

力扣高频 SQL 50 题阶段总结（六）

LeetCode 高频 SQL 刷题总结（详细思路强化版）

本篇总结 7 道经典基础 SQL 高频题：

| 1667 / 1527 / 196 / 176 / 1484 / 1327 / 1517

这些题覆盖了 SQL 面试最常见的模型：

- 字符串处理
 - 条件匹配
 - 删除重复数据
 - TopN 查询
 - 分组聚合
 - 日期过滤
 - 正则表达式校验
-
-

1667. Fix Names in a Table（修正姓名格式）

题目要求

表 `Users`：

<code>user_id</code>	<code>name</code>
1	aLice
2	bOB

输出时要求：

- 名字首字母必须大写
- 后面的所有字母必须小写

结果:

user_id	name
1	Alice
2	Bob

💡 解题思路 (详细拆解)

这题的本质是:

对字符串的不同部分做不同的大小写处理, 然后拼接回去

Step 1: 拆分字符串

一个名字可以拆成:

- 第一个字符 (首字母)
- 剩下的部分

例如:

代码块

```
1 aLice
2 ↑
3 首字母: a
4
5 剩余部分: Lice
```

Step 2: 首字母大写

SQL 中:

- `LEFT(name, 1)` 取第一个字符
- `UPPER()` 转成大写

代码块

```
1 UPPER(LEFT(name, 1))
```

Step 3: 剩余部分全部小写

SQL 中:

- `SUBSTRING(name, 2)` 从第 2 个字符开始截取
- `LOWER()` 转小写

代码块

```
1 LOWER(SUBSTRING(name, 2))
```

Step 4: 拼接

使用 `CONCAT()` 拼接:

代码块

```
1 CONCAT(  
2     UPPER(LEFT(name, 1)),  
3     LOWER(SUBSTRING(name, 2))  
4 )
```

✅ 最终 SQL

代码块

```
1 SELECT  
2     user_id,  
3     CONCAT(  
4         UPPER(LEFT(name, 1)),  
5         LOWER(SUBSTRING(name, 2))  
6     ) AS name  
7 FROM Users  
8 ORDER BY user_id;
```

⚠️ 易错点

- `SUBSTRING(name, 2)` 不是截取 2 个字符，而是从位置 2 开始一直取到结尾

- 题目要求按 `user_id` 排序
-
-

✓ 1527. Patients With a Condition (筛选糖尿病患者)

题目要求

表 `Patients` :

patient_id	conditions
1	DIAB1
2	SICK DIAB1
3	DIAB100

要求找出:

- `conditions` 中包含 **DIAB1** 的患者

但注意:

- `DIAB100` 不算
-

解题思路 (详细拆解)

这题的难点在于:

DIAB1 必须是一个完整的疾病编码, 而不是前缀匹配

Step 1: 疾病字段是字符串集合

`conditions` 可能是:

- 单个疾病: `DIAB1`
- 多个疾病: `SICK DIAB1 ASTHMA`

疾病之间用空格分隔。

Step 2: 不能直接 `%DIAB1%`

如果写:

代码块

```
1 WHERE conditions LIKE '%DIAB1%'
```

那么：

代码块

```
1 DIAB100
```

也会被误匹配。

Step 3: 正确匹配方式

DIAB1 出现只有两种情况：

情况 1: 开头就是 DIAB1

代码块

```
1 conditions LIKE 'DIAB1%'
```

情况 2: 中间出现，并且前面有空格

代码块

```
1 conditions LIKE '% DIAB1%'
```

最终 SQL

代码块

```
1 SELECT *
2 FROM Patients
3 WHERE conditions LIKE 'DIAB1%'
4 OR conditions LIKE '% DIAB1%';
```

⚠ 易错点

- `%DIAB1%` 会误匹配 `DIAB100`
- 必须考虑 DIAB1 在开头和中间两种情况

✅ 196. Delete Duplicate Emails (删除重复邮箱)

📌 题目要求

表 `Person` :

id	email
1	a@x
2	a@x
3	b@x

要求:

- 删除重复邮箱
- 只保留 id 最小的一条

结果:

id	email
1	a@x
3	b@x

💡 解题思路 (详细拆解)

这题是经典模型:

删除重复数据时, 保留最小 id

Step 1: 重复的判定标准

两行重复的条件:

```
代码块 p1.email = p2.email
```

Step 2: 删除哪一行?

题目要求保留最小 id, 所以删除:

代码块

```
1 p1.id > p2.id
```

即:

- p1 是“重复且 id 更大”的那条

Step 3: 自连接定位删除对象

SQL 删除必须明确删除表别名:

代码块

```
1 DELETE p1
2 FROM Person p1
3 JOIN Person p2
```

✅ 最终 SQL

代码块

```
1 DELETE p1
2 FROM Person p1
3 JOIN Person p2
4 ON p1.email = p2.email
5 AND p1.id > p2.id;
```

⚠️ 易错点

- 删除时必须指定 `DELETE p1`

- 条件必须是 `>` 才能保留最小 id

✓ 176. Second Highest Salary (第二高工资)

📌 题目要求

找第二高工资：

- 如果只有一个工资，返回 null

💡 解题思路 (详细拆解)

这题考察：

| TopN 查询 + NULL 返回机制

Step 1: 第二高 = 去重后排序第 2 个

工资可能重复：

salary
100
100
200

第二高应该是 100，不是 NULL。

所以必须：

代码块

```
1  DISTINCT salary
```

Step 2: 排序

代码块

```
1 ORDER BY salary DESC
```

Step 3: 取第 2 条

MySQL 用:

代码块

```
1 LIMIT 1 OFFSET 1
```

解释:

- OFFSET 1: 跳过第一条 (最高工资)
- LIMIT 1: 取下一条

Step 4: 不存在时自动返回 NULL

外层 SELECT 包裹后:

- 如果没有第二条, 子查询为空 → 返回 NULL

✅ 最终 SQL

代码块

```
1 SELECT (  
2     SELECT DISTINCT salary  
3     FROM Employee  
4     ORDER BY salary DESC  
5     LIMIT 1 OFFSET 1  
6 ) AS SecondHighestSalary;
```

⚠ 易错点

- 必须 DISTINCT
- OFFSET 从 0 开始

✓ 1484. Group Sold Products By The Date (按日期统计产品)

📌 题目要求

每天卖出的：

- 产品数量（去重）
- 产品列表（按字典序拼接）

💡 解题思路（详细拆解）

Step 1: 按日期分组

代码块

```
1 GROUP BY sell_date
```

Step 2: 统计产品种类数

代码块

```
1 COUNT(DISTINCT product)
```

Step 3: 拼接产品名

MySQL 提供：

代码块

```
1 GROUP_CONCAT()
```

并要求：

- DISTINCT 去重

- ORDER BY product 字典序
-

Step 4: 最终组合

✓ SQL

代码块

```
1 SELECT
2     sell_date,
3     COUNT(DISTINCT product) AS num_sold,
4     GROUP_CONCAT(DISTINCT product ORDER BY product) AS products
5 FROM Activities
6 GROUP BY sell_date
7 ORDER BY sell_date;
```

⚠ 易错点

- GROUP_CONCAT 是 MySQL 特有
 - 必须加 ORDER BY 才符合题意
-

✓ 1327. List Products Ordered in a Period (时间段订单统计)

📌 题目要求

找出:

- 2020年2月
 - 总销量 ≥ 100 的产品
-

💡 解题思路 (详细拆解)

Step 1: 先过滤时间

日期过滤必须在 WHERE:

代码块

```
1 WHERE order_date BETWEEN '2020-02-01' AND '2020-02-29'
```

Step 2: 按产品分组

代码块

```
1 GROUP BY product_id
```

Step 3: 统计销量

代码块

```
1 SUM(unit)
```

Step 4: 筛选销量 \geq 100

聚合后的筛选必须用 HAVING:

代码块

```
1 HAVING SUM(unit) >= 100
```

SQL

代码块

```
1 SELECT
2     p.product_name,
3     SUM(o.unit) AS unit
4 FROM Orders o
5 JOIN Products p
```

```
6 ON o.product_id = p.product_id
7 WHERE o.order_date BETWEEN '2020-02-01' AND '2020-02-29'
8 GROUP BY o.product_id
9 HAVING SUM(o.unit) >= 100;
```

⚠ 易错点

- WHERE 过滤行
- HAVING 过滤聚合结果

✅ 1517. Find Users With Valid Emails (合法邮箱)

📌 题目要求

邮箱必须：

- 字母开头
- 只含字母数字 _ .
- 以 [@leetcode.com](https://leetcode.com) 结尾

💡 解题思路 (详细拆解)

这题最适合用正则：

正则结构

代码块

```
1  ^           开头
2  [a-zA-Z]   首字符必须是字母
3  [a-zA-Z0-9_.-]* 后面允许的字符集合
4  @leetcode\.com 固定结尾
5  $           结束
```

✓ SQL

代码块

```
1 SELECT *
2 FROM Users
3 WHERE mail REGEXP '^[a-zA-Z][a-zA-Z0-9_.-]*@leetcode\\.com$';
```

⚠ 易错点

- `.` 必须转义 `\\.`

✓ 本组题核心模板总结

模型	题号
字符串处理	1667
精准 LIKE 匹配	1527
删除重复 (保留最小)	196
TopN 查询	176
分组聚合 + 拼接	1484
日期过滤 + HAVING	1327
正则校验	1517

🚀 下一步建议

你现在已经把 SQL 高频基础模型刷完了

下一阶段就是：

- 窗口函数 (rank / lag / lead)
- TopN per Group (185)
- 滑动窗口 (1321)

只要你回复一句：

继续总结下一组题

我马上接着写。