

实验四：数据高级查询

一、实验目的

- 熟练掌握设计正确查询语句以实现数据高级查询的方法
- 熟练掌握 mysql 连接查询、子查询和集合查询的方法
 - (内)连接、(全)外连接、左外连接、右外连接
 - 子查询(嵌套查询)
 - 不相关子查询与相关子查询
 - EXISTS/NOT EXISTS 的使用
 - ANY 的使用
 - ALL 的使用
 - 集合运算：UNION 【ALL】，INTERSECT，EXCEPT
- 理解不相关子查询与相关子查询的各自特点
- 熟练掌握基于派生表的查询构造方法

建议：尽量使用不同的查询语句实现同一查询要求。如，所有带 IN 谓词、比较运算符、ANY 或 ALL 谓词的子查询都能用带 EXISTS 谓词的子查询等价替换

二、实验要求

- 设计正确的 SQL 查询语句并测试其是否实现了查询要求
- 完成实验并撰写实验报告（按实验报告模板）
- 实验报告提交截止日期：2025 年 4 月 26 日 00:00（星期六）
- **选项：**自行完成教材上的例题，但无需放入实验报告中

三、实验内容与步骤

• 连接运算

1. 创建两种表：palette_a 和 palette_b，它们有相同的结构

字段	数据类型	说明
id	INT	PRIMARY KEY
color	VARCHAR(100)	NOT NULL

2. 为表 palette_a 添加样例数据：{(1, 'Red'), (2, 'Green'), (3, 'Blue'), (4, 'Purple')}
3. 为表 palette_b 添加样例数据：{(1, 'Green'), (2, 'Red'), (3, 'Cyan'), (4, 'Brown')}
4. 查询两张表中相同颜色的所有信息
5. 查询 palette_a 表中颜色不出现在 palette_b 表中的 id 和颜色（要求使用左外连接实现）
6. 查询 palette_b 表中颜色不出现在 palette_a 表中的 id 和颜色（要求使用右外连接实现）
7. 查询 5 和 6 的并集（注意：mysql8 不支持 full outer join，但可以通过 5 UNION 6 实现）

- 子查询

8. 查询产品表 products 中的 product_id, product_name, list_price 信息, 要求产品定价 list_price 大于其平均定价 list_price (要求使用子查询实现)
9. 查询产品表 products 中最便宜产品的 product_id, product_name, list_price (要求使用子查询实现)
10. 查询没有一个订单的顾客姓名 (要求使用 NOT IN 实现+其他方法 (如果能找到, 鼓励, 下同))
11. 查询产品表 products 中产品的 product_id, product_name, list_price, 要求产品定价 list_price 大于其同类产品 (可由 category_id 表达) 的平均定价 (要求使用相关子查询和基于派生表查询的两种方法实现)
12. 查询有订单 order 的所有顾客 customer 姓名 (查询涉及 customers 表和 orders 表) (要求使用 EXISTS 实现)
13. 执行以下三条与 NULL 相关的语句, 观察各自执行的结果, 能否从中得出某些初步结论? (EXISTS 与 IN 的不同)

```
SELECT * FROM customers WHERE customer_id IN (NULL);  
SELECT NULL FROM customers;  
SELECT * FROM customers WHERE EXISTS (SELECT NULL FROM customers);
```

14. 找出所有没有订单的顾客姓名 (查询涉及 customers 表和 orders 表) (要求使用 NOT EXISTS 实现)
15. 查询产品表 products 中的产品名 product_name 和定价 list_price, 要求其定价高于产品种类 1 中的任何产品定价 (要求使用 ANY 实现)
16. 查询产品表 products 中的产品名 product_name 和定价 list_price, 要求其定价高于产品种类 1 中的所有定价 (要求使用 ALL 实现)

17. 查询产品表 products 中的产品名 product_name 和定价 list_price, 要求其定价低于产品种类的
所有平均定价 (要求使用 ALL 实现)

- **集合查询**

18. 查询 contacts 表和 employees 表中的所有 last_name, 并以 last_name 升序显示 (要求**分别使用**
UNION 和 **UNION ALL** 实现, 统计各自结果的记录数, 比较两者数目是否相同, 为什么?)

19. 查询同时出现在 contacts 表和 employees 表中的所有 last_name (要求使用 **INNERSECT**+其他方法
实现 (如果能找到))

20. 查询在产品表 products 中而不在库存表 inventories 中的产品号 product_id (要求使用 **EXCEPT**+
其他方法实现 (如果能找到))

四、实验思考

1.请分析子查询的特点及归纳总结什么场景下的查询**必须使用子查询**才能实现, 举例说明之。

2.分析总结相关子查询与不相关子查询的区别, 试举例说明使用相关子查询的典型场景, 如何将相关子查询
转换为一般查询实现? (说明: 一般查询是指不一定必须使用子查询的查询)

注: 将问题的解答放到实验报告的实验总结部分