第二章 关系模型

Employee(EID, Name, DeptID, Salary)`

Department(DeptID, DeptName, Location)`

Project(PID, PName, DeptID)`

WorksOn(EID, PID, Hours)`

1. 查询所有薪资高于 5000 的员工。

2. 查询所有员工的姓名和所在部门编号。

3. 查询员工姓名及其所在部门名称（使用一般连接）。

4. 找出所有员工和所有项目的可能组合（不加任何条件）。

5. 查询每个项目所属的部门名称（使用自然连接）。

6. 找出所有员工和所有项目中出现的部门编号（去重）。

7. 找出没有参与任何项目的员工编号。

8. 将所有薪资高于 8000 的员工关系赋值给一个临时关系 `HighSalaryEmp`。

9. 将 `Employee` 表重命名为 `Emp`，并将其属性重命名为 `(ID, Name, D, Sal)`。

10. 请创建表：

表 Department 包含：

dept\_id：主键

dept\_name：唯一且非空

表 Student 包含：

student\_id：主键

name：非空

email：唯一

age：检查年龄必须在 18 到 60 岁之间

dept\_id：外键，引用 Department(dept\_id)

第三章 基础 SQL / 第四章 中级SQL

1. 写一条查询语句，查询员工表 `employees` 的 `salary` 字段，并将其结果列命名为 `monthly\_salary`。

2. 查询 `customers` 表中，姓名以“Li”开头的客户记录。

3. 写一条查询语句，返回 `products` 表的所有列所有记录。

4. 查询 `orders` 表所有订单，按订单日期 `order\_date` 从最近到最早排序。

5. 查询 `employees` 表中，工资在 3000 到 6000 之间的员工。

6. 查询 `customers` 表中，客户 ID 等于 1001 的客户信息。

7. 写两条查询，分别查询 `employees` 表和 `managers` 表的姓名字段，将两表姓名合并去重显示。

8. 查询 `orders` 表中，所有订单的总金额和订单数量。

9. 查询 `employees` 表，按部门 `dept` 统计每个部门的员工数量。

10. 查询 `employees` 表，统计每个部门的员工数量，只显示员工数量超过 10 的部门。

11.

查询 orders 表中，客户ID在 vip\_customers 表中的订单。

查询 employees 表中，薪资高于部门 dept\_salaries 表中最低薪资的员工。

查询 products 表中，价格高于所有 discounted\_products 表中折扣后价格的产品。

查询所有有订单记录的客户。

查询 employees 表中，部门 dept 是否只有唯一一位员工（即部门唯一）。

12. 用 `WITH` 写一个临时表，先筛选出薪资大于 5000 的员工，再查询这些员工的姓名和薪资。

13.

查询 orders 和 customers 表，自动按相同字段连接，显示订单号和客户姓名。

查询 orders 和 customers 表，按 customer\_id 连接，显示订单号和客户姓名。

查询 orders 和 customers 表，用 USING(customer\_id) 连接，显示订单号和客户姓名。

14. 写一条查询，查询所有客户及其订单信息，即使客户没有订单也要显示（使用左连接）。

15. 创建一个视图 `high\_salary\_employees`，只包含薪资高于 8000 的员工。

16.

查询 employees 表中，入职日期在 2024-01-01 之后的员工。

查询 shifts 表中，班次开始时间在上午 9 点之后的所有班次。

查询 logs 表中，日志时间在 2025-06-01 08:00:00 到 2025-06-01 12:00:00 之间的记录。

17.更新 `employees` 表中，所有属于“销售部”的员工薪资增加 10%。

18.向 `products` 表插入一条新产品记录，包含产品名和价格。

19.删除 `customers` 表中，状态为“inactive”的客户记录。

20.授予用户 `john` 查询 `employees` 表的权限，然后撤销该权限。

第五章 高级SQL

1. 函数

请编写一个 SQL 函数 `calculate\_discount`，输入参数为顾客的总购买金额 `total\_amount`，根据以下规则返回折扣金额：

购买金额大于等于 1000，折扣 10%

购买金额大于等于 500 且小于 1000，折扣 5%

购买金额小于 500，无折扣

2. 过程

请编写一个存储过程 `add\_employee`，参数包括 `emp\_name`（员工姓名）、`emp\_dept`（部门）、`emp\_salary`（薪资）。存储过程执行时，将该员工信息插入到 `employees` 表中，且如果 `emp\_salary` 小于 3000，则将其自动提升到 3000。

3. 触发器

请设计一个触发器，当对 `orders` 表执行 `INSERT` 操作时，自动将该订单的金额 `order\_amount` 累加到对应客户 `customers` 表中 `total\_spent` 字段里，实现客户累计消费金额的自动更新。

4.写出SQL中语言结构：

case

while

repeat

if