Everything about primary key

- Primary key can only contain the unique values
- No null values are permitted
- Only one primary key is permitted per table
- Recommended the integer data type for primary key
- Recommended to have the auto-increment on primary key
- Internally the index named “Primary” is created on the column.

Aggregate Functions

Aggregate Function	Descriptions
count()	It returns the number of rows, including rows with NULL values in a group.
sum()	It returns the total summed values (Non-NULL) in a set.
average()	It returns the average value of an expression.
min()	It returns the minimum (lowest) value in a set.
max()	It returns the maximum (highest) value in a set.
groutp_concat()	It returns a concatenated string.

Subquery

A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators.

- The inner query executed first gives the result to the outer query, and then the main/outer query will be performed.
- Must be closed within parenthesis
- We cannot use the **ORDER BY** clause in a subquery, although it can be used inside the main query.

```
mysql> select * from employee1;
+--------+----------+---------+------------+---------+
| emp_id | emp_name | emp_age | city       | income  |
+--------+----------+---------+------------+---------+
|    101 | Peter    |      32 | Newyork    |  200000 |
|    102 | Mark     |      32 | California |  300000 |
|    103 | Donald   |      40 | Arizona    | 1000000 |
|    104 | Obama    |      35 | Florida    | 5000000 |
|    105 | Linkon   |      32 | Georgia    |   25000 |
|    106 | Kane     |      45 | Alaska     |  450000 |
+--------+----------+---------+------------+---------+
```

```
mysql> select emp_name, city, income from employee1
    -> where emp_id IN(select emp_id FROM employee1);
+----------+------------+---------+
| emp_name | city       | income  |
+----------+------------+---------+
| Peter    | Newyork    |  200000 |
| Mark     | California |  300000 |
| Donald   | Arizona    | 1000000 |
| Obama    | Florida    | 5000000 |
| Linkon   | Georgia    |   25000 |
| Kane     | Alaska     |  450000 |
+----------+------------+---------+
```

```
mysql> select * from employee1
    -> where emp_id IN(select emp_id from
    -> employee1 where income>350000);
+--------+----------+---------+---------+---------+
| emp_id | emp_name | emp_age | city    | income  |
+--------+----------+---------+---------+---------+
|    103 | Donald   |      40 | Arizona | 1000000 |
|    104 | Obama    |      35 | Florida | 5000000 |
|    106 | Kane     |      45 | Alaska  |  450000 |
+--------+----------+---------+---------+---------+
```

```
mysql> select * from employee1
    -> where
    -> income=(select max(income) from employee1);
+--------+----------+---------+---------+---------+
| emp_id | emp_name | emp_age | city    | income  |
+--------+----------+---------+---------+---------+
|    104 | Obama    |      35 | Florida | 5000000 |
+--------+----------+---------+---------+---------+
1 row in set (0.01 sec)
```

All about index

- Performance matters
- Be wise on designing the index

When to avoid in

1. Indexes should not be used on tables containing few records.
2. Tables that have frequent, large batch updates or insert operations.
3. Indexes should not be used on columns that contain a high number of **NULL** values.
4. Indexes should not be used on the columns that are frequently manipulated.

Altering the table

- Tasks
  - Dropping the column

```
ALTER TABLE Customers DROP COLUMN Email;
```

  - Adding the column

```
ALTER TABLE Persons ADD DateOfBirth date;
```

  - Modifying the column

```
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

Views

A view consists of a stored query accessible as a virtual table composed of the result set of a query.

- CREATE VIEW CheapCars AS SELECT Name FROM Cars WHERE Cost<25000;
- ALTER VIEW CheapCars AS SELECT Name FROM Cars WHERE Cost<30000;
- Modifying the view
- Droping the view
- Show full tables;
- Check table <table name>
- You can also create view from view

Restrictions on Views

- **Can’t create an index of views**

- **MySQL invalidates the view**

- **MySQL views cannot be updateable in some situations –** Actually, the simple view can be updateable but a view created on a complex SELECT statement with JOIN or SUBQUERY etc. cannot be updateable.

- **MySQL does not support materialized views**

- **Using subquery in the FROM clause of view depends on MySQL version –** Actually, we can use a subquery in the FROM clause of view if MySQL version is less than 5.7.7.

- **Cannot create a TEMPORARY view –** Actually, the definition cannot refer to a TEMPORARY table hence we cannot create a TEMPORARY view.

- **Cannot associate a trigger with a view –** we cannot associate a trigger with a view.

Temporary table

```
CREATE TEMPORARY TABLE temp_table_name
SELECT * FROM original_table
LIMIT 0;
```