

# 《数据库概论》第 2 次课后作业

## 一、填空题

1. 关系数据库的规范化理论主要研究**函数依赖**和**多值依赖**及其对关系模式的影响。
2. 函数依赖分为完全函数依赖、部分函数依赖和**传递函数依赖**。
3. 关系模式满足第一范式（1NF）的条件是**每个属性值不可分割**。
4. 第二范式（2NF）要求关系模式中的每个非主属性**完全函数依赖于关键字**。
5. 第三范式（3NF）要求关系模式中不存在非主属性对关键字的**传递函数依赖**。
6. BCNF 范式要求每一个函数依赖的决定因素**必包含关键字**。
7. 多值依赖的符号表示为  $X \rightarrow\!\!\! \rightarrow Y$ 。
8. 第四范式（4NF）要求关系模式中不存在**非平凡多值依赖**。
9. 模式分解的两个关键特性是**无损联接性和依赖保持性**。
10. 数据库设计的四个阶段包括需求分析、概念设计、**逻辑设计**和物理设计。
11. 需求分析阶段的主要工具是 \*\* 数据流图（DFD）\*\* 和数据字典。
12. 概念设计阶段常用的模型是 **E-R 模型**（实体 - 联系模型）。
13. 逻辑设计阶段的任务是将概念模型转换为**关系数据模型**。
14. 物理设计阶段主要考虑数据库的**存储结构**和存取方法。
15. 函数依赖集的闭包用符号  $F^+$  表示。
16. Armstrong 公理系统包括自反规则、增广规则和**传递规则**。
17. 候选关键字是指能够**唯一标识元组且最小**的属性集。
18. 关系模式分解到 3NF 时，需消除**部分依赖**和传递依赖。
19. 数据字典包含数据项、数据结构、数据流、**数据存储**和数据处理。
20. 实体间的联系类型包括一对**一**、一对**多**和**多对多**。
21. 视图集成过程中可能出现的冲突包括命名冲突、概念冲突、域冲突和**约束冲突**。
22. 物理设计中的存取方法包括索引、集簇和 **HASH**。
23. 关系模式 SCG (S#, Sn, Sd, Sa, C#, G) 的关键字是 \*\*(S#, C#)\*\*。
24. 多值依赖产生的原因是关系中存在两个**相互独立的一对多联系**。
25. 数据库设计的生命周期法包括需求分析、概念设计、逻辑设计、物理设计、编码、测试、运行和**进一步修改**。
26. 函数依赖集 F 的最小覆盖需满足三个条件：依赖因素为单个属性、无冗余依赖、**决定因素无冗余属性**。
27. 实体集转换为关系模式时，实体的属性变为关系的属性，实体的标识属性变为关系的**主关键字**。
28. 二元联系的转换方式根据**参与度**和函数对应关系分为不同情况。
29. 物理设计中，集簇的优点是提高特定查询效率，缺点是**建立开销高且可能影响其他查询**。
30. 逻辑设计中，继承关系的转换方式包括每个实体集独立转换、仅转换叶子实体集和**合并为单个关系**。
31. 关系视图的作用包括提供逻辑独立性、适应用户需求和**数据保密**。
32. 数据流图的符号包括数据源 / 终点、数据流、处理过程和**数据存储**。
33. 函数依赖  $X \rightarrow Y$  中，X 称为**决定因素**，Y 称为**依赖因素**。
34. 无损联接性的分解要求分解后的关系通过自然连接能**还原原关系**。
35. 数据库设计的核心任务是根据用户需求设计出性能良好的**数据模式**。

## 问答题

1. 什么是**函数依赖**？请举例说明完全函数依赖和部分函数依赖的区别。

函数依赖是指关系模式中属性间的依存关系，即一个属性集的值确定后，另一个属性集的值也随之确定。

- **完全函数依赖**：若  $X \rightarrow Y$ ，且对 X 的任何真子集  $X'$ ,  $X'$  不能决定 Y，则 Y 完全依赖于 X。例如， $(S#, C#) \rightarrow G$ （学生选课成绩，学号和课程号共同决定成绩，缺一不可）。

- **部分函数依赖**: 若  $X \rightarrow Y$ , 但  $Y$  依赖于  $X$  的某个真子集。例如,  $(S\#, C\#) \rightarrow Sn$  (学生姓名仅依赖于学号, 因此  $Sn$  部分依赖于  $(S\#, C\#)$ )。

## 2. 简述数据库设计的四个阶段及其主要任务。

- **需求分析**: 调查用户需求, 生成数据流图和数据字典。
- **概念设计**: 构建 E-R 模型, 描述实体、属性及联系。
- **逻辑设计**: 将 E-R 模型转换为关系模式, 优化并确定外模式。
- **物理设计**: 设计数据库的存储结构和存取方法, 提高访问效率。

## 3. 什么是模式分解的无损联接性和依赖保持性? 为什么它们很重要?

- **无损联接性**: 分解后的关系通过自然连接能完全还原原关系, 确保数据不丢失。
  - **依赖保持性**: 分解后的关系模式能保留原函数依赖, 确保数据约束不被破坏。
- 重要性: 两者保证分解后的数据库模式既不丢失信息, 又能维持数据完整性和一致性。

## 4. 简述 E-R 模型转换为关系模式的规则 (以二元联系为例)。

- **一对一联系**: 可转换为独立关系, 或与任意一端实体合并。
- **一对多联系**: 联系转换为独立关系, 或与多端实体合并 (加入一端主键作为外键)。
- **多对多联系**: 必须转换为独立关系, 包含两端主键和联系属性, 主键为两端主键的组合。

## 5. 为什么需要对关系模式进行规范化? 规范化的主要步骤是什么?

- **目的**: 消除数据冗余、插入异常、删除异常和更新异常, 提高数据完整性。
- **步骤**:
  - a. 消除非主属性对关键字的部分依赖 (达到 2NF)。
  - b. 消除非主属性对关键字的传递依赖 (达到 3NF)。
  - c. 消除主属性间的部分或传递依赖 (达到 BCNF)。
  - d. 消除非平凡多值依赖 (达到 4NF)。