STATEMENTS ARE CLASSIFIED INTO 5 DIFFERENT TYPES

- DATA DEFINITION LANGUAGE (DDL)
- DATA MANIPULATION LANGUAGE (DML)
- TRANSACTION CONTROL LANGUAGE (TCL)
- DATA CONTROL LANGUAGE (DCL)
- DATA QUERY LANGUAGE (DQL)

1. DATA DEFINITION LANGUAGE (DDL):

DDL is used to construct an object in the database and deals with the Structure of the Object.

It has 5 statements :

1. CREATE
2. RENAME
3. ALTER
4. TRUNCATE
5. DROP

1. CREATE : IT IS USED TO BUILD / CONSTRUCT AN OBJECT "

Object / Entity can be a Table or a View (Virtual Table) .

How to Create a Table :

- Name of the table. Tables cannot have same names.
- Number of Columns.
- Names of the columns.
- Assign datatypes for the Columns. (not mandatory)

Example 1:
Table_name : **Customer** , Number of columns : **4**

Column_Name	**CID**	**CNAME**	**CNO**	**ADDRESS**
Datatypes	Number(2)	Varchar(10)	Number (10)	Varchar(15)
Null / Not Null	Not Null	Not Null	Not Null	Null
Unique	Unique		Unique	
Check			Check (length(CNO) = 10)	
Primary Key	Primary Key			
FOREIGN KEY				

Syntax to create a table :

```
CREATE  TABLE   Table_Name(

          Column_Name1  datatype  constraint_type ,
          Column_Name2  datatype  constraint_type ,
          Column_Name3  datatype constraint_type ,

          .

          .

          Column_NameN  datatype   constraint_type
```

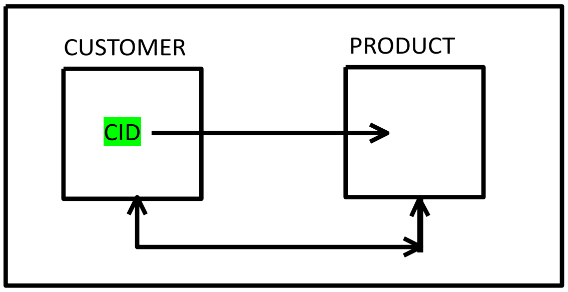
Example :

```
CREATE TABLE CUSTOMER(

          CID Number(2) primary key ,CNAME
          Varchar(10) ,
          CNO Number(10) not null check( length( CNO ) = 10 ) ,
          ADDRESS Varchar(15)

          );
```

NOTE :

To Describe the table:

```
Syntax:     DESC Table_Name ;
```



Example 2:

Table_Name : **PRODUCT**

Number of Columns : 4

Product

Column_Name	**PID**	**PNAME**	**PRICE**	**CID**
Datatypes	Number(2)	Varchar(10)	Number (7,2)	Number(2)
Null / Not Null	Not Null	Not Null	Not Null	Null
Unique	Unique			
Check			Check (Price > 0)	
Primary Key	Primary Key			
Foreign Key				**Foreign Key**

Syntax to create a table :

```
CREATE  TABLE   Table_Name(

          Column_Name1 datatype constraint_type ,
          Column_Name2 datatype constraint_type ,
          Column_Name3 datatype constraint_type ,

          .

          .

          Column_NameN datatype ,

          Constraint Foreign key references Parent_Table_Name(Column_Name)

     );
```

Example :

```
CREATE TABLE PRODUCT(
          PID Number(2) primary key ,PNAME
          Varchar(10) , PRICE Number(7,2)
          check( Price > 0) ,CID Number(2) ,

          Constraint CID_FK Foreign Key(CID) references CUSTOMER( CID )

          );
```

2.**RENAME :** IT IS USED TO CHANGE THE NAME OF THE OBJECT

Syntax: RENAME Table_Name TO New_Name ;

Example :

```
RENAME Customer TO Cust ;
```

3. **ALTER :** IT IS USED TO MODIFY THE STRUCTURE OF THE TABLE

- **TO ADD A COLUMN :**

Syntax:
```
ALTER  TABLE  Table_Name
       ADD Column_Name Datatype Constraint_type ;
```

Example : ALTER TABLE Cust
```
ADD MAIL_ID Varchar(15) ;
```

- **TO DROP A COLUMN :**

Syntax: ALTER TABLE Table_Name DROP
COLUMN Column_Name ;

Example : ALTER TABLE Cust
DROP COLUMN MAIL_ID ;

- **TO RENAME A COLUMN :**

Syntax: ALTER TABLE Table_Name
RENAME COLUMN Column_Name TO new_Column_Name ;

Example : ALTER TABLE Cust
RENAME COLUMN CNO TO PHONE_NO ;

- **TO MODIFY THE DATATYPE :**

Syntax: ALTER TABLE Table_Name
MODIFY COLUMN_NAME New_Datatype;

Example : ALTER TABLE Cust
MODIFY CNAME CHAR(10) ;

- **TO MODIFY NOT NULL CONSTRAINTS :**

Syntax: ALTER TABLE Table_Name
MODIFY COLUMN_NAME Existing_datatype [NULL]/NOT NULL;

Example : ALTER TABLE Cust
MODIFY ADDRESS Varchar(15) Not Null ;

4. **TRUNCATE :** IT IS USED TO REMOVE ALL THE RECORDS FROM THE TABLE PERMANENTLY.

Syntax: TRUNCATE TABLE Table_Name ;

Cust

Cid	**Cname**	**Phone_no**	**Address**
1	A	1234567890	BANGALORE
2	B	1234567899	MYSORE
3	C	1234567880	MANGALORE

Example : TRUNCATE TABLE Cust ;

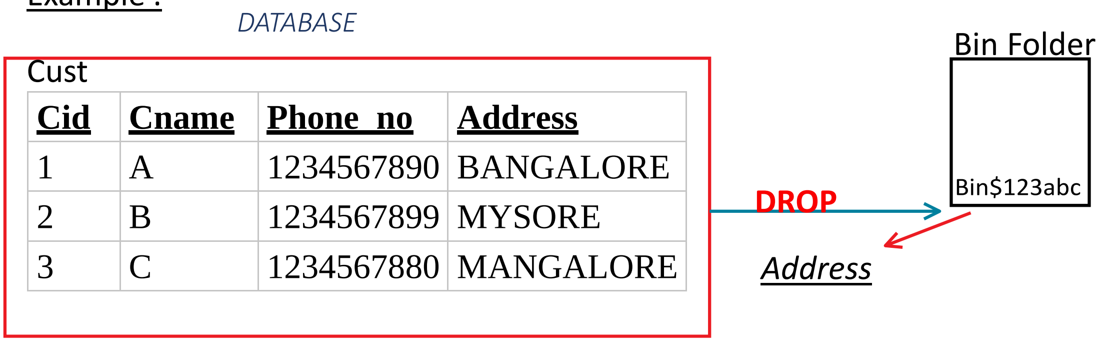
Cust

Cid	**Cname**	**Phone_no**	**Address**

5.DROP : IT IS USED TO REMOVE THE TABLE FROM THE DATABASE.

Syntax: DROP TABLE Table_Name ;

Example :



DATABASE

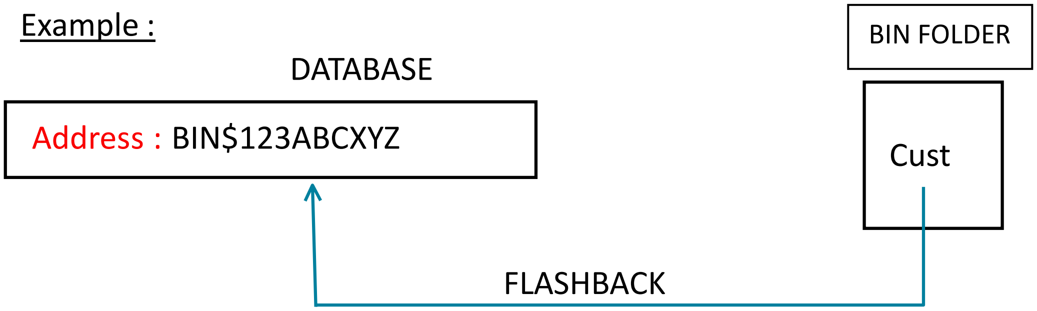
Cust

Cid	**Cname**	**Phone_no**	**Address**
1	A	1234567890	BANGALORE
2	B	1234567899	MYSORE
3	C	1234567880	MANGALORE

TO RECOVER THE TABLE :

Syntax: FLASHBACK TABLE Table_Name TO BEFORE DROP ;

Example :



EXAMPLE :

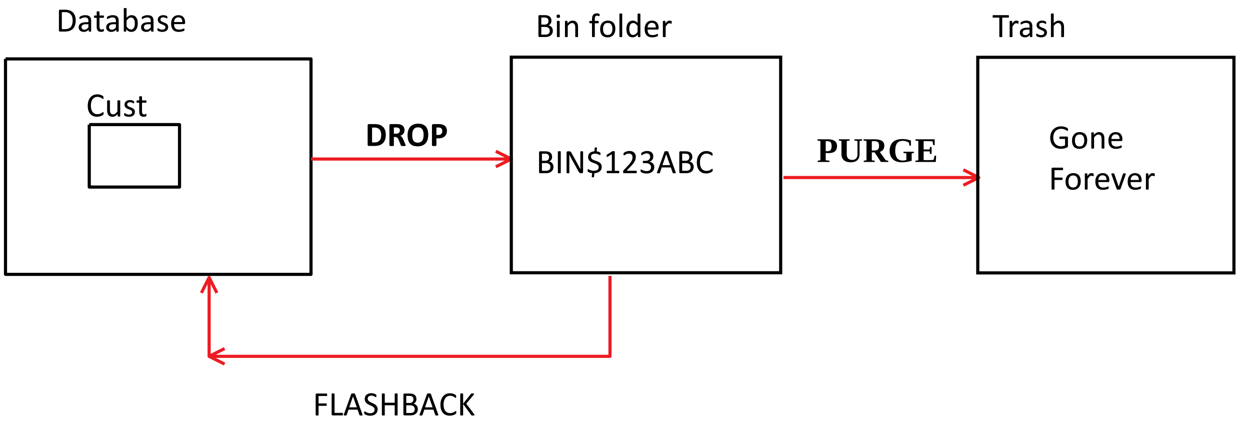
FLASHBACK TABLE CustTO

BEFORE DROP ;

TO DELETE THE TABLE FROM BIN FOLDER :

Syntax: PURGE TABLE Table_Name ;

Example : PURGE TABLE Cust ;



NOTE : DDL STATEMENTS ARE AUTO-COMMIT STATEMENTS

DATA MANIPULATION LANGUAGE (DML)

It is used to Manipulate the Object by performing insertion , updating and deletion .

1. INSERT
2. UPDATE
3. DELETE

1. **INSERT :** It is used to insert / create records in the table .

```
Syntax: INSERT INTO Table_Name VALUES( v1 , v2 , v3 …… ) ;
```

CUSTOMER

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)

```
INSERT INTO CUSTOMER VALUES( 1 , 'DINGA' , 9876543210 ,'BANGALORE' ) ;
```

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	DINGA	9876543210	BANGALORE

```
INSERT INTO CUSTOMER VALUES( 2 , 'DINGI' , 9876543211 ,'MANGALORE' ) ;
```

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	DINGA	9876543210	BANGALORE
2	DINGI	9876543211	MANGALORE

PRODUCT

PID	PNAME	PRICE	CID
NUMBER(2)	VARCHAR(10)	NUMBER(6,2)	NUMBER(3)

INSERT INTO PRODUCT VALUES(11 , 'iPhone' , 10000 , 2);

PID	PNAME	PRICE	CID
NUMBER(2)	VARCHAR(10)	NUMBER(6,2)	NUMBER(3)
11	iPhone	10000	2

INSERT INTO PRODUCT VALUES(22 , 'Mac Book' , 20000 , 1);

PID	PNAME	PRICE	CID
NUMBER(2)	VARCHAR(10)	NUMBER(6,2)	NUMBER(3)
11	iPhone	10000	2
22	Mac Book	20000	1

2.UPDATE : It is used to modify an existing value .

Syntax: **UPDATE** Table_Name
SET Col_Name = Value , Col_Name = Value ,,,,,
[WHERE stmt] ;

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	ABHI	1234567890	BANGALORE
2	ABDUL	9876543210	MANGALORE

- WAQT update the phone number of Abdul to 7778889994

  UPDATE CUSTOMER SET
  CNO = 7778889994

  WHERE CNAME ='ABDUL' ;

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	ABHI	1234567890	BANGALORE
2	ABDUL	7778889994	MANGALORE

- WAQT change the address of the customer to Mysore whose cid is 1 .

```
UPDATE  CUSTOMER

SET  ADDRESS = 'MYSORE'

WHERE CID = 1 ;
```

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	ABHI	1234567890	MYSORE
2	ABDUL	7778889994	MANGALORE

3.DELETE : It is used to remove a particular record from the table .

Syntax: **DELETE** FROM Table_Name[
WHERE stmt];

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	ABHI	1234567890	BANGALORE
2	ABDUL	1234567891	MANGALORE

- WAQT remove abdul from the list of customers .

```
DELETE  FROM  CUSTOMERWHERE
CNAME ='ABDUL' ;
```

CID	CNAME	CNO	ADDDRESS
NUMBER(2)	VARCHAR(10)	NUMBER(10)	VARCHAR(20)
1	ABHI	1234567890	BANGALORE

ASSIGNMENT ON DML STATEMENTS :

1. WAQT update the salary of employee to double their salary if He is working as a manager .

2. WAQT change the name of SMITH to SMIITH .

3. WAQT modify the job of KING to 'PRESIDENT' .

4. WAQT to change name of ALLEN to ALLEN MORGAN .

5. WAQT hike the salary of the employee to 10% . If employees earn less than 2000 as a salesman .

6. WAQ TO delete the employees who don’t earn commission .

7. WAQ to remove all the employees hired before 1987 in dept 20

8. Differentiate between TRUNCATE and DELETE statements .

TRUNCATE	**DELETE**
Belongs to DDL	Belongs to DML
Removes all the records from the Table permanently .	Removes a particular record from the Table .
Auto COMMIT	Not auto COMMIT .

3. TRANSACTION CONTROL LANGUAGE (TCL)

"It is used to control the transactions done on the database ".

The DML Operations performed on the Database are known as Transactions such as Insertion , Updating and Deletion .

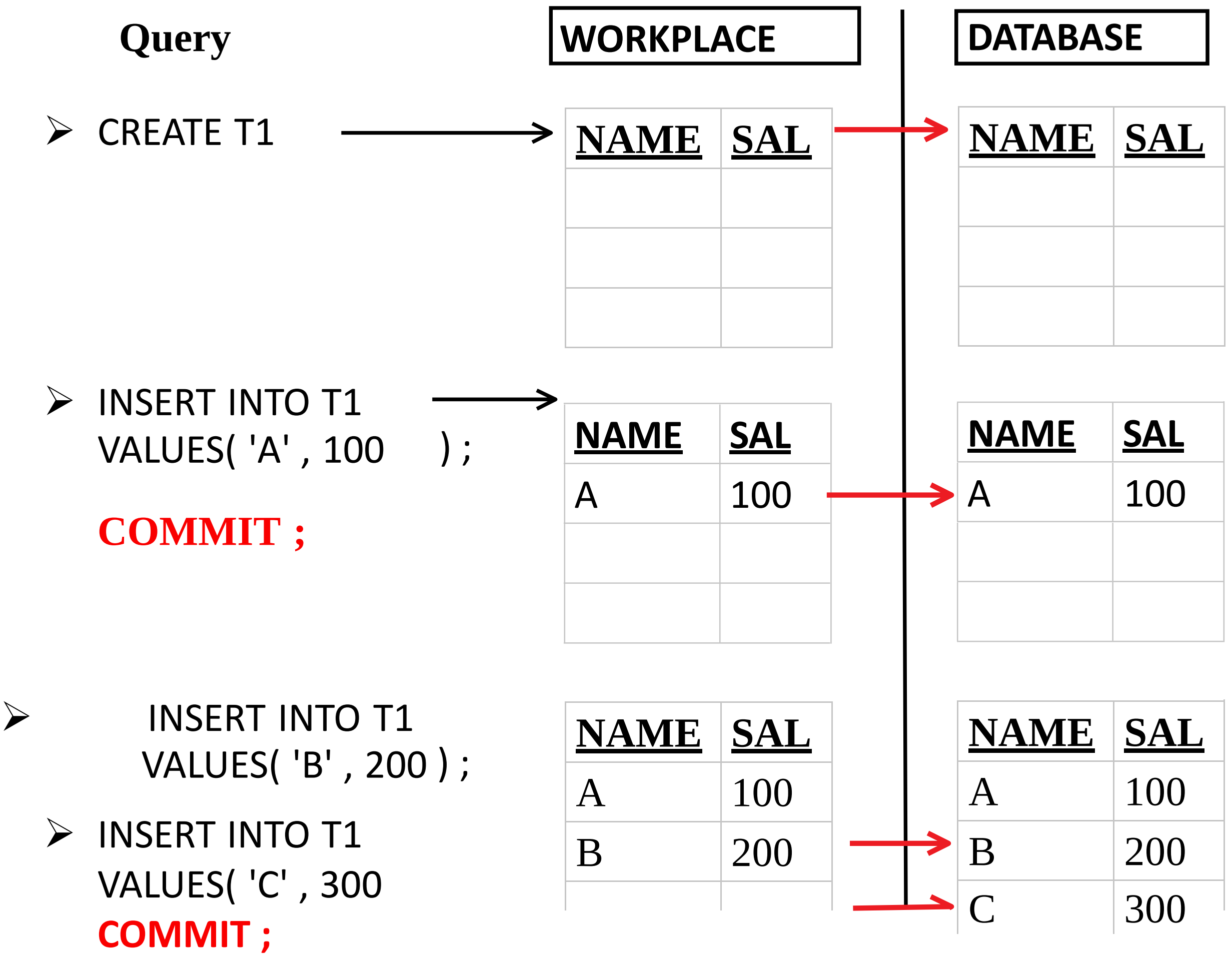
We have 3 Statements :

1. COMMIT
2. ROLLBACK
3. SAVEPOINT

1.COMMIT : "This statement is used to SAVE the transactions into the DB ".

Syntax: **COMMIT ;**

Example :



Query

WORKPLACE

DATABASE

- CREATE T1

NAME	SAL

NAME	SAL

- INSERT INTO T1 VALUES('A' , 100) ;

COMMIT ;

NAME	SAL
A	100

NAME	SAL
A	100

- INSERT INTO T1 VALUES('B' , 200) ;

- INSERT INTO T1 VALUES('C' , 300

COMMIT ;

NAME	SAL
A	100
B	200

NAME	SAL
A	100
B	200
C	300

VALUES('C' , 300) ;

COMMIT ;

B	200
C	300

B	200
C	300

- UPDATE T1 SET
  SAL = 1000
  WHERE NAME = 'A' ;

NAME	**SAL**
A	1000
~~B~~	200
C	300

- **COMMIT ;**

NAME	**SAL**
A	1000
B	200
C	300

2. ROLLBACK :

This statement is used to Obtain only the saved data from the DB .

It will bring you to the point where you have committed for the last time .

SYNTAX: **ROLLBACK ;**

3. SAVEPOINT :

This statement is used to mark the positions or restoration points . (nothing related to DB) .

SYNTAX: **SAVEPOINT** Savepoint_Name ;

Example :

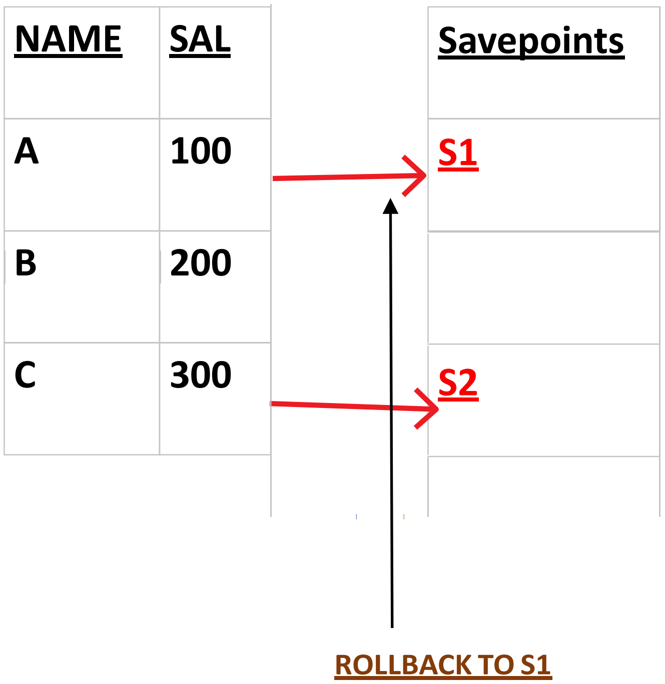
Query

- INSERT INTO T1 VALUES('A',100) ;
- SAVEPOINT S1 ;
- INSERT INTO T1 VALUES('B' , 200) ;
- INSERT INTO T1

  VALUES('B' , 200) ;
- INSERT INTO T1 VALUES('C' , 300) ;
- SAVEPOINT S2 ;
- INSERT INTO T1 VALUES('D' , 400) ;
- INSERT INTO T1 VALUES('E' , 500) ;
- SAVEPOINT S3 ;
- INSERT INTO T1 VALUES('F' , 600) ;

WORKPLACE

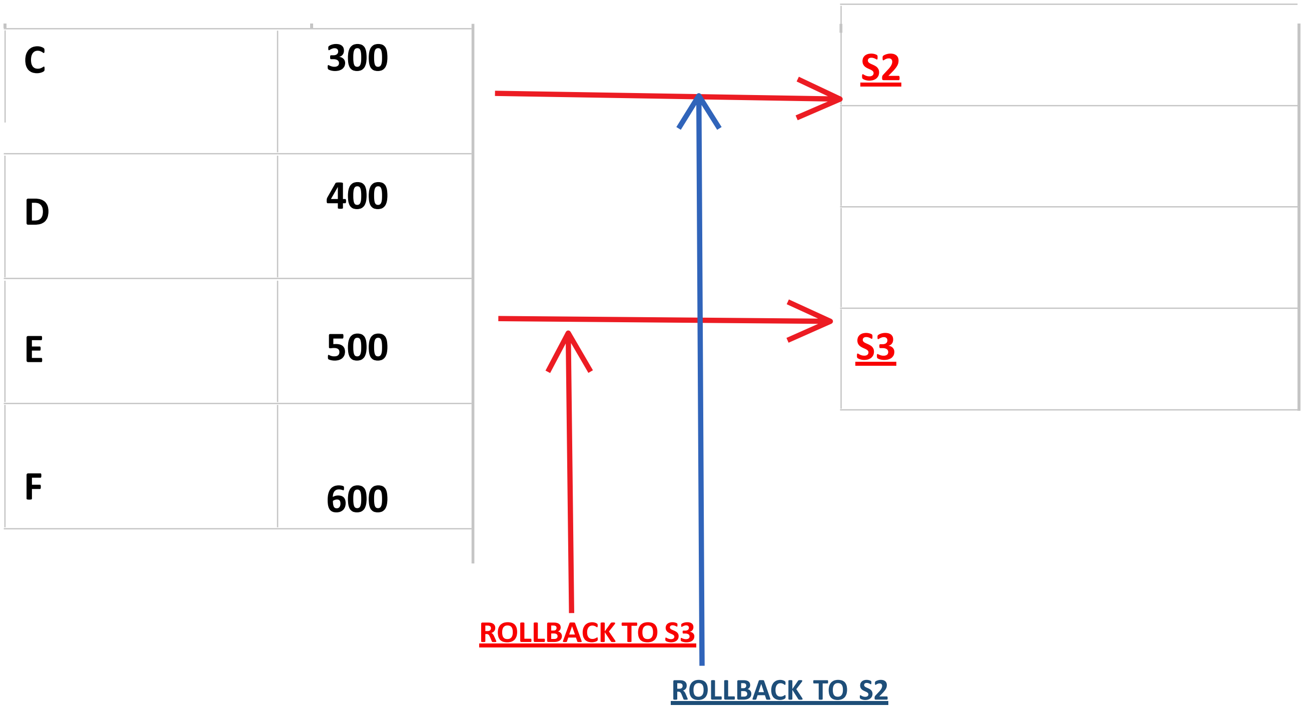
SAVEPOINT



NAME	SAL
A	100
B	200
C	300

Savepoints
S1
S2

ROLLBACK TO S1



C	300
D	400
E	500
F	600

S2
S3

ROLLBACK TO S3

ROLLBACK TO S2

SYNTAX: **ROLLBACK TO** Savepoint_Name ;

4. DATA CONTROL LANGUAGE :

"This statement is used to control the flow of data between the users ".

We have 2 statements :

1. GRANT
2. REVOKE

1.GRANT : THIS STATEMENT IS USED TO GIVE PERMISSION TO AUSER .

SYNTAX: **GRANT** SQL_STATEMENT
ON TABLE_NAME

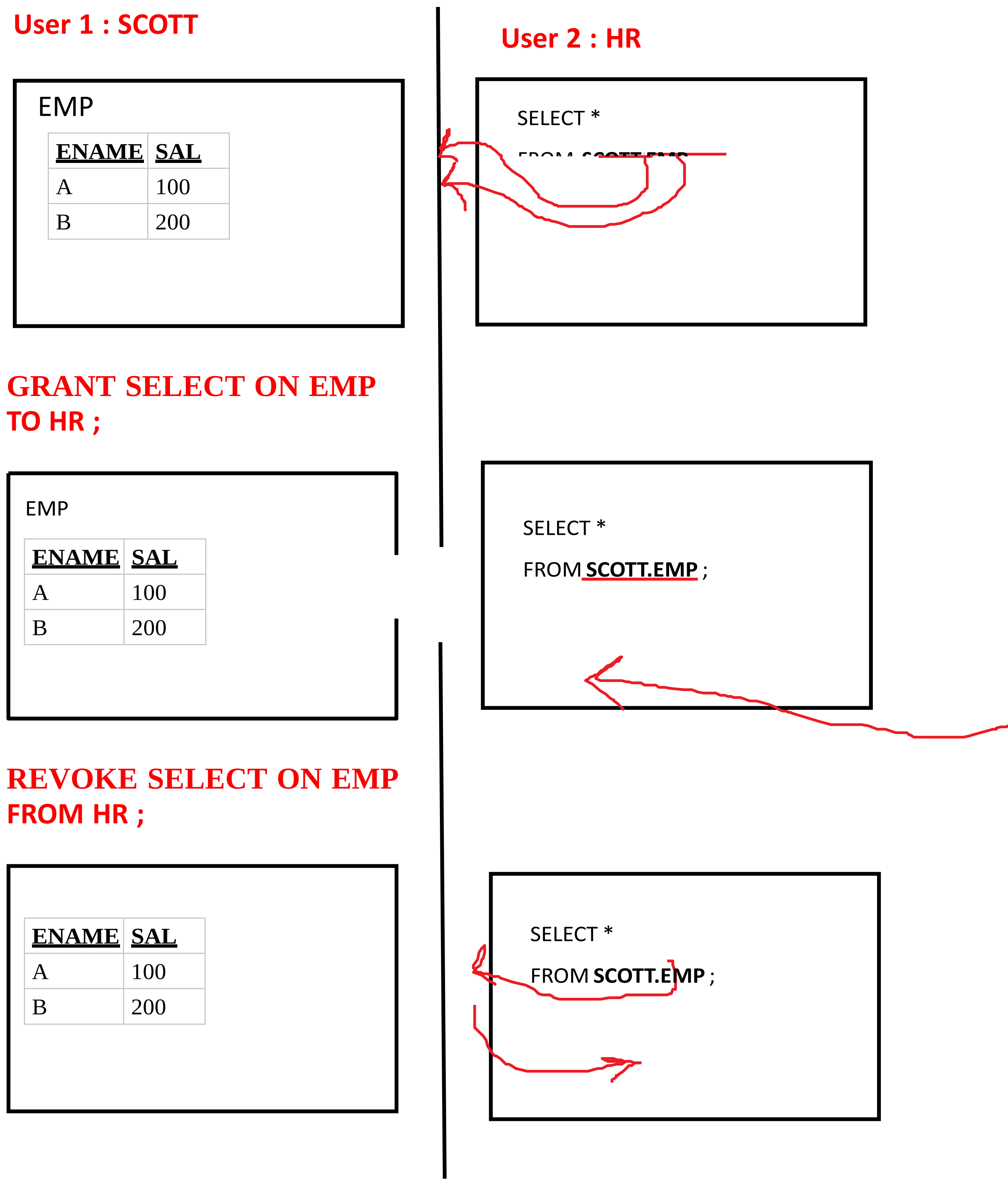
2.REVOKE : THIS STATEMENT IS USED TO TAKE BACK THEPERMISSION FROM THE USER .

SYNTAX: **REVOKE** SQL_STATEMENT
ON TABLE_NAME FROM
USER_NAME ;

Example :

User 1 : SCOTT

User 2 : HR



User 1 : SCOTT

EMP

ENAME	SAL
A	100
B	200

User 2 : HR

SELECT *

~~FROM SCOTT.EMP~~

GRANT SELECT ON EMP TO HR ;

EMP

ENAME	SAL
A	100
B	200

SELECT *

FROM **SCOTT.EMP** ;

REVOKE SELECT ON EMP FROM HR ;

ENAME	SAL
A	100
B	200

SELECT *

FROM **SCOTT.EMP** ;

TRY !!!!

```
SQL> SHOW USER ;
USER is "SCOTT"
SQL> CONNECT

Enter user-name: HR
Enter password: *****
Connected.

SQL> SHOW USER ;
USER is "HR"
```

```
Connected.

SQL> SHOW USER ;
USER is "HR"

SQL> SELECT *

  2  FROM SCOTT.EMP;
FROM SCOTT.EMP

           *

ERROR at line 2:

ORA-00942: table or view does not exist


SQL> CONNECT

Enter  user-name:  SCOTT
Enter password: *****
Connected.

SQL> GRANT SELECT ON EMP TO HR;

Grant succeeded.

SQL>  CONNECT

Enter user-name: HR

Enter password: *****
Connected.

SQL> SELECT *

  2  FROM SCOTT.EMP;
```

```
EMPNO ENAME      JOB              MGR       HIREDATE          SAL       COMM      DEPTNO
-------    ----------     ---------        ----------      ---------------      ----------  ----------   ------------
   7369 SMITH      CLERK           7902       17-DEC-80       800                    20
   7499 ALLEN      SALESMAN        7698       20-FEB-81      1600       300          30
   7521 WARD       SALESMAN        7698       22-FEB-81      1250       500          30
   7566 JONES      MANAGER         7839       02-APR-81      2975                    20
   7654 MARTIN     SALESMAN        7698       28-SEP-81      1250      1400          30
   7698 BLAKE      MANAGER         7839       01-MAY-81      2850                    30
   7782 CLARK      MANAGER         7839       09-JUN-81      2450                    10
   7788 SCOTT      ANALYST         7566       19-APR-87      3000                    20
   7839 KING       PRESIDENT                  17-NOV-81      5000                    10
   7844 TURNER     SALESMAN        7698       08-SEP-81      1500         0          30
   7876 ADAMS      CLERK           7788       23-MAY-87      1100                    20
   7900 JAMES      CLERK           7698       03-DEC-81       950                    30
   7902 FORD       ANALYST         7566       03-DEC-81      3000                    20
   7934 MILLER     CLERK           7782       23-JAN-82      1300                    10
```