**DDL: Example:**

2. **Alter:**

Alter command is used for alteration of table structures. There are various uses of alter command, such as,

1. to add a column to existing table
2. to rename any existing column
3. to change datatype of any column or to modify its size.
4. alter is also used to drop a column.
5. **to add a column to existing table:**

Using alter command we can add a column to an existing table.

**Syntax**:

alter table table-name add(column-name datatype);

**Example:**

1. alter table customers add DESIG varchar2(20);
2. alter table customers add (hobby varchar2(20));

**SQL> select \* from customers;**

**ID NAME AGE ADDRESS SALARY DESIG HOBBY**

**---------- -------------------- ---------- ------------------------- ---------- -------------------- --------------- 1 Ramesh 32 Ahmedabad 2000**

**2 Khilan 25 Delhi 1500**

**3 kaushik 23 Kota 2000**

**4 Chaitali 25 Mumbai 6500**

**5 Hardik 27 Bhopal 8500**

**6 Komal 22 MP 4500**

**6 rows selected.**

* 1. **To Add Multiple Column to existing Table**

Using alter command we can even add multiple columns to an existing table.

**Syntax:**

alter table table-name add(column-name1 datatype1, column-name2 datatype2, column-name3 datatype3);

**Example:**

SQL> alter table customers add (

F\_name varchar2(10),

M\_name varchar2(10),

Pan varchar2(10));

select \* from customers;

**ID NAME AGE ADDRESS SALARY DESIG HOBBY F\_NAME M\_NAME PAN**

**-- ------------ ----------------- ------- -------- ---------- ---------- ----------**

**1 Ramesh 32 Ahmedabad 2000**

**2 Khilan 25 Delhi 1500**

**3 kaushik 23 Kota 2000**

**4 Chaitali 25 Mumbai 6500**

**5 Hardik 27 Bhopal 8500**

**6 Komal 22 MP 4500**

**1.2 To Add column with Default Value**

alter command can add a new column to an existing table with default values.

**Syntax:**

alter table table-name add(column-name1 datatype1 default data);

**Example:**

alter table Student add(dob date default '1-Jan-99');

**ID NAME AGE ADDRESS SALARY DESIG HOBBY F\_NAME M\_NAME PAN DOB**

**-- ------------ ----------------- ------- -------- ---------- ---------- --- ----------**

**1 Ramesh 32 Ahmedabad 2000 01-01-2000**

**2 Khilan 25 Delhi 1500 01-01-2000**

**3 kaushik 23 Kota 2000 01-01-2000**

**4 Chaitali 25 Mumbai 6500 01-01-2000**

**5 Hardik 27 Bhopal 8500 01-01-2000**

**6 Komal 22 MP 4500**  **01-01-2000**

1. **To Rename a column**

Using alter command you can rename an existing column.

**Syntax:**

alter table table-name rename column old-column-name to column-name;

**Example:**

**alter table customers rename column SALARY to PAYMENT;**

**select \* from customers;**

**ID NAME AGE ADDRESS PAYMENT**

**------ -------------------- ---------- ------------------------- ----------**

**1 Ramesh 32 Ahmedabad 2000**

**2 Khilan 25 Delhi 1500**

**3 kaushik 23 Kota 2000**

**4 Chaitali 25 Mumbai 6500**

**5 Hardik 27 Bhopal 8500**

**6 Komal 22 MP 4500**

1. **To Modify an existing Column:**

alter command is used to modify data type of an existing column.

**Syntax:**

alter table table-name modify (column-name datatype);

**Example:**

alter table customers modify (desig varchar2(10));

**Before Modification:**

DESIG VARCHAR2(20)

**After Modification:**

DESIG VARCHAR2(10)

1. **To Drop a Column**

alter command is also used to drop columns also.

**Syntax:**

alter table table-name drop(column-name);

**Example:**

alter table customers drop(PAN);

**NOTE:**

**Constraints:** Constraints are rules which the database enforces to maintain the integrity of the data.

**\*\*\* JOIN EXAMPLES:**

**SQL> select \* from class SQL> select \* from class\_info;**

**ID NAME ID ADDRESS**

**---------- -------------------- ---------- --------------------**

**1 Ramesh 1 Mumbai**

**2 Khilan 2 Kalyan**

**4 kaushik 3 Pune**

1. **INNER Join or EQUI Join**

SQL> select \* from class;

ID NAME

---------- --------------------

1 Ramesh

2 Khilan

4 kaushik

SQL> select \* from class\_info;

ID ADDRESS

---------- --------------------

1 Mumbai

2 Kalyan

3 Pune

SQL>

SQL>

SQL> SELECT \* from class, class\_info where class.id = class\_info.id;

ID NAME ID ADDRESS

---------- -------------------- ---------- --------------------

1 Ramesh 1 Mumbai

2 Khilan 2 Kalyan

SQL> SQL> select class.id,name,ADDRESS from class, class\_info where class.id = class\_info.id;

ID NAME ADDRESS

---------- -------------------- --------------------

1 Ramesh Mumbai

2 Khilan Kalyan

1. **NATURAL JOIN:**

SQL> SELECT \* from class NATURAL JOIN class\_info;

ID NAME ADDRESS

---------- -------------------- --------------------

1 Ramesh Mumbai

2 Khilan Kalyan

1. **OUTER JOIN:**
2. **Left Outer Join**

SQL> SELECT \* FROM class LEFT OUTER JOIN class\_info ON (class.id=class\_info.id);

**ID NAME ID ADDRESS**

**---------- -------------------- ---------- --------------------**

**1 Ramesh 1 Mumbai**

**2 Khilan 2 Kalyan**

**4 kaushik**

1. **Right Outer Join**

SQL> SELECT \* FROM class RIGHT OUTER JOIN class\_info on (class.id=class\_info.id);

**ID NAME ID ADDRESS**

**---------- -------------------- ---------- --------------------**

**1 Ramesh 1 Mumbai**

**2 Khilan 2 Kalyan**

**3 Pune**

1. **Full Outer Join**

**SQL> SELECT \* FROM class FULL OUTER JOIN class\_info on (class.id=class\_info.id);**

**ID NAME ID ADDRESS**

**---------- -------------------- ---------- --------------------**

**1 Ramesh 1 Mumbai**

**2 Khilan 2 Kalyan**

**3 Pune**

**4 kaushik**

**How to create user in oracle : (We cannot create trigger in SYS user so create a new user)**

SQL> create user test identified by test1

  2  /

User created.

SQL> show user

USER is "SYS"

SQL> GRANT CONNECT , RESOURCE, DBA to test;

Grant succeeded.

SQL> GRANT SELECT , UPDATE , INSERT ON test123 to TEST

  2  ;

Grant succeeded.

SQL> CONNECT TEST

Enter password:

Connected.

SQL> show user

USER is "TEST"

SQL>

**PACKAGE PROGRAMS/EXAMPLES:**

SQL> CREATE OR REPLACE PACKAGE c\_package AS

2 -- Adds a customer

3 PROCEDURE addCustomer(c\_id customers.id%type,

4 c\_name customerS.Name%type,

5 c\_age customers.age%type,

6 c\_addr customers.address%type,

7 c\_sal customers.salary%type);

8

9 -- Removes a customer

10 PROCEDURE delCustomer(c\_id customers.id%TYPE);

11 END c\_package;

/

12

Package created.

1 CREATE OR REPLACE PACKAGE BODY c\_package AS

2 PROCEDURE addCustomer(c\_id customers.id%type,

3 c\_name customerS.Name%type,

4 c\_age customers.age%type,

5 c\_addr customers.address%type,

6 c\_sal customers.salary%type)

7 IS

8 BEGIN

9 INSERT INTO customers (id,name,age,address,salary)

10 VALUES(c\_id, c\_name, c\_age, c\_addr, c\_sal);

11 END addCustomer;

12 PROCEDURE delCustomer(c\_id customers.id%type) IS

13 BEGIN

14 DELETE FROM customers

15 WHERE id = c\_id;

16 END delCustomer;

17\* END c\_package;

18 /

Package body created.

DECLARE

code customers.id%type:= 7;

BEGIN

c\_package.addcustomer(10, 'Rajnish', 25, 'Chennai', 3500); -- Change the values for unique constraint.

c\_package.addcustomer(11, 'Subham', 32, 'Delhi', 7500); -- Change the values for unique constraint.

c\_package.delcustomer(code);

END;

/

SQL> select \* from customers;

**ID NAME AGE ADDRESS SALARY**

**---------- -------------------- ---------- ------------------------- ----------**

**1 Ramesh 32 Ahmedabad 2000**

**2 Khilan 25 Delhi 1500**

**3 kaushik 23 Kota 2000**

**4 Chaitali 25 Mumbai 6500**

**5 Hardik 27 Bhopal 8500**

**6 Komal 22 MP 4500**

**10 Rajnish 25 Chennai 3500**

**11 Subham 32 Delhi 7500**

**8 rows selected.**

Using Package, we have added two records in customer table and deleted record have ID = 7