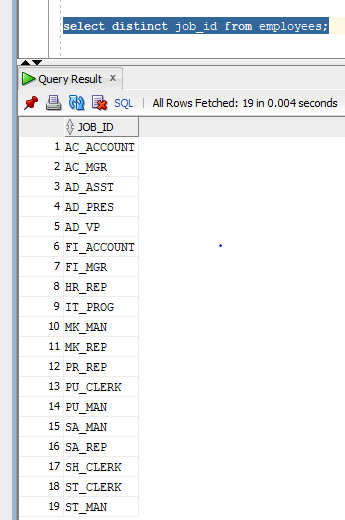
**DISTINCT**

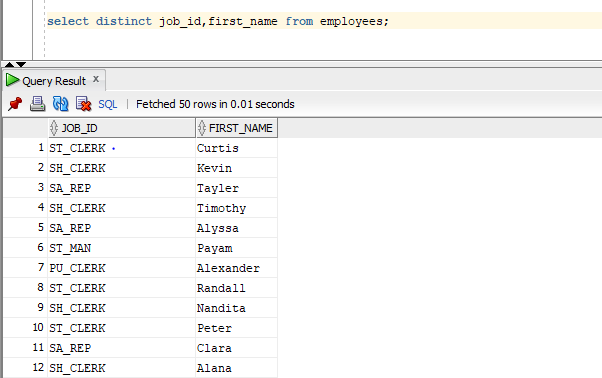
Distinct keyword is used to display distinct rows are unique data.

It will eliminate the duplicate records.

select distinct job\_id from employees;



We can use distinct not only with one column , we can use along with other columns. That time records will be returned more records based on second column.



**RENAME**

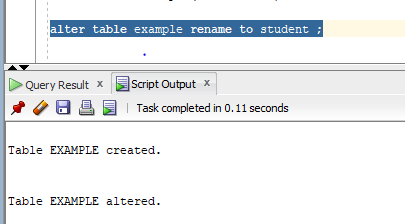
Rename is used to change the table name or column name.

Example

**To change table name**

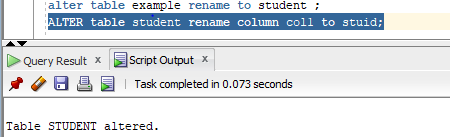
create table example(col1 NUMBER);

changing this table to student



To change column name

ALTER table student rename column col1 to stuid;



**To modify the size of column.**

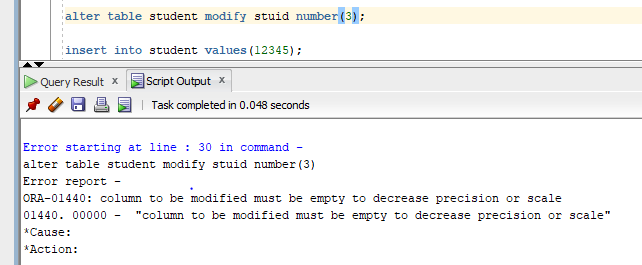
alter table student modify stuid number(10);

Table altered

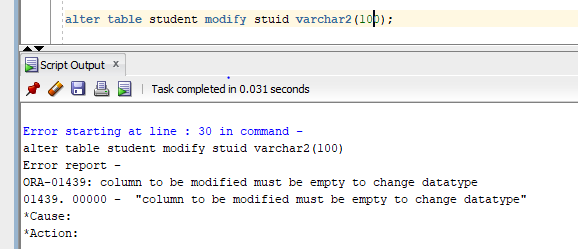
Incase data is having data more than the size we are changing then it gives error

Now student id is having 12345

Trying to change to size 3



Incase if column type is number and trying to change to varchar then also we get error.



**Create user**

**Syntax**

CREATE USER username IDENTIFIED by password

DEFAULT TABLESPACE tablespacename TEMPORARY TABLESPACE tablespace

QUOTA size/unlimited on tablespace\_name

PROFILE profile\_name

Password expire

ACCOUNT lock/unlock;

Create user test\_schema identified by Welcome1

DEFAULT TABLESPACE user TEMPORARY TABLESPACE temp

QUOTA 20M ON users

ACCOUNT unlock;

**To drop user**

DROP USER test\_Schema cascade;

To drop the connected user

1. Should be connected as sys or user with sys privilege
2. We should make database as restricted

Alter SYSTEM enable restricted session;

1. Need to get session id

Select sid,serial# from v$session where username=’TEST\_SCHEMA’;

1. Alter system kill session ’67,638’; (output from above query sid,serial#)
2. Drop user test\_Schema cascade;
3. Alter SYSTEM disable restricted session;

**To unlock user**

**Should be connects as SYS**

ALTER USER user\_name ACCOUNT UNLOCK;

**To change password**

ALTER USER user\_name IDENTIFIED BY password;

**CASE statement**

Using case we can achieve if then else logic without invoking any kind of SQL procedure.

There are two types of case expression

1. Simple case expression
2. Searched case expression

**Syntax**

CASE search\_Expression

WHEN input\_Expression\_1 THEN output\_result1

WHEN input\_expression\_2 THEN output\_Result2

.

.

.

WHEN input\_Expression\_n THEN output\_resultn

ELSE

Elseresult

End;

Here input\_expression and search\_Exprssion datatype must be same.

Result datatype along with ouput result must be same

Maximum number of arguments for CASE is 255

Example

SELECT case when comm is null THEN 0 ELSE comm END

From emp;

Example2

SELECT case deptno WHEN 10 THEN ‘Computer’

When 20 THEN ‘Science’

When 30 THEN ‘Maths’

ELSE ‘Sorry’

END

FROM emp;

Below the statemwnt will throw error

SELECT case deptno WHEN **‘10’** THEN ‘Computer’

When 20 THEN ‘Science’

When 30 THEN ‘Maths’

ELSE ‘Sorry’

END

FROM emp;

SELECT case deptno WHEN 10 THEN ‘Computer’

When 20 THEN ‘Science’

When 30 THEN ‘Maths’

**ELSE 10**

END

FROM emp;

**Error :- inconsistent datatypes**

**When none of case condition is matched then else will be returned**

Select (case ‘dog’ when ‘cat’ then ‘1 true’

When ‘cattle’ then ‘2 true’

When ‘rat’ then ‘3 true’

ELSE ‘sorry’ ) as result

From dual;

Result

Sorry

**When two condition matches then first result wll be produced**

Select (case ‘dog’ when ‘cat’ then ‘1 true’

When ‘dog’ then ‘2 true’

When ‘rat’ then ‘3 true’

When ‘dog’ then ‘4 true’

ELSE ‘sorry’ ) as result

From dual;

Result

2 true

**If search expression doesn’t match is any expression and else is omitted then it returns**

**NULL**

Select (case ‘dog’ when ‘cat’ then ‘1 true’

When ‘monkey’ then ‘2 true’

When ‘rat’ then ‘3 true’

When ‘baffalo’ then ‘4 true’

) as result

From dual;

Result

NULL

**Search case expression**

**Syntax**

CASE

WHEN condition\_1 THEN output\_result1

WHEN condition\_2 THEN output\_Result2

.

.

.

WHEN condition\_n THEN output\_resultn

ELSE

Elseresult

End;

**Example**

Select alert\_id,case when status\_internal\_id=2278 and deleted=0 THEN ‘Initiated and visible’

WHEN status\_internal\_id=3468 and deleted=0 THEN ‘investigate and visible’

ELSE ‘others’ END

From alerts;

Select empno (case when salary<=9000 THEN ‘Average salary’

When first\_name=’KING’ and manager\_id is null THEN ‘Head of Company’

When salary between 10000 and 13500 THEN ‘Medium Salary’

When salary in (2000,4000,8000) THEN ‘low salary’

Else other END) As result

FROM emp;

**CASE in PLSQL**

Set serveroutput on;

DECLARE

V\_Sal NUMBER;

V\_msg VARCHAR2(200);

BEGIN

SELECT sal

INTO v\_sal

FROM emp

WHERE empno=7689;

CASE WHEN v\_Sal <2000 THEN

V\_msg:=’Low’;

CASE WHEN v\_Sal >=2000 AND v\_Sal<3000 THEN

V\_msg:=’Fair’

CASE WHEN v\_Sal>3000 THEN

V\_msg:=’High’

END CASE;

DBSM\_OUTPUT.PUT\_LINE(‘Salary is ‘||v\_msg);

END;

/

**DECODE**

Decode is old way to perform IF THEN ELSE logic

CASE expression is easy to read compare to decode expression

Using decode we can perform only equality tests while using case we can perform logical test

CASE expression is supported by PL/SQL while decode only by SQL

**Syntax**

Decode(value,search\_value,result,default\_value)

Select decode(‘ramya’,’ramya’,’True ,string are same’,’False, string are not equal)

From dual;

Decode in plsql

Set serveroutput on;

DECLARE

V\_Sal NUMBER;

V\_msg VARCHAR2(200);

BEGIN

SELECT sal

INTO v\_sal

FROM emp

WHERE empno=7689;

DECODE(v\_Sal,2000,’Low’);

DBSM\_OUTPUT.PUT\_LINE(‘Salary is ‘||v\_msg);

END;

/

ERROR :- Decode is not a procedure or undefined

|  |  |
| --- | --- |
| **CASE** | **DECODE** |
| Compiles with ANSII SQL | Oracle propritary |
| Can work with logical operators other than equality | Works with only =/Like operator |
| Can work with predicates and searchable queries | Expressions are scalar values only |
| Needs data consistency | Data consistency is not needed |
| NULL=NULL returns FALSE | NULL is NULL return true |
| Can be used in PLSQL block | Can be used only in SQL statement |
| Can be used in parameter while calling procedure | Cannot be used as parameter while calling procedure |