St. Aloysius S.S.School, Cantt., Jabalpur

Class 12 (IP)

Unit 1 :

More on MySql Dated – 30/03/2020

Welcome children –

In class XI , you had studied the concepts of MySql (creating and handling data in tables). So why not take a quick review of the past. Hope you have not forgotten all the previous knowledge. At the end of the learning there will be questions for you to examine yourself.

No need to purchase any copy separately for these tasks. You can note down the important points (underlined for you) in any blank pages and also learn.

Revision

Commands are –

- **Use – to connect to a database**
  Example – use Jackson
  This will connect you to a database Jackson.

- **Drop – to remove a table or a database**
  Example – drop table stores;
  Example - drop database Jackson;
  The first will remove the table stores (the table should be empty).
  The second will remove the database Jackson.
- **Delete – to erase a single record (row or tuple) or many records.**
  Example – delete from stores;
  This will erase all the records from the table stores.
  Example – delete from stores where quantity > 400 ;
  This will erase all the records from the table stores for value more than 400 in column quantity.
- **Create – to create a new database or a new table in a database.**
  Example – create database Jackson;
  This will create a new database named Jackson.
  Example – create table stores (icode int not null , iname char(20), quantity int, price int);
  This will create a new table named stores and also has 4 columns (attributes) namely icode, iname, quantity and price. The data type int stands for integer data type and char(20) specifies the size limited to 20 characters storage.

- **Insert – to add a new record in the table already created.**

  Example – insert into stores values (500, 'crack jack', 250, 20);

  This will add a new record in the table stores in the same sequence as specified which is according to the sequence used at the creation of table stores.

  insert into stores values (520, 'safola', 100, 190);

  insert into stores values (600, 'glucose', 200, 20);

- **Select – to view the records from the table as rows and columns.**

  Example – select * from stores;

  This will display the attributes and also the first record inserted. If there were 10 records in all then all the records will be displayed one below the other.

Table : stores

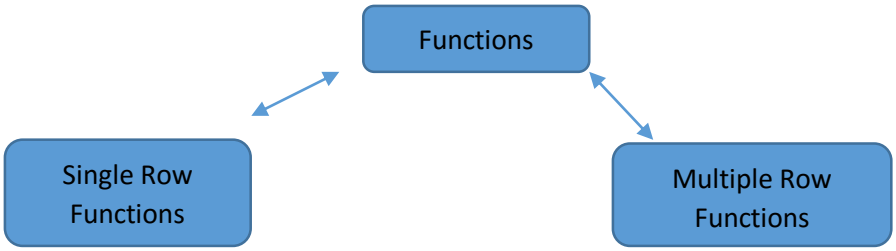
Icode	iname	quantity	price
500	crack jack	250	20
520	safola	100	190
600	glucose	200	20

More to learn

Functions in MySql – A function can be define aa a set of pre-defined commands which, when called, performs certain operations and returns a single value.

Functions in MySql are classified into two types :

- **Single Row functions**
- **Multiple Row functions**



- **Single Row functions :**

Single row functions operate on a single value to return a single value to return a single value. They can accept one or more arguments but return only one result per row. When applied to a table they return a single result for every row of the queried table. They can be used with Select, where and order by clause. Example concat( ),

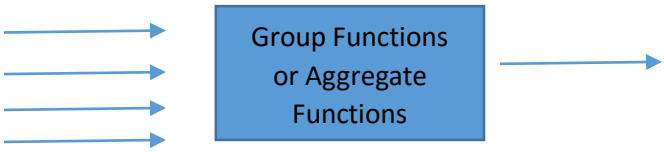
Left ( ), right( ), trim( ) , ltrim( )

They are further categorized into :

1. String functions
2. Numeric functions
3. Date and Time functions

- **Multiple Row Functions :** These functions are termed as Aggregate Functions because they summarize the results of a query, and return a single value calculated from values in a column instead of providing the listing of all of the rows.

Aggregate functions are also termed as Group Functions and are so called because they operate on group or aggregate of tuples (records).



MySql has a large collection of in-built functions also called library functions used for performing calculations on data. These functions summarize the results of a query and returns a single result or value.

It helps in summarizing large volumes of data. These functions give a single output for an entire group of rows or entire rows in the table.

Functions are –

* AVG – This function calculates the average of a set of values.
* MAX – This function gets the maximum value in a set of values.
* MIN – This function gets the minimum value in a set of values.
* Count - This function gets count of rows in a specified table.
* Sum - This function calculates the sum of values of a single attribute.

You need to know that –

- Sum ( ) and Avg( ) functions always takes argument of integer type only.
- Average of string and date type data is not defined. NULL values are not included in the calculations.

Examples :

* Select avg(price) from stores;

* Select sum(quantity) from stores;

* Select max(price) from stores;

* Select min(icode) from stores;

* Select count(price) from stores;
  OR
  Select count(*) from stores;

Important – count( ) and be written with (*) or the (column name).

- **Distinct** - The keywords distinct and count can be used together to count the number of records without taking duplicate values.

  Syntax : select count (distinct column_name) from <table_name>;

  Example : To display the total number of items in the stores table without including the duplicate items or ignoring the duplicate values.

  Select count(distinct iname) from stores;

  In the table taken as reference from the beginning there are no items which are repeated and therefore the result of the next statement will be same as above.

  Select count(iname) from stores;

  Can you guess the answer ??????????????

Arranging records in an ordered form :

The ordered by clause is used to sort data in ascending or descending order based on one or more columns. The order by keyword is used to sort the result by one or more fields in a table.

There are two ways in which you can arrange your records at the time you view the records using select.

- Ascending order
- Descending order

In both the cases you need the order by clause.
There are two more reserve words used to assist in doing this job.

Asc and Desc

Example 1 :

To display the itemcode, itemname , quantity and price from the table stores on the basis of the quantity in ascending order –

Select icode, iname, quantity, price from stores order by quantity;

OR

Select icode, iname, quantity, price from stores order by quantity asc ;

Noticeable is that in both the cases the result will be the same i.e. the result will be displayed on the basis of quantity but in ascending order. Therefor the reserve word asc added with the column name quantity yields the same result. We can say that not using it also gives the same result. So it is optional when you want the records to be displayed in ascending order.

Example 2 :

To display the itemcode, itemname , quantity and price from the table stores on the basis of the quantity in descending order –

Select icode, iname, quantity, price from stores order by quantity desc;

Noticeable is that in the above case the result will be displayed on the basis of quantity but in descending order. Therefor the reserve word desc added with the column name quantity means descending . So it is not optional when you want the records to be arranged in descending order you have to use it.

- **Sorting on column Alias –**

  Alias means another name given to the column name which can be used instead of the actual column name –

  Example –

  - Select icode, iname , quantity AS quantity_in_store from stores order by quantity_in_store asc ;

    This has an alias name quantity_in_store instead of quantity and is applied with order by clause.

    What is so new in this topic ?

  For your reading –

  [Aliases" redirects here. For other uses, see Alias (disambiguation).

  A **pseudonym** () or **alias** () is a name that a person or group assumes for a particular purpose, which can differ from their first or true name (orthonym). The term is not used when a new name entirely replaces an individual's own.]

- MySQL `GROUP BY` clause

  The `GROUP BY` clause groups a set of rows into a set of summary rows by values of columns or expressions. The `GROUP BY` clause returns one row for each group. In other words, it reduces the number of rows in the result set.

  You often use the `GROUP BY` clause with aggregate functions such as SUM, AVG, MAX, MIN, and COUNT. The aggregate function that appears in the `SELECT` clause provides information about each group.

  The `GROUP BY` clause is an optional clause of the `SELECT` statement. The following shows the `GROUP BY` clause syntax:

  - SELECT c1, c2,..., cn, aggregate_function(ci) FROM table WHERE where_conditions GROUP BY c1 , c2,...,cn ;

The `GROUP BY` clause must appear after the `FROM` and `WHERE` clauses.

Following the `GROUP BY` keywords is a list of comma-separated columns or expressions that you want to use as criteria to group rows.

MySQL evaluates the `GROUP BY` clause after
the `FROM`, `WHERE` and `SELECT` clauses and before the `HAVING` , `ORDER BY` clauses:

```
SELECT   price , COUNT(*) FROM stores  GROUP BY price ;

SELECT  iname,  SUM(quantity * price) AS total FROM stores GROUP BY  iname ;

SELECT icode    SUM(quantity * price) AS total FROM stores GROUP BY  icode ;
```

- Restricting query results using the HAVING clause

Note this – where clause is never used with group by clause but instead

Having clause is used.

It's not always that we will want to perform groupings on all the data in a given table. There will be times when we will want to restrict our results to a certain given criteria. In such cases , we can use the HAVING clause

Suppose we want to know all the release years for movie category id is 7. We would use the following script to achieve our results.

```
SELECT * FROM stores  GROUP BY iname HAVING length(iname) = 7;
```

Note only movies with length of iname = 7 will be affected by our GROUP BY clause.

Here HAVING IS USED AND NOT WHERE

Summary

- The GROUP BY Clause is used to group rows with same values .

- The GROUP BY Clause is used together with the SQL SELECT statement.

- The SELECT statement used in the GROUP BY clause can only be used contain column names, aggregate functions, constants and expressions.
- The HAVING clause is used to restrict the results returned by the GROUP BY clause.

Where Vs Having -

To be explained by YOU……

……

Most important fact about aggregation functions is that these functions do not take NULL into consideration at all.

You need to think now whether the following two statements mean the same –

Select sum(price) from stores where quantity = 450;

And

Select sum(price) from stores group by quantity;

Points to revise –

- Type of Mysql functions
- Aggregation functions
- Compare the aggregation functions with single row functions
- Arranging records in a table
- Arranging records on the basis if common column values
- Use of having clause
- Where Vs having clause

Also form at least 10 different statements of your own using where and having clause which is not discussed here.

You can use a new table and new set of records of your choice and work.

…… End ……