**My SQL Commands**

this command display all database present in your account

filter the data using where clause

1. Relational operator: >,>=, <, <=, =,!=

Select \* from tableName where columnName RO value;

select \* from employee where salary > 15000;

select \* from employee where job\_id='ST\_clerk';

select \* from employee where hire\_date > '1995-01-01';

1. Between operator : range value

select \* from tableName where columnName between minValue and maxValue

select \* from employee where salary between 10000 and 15000;

select \* from employee where hire\_date between '1995-01-01' and '1997-12-31';

1. In operator : using in operator we can pass more the value

Select \* from tablename where columnName in (v1,v2,v3);

select \* from employee where employee\_id in(100,120,106);

select \* from employee where job\_id in('AD\_PRES','ST\_MAN')

1. Like operator : like operator check equality with extra features as start with, end with and contains specific character.

select \* from tableName where columnName like ‘s%’

Start with s character and % means 0 or 1 or many

select \* from tableName where columnName like ‘%a’

name end with a character

select \* from tableName where columnName like ‘%g%’

contains g character

select \* from tableName where columnName like ‘s%’

select \* from employee where first\_name like '%a';

select \* from employee where first\_name like '%g%';

1. Is null : display those records value or field value is missing not present.

select \* from employee where manager\_id is null;

select \* from employee where manager\_id is not null;

**Order by clause** : which help do display the in ascending or descending order.

Select \* from tableName order by columName asc/desc

By default ascending order consider.

select \* from employee order by employee\_id desc

select \* from employee order by first\_name desc;

select \* from employee order by first\_name;

or

select \* from employee order by first\_name asc;

**Join**

Using join we can retrieve more than one table value with or without conditions.

Select fewColumnsFromFirstTable, fewColumnNameFromSecondTable FirsttableName,SecondTableName;

Cross join or Cartesian product

First table m record \* second table n records

5 records : employee

3 records : technologies

select first\_name,salary,department\_name from employee,department;

select first\_name,salary,department\_id,department\_name from employee,department;

retrieve more than one column from more than one table with tableName.columnName

select employee.first\_name,employee.salary,department.department\_id,department.department\_name from employee,department;

table alias

select e.first\_name,e.salary,d.department\_id,d.department\_name from employee e,department d;

**Inner Join**

**It retrieve common records present in both the tables.**

**select emp.first\_name,emp.salary,dept.department\_name from employee emp inner join department dept on emp.department\_id=dept.department\_id;**

**select emp.first\_name,job.job\_title from employee emp inner join jobs job on emp.job\_id=job.job\_id;**

**Left outer join : common as well as left table remaining records**

**select emp.first\_name,emp.salary,dept.department\_name from employee emp left outer join department dept on emp.department\_id=dept.department\_id;**

**Right outer join : common as well as right table remining records**

**select emp.first\_name,emp.salary,dept.department\_name from employee emp right outer join department dept on emp.department\_id=dept.department\_id;**

**rules of inner join**

**number of tables is n then minimum condition must be n-1.**

**Inner join with 3 tables**

**select emp.first\_name,dept.department\_name,job.job\_title from department dept inner join employee emp inner join jobs job on emp.department\_id=dept.department\_id and emp.job\_id=job.job\_id;**

**select emp.first\_name,dept.department\_name,job.job\_title from department dept inner join employee emp inner join jobs job on emp.department\_id=dept.department\_id and emp.job\_id=job.job\_id and job.job\_title='Stock Manager';**

**MySQL functions**

Set contains set of instruction to perform a specific task.

Mainly divided into 2 types.

1. Pre defined or built in function
2. User defined 🡪 Pl SQL

Function and number of parameter and return type.

Pre defined function it divided into 2 types.

1. Single row function: the function functionality apply for each records individually.
   1. String function
   2. Number function
   3. Date function
   4. Conversion function
2. Multi row or aggregate function: the function functionality apply for more than one records base upon group by default group is whole table consider.

Syntax to user pre defined single row function

Select functionName as Columname

String function

select length('Welcome to my sql training') as Result;

select upper('Welcome to my sql training') as Result;

select substr('Welcome to my sql training',4) as Result;

select substr('Welcome to my sql training',4,10) as Result;

**date function**

curdate()

curtime()

now()

select date\_format(now(),'%d-%m-%Y');

select upper(first\_name),date\_format(hire\_date,'%d-%m-%Y') from employee;

select first\_name,hire\_date,floor(datediff(now(),hire\_date)/365) year\_of\_exp from employee;

**Multi row functions**

Sum()

Max()

Min()

Count()

Avg()

select sum(salary) as total\_salary from employee;

select max(salary) as max\_salary from employee;

select min(salary) as min\_salary from employee;

select avg(salary) as avg\_salary from employee;

select count(first\_name) as number\_of\_emp from employee;

count function ignore null value.

select count(employee\_id) as number\_of\_emp from employee;

select count(\*) as number\_of\_emp from employee;

**aggregate function with sub group**

select sum(salary) as total\_salary from employee group by department\_id;

select department\_id,sum(salary) as total\_salary from employee group by department\_id;

group by with where clause

select department\_id,sum(salary) as total\_salary from employee where department\_id is not null group by department\_id;

having clause: where clause and having clause we use for apply condition. Where clause apply the condition for individual records and having clause apply the condition for group of records. Where clause must be before group by. But having after group by.

select department\_id,sum(salary) as total\_salary from employee where department\_id is not null group by department\_id having sum(salary) > 55000;

order by clause

select department\_id,sum(salary) as total\_salary from employee where department\_id is not null group by department\_id having sum(salary) > 55000 order by department\_id desc;