2025 年期中考试

Problem 1

1.1 关系数据库

给出关系数据库的定义.

Solution:
关系数据库是一个由若干关系组成的数据库，
其中每个关系是一个由元组构成的二维表格，每个表格有一组命名的属性，
属性的取值来自定义域，并满足特定的数据完整性约束条件.

1.2 超键、候选键、主键和外键

给出超键、候选键、主键和外键的定义.

Solution:

- **超键:** 能够唯一标识关系中每个元组的一个属性集.
- **候选键:** 小到不能再小的超键，即其任意真子集都不能构成关系的超键.
- **主键:** 数据库设计者选定的候选键，用于唯一标识关系中的元组.
- **外键:** 一个关系中的属性集，它引用了另一个关系的主键.

1.3 有损分解和无损分解

给出有损分解和无损分解的定义.

Solution:
假设属性集 R_1 和 R_2 构成 R 的分解 (即 $R = R_1 \cup R_2$)
考虑关系模式 $r(R)$ 的任意实例 r
若投影结果的自然连接是原始关系的真超集 (即 $r \subset \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$)，则 $R = R_1 \cup R_2$ 为有损分解.
若投影结果的自然连接恰好是原始关系 (即 $r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$)，则 $R = R_1 \cup R_2$ 为无损分解.

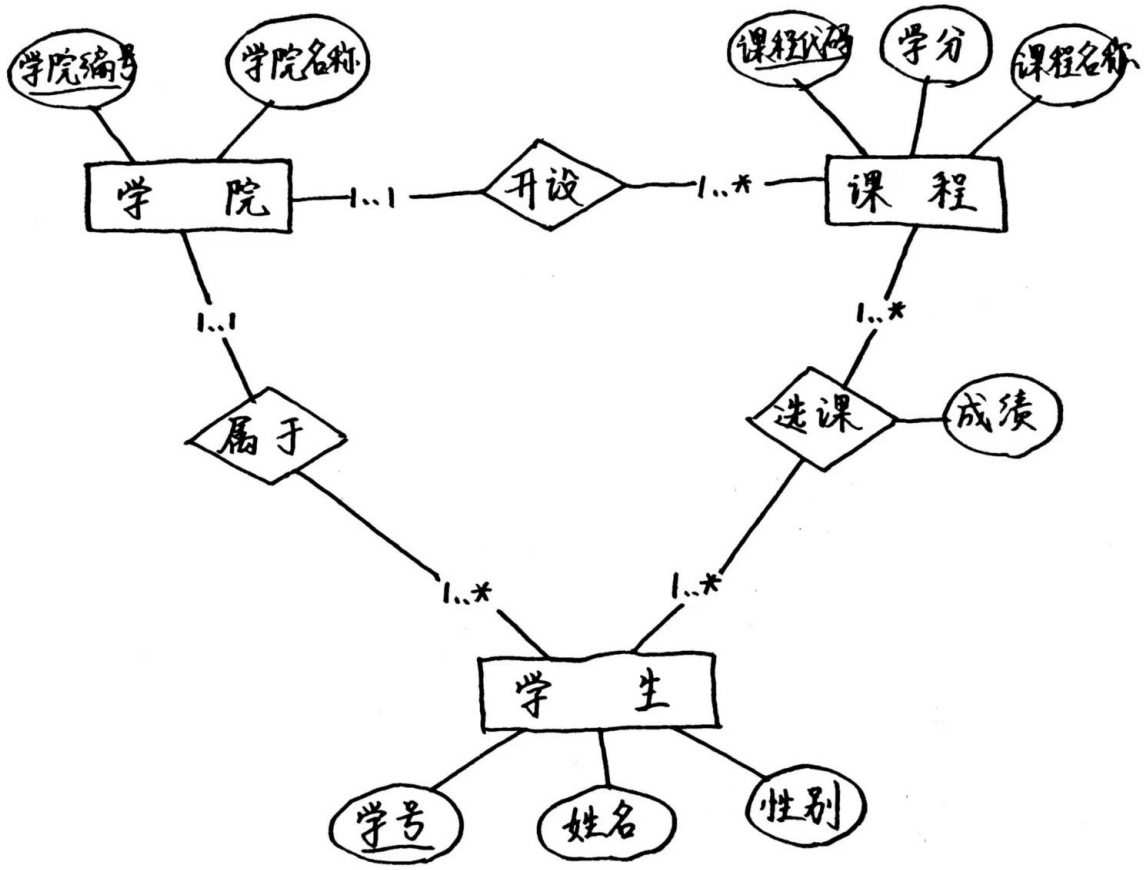
Problem 2

考虑一个大学数据库.
每个学院开设若干个课程，每一个课程属于一个学院 (一对多).
每一个学院有若干个学生，每一个学生只能属于一个学院 (一对多).
一名学生可能选了多门课程，而且每一个课程有若干名学生参与 (多对多).
根据学生在课程中的表现获得分数 (为选课这个关系添加成绩这个属性).

2.1 ER 图

画出 ER 图，并在图上注明属性、联系的类型.

Solution:



2.2 关系模式

将 ER 图转换成关系模式，并注明主键和外键.

Solution:

$$
\begin{aligned}
&\text{学院}(\underline{\text{学院编号}},\ \text{学院名称})\\
&\text{课程}(\underline{\text{课程代码}},\ \underline{\text{学院编号}},\ \text{课程名称},\ \text{学分})\\
&\text{学生}(\underline{\text{学号}},\underline{\text{学院编号}},\ \text{姓名},\ \text{性别})\\
&\text{选课}(\underline{\text{学号}},\underline{\text{课程代码}},\ \text{成绩})
\end{aligned}
$$

Problem 3

考虑以下数据库模式:

$$
\begin{aligned}
&\text{Customer}(\underline{\text{cid}},\ \text{phone},\ \text{email})\\
&\text{Product}(\underline{\text{pid}},\ \text{price})\\
&\text{Order}(\underline{\text{oid}},\ \text{cid}(\text{FK}\rightarrow\text{Customer}),\ \text{pid}(\text{FK}\rightarrow\text{Product}),\ \text{quantity})
\end{aligned}
$$

3.1 关系代数-1

查找购买了编号为 "001" 产品的顾客的编号，电话号码和邮箱地址.

Solution:

$$
\text{result1}=\prod_{\text{cid,phone,email}}(\text{Customer}\bowtie\sigma_{\text{pid}=001}(\text{Order}))
$$

3.2 关系代数-2

查找每份订单购买数量都为 1 的顾客的编号.

Solution:
本题考察差集运算:

$$\text{result2} = \Pi_{\text{cid}}(\text{Order}) - \Pi_{\text{cid}}(\sigma_{\text{quantity}\neq 1}(\text{Order}))$$

不能直接筛选 $\text{quantity} = 1$ 的订单，
因为这相当于查找 "存在订单购买数量为 1 的顾客的编号"，与题意不符.

3.3 关系代数-3

查找编号为 "002" 产品的总销售数量和总销售金额.

Solution:
本题考察聚合操作:

$$\text{result3} = g_{\text{sum(quantity) as total\_quantity, sum(quantity}\times\text{price) as total\_amount}}(\sigma_{\text{pid}=002}(\text{Order} \bowtie \text{Product}))$$

3.4 关系代数-4

查找价格在 $[10, 100)$ 之间的产品的编号和价格.

Solution:

$$\text{result4} = \Pi_{\text{pid,price}}(\sigma_{\text{price}\geq 10 \wedge \text{price} < 100}(\text{Product}))$$

3.5 关系代数-5 $(\star)$

对于每个产品，查找曾购买过这个产品的不同用户的个数.

Solution:
本题考察分组聚合操作:

$$\text{result5} = {}_{\text{pid}}g_{\text{count(cid) as total\_users}}\left(\Pi_{\text{pid,cid}}(\text{Order})\right)$$

3.6 关系代数-6

查找购买编号为 "003" 产品总数量最多的顾客的编号.

Solution:
先筛选 "003" 产品的订单，并按顾客分组计算总购买数量:

$$\text{product\_003\_sum} = {}_{\text{cid}}g_{\text{sum(quantity) as total\_quantity}}(\sigma_{\text{pid}=003}(\text{Order}))$$

再找出最大值对应的顾客编号:

$$\text{result6} = \Pi_{\text{cid}}\left(\sigma_{\text{total\_quantity}=\text{max\_quantity}}\left(\text{product\_003\_sum} \times \left(g_{\text{max(total\_quantity) as max\_quantity}}\text{product\_003\_sum}\right)\right)\right)$$

Problem 4

考虑以下数据库模式:

$$\begin{aligned}
&\text{Customer}(\underline{\text{name}},\ \text{phone},\ \text{city})\\
&\text{Agent}(\underline{\text{branch}},\ \text{city})\\
&\text{Flight}(\underline{\text{no}},\ \text{airline},\ \text{depart},\ \text{arrive})\\
&\text{Ticket}(\underline{\text{no}(\text{FK}\rightarrow\text{Flight}),\ \text{branch}(\text{FK}\rightarrow\text{Agent}),\ \text{name}(\text{FK}\rightarrow\text{Customer})},\ \text{price})
\end{aligned}$$

4.1 SQL 查询-1

列出所有购买机票飞往上海的乘客的姓名和电话号码.

Solution:

```sql
SELECT c.name, c.phone
FROM Customer AS c
    JOIN Ticket AS t ON c.name = t.name
    JOIN Flight AS f ON t.no = f.no
WHERE f.arrive = 'Shanghai';
```

4.2 SQL 查询-2

找出所有通过自己所住的城市的机票代理点购买过机票的乘客的姓名.

Solution:
注意使用 `DISTINCT` 关键字进行去重:

```sql
SELECT DISTINCT c.name
FROM Customer AS c
    JOIN Ticket AS t ON c.name = t.name
    JOIN Agent AS a ON t.branch = a.branch
WHERE c.city = a.city;
```

4.3 SQL 查询-3

列出所有有航班到达却没有航班出发的城市.

Solution:
注意使用 `DISTINCT` 关键字进行去重:

```sql
SELECT DISTINCT arrive AS city
FROM Flight AS f1
WHERE NOT EXISTS(
    SELECT 1
    FROM Flight AS f2
    WHERE f2.depart = f1.arrive
);
```

4.4 SQL 查询-4

请计算每家航空公司从上海飞往北京的所有售出机票的平均价格.

Solution:
考察 `GROUP BY` 分组聚合操作:

```sql
SELECT f.airline, ROUND(AVG(t.price), 2) AS avg_price
FROM Flight AS f
    JOIN Ticket AS t ON f.no = t.no
WHERE f.depart = 'Shanghai'
    AND f.arrive = 'Beijing'
GROUP BY f.airline;
```

4.5 SQL 查询-5

对于每家航空公司，计算有这家航空公司的航班直接往来的城市对的个数 (这里城市对是一个无序对).

Solution:
第一个子查询直接保留自然顺序的航班，第二个子查询处理反向航班，并转化为正常顺序.
然后合并正向和反向筛选的结果，允许重复记录.
最后外层查询按航空公司分组聚合，对城市对去重，统计唯一组合数量.

```sql
SELECT airline, COUNT(DISTINCT city1, city2) AS city_pair_count -- 按航空公司分组聚合
FROM(
    SELECT airline, depart AS city1, arrive AS city2 -- 查询正向城市对 (即自然顺序)
    FROM Flight
    WHERE depart <= arrive
    UNION ALL
    SELECT airline, arrive AS city1, depart AS city2 -- 查询反向城市对
    FROM Flight
    WHERE depart > arrive
) AS all_pairs
GROUP BY airline;
```

Problem 5

5.1 属性闭包

设关系模式 $R=(A,B,C,D,E)$ 上存在函数依赖集 $\mathcal{F}=\{A\to BC, CD\to E, B\to D\}$.
计算 $\mathrm{cl}(B)$ 和 $\mathrm{cl}(BC)$，并列出该关系模式的全部候选键.

Solution:

- ① 计算 B 的属性闭包 $\mathrm{cl}(B)$:
  从 $\mathrm{result}=\{B\}$ 出发，应用函数依赖 $B\to D$ 可置 $\mathrm{result}=\{B,D\}$.
  无其他函数依赖可用，因此 $\mathrm{cl}(B)=\{B,D\}$.
- ② 计算 BC 的属性闭包 $\mathrm{cl}(BC)$:
  从 $\mathrm{result}=\{B,C\}$ 出发，应用函数依赖 $B\to D$ 可置 $\mathrm{result}=\{B,C,D\}$.
  应用函数依赖 $CD\to E$ 可置 $\mathrm{result}=\{B,C,D,E\}$.
  无其他函数依赖可用，因此 $\mathrm{cl}(BC)=\{B,C,D,E\}$.
- ③ 列出该关系模式的全部候选键:
  根据 $\mathrm{cl}(BC)=\{B,C,D,E\}$ 可知任何不包含 A 的属性集都不能构成超键.
  考虑计算单属性 A 的闭包:
  从 $\mathrm{result}=\{A\}$ 出发，应用函数依赖 $A\to BC$ 可置 $\mathrm{result}=\{A,B,C\}$.
  应用函数依赖 $B\to D$ 可置 $\mathrm{result}=\{A,B,C,D\}$.
  应用函数依赖 $CD\to E$ 可置 $\mathrm{result}=\{A,B,C,D,E\}$.
  无其他函数依赖可用，因此 $\mathrm{cl}(A)=\{A,B,C,D,E\}$，
  这意味着 A 是关系模式 R 的超键 (同时也是候选键，因为它已经小到不能再小了)

  注意到关系模式 R 的任意超键 (例如 AB) 一定包含属性 A，
  而 A 本身已是候选键，因此 A 是唯一的候选键.

5.2 无损分解

设关系模式 $R=(A,B,C,D,E)$ 上存在函数依赖集 $\mathcal{F}=\{A\to C, BC\to D, E\to A, CF\to E\}$.
将 R 分解为 $R_1=(A,B,C,D)$ 和 $R_2=(A,B,E,F)$ 是否是无损分解?

Solution:
要判断 R 分解为 R_1 和 R_2 是否为无损分解，
即验证 $R_1\cap R_2$ 是否为 R_1 或 R_2 的超键，
因此只需检查函数依赖 $R_1\cap R_2\to R_1$ 和 $R_1\cap R_2\to R_2$ 是否至少有一个在 R 上成立.

考虑计算 $R_1\cap R_2=\{A,B\}$ 的闭包 $\mathrm{cl}(AB)$:
从 $\mathrm{result}=\{A,B\}$ 出发，应用函数依赖 $A\to C$ 可置 $\mathrm{result}=\{A,B,C\}$.
应用函数依赖 $BC\to D$ 可置 $\mathrm{result}=\{A,B,C,D\}$.
无其他函数依赖可用，因此 $\mathrm{cl}(AB)=\{A,B,C,D\}=R_1$.
这意味着 $R_1\cap R_2$ 是 R_1 的超键，
因此将 R 分解为 $R_1=(A,B,C,D)$ 和 $R_2=(A,B,E,F)$ 是无损分解.

5.3 依赖保持

设关系模式 $R=(A,B,C)$ 上存在函数依赖集 $\mathcal{F}=\{A\to C, C\to B\}$.
将 R 分解为 $R_1=(A,C)$ 和 $R_2=(B,C)$ 是否保持依赖?

Solution:
逐一检验 $\mathcal{F}$ 中的每个函数依赖在 R 分解成 R_1,R_2 后是否被保持:

- ① $\mathrm{cl}(A\cap R_1)\cap R_1=AC$ 包含 C，函数依赖 $A\to C$ 被保持.
- ② $\mathrm{cl}(C\cap R_2)\cap R_2=BC$ 包含 B，函数依赖 $C\to B$ 被保持.

因此上述分解保持依赖.

5.4 Boyce-Codd 范式

设关系模式 $R=(A,B,C)$ 中 A,B 为联合主键.
若 R 在函数依赖集 $\mathcal{F}$ 上符合 BCNF，给出 $\mathcal{F}$ 可以包含的所有函数依赖.

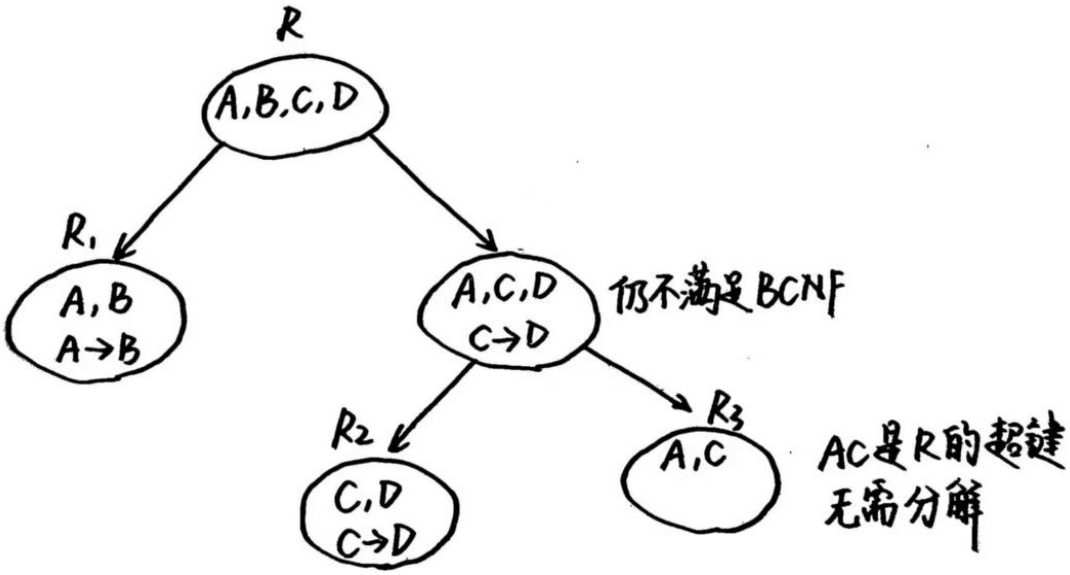
Solution:
BCNF 要求所有非平凡函数依赖 $\alpha\to\beta$ 中，α 必须是 R 的超键.
注意到 A,B 为联合主键，
故非平凡函数依赖只能是 $AB\to C,\ AB\to AC,\ AB\to BC,\ AB\to ABC$.
$\mathcal{F}$ 可以包含上述非平凡函数依赖，以及下列平凡函数依赖:

$$
\begin{gathered}
A\to A,\ B\to B,\ C\to C\\
AB\to AB,\ AB\to A,\ AB\to B\\
AC\to AC,\ AC\to A,\ AC\to C\\
BC\to BC,\ BC\to B,\ BC\to C\\
ABC\to ABC\\
ABC\to AB,\ ABC\to AC,\ ABC\to BC\\
ABC\to A,\ ABC\to B,\ ABC\to C
\end{gathered}
$$

5.5 BCNF 分解算法

设关系模式 $R=(A,B,C,D)$ 上存在函数依赖集 $\mathcal{F}=\{A\to B, C\to D\}$.
判断 R 是否满足 BCNF.
若符合请给出判断依据; 若不符合，请按照 BCNF 分解算法给出无损分解后的关系模式.

Solution:
注意到 A 的闭包 $\mathrm{cl}(A)=\{A,B\}$，因此 A 不是 R 的超键.
这意味着函数依赖 $A\to B$ 的左侧不是 R 的超键，因此 R 的满足 BCNF.
按照 BCNF 分解算法得到的无损分解如下:



$$R = \{A, B, C, D\} \quad \Rightarrow \quad \begin{cases} R_1 = \{A, B\} \\ R_2 = \{C, D\} \\ R_3 = \{A, C\} \end{cases}$$

The End