

# 2019年春季学期《数据库系统及应用》期末试题

---

Edited by [Lyncien](#)

2019.06.25

该部分为个人答案，仅供参考

## 一、选择题 5 \* 2%

1. 下面哪个不是数据库系统相对于文件系统的优点
  - A. 一致性高
  - B. 冗余小
  - C. 独立性高
  - D. 结构化程度高
2. 现实世界中的一个实体是通过关系数据库中的来表示的
  - A. 元组
  - B. 关系
  - C. 主码
  - D. 候选码
3. Student表的gender属性要求不为空，如果插入时空，则自动填充“NA”，下面哪种方式无法实现该功能
  - A. 触发器
  - B. 存储过程
  - C. Default
  - D. Check约束
4. 下面哪个操作不能在视图上操作
  - A. 创建视图
  - B. 删除记录
  - C. 聚集查询
  - D. 修改记录
5. 建立索引不需要考虑的步骤是
  - A. 决定索引文件的具体存储位置
  - B. 在哪些表上建立索引
  - C. 在哪些列上建立索引
  - D. 采用哪种索引结构

1. D
2. A
3. D

4. A

5. A

## 二、判断题 10 \* 2%

1. ER模型的实体不能只有一个属性
2. 市面上主流的数据库一般都采用自主访问控制
3. 数据库日志将事务对数据库的详细操作都记录下来
4. SQL的基本表可以没有Foreign Key约束，但是必须要有Primary Key约束
5. 外码参照的属性必须被Primary Key约束或者Unique约束
6. 按Redo日志，数据一旦write就马上存储到磁盘
7. 如果DBMS不支持多粒度锁，就没必要支持意向锁
- 8.
- 9.
- 10.

1. F

2. F

3. T

4. F

5. T

6. F

7. T

8.

9.

10.

## 三、回答以下数据库体系结构的问题 15%

1. 数据库的三级模式在关系数据库中是如何实现的？
2. 关系数据库中操作三种模式的语句有哪些？（三种分开写）
3. 解释逻辑独立性并举例

1.

2.

3. 当概念模式发生改变时，只要修改外模式/模式映像，可保持外模式不变，从而保持用户应用程序不变，保证了数据与用户程序的逻辑独立性；

## 四、回答以下数据库并发的问题 15%

1. 什么是两阶段锁？
2. 隔离级别“可重复读”的含义是什么？
3. 采用两阶段锁的事务是否一定不会出现不可重复读的问题？若是，说明理由，若否，举反例

1. (1)事务在对任何数据进行读写之前，首先要获得该数据上的锁 (2)在释放一个锁之后，事务不再获得任何锁

2. (1)保证事务在事务内部如果重复访问同一数据（记录集），数据不会发生改变。即，事务在访问数据时，其他事务不能修改正在访问的那部分数据
  - (2)可重复读可以防止脏读和不可重复读取，但不能防止幻像
  - (3)事务必须在所访问数据上加S锁，防止其他事务修改数据，而且S锁必须保持到事务结束
3. 否

## 五、 $F = \{A \rightarrow BC, B \rightarrow CE, A \rightarrow B, AB \rightarrow C, AC \rightarrow DE, E \rightarrow A\}$ 16%

1. 求最小函数依赖集
2. 求候选码
3. 满足第几范式？为什么？
4. 无损且保持函数依赖地分解到3NF

1. (1)将右边写出单属性并去除重复FD（分解律）

$A \rightarrow BC$ 分为 $A \rightarrow B, A \rightarrow C$ ,  $B \rightarrow CE$ 分为 $B \rightarrow C, B \rightarrow E$ ,  $AC \rightarrow DE$ 分为 $AC \rightarrow D, AC \rightarrow E$

$F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow E, AC \rightarrow D, AC \rightarrow E, E \rightarrow A\}$

- (2)消去左部冗余属性

$A \rightarrow C$ , 则 $AA \rightarrow AC$ , 即 $A \rightarrow AC$ , 又 $AC \rightarrow D$ , 所以 $A \rightarrow D$ , 所以可以去除 $AC \rightarrow D$

同理, 加入 $A \rightarrow E$ , 去除 $AC \rightarrow E$

$F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow E, A \rightarrow D, A \rightarrow E, E \rightarrow A\}$

- (3)消去冗余函数依赖

$A \rightarrow C$ 可由 $A \rightarrow B$ 和 $B \rightarrow C$ 推出,  $A \rightarrow E$ 可由 $A \rightarrow B$ 和 $B \rightarrow E$ 推出

$F = \{A \rightarrow B, B \rightarrow C, B \rightarrow E, A \rightarrow D, E \rightarrow A\}$

2.  $U = \{A, B, C, D, E\}$ , 由于 $\{A\} \rightarrow U$ 属于 $F^+$ , 所以 $\{A\}$ 是超码, 又显然不存在 $\{A\}$ 的真子集 $Y$ 使得 $Y \rightarrow U$ , 所以 $\{A\}$ 是候选码
3. 只要是关系模式就满足1NF; 由于 $A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E$ , 所以非主属性都完全函数依赖于主码 $\{A\}$ , 故满足2NF; 由于 $A \rightarrow B, B \rightarrow E$ ,  $E$ 传递依赖于 $A$ , 故不满足3NF。
4. 保持函数依赖: 对 $F$ 按相同的左部分组, 并将每组涉及的所有属性作为一个关系模式输出。

$\{A \rightarrow B, A \rightarrow D\}, \{B \rightarrow C, B \rightarrow E\}, \{E \rightarrow A\}$

$p = \{R1(A, B, D), R2(B, C, E), R3(A, E)\}$

无损连接: 主码为 $\{A\}$ ,  $R4(A)$ 应该加入 $p$ , 但 $R4$ 是 $R1$ 的子集, 因此去除

最后 $p = \{R1(A, B, D), R2(B, C, E), R3(A, E)\}$

## 六、学生表student(sid#, sname, department), 课程表course(cid#, cname, ctype), 选课表SC(sid#, cid#, score), 按要求编写SQL语句 17%

1. 姓'李'的学生学号、姓名
2. 学号为'S001'的学生选修的课程的课程号、课程名
3. 按照department统计选了'DB'课程但没有成绩的学生人数（结果两列，department和人数）
4. 选了'DB'课程和'WEB'课程且分数一样的学生的学号、姓名
5. 选修课程数不小于5且每门课分数都不低于90分的学生的学号、姓名、选修课程数、平均成绩，结果按平均成绩降序排序（结果四列）

1. 

```
select sid, sname
from student
where sname like '李%'
```

2.

```
select cid, cname
from student S, course C, SC
where S.sid = 'S001' and S.sid = SC.sid and SC.cid = C.cid
```

3.

```
select department, count(*) as snum
from student S, course C, SC
where C.cname = 'DB' and SC.score is null and S.sid = SC.sid and SC.cid = C.cid
```

4.

```
select S.sid, S.sname
from student S, course C1, SC SC1, course C2, SC SC2
where C1.cname = 'DB' and C2.cname = 'WEB' and SC1.score = SC2.score and S.sid = SC1.sid and SC1.cid = C1.cid and S.sid = SC2.sid and SC2.cid = C2.cid
```

```
select S.sid, S.sname, count(cid) as cnum, avg(SC.score) as avg_score
from student S, SC
where S.sid = SC.sid
group by S.sid, S.sname
having count(cid) >= 5 and min(score) >= 90
```

5.

## 七、关系数据库的缺点，举出3点并详细说明 7%

1. 可扩展性较差，无法较好支持海量数据存储
2. 无法满足数据高并发的需求
3. 事务机制影响系统的整体性能等

### 三、 1.

数据库的三级模式架构包括外模式（视图）、概念模式（逻辑模式）和内模式（物理模式）。在关系数据库中，这三级模式的实现如下：

#### a. 外模式（External Schema）：

外模式也称为用户视图，是用户或应用程序与数据库交互的接口。每个用户可以有不同的视图，视图是从基本表中派生出来的。

在关系数据库中，外模式通过视图（VIEW）来实现。视图可以限制用户访问数据的范围，提供特定的数据表示形式。

#### b. 概念模式（Conceptual Schema）：

概念模式是数据库的整体逻辑结构，描述了所有数据及其关系。它独立于物理存储细节。

在关系数据库中，概念模式通过数据库的逻辑结构实现，包括表（TABLE）、约束（CONSTRAINT）、索引（INDEX）等。

#### c. 内模式（Internal Schema）：

内模式是数据库的物理存储结构，描述数据在存储设备中的组织方式和存取方法。

在关系数据库中，内模式通过存储结构、文件组织、索引结构等实现。数据库管理系统（DBMS）负责管理这些物理存储细节。

### 2.

外模式（视图）操作语句：

创建视图：

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

删除视图：

```
DROP VIEW view_name;
```

概念模式（表、约束、索引）操作语句：

创建表：

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    ...  
);
```

添加约束：

```
ALTER TABLE table_name  
ADD CONSTRAINT constraint_name  
PRIMARY KEY (column1, column2, ...);
```

### 3.

逻辑独立性（Logical Independence）是指在不改变外模式（用户视图）的情况下，对概念模式（逻辑模式）进行修改的能力。逻辑独立性允许数据库管理员在不影响应用程序或用户视图的情况下，修改数据库的逻辑结构，如添加新字段、删除字段或改变字段的类型。

举例：

假设有一个员工数据库，初始的概念模式如下：

```
CREATE TABLE Employee (  
    EmpID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Department VARCHAR(100)  
);
```

用户视图：

```
CREATE VIEW EmployeeView AS  
SELECT EmpID, Name, Department  
FROM Employee;
```

现在，数据库管理员决定在 Employee 表中添加一个新的字段 Email：

```
ALTER TABLE Employee  
ADD Email VARCHAR(100);
```

此时，用户视图 EmployeeView 仍然可以正常工作，因为它没有依赖于新添加的 Email 字段。这就是逻辑独立性：概念模式可以修改，而不影响外模式（用户视图）。