

175. 组合两个表

表1: Person

列名	类型
PersonId	int
FirstName	varchar
LastName	varchar

PersonId 是上表主键

表2: Address

列名	类型
AddressId	int
PersonId	int
City	varchar
State	varchar

AddressId 是上表主键

编写一个 SQL 查询，满足条件：无论 person 是否有地址信息，都需要基于上述两表提供 person 的以下信息：

FirstName, LastName, City, State

答案

```
SELECT
    FirstName,
    LastName,
    City,
    State
FROM Person
LEFT JOIN Address
    ON Person.PersonId = Address.PersonId;
```

176. 第二高的薪水

编写一个 SQL 查询，获取 Employee 表中第二高的薪水（Salary）。

Id	Salary
1	100
2	200
3	300

例如上述 `Employee` 表，SQL 查询应该返回 `200` 作为第二高的薪水。如果不存在第二高的薪水，那么查询应返回 `null`。

SecondHighestSalary
200

答案

```
SELECT CASE WHEN count(Salary) > 1
  THEN (
    SELECT DISTINCT Salary FROM Employee
    ORDER BY Salary DESC
    LIMIT 1, 1)
  ELSE NULL END AS SecondHighestSalary
FROM Employee;
```

181. 超过经理收入的员工

`Employee` 表包含所有员工，他们的经理也属于员工。每个员工都有一个 `Id`，此外还有一列对应员工的经理的 `Id`。

Id	Name	Salary	ManagerId
1	Joe	70000	3
2	Henry	80000	4
3	Sam	60000	NULL
4	Max	90000	NULL

给定 `Employee` 表，编写一个 SQL 查询，该查询可以获取收入超过他们经理的员工的姓名。在上面的表格中，`Joe` 是唯一一个收入超过他的经理的员工。

Employee
Joe

答案

```
SELECT a.Name AS `Employee`
FROM Employee AS a,
      Employee AS b
WHERE a.ManagerId = b.Id
      AND a.Salary > b.Salary;
```

182. 查找重复的电子邮箱

编写一个 SQL 查询，查找 Person 表中所有重复的电子邮箱。

示例：

Id	Email
1	a@b.com
2	c@d.com
3	a@b.com

根据以上输入，你的查询应返回以下结果：

Email
a@b.com

说明：所有电子邮箱都是小写字母。

答案

```
SELECT Email FROM Person
GROUP BY Email
HAVING COUNT(Email) > 1;
```

183. 从不订购的客户

某网站包含两个表，Customers 表和 Orders 表。编写一个 SQL 查询，找出所有从不订购任何东西的客户。

Customers 表:

Id	Name
1	Joe
2	Henry
3	Sam
4	Max

Orders 表:

Id	CustomerId
1	3
2	1

例如给定上述表格，你的查询应返回:

Customers
Henry
Max

答案

```
SELECT a.Name AS `Customers` FROM `Customers` a
LEFT JOIN `Orders` b ON a.id = b.CustomerId
WHERE b.id IS NULL;
```

196. 删除重复的电子邮箱

编写一个 SQL 查询，来删除 Person 表中所有重复的电子邮箱，重复的邮箱里只保留 Id 最小的那个。

```
+-----+
| Id | Email |
+-----+
| 1 | john@example.com |
| 2 | bob@example.com |
| 3 | john@example.com |
+-----+
Id 是这个表的主键。
```

例如，在运行你的查询语句之后，上面的 `Person` 表应返回以下几行：

```
+-----+
| Id | Email |
+-----+
| 1 | john@example.com |
| 2 | bob@example.com |
+-----+
```

答案

```
DELETE FROM `Person`
WHERE
  `Id`
  IN (
    SELECT `Id`
    FROM (
      SELECT p2.Id Id
      FROM
        Person p1, Person p2
      WHERE
        p1.Id < p2.Id
        AND p1.Email = p2.Email
    ) tmp
  );
```

197. 上升的温度

给定一个 `Weather` 表，编写一个 SQL 查询，来查找与之前（昨天的）日期相比温度更高的所有日期的 `Id`。

Id(INT)	RecordDate DATE	Temperature(INT)
1	2015-01-01	10
2	2015-01-02	25
3	2015-01-03	20
4	2015-01-04	30

例如，根据上述给定的 Weather 表格，返回如下 Id:

Id
2
4

答案

```
SELECT a.Id AS `Id` FROM `Weather` a LEFT JOIN `Weather` w
      ON datediff(a.RecordDate, w.RecordDate) = 1
WHERE a.Temperature > w.Temperature;
```

595. 大的国家

这里有张 World 表

name	continent	area	population	gdp
Afghanistan	Asia	652230	25500100	20343000
Albania	Europe	28748	2831741	12960000
Algeria	Africa	2381741	37100000	188681000
Andorra	Europe	468	78115	3712000
Angola	Africa	1246700	20609294	100990000

如果一个国家的面积超过 300 万平方公里，或者人口超过2500万，那么这个国家就是大国家。

编写一个 SQL 查询，输出表中所有大国家的名称、人口和面积。

例如，根据上表，我们应该输出：

name	population	area
Afghanistan	25500100	652230
Algeria	37100000	2381741

答案

```
SELECT
    `name`,
    `population`,
    `area`
FROM `World`
WHERE
    `population` > 25000000 OR area > 3000000;
```

596. 超过5名学生的课

有一个courses 表，有: **student** (学生) 和 **class** (课程)。

请列出所有超过或等于5名学生的课。

例如,表：

student	class
A	Math
B	English
C	Math
D	Biology
E	Math
F	Computer
G	Math
H	Math
I	Math

应该输出：

```
+-----+
| class |
+-----+
| Math  |
+-----+
```

Note: 学生在每个课中不应被重复计算。

答案

```
SELECT a.class FROM courses a
GROUP BY class
HAVING count(DISTINCT (student)) >= 5;
```

620. 有趣的电影

某城市开了一家新的电影院，吸引了很多人过来看电影。该电影院特别注意用户体验，专门有个 LED 显示板做电影推荐，上面公布着影评和相关电影描述。

作为该电影院的信息部主管，您需要编写一个 SQL 查询，找出所有影片描述为**非 boring** (不无聊) 的并且 **id 为奇数** 的影片，结果请按等级 **rating** 排列。

例如，下表 cinema:

id	movie	description	rating
1	War	great 3D	8.9
2	Science fiction		8.5
3	irish	boring	6.2
4	Ice song	Fantasy	8.6
5	House card	Interesting	9.1

对于上面的例子，则正确的输出是为：

5	House card	Interesting	9.1
1	War	great 3D	8.9

答案


```
SELECT * FROM `Cinema`  
WHERE `description` != 'boring' AND id % 2 = 1  
ORDER BY `rating`;
```

627. 交换工资

给定一个 salary 表，如下所示，有 m=男性 和 f=女性的值。交换所有的 f 和 m 值(例如，将所有 f 值更改为 m，反之亦然)。要求使用一个更新查询，并且没有中间临时表。

例如：

id	name	sex	salary
1	A	m	2500
2	B	f	1500
3	C	m	5500
4	D	f	500

运行你所编写的查询语句之后，将会得到以下表：

id	name	sex	salary
1	A	f	2500
2	B	m	1500
3	C	f	5500
4	D	m	500

答案

```
UPDATE `Salary`  
SET sex = if(sex = 'm', 'f', 'm');
```

177. 第N高的薪水

编写一个 SQL 查询，获取 Employee 表中第 n 高的薪水 (Salary)。

Id	Salary
1	100
2	200
3	300

例如上述 `Employee` 表， $n = 2$ 时，应返回第二高的薪水 `200`。如果不存在第 n 高的薪水，那么查询应返回 `null`。

<code>getNthHighestSalary(2)</code>
<code>200</code>

答案

```
CREATE FUNCTION getNthHighestSalary(N INT)
  RETURNS INT
  BEGIN
    DECLARE M INT;
    SET M = N - 1;
    RETURN (
      SELECT DISTINCT `Salary`
      FROM Employee
      ORDER BY `Salary` DESC
      LIMIT M, 1
    );
  END;
```

178. 分数排名

编写一个 SQL 查询来实现分数排名。如果两个分数相同，则两个分数排名 (Rank) 相同。请注意，平分后的下一个名次应该是下一个连续的整数值。换句话说，名次之间不应该有“间隔”。

Id	Score
1	3.50
2	3.65
3	4.00
4	3.85
5	4.00
6	3.65

例如，根据上述给定的 `Scores` 表，你的查询应该返回（按分数从高到低排列）：

Score	Rank
4.00	1
4.00	1
3.85	2
3.65	3
3.65	3
3.50	4

答案

```
SELECT
    `Score`,
    CONVERT(
        CASE
            WHEN @preValue = `Score`
            THEN @preRank
            WHEN (@preValue := `Score`) >= 0
            THEN @preRank := @preRank + 1
            END, UNSIGNED) AS `Rank`
FROM `Scores`,
(
    SELECT @preValue := NULL) p,
(
    SELECT @preRank := 0) v
ORDER BY `Score` DESC;
```

180. 连续出现的数字

编写一个 SQL 查询，查找所有至少连续出现三次的数字。

Id	Num
1	1
2	1
3	1
4	2
5	1
6	2
7	2

例如，给定上面的 Logs 表，1 是唯一连续出现至少三次的数字。

ConsecutiveNums
1

答案

```
SELECT DISTINCT l1.Num `ConsecutiveNums` FROM `Logs` l1
LEFT JOIN Logs l2 ON l1.Id = l2.Id - 1
LEFT JOIN Logs l3 ON l1.Id = l3.Id - 2
WHERE l1.Num = l2.Num AND l2.Num = l3.Num;
```

626. 换座位

小美是一所中学的信息科技老师，她有一张 `seat` 座位表，平时用来储存学生名字和与他们相对应的座位 `id`。

其中纵列的 `id` 是连续递增的

小美想改变相邻俩学生的座位。

你能不能帮她写一个 SQL query 来输出小美想要的结果呢？

示例：

id	student
1	Abbot
2	Doris
3	Emerson
4	Green
5	Jeames

假如数据输入的是上表，则输出结果如下：

id	student
1	Doris
2	Abbot
3	Green
4	Emerson
5	Jeames

注意：

如果学生人数是奇数，则不需要改变最后一个同学的座位。

答案

```
SELECT
    if(
        id < (
            SELECT count(*)
            FROM `Seat`),
        if(id % 2 = 0, id - 1, id + 1),
        if(id % 2 = 0, id - 1, id)
    ) AS id, student
FROM
    `Seat`
ORDER BY id ASC;
```

185. 部门工资前三高的员工

Employee 表包含所有员工信息，每个员工有其对应的 Id, salary 和 department Id。

Id	Name	Salary	DepartmentId
1	Joe	70000	1
2	Henry	80000	2
3	Sam	60000	2
4	Max	90000	1
5	Janet	69000	1
6	Randy	85000	1

Department 表包含公司所有部门的信息。

Id	Name
1	IT
2	Sales

编写一个 SQL 查询，找出每个部门工资前三高的员工。例如，根据上述给定的表格，查询结果应返回：

Department	Employee	Salary
IT	Max	90000
IT	Randy	85000
IT	Joe	70000
Sales	Henry	80000
Sales	Sam	60000

答案

```

SELECT
    Department,
    Employee,
    Salary
FROM (
    SELECT
        Department,
        Employee,
        Salary,
        CASE
            WHEN @dpId = DepartmentId AND @sal > Salary
            THEN @rk := @rk + 1
            WHEN @dpId = DepartmentId
            THEN @rk
            WHEN @dpId := DepartmentId
            THEN @rk := 1
        END AS rk,
        @sal := Salary AS sal
    FROM
        (
            SELECT
                d.name AS Department,
                e.name AS Employee,
                Salary,
                e.DepartmentId
            FROM
                Employee e,
                Department d
            WHERE e.DepartmentId = d.Id
            ORDER BY d.id ASC,
                e.Salary DESC) od,
        (
            SELECT
                @rk := 1,
                @dpId := NULL,
                @sal := NULL) b) de
WHERE de.rk <= 3;

```

262. 行程和用户

Trips 表中存所有出租车的行程信息。每段行程有唯一键 Id, Client_Id 和 Driver_Id 是 Users 表中 Users_Id 的外键。Status 是枚举类型, 枚举成员为 ('completed', 'cancelled_by_driver', 'cancelled_by_client')。

Id	Client_Id	Driver_Id	City_Id	Status	Request_at
1	1	10	1	completed	2013-10-01
2	2	11	1	cancelled_by_driver	2013-10-01
3	3	12	6	completed	2013-10-01
4	4	13	6	cancelled_by_client	2013-10-01
5	1	10	1	completed	2013-10-02
6	2	11	6	completed	2013-10-02
7	3	12	6	completed	2013-10-02
8	2	12	12	completed	2013-10-03
9	3	10	12	completed	2013-10-03
10	4	13	12	cancelled_by_driver	2013-10-03

Users 表存所有用户。每个用户有唯一键 Users_Id。Banned 表示这个用户是否被禁止, Role 则是一个表示 ('client', 'driver', 'partner') 的枚举类型。

Users_Id	Banned	Role
1	No	client
2	Yes	client
3	No	client
4	No	client
10	No	driver
11	No	driver
12	No	driver
13	No	driver

写一段 SQL 语句查出 **2013年10月1日** 至 **2013年10月3日** 期间非禁止用户的取消率。基于上表，你的 SQL 语句应返回如下结果，取消率（Cancellation Rate）保留两位小数。

Day	Cancellation Rate
2013-10-01	0.33
2013-10-02	0.00
2013-10-03	0.50

答案

```

SELECT
    Request_at Day,
    round(sum(CASE WHEN Status IN ('cancelled_by_driver',
    'cancelled_by_client')
        THEN 1
        ELSE 0 END) / count(1), 2) "Cancellation Rate"
FROM
    (
        SELECT Users_Id FROM Users
        WHERE Banned = 'No') a
JOIN
    (
        SELECT
            Client_Id,
            Status,
            Request_at FROM Trips
        WHERE Request_at >= '2013-10-01' AND Request_at <= '2013-10-
03') b
    ON a.Users_Id = b.Client_Id
GROUP BY Request_at
ORDER BY Request_at

```


601. 体育馆的人流量

X 市建了一个新的体育馆，每日人流量信息被记录在这三列信息中：序号 (id)、日期 (date)、人流量 (people)。

请编写一个查询语句，找出高峰期时段，要求连续三天及以上，并且每天人流量均不少于100。

例如，表 stadium:

id	date	people
1	2017-01-01	10
2	2017-01-02	109
3	2017-01-03	150
4	2017-01-04	99
5	2017-01-05	145
6	2017-01-06	1455
7	2017-01-07	199
8	2017-01-08	188

对于上面的示例数据，输出为：

id	date	people
5	2017-01-05	145
6	2017-01-06	1455
7	2017-01-07	199
8	2017-01-08	188

Note: 每天只有一行记录，日期随着 id 的增加而增加。

答案

```
SELECT
  a.id,
  a.date,
  a.people FROM stadium AS a,
  (
    SELECT
      MAX(p.id) AS mid,
      MAX(p.c) AS mc
    FROM
```

```

(
    SELECT
        l.id,
        l.date,
        l.people,
        CASE
            WHEN l.people >= 100
                THEN @cur := @cur + 1
            ELSE @cur := 0
        END AS c,
        CASE
            WHEN @cur > 0
                THEN @index := @index
            ELSE @index := @index + 1
        END AS d
    FROM stadium AS l,
    (
        SELECT
            @cur := 0,
            @index := 0) t) p
    GROUP BY p.d
    HAVING count(p.c) > 3) t
WHERE a.id <= t.mid
      AND a.id > t.mid - t.mc

```