

一、选择题（单选题，本题共 10 分，每小题 1 分）

1. 下列实体类型的联系中，属于一对一联系的是 (C)
A) 教研室对教师的所属联系 B) 父亲对孩子的亲生联系
C) 省对省会的所属联系 D) 供应商与工程项目的供货联系
2. 在关系代数的连接操作中，哪一种连接操作需要删除重复列？ (A)
A) 自然连接 B) 笛卡尔积 C) 等值连接 D) θ 连接
3. 设有关系 $R(A, B, C)$ 和 $S(C, D)$ ，与关系代数表达式 $\pi_{A, B, D}(\sigma_{R.C=S.C}(R \times S))$ 等价的 SQL 语句是 (B)。
A) `SELECT * FROM R, S WHERE R.C=S.C`
B) `SELECT A, B, D FROM R, S WHERE R.C=S.C`
C) `SELECT A, B, D FROM R, S WHERE R=S`
D) `SELECT A, B FROM R WHERE (SELECT D FROM S WHERE R.C=S.C)`
4. 已知关系模式 $R(A, B, C, D, E)$ 及其上的函数依赖集 $F=\{A \rightarrow D, B \rightarrow C, E \rightarrow A\}$ ，该关系模式的候选码是 (B)。
A) AB B) BE C) CD D) DE
5. 下列关于关系模式的码的叙述中，哪一项是不正确的？ (C)
A) 当候选码多于一个时，选定其中一个作为主码
B) 主码可以是单个属性，也可以是属性组
C) 不包含在主码中的属性称为非主属性
D) 若一个关系模式中的所有属性构成码，则称为全码
6. 下列哪一条不是事务的基本性质？ (B)
A) 原子性 B) 安全性 C) 一致性 D) 隔离性
7. 索引是为了解决关系数据库中 (B) 问题而引入的。
A) 插入异常、删除异常和数据冗余 B) 提高查询速度
C) 减少数据操作的复杂性 D) 保证数据的安全性和完整性。
8. 数据库中的封锁机制是 (C) 的主要方法。
A) 完整性 B) 安全性 C) 并发控制 D) 恢复
9. 游标的使用主要有四个步骤，请问下面哪个步骤不是游标使用中四个主要步骤所涉及到的 (B)
A) CLOSE B) SELECT C) FETCH D) OPEN
10. 关系模式的规范化程度越高，可能会使得下面的 (C) 操作性能更差？
A) INSERT B) UPDATE C) SELECT D) DELETE

二、简答题（本题共 20 分）

1. 假设要开发一个高校的教务管理系统，系统的主要用户角色包括学生、教师、学院教务员、学校教务员。请为该系统的数据库设计合适的安全控制策略。（8 分）

参考答案：

- 1) **用户标识和鉴别**。所有用户在使用系统时需要注册账号、设置密码；进入系统时需要进行身份鉴别。（2 分）
- 2) 账号和密码在从客户端向服务器端传送时需要**加密**；在服务器端存储时加密。（2 分）
- 3) 不同的用户角色通过**授权**机制赋予不同的权限，同时可以采用**视图**辅助权限控制。例如对于某学院的教务员可以通过视图限制其只能访问该学院学生的选课数据和成绩数据。（2 分）
- 4) 对重要的表，例如成绩表进行**审计**。

2. 数据库系统采用三级模式结构。试述三级模式结构的含义以及采用这种体系结构的优点。（6 分）

参考答案：

三级模式结构包括外模式、模式、内模式以及外模式/模式映像和模式/内模式映像。模式是数据库中全体数据的逻辑结构和特征的描述；外模式是局部数据的逻辑结构和特征的描述；内模式是数据库存储结构的描述。（3 分）

外模式/模式映像提高了数据的逻辑独立性；模式/内模式映像提高了数据的物理独立性；此外，三级模式结构还具有简化用户接口、利于数据共享、利于安全保密的优点。（3 分）

3. 关系数据库有哪几类完整性？请分别举例说明其实现方法。（6 分）

参考答案：

关系数据库完整性有实体完整性、参照完整性和用户自定义完整性。（3 分）

实体完整性是指关系中的主属性不能取 NULL 值且主键取值要保持唯一性，通过定义主键来实现（举例略）。参照完整性是指关系中外键的取值必须是另一个关系主键的有效值，或者取 NULL 值，通过定义外键来实现（举例略）。用户自定义完整性是指具体应用中数据要满足的约束条件，可通过定义约束（constraints）、规则（rules）或者编写程序（触发器、存储过程）来实现（举例略）。（3 分）

三、设计题 (本题共 20 分)

宠物管理机构需要设计开发一套宠物管理系统，包括宠物、宠物店及宠物主人等相关对象进行信息与服务管理，其主要需求与主要属性、关联关系有：

(1) **宠物**（宠物 ID、品种、毛色、年龄、特征描述）

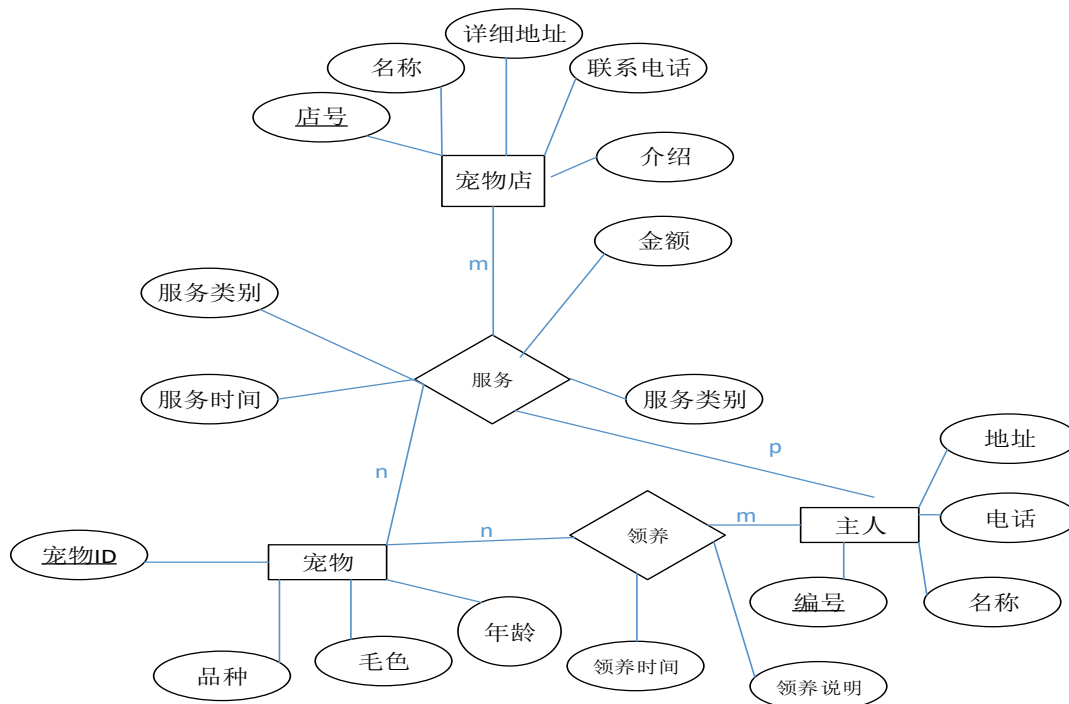
(2) **宠物店**（店号、名称、详细地址、联系电话、介绍），主要为宠物主人提供**宠物本身的买、卖及清洁、治疗、疫苗接种等服务及宠物饲料、玩具等购买服务**（提示：可以统一为服务类别），要求记录服务时间与服务金额及服务具体内容。

(3) **宠物主人**（编号, 名称（可能不是个人），电话，地址），主人与宠物需要有一个直接的**领养关系**记录，宠物可能存在不同领养时间被不同主人领养的情况。

1. 请根据需求画出 ER 图（10 分）。
2. 将 ER 图转换成关系模型（10 分）。

1. 请根据需求画出 ER 图（10 分）。

参考 ER 图如下，其中包括 3 个实体（包含主要属性），3 分；2 个联系及联系的主要属性、联系的类别（1-n, m-n），4 分（仅有服务联系，记 3 分）；视是否标记键属性、实体及联系的属性是否完整，1-3 分。



2. 将 ER 图转换成关系模型（10 分）。

参考答案：

按转换规则，ER 图中实体可以直接转换为关系，可以是关系模式（仅需要关系及属性名），也可以是关系（表、有表名、字段名、类型、长度、主键、外键）：

(1) **宠物**（宠物 ID、品种、毛色、年龄、特征描述）

- (2) 宠物店 (店号、名称、详细地址、联系电话、介绍)
- (3) 宠物主人 (主人编号, 名称, 电话, 地址),
- (4) 服务 (店号、主人编号, 宠物 ID, 服务类别, 服务时间, 服务金额, 服务内容描述)
- (5) 领养 (主人编号, 宠物 ID, 领养时间, 领养说明)

有 3 个实体转化的关系或关系模式, 3 分, 有关键属性标识, 2 分

联系的转化:

服务转化为关系模式, 能体现 3 个实体的关联关系, 同时能体现服务类别等关键属性, 3 分

领养为关系模式, 能体现主人与宠物的关联关系, 同时能体现领养时间等关键属性, 2 分

如果没有将领养单独形成服务, 直接归并到服务中, 可记 1 分 (总分 9 分)

四、分析题 (本题共 15 分)

教学管理关系 TM (S (学号), C (课程号), G (成绩), T (教师), D (教师所在系)), 满足的函数依赖 $F=\{(S,C)\rightarrow G, C\rightarrow T, T\rightarrow D\}$, 请解答:

1. 如果用上述属性组成一个关系模式, 那么其属于第几范式? 并说明理由 (4 分)
2. 举例说明在进行增、删操作时的异常现象 (3 分)
3. $(CG)_F^+$ (4 分)
4. 将其分解使其满足 3NF (4 分)

参考答案:

(1) 1NF, 该关系模式的候选码为(S, C), 则非主属性有 G、T、D; 又 F 中有 $C\rightarrow T$, 故存在非主属性 T 对候选码(S, C)的部分函数依赖。

(2) 若新增一门课程而暂时还没有学生选修时, 则因缺少主属性 S 值而不能进行插入操作; 若某个教师调离学校要删除其有关信息是, 会将不改伤处的课程 (C) 信息删除。

(3) $(CG)_F^+=CGTD$ 。

(4) $R_1=(S, C, G) \quad F_1=\{(S,C)\rightarrow G\}$

$R_2=(C, T) \quad F_2=\{C\rightarrow T\}$

$R_3=(T, D) \quad F_3=\{T\rightarrow D\}$

五、程序题 (本题 35 分)

设有如下表结构:

Employee (员工表) (EmpID, EmpName, Gender, Age, DeptID, Salary) (工号, 姓名, 性别, 年龄, 部门 ID, 月薪);

Department (部门表) (DeptID, DeptName, Location) (部门 ID, 部门名称, 所在地)

Project (项目表) (ProjID, ProjName, ManagerID, StartDate, EndDate) (项目 ID, 项目名称, 项目经理工号, 开始日期, 结束日期)

EmployeeProject (员工参与项目表) (EmpID, ProjID, Role) (工号, 项目 ID, 角色)

请使用关系代数表达式、SQL 语句或编写程序完成下列任务:

1. [关系代数表达式] 查询某项目 (项目 ID 为 “202401”) 中年龄在 30-40 岁之间的员工姓名和角色。(3 分)
2. [关系代数表达式] 查询所有未参与任何项目的员工工号和姓名。(3 分)
3. [SQL 语句] 在员工表 (Employee) 中为新入职的员工插入一条记录, 工号为 “E1008”, 姓名为 “张三”, 性别为 “男”, 年龄为 “22”, 部门 ID 为 “002”, 月薪为 “8000”。(3 分)
4. [SQL 语句] 修改部门所在地在 “长沙” 的所有员工的月薪, 使其增加 10%。(3 分)
5. [SQL 语句] 创建一个视图, 展示 2020 年 1 月 1 日及其之后开始的所有项目的全部信息, 以及项目经理的姓名。结果按项目开始日期降序排列 (5 分)
6. [SQL 语句] 列出不同所在地员工的数量和平均月薪 (5 分)

7. [SQL 语句] 查询所有参与了至少三个项目的员工的工号和姓名。(6 分)

8. [编写触发器] 创建一个触发器, 当在员工参与项目表 (EmployeeProject) 中插入一条新记录时 (即某项目有一个新员工的加入), 会根据新加入项目的员工年龄大小, 在 Employee 表中增加对应员工的月薪。如果员工年龄小于 30 岁, 月薪提高 500 元, 如果年龄在 30 岁以上 (包括 30 岁), 月薪提高 800 元。(7 分)

参考答案 (以 SQL SERVER 标准编写, 部分题目会有多种程序写法, 特别是表连接查询语法具有多样性, 请老师们酌情批改):

1、 $\pi_{EmpName, Role}(\sigma_{ProjID='202401' \wedge Age \geq 30 \wedge Age \leq 40}(Employee \bowtie EmployeeProject))$

1分

1分

1分

2、 $\pi_{EmpID, EmpName}(Employee) - \pi_{EmpID, EmpName}(Employee \bowtie EmployeeProject)$

1 分

1 分

1 分

3、INSERT INTO Employee (EmpID, EmpName, Gender, Age, DeptID, Salary) 2分
VALUES ('E1008', '张三', '男', 22, '002', 8000), 1分

4、UPDATE E

SET E.Salary = E.Salary * 1.1 1 分

FROM Employee E

INNER JOIN Department D ON E.DeptID = D.DeptID 1 分

WHERE D.Location = '长沙'; 1 分

5、CREATE VIEW ProjectsAfter2020 AS 1 分

SELECT P. *, E.EmpName 1 分

FROM Project P

JOIN Employee E ON P.ManagerID = E.EmpID 1 分

WHERE P.StartDate >= '2020-01-01' 1 分

ORDER BY P.StartDate DESC; 1 分

6、SELECT D.Location, COUNT(E.EmpID) AS EmpCount, 2 分

AVG(E.Salary) AS AvgSalary

FROM Employee E

JOIN Department D 1 分

ON E.DeptID = D.DeptID

GROUP BY D.Location; 2 分

7、SELECT EmpID, EmpName 1 分

FROM Employee

WHERE EmpID IN (2 分

SELECT EmpID

FROM EmployeeProject

GROUP BY EmpID 1 分

HAVING COUNT(ProjID) >= 3 2 分

);

8、CREATE TRIGGER IncreaseSalary

```
ON EmployeeProject
AFTER INSERT          --定义触发器并指定触发时机 1 分
AS
BEGIN
    -- 声明变量 1 分
    DECLARE @EmpID INT, @Age INT, @SalaryIncrease INT, @NewSalary INT;

    -- 从临时表中获取员工号，然后连接员工表获取年龄信息 2 分
    SELECT @EmpID = I.EmpID, @Age = E.Age
    FROM INSERTED I
    JOIN Employee E ON I.EmpID = E.EmpID;

    -- 根据年龄判断月薪增加的额度 2 分
    IF @Age < 30
    BEGIN
        SET @SalaryIncrease = 500;
    END
    ELSE
    BEGIN
        SET @SalaryIncrease = 800;
    END
    -- 更新 Employee 表中的月薪 1 分
    UPDATE Employee
    SET Salary = Salary + @SalaryIncrease
    WHERE EmpID = @EmpID;
END
```