DDL,DML,DCL,TCL COMMANDS

Data Definition Language: (DDL)

Data Definition Language (DDL) statements are used to define the database structure or schema. Some examples:

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed

Creating a table:

CREATE TABLE command is used to create a new table into the database. A table creation command requires three things:

- Name of the table
- Names of fields
- Definitions for each field /datatype

Syntax:

create table tablename(colname1 datatype,colname2 datatype ,colname2 datatype);

Eg:

create table employee(empname varchar2(20),empid number(10), emp_dob date);

Output:

```
create table employee(empname varchar2(20),empid number(10), emp_dob date);
```

Results Explain Describe Saved SQL History

```
Table created.
```

To view the structure of the table created:

Once the table is created the table structure can be viewed using desc command

Syntax:

describe tablename;
(OR)
desc tablename;

Eg:

Describe employee;

desc employee;

Output:

```
desc employee;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMPNAME	Varchar2	20	-	-	-	✓	-	-
	EMPID	Number	-	10	0	-	✓	-	-
	EMP_DOB	Date	7	-	-	-	✓	-	-

1 - 3

Alter:

ALTER statement is used when we want to change the name of the table or any table field. It is also used to add or delete an existing column in a table.

The ALTER statement is always used with "ADD", "DROP",”RENAME” and "MODIFY" commands .

1) **Add a column in table:**

Syntax:

alter table tablename add columnname datetype;

Eg:

alter table empp add empaddr varchar2(10);

Output:

```
alter table empp add empaddr varchar2(10);
```

Results Explain Describe Saved SQL History

```
Table altered.
```

2) **Add a multiple column:**

Syntax:

alter table tablename add(col1 datetype,col2 datetype .....)

Eg:

alter table empp add(gender varchar2(10),doj date);

Output:

```
alter table empp add(gender varchar2(10),doj date);
```

Results Explain Describe Saved SQL History

```
Table altered.
```

After adding a new column to the table then view the table structure using desc command.

```
desc empp ;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPP	EMPNAME	Varchar2	20	-	-	-	✓	-	-
	EMPID	Number	-	10	0	-	✓	-	-
	EMP_DOB	Date	7	-	-	-	✓	-	-
	EMPADDR	Varchar2	10	-	-	-	✓	-	-
	GENDER	Varchar2	10	-	-	-	✓	-	-
	DOJ	Date	7	-	-	-	✓	-	-

1 - 6

3) Modify column in a table:

MODIFY command is used to change the column definition of the table.

Syntax:

alter table tablename modify columname datatype;

Eg:

alter table empp modify empid varchar2(10);

Output:

```
alter table empp modify empid varchar2(10);
```

Results Explain Describe Saved SQL History

```
Table altered.
```

After modifying the column datatype then view the table structure using desc command.

```
desc empp;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPP	EMPNAME	Varchar2	20	-	-	-	✓	-	-
	EMPID	Varchar2	10	-	-	-	✓	-	-
	EMP_DOB	Date	7	-	-	-	✓	-	-
	EMPADDR	Varchar2	10	-	-	-	✓	-	-
	GENDER	Varchar2	10	-	-	-	✓	-	-
	DOJ	Date	7	-	-	-	✓	-	-

1 - 6

4) Rename a column in a table:

Syntax:

alter table tablename rename column oldcolumnname to newcolumnname;

Eg:

alter table empp rename column emp_dob to dateofbirth;

Output:

```
alter table empp rename column emp_dob to dateofbirth;
```

Results Explain Describe Saved SQL History

```
Table altered.
```

0.03 seconds

After renaming the column name then view the table structure using desc command.

```
desc empp;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPP	EMPNAME	Varchar2	20	-	-	-	✓	-	-
	EMPID	Varchar2	10	-	-	-	✓	-	-
	DATEOFBIRTH	Date	7	-	-	-	✓	-	-
	EMPADDR	Varchar2	10	-	-	-	✓	-	-
	GENDER	Varchar2	10	-	-	-	✓	-	-
	DOJ	Date	7	-	-	-	✓	-	-

1 - 6

5) **Rename a table:**

Syntax:

alter table oldtablename rename to newtablename;

Eg:

alter table employe rename to empp;

Output:

```
alter table employee rename to empp;
```

Results Explain Describe Saved SQL History

```
Table altered.
```

6) **Drop a column in a table:**

Syntax:

alter table tablename drop column columnname;

Eg:

alter table emp drop column gender;

Output:

```
alter table empp drop column gender;
```

Results Explain Describe Saved SQL History

```
Table dropped.
```

0.89 seconds

After dropping the column check whether the column is available in the table by issuing desc command.

```
desc empp;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPP

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPP	EMPNAME	Varchar2	20	-	-	-	✓	-	-
	EMPID	Varchar2	10	-	-	-	✓	-	-
	DATEOFBIRTH	Date	7	-	-	-	✓	-	-
	EMPADDR	Varchar2	10	-	-	-	✓	-	-
	DOJ	Date	7	-	-	-	✓	-	-

1 - 5

7) Truncate a table:

Syntax:

truncate table tablename;

Eg:

truncate table emp;

Output:

Autocommit Display 10

```
truncate table empp;
```

Results Explain Describe Saved SQL History

```
Table truncated.
```

0.12 seconds

After truncating the table check whether the table contents are available in the table by issuing select command.

```
Autocommit  Display 10
select * from empp;
Results  Explain  Describe  Saved SQL  History
no data found
```

8) **Drop a table:**

Syntax:

drop table tablename;

Eg:

drop table emp;

Output:

```
drop table empp;
Results  Explain  Describe  Saved SQL  Histor

Table dropped.

0.02 seconds
```

After the table is dropped check whether the table is available by issuing describe command.

```
desc empp;
Execute  Load Script  Save Script  Cancel
ERROR:
ORA-04043: object empp does not exist
```

Data Manipulation Language:(DML)

Data Manipulation Language (DML) statements are used for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database

- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

1) INSERT VALUES INA TABLE:

INSERT statement is used to insert a single record or multiple records into a table in Oracle.There are 3 types of insertion:

- static insertion
- dynamic insertion
- specific insertion

STATIC INSERTION: (Inserting a single record using the Values keyword):
The simplest way to create an Oracle INSERT query to list the values using the VALUES keyword

Syntax:

insert into tablename values(values1,values2,values3…..valueN)

Eg:

insert into empp values('anand','16E101','16-may-1997','coimbatote','12-apr-2015');

Output:

```
insert into empp values('anand','16E101','16-may-1997','coimbatote','12-apr-2015');
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

DYNAMIC INSERTION:

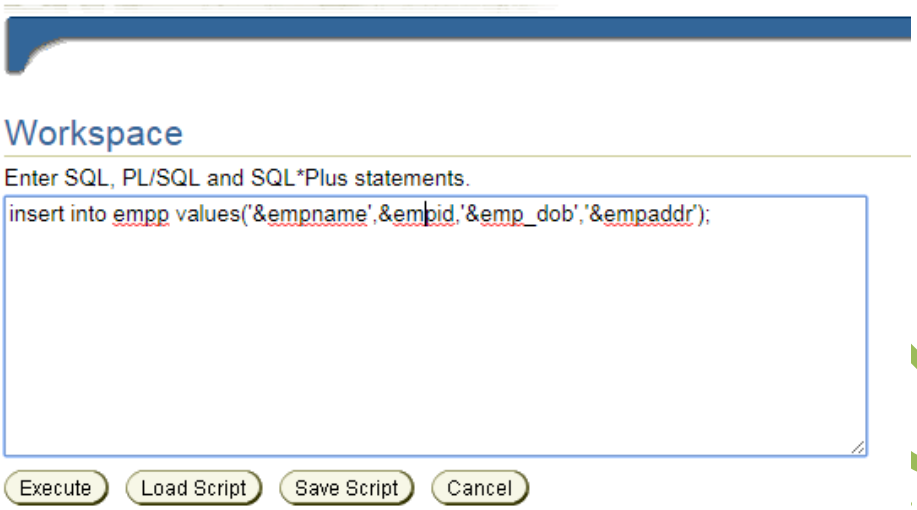
To insert the column values in the table by receiving inputs from the user at the runtime.

Syntax:

insert into tablename values(‘&col1’,’&col2’,&col3);

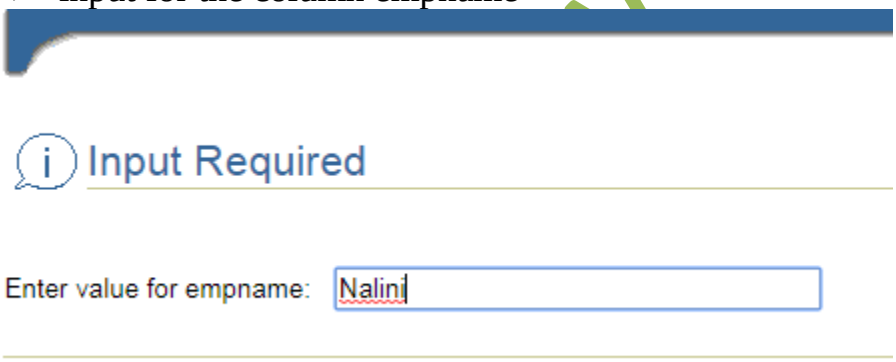
Eg:

insert into empp values('&empname',&empid,'&emp_dob','&empaddr');

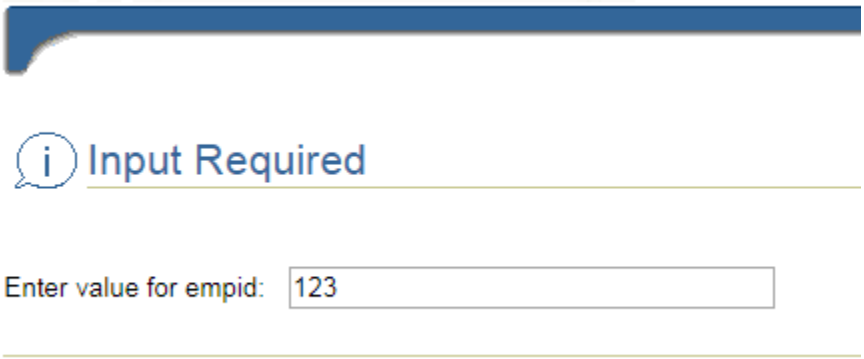


Once the query is issued then the inputs are received from the user at runtime as follows

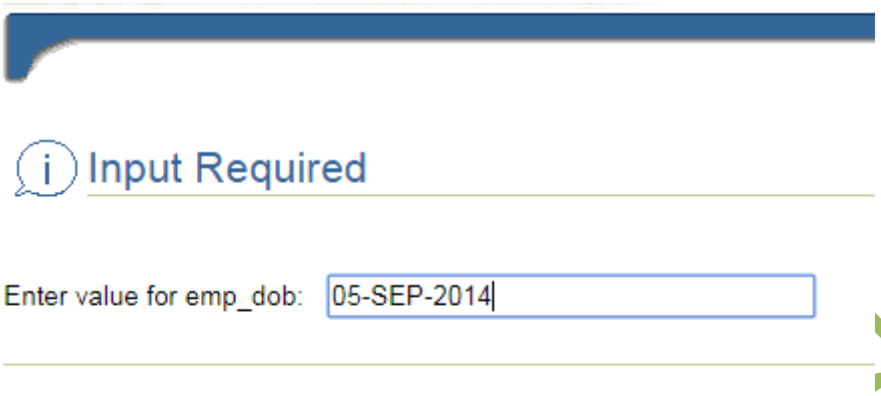
- input for the column empname



- input for the column empid



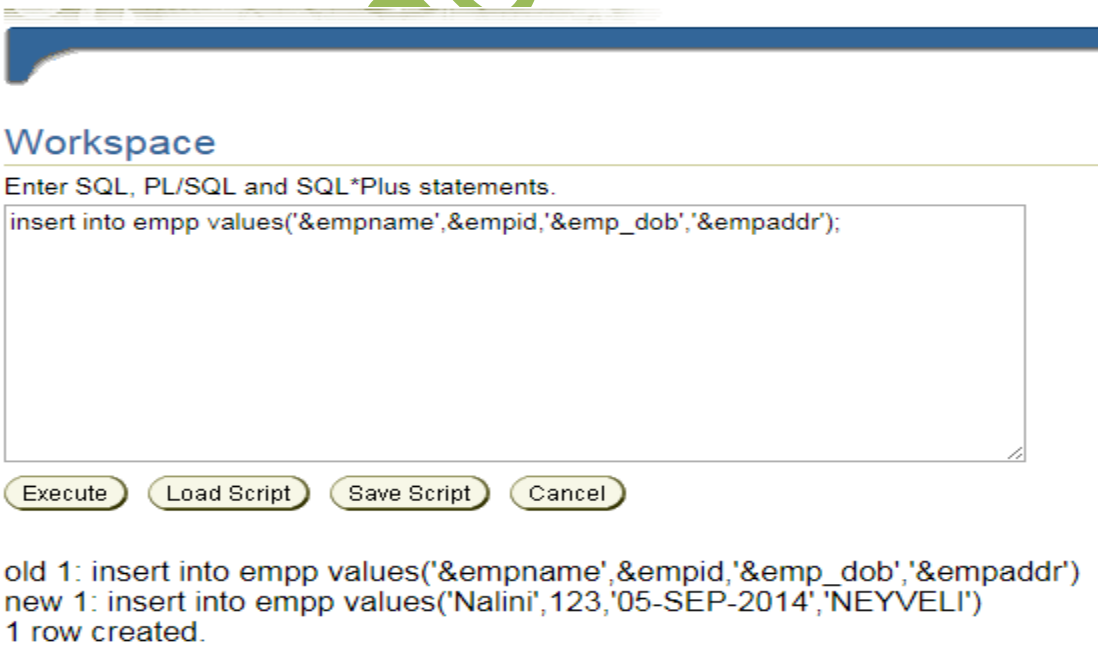
- input for the column emp_dob



- input for the column empaddr

After receiving all the inputs from the user then the values are inserted into the table.

Output:



SPECIFIC INSERTION:

Insert values to two or more specific columns in a table.

Syntax:

insert into tablename (col1,col2) values(col_values1,col_values2);

Eg:

insert into empp(empname,empid,doj) values('kumar','16m201','13-oct-2001');

Output:

```
insert into empp(empname,empid,doj) values('kumar','16m201','13-oct-2001');
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

0.01 seconds

After inserting ,issue the select command to display the values inserted in a table.

2) **SELECT Statement:**

The SELECT statement is used to fetch data from the one or more tables in Oracle We can retrieve records of all fields or specified fields.

Syntax for all fields:

SELECT * FROM tables [WHERE conditions];

Syntax for selecting particular column/field:

SELECT expressions
FROM tables
[WHERE conditions];

SELECT Example 1: for particular fields

SELECT empid,empname,doj
FROM empp;

Output:

```
SELECT empid,empname,doj FROM empp;
```

Results Explain Describe Saved SQL History

EMPID	EMPNAME	DOJ
16E101	anand	12-APR-15
16m201	kumar	13-OCT-01
16E102	akila	22-DEC-14

3 rows returned in 0.00 seconds CSV Export

SELECT Example 2: for all fields

Specify either all fields or * (asterisk) symbol.

Eg:

Select * from empp;

Output:

```
select * from empp;
```

Results Explain Describe Saved SQL History

EMPNAME	EMPID	DATEOFBIRTH	EMPADDR	DOJ
anand	16E101	16-MAY-97	coimbatote	12-APR-15
kumar	16m201	-	-	13-OCT-01
akila	16E102	14-NOV-98	erode	22-DEC-14

3 rows returned in 0.00 seconds CSV Export

3) UPDATE Query:

UPDATE statement is used to update data of the MySQL table within the database. It is used to modify the table.

Syntax:

UPDATE table_name SET field1=new-value1, field2=new-value2
[WHERE Clause]

Note:

- One or more field can be updated altogether.
- Any condition can be specified by using WHERE clause.
- we can update values in a single table at a time.
- WHERE clause is used to update selected rows in a table.

Eg:

update empp set empname='anandhi' where empid='16E101';

Output:

```
update empp set empname='anandhi' where empid='16E101';
```

Results Explain Describe Saved SQL History

```
1 row(s) updated.
```

0.00 seconds

After updating the table ,to check whether the values is updated issue select command.

```
select * from empp;
```

Results Explain Describe Saved SQL History

EMPNAME	EMPID	DATEOFBIRTH	EMPADDR	DOJ
anandhi	16E101	16-MAY-97	coimbatote	12-APR-15
kumar	16m201	-	-	13-OCT-01
akila	16E102	14-NOV-98	erode	22-DEC-14

3 rows returned in 0.00 seconds CSV Export

4) DELETE Statement:

DELETE statement is used to delete data from the table within the database. By using delete statement, we can delete records on the basis of conditions.

Syntax:

DELETE FROM table_name
WHERE
(Condition specified);

Example:

DELETE FROM empp
WHERE empid='16E102';

Output:

```
DELETE FROM empp  WHERE empid='16E102';
```

Results Explain Describe Saved SQL History

```
1 row(s) deleted.
```

0.00 seconds

After deleting the table ,to check whether the values are deleted, issue select command.

```
select * from empp;
```

Results Explain Describe Saved SQL History

EMPNAME	EMPID	DATEOFBIRTH	EMPADDR	DOJ
anandhi	16E101	16-MAY-97	coimbatote	12-APR-15
kumar	16m201	-	-	13-OCT-01

2 rows returned in 0.02 seconds CSV Export

DCL COMMANDS:(Data Control Language)

Data Control Language(DCL) is used to control privilege in Database. To perform any operation in the database, such as for creating tables, sequences or views we need privileges. Privileges are of two types,

- **System :** creating session, table etc are all types of system privilege.
- **Object :** any command or query to work on tables comes under object privilege.

DCL defines two commands,

- **Grant :** Gives user access privileges to database.
- **Revoke :** Take back permissions from user.

GRANT:

We can grant users various privileges to tables. These permissions can be any combination of SELECT, INSERT, UPDATE, DELETE, INDEX, CREATE, ALTER, DROP, GRANT OPTION or ALL.

Syntax:

GRANT privileges ON object TO user;

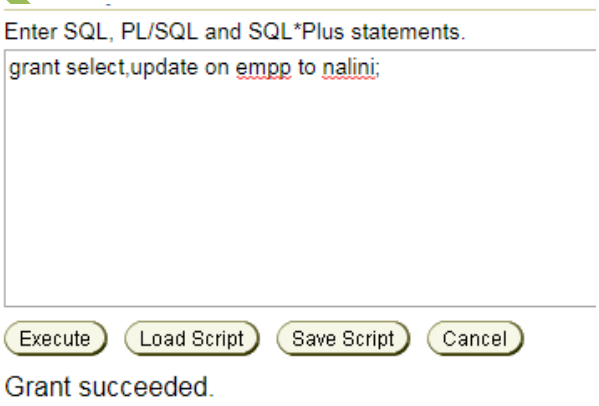
Privileges:	It can be any of the following values: **Privilege** — **Description** SELECT — Ability to perform SELECT statements on the table. INSERT — Ability to perform INSERT statements on the table. UPDATE — Ability to perform UPDATE statements on the table. DELETE — Ability to perform DELETE statements on the table. INDEX — Ability to create an index on an existing table. CREATE — Ability to perform CREATE TABLE statements. ALTER — Ability to perform ALTER TABLE statements to change the table definition. DROP — Ability to perform DROP TABLE statements. GRANT OPTION — Allows you to grant the privileges that you possess to other users. ALL — Grants all permissions except GRANT OPTION.
Object :	The name of the database object that you are granting permissions for. In the case of granting privileges on a table, this would be the table name.
User :	The name of the user that will be granted these privileges.

Example:

```
GRANT SELECT,UPDATE ON empp TO 'nalini';
```

To grant SELECT,UPDATE privileges on a table called *empp* to a user name *nalini*, we have to run the following GRANT statement:

Output:



REVOKE:

Once we have granted privileges, we may need to revoke some or all of these privileges. To do this, we can run a revoke command. we can revoke any combination of SELECT, INSERT, UPDATE,

DELETE, REFERENCES, ALTER, or ALL.

SYNTAX:

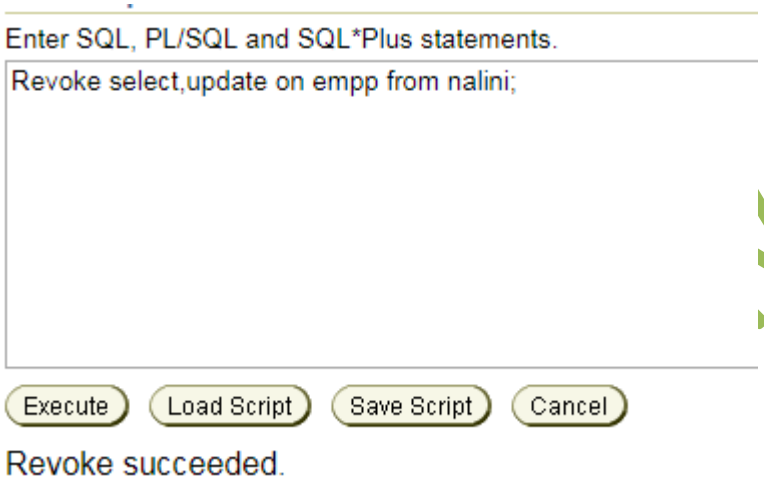
REVOKE privileges ON object FROM user;

Example

REVOKE SELECT, UPDATE ON empp FROM 'nalini';

To revoke SELECT and UPDATE privileges on a table called *empp* from a user named nalini, we would run the following REVOKE statement:

Output:



TCL COMMANDS:

Transaction control statements manage changes made by DML statements.TCL Statements available in Oracle are

COMMIT : Make changes done in transaction permanent.
ROLLBACK : Rollbacks the state of database to the last commit point.
SAVEPOINT : Use to specify a point in transaction to which later you can rollback.

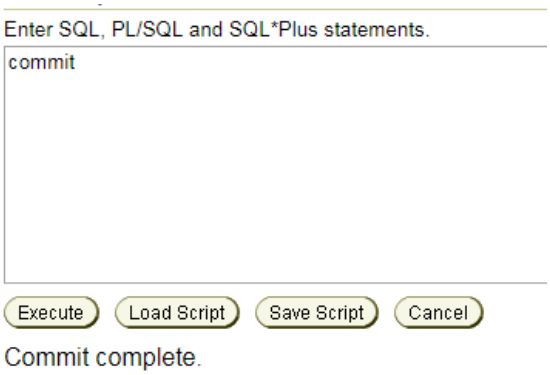
1) Commit :

Commit command is used to permanently save any transaaction into database.

Syntax:

commit;

Output:



2) Savepoint :

Savepoint command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.

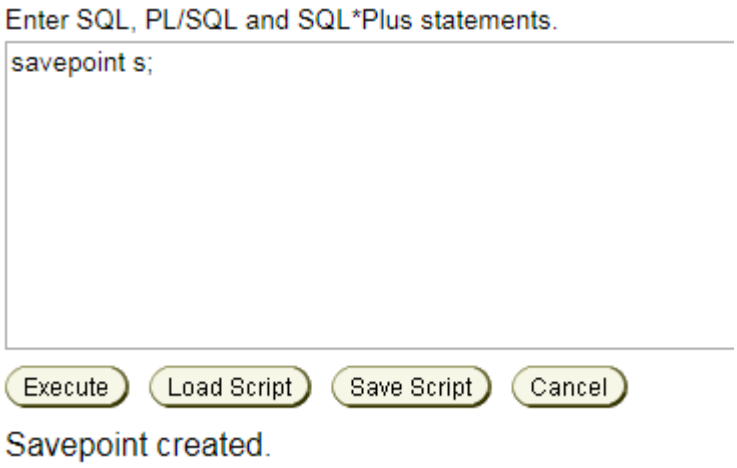
Syntax:

savepoint *savepoint-name*;

Eg:

savepoint S;

Output:



3) Rollback:

This command restores the database to last commited state. It is also use with savepoint command to jump to a savepoint in a transaction.

Syntax:

STEP 1:

Perform some manipulation work

STEP2:

Create a savepoint S1;

STEP 3:

Perform some manipulation work

STEP4:

Create a savepoint S2;

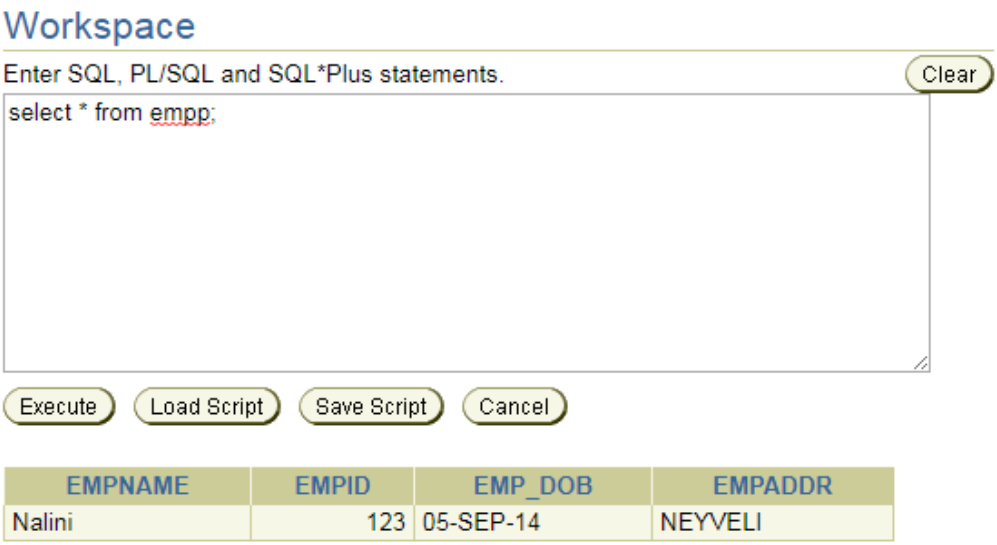
STEP 5:

Rollback to S1;

Example:

STEP 1:

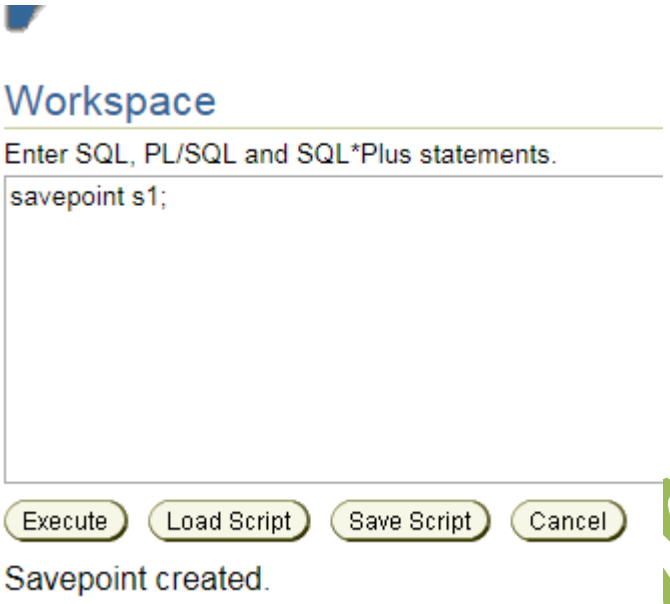
Initially view the content of the table



EMPNAME	EMPID	EMP_DOB	EMPADDR
Nalini	123	05-SEP-14	NEYVELI

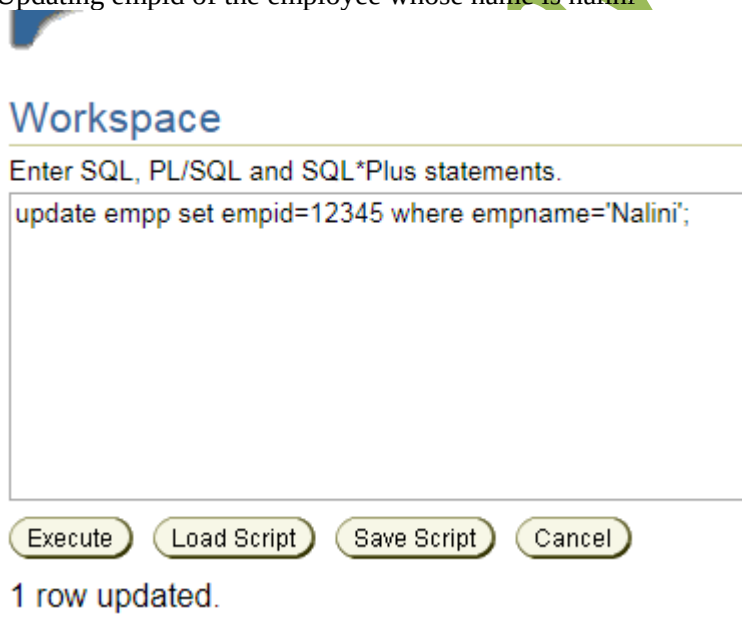
STEP2:

Create a savepoint S1;



STEP 3:

Updating empid of the employee whose name is nalini



Connected as SAKTHI@

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

```
select *from empp;
```

Execute | Load Script | Save Script | Cancel

EMPNAME	EMPID	EMP_DOB	EMPADDR
Nalini	12345	05-SEP-14	NEYVELI

STEP4:

Create a savepoint S2;

Workspace

Enter SQL, PL/SQL and SQL*Plus statements.

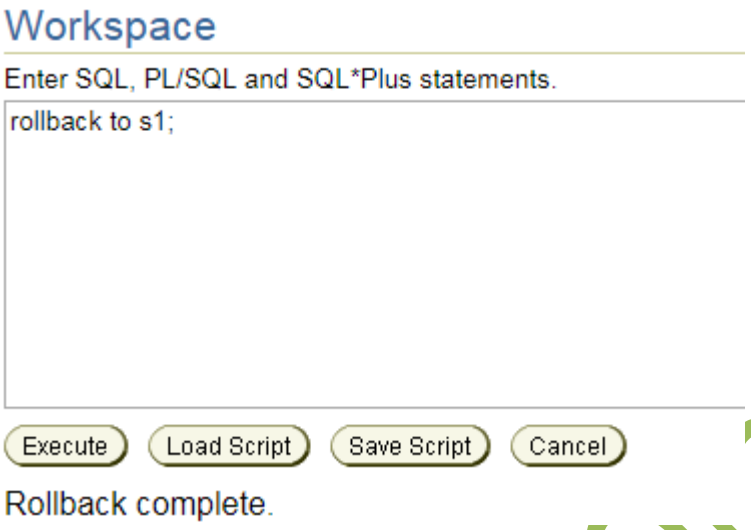
```
savepoint s2;
```

Execute | Load Script | Save Script | Cancel

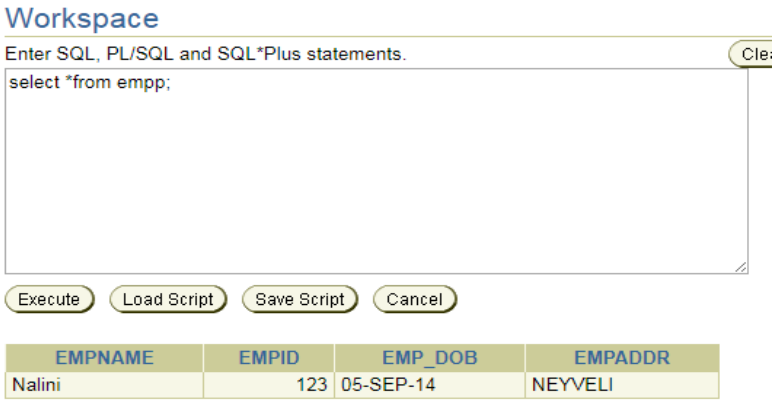
Savepoint created.

STEP 5:

Rollback to S1;



Issue select command:



After issuing rollback command the empid of the employee is same as the intial one.The updation performed after the savepoint s1 is not included.(undo operation has been performed)