正确的是 ()。

A.SELECT 会员号, 项目号, 健身时长 FROM 健身 WHERE 健身日期 = '2022-2-1'

C.SELECT 会员号, 项目号, 健身时长 FROM 健身 WHERE 健身日期 = '2022-02-01'

(4) 在项目表中插入一行数据，项目号：X100，项目名：新项目，类别：跑步，费用：待定。下列语句正确的是 ()。

B.INSERT INTO 项目 VALUES('X100','新项目','跑步',NULL)

C.INSERT INTO 项目(项目号, 项目名, 类别) VALUES('X100','新项目','跑步')

(5) 有如下定义项目表和健身表的语句, 请补全空白部分。
CREATE TABLE 项目 (项目号 char(10) (1) D.primary key, 项目名 (2) G.nvarchar(10) (3) A.not null, 类别 char(6) (4) F.check (类别 in ('瑜伽','跑步','健美操')), 费用 int (5) B.default 100)

CREATE TABLE 健身 (项目号 char(10), 会员号 (6) M.char(18), 健身日期 (7) C.date, 健身时长 int , (8) K.foreign key (项目号) (9) I.references 项目 ((10) E.项目号))

(6) 现要创建满足如下要求的视图: 查询参加“健美操”项目的情况, 列出会员名、健身日期和健身时长。请补全空白部分 (设视图名为: V1)。

CREATE (1) D.VIEW V1 AS SELECT (2) E.会员名, 健身日期, 健身时长 FROM 项目 a JOIN 健身 b ON (3) C.a. 项目号 = b. 项目号 JOIN 会员 c ON (4) A.c. 会员号 = b. 会员号 WHERE (5) B.项目名 = '健美操'

52、有如下项目、会员和健身三张表, 各表结构如下:

“项目”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 主键。

项目名: 统一编码可变量长字符类型, 长度为 10, 非空。

类别: 普通编码定长字符类型, 长度为 6, 取值: 瑜伽、跑步、健美操。

费用: 整型, 默认值 100。

“会员”表结构:

会员号: 普通编码定长字符类型, 长度为 18, 主键。

会员名: 统一编码可变量长字符类型, 长度为 10, 非空。

性别: 统一编码定长字符类型, 长度为 1。

联系电话: 普通编码定长字符类型, 长度为 11。

“健身”记录表结构:

项目号: 普通编码定长字符类型, 长度为 10, 引用“项目”表的外键。

会员号: 普通编码定长字符类型, 长度为 18, 引用“会员”表的外键。

健身日期: 日期型。

健身时长: 整型。(注: 以小时为单位)

主键: (项目号, 会员号, 健身日期)。

(1) 有如下定义项目表和健身表的语句, 请补全空白部分。(每空 1 分, 共 10 分)

“项目”表结构:

CREATE TABLE 项目(

项目号 char(10) (1) [D.] primary key,

项目名 (2) [G.] nvarchar(10) (3) [A.] not null,

类别 char(6) (4) [F.] check (类别 in ('瑜伽','跑步','健美操')),

费用 int (5) [B.] default 100

)

“健身”记录表结构:

CREATE TABLE 健身(

项目号 char(10),

会员号 (6) [M.] char(18),

健身日期 (7) [C.] date,

健身时长 int ,

(8) [K.] foreign key (项目号) (9) [I.] references 项目 ((10) [E.] 项目号)

)

(2) 完成下列多选题 (20 分)

①查询单次健身时长在 1-3 小时的会员号、项目号和健身日期。下列语句正确的是 ()。

B.SELECT 会员号, 项目号, 健身日期 FROM 健身

WHERE 健身时长 BETWEEN 1 AND 3

D.SELECT 会员号, 项目号, 健身日期 FROM 健身

WHERE 健身时长 >= 1 AND 健身时长 <= 3

②在项目表中插入一行数据, 项目号: X100, 项目名: 新项目, 类别: 跑步, 费用: 待定。下列语句正确的是 ()。

B.INSERT INTO 项目 VALUES('X100','新项目','跑步',NULL)

C.INSERT INTO 项目(项目号, 项目名, 类别)

VALUES('X100','新项目','跑步')

③将项目号为“X111”的项目费用降低 5%。下列语句错误的是 ()。

A.UPDATE 项目 SET 费用 = 费用 * 95% WHERE 项目号 = 'X111'

C.UPDATE 费用 SET 费用 = 费用 * 95% WHERE 项目号 = 'X111'

D.UPDATE 费用 SET 费用 = 费用 * 0.95 WHERE 项目号 = 'X111'

④统计每个会员的健身总时长。下列语句错误的是 ()。

A.SELECT 会员号, MAX(健身时长) FROM 健身 GROUP BY 会员号

B.SELECT 会员号, COUNT(健身时长) FROM 健身 GROUP BY 会员号

C.SELECT 会员号, TOTAL(健身时长) FROM 健身 GROUP BY 会员号

(3) 现要创建满足如下要求的视图: 查询“2022 年 10 月 10 日”的健身情况, 列出会员名、项目名和健身时长。请补全空白部分 (设视图名为: V1)。(每空 2 分, 共 10 分)

CREATE VIEW V1 AS

SELECT (1) [B.] 会员名, 项目名, 健身时长 FROM (2) [C.] 项目 a

JOIN 健身 b ON (3) [E.] a.项目号 = b.项目号

JOIN 会员 c ON (4) [D.] b.会员号 = c.会员号

WHERE (5) [A.] 健身日期 = '2022-10-10'

53、有如下项目、会员和健身三张表, 各表结构如下:

“项目”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 主键。

项目名: 统一编码可变量长字符类型, 长度为 10, 非空。

类别: 普通编码定长字符类型, 长度为 6, 取值: 瑜伽、跑步、健美操。

费用: 整型, 默认值 100。

“会员”表结构:

会员号: 普通编码定长字符类型, 长度为 18, 主键。

会员名: 统一编码定长字符类型, 长度为 10, 非空。

性别: 统一编码定长字符类型, 长度为 1, 默认值“男”。

联系电话: 普通编码定长字符类型, 长度为 11。

“健身”记录表结构:

项目号: 普通编码定长字符类型, 长度为 10, 引用“项目”表的外键。

会员号: 普通编码定长字符类型, 长度为 18, 引用“会员”表的外键。

健身日期: 日期型。

健身时长: 整型, 取值大于等于 1。(注: 以小时为单位)

主键: (项目号, 会员号, 健身日期)。

(1) 查询费用在 100-200 之间 (包括边界值) 的项目名和费用。下列语句正确的是 ()。

B.SELECT 项目名, 费用 FROM 项目 WHERE 费用 BETWEEN 100 AND 200

D.SELECT 项目名, 费用 FROM 项目 WHERE 费用 >= 100 AND 费用 <= 200

(2) 现要创建满足如下要求的视图: 查询会员“张三”的健身情况, 列出项目名、健身日期和健身时长。请补全空白部分 (设视图名为: V1)。CREATE VIEW V1 AS SELECT (1) E.项目名, 健身日期, 健身时长 FROM 项目 a JOIN (2) D.健身 b ON (3) C.a. 项目号 = b. 项目号 JOIN 会员 c ON (4) A.c. 会员号 = b. 会员号 WHERE (5) B.会员名 = '张三'

(3) 将“瑜伽”类项目的费用增加 5%。下列语句正确的是 ()。

A.UPDATE 项目 SET 费用 = 费用 * 1.05 WHERE 类别 = '瑜伽'

C.UPDATE 项目 SET 费用 = 费用 + 费用 * 0.05 WHERE 类别 = '瑜伽'

(4) 统计“瑜伽”类项目的项数。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 项目 WHERE 类别 = '瑜伽'

B.SELECT COUNT(项目号) FROM 项目 WHERE 类别 = '瑜伽'

C.SELECT COUNT(项目名) FROM 项目 WHERE 类别 = '瑜伽'

(5) 在会员表中插入一行数据, 会员号: H321, 会员名: 王平, 性别: 男, 联系电话: 未知。下列语句正确的是 ()。

B.INSERT INTO 会员 VALUES(H321,'王平','男',NULL)

C.INSERT INTO 会员(会员号,会员名,性别) VALUES('H321','王平','男')

D.INSERT INTO 会员(会员号,会员名,性别,联系电话) VALUES('H321','王平','男',NULL)

(6) 有如下定义会员表和健身表的语句,请补全空白部分。CREATE TABLE 会员(会员号 char(18) (1) D.primary key, 会员名 (2) N.nchar(10) not null, 性别 (3) A.nchar(1) (4) B.default '男' 联系电话 char(11)) CREATE TABLE 健身(项目号 (5) M.char(10), 会员号 char(18), 健身日期 (6) C.date, 健身时长 int (7) F.check (健身时长 >= 1), (8) K.foreign key (会员号) (9) I.references (10) H.会员 (会员号))

54、有如下项目、会员和健身三张表,各表结构如下:

“项目”表结构:

项目号:普通编码定长字符类型,长度为10,主键。

项目名:统一编码可变长字符类型,长度为10,非空。

类别:普通编码定长字符类型,长度为6,取值:瑜伽、跑步、健美操。

费用:整型。

“会员”表结构:

会员号:普通编码定长字符类型,长度为18,主键。

会员名:统一编码定长字符类型,长度为10,非空。

性别:统一编码定长字符类型,长度为1,默认值“男”。

联系电话:普通编码定长字符类型,长度为11。

“健身”记录表结构:

项目号:普通编码定长字符类型,长度为10,引用“项目”表的外键。

会员号:普通编码定长字符类型,长度为18,引用“会员”表的外键。

健身日期:日期型。

健身时长:整型,取值大于等于1。(注:以小时为单位)

主键:(项目号,会员号,健身日期)。

(1) 有如下定义会员表和项目表的语句,请补全空白部分。

CREATE TABLE 会员(

会员号(1) [M] char(18) primary key,

会员名(2) [N] nchar(10) not null,

性别(3) [A] nchar(1) (4) [B] default '男' 联系电话(5) [E] char(11)

)

CREATE TABLE 项目(

项目号(6) [C] char(10) primary key,

项目名(7) [G] nvarchar(10) not null,

类别 char(6) (8) [F] check (9) [H] 类别 in ('瑜伽','跑步','健美操')),

费用(10) [D] int

)

(2) 【多选题】

1) 查询单次健身时长在1-3小时的会员号、项目号和健身日期。下列语句正确的是()。

[B.] SELECT 会员号,项目号,健身日期 FROM 健身 WHERE 健身时长 BETWEEN 1 AND 3

[D.] SELECT 会员号,项目号,健身日期 FROM 健身 WHERE 健身时长 >= 1 AND 健身时长 <= 3

2) 统计“瑜伽”类项目的最高费用和最低费用。下列语句错误的是()。

[B.] SELECT TOP(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

[C.] SELECT MAX(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

[D.] SELECT LAG(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

3) 将项目号为“X111”的项目类别改为“瑜伽”。下列语句正确的是()。

[A.] UPDATE 项目 SET 类别 = '瑜伽' WHERE 项目号 = 'X111'

[B.] UPDATE 项目 SET 类别 = '瑜伽' WHERE 项目号 LIKE 'X111'

4) 删除联系电话后3位为“123”的会员记录。下列语句错误的是()。

[B.] DELETE FROM 会员 WHERE 联系电话 LIKE '123'

[C.] DROP FROM 会员 WHERE 联系电话 LIKE '%123'

[D.] DROP FROM 会员 WHERE 联系电话 LIKE '123'

(3) 现要创建满足如下要求的视图:查询参加费用超过200项目的会员名、项目名和费用。请补全空白部分(设视图名为:V1)。

CREATE VIEW V1 AS

SELECT (1) [A] 会员名,项目名,费用 FROM 项目 a JOIN (2) [D] 健身 b ON

a.项目号 = b.项目号 JOIN (3) [C] 会员 c ON (4) [E] c.会员号 = b.会员号 WHERE (5) [B] 费用 > 200

55、有如下项目、会员和健身三张表,各表结构如下:

“项目”表结构:

项目号:普通编码定长字符类型,长度为10,主键。

项目名:统一编码可变长字符类型,长度为10,非空。

类别:普通编码定长字符类型,长度为6,取值:瑜伽、跑步、健美操。

费用:整型。

“会员”表结构:

会员号:普通编码定长字符类型,长度为18,主键。

会员名:统一编码定长字符类型,长度为10,非空。

性别:统一编码定长字符类型,长度为1,默认值“男”。

联系电话:普通编码定长字符类型,长度为11。

“健身”记录表结构:

项目号:普通编码定长字符类型,长度为10,引用“项目”表的外键。

会员号:普通编码定长字符类型,长度为18,引用“会员”表的外键。

健身日期:日期型。

健身时长:整型,取值大于等于1。(注:以小时为单位)

主键:(项目号,会员号,健身日期)。

(1) 有如下定义会员表和项目表的语句,请补全空白部分。

CREATE TABLE 会员(会员号 (1) M.char(18) primary key, 会员名 (2) N.nchar(10) not null, 性别 (3) A.nchar(1) (4) B.default '男' 联系电话 (5) E.char(11))

CREATE TABLE 项目(项目号 (6) C.char(10) primary key, 项目名 (7) G.nvarchar(10) not null, 类别 char(6) (8) F.check (9) H.类别 in ('瑜伽','跑步','健美操')), 费用 (10) D.int)

(2) 查询单次健身时长在1-3小时的会员号、项目号和健身日期。下列语句正确的是()。

B.SELECT 会员号,项目号,健身日期 FROM 健身 WHERE 健身时长 BETWEEN 1 AND 3

D.SELECT 会员号,项目号,健身日期 FROM 健身 WHERE 健身时长 >= 1 AND 健身时长 <= 3

(3) 统计“瑜伽”类项目的最高费用和最低费用。下列语句错误的是()。

B.SELECT TOP(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

C.SELECT MAX(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

D.SELECT LAG(费用),MIN(费用) FROM 项目 WHERE 类别 = '瑜伽'

(4) 现要创建满足如下要求的视图:查询参加费用超过200项目的会员名、项目名和费用。请补全空白部分(设视图名为:V1)。

CREATE VIEW V1 AS SELECT (1) A.会员名,项目名,费用 FROM 项目 a JOIN (2) D.健身 b ON a.项目号 = b.项目号 JOIN (3) C.会员 c ON (4) E.c.会员号 = b.会员号 WHERE (5) B.费用 > 200

(5) 将项目号为“X111”的项目类别改为“瑜伽”。下列语句正确的是()。

A.UPDATE 项目 SET 类别 = '瑜伽' WHERE 项目号 = 'X111'

B.UPDATE 项目 SET 类别 = '瑜伽' WHERE 项目号 LIKE 'X111'

(6) 删除联系电话后3位为“123”的会员记录。下列语句错误的是()。

B.DELETE FROM 会员 WHERE 联系电话 LIKE '123'

C.DROP FROM 会员 WHERE 联系电话 LIKE '%123'

D.DROP FROM 会员 WHERE 联系电话 LIKE '123'

56、有如下项目、会员和健身三张表,各表结构如下:

“项目”表结构:

项目号:普通编码定长字符类型,长度为10,主键。

项目名：统一编码可变长字符类型，长度为10，非空。
类别：普通编码定长字符类型，长度为6，取值：跑步、举重。
费用：整型，默认值100。

“会员”表结构：

会员号：普通编码定长字符类型，长度为18，主键。
会员名：统一编码可变长字符类型，长度为10，非空。
性别：统一编码定长字符类型，长度为1，默认值“男”。
联系电话：普通编码定长字符类型，长度为11。

“健身”记录表结构：

项目号：普通编码定长字符类型，长度为10，引用“项目”表的外键。
会员号：普通编码定长字符类型，长度为18，引用“会员”表的外键。
健身日期：日期型。
健身时长：整型。（注：以小时为单位）
主键：（项目号，会员号，健身日期）。

(1) 查询联系电话以“138”开始的会员名和联系电话。下列语句错误的是 ()。

A.SELECT 会员名, 联系电话 FROM 会员 WHERE 联系电话 LIKE '138'
C.SELECT 会员名, 联系电话 FROM 会员 WHERE 联系电话 = '138'
D.SELECT 会员名, 联系电话 FROM 会员 WHERE 联系电话 = '138%'

(2) 将项目号为“X111”的项目费用降低5%。下列语句错误的是 ()。

A.UPDATE 项目 SET 费用 = 费用 * 95% WHERE 项目号 = 'X111'
C.UPDATE 费用 SET 费用 = 费用 * 95% WHERE 项目号 = 'X111'
D.UPDATE 费用 SET 费用 = 费用 * 0.95 WHERE 项目号 = 'X111'

(3) 删除2020年1月1日之前的健身记录。下列语句正确的是 ()。

A.DELETE FROM 健身 WHERE 健身日期 < '2020-1-1'
B.DELETE FROM 健身 WHERE 健身日期 < '2020-01-01'
C.DELETE 健身 WHERE 健身日期 < '2020-1-1'
D.DELETE 健身 WHERE 健身日期 < '2020-01-01'

(4) 现要创建满足如下要求的视图：查询“男”性会员的健身情况，列出会员名、项目名和健身日期。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) B. 会员名, 项目名, 健身日期 FROM 项目 a JOIN (2) D. 健身 b ON a. 项目号 = b. 项目号 JOIN (3) C. 会员 c ON (4) E.c. 会员号 = b. 会员号 WHERE (5) A. 性别 = '男'

(5) 统计每个会员的健身总时长。下列语句错误的是 ()。

A.SELECT 会员号, MAX(健身时长) FROM 健身 GROUP BY 会员号
B.SELECT 会员号, COUNT(健身时长) FROM 健身 GROUP BY 会员号
C.SELECT 会员号, TOTAL(健身时长) FROM 健身 GROUP BY 会员号

(6) 有如下定义项目表和健身表的语句，请补全空白部分。

CREATE TABLE 项目 (项目号 char(10) (1) A.primary key, 项目名 (2) G.nvarchar(10) not null, 类别 char(6) (3) F.check (类别 (4) H.in ('跑步','举重')), 费用 int (5) B.default 100)
CREATE TABLE 健身 (项目号 char(10), 会员号 (6) M.char(18), 健身日期 (7) C.date, 健身时长 int, (8) D.foreign key (项目号),(9) E.references 项目 ((10) I.项目号))

57、有如下项目、会员和健身三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为10，主键。
项目名：统一编码可变长字符类型，长度为10，非空。
类别：普通编码定长字符类型，长度为6，取值：跑步、举重。
费用：整型，默认值100。

“会员”表结构：

会员号：普通编码定长字符类型，长度为18，主键。
会员名：统一编码可变长字符类型，长度为10，非空。
性别：统一编码定长字符类型，长度为1，默认值“男”。
联系电话：普通编码定长字符类型，长度为11。

“健身”记录表结构：

项目号：普通编码定长字符类型，长度为10，引用“项目”表的外键。
会员号：普通编码定长字符类型，长度为18，引用“会员”表的外键。
健身日期：日期型。
健身时长：整型，取值范围1-4。（注：以小时为单位）
主键：（项目号，会员号，健身日期）。

(1) 在健身表中插入一行数据：项目号：X123，会员号：H321，健身日期：2022-11-16，健身时长：2。下列语句正确的是 ()。

A.INSERT INTO 健身 VALUES('X123', 'H321', '2022-11-16', 2)
B.INSERT 健身 VALUES('X123', 'H321', '2022-11-16', 2)
C.INSERT INTO 健身(项目号, 会员号, 健身日期, 健身时长) VALUES('X123', 'H321', '2022-11-16', 2)
D.INSERT INTO 健身(项目号, 会员号, 健身时长, 健身日期) VALUES('X123', 'H321', 2, '2022-11-16')

(2) 查询“跑步”类项目的名称和费用。下列语句正确的是 ()。

A.SELECT 项目名, 费用 FROM 项目 WHERE 类别 LIKE '跑步'
B.SELECT 项目名, 费用 FROM 项目 WHERE 类别 = '跑步'

(3) 删除费用超过500的项目。下列语句正确的是 ()。

A.DELETE FROM 项目 WHERE 费用 > 500
B.DELETE 项目 WHERE 费用 > 500

(4) 统计“男”性会员人数。下列语句正确的是 ()。

B.SELECT COUNT(会员号) FROM 会员 WHERE 性别 = '男'
D.SELECT COUNT(*) FROM 会员 WHERE 性别 = '男'

(5) 现要创建满足如下要求的视图：查询“2022年10月10日”的健身情况，列出会员名、项目名和健身时长。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) B. 会员名, 项目名, 健身时长 FROM (2) C. 项目 a JOIN 健身 b ON (3) E.a. 项目号 = b. 项目号 JOIN 会员 c ON (4) D.b. 会员号 = c. 会员号 WHERE (5) A. 健身日期 = '2022-10-10'

(6) 有如下定义会员表和健身表的语句，请补全空白部分。

CREATE TABLE 会员 (会员号 char(18) (1) A.primary key, 会员名 (2) G.nvarchar(10) not null, 性别 (3) E.nchar(1) (4) B.default '男', 联系电话 (5) N.char(11))

CREATE TABLE 健身 (项目号 (6) M.char(10), 会员号 char(18), 健身日期 (7) C.date, 健身时长 int (8) F.check (健身时长 BETWEEN 1 AND 4), Foreign key(会员号) references (9) K.会员 ((10) I.会员号))

58、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为10，主键。
项目名：统一编码可变长字符类型，长度为12，非空。
类别：统一编码定长字符类型，长度为2，取值范围：{田径，游泳}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为18，主键。
运动员名：统一编码可变长字符类型，长度为10，非空。
性别：统一编码定长字符类型，长度为1，默认值：“男”。
出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为10，引用“项目”表的外键。
运动员号：普通编码定长字符类型，长度为18，引用“运动员”表的外键。
比赛日期：日期型。
比赛名次：整型。

主键：（项目号，运动员号，比赛日期）。

(1) 现要创建满足如下要求的视图：查询“张三”运动员参加比赛的项目名、比赛日期和比赛名次。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) B. 项目名, 比赛日期, 比赛名次 FROM (2) C. 项目 a JOIN 比赛 b

ON (3) D.a. 项目号 = b. 项目号 JOIN 运动员 c ON (4) A.c. 运动员号 = b. 运动员号 WHERE (5) E. 运动员名 = '张三'

(2) 统计 2022 年 10 月 15 日的比赛项目数。下列语句错误的是 ()。

A.SELECT COUNT(运动员号) FROM 比赛 WHERE 比赛日期 = '2022-10-15'

B.SELECT COUNT(项目号) FROM 比赛 WHERE 比赛日期 = '2022-10-15'

C.SELECT COUNT(DISTINCT 运动员号) FROM 比赛 WHERE 比赛日期 = '2022-10-15'

(3) 将“X001”号项目的类别改为“游泳”。下列语句错误的是 ()。

A.ALTER 项目 SET 类别 = '游泳' WHERE 项目号 = 'X001'

B.ALTER 类别 SET 类别 = '游泳' WHERE 项目号 = 'X001'

D.UPDATE 类别 SET 类别 = '游泳' WHERE 项目号 = 'X001'

(4) 查询比赛名次前 3 名的运动员号、项目号和比赛日期。下列语句错误的是 ()。

A.SELECT 运动员号, 项目号, 比赛日期 FROM 项目 WHERE 比赛名次 <= 3

C.SELECT 运动员号, 项目号, 比赛日期 FROM 运动员 WHERE 比赛名次 <= 3

D.SELECT 运动员号, 项目号, 比赛日期 FROM 比赛次数 WHERE 比赛名次 <= 3

(5) 在运动员表中插入一行数据: 运动员号: Y121, 运动员名: 张三, 性别: 男, 出生日期: 未知。下列语句正确的是 ()。

A.INSERT INTO 运动员(运动员号, 运动员名, 性别) VALUES('Y121','张三','男')

B.INSERT INTO 运动员(运动员号, 运动员名, 性别, 出生日期) VALUES('Y121','张三','男', NULL)

D.INSERT INTO 运动员 VALUES('Y121','张三','男', NULL)

(6) 有如下定义运动员表和项目表的语句, 请补全空白部分。

CREATE TABLE 运动员 (运动员号 (1) A.char(18) primary key, 运动员名 (2) G.nvarchar(10) not null, 性别 (3) C.nchar(1)(4) D.default '男', 出生日期 (5) I.date)

CREATE TABLE 项目 (项目号 (6) F.char(10) primary key, 项目名 (7) K.nvarchar(12) not null, 类别 (8) N.nchar(2) (9) H.check ((10) E.类别 in('田径','游泳'))

59、有如下项目、运动员和比赛三张表, 各表结构如下:

“项目”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 主键。

项目名: 统一编码可变长字符类型, 长度为 12, 非空。

类别: 统一编码定长字符类型, 长度为 2, 取值范围为: {田径, 游泳}

“运动员”表结构:

运动员号: 普通编码定长字符类型, 长度为 18, 主键。

运动员名: 统一编码可变长字符类型, 长度为 10, 非空。

性别: 统一编码定长字符类型, 长度为 1, 默认值: “男”。

出生日期: 日期型。

“比赛”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 引用“项目”表的外键。

运动员号: 普通编码定长字符类型, 长度为 18, 引用“运动员”表的外键。

比赛日期: 日期型。

比赛名次: 整型。

主键: (项目号, 运动员号, 比赛日期)。

(1) 现要创建满足如下要求的视图: 查询参加“男子 100 米跑”项目的运动员名和比赛名次, 结果按比赛名次升序排序。请补全空白部分 (设视图名为: V1)。

CREATE VIEW V1 AS SELECT (1) D.运动员名, 比赛名次 FROM 项目 a JOIN 比赛 b ON a. 项目号 = b. 项目号 JOIN (2) E.运动员 c ON (3) A.c. 运动员号 = b. 运动员号(4) B.WHERE 项目名 = '男子 100 米跑' (5) C.ORDER BY 比赛名次 ASC

(2) 查询 2022 年 10 月 15 日举行比赛的项目号和参赛运动员号。下列语句正确的是 ()。

A.SELECT 项目号, 运动员号 FROM 比赛 WHERE 比赛日期 = '2022-10-15'

C.SELECT 项目号, 运动员号 FROM 比赛 WHERE 比赛日期 IN ('2022-10-15')

(3) 有如下定义项目表和比赛表的语句, 请补全空白部分。

CREATE TABLE 项目 (项目号 char(10) (1) F.primary key, 项目名 (2) K.nvarchar(12) not null, 类

别 (3) N.nchar(2) (4) D.check ((5) C.类别 in('田径','游泳'))

CREATE TABLE 比赛 (项目号 char(10), 运动员号 (6) A.char(18), 比赛日期 (7) I.date, 比赛名次 (8) B.int, (9) G.foreign key (项目号) references (10) E.项目(项目号))

(4) 删除比赛名次大于 3 的比赛记录。下列语句正确的是 ()。

A.DELETE FROM 比赛 WHERE 比赛名次 > 3

B.DELETE 比赛 WHERE 比赛名次 > 3

(5) 在比赛表中插入一行数据: 项目号: X100, 运动员号: Y121, 比赛日期: 2022 年 10 月 10 日, 比赛名次: 未知。下列语句正确的是 ()。

A.INSERT INTO 比赛(项目号, 运动员号, 比赛日期) VALUES('X100','Y121','2022-10-10')

C.INSERT INTO 比赛(项目号, 运动员号, 比赛日期, 比赛名次) VALUES('X100','Y121','2022-10-10', NULL)

D.INSERT INTO 比赛 VALUES('X100','Y121','2022-10-10', NULL)

(6) 统计 2022 年 1 月 1 日之后举行的比赛项目数。下列语句错误的是 ()。

A.SELECT COUNT(运动员号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

B.SELECT COUNT(项目号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

C.SELECT COUNT(DISTINCT 运动员号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

60、有如下项目、运动员和比赛三张表, 各表结构如下:

“项目”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 主键。

项目名: 统一编码可变长字符类型, 长度为 12, 非空。

类别: 统一编码定长字符类型, 长度为 2, 取值范围为: {田径, 游泳}

“运动员”表结构:

运动员号: 普通编码定长字符类型, 长度为 18, 主键。

运动员名: 统一编码可变长字符类型, 长度为 10, 非空。

性别: 统一编码定长字符类型, 长度为 1, 默认值: “男”。

出生日期: 日期型。

“比赛”表结构:

项目号: 普通编码定长字符类型, 长度为 10, 引用“项目”表的外键。

运动员号: 普通编码定长字符类型, 长度为 18, 引用“运动员”表的外键。

比赛日期: 日期型。

比赛名次: 整型, 取值大于等于 1。

主键: (项目号, 运动员号, 比赛日期)。

(1) 删除“2021 年 10 月 10 日”之前的比赛记录。下列语句正确的是 ()。

A.DELETE FROM 比赛 WHERE 比赛日期 < '2021-10-10'

B.DELETE 比赛 WHERE 比赛日期 < '2021-10-10'

(2) 统计 2022 年 10 月 10 日参加比赛的运动员数。下列语句错误的是 ()。

A.SELECT COUNT(运动员号) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

C.SELECT COUNT(DISTINCT 项目号) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

D.SELECT COUNT(*) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

(3) 现要创建满足如下要求的视图: 查询 2000 年 1 月 1 日及以后出生的运动员参加比赛的情况, 列出项目名、运动员名、比赛日期、比赛名次。请补全空白部分 (设视图名为: V1)。

CREATE (1) C.VIEW V1 AS SELECT (2) D.项目名, 运动员名, 比赛日期, 比赛名次 FROM 项目 a JOIN 比赛 b ON (3) A.a. 项目号 = b. 项目号 JOIN (4) E.运动员 c ON c. 运动员号 = b. 运动员号 WHERE (5) B.出生日期 >= '2000-1-1'

(4) 有如下定义运动员表和比赛表的语句, 请补全空白部分。

CREATE TABLE 运动员 (运动员号 char(18) (1) F.primary key, 运动员名 (2) K.nvarchar(10) not null, 性别 (3) N.nchar(1) (4) C.default '男', 出生日期 date)

CREATE TABLE 比赛 (项目号 char(10), 运动员号 (5) A.char(18), 比赛日期 date, 比赛名次 (6) H.int check ((7) .比赛名次 >= 1), (8) G.foreign key (运动员号) references (9) E.运动员((10) B.

运动员号))

(5) 查询“张三”运动员的运动员号和出生日期。下列语句正确的是 ()。

A.SELECT 运动员号, 出生日期 FROM 运动员 WHERE 姓名 = '张三'
B.SELECT 运动员号, 出生日期 FROM 运动员 WHERE 姓名 LIKE '张三'

(6) 将项目号为“X101”的项目名改为“新项目”。下列语句错误的是 ()。

B.UPDATE 项目名 SET 项目名 = '新项目' WHERE 项目号 = 'X101'
C.ALTER 项目 SET 项目名 = '新项目' WHERE 项目号 = 'X101'
D.ALTER 项目名 SET 项目名 = '新项目' WHERE 项目号 = 'X101'

61、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 12，非空。

类别：统一编码定长字符类型，长度为 2，取值范围为：{田径，游泳}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为 18，主键。

运动员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1，默认值：“男”。

出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

运动员号：普通编码定长字符类型，长度为 18，引用“运动员”表的外键。

比赛日期：日期型。

比赛名次：整型。主键：(项目号，运动员号，比赛日期)。

(1) 有如下定义项目表和比赛表的语句，请补全空白部分。

CREATE TABLE 项目(

项目号 char(10) (1) 【F】 primary key,

项目名(2) 【K】 nvarchar(12) not null,

类别(3) 【N】 nchar(2) (4) 【D】 check ((5) 【C】 类别 in('田径','游泳'))

CREATE TABLE 比赛(

项目号 char(10),

运动员号 (6) 【A】 char(18),

比赛日期 (7) 【I】 date,

比赛名次(8) 【B】 int, (9) 【G】 foreign key (项目号) references (10) 【E】 项目(项目号)

)

(2) 多项选择题

①查询 2022 年 10 月 15 日举行比赛的项目号和参赛运动员号。下列语句正确的是 ()。

【A.】 SELECT 项目号, 运动员号 FROM 比赛 WHERE 比赛日期 = '2022-10-15'

【C.】 SELECT 项目号, 运动员号 FROM 比赛 WHERE 比赛日期 IN ('2022-10-15')

②统计 2022 年 1 月 1 日之后举行的比赛项目数。下列语句错误的是 ()。

【A.】 SELECT COUNT(运动员号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

【B.】 SELECT COUNT(项目号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

【C.】 SELECT COUNT(DISTINCT 运动员号) FROM 比赛 WHERE 比赛日期 > '2022-1-1'

③删除比赛名次大于 3 的比赛记录。下列语句正确的是 ()。

【A.】 DELETE FROM 比赛 WHERE 比赛名次 > 3

【B.】 DELETE 比赛 WHERE 比赛名次 > 3

④在比赛表中插入一行数据：项目号：X100，运动员号：Y121，比赛日期：2022 年 10 月 10 日，比赛名次：未知。下列语句正确的是 ()。

【A.】 INSERT INTO 比赛(项目号, 运动员号, 比赛日期) VALUES('X100','Y121','2022-10-10')

【C.】 INSERT INTO 比赛(项目号, 运动员号, 比赛日期, 比赛名次) VALUES('X100','Y121','2022-10-10', NULL)

【D.】 INSERT INTO 比赛 VALUES('X100','Y121','2022-10-10', NULL)

(3) 现要创建满足如下要求的视图：查询参加“男子 100 米跑”项目的运动员名和比赛名次，结果按比赛名次升序排序。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS

SELECT (1) 【D】 运动员名, 比赛名次 FROM 项目

a JOIN 比赛 b ON

a.项目号 = b.项目号 JOIN (2) 【E】 运动员 c ON (3) 【A】 c.运动员号 = b.运动员号 (4) 【B】 WHERE 项目名 = '男子 100 米跑' (5) 【C】 ORDER BY 比赛名次 ASC

62、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 12，非空。

类别：统一编码定长字符类型，长度为 2，取值范围为：{田径，游泳}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为 18，主键。

运动员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1，默认值：“男”。

出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

运动员号：普通编码定长字符类型，长度为 18，引用“运动员”表的外键。

比赛日期：日期型。

比赛名次：整型，取值大于等于 1。

主键：(项目号，运动员号，比赛日期)。

(1) 有如下定义运动员表和比赛表的语句，请补全空白部分。

CREATE TABLE 运动员(

运动员号 char(18) (1) 【F】 primary key,

运动员名 (2) 【K】 nvarchar(10) not null,

性别 (3) 【N】 nchar(1) (4) 【C】 default '男',

出生日期 date

)

CREATE TABLE 比赛(

项目号 char(10),

运动员号 (5) 【A】 char(18),

比赛日期 date,

比赛名次 (6) 【H】 int check ((7) 【I】 比赛名次 >= 1),

(8) 【G】 foreign key (运动员号) references (9) 【E】 运动员 ((10) 【B】 运动员号)

)

(2) 多项选择题

①查询“张三”运动员的运动员号和出生日期。下列语句正确的是 ()。

【A.】 SELECT 运动员号, 出生日期 FROM 运动员 WHERE 姓名 = '张三'

【B.】 SELECT 运动员号, 出生日期 FROM 运动员 WHERE 姓名 LIKE '张三'

②统计 2022 年 10 月 10 日参加比赛的运动员数。下列语句错误的是 ()。

【A.】 SELECT COUNT(运动员号) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

【C.】 SELECT COUNT(DISTINCT 项目号) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

【D.】 SELECT COUNT(*) FROM 比赛 WHERE 比赛日期 = '2022-10-10'

③删除“2021 年 10 月 10 日”之前的比赛记录。下列语句正确的是 ()。

【A.】 DELETE FROM 比赛 WHERE 比赛日期 < '2021-10-10'

【B.】 DELETE 比赛 WHERE 比赛日期 < '2021-10-10'

④将项目号为“X101”的项目名改为“新项目”。下列语句错误的是 ()。

【B.】 UPDATE 项目 SET 项目名 = '新项目' WHERE 项目号 = 'X101'

【C.】 ALTER 项目 SET 项目名 = '新项目' WHERE 项目号 = 'X101'

[D.] ALTER 项目名 SET 项目名 = '新项目' WHERE 项目号 = 'X101'

(3) 现要创建满足如下要求的视图：查询 2000 年 1 月 1 日及以后出生的运动员参加比赛的情况，列出项目名、运动员名、比赛日期、比赛名次。请补全空白部分（设视图名为：V1）。

CREATE (1) **[C.] VIEW V1 AS**

SELECT (2) **[D.] 项目名, 运动员名, 比赛日期, 比赛名次** FROM 项目
a JOIN 比赛 b ON (3) **[A.] a.项目号 = b.项目号** JOIN (4) **[E.] 运动员 c**
ON c.运动员号 = b.运动员号

WHERE (5) **[B.] 出生日期 >= '2000-1-1'**

63、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 10，非空。

类别：统一编码定长字符类型，长度为 2，取值范围为：{田径，游泳，体操}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为 18，主键。

运动员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1。

出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

运动员号：普通编码定长字符类型，长度为 18，引用“运动员”表的外键。

比赛日期：日期型。

比赛名次：整型。

主键：(项目号，运动员号，比赛日期)。

(1) 现要创建满足如下要求的视图：查询“游泳”类项目的比赛情况，列出项目名、比赛日期、运动员名。请补全空白部分（设视图名为：V1）。

CREATE (1) **D.VIEW V1 AS** SELECT (2) **E.项目名, 比赛日期, 运动员名** FROM 项目 a JOIN 比赛 b
ON (3) **C.a. 项目号 = b. 项目号** JOIN 运动员 c ON (4) **A.c. 运动员号 = b. 运动员号** WHERE (5) **B. 类别 = '游泳'**

(2) 将“Y101”运动员参加的“X001”项目号的比赛名次改为第 2 名。下列语句正确的是 ()。

A.UPDATE 比赛 SET 比赛名次 = 2 WHERE 运动员号 = 'Y101' AND 项目号 = 'X001'

B.UPDATE 比赛 SET 比赛名次 = 2 WHERE 项目号 = 'X001' AND 运动员号 = 'Y101'

(3) 统计“游泳”类的项目数量。下列语句正确的是 ()。

A.SELECT COUNT(*) FROM 项目 WHERE 类别 = '游泳'

B.SELECT COUNT(项目号) FROM 项目 WHERE 类别 = '游泳'

C.SELECT COUNT(项目名) FROM 项目 WHERE 类别 = '游泳'

(4) 查询 2000 年 10 月 10 日（包括该日期）之后出生的所有运动员名、性别和出生日期。下列语句错误的是 ()。

B.SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 >= 2000-10-10

C.SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 IS >= '2000-10-10'

D.SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 IS >= 2000-10-10

(5) 删除“体操”类的运动项目。下列语句正确的是 ()。

A.DELETE FROM 项目 WHERE 类别 = '体操'

B.DELETE FROM 项目 WHERE 类别 LIKE '体操'

(6) 有如下定义项目表和比赛表的语句，请补全空白部分。

CREATE TABLE 项目 (项目号 char(10) (1) **D.primary key**, 项目名 (2) **G.nvarchar(10)** (3) **A.not null**, 类别 (4) **H.nchar(2)** (5) **F.check** (类别 in ('田径','游泳','体操'))

CREATE TABLE 比赛 (项目号 char(6), 运动员号 (6) **M.char(18)**, 比赛日期 (7) **C.date**, 比赛名次 int , (8) **K.foreign key** (项目号) (9) **I.references** 项目((10) **E.项目号**))

B.char(2) N.nchar(18)

64、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 10，非空。

类别：统一编码定长字符类型，长度为 2，取值范围为：{田径，游泳，体操}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为 18，主键。

运动员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1。

出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

运动员号：普通编码定长字符类型，长度为 18，引用“运动员”表的外键。

比赛日期：日期型。

比赛名次：整型。

主键：(项目号，运动员号，比赛日期)。

(1) 有如下定义项目表和比赛表的语句，请补全空白部分。

CREATE TABLE 项目(

项目号 char(10) (1) **[D.] primary key**,

项目名(2) **[G.] nvarchar(10)** (3) **[A.] not null**,

类别(4) **[H.] nchar(2)** (5) **[F.] check** (类别 in ('田径','游泳','体操'))

)

CREATE TABLE 比赛(

项目号 char(6),

运动员号(6) **[M.] char(18)**,

比赛日期(7) **[C.] date**,

比赛名次 int,

(8) **[K.] foreign key** (项目号) (9) **[I.] references** 项目((10) **[E.] 项目号**)

)

(2) **【多选题】**

1) 查询 2000 年 10 月 10 日（包括该日期）之后出生的所有运动员名、性别和出生日期。下列语句错误的是 ()。

[B.] SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 >= 2000-10-10

[C.] SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 IS >= '2000-10-10'

[D.] SELECT 运动员名, 性别, 出生日期 FROM 运动员 WHERE 出生日期 IS >= 2000-10-10

2) 统计“游泳”类的项目数量。下列语句正确的是 ()。

[A.] SELECT COUNT(*) FROM 项目 WHERE 类别 = '游泳'

[B.] SELECT COUNT(项目号) FROM 项目 WHERE 类别 = '游泳'

[C.] SELECT COUNT(项目名) FROM 项目 WHERE 类别 = '游泳'

3) 将“Y101”运动员参加的“X001”项目号的比赛名次改为第 2 名。下列语句正确的是 ()。

[A.] UPDATE 比赛 SET 比赛名次 = 2 WHERE 运动员号 = 'Y101' AND 项目号 = 'X001'

[B.] UPDATE 比赛 SET 比赛名次 = 2 WHERE 项目号 = 'X001' AND 运动员号 = 'Y101'

4) 删除“体操”类的运动项目。下列语句正确的是 ()。

[A.] DELETE FROM 项目 WHERE 类别 = '体操'

[B.] DELETE FROM 项目 WHERE 类别 LIKE '体操'

(3) 现要创建满足如下要求的视图：查询“游泳”类项目的比赛情况，列出项目名、比赛日期、运动员名。请补全空白部分（设视图名为：V1）。

CREATE (1) **[D.] VIEW V1 AS** SELECT (2) **[E.] 项目名, 比赛日期, 运动员名** FROM

项目 a JOIN 比赛 b ON (3) **[C.] a.项目号 = b.项目号**

JOIN 运动员 c ON (4) **[A.] c.运动员号 = b.运动员号**

WHERE (5) **[B]** 类别 = '游泳'

65、有如下项目、运动员和比赛三张表，各表结构如下：

“项目”表结构：

项目号：普通编码定长字符类型，长度为 10，主键。

项目名：统一编码可变长字符类型，长度为 10，非空。

类别：统一编码定长字符类型，长度为 2，取值范围为：{田径，游泳，体操}

“运动员”表结构：

运动员号：普通编码定长字符类型，长度为 18，主键。

运动员名：统一编码可变长字符类型，长度为 10，非空。

性别：统一编码定长字符类型，长度为 1，取值：“男”、“女”。

出生日期：日期型。

“比赛”表结构：

项目号：普通编码定长字符类型，长度为 10，引用“项目”表的外键。

运动员号：普通编码定长字符类型，长度为 18，引用“运动员”表的外键。

比赛日期：日期型。

比赛名次：整型。

主键：(项目号，运动员号，比赛日期)。

(1) 现要创建满足如下要求的视图：查询 2022 年 10 月 18 日举行的比赛情况，列出项目名、运动员名和比赛名次，结果按比赛名次降序排序。请补全空白部分（设视图名为：V1）。

CREATE VIEW V1 AS SELECT (1) E.项目名, 运动员名, 比赛名次 FROM 项目 a JOIN (2) C.比赛 b ON a. 项目号 = b. 项目号 JOIN 运动员 c ON (3) A.c. 运动员号 = b. 运动员号 (4) B.WHERE 比赛日期 = '2022-10-18' (5) D.ORDER BY 比赛名次 DESC

(2) 统计每类项目的的项目数。下列语句正确的是 ()。

A.SELECT 类别, COUNT(*) FROM 项目 GROUP BY 类别

B.SELECT 类别, COUNT(项目号) FROM 项目 GROUP BY 类别

(3) 在项目表中插入一行数据，项目号：X100，项目名：男子 100 米跑，类别：田径。下列语句正确的是 ()。

A.INSERT INTO 项目 VALUES('X100','男子 100 米跑','田径')

C.INSERT 项目 VALUES('X100','男子 100 米跑','田径')

(4) 有如下定义运动员表和比赛表的语句，请补全空白部分。

CREATE TABLE 运动员 (运动员号 char(18) (1) D.primary key, 运动员名 (2) G.nvarchar(10) not null, 性别 (3) N.nchar(1) check((4) E.性别) (5) C.in ('男','女'), 出生日期 date)

CREATE TABLE 比赛 (项目号 (6) A.char(10), 运动员号 char(18), 比赛日期 date, 比赛名次 (7) M.int, (8) K.foreign key (项目号) (9) I.references 项目((10) H.项目号))

(5) 查询“田径”和“游泳”两类项目的项目号和项目名。下列语句正确的是 ()。

C.SELECT 项目号, 项目名 FROM 项目 WHERE 类别 = '田径' OR 类别 = '游泳'

D.SELECT 项目号, 项目名 FROM 项目 WHERE 类别 in ('田径', '游泳')

(6) 将“张三”运动员的出生日期改为“2000 年 2 月 4 日”。下列语句正确的是 ()。

C.UPDATE 运动员 SET 出生日期 = '2000-2-4' WHERE 运动员名 = '张三'

D.UPDATE 运动员 SET 出生日期 = '2000-02-04' WHERE 运动员名 = '张三'