

中国科学技术大学

2019--2020 学年第二学期考试试卷

考试形式：闭卷笔试，计算器禁用

考试科目：数据库系统及应用 得分：

学生所在系： 姓名： 学号：

[注意]：所有答案（包括选择和判断题）都写到答题纸上。交卷时答题纸、试卷和草稿纸一起上交。

一、单项选择题（10 分）

1. 下面哪一项是数据库技术与文件系统相比的优点？
B A. 支持数据共享 B. 数据一致性高 C. 数据无冗余 D. 支持海量数据存储
2. 在关系数据模型中，现实世界中的一个实体是通过下面哪一个概念进行表示的？
A A. 元组 B. 超码 C. 主码 D. 候选码
3. 现有一个 student 表，其中有个字符串类型的字段 gender 要求不能为空。因此，我们希望在插入一条 student 记录时，如果 gender 为空则自动填上“NA”，请问下列哪一项技术无法实现这一功能？
D A. 触发器 B. Default 值 C. 存储过程 D. Check 约束
4. 下列哪个操作不可能在视图上执行？
B A. 定义新视图 B. 通过视图修改基本表结构 C. 聚集查询 D. 删除记录
5. 下面哪一项不是数据库物理设计阶段的任务？
A A. 设计必要的存储过程和触发器 B. 确定某个表的哪些列需要设计索引
C. 确定索引文件的具体存储位置 D. 确定数据库文件的磁盘块大小

二、判断题。正确的打√，错误的打×（20 分）

1. 事务的“可重复读”隔离级别意味着一个事务内部重复执行同一条 Select 语句必定返回相同的记录集 (X) *phantom read*
2. 如果 DBMS 不支持多粒度锁，则没必要实现意向锁 (✓)
3. 按照 Redo 日志，事务在执行 write 操作后可以立即将更新的数据写入磁盘 (X) *write commit first*
4. 安全 DBMS 或者可信 DBMS 要求必须实现强制访问控制，且达到 B1 安全级别 (✓)
5. 一个 SQL 基本表可以没有 Unique 约束，但不能没有 Primary Key 约束 (X)
6. 在数据库系统中，对用户使用的数据视图的描述称为概念模式 (X)
7. 任何一个满足 BCNF 的关系模式也必定满足 3NF (✓)
8. SQL 中的 Foreign Key 所引用的列必须要有 Unique 或者 Primary Key 约束 (✓)
9. 在 ER 模型中，不允许出现只包含一个属性的实体。 (X)
10. 如果调度中的事务都遵循 2PL，则该调度必定可串，但不一定冲突可串 (✓)

外模式——视图
概念模式——基本表
内模式——文件

三、(10 分) 回答下面关于数据库体系结构的问题：

1. 数据库三级模式结构在 SQL 数据库中是如何实现的？
2. 什么是数据的逻辑独立性？请举例说明。

四、(15 分) 请回答下面关于事务和日志的问题：不需要，事务日志保证数据一致性，select 等

1. 事务日志是否需要记录事务的所有 DML 操作？为什么？只读句不用记录。
2. 目前许多 DBMS 例如 MySQL 都默认不支持嵌套事务（即在一个事务内部又启动了另一个事务），请分析一下：如果 DBMS 支持嵌套事务，将面临哪些问题（至少写出 2 点并且要给出自己的分析）？①嵌套事务 commit 后，外层事务 rollback 无法撤销已完成事务 ②锁机制下两个事务读写同一数据，会死锁
3. 若 DBMS 同时采用 Redo 日志和 2PL，还会出现脏读问题吗？如果会出现，请给出一个例子；如果不会出现，请给出证明。会。

五、(15 分) 已知有关系模式 $R(A, B, C, D, E)$ ， R 上的一个函数依赖集如下：

1. ① $A \rightarrow B, A \rightarrow D, BC \rightarrow D, CE \rightarrow A, D \rightarrow B, E \rightarrow D$
 $E \rightarrow ED \Rightarrow CE \rightarrow DCE \Rightarrow CE \rightarrow A \Rightarrow CE \rightarrow A$
 ② $A \rightarrow D, D \rightarrow B \Rightarrow A \rightarrow B$
 $A \rightarrow D, BC \rightarrow D, CE \rightarrow A, D \rightarrow B, E \rightarrow D$

$$F = \{A \rightarrow BD, BC \rightarrow D, DCE \rightarrow A, D \rightarrow B, E \rightarrow D\}$$

2. $\{C, E\}$

1. 求出 F 的最小函数依赖集
2. 求 R 的候选码
3. R 属于第几范式？为什么？1NF，D 部分函数依赖于主码
4. 请将 R 无损连接并且保持函数依赖地分解到 3NF。

T_1
 $XLCAI$
 $Read(A,t)$
 $t = t + 1$
 $Write(A,t)$
 $XU(A)$
 $commit$
 T_2
 $SL(A)$
 $read(A,t)$
 $SU(A)$
 $commit$

六、(20 分) 给定下面的基本表：学生 (student)、课程 (course)、系 (department)、教师 (faculty) 和选课 (SC)：student(sid, sname, did), course(cid, cname, room, fid), department(did, dname, location), faculty(fid, fname, sex, did), SC(sid, cid, score)

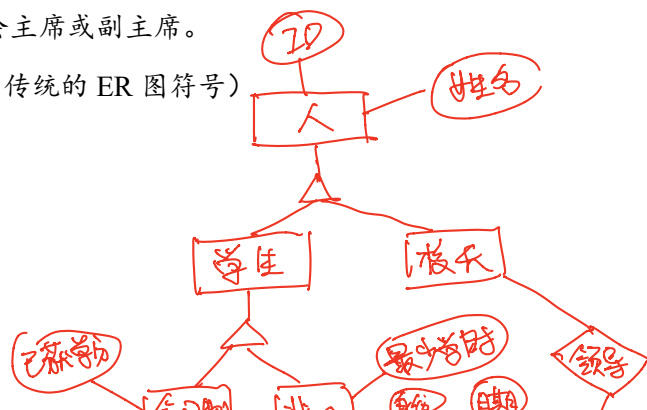
其中加下划线的字段是主键，score 字段是整型，其它字段都是字符串类型。

请用 SQL 语句完成下列查询（要求：只能用一个 SQL 语句）：

1. 查询在“3C102”上课的所有姓“赵”的“计算机”系老师姓名；
2. 查询选过课的学生中只选了“张三”老师所授课程的学生学号和姓名；
3. 查询每个系选修了“DB”课程但缺少成绩的学生人数，要求返回两列：一列显示系名称 (department)，另一列显示学生人数，并且查询结果按学生人数降序排列；
4. 查询选修过“DB”和“AI”课程但“DB”课程成绩不低于“AI”成绩的学生学号和姓名；
5. 查询选课数不少于 4 门并且各科成绩均不低于 95 的学生姓名、选修课程数和平均成绩。

七、(10 分) 假设我们准备设计一个数据库用于存储高校的相关信息。已知该数据库有下面的一些特性：1) 每所高校需要记录校名以及一个唯一的 ID；2) 每所高校至少要有一名学生和一位校长；3) 每名学生需要记录姓名和唯一的 ID；一名可以属于一个或多个高校；我们还需要记录每名学生进入某所高校的日期以及身份 (本科生还是研究生)；4) 学生可以是全日制的，也可以是非全日制的；对于非全日制学生，需要记录他们每周的最少学时数；对于全日制学生，需要记录他们已经取得的总学分；5) 校长需要记录其姓名和一个唯一的 ID，且只能担任一所学校的校长；6) 每所学校可以设置一名学生会主席和若干名副主席，要求学生会主席或副主席只能是非全日制学生，并且一名学生只能担任一所学校的学生会主席或副主席。

1. 请根据上述需求画出 ER 图（使用传统的 ER 图符号）
2. 将 ER 模型转换为关系模型



1. select faculty.fname
 from course, department, faculty
 where course.fid = faculty.fid and
 faculty.did = department.did and
 department.dname = "计算机" and
 faculty.fname LIKE "赵%" and
 course.room = "3C301"



2. select sid, sname from student
 where exists
 select * from student SC where
 SC.sid = student.sid)
 and not exists (
 select * from faculty, SC, course
 where course.fid = faculty.fid and
 student.sid = SC.sid and
 course.cid = SC.cid and
 faculty.fname <> "赵三")

3. select department.dname, count(*)
 from student, SC, course, department
 where student.sid = SC.sid and
 course.cid = SC.cid and
 department.did = student.did and
 course.cname = "DB" and
 SC.score is NULL
 group by department.did
 order by count(*) DESC

4. select student.sid, student.sname
 from student, (select SC.sid as sid, score
 from SC, course where
 SC.cid = course.cid and
 course.cname = "AI") as AI
 (select SC.sid as sid, score
 from SC, course where
 SC.cid = course.cid and
 course.cname = "DB") as DB
 where student.sid = AI.sid and
 student.sid = DB.sid and
 AI.score <= DB.score

```
5. select student.name, TB.total, TB.ave_score
from (select sid, AVE(score) as ave_score, COUNT(*) as total
      from SC group by sid) as TB, student
where TB.sid = student.sid and
      TB.total >= 4 and
      TB.ave_score >= 95
```