SQL COMMANDS

1. DDL - Create, Alter, Rename, Truncate, Drop
2. DML - Insert, Update, Delete
3. DQL - Select
4. DCL - Grant Revoke

DDL

1. Create
2. Alter - Add, Modify, Rename*, Drop*
3. Rename*
4. Truncate
5. Drop*

DDL: Create

Used for creating new table in database

Syntax:

```
Create table <tablename> (<Columnname1><datatype>(<size>), <Columnname2><datatype>(<size>));

Create table <tablename> (
<Columnname1><datatype>(<size>),
<Columnname2><datatype>(<size>)
);
```

Example:

```
Create table Customer(
Cid varchar(10),
Name Char(20),
Gender Char(10),
Age Number(3)
);
```

DDL: Alter

Used for doing modification or changes in preexisting table columns

1. **Add (Add a new Column to pre existing table)**
2. **Modify (Modify datatype, size of pre-existing table or add Not Null, Default Constraint to table)**
3. **Rename (Rename the existing column name of table)**
4. **Drop (Removes the column from the table)**

DDL: Alter - Add

(Add a new Column to pre existing table)

Syntax:

```
Alter table <tablename> add (<new Columnname><datatype>(<size>));

Alter table <tablename> add (
<new Columnname><datatype>(<size>));
```

Example:

```
Alter table Customer add(
Phone number(10),
);
```

DDL: Alter - Modify

(Modify datatype, size of pre-existing table or add Not Null, Default Constraint to table)

Syntax:

```
Alter table <tablename> modify (<Columnname><datatype>(<size>));

Alter table <tablename> modify (
<Columnname><datatype>(<size>));
```

Example:

```
Alter table Customer modify(
name char(30),
Phone number(12)
);
```

DDL: Alter - Rename

(Renames the column name of pre-existing table)

Syntax:

Alter table <tablename> rename column <old column name> to <new column name>;

Alter table <tablename> rename column
<old column name> to <new column name>;

Example:

Alter table Customer rename column name to Username;

 # Only command to use COLUMN Keyword

DDL: Alter - Drop

(Removes the column from the table)

Syntax:

Alter table <tablename> drop(<column name>);

Alter table <tablename> drop(
(<column name>);

Example:

Alter table Customer drop(age);

DDL: Rename

Used for renaming table in database

Syntax:

Rename <old tablename> to <new tablename>;

Example:

Rename Customers to Clients;

Rename Clients to Customers;

 # Only DDL command to not use TABLE Keyword

DDL: Truncate

Removes all data except structure

Syntax:

Truncate table <tablename>;

Example:

Truncate table Customers;

DDL: Drop

Removes all data and structure

Syntax:

Drop table <tablename>;

Example:

Drop table Customers;

DML

1. Insert
2. Update
3. Delete

DML: Insert

Used to Insert data in the table

Syntax:

```
Insert into <tablename> values(value1,value2);
```

Example:

```
Insert into Customer (Cid,Name,Gender,Age) values('E101','Raman','Male',20);

Insert into Customer values('E101','Raman','Male',20);
Insert into Customer values('E102','Daman','Female',22);
Insert into Customer values('E103','Chaman','Male',20);
Insert into Customer values('E104','Harman','Female',27);
Insert into Customer values('E105','Samman','Male',32);
```

```
Select * from Customer;
```

DML: Update

Used to Update the data inside the table

Syntax:

Update <table_name> set <column_name>=<new_value> where <condition>

Example:

Update customer set age=35 where name='Samman';

Update customer set gender='Male' where Eid='E104';

Select * from Customer;

DML: Delete

Used to Delete the rows from the table

Syntax:

Delete from <tablename> where <condition>

Example:

Delete from customer where name='Chaman';

Delete from customer where Eid='E105';

Select * from Customer;

DQL: Select

Used to Query from the table

Syntax:

```
Select <Column_name> from <Table_name>;
Select <Column_name> from <Table_name> where <condition>;
Select * from <Table_name>;
```

Example:

```
Select Name,age from Customer;

Select Name, age, gender from Customer where gender = 'male';

Select * from Customer;
```

DQL: Select

Used to Query from the table

Syntax:

```
Select <Column_name> from <Table_name>;
Select <Column_name> from <Table_name> where <condition>;
Select * from <Table_name>;
```

Example:

```
Select Name,age from Customer;

Select Name, age, gender from Customer where gender = 'male';

Select * from Customer;
```

DQL: Select

Used to Query from the table

- **Find the Customers having Age above 25 years**
- **Find the Customers Age and Gender whose Customerid is E102**
- **Find the Customers Name and Age those are females**
- **Find the detail of customer whose name is Harman**
- **Find the Gender of customer whose age is less than 25 years**
- **Find the Customer id of all the customers those are male**

CONSTRAINTS

1. Not Null
2. Unique
3. Primary Key
4. Foreign Key
5. Check
6. Default

CONSTRAINTS

Constraints can be applied using

- **Create Command**
- **Alter Command**
  - **Using ADD Keyword**
    1. Unique
    2. Check
    3. Primary Key
    4. Foreign Key
  - **Using Modify Keyword**
    1. Not Null
    2. Default

CONSTRAINTS

Constraints with Create Command

- Create table product (pid varchar(10) primary key, pname varchar(20));

- Create table Customer (Cid varchar(10) primary key, name char(20) not null, gender char(10) not null, age number(3) check(age<120), Phone number(10) unique, website char(10) default 'flipkart', productid varchar(10) references product(pid));

Pid	Pname
p1	Sunglass
p2	Earphones
p4	Shoes

PRODUCT TABLE

Cid	Name	Gender	Age	Phone	Website	ProductID
C1	Man	Male	20	90	Flipkart	p2
C2	Aman	Female	100	89	Flipkart	p1
C3	Raman	Female	40	98	Flipkart	p4
C4	Chaman	Male	28	94	Flipkart	p2
C5	Daman	Male	19	78	Flipkart	p2

CUSTOMER TABLE

CONSTRAINTS

Constraints with Create Command

- Create table product (pid varchar(10), pname varchar(20));
  - Alter table product add primary key(pid);

- Create table Customer (Cid varchar(10), name char(20), gender char(10), age number(3), Phone number(10), website char(10), productid varchar(10));

  - Alter table Customer add primary key(Cid);
  - Alter table Customer modify name char(20) not null;
  - Alter table Customer modify gender char(10) not null;
  - Alter table Customer add check(age<120);
  - Alter table Customer add unique(Phone);
  - Alter table Customer modify website char(10) default ‘flipkart’;
  - Alter table Customer add foreign key (productid) references product(pid);

DROP CONSTRAINTS

Constraints Drop Commands

Before dropping constraint you must know constraint name.

Alter table <tablename> drop constraint <constraintname>;

For finding constraint name

i. Either give constraint name while applying constraint to table

Alter table product add constraint productname_unique unique(pname);

ii. Find the system generated constraint name

Select Constraint_name,Constraint_type from all_constraints where table_name='YOUR_TABLE_NAME_IN_CAPITAL_CASE";

DQL KEYWORDS

- **And, In, Or and Between**
- **Like and wildcards**
- **Aggregate Functions (Min, Max, Avg, Sum, Count)**
- **Is Null**
- **Distinct**

And, In, Or and Between KEYWORDS

AND	• Select name from customer where Gender = ‘Male’ and Age = 32;
OR	• Select name from customer where Age = 25 or Age = 32;
IN	• Select name from customer where Age in (25,32);
BETWEEN	• Select name from customer where Age between 20 and 30;

DQL KEYWORDS

% n Characters
_ 1 Characters

LIKE

- Select name from customer where name like 'M%';
- Select cid from customer where cid like '_101';

DQL KEYWORDS Aggregate Functions

Aggregate Functions

- **Min()** — **Select min(age) from customer;**
- **Max()** — **Select max(age) from customer;**
- **Avg()** — **Select avg(age) from customer;**
- **Sum()** — **Select sum(age) from customer;**
- **Count()** — **Select count(age) from customer;**

DQL KEYWORDS

- **Is Null** **Select name from customer where age is null;**
- **Distinct** **Select distinct(name) from customer**

Order By (Use to sort data)

- **Select name from customer order by age**
- **Select name from customer order by age desc**
- **Select name from customer order by (age,gender)**

Group By (Use to group data)

- **Select gender from customer group by gender;**
- **Select avg(age) from customer group by age;**
- **Select count(Cid) from customer group by gender;**
- **select age,count(age) from customer group by age;**

Having (Use to search data inside group)

- **Select count(Cid) from customer group by gender having gender=‘female’;**

- **Select count(Cid),gender from customer group by gender having count(Cid)>2;**

- **select age,count(age) from customer group by age having count(age)>1;**

DDL QUERIES

/* CREATE TABLE */

```
Create table Customer(
Cid varchar(10),
Name Char(20),
Gender Char(10),
Age Number(3)
);
```

/* VIEW STRUCTURE */

```
Desc Customer;
```

/* ALTER STRUCTURE ADD COLUMN IN TABLE */

```
Alter table Customer add(Phone number(10));
```

/* VIEW STRUCTURE */

```
Desc Customer;
```

/* ALTER STRUCTURE CHANGE DATATYPE AND SIZE OF COLUMN IN TABLE */

```
Alter table Customer modify(name char(30),Phone number(12));
```

/* VIEW STRUCTURE */

```
Desc Customer;
```

/* ALTER STRUCTURE RENAME COLUMN IN TABLE */

```
Alter table Customer rename column name to Username;
Desc Customer;
```

/* ALTER STRUCTURE REMOVE COLUMN FROM TABLE */

```
Alter table Customer drop(age);
Desc Customer;
```

DDL QUERIES

/* RENAME TABLE */

Rename Customer to Client;

/* INSERTING ROW/DATA IN TABLE FOR CHECKING TRUNCATE COMMAND */

insert into client values('C1','Raman','Male',9041241209);

/* VIEW DATA INSIDE TABLE */

Select * from client;

/* VIEW STRUCTURE */

Desc Client;

/* TRUNCATE COMMAND */

Truncate table client;

/* VIEW DATA INSIDE TABLE */

Select * from client;

/* VIEW STRUCTURE */

Desc Client;

/* INSERTING ROW/DATA IN TABLE FOR CHECKING DROP COMMAND */

insert into client values('C1','Raman','Male',9041241209);

/* VIEW DATA INSIDE TABLE */

Select * from client;

/* VIEW STRUCTURE */

Desc Client;

/* DROP COMMAND */

Drop table client;

/* VIEW DATA INSIDE TABLE */

Select * from client;

/* VIEW STRUCTURE */

Desc Client;

DML

```
Create table Customer(Cid varchar(10),Name Char(20),Gender Char(10),Age Number(3));

Insert into Customer (Cid,Name,Gender,Age) values('E101','Raman','Male',20);
Insert into Customer values('E102','Daman','Female',22);
Insert into Customer values('E103','Chaman','Male',20);
Insert into Customer values('E104','Harman','Female',27);
Insert into Customer values('E105','Samman','Male',32);

Select * from Customer;

Insert into Customer (Cid,Name,Gender) values('E101','Raman','Male');
Insert into Customer (Cid,Name,Gender,Age) values('E107',NULL,'Male',29);
Insert into Customer (Cid,Gender,Age) values('E107','Male',29);

Update Customer set gender='Female' where Cid='E101';
Update Customer set age=30 where Cid='E105';
Update Customer set age=29, gender='female' where Cid='E103';

Select * from Customer;

Delete from Customer where cid='E105';
Delete from Customer where gender='Male';

insert into customer values('E108','Sharvan','Male',NULL);

Delete from customer where age is NULL;

select * from customer;
```

DQL

```
Create table Customer(Cid varchar(10),Name Char(20),Gender Char(10),Age
Number(3));

Insert into Customer (Cid,Name,Gender,Age) values('E101','Raman','Male',20);

Insert into Customer values('E102','Daman','Female',22);

Insert into Customer values('E103','Chaman','Male',20);

Insert into Customer values('E104','Harman','Female',27);

Insert into Customer values('E105','Samman','Male',32);

Select * from Customer;

/*Find the Customers having Age above 25 years*/

Select * from customer where age>25;

/*Find the Customers Age and Gender whose Customerid is E102*/

Select age,gender from customer where cid='E102';

/*Find the Customers Name and Age those are females */

Select name,age from customer where gender='Female';

/*Find the detail of customer whose name is Harman*/

Select * from customer where name='Harman';

/*Find the Gender of customer whose age is less than 25 years*/

Select name,gender from customer where age<25;

/*Find the Customer id of all the customers those are male*/

Select cid from customer where gender='Male';
```

CONSTRAINTS WITH CREATE TABLE COMMAND

```
Create table product(Pid varchar(10) Primary key,Pname varchar (20));
Desc product;

Create table customer(Cid varchar(10) Primary key,
Name char(10) not null,
Gender char(10) not null,
Age number(3) check(age<120),
Phone number(10) unique,
Website varchar(10) default 'Flipkart',
Productid varchar(10) references product(Pid));

Insert into product values('P2','Sunglasses');
Select * from product;

Insert into customer values('C101','Man','Male',32,980,Default,'P2');
Select * from customer;
```

CONSTRAINTS WITH ALTER TABLE COMMAND

```
Create table product(Pid varchar(10),Pname varchar (20));
Desc product;

Create table customer(Cid varchar(10),
Name char(10),
Gender char(10),
Age number(3),
Phone number(10),
Website varchar(10),
Productid varchar(10));
Desc Customer;

Alter table Customer add primary key(Cid);
Alter table Customer modify name char(20) not null;
Alter table Customer modify gender char(10) not null;
Alter table Customer add check(age<120);
Alter table Customer add unique(Phone);
Alter table Customer modify website char(10) default 'flipkart';

Alter table Customer add foreign key (productid) references product(pid);
Alter table Product add primary key(pid);
Alter table Customer add foreign key (productid) references product(pid);

Insert into product values('P2','Sunglasses');
Select * from product;
Insert into customer values('C101','Man','Male',32,980,Default,'P2');
Select * from customer;
```

Download Joins explanation ppt from:

https://docs.google.com/presentation/d/1qZe8Iv0lemXX4G2S4ZjuiRoJ7Eg3SV-T/edit#slide=id.p1

QUERIES

```
create table customer(id number(10),name varchar2(10),age number(10),address varchar2(10),gender char(10));

insert into customer values(1,'Ramesh',32,'MP','Male');
insert into customer values(2,'Khilan',25,'Delhi','Male');
insert into customer values(3,'Suraj',56,'Chandigarh','Male');
insert into customer values(4,'Nish',45,'Ludhiana','Female');
insert into customer values(5,'Anu',48,'Mumbai','Female');

create table orders(oid number(10),odate date,customer_id number(10),amount number(10));
insert into orders values(105,'28-oct-2014',1,1500);
insert into orders values(104,'26-mar-2015',3,11500);
insert into orders values(108,'28-oct-2016',4,15500);
```

CUSTOMER					
CID	NAME	AGE	GENDER	PHONE	CITY
C1	MAN	22	M	9041	MUMBAI
C2	AMAN	32	F	9888	BANGALORE
C3	RAMAN	45	F	9766	MUMBAI
C4	CHAMAN	28	M	8045	PUNE
C5	HARMAN	23	M	9826	CHANDIGARH

PRODUCTS			
PID	PNAME	PRICE	DESCRIPTION
P1	WATCH	8000	Fossil
P2	EARPHONE	4000	Boat
P3	SUNGLASSES	7000	Dita
P4	PHONES	70000	Apple
P5	SHOES	4500	Mochi
P6	BOTTLE	2000	Tupperware

ORDERS			
ORDER ID	CID	PID	UNITS
O1	C1	P2	2
O2	C3	P3	1
O3	C4	P5	1
O4	C4	P6	1
O5	C5	P3	1
O6	C3	P4	1

Download Joins explanation ppt from:

https://docs.google.com/presentation/d/1qZe8Iv0lemXX4G2S4ZjuiRoJ7Eg3SV-T/edit#slide=id.p1

//Cross Join

SELECT ID, NAME, AMOUNT, ODATE FROM CUSTOMER, orders;

//Inner Join

select id,name,amount,odate from customer inner join orders on customer.id=orders.customer_id;

//Natural Join

select id,name,age,amount from customer natural join orders;

//Equi Join

select id,name,age,amount from customer,orders where customer.id=orders.customer_id;

Download Joins explanation ppt from:

https://docs.google.com/presentation/d/1qZe8Iv0lemXX4G2S4ZjuiRoJ7Eg3SV-T/edit#slide=id.p1

//Outer Join

//Left Join

select id,name,amount,odate from customer left join orders on customer.id=orders.customer_id;

Select id,name,amount,odate,age from customer,orders where customer.id=orders.customer_id(+);

//Right Join

select id,name,amount,odate from customer right join orders on customer.id=orders.customer_id;

Select id,name,amount,odate,age from customer,orders where customer.id(+)=orders.customer_id;

//Full Outer Join

select id,name,amount,odate from customer full outer join orders on customer.id=orders.customer_id;

```
create table customer (CID varchar(10), Cname varchar(10), Age number(3), Gender varchar(10), Phone number(10),City varchar(20));

insert into customer values('C1', 'Man',22,'M',9041,'Mumbai');

insert into customer values('C2', 'Aman',32,'F',9888,'Bangalore');

insert into customer values('C3', 'Raman',45,'F',9766,'Mumbai');

insert into customer values('C4', 'Chaman',28,'M',8045,'Pune');

insert into customer values('C5', 'Harman',23,'M',9826,'Chandigarh');


Create table products(PID varchar(10), Pname varchar(10), Price number(10), Description varchar(20));

insert into products values('P1', 'Watch',8000,'Fossil');

insert into products values('P2', 'Earphones',4000,'Boat');

insert into products values('P3', 'Sunglasses',7000,'Dita');

insert into products values('P4', 'Phones',70000,'Apple');

insert into products values('P5', 'Shoes',4500,'Mochi');

insert into products values('P6', 'Bottle',2000,'Tupperware');


Select * from customer;

Select * from products;


create table orders(OID varchar(10), CID varchar(10), PID varchar(10), Units number(10));

insert into orders values('O1', 'C1','P2',2);

insert into orders values('O2', 'C3','P3',1);

insert into orders values('O3', 'C4','P5',1);

insert into orders values('O4', 'C4','P6',1);

insert into orders values('O5', 'C5','P3',1);

insert into orders values('O6`', 'C3','P4',1);

Select * from orders;
```

```
select cname,age,gender,pname,price from customer,products,orders where customer.cid=orders.cid
and products.pid=orders.pid;

select cname,pname from customer,products,orders where customer.cid=orders.cid and
products.pid=orders.pid and cname='Raman';


select cname,age,gender,units,pid from customer left join orders on customer.cid=orders.cid;

select pname,price,description,units,products.pid from orders right join products on
orders.pid=products.pid;


select cname,age,pid from customer full outer join orders on  customer.cid=orders.cid;



select cname,pname from customer, orders,products where customer.cid=orders.cid and
products.pid=orders.pid and pname='Sunglasses';



select * from customer, orders,products;
```

EID	ENAME	SALARY	DEPTNAME
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	CHAMAN	25000	HR
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

DID	DEPTNAME	NoofEmployees
D1	HR	10
D2	SALES	20
D3	ADVERTISING	10
D4	IT	50

SUBQUERY

EID	ENAME	SALARY	DEPTNAME
E1	**RAM**	**10000**	**HR**
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	**CHAMAN**	**25000**	**HR**
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

DID	DEPTNAME	NoofEmployees
D1	HR	10
D2	SALES	20
D3	ADVERTISING	10
D4	IT	50

Find the name of employee having DID = D1

Select ename from employee where deptname in (Select deptname from department where DID='D1')

JOINS

EID	ENAME	SALARY	DEPTNAME
E1	**RAM**	**10000**	**HR**
E1	RAM	10000	HR
E1	RAM	10000	HR
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E2	SHAM	20000	SALES
E2	SHAM	20000	SALES
E2	SHAM	20000	SALES

DID	DEPTNAME	NoofEmployees
D1	**HR**	**10**
D2	SALES	20
D3	ADVERTISING	10
D4	IT	50
D1	HR	10
D2	SALES	20
D3	ADVERTISING	10
D4	IT	50

Find the name of employee having DID = D1

Select ename from employee,department where employee.deptname=department.deptname and DID='D1';

EID	ENAME	SALARY	DEPARTMENT
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	CHAMAN	25000	HR
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

```
Create table employees (EID varchar(10), Ename varchar(20), Salary number(10), Dept varchar(20));
Insert into employees values('E1','RAM',10000,'HR');
Insert into employees values('E2','SHAM',20000,'Sales');
Insert into employees values('E3','RAMAN',15000,'Sales');
Insert into employees values('E4','CHAMAN',25000,'HR');
Insert into employees values('E5','DAMAN',40000,'ADVERTISING');
Insert into employees values('E6','HARMAN',25000,'ADVERTISING');
Insert into employees values('E7','SAMMAM',35000,'IT');
```

EID	ENAME	SALARY	DEPARTMENT
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	CHAMAN	25000	HR
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

1. **Find the max salary?**
2. **Find the name of the employee earning max salary?**
3. **Find the max salary in each department?**
4. **Find the name of the employee earning max salary in their respective departments?**
5. **Find the department name and amount to whom we are paying the max amount in terms of salary?**
6. **Find the department name over which we are spending most?**

```
Create table employees (EID varchar(10), Ename varchar(20), Salary number(10), Dept varchar(20));

Insert into employees values('E1','RAM',10000,'HR');
Insert into employees values('E2','SHAM',20000,'Sales');
Insert into employees values('E3','RAMAN',15000,'Sales');
Insert into employees values('E4','CHAMAN',25000,'HR');
Insert into employees values('E5','DAMAN',40000,'ADVERTISING');
Insert into employees values('E6','HARMAN',25000,'ADVERTISING');
Insert into employees values('E7','SAMMAM',35000,'IT');

Select * from employees;
/* Find max salary */
Select max(salary) from employees;

/* Find max salary */
Select ename from employees where salary in (Select max(salary) from employees);

/* Find max salary in each department*/
Select dept,max(salary) from employees group by dept;

/* Find name of employee earning max salary in their respective department */
Select ename from employees where (salary,dept) in (select max(salary),dept from employees group by
dept);

/*Find the name of the employee earning max salary in their respective departments?*/
/*Find the department name and amount to whom we are paying the max amount in terms of salary?*/
/*Find the department name over which we are spending most?*/
```

```
Create table employees (EID varchar(10), Ename varchar(20), Salary number(10), Dept varchar(20));

Insert into employees values('E1','RAM',10000,'HR');
Insert into employees values('E2','SHAM',20000,'Sales');
Insert into employees values('E3','RAMAN',15000,'Sales');
Insert into employees values('E4','CHAMAN',25000,'HR');
Insert into employees values('E5','DAMAN',40000,'ADVERTISING');
Insert into employees values('E6','HARMAN',25000,'ADVERTISING');
Insert into employees values('E7','SAMMAM',35000,'IT');

/* Find name of employee earning max salary in their respective department */
Select ename from employees where (salary,dept) in (select max(salary),dept from employees group by dept);

/*Find the department name over which we are spending most?*/
select dept from employees group by dept having sum(salary)=(select max(amount) from (select sum(salary)as amount,dept from employees group by dept));;

/*Find the department name and amount to whom we are paying the max amount in terms of salary?*/
select sum(salary),dept from employees group by dept having sum(salary)=(select max(sum(salary)) from employees group by dept);

/*Find the 2nd Maximum Salary*/
Select max (salary) from employees where salary<(Select max (salary) from employees)
Select max(salary) from employees where salary in (Select salary from employees minus select max(salary) from employees);
```

/*Find the nth Maximum Salary*/

select ename,salary from employees e1 where (1-1)= (select count(distinct(salary)) from employees e2 where e1.salary<e2.salary);

/*Delete the n Redundant Rows*/

Insert into employees values('E6','HARMAN',25000,'ADVERTISING');

Insert into employees values('E7','SAMMAM',35000,'IT');

Insert into employees values('E6','HARMAN',25000,'ADVERTISING');

Insert into employees values('E7','SAMMAM',35000,'IT');

Insert into employees values('E6','HARMAN',25000,'ADVERTISING');

Insert into employees values('E7','SAMMAM',35000,'IT');

Insert into employees values('E6','HARMAN',25000,'ADVERTISING');

Insert into employees values('E7','SAMMAM',35000,'IT');

Insert into employees values('E6','HARMAN',25000,'ADVERTISING');

Insert into employees values('E7','SAMMAM',35000,'IT');

Select * from employees;

Delete from employees where rowid not in (select min(rowid) from employees group by (Eid,Ename,Salary,Dept));

Drop table backup1

/* Create Duplicate Table with same schema but without data*/

Create table backup as select * from employees where 1=0;

Select * from backup;

Desc backup;

/* Create Duplicate Table with same schema but with data*/

Create table backup1 as select * from employees where 1=1;

Select * from Backup1;

EID	ENAME	SALARY	DEPARTMENT
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	CHAMAN	25000	HR
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

```
Create table employees (EID varchar(10), Ename varchar(20), Salary number(10), Dept varchar(20));
Insert into employees values('E1','RAM',10000,'HR');
Insert into employees values('E2','SHAM',20000,'Sales');
Insert into employees values('E3','RAMAN',15000,'Sales');
Insert into employees values('E4','CHAMAN',25000,'HR');
Insert into employees values('E5','DAMAN',40000,'ADVERTISING');
Insert into employees values('E6','HARMAN',25000,'ADVERTISING');
Insert into employees values('E7','SAMMAM',35000,'IT');
```

EID	ENAME	SALARY	DEPARTMENT
E1	RAM	10000	HR
E2	SHAM	20000	SALES
E3	RAMAN	15000	SALES
E4	CHAMAN	25000	HR
E5	DAMAN	40000	ADVERTISING
E6	HARMAN	25000	ADVERTISING
E7	SAMMAN	35000	IT

1. **Find the department name over which we are spending most?**
2. **Find the department name and amount to whom we are paying the max amount in terms of salary?**
3. **Find the 2nd Maximum Salary?**
4. **Find the nth Maximum Salary?**
5. **Delete the n Redundant Rows?**
6. **Create Duplicate Table with same schema but without data?**
7. **Create Duplicate Table with same schema but with data?**

```
select * from employees;

/*Simple View Updatable*/
Create or replace view V1 as select ename,salary from employees;
Select * from V1;

Create or replace view V1 as select ename,salary from employees where salary>30000;
Select * from V1;

insert into V1 values ('Abhishek',50000);
Select * from V1;
select * from employees;

insert into V1 values ('Abhishek',25000);

/*Simple View Read Only*/
Create or replace view V2 as select ename,dept from employees with read only;
select * from V2;

Insert into V2 values('Rahul','IT');

/* Complex view */
create or replace view V3 as select dept from employees group by dept;
Create or replace view V3 as select dept, sum(salary) as salary  from employees group by dept;
select * from V3;
Insert into v3 values('youth',35000);


select * from customer;
```

```
select * from product;
select * from orders;

Create or replace view V5 as (select cname,age,gender,pname,price from customer,product,orders
where customer.cid=orders.cid and product.pid=orders.pid);
select * from V5;

insert into V5 values('nam',22,'F','asssdfg',200);
```

VIEW

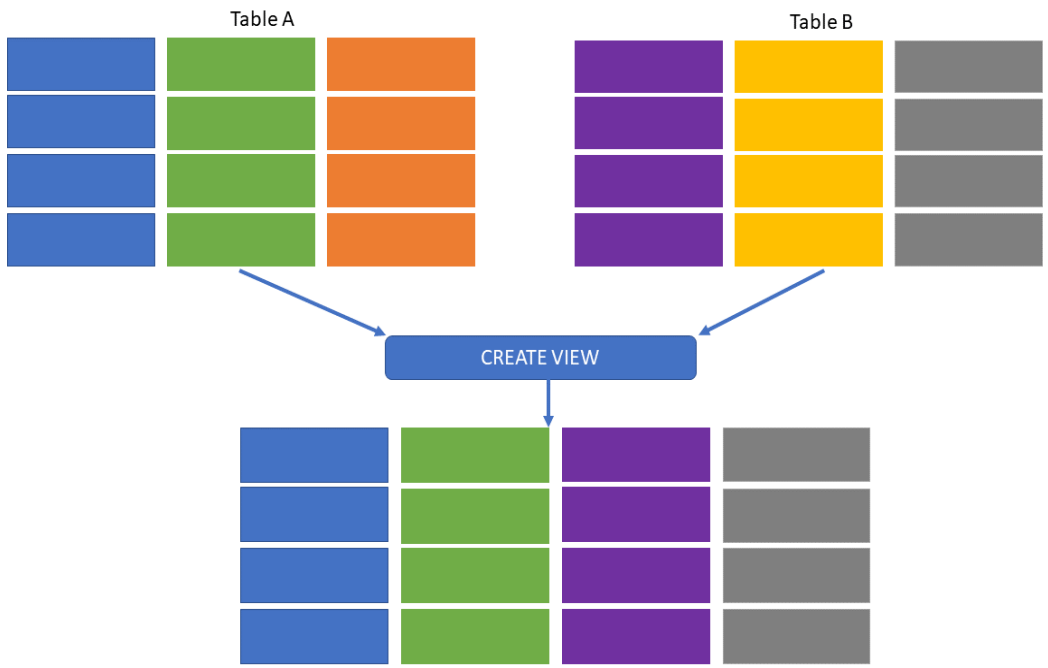


Table A
Table B
CREATE VIEW

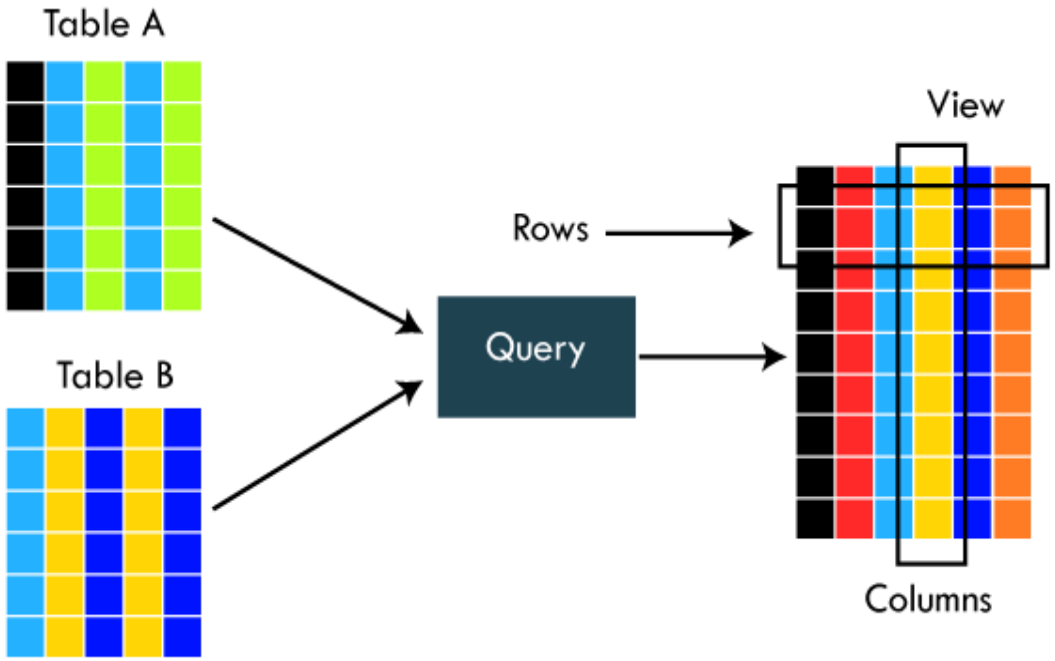
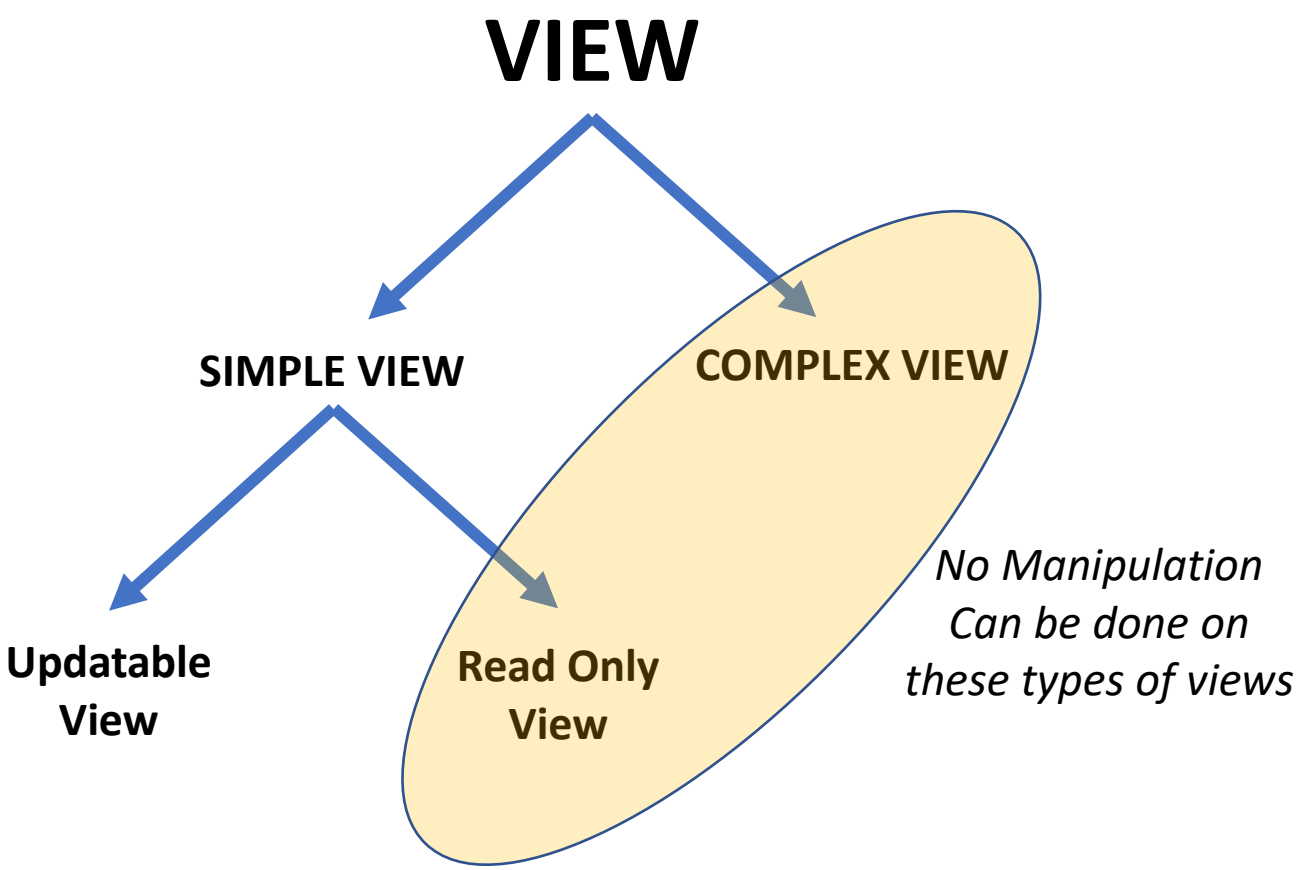


Table A
Table B
Query
Rows
View
Columns


VIEW
SIMPLE VIEW
COMPLEX VIEW
Updatable View
Read Only View
No Manipulation Can be done on these types of views

PL/SQL Block Types

Anonymous

```
DECLARE
……................................
……................................

BEGIN
……..................................
-statements
……..................................

END;
```

PL/SQL Variable Types

- **Variable-name datatype(size);**

```
DECLARE
  a number := 10;
  b number := 20;
  c number;
```

Declaring a Constant

- **PI CONSTANT NUMBER := 3.141592654;**

```
DECLARE
  -- constant declaration
```

Assignment

1. :=

```
A:=10;
Sum:=A+B+C;
```

2. Get value from data base object.

SELECT INTO

```
Select salary into SAL from employee  where empid=12;
```

PL/SQL Literals

Literal Type	Example:
Numeric Literals	050 78 -14 0 +32767 6.6667 0.0 -12.0 3.14159 +7800.00 6E5 1.0E-8 3.14159e0 -1E38 -9.5e-3
Character Literals	'A' '%' '9' ' ' 'z' '('
String Literals	'Hello, world!' 'Tutorials Point' '19-NOV-12'
BOOLEAN Literals	TRUE, FALSE, and NULL.
Date and Time Literals	DATE '1978-12-25'; TIMESTAMP '2012-10-29 12:01:01';

PL/SQL Operators

Operator	Description	Example
+	Adds two operands	A + B will give 15
-	Subtracts second operand from the first	A - B will give 5
*	Multiply both operands	A * B will give 50
/	Divide numerator by de-numerator	A / B will give 2
**	Exponentiation operator, raises one operand to the power of other	A ** B will give 100000

PL/SQL Relational Operators

Operator	Description	Example
=	Checks if the value of two operands is equal or not, if yes then condition becomes true.	(A = B) is not true.
!= <> ~=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

PL/SQL Comparison Operators

Operator	Description	Example
LIKE	The LIKE operator compares a character, string, or CLOB value to a pattern and returns TRUE if the value matches the pattern and FALSE if it does not.	If 'Zara Ali' like 'Z% A_i' returns a Boolean true, whereas, 'Nuha Ali' like 'Z% A_i' returns a Boolean false.
BETWEEN	The BETWEEN operator tests whether a value lies in a specified range. x BETWEEN a AND b means that x >= a and x <= b.	If x = 10 then, x between 5 and 20 returns true, x between 5 and 10 returns true, but x between 11 and 20 returns false.
IN	The IN operator tests set membership. x IN (set) means that x is equal to any member of set.	If x = 'm' then, x in ('a', 'b', 'c') returns boolean false but x in ('m', 'n', 'o') returns Boolean true.
IS NULL	The IS NULL operator returns the BOOLEAN value TRUE if its operand is NULL or FALSE if it is not NULL. Comparisons involving NULL values always yield NULL.	If x = 'm', then 'x is null' returns Boolean false.

PL/SQL Logical Operators

Operator	Description	Example
and	Called logical AND operator. If both the operands are true then condition becomes true.	(A and B) is false.
or	Called logical OR Operator. If any of the two operands is true then condition becomes true.	(A or B) is true.
not	Called logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true then Logical NOT operator will make it false.	not (A and B) is true.

PL/SQL Comments

- A:=5; -- assign value 5 to variable A.
- A:=b+c; /* the value of variable A and B are added and assign to variable A */

PL/SQL Block

```
set serveroutput on;
Declare
        a number(2);
         b number (2);
         c number(2);
Begin
        a:=5;
        b:=2;
        c:=a+b;
    dbms_output.put_line('Sum=' || c);
End;
```

PL/SQL Block

```
set serveroutput on;
Declare
         a number(2);
         b number (2);
         c number(2);
Begin
        a:=&a;
        b:=&b;
        c:=a+b;
    dbms_output.put_line('Sum=' || c);
End;
```

Employee Table (Fetch data from table)

Emp_name	Emp_id	TA	DA	Total	Branch_City
abc	10	1200	1345		Delhi
xyz	12	1100	1200		Mumbai
def	14	1700	1500		Chandigarh

```
Create table emp(Emp_name varchar(10), Emp_id number(10), TA number(10), DA number(10), Total
number(10), Branch_city varchar(10));

Insert into emp values ('abc',10,1200,1345,NULL,'Delhi');
Insert into emp values ('xyz',12,1100,1200,NULL,'Mumbai');
Insert into emp values ('def',14,1700,1500,NULL,'Chandigarh');

Select * from emp;
```

```
Declare
   a number(5);
   b number(5);
   t number(5);

Begin
  select TA,DA into a,b from emp where Emp_id=10;
  t:=a+b;
  update emp set total=t where Emp_id=10;

End;
```

Employee Table (Using %type)

Emp_name	Emp_id	TA	DA	Total	Branch_City
abc	10	1200	1345	2545	Delhi
xyz	12	1100	1200		Mumbai
def	14	1700	1500		Chandigarh

```
Declare
        a emp.TA%TYPE;
        b emp.DA%TYPE;
        t emp.Total%TYPE;

Begin

        Select TA,DA into a,b from emp where Emp_id=12;
        t=a+b;

        Update emp set Total=t where Emp_id=12;

End;

Select * from emp;
```

Employee Table (Using % rowtype)

%ROWTYPE has all properties of **%TYPE** and one additional that we required only one variable to access any number of columns.

Emp_name	Emp_id	TA	DA	Total	Branch_City
abc	10	1200	1345	2545	Delhi
xyz	12	1100	1200	2300	Mumbai
def	14	1700	1500		Chandigarh

```
Declare
    Record  emp%ROWTYPE;

Begin
    Select * into Record from emp where Emp_id=14;
    Record.Total=Record.TA+Record.DA;
    Update emp set Total=Record.Total where Emp_id=14;

End;
```

Conditional / Selection

```
Declare

        Num1 number(2);
        Num2 number(2);

Begin

        Num1:=&Num1;
        Num2:=&Num2;

If Num1>Num2 then

        dbms_output.put_line('greater number is:=' || Num1);
Else
        dbms_output.put_line('greater number is:=' || Num2);

End if;

End;
```

Loop

```
Declare

    I number(2);

Begin
    I:=1;

Loop
    Dbms_output.put_line(I);
    I:=I+1;

Exit when I>10;

End loop;

End;
```

For Loop

```
Declare

Total number(4);
i number(2);

Begin

For i in 1..10
Loop

Total:=2*i;

Dbms_output.put_line('2*'||i||'='||Total);

End loop;

End;
```

PROCEDURES

LOCAL PROCEDURE

As it Local so it executes in the same session due to which

- Initialization of procedure
- Calling of procedure

To be done in same programme

STORED PROCEDURE

As it Stored so it can be stored & can easily executes later on.

- Initialization of procedure
- Calling of procedure

Can be done differently

PROCEDURES

/* Creating and Declaring Procedure */

```
Declare
Num1 number(2);
Num2 number(2);
Mul number(4);

Procedure multiplication (Num1 IN number, Num2 IN number, Mul OUT number) IS

Begin

Mul:= Num1*Num2;

End multiplication;
```

/* Calling Procedure*/

```
Begin
Num1:=&num1;
Num2:=&num2;
Multiplication (num1,num2,mul);
Dbms_output.put_line(mul);
End;
```

Stored Procedures

/* Creating and Declaring Procedure */

```
Create or replace procedure addition (Num1 IN number, Num2 IN number, Sum1 OUT number) IS
Begin
Sum1:= Num1+Num2;
End;
```

/* Calling Procedure*/

```
Declare

    Num1 number(2);
    Num2 number(2);
    Sum1 number(4);

Begin
    Num1:=&num1;
    Num2:=&num2;
    addition(num1,num2,sum1);

Dbms_output.put_line(sum1);

End;
```

Cursor

What is?

A cursor is a work area where the result of a SQL query is stored at server side.

- **Known as active data set**
  - Declare a cursor
  - Open a cursor
  - Read from Cursor
  - Close Cursor

Types of Cursor

Implicit Cursor

Explicit Cursor

- User Defined (Define in Declare Section of PL/SQL Block)

Cursor Attributes	Implicit Cursor Attributes	Explicit Attributes
➢ %ISOPEN	➢ SQL%ISOPEN	➢ *Cursorname*%ISOPEN
➢ %FOUND	➢ SQL%FOUND	➢ *Cursorname* %FOUND
➢ %NOTFOUND	➢ SQL%NOTFOUND	➢ *Cursorname* %NOTFOUND
➢ %ROWCOUNT	➢ SQL%ROWCOUNT	➢ *Cursorname* %ROWCOUNT

Cursor (Implicit)

Write a PL/SQL block to display message that whether a record is updated or not.

```
Begin
        Update emp set Branch_City = ‘Delhi’ where Emp_id=&Emp_id;

If SQL%FOUND  then
Dbms_output.put_line(‘record updated’);
End if;

If SQL%NOTFOUND  then
Dbms_output.put_line(‘record not updated’);
End if;

End;
```

Cursor (Implicit)

Write a PL/SQL block to count the number of rows affected by an update statement.

Emp_name	Emp_id	TA	DA	Total	Branch_City
abc	10	1200	1345	2545	Delhi
xyz	12	1100	1200	2300	Mumbai
def	14	1700	1500	3200	Chandigarh

```
Declare
Num number(2);

Begin

Update Emp set TA =1500 where TA < 1600;
Num:=SQL%ROWCOUNT;
Dbms_output.put_line('total rows affected =' || Num);

End;
```

Cursor (Explicit)

Steps of Execution

- Declare the cursor
- Open the cursor
- Using loop, fetch the data from cursor one row at a time and store in memory variable
- Exit from the loop
- Close the cursor

```
Declare

        Cursor C123 is select Emp_name from emp where Emp_id >11;
        my_name emp.Emp_name%type;
Begin
        Open C123;

        Loop
        Fetch C123 into my_name;
        Exit when C123%NotFound;
        dbms_output.put_line(my_name);
        End loop;

        Close C123;
End;
```

Trigger

What is?

Stored procedures that automatically executed when some event occurs to data base.

- **Events are**

  - Insert - Update - Delete

Parts of Trigger

- Triggering event or statement
- Trigger restriction (Is boolean value true or false)
- Trigger action

Trigger

Create PL/SQL trigger which will tell about the operation performed on database.

```
Create or replace trigger t1
      Before INSERT or UPDATE or DELETE

ON emp

Begin

IF INSERTING then
Dbms_output.put_line("operation performed inserting");

ELSEIF UPDATING  then
Dbms_output.put_line("operation performed Updating");

ELSE
Dbms_output.put_line("operation performed Deletion");

End if;
End;
```

- Insert into emp values ('jki',18,1200,3200,4400,'Meerut');
- Update emp set TA=1400 where Emp_id=18;
- Delete from emp where Emp_id=18;

Trigger

Create PL/SQL trigger which will convert the name of the Employee to uppercase before inserting or updating the name column of Employee database.

```
Create or replace trigger t2
      Before INSERT or UPDATE of NAME

ON emp
For each row

Begin
:NEW.Emp_name:=UPPER(:NEW.Emp_name);

End;
```

- Insert into emp values (‘jki’,18,1200,3200,4400,’Meerut’);
- Update emp set TA=1400 where Emp_id=18;
- Delete from emp where Emp_id=18;

Trigger

Create PL/SQL trigger which will insert the detail of the employee in emp_backup table when particular record is deleted from database.

```
Create or replace trigger t3
      Before delete

On emp
For each row

Begin
Insert into emp_backup values (:OLD.Emp_name,:OLD.Emp_id)
End;
```

- Create table emp_backup (Emp_name varchar(20), Emp_id number(10));

- Delete from emp where Emp_id=18;