**Practical - 4**

**MySQL Alter table & Select Query**

MySQL ALTER TABLE To Change Table Structure

MySQL ALTER TABLE statement that changes existing table structure such as adding or removing columns, changing column attributes.

Syntax : ALTER TABLE table\_name action1[,action2,…]

Alter table is used to

* Change existing schema
* Add/Remove column
* Add / Remove Constraint
* ALTER TABLE FOR CHANGING COLUMN  
  Syntax : ALTER TABLE table\_name   
   CHANGE COLUMN col\_name datatype [constraint]

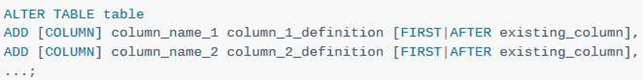
e.g

mysql> **alter table per\_tab1 change name varchar(20) DEFAULT 'xxx' ;**

* ALTER TABLE FOR ADD COLUMN

It will create a new column.

Syntax :



Add a new column called salary in per\_tab1 at the end

E.g mysql> **alter table per\_tab1 add column salary decimal(10,2) ;**

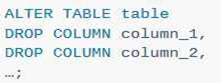
mysql> describe per\_tab1;

mysql > **alter table per\_tab1**

**add column state varchar(10) after city ;**

it will add the new column start after the column city.

* ALTER TABLE FOR DELETE COLUMN



Delete two columns state and salary form per\_tab1;

**e.g mysql> alter table per\_tab1  
 DROP COLUMN state ,  
 DROP COLUMN salary ;**

* **ALTER TABLE FOR ADD CONSTRAINT**

Syntax : ALTER TABLE table\_name

ADD CONSTRAINT constraint\_name

Constraint\_defenation;

ADD Primary Key

ALTER TABLE table\_name

ADD CONSTRAINT pk\_const

PRIMARY KEY(col\_name) ;

e.g

Add the column id at first position in per\_tab1

1. **mysql> alter table per\_tab1**

**add column id int(3) first ;**

2. Add primary key constraint to field id.

**mysql> alter table per\_tab1**

**add constraint pk\_const**

**primary key(id) ;**

* **ALTER TABLE for FOREIGN KEY**

Syntax : ALTER TABLE table\_name

ADD Constraint const\_name

Foreign key (col\_name)

References ref\_table\_name(ref\_col\_name)

**mysql>create table category(cat\_id int(4) primary key, name varchar(50) );**

**mysql>insert into category values(1,'science'),(2,'spritual'),(3,'business'),(4,'food'),(5,'health');**

**mysql> create table post (id int(4) , title varchar(50), content TEXT , createdon date);**

Table category which have c**at\_id as primary key.**

Table post contains information about post. Add the constraint of foreignkey in post table

**mysql>alter table post**

**add constraint fk\_const**

**foreign key(id)**

**references category(cat\_id) ;**

**describe post;**

Don’t allowed to drop column as id contain the foreign key constraint.

**Mysql> alter table post**

**drop column id;**

**ERROR 1828 (HY000): Cannot drop column 'id': needed in a foreign key constraint 'fk\_const'**

Column id is foreign key so it don’t allowed to delete

* **DROP the CONSTRAINT**
  + It allowed to drop existing constraint.

Syntax : ALTER TABLE table\_name

DROP Constraint <constraint\_name>

e.g **alter table post drop constraint fk\_const**;

* DROP TABLE

It deletes the table with its all content.

Syntax : DROP TABLE tablename.

**SELECT STATEMENT**

* SELECT Query

The SELECT statement allows you to get the data from tables or views. A table consists of rows and columns.

Syntax :

SELECT   
 Column1,column2,…  
 FROM table\_name  
 [INNER|LEFT|RIGT JOIN table 2 ON condition]  
 [ Where  
 conditions ]

[ GROUP BY column1 ]

[ HAVING groupby condition ]

[ ORDER BY column1 ]

[ LIMIT offset, length; ]

* Column1 , column2.. : are the columns whose values to be displayed. \* indicates all columns of the table
* Table\_name : Name of the table
* JOIN : gets related data from other tables[table2,table3..] based on specific join
* Conditions
* WHERE clause filters row in the result set.
* GROUP BY clause groups a set of rows into groups and applies aggregate functions on each group.
* HAVING clause filters group based on groups defined by GROUP BY clause.
* ORDER BY clause specifies a list of columns for sorting.
* LIMIT constrains the number of returned rows.

1. Select all the record from table.

e.g mysql> select \* from employee;

2. select field from table

Eg mysql > select name,age,designation from employee;

* DISTINCT to eliminate duplicates

In order to remove these duplicate rows, you use the DISTINCT clause in the SELECT statement.

Syntax : SELECT **DISTINCT**  
 Column FROM tablename

[Where condition]

e.g select department available to employee table

**mysql> select distinct department from employee;**

e.g select distinct department and city of employee table.

**mysql> select distinct department,city from employee;**

it will check combinely unique values for department and city together and display the result

+------------+--------+

| department | city |

+------------+--------+

| sales | NULL |

| admin | NULL |

| production | baroda |

| production | pune |

| sales | surat |

| admin | surat |

| sales | pune |

+------------+--------+

* DISTINCT for NULL values

**mysql> select distinct city from employee;**

+--------+

| city |

+--------+

| NULL | -> NULL repeated only one time even multiple records with city NULL values

| baroda |

| pune |

| surat |

+--------+

* Display records in sorted order

Syntax : SELECT column

FROM tablename

ORDER BY col\_name [desc];

Col\_name : name of column according to which the data will be displayed in sorted order

Desc : for sort into descending order

e.g display the name and age of the employee according to the age.

**mysql> select name, age from employee order by age;**

+--------------+------+

| name | age |

+--------------+------+

| Hares | 24 |

| sunit | 24 |

| suresh | 25 |

| Rohan patel | 26 |

| reema | 26 |

| virat | 32 |

e.g **mysql> select name,age from employee order by age desc;**

virat | 32 |

| sameer | 32 |

| Rohan patel | 26 |

| reema | 26 |

| suresh | 25 |

| Hares | 24 |

| sunit | 24 |

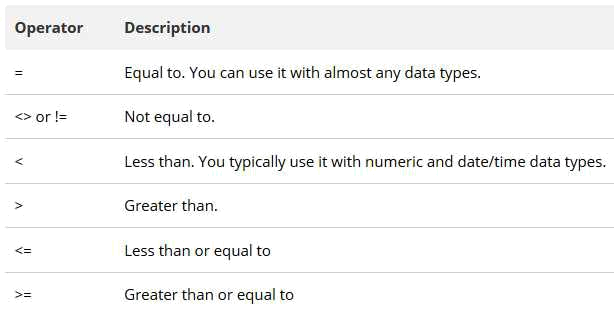
+--------------+------+

* **SELECT WITH WHERE CLAUSE**

Where clause is used to filter the records

Syntax : SELECT columns FROM tablename WHERE conditions;

**Operators used in conditions**

****

**1.** display record which having department sales

**mysql> select \*from employee where department='sales';**

+-------------+------+-------+-------------+------------+----------+------------+

| name | age | city | designation | department | salary | joindate |

+-------------+------+-------+-------------+------------+----------+------------+

| Rohan patel | 26 | NULL | salesman | sales | 9000.00 | NULL |

| Hares | 24 | NULL | salesman | sales | 11000.00 | NULL |

| reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29 |

| suhana | 34 | pune | manager | sales | 30000.00 | 2014-08-23 |

+-------------+------+-------+-------------+------------+----------+------------+

**2** . select the name and city of employee who are not from baroda

**mysql> select name,age,city from employee where city <> 'baroda';**

| name | age | city |

+--------------+------+-------+

| Vimal khatri | 38 | pune |

| reema | 26 | surat |

| ruhan | 33 | surat |

| suhana | 34 | pune |

+--------------+------+-------+

3 . display the employee details whose age is greater than 30

**mysql> select \* from employee where age >=30**

+--------------+------+--------+-------------+------------+----------+------------+

| name | age | city | designation | department | salary | joindate |

+--------------+------+--------+-------------+------------+----------+------------+

| virat | 32 | NULL | admin | admin | 10000.00 | NULL |

| sameer | 32 | NULL | accountant | admin | 12000.00 | NULL |

| Hema trivedi | 36 | baroda | cleark | production | 18000.00 | 1997-03-14 |

| Vimal khatri | 38 | pune | manager | production | 25000.00 | 2001-07-23 |

| ruhan | 33 | surat | cleark | admin | 22000.00 | 2008-05-22 |

| suhana | 34 | pune | manager | sales | 30000.00 | 2014-08-23 |

+--------------+------+--------+-------------+------------+----------+------------+

* **WHERE CLAUSE with AND OPERATOR**

The AND operator is a logical operator that combines two or more Boolean expressions and returns true only if both expressions evaluate to true.

it is normally used when two conditions on different fields are checking at the same time.

1. select employee who are manager and work in production department

**mysql> select \* from employee where designation='manager' and department='production';**

output:

| Vimal khatri | 38 | pune | manager | production | 25000.00 | 2001-07-23 |

2. display employee those are working in sales department and age is less than or equal to 30

**mysql> select \*from employee where department='sales' and age<=30;**

Rohan patel | 26 | NULL | salesman | sales | 9000.00 | NULL |

| Hares | 24 | NULL | salesman | sales | 11000.00 | NULL |

| reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29|

* **WHERE CLAUSE with OR operator**

The MySQL OR operator combines two Boolean expressions and returns true when either condition is true**.**



**Display employee who are living in surat,baroda or nadiad**

**mysql> select \*from employee where city='surat' or city='pune' or city='nadiad' ;**

Vimal khatri | 38 | pune | manager | production | 25000.00 | 2001-07-23 |

| reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29 |

| ruhan | 33 | surat | cleark | admin | 22000.00 | 2008-05-22 |

| suhana | 34 | pune | manager | sales | 30000.00 | 2014-08-23

**Display employee who are living in surat or having salary greater than 25000**

**mysql> select \*from employee where city='surat' or salary>25000 ;**

|reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29 |

| ruhan | 33 | surat | cleark | admin | 22000.00 | 2008-05-22 |

| suhana | 34 | pune | manager | sales | 30000.00 | 2014-08-23 |

* **WHERE CLAUSE with IN operator**

The IN operator allows you to determine if a specified value matches any value in a set of values or returned by a subquery.

Syntax : SELECT column1,column2..

FROM tablename

WHERE

(exp|column1) IN ( value1,value2,…);

**Display employee who are living in nadiad,Mumbai,surat, or pune**

**Mysql>select \*from employee where city in('nadiad','mumbai','surat','pune');**

|Vimal khatri | 38 | pune | manager | production | 25000.00 | 2001-07-23 |

| reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29 |

| ruhan | 33 | surat | cleark | admin | 22000.00 | 2008-05-22 |

| suhana | 34 | pune | manager | sales | 30000.00 | 2014-08-23|

**Display employee who are NOT living in nadiad,Mumbai,surat, or pune**

**Mysql>select \*from employee where city NOT in('nadiad','mumbai','surat','pune');**

**WHERE CLAUSE with BETWEEN operator**

The BETWEEN operator allows you to specify a range to test. You often use the BETWEEN operator in the WHERE clause of the SELECT, UPDATE, and DELETE statements.

Syntax SELECT column1,column1,…

FROM table name

WHERE

Expr|column [NOT] BETWEEN begin\_expr AND end\_expr;

**Display the employee details who join from 2006 to 2012.**

**mysql>select \* from employee where joindate between '2006-01-01' and '2012-12-30';**

|reema | 26 | surat | worker | sales | 8800.00 | 2007-08-29 |

| ruhan | 33 | surat | cleark | admin | 22000.00 | 2008-05-22 |

| suresh | 25 | baroda | peon | production | 5000.00 | 2012-02-14|

* **WHERE CLAUSE with LIKE operator**

The LIKE operator is commonly used to select data based on patterns.

Using the LIKE operator in the right way is essential to increase the query performance.

MySQL provides two wildcard characters for using with the LIKE operator, the percentage % and underscore \_ .

* + The percentage ( % ) wildcard allows you to match any string of zero or more characters.
  + The underscore ( \_ ) wildcard allows you to match any single character.

**Display the employee name and age whose name start with h**

**mysql>select name,age from employee where name like 'h%';**

**Display the employee name and age whose name ends with h**

**mysql> select name,age from employee where name like '%h';**

+--------+------+

| name | age |

+--------+------+

| suresh | 25 |

+--------+------+

Display the name , age of employee whose name start with s and 3rd character is r

**mysql> select name,age from employee where name like 's\_r%';**

suresh | 25

**EXERCISE**

1. ADD column emp\_id int(3) in employee table at first position.
2. ADD constraint primary key for emp\_id in employee table.
3. Drop constraint of primary key from emp\_id in employee table.
4. Delete column emp\_id from employee table.

**FOR EMPLOYEE TABLE**

**NOTE : insert necessary records in employee table which satisfy query results[atleast 10 records]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **datatype** | **size** | **Null** | **Attr ibute** |
| **Name** | **varchar** | **50** | **Not null** |  |
| **Age** | **Int** | **03** |  |  |
| **City** | **Varchar** | **30** |  |  |
| **Designation** | **varchar** | **30** |  |  |
| **Department** | **varchar** | **25** | **Not Null** |  |
| **Salary** | **decimal** | **10,2** |  |  |
| **Joindate** | **Date** |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **age** | **city** | **designation** | **department** | **salary** | **joindate** |
| Rohan patel | 26 | NULL | salesman | sales | 9000 | NULL |
| virat | 32 | NULL | admin | admin | 10000 | NULL |
| sameer | 32 | NULL | accountant | admin | 12000 | NULL |
| Hares | 24 | NULL | salesman | sales | 11000 | NULL |
| Hema trivedi | 36 | baroda | cleark | production | 18000 | 1997-03-14 |
| Vimal khatri | 38 | pune | manager | production | 25000 | 2001-07-23 |
| sunit | 24 | baroda | worker | production | 9000 | 2005-06-19 |
| reema | 26 | surat | worker | sales | 8800 | 2007-08-29 |
| ruhan | 33 | surat | cleark | admin | 22000 | 2008-05-22 |
| suresh | 25 | baroda | peon | production | 5000 | 2012-02-14 |
| suhana | 34 | pune | manager | sales | 30000 | 2014-08-23 |

1. Display all details of employee details who are working in sales department
2. Display all details of employee details who are not working in sales department.
3. Display all details of employee whose age is greater than 30
4. Display name, department, designation of employee whose salary is less than 10000
5. Display all employee who are not a clerk or salesman.
6. Display name, age , salary of employee whose joining date after 1-1-2011.
7. Display all details of employee whose age is between 25 to 35.
8. Display all details of employee working in production or sales department.
9. Display name, age, salary, joindate of employee whose salary is between 15000 to 28000
10. Display all details of employee whose living in surat and working in production department.
11. Display all details of employee those are manager in sales department
12. Display name,age,joindate of employee who joins from 1-1-2007 to 31-12-2014.
13. Display names of employee which are strat from s and ends form s.
14. Display name of employee whose name does not start with H
15. Display name of employee who is cleark in admin department or salesman in sales department.
16. Display all the department available in employee table.[each department name must be displayed only onces]
17. Display all worker lives in Surat city.
18. Display all employee who joins before 12-10-2007
19. Display all employee salary wise and name wise.[first salary,second name]
20. Display all employee whose lives in pune with designation manager or lives in Mumbai with designation salesman.

**Queries based on practical-3 tables**

1. display all product details from product\_master where quentyonhand grater than 100 and salesprice is graterthan 300.
2. display all order detail which is placed before orderdate 30-may-2004 from sales\_order table.
3. display all orders detail which are fulfilled from sales\_order table.
4. display all ordernumber which have Qtyordered and Otydisp is same from sales\_order\_detail.
5. Display order details of all order which are full and not fulfilled from sale\_order table.
6. Display salesman detail who are not from maharastra.