**## Quick View**

- SELECT Clause

- WHERE Clause

- Logical Operators

- IN Operators

- BETWEEN Operators

- LIKE Operators

- REGEXP Operator

- IS NULL Operator

- ORDER BY Clause

- LIMIT Clause

- Inner Joins

- Outer Joins

- USING Clause

- Cross Joins

- Unions

- Inserting Data

- --

- UPDATE

- LEFT

- SUBSTRING

- UPPER/LOWER

- CONCAT

- GROUP\_CONCAT

**## Notes**

1. 带null的无法与值做比较，需要先把null转为0，或者多使用个 IS NULL 判断条件

```

SELECT name

FROM customer

WHERE referee\_id != '2' OR referee\_id IS NULL

```

```

SELECT name

FROM customer

WHERE IFNULL(referee\_id,0)<>2

```

2. 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。有三种思路，NOT EXISTS, NOT IN, 还有JOIN\

另：SELECT 1在作为子查询中的判断子查询结果是否存在条件时效率较高（大量数据情况下）, 因为不用查字典表

```

SELECT Name AS Customers

FROM Customers c

WHERE NOT EXISTS (

    SELECT 1

    FROM Orders o

    WHERE o.CustomerId=c.Id

    )

```

```

SELECT Name AS Customers

FROM Customers

WHERE Customers.Id NOT IN (

    SELECT CustomerID

    FROM Orders

)

```

```

SELECT Name AS Customers

FROM Customers c

LEFT JOIN Orders o ON c.Id=o.CustomerId

WHERE o.Id IS NULL

```

3. UPDATE语句，更新；IF语句，判断

```

UPDATE salary SET sex = if(sex='m','f','m')

```

4. CONCAT, UPPER, LOWER, LEFT, SUBSTR 用法

LEFT(string, number\_of\_chars)

SUBSTR(string, start, length)

```

SELECT user\_id, concat(upper(left(name,1)),lower(substr(name,2))) as name

FROM Users

ORDER BY user\_id

```

5. GROUP\_CONCAT 用法

GROUP\_CONCAT(DISTINCT XXX ORDER BY XXX SEPARATOR "X")

```

SELECT

    sell\_date,

    COUNT(DISTINCT product) AS num\_sold,

    GROUP\_CONCAT(DISTINCT product ORDER BY product SEPARATOR ',') AS products

FROM Activities

GROUP BY sell\_date

ORDER BY sell\_date

```

6. LIKE 用法

LIKE 'a%'/'%a'/'%or%'/'a\_%\_%'

\* 'a\_%\_%' : start with a with length at least 3

```

SELECT patient\_id, patient\_name, conditions

FROM Patients

WHERE conditions LIKE '% DIAB1%' or conditions LIKE 'DIAB1%'

```

7. ORDER BY

```

SELECT \* FROM Customers

ORDER BY Country ASC, CustomerName DESC;

```

8. Find missing information

```

SELECT employee\_id FROM Employees WHERE employee\_id NOT IN(SELECT employee\_id FROM Salaries)

UNION

SELECT  employee\_id FROM Salaries  WHERE employee\_id NOT IN(SELECT employee\_id FROM Employees)

ORDER BY employee\_id;

```

9. Transfer between Columns and Rows

```

SELECT

  product\_id,

  SUM(IF(store = 'store1', price, NULL)) 'store1',

  SUM(IF(store = 'store2', price, NULL)) 'store2',

  SUM(IF(store = 'store3', price, NULL)) 'store3'

FROM

  Products1

GROUP BY product\_id ;

```

```

SELECT product\_id, 'store1' AS store, store1 AS price FROM products WHERE store1 IS NOT NULL

UNION

SELECT product\_id, 'store2' AS store, store2 AS price FROM products WHERE store2 IS NOT NULL

UNION

SELECT product\_id, 'store3' AS store, store3 AS price FROM products WHERE store3 IS NOT NULL;

```