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;
```