Birla Institute of Technology & Science, Pilani, K. K. BIRLA Goa campus
Database Systems and Applications (IS F243)
Second Semester 2012-2013
Lab-2

To study DQL commands.

Retrieving information from a table

The SELECT statement is used to pull information from a table.

Syntax:

SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy
what_to_select can be a list of columns, or * to indicate all columns.

Selecting all data

mysql> SELECT * FROM <tablename>;

Example://to display all information of all students

mysql >SELECT * FROM students;

Selecting distinct rows

mysql>SELECT DISTINCT <column name> FROM<tablename>;

Example://to display names of all students ignoring duplicate names

mysql > SELECT DISTINCT name from student;

Selecting particular rows i.e; select with where clause

mysql>select column name1, column name2 ….. from tablename where condition (and or conditions);

Examples:

/*to display name of a particular student*/
mysql >SELECT name FROM students where id = 'PS99305018';

/*to display id and name of students in hostel no 10 having percentage more than 90.6*/
mysql >SELECT id,name FROM students where percentage>90.6 **and** hostel=10;

/*to display id and name of students having percentage more than 90.6 but do not belong to hostel no 10*/
mysql >SELECT id,name FROM students where percentage>90.6 and hostel **!**= 10;

OR

mysql >SELECT id,name FROM students where percentage>90.6 and **not** hostel = 10;

/*to display id and name of students having percentage between 60 **and** 90.6*/
mysql >SELECT id,name FROM students where percentage>=60.0 **and** percentage<=90.6;

/*to display id and name of students having percentage 60 **and** 90.6*/
mysql >SELECT id,name FROM students where percentage=60.0 **or** percentage=90.6;

AND and OR may be intermixed. If you do that, it's a good idea to use parentheses to indicate how conditions should be grouped.

```
/*to view current_date*/
mysql> select curdate( );
```

```
/*to view current date and time*/
mysql> select now( ) ; OR show now( ); OR show sysdate( );
```

TRY THIS /*to do arithmetic in select clause*/

```
mysql>select percentage * 10.0 from students where hostel = 10;
```

```
/* to name the result */
mysql>select percentage * 10.0  product from students where hostel = 10;
```

This is just to view the result of arithmetic operation and not to make any changes in the table. To do so we need another DML command named update.

Using inbuilt functions

```
/* display the names of those students whose birthday is in month of june. */
mysql> select name from students where month(bdate)=6;
```

similarly there are inbuilt functions for arithmetic, string, numeric and date operations.

Selecting particular columns

select column name ….from table name

Example:

```
mysql >SELECT  id FROM students;
mysql >SELECT  id, name FROM students;
```

SORTING ROWS

You may have noticed in the preceding examples that the result rows are displayed in no particular order. However, it's often easier to examine query output when the rows are sorted in some meaningful way. To sort a result, use an ORDER BY clause.

Example://to display names of all students sorted by percentage in ascending order

```
mysql> SELECT name, percentage FROM students ORDER BY percentage;
```

To sort in reverse order, add the DESC (descending) keyword to the name of the column you are sorting by:

```
mysql> SELECT name, percentage FROM students ORDER BY percentage DESC;
```

You can sort on multiple columns. For example, to sort by hostel no of students, then by name, use the following query:

```
mysql> SELECT name, hostelno, percentage FROM students ORDER BY hostelno, name
DESC;
```

SQL pattern matching

SQL pattern matching allows you to use ‘_’ (underscore) to match any single character, and ‘%’ (percentage) to match an arbitrary number of characters (including zero characters). In MySQL, SQL patterns are case insensitive by default. Some examples are shown below. Note that you do not use = or != when you use SQL patterns; use the LIKE or NOT LIKE comparison operators instead.

Examples:

To find names beginning with `b':

```
mysql> SELECT * FROM students WHERE name LIKE "b%";
```

To find names ending with `fy':

```
mysql> SELECT * FROM students WHERE name LIKE "%fy";
```

To find names containing a `w':

```
mysql> SELECT * FROM students WHERE name LIKE "%w%";
```

To find names containing exactly five characters, use the `_' pattern character:

```
mysql> SELECT * FROM students WHERE name LIKE "_____";
```

SQL AGGREGATE FUNCTIONS:

SQL provides specialized functions to perform operations using the data manipulation commands. A function takes one or more arguments and returns a value.

1. Average function **avg( ):** This function returns the average of values of the column specified in the argument.
   Example://to find average percentage of students
   ```
   mysql> SELECT AVG(percentage) FROM students;
   ```

2. Min function **min( ):** This function gives the least of all values of the column present in the argument.
   Example://to find minimum percentage of students
   ```
   mysql> SELECT MIN(percentage) FROM students;
   ```

3. Max function **max( )**: This function gives the maximum of a set of values of the column present in the argument.
   Example://to find maximum percentage of students
   ```
   mysql> SELECT MAX(percentage) FROM students;
   ```

4. Sum function **sum( ):** This function is used to obtain the sum of a range of values of a record set.
   Example://to find sum of percentage of all students
   ```
   mysql> SELECT SUM(percentage) FROM students;
   ```

5. Count function **count( )**: This function is used to count the number of non-NULL results.
   Example://to count total number of students
   ```
   mysql> SELECT COUNT(*) FROM students;
   ```