



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

FACULTY OF ENGINEERING & TECHNOLOGY

(Formerly SRM University, Under section 3 of UGC Act, 1956)

**S.R.M. NAGAR, KATTANKULATHUR –603 203, KANCHEEPURAM
DISTRICT**

SCHOOL OF COMPUTING

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

Course Code: 18CSC303J

Course Name: Database Management Systems

LAB REPORT

NAME: Arnav Kumar

REG.NO.: RA1911027010040

SECTION: N1

CSE – BIG DATA ANALYTICS

TABLE OF CONTENT

SL.NO	NAME OF EXPERIMENT
1	BASIC SQL COMMANDS
2	CONSTRAINTS FOR HOSPITAL MANAGEMENT SYSTEMS
3	DATABASE FOR A CAR INSURANCE COMPANY
4	USE OF FUNCTIONS LIKE SUM AND COUNT
5	USE OF DIFFERENT TYPE OF FUNCTIONS IN SQL
6	MODIFICATION AND DELETION OF TABLE VALUES
7	IMPLEMENTATION OF JOINS IN SQL
8	FACTORIAL IN PL/SQL
9	INTRODUCTION TO PL/SQL CODES
10	EXECUTE PL/SQL COMMAND TO GIVEN CONDITION
11	MINI PROJECT

Date: 17/1/2022 Ex No:1	Title of the Lab BASIC SQL COMMANDS	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
--	--	--

Aim:- To implement basic commands in SQL like insert, modify select etc.

Description about the problem:-

1. Create a table in SQL for patients and add patient details.
2. Display the inserted table.
3. Select specific rows from the table
4. Altering the tables data
5. Updating the tables values with arithmetic operations
6. Deleting an entry in a table
7. These are the basic commands that are required to perform

Queries :-

```
create table myt2(id_no integer, cname varchar2(50), dob date,ra_nk integer, loc
varchar2(20), contact varchar2(10), section varchar2(5));
```

```
insert into myt2 values(1,'ab','9-jan-2000',1,'a','987654','A1');
```

```
insert into myt2 values(2,'cd','1-jun-2000',2,'b','23456','B1');
```

```
insert into myt2 values(4,'gh','8-mar-2000',4,'d','56432','D1');
```

```
insert into myt2 values(5,'ij','21-oct-2000',5,'e','4365656','E1');
```

```
insert into myt2 values(6,'kl','17-feb-2000',6,'f','7653','F1');
```

```
insert into myt2 values(7,'mn','3-oct-2000',7,'g','674564','G1');
```

```
insert into myt2 values(3,'ef','3-sep-2000',7,'c','7443','C1');
```

```
select * from myt2 ;
```

```
select cname,loc from myt2 ;
```

```
select cname,ra_nk from myt2 where ra_nk>5;
```

```
select * from myt2 where ra_nk<5;
```

```
select * from myt2 where ra_nk<5 and id_no=4;
```

```
alter table myt2 add(gender char);
```

```
update myt2 set gender='M';
```

```
select * from myt2;
```

```
update myt2 set gender='F' where id_no=2;
```

```
select * from myt2;
```

```
alter table myt2 drop column section;
```

```
alter table myt2 modify cname varchar(25);
```

```
desc myt2
```

Output:-

```
create table myt2(id_no integer, cname varchar2(50), dob date,ra_nk integer, loc varchar2(20), contact varchar2(10), section varchar2(5))
```

Table created.

```
insert into myt2 values(1,'ab','9-jan-2000',1,'a','987654','A1')
```

1 row(s) inserted.

```
insert into myt2 values(2,'cd','1-jun-2000',2,'b','23456','B1')
```

1 row(s) inserted.

```
insert into myt2 values(4,'gh','8-mar-2000',4,'d','56432','D1')
```

1 row(s) inserted.

```
select * from myt2
```

ID_NO	CNAME	DOB	RA_NK	LOC	CONTACT	SECTION
5	ij	21-OCT-00	5	e	4365656	E1
6	kl	17-FEB-00	6	f	7653	F1
1	ab	09-JAN-00	1	a	987654	A1
3	ef	03-SEP-00	7	c	7443	C1
2	cd	01-JUN-00	2	b	23456	B1
4	gh	08-MAR-00	4	d	56432	D1
7	mn	03-OCT-00	7	g	674564	G1

7 rows selected.

```
select * from myt2 where ra_nk<5
```

ID_NO	CNAME	DOB	RA_NK	LOC	CONTACT	SECTION
1	ab	09-JAN-00	1	a	987654	A1
2	cd	01-JUN-00	2	b	23456	B1
4	gh	08-MAR-00	4	d	56432	D1

3 rows selected.

```
select cname,loc from myt2
```

CNAME	LOC
ij	e
kl	f
ab	a
ef	c
cd	b
gh	d
mn	g

7 rows selected.

```
select cname,ra_nk from myt2 where ra_nk>5
```

CNAME	RA_NK
kl	6
ef	7
mn	7

3 rows selected.

```
select * from myt2 where ra_nk<5
```

ID_NO	CNAME	DOB	RA_NK	LOC	CONTACT	SECTION
1	ab	09-JAN-00	1	a	987654	A1
2	cd	01-JUN-00	2	b	23456	B1
4	gh	08-MAR-00	4	d	56432	D1

3 rows selected.

```
select * from myt2 where ra_nk<5 and id_no=4
```

ID_NO	CNAME	DOB	RA_NK	LOC	CONTACT	SECTION
4	gh	08-MAR-00	4	d	56432	D1

```
alter table myt2 add(gender char)
```

```
update myt2 set gender='F' where id_no=2
```

1 row(s) updated.

Result: Successfully implemented the commands for insert, create, update, delete and alter.

Date: 31/2/2022 Ex No:2	Title of the Lab CONSTRAINTS FOR HOSPITAL MANAGEMENT SYSTEMS	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
--	---	--

Aim:- Creating a hospital database where patient details are added in a table and modified as required.

Description about the problem:-

- Create a table in SQL for patients and add patient details.
- Display the inserted table.
- Select specific rows from the table
- Altering the tables data
- Updating the tables values with arithmetic operations
- Deleting an entry in a table

These are the basic commands that are required to perform

Queries:-

```
create table hosp(p_id int NOT NULL PRIMARY KEY,p_name varchar(25),gender char,amount int NOT NULL,due_amount int ,CHECK (amount>=500));
```

```
insert into hosp values(1,'absa','M',10000,2000);
```

```
insert into hosp values(2,'bbb','F',2090,0);
```

```
alter table hosp add (account_no int,UNIQUE(account_no));
```

```
update hosp set account_no=132435 where p_id=1;
```

```
update hosp set account_no=132435 where p_id=2;
```

```
select * from hosp;
```

```
update hosp set account_no=232435 where p_id=2;
```

Output:-

```
create table hosp(p_id int NOT NULL PRIMARY KEY,p_name varchar(25),gender char,amount int NOT NULL,due_amount int ,CHECK (amount>=500))
```

Table created.

```
insert into hosp values(1,'absa','M',10000,2000)
```

1 row(s) inserted.

```
insert into hosp values(2,'bbb','F',2090,0)
```

```
alter table hosp add (account_no int,UNIQUE(account_no))
```

Table altered.

```
update hosp set account_no=132435 where p_id=1
```

1 row(s) updated.

```
update hosp set account_no=232435 where p_id=2
```

1 row(s) updated.

```
select * from hosp
```

P_ID	P_NAME	GENDER	AMOUNT	DUE_AMOUNT	ACCOUNT_NO
1	absa	M	10000	2000	132435
2	bbb	F	2090	0	232435

[Download CSV](#)

2 rows selected.

Result:- Successfully implemented the commands for a hospital database where patients data can be added and modified

Date: 7/2/2022 Ex No:3	Title of the Lab DATABASE FOR A CAR INSURANCE COMPANY	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
---	--	--

Aim:- To implement commands to create and find data from the table as required below.

Description about the problem:

Consider the insurance database given below.

1. PERSON (driver_id, name, address) CAR (regno, model, year)
2. PARTICIPATED (driver_id, regno, report_number, damage_amount)
3. Create the above tables by properly specifying the primary keys and foreign keys and enter atleast five tuples for each relation.
4. Update the damage amount for the car with specific regno in the accident with reportnumber 12 to 25000.
5. Add a new car to the database.
6. Find the total number of people who owned BMW cars before 2009
7. List the names of the person whose name contain substring 'LA'.
8. List the driver details who damage amount is between 10000-20000
9. list the person belongs to 'chennai' and 'mumbai'
10. list the year of the car in descending order
11. list the car regno, model, driver id.
12. remove the car with year of manufacture is <2000

Queries:-

```
create table person(driver_id int primary key, d_name varchar(20) not null, address varchar(50) not null);
```

```
create table car(reg_no int primary key, model_name varchar(20) not null, year_p int not null, color varchar(10));
```

```
create table participated(driver_id int not null, reg_no int not null,
                        rep_no int not null, damage_no int not null,
                        foreign key (driver_id) references person(driver_id),
```



```
foreign key (reg_no) references car(reg_no));
```

```
insert into person values(100,'LABhsd','Bangalore');  
insert into person values(200,'ddvLA','Mumbai');  
insert into person values(300,'ddLAd','Chennai');  
insert into person values(400,'abcd','Chennai');  
insert into person values(500,'ablacd','Bangalore');
```

```
insert into car values(001,'BMW',2000,'Red');  
insert into car values(002,'AUDI',2001,'Blue');  
insert into car values(003,'FORD',1999,'Yellow');  
insert into car values(004,'BMW',2005,'White');  
insert into car values(005,'BMW',2003,'Black');
```

```
insert into participated values(100,001,11,14000) ;  
insert into participated values(200,002,12,11000)  
insert into participated values(300,03,13,21000)  
insert into participated values(400,05,45,17000)  
insert into participated values(500,05,63,30000)  
insert into participated values(500,04,54,25000)
```

```
update participated set damage_no=25 where rep_no=12;  
SELECT * FROM participated;
```

```
update participated set damage_no=25000 where rep_no=12;
```

```
select count(year_p) from car where model_name in ('BMW');  
update car SET year_p=2021 where reg_no=2005;  
update car SET year_p=2021 where reg_no=4;
```

```
select count(year_p) from car where model_name='BMW' and year_p<2009;  
alter table car modify year_p int null;
```

Output:-

Statement 1	<code>create table person(driver_id int primary key, d_name varchar(20) not null, address varchar(50) not null)</code>
	Table created.
Statement 2	<code>create table car(reg_no int primary key, model_name varchar(20) not null, year_p int not null,color varchar(10))</code>
	Table created.

Statement
20

`SELECT * FROM person`

DRIVER_ID	D_NAME	ADDRESS
200	ddvLA	Mumbai
500	ablaod	Bangalore
100	LAhbsd	Bangalore
400	abod	Chennai
300	ddLAd	Chennai

5 rows selected.

Statement
21

`SELECT * FROM car`

REG_NO	MODEL_NAME	YEAR_P	COLOR
1	BMW	2000	Red
2	AUDI	2001	Blue
4	BMW	2005	White
3	FORD	1999	Yellow
5	BMW	2003	Black

5 rows selected.

`select count(year_p) from car where model_name in ('BMW')`

COUNT(YEAR_P)
5

<code>update car SET year_p=2021 where reg_no=4</code>
1 row(s) updated.

Statement 34	<code>update participated set damage_no=25 where rep_no=12</code>
	1 row(s) updated.

Statement 64	<code>alter table car modify year_p int null</code>
	Table altered.

Result:- Successfully implemented the commands for the given situation of car insurance company.

Date: 17/2/2022 Ex No:4	Title of the Lab USE OF FUNCTIONS LIKE SUM AND COUNT	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
--	---	--

Aim:- To implement commands to create and find data from the table as required below and use function like SUM, COUNT In the commands

Description about the problem:

1. Create a Product and sales table
2. Find the product which is sold maximum in the year 2020
3. Find the minimum cost of a particular product
4. what is the total sale of a product in a particular year
5. what is average sale of the product in a year?
6. list the number of products available with their name
7. what is the total sales amount done for a particular product?

Queries:-

```
CREATE TABLE p_s1(prod_name varchar(25), sale_year int, total_sales int,co_st int);
```

```
CREATE TABLE p_s2(prod_name varchar(25), sale_year int, total_sales int,co_st int);
```

```
INSERT INTO p_s1 VALUES('p1',2017,900,2000);
```

```
INSERT INTO p_s1 VALUES('p2',2017,500,1500);
```

```
INSERT INTO p_s1 VALUES('p3',2017,700,2100);
```

```
INSERT INTO p_s1 VALUES('p1',2019,1000,2100);
```

```
INSERT INTO p_s1 VALUES('p2',2019,300,1700);
```

```
INSERT INTO p_s1 VALUES('p3',2019,400,2500);
```

```
SELECT * FROM p_s1 WHERE total_sales=(SELECT MAX(total_sales) FROM p_s1 WHERE sale_year=2020);
```

```
SELECT * FROM p_s1 WHERE co_st=(SELECT MIN(co_st) FROM p_s1 WHERE prod_name='p3');
```

```
SELECT MIN(co_st) FROM p_s1 WHERE prod_name='p3';
```

```
SELECT prod_name,MIN(co_st) FROM p_s1 WHERE prod_name='p3';
```

```
SELECT SUM(total_sales) FROM p_s1 WHERE prod_name='p3' AND sale_year=2017;
```

```
SELECT AVG(total_sales) FROM p_s1 WHERE prod_name='p1' AND sale_year=2021;
```

```
SELECT * FROM p_s1;
```

SELECT DISTINCT prod_name FROM p_s1;

SELECT SUM(total_sales) FROM p_s1 WHERE prod_name='p2';

SELECT SUM(co_st) FROM p_s1 WHERE prod_name='p2';

SELECT SUM(co_st) FROM p_s1 WHERE prod_name='p3' AND sale_year=2017;

Output:-

```
SELECT * FROM p_s1 WHERE total_sales=(SELECT MAX(total_sales) FROM p_s1 WHERE sale_year=2020)
```

PROD_NAME	SALE_YEAR	TOTAL_SALES	CO_ST
p1	2020	1400	1800

```
SELECT * FROM p_s1 WHERE co_st=(SELECT MIN(co_st) FROM p_s1 WHERE prod_name='p3')
```

PROD_NAME	SALE_YEAR	TOTAL_SALES	CO_ST
p1	2017	900	2000
p3	2020	700	2000
p2	2021	700	2000

3 rows selected.

```
SELECT * FROM p_s1 WHERE total_sales =(SELECT MIN(total_sales ) FROM p_s1 WHERE prod_name='p3')
```

PROD_NAME	SALE_YEAR	TOTAL_SALES	CO_ST
p3	2019	400	2500

```
SELECT AVG(total_sales) FROM p_s1 WHERE prod_name='p1' AND sale_year=2021
```

AVG(TOTAL_SALES)
1250

```
SELECT DISTINCT prod_name FROM p_s1
```

PROD_NAME
p1
p2
p3

3 rows selected.

```
SELECT SUM(total_sales) FROM p_s1 WHERE prod_name='p2'
```

SUM(TOTAL_SALES)
2500

Result:- Successfully implemented the commands by including functions like SUM and COUNT in the commands

Date: 21/2/2022 Ex No:5	Title of the Lab USE OF DIFFERENT TYPE OF FUNCTIONS IN SQL	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
--	---	--

Aim:- To implement commands to create and find data from the table using the Functions like Numeric, Conversion, Characters etc

Description about the problem:




- Use of functions like last_day(d),next_day(d,day) etc
- Character Function usage where quarries like replace translate are performed
- Conversion Functions like to_char, to_number on a table
- Numeric Functions like Abs(n), Exp(n), Power(m,n) and other
- arithmetic functions as well.

Queries:-

- `select abs(-750) from dual`
- `select ceil(254.559) from dual`
- `select cos(45) from dual`
- `select exp(6) from dual`
- `select power(25,2) from dual`
- `select sqrt(256) from dual`
- `select trunc(to_date('03-sept-2000','dd-mon-yy')) from dual`
- `select to_date('21-oct-2001','dd-mm-yy') from dual`
- `select to_char(sysdate, 'DL') from dual`
- `select initcap('dmbs') from dual`
- `select ltrim('34570DBMS', '34570') from dual`
- `select replace('Computer science AI and ML', 'science', 'Science and Engineering') from dual`
- `select add_months(DATE '1972-01-09', 5) from dual`
- `select last_day(sysdate) from dual`
- `select next_day(sysdate,2) from dual`
- `select months_between(DATE '2022-01-09', DATE '2017-06-01') from dual`

Output:-

Statement 5






select abs(-750) from dual

ABS(-750)
750

Download CSV

Statement 6






select ceil(254.559) from dual

CEIL(254.559)
255

Download CSV

Statement 7






select cos(45) from dual

COS(45)
.5253219888177296960474644048235629189251

Download CSV




Statement 8



select exp(6) from dual

EXP(6)
403.428793492735122608387180543388279609

Statement 9






select power(25,2) from dual

POWER(25,2)
625

Download CSV

Statement 10






select sqrt(256) from dual

SQRT(256)
16

Download CSV

Statement 11






select trunc(to_date('03-sep-2000','dd-mon-yy')) from dual

TRUNC(TO_DATE('03-SEP-2000','DD-MON-YY'))
03-SEP-00

Download CSV

Statement 12






select to_date('21-oct-2001','dd-mm-yy') from dual

TO_DATE('21-OCT-2001','DD-MM-YY')
21-OCT-01

Download CSV

Statement 13



select initcap('dmbs') from dual

INITCAP('DMBS')
Dmbs

Statement 14



```
select ltrim( '34578DBMS', '34578' ) from dual
```

```
LTRIM('34578DBMS','34578')
```

DBMS

[Download CSV](#)

Statement 15



```
select replace( 'Computer science AI and ML', 'science', 'Science and Engineering' ) from dual
```

```
REPLACE('COMPUTERSCIENCEAIANDML','SCIENCE','SCIENCEANDENGINEERING')
```

Computer Science and Engineering AI and ML

[Download CSV](#)

Statement 16



```
select add_months(Date'1972-01-09', 5 ) from dual
```

```
ADD_MONTHS(Date'1972-01-09',5)
```

09-JUN-72

[Download CSV](#)

Statement 17



```
select last_day(sysdate) from dual
```

```
LAST_DAY(SYSDATE)
```

31-MAY-22

[Download CSV](#)

Statement 18



```
select next_day(sysdate,2) from dual
```

```
NEXT_DAY(SYSDATE,2)
```

09-MAY-22

[Download CSV](#)

Statement 19



```
select months_between(Date '2022-01-09', Date '2017-06-01' ) from dual
```

```
MONTHS_BETWEEN(Date'2022-01-09',Date'2017-06-01')
```

55.25806451612903225806451612903225806452

Result:- Successfully implemented the commands by using the Functions like Numeric, Conversion, Characters etc

Date: 3/3/2022 Ex No:6	Title of the Lab MODIFICATION AND DELETION OF TABLE VALUES	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
---	---	--

Aim:- To implement commands to create and find data from the table for the below queries where modification and deletions are performed.

Description about the problem:

- Update the view empv10 by increasing 10% salary of the employees who work as 'CLERK'. Also confirm the modifications in emp table.
- Modify the view empv10 which contains the data empno, ename, job, sal. Add an alias for each column name.
- Using emp table, create a view pay which contains ename, monthly_sal, annual_sal, deptno.
- Create a view dept_stat which contains department no., department name, minimum salary, maximum salary, total salary.
- Execute the following query and then try to delete the row with dept no 20. Now write in words that you understand

Queries:-

```
create table emp(empno int primary key , ename varchar(25),job varchar(25),salary int, deptno int);
```

```
insert into emp values(1,'ABC','CLERK',300000,12);
insert into emp values(2,'DEF','JUDGE',1000000,22);
insert into emp values(3,'GHI','DOCTOR',1500000,10);
insert into emp values(4,'JKL','POLICE',800000,9);
insert into emp values(5,'MNO','ACTOR',2000000,4);
insert into emp values(5,'PQR','DRIVER',400000,8);
insert into emp values(6,'PQR','DRIVER',400000,8);
insert into emp values(7,'STU','ACCOUNTANT',600000,13);
insert into emp values(8,'VWX','COOK',500000,16);
insert into emp values(9,'YZA','MANAGER',1100000,11);
insert into emp values(10,'BCD','HR',2100000,14);
```



```
create table dept(dno int PRIMARY KEY ,dname varchar(25),min_sal int,max_sal int);
```

```
insert into dept values(1,'ABC',10000,100000);
insert into dept values(2,'DEF',100000,500000);
insert into dept values(3,'GHI',50000,250000);
insert into dept values(4,'JKL',200000,700000);
insert into dept values(5,'MNO',500000,1000000);
insert into dept values(6,'PQR',100000,600000);
insert into dept values(7,'STU',700000,1400000);
insert into dept values(8,'VWX',100000,350000);
insert into dept values(9,'YZA',50000,200000);
insert into dept values(10,'BCD',500000,700000);
```

```
update emp10
set salary=salary+(10*salary)
where job='CLERK';
```

```
create or replace view emp10 as
select empno as eno,ename as en,job as j,salary as sal
from emp;
```

```
select * from emp10;
```

```
alter view emp10
add monthly_sal;
```

```
alter table emp
add monthly_sal int;
```

```
update emp
set monthly_sal=salary/12;
```

```
select * from emp;
```

```
create or replace view emp10 as
```

```
select ename,monthly_sal,salary as annual_salary,deptno as deptno  
from emp;
```

```
select * from emp10;
```

```
create view dept_stat as  
select dno as department_no, dname as department_name, min_sal as minimum_salary, max_sal as  
maximun_salary  
from dept;
```

```
alter table dept  
add total_salary int  
;
```

```
update dept  
set total_salary=(min_sal+max_sal)/2;
```

```
select * from dept;
```

```
create or replace view dept_stat as  
select dno as department_no, dname as department_name, min_sal as minimum_salary, max_sal as  
maximun_salary,total_salary  
from dept;
```

Output:-

```
create table emp(empno int primary key , ename varchar(25),job varchar(25),salary int, deptno int)
```

Table created.

```
insert into emp values(1,'ABC','CLERK',300000,12)
```

1 row(s) inserted.

```
insert into emp values(2,'DEF','JUDGE',1000000,22)
```

1 row(s) inserted.

```
insert into emp values(3,'GHI','DOCTOR',1500000,10)
```

```
select * from emp10
```

EMPNO	ENAME	JOB	SALARY	DEPNO
2	DEF	JUDGE	1000000	22
7	STU	ACCOUNTANT	800000	13
9	YZA	MANAGER	1100000	11
1	ABC	CLERK	300000	12
8	VWX	COOK	500000	16
3	GHI	DOCTOR	1500000	10
4	JKL	POLICE	800000	9
5	MNO	ACTOR	2000000	4
6	PQR	DRIVER	400000	8
10	BCD	HR	2100000	14

10 rows selected.

```
update emp10  
set salary=salary+(10*salary)  
where job='CLERK'
```

1 row(s) updated.

```
select * from emp10 where job='CLERK'
```

EMPNO	ENAME	JOB	SALARY	DEPNO
1	ABC	CLERK	3300000	12

```
update emp10  
set salary=salary+(10*salary)  
where job='CLERK'
```

1 row(s) updated.

```
select * from emp10 where job='CLERK'
```

EMPNO	ENAME	JOB	SALARY	DEPNO
1	ABC	CLERK	3300000	12

```
create or replace view emp10 as  
select empno as eno,ename as en,job as j,salary as sal  
from emp
```

View created.

```
t  
> alter table emp  
> add monthly_sal int
```

Table altered.

```
t  
r update emp  
r set monthly_sal=salary/12
```

10 row(s) updated.

```
t  
> select * from emp
```

```
alter table dept  
add total_salary int
```

Table altered.

```
update dept  
set total_salary=(min_sal+max_sal)/2
```

12 row(s) updated.

```
t  
> create view dept_stat as  
> select dno as department_no, dname as department_name, min_sal as minimum_salary, max_sal as maximum_salary  
from dept
```

View created.

```
t  
f alter table dept  
f add total_salary int
```

Table altered.

```
t  
> update dept  
> set total_salary=(min_sal+max_sal)/2
```

12 row(s) updated.

Result:- Successfully implemented commands to create and find data from the table for the below queries where modification and deletions are performed

Date: 3/4/2022 Ex No:7	Title of the Lab IMPLEMENTATION OF JOINS IN SQL	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
---	--	--

Aim:- To practice all joins using emp and dept table simple joins self joins equ joins inner joins outer joins left & right union intersect minus set difference

Description about the problem:

- Create 2 different tables and enter values
- Select some rows only from the table of entered data
- Use Order by function
- Use all kind of join functions starting inner, outer, self-etc
- Joins where we add tables at left or right side of the table
- Union function and intersection too

Queries:-

```
create table employee(empno int primary key,na_me varchar(20),age int ,job_role
varchar(20),officecode int);
```

```
insert into employee values (12,'abc',25,'manager',9);
```

```
insert into employee values (11,'def',22,'helper',9);
```

```
insert into employee values (21,'ghi',21,'hr',7);
```

```
insert into employee values (28,'jjk',28,'hr',7);
```

```
insert into employee values (15,'fdf',26,'sales',6);
```

```
insert into employee values (18,'bgfbg',31,'marketing',9);
```

```
insert into employee values (17,'bgdbg',29,'pr',8);
```

```
create table office(officeno int primary key,officecode int,na__me varchar(15),loc
varchar(10),pincode int);
```

```
insert into office values (1,7,'ggk','B',001)
```

```
insert into office values (2,7,'fgfd','C',002)
```

```
insert into office values (3,7,'dfdfe','A',003)
```

```
insert into office values (4,7,'ddscd','B',003)
```

```
insert into office values (5,6,'dvds', 'C',004)
```

```
insert into office values (6,6,'dsdsv','A',004)
```

```
insert into office values (7,6,'dsdsv','B',005)
```

```
SELECT employee.na_me AS EmployeeName, office.loc AS Location_, office.pincod
FROM employee,office
WHERE employee.officocode= office.officocode
ORDER BY employee.empno;
```

```
SELECT A.job_role, B.na_me,B.age
FROM employee A,employee B
WHERE A.job_role <> B.job_role
AND A.officocode = B.officocode
ORDER BY A.empno;
```

```
SELECT A.empno, B.job_role,B.age
FROM employee A,employee B
WHERE A.job_role <> B.job_role
AND A.officocode = B.officocode
ORDER BY A.empno;
```

```
SELECT employee.NA_ME,employee.JOB_ROLE,office.PINCODE,office.LOC
FROM employee,office
WHERE employee.EMPNO=office.OFFICENO;
```

```
SELECT employee.NA_ME,employee.AGE,employee.JOB_ROLE,office.LOC,office.PINCODE
FROM employee
INNER JOIN office ON employee.OFFICECODE=office.OFFICECODE;
```

```
SELECT employee.NA_ME,employee.JOB_ROLE,office.PINCODE,office.LOC
FROM employee
FULL OUTER JOIN office ON employee.OFFICECODE=office.OFFICECODE;
```

```
SELECT employee.NA_ME,employee.AGE,employee.JOB_ROLE,office.LOC,office.PINCODE
FROM employee
LEFT JOIN office ON employee.OFFICECODE=office.OFFICECODE;
```

```
SELECT employee.NA_ME,employee.JOB_ROLE,office.PINCODE,office.LOC
FROM employee
```

```
RIGHT JOIN office ON employee.OFFICECODE=office.OFFICECODE;
```

```
SELECT OFFICECODE FROM employee
```

```
UNION
```

```
SELECT OFFICECODE FROM office
```

```
ORDER BY OFFICECODE;
```

```
SELECT EMPNO FROM employee
```

```
INTERSECT
```

```
SELECT OFFICENO FROM office;
```

```
SELECT EMPNO FROM employee
```

```
MINUS
```

```
SELECT OFFICENO FROM office;
```

Output:-

```
create table employee(empno int primary key,na_me varchar(20),age int ,job_role varchar(20),officecode int)
```

Table created.

```
create table office(officeno int primary key,officecode int,na__me varchar(15),loc varchar(10),pincode int)
```

Table created.

```
SELECT employee.na_me AS EmployeeName, office.loc AS Location_, office.pincode  
FROM employee,office  
WHERE employee.officecode= office.officecode  
ORDER BY employee.empno
```

EMPLOYEEENAME	LOCATION_	PINCODE
nhgr	A	6
nhgr	C	5
nhgr	B	6
nhgr	B	3
def	B	6
def	A	6
def	C	5

```

SELECT A.job_role, B.na_me, B.age
FROM employee A, employee B
WHERE A.job_role <> B.job_role
AND A.officecode = B.officecode
ORDER BY A.empno

```

JOB_ROLE	NA_ME	AGE
pr	def	22
pr	abc	25
pr	bgfbg	31
helper	nhgr	25
helper	abc	25
helper	bgfbg	31

```

SELECT A.empno, B.job_role, B.age
FROM employee A, employee B
WHERE A.job_role <> B.job_role
AND A.officecode = B.officecode
ORDER BY A.empno

```

EMPNO	JOB_ROLE	AGE
10	helper	22
10	manager	25
10	marketing	31
11	pr	25
11	manager	25

```

SELECT employee.NA_ME, employee.JOB_ROLE, office.PINCODE, office.LOC
FROM employee, office
WHERE employee.EMPNO=office.OFFICENO

```

NA_ME	JOB_ROLE	PINCODE	LOC
nhgr	pr	6	B
def	helper	3	A
abc	manager	3	B
sgs	advertising	7	C

4 rows selected.

```

SELECT employee.NA_ME, employee.AGE, employee.JOB_ROLE, office.LOC, office.PINCODE
FROM employee
INNER JOIN office ON employee.OFFICECODE=office.OFFICECODE

```

NA_ME	AGE	JOB_ROLE	LOC	PINCODE
ghi	21	hr	B	1
ijk	28	hr	B	1
tew	30	hr	B	1
aaa	21	manager	B	1
ghi	21	hr	C	2
ijk	28	hr	C	2
tew	30	hr	C	2
aaa	21	manager	C	2


```
SELECT employee.NA_ME,employee.JOB_ROLE,office.PINCODE,office.LOC
FROM employee
RIGHT JOIN office ON employee.OFFICECODE=office.OFFICECODE
```

NA_ME	JOB_ROLE	PINCODE	LOC
abc	manager	5	C
abc	manager	6	A
abc	manager	6	B
abc	manager	3	B
def	helper	5	C

```
SELECT OFFICECODE FROM employee
UNION
SELECT OFFICECODE FROM office
ORDER BY OFFICECODE
```

OFFICECODE
6
7
8
9

4 rows selected.

```
SELECT EMPNO FROM employee
INTERSECT
SELECT OFFICENO FROM office
```

EMPNO
10
11
12
13

4 rows selected.

```
SELECT EMPNO FROM employee
MINUS
SELECT OFFICENO FROM office
```

EMPNO
15
16
17
18
20
21

Result:- Successfully practiced all joins using emp and dept table simple joins self joins equ joins inner joins outer joins left & right union intersect minus set difference.

Date: 7/4/2022 Ex No:8	Title of the Lab INTRODUCTION TO PL/SQL CODES	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
---	--	--

Aim:- Implement the given PL/SQL codes in SQL.

Description about the problem:

- Get the data for specific conditions
- Implement Maximum code
- Compare the values of variables and print the maximum variable

Queries:-

```
Declare
a number;
b number;
c number;
Begin
dbms_output.put_line('Enter a:');
a:=&a;
dbms_output.put_line('Enter b:');
b:=&b;
dbms_output.put_line('Enter c:');
c:=&c;
if (a>b) and (a>c) then
dbms_output.putline('A is Maximum');
elsif (b>a) and (b>c) then
dbms_output.putline('B is Maximum');
else
dbms_output.putline('C is Maximum');
end if;
End;
```

Output:-

```
Declare
  a number;
  b number;
  c number;
Begin
  dbms_output.put_line('Enter a:');
  a:=21;
  dbms_output.put_line('Enter b:');
  b:=4;
  dbms_output.put_line('Enter c:');
  c:=7;
  if (a>b) and (a>c) then
    dbms_output.put_line('A is Maximum');
  elsif (b>a) and (b>c) then
    dbms_output.put_line('B is Maximum');
  else
    dbms_output.put_line('C is Maximum');
  end if;
End;
```

```
Statement processed.
Enter a:
Enter b:
Enter c:
A is Maximum
```

Result:- Successfully implemented PL/SQL for given codes.

Date: 14/4/2022 Ex No:9	Title of the Lab FACTORIAL IN PL/SQL	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
--	---	--

Aim:- To implement commands to create and find data from the table as required below.

Description about the problem:

- Write a pl/SQL program for finding the factorial of a number
- The factorial of a number is the product of all the integers from 1 to that number. For example, the factorial of 7 is $1*2*3*4*5*6*7 = 5040$.
- Factorial is not defined for negative numbers, and the factorial of zero is one, $0! = 1$

Queries:-

```
declare
```

```
n number := 7;
```

```
fact number := 1;
```

```
temp number;
```

```
begin
```

```
temp :=n;
```

```
while( temp>0 )
```

```
loop
```

```
fact := fact*temp;
```

```
temp := temp-1;
```

```
end loop;
```

```
dbms_output.put_line('factorial of the ' || n || ' is ' || fact);
```

```
end;
```

Output:-

```
declare
n number := 7;
fact number := 1;
temp number;

begin

temp := n;

while( temp>0 )
loop
fact := fact*temp;
temp := temp-1;
end loop;

--
dbms_output.put_line('factorial of the '|| n || ' is ' || fact);
end;
```

```
Statement processed
factorial of the 7 is 5040
```

Result:- Successfully implemented PL/SQL code to find factorial of a number.

Date: 21/4/2022 Ex No:10	Title of the Lab EXECUTE PL/SQL COMMAND TO GIVEN CONDITION	Name: Arnav Kumar Registration Number: RA1911027010040 Section:N1 Lab Batch:2 Day Order:1
---	---	--

Aim:- Write PL/SQL program for the given condition.

Description about the problem:

Write a PL/SQL procedure to fetch the employee salary details and update 10% allowances for employees' salaries > 20000 and < 50000.

Queries:-

```
create table emp_sal( emp_id number(6),emp_name varchar(20), dept varchar(20),salary
int);
```

```
insert into emp_sal values (10,'Seb','HR',50000);
```

```
insert into emp_sal values (11,'Lance','Sales',30000);
```

```
insert into emp_sal values (12,'Lawrence','Admins',90000);
```

```
insert into emp_sal values (13,'George','HR',40000);
```

```
insert into emp_sal values (14,'Toto','Admins',33000);
```

```
insert into emp_sal values (15,'Lewis','Marketing',35000);
```

```
insert into emp_sal values (16,'Valterri','Sales',29000);
```

```
select * from emp_sal;
```

```
DECLARE
```

```
    CURSOR employee_cur IS
```

```
        SELECT emp_id,salary
```

```
        FROM    emp_sal
```

```
        FOR UPDATE;
```

```
    incr_sal NUMBER;
```

```
BEGIN
```

```
    FOR employee_rec IN employee_cur LOOP
```

```
        IF employee_rec.salary < 50000 AND employee_rec.salary > 20000 THEN
```

```
            incr_sal := .10;
```

```
ELSE
```

```
    incr_sal := 0;
```

```
END IF;
```

```
UPDATE emp_sal
```

```
SET    salary = salary + salary * incr_sal
```

```
WHERE  CURRENT OF employee_cur;
```

```
END LOOP;
```

```
END;
```

```
/
```

```
select * from emp_sal;
```

Output:-

```
create table emp_sal( emp_id number(6),emp_name varchar(20), dept varchar(20),salary int)
```

```
Table created.
```

```
desc emp_sal
```

TABLE EMP_SAL

Column	Null?	Type
EMP_ID	-	NUMBER(6,0)
EMP_NAME	-	VARCHAR2(20)
DEPT	-	VARCHAR2(20)
SALARY	-	NUMBER

[Download CSV](#)

4 rows selected.

```
select * from emp_sal
```

EMP_ID	EMP_NAME	DEPT	SALARY
10	Seb	HR	50000
11	Lance	Sales	33000
12	Lawrence	Admins	90000
13	George	HR	44000
14	Toto	Admins	36300
15	Lewis	Marketing	38500
16	Valterri	Sales	31900

[Download CSV](#)

7 rows selected.

Result:- Successfully implemented PL/SQL code and found the output.