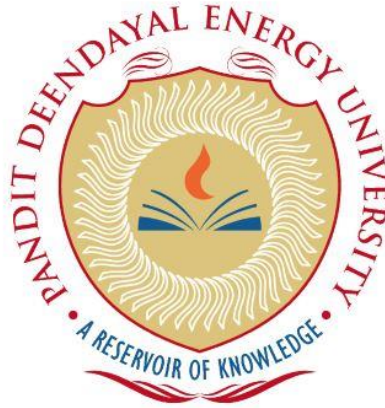


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF TECHNOLOGY**

**PANDIT DEENDAYAL ENERGY UNIVERSITY**

**SESSION 2022-23**



**SUBMITTED BY**

**NAME : Padshala Smit Jagdishbhai**  
**ROLL NO. : 21BCP187**  
**DIVISION : 3**  
**COURSE NAME : DBMS LAB**  
**COURSE CODE : 20CP208P**

**SUBMITTED TO**

**Dr. Sonam Nahar**

**Assistant Professor**

**Department of Computer Science and Engineering**

**Pandit Deendayal Energy University**

**INDEX**

<b>Exp. No.</b>	<b>Title of lab work</b>	<b>Date of lab work</b>	<b>Page</b>	<b>Signature</b>
<b>1</b>	DDL (Data Definition Language) commands	10/01/23	3	
<b>2</b>	DML commands with constraints	17/01/23	5	
<b>3</b>	DDL (Data Definition Language) commands with Data Constraints	24/01/23	8	
<b>4</b>	DDL (Data Definition Language) commands with Data Constraints	31/01/23	11	
<b>5&amp;6</b>	Use of Inbuilt functions and relational algebra operation	07/02/23	13	
<b>7&amp;8</b>	Nested SQL queries or Subqueries	17/02/23	17	
<b>9</b>	Group by & having clause	14/03/23	22	
<b>10</b>	Joins in SQL	22/03/23	25	

## Experiment 1: - DDL (Data Definition Language) command

**Objective:** To understand the concept of designing issue related to the database with creating, populating the tables

### SQL PROGRAM: -

```
create database dbms_lab;
```

```
use dbms_lab;
```

```
create table CLIENT_MASTER(ClientNO varchar(6), NAME varchar(20), CITY varchar(15),  
PINCODE integer, STATE varchar(15), BALDUE decimal (10,2));
```

```
create table PRODUCT_MASTER(PRODUCTNO varchar(6), DESCRIPTION Varchar(15),  
PROFITPERCENT Decimal (4,2) ,UNITMEASURE Varchar (10),QTYONHAND  
Integer,REORDERL_VL Integer,SELLPRICE Decimal(8,2),COSTPRICE Decimal(8,2));
```

```
create table SALESMAN_MASTER(SALESMANNO Varchar(6), SALESMANNAME  
Varchar(20),ADDRESS1 Varchar(30),ADDRESS2 Varchar(30) , CITY Varchar(20) , PINCODE  
Integer ,STATE Varchar(20) , SALAMT Real, TGTTOGET Decimal , YTDSALES Double(6,2) ,  
REMARKS Varchar(60));
```

```
insert into CLIENT_MASTER values
```

```
("C00001","Ivan bayross" , "Mumbai", 400054," Maharashtra", 15000),  
("C00002"," Mamta muzumdar" , "Madras", 780001," Tamil nadu", 0),  
("C00003","Chhaya bankar" , "Mumbai", 400057," Maharashtra", 5000),  
("C00004","Ashwini joshi" , "Bangalore ",560001 , " Karnataka", 0),  
("C00005","Hansel colaco" , "Mumbai", 400060," Maharashtra", 2000),  
("C00006","Deepak sharma" , "Mangalore", 560050," Karnataka", 0);
```

```
insert into PRODUCT_MASTER values
```

```
("P00001","T-Shirt" , 5 , "Piece" , 200 , 50 , 350 , 250),  
("P0345","Shirt" , 6 , "Piece" , 150 , 50 , 500 , 350),  
("P06734","Cotton jeans" , 5 , "Piece" , 100 , 20 , 600 , 450),  
("P07865","Jeans" , 5 , "Piece" , 100 , 20 , 750 , 500),  
("P07868","Trousers" , 2 , "Piece" , 150 , 50 , 850 , 550),  
("P07885","Pull Overs" , 2.5 , "Piece" , 80 , 30 , 350*2 , 450),  
("P07965","Denim jeans" , 4 , "Piece" , 100 , 40 , 350 , 250),  
("P07975","Lycra tops" , 5 , "Piece" , 70 , 30 , 300 , 175),
```

```
("P08865","Skirts" , 5 , "Piece" , 75 , 30 , 450 , 300);
```

insert into SALESMAN\_MASTER values

```
("S00001","Aman", "A/14", "Worli", "Mumbai", 400002,"Maharashtra",3000,100,50,"Good"),
("S00001","Omkar", "65", "Nariman", "Mumbai",
400001,"Maharashtra",3000,200,100,"Good"),
("S00001","Raj", "P-7", "Bandra", "Mumbai", 400032,"Maharashtra",3000,200,100,"Good"),
("S00001","Ashish", "A/5", "Jihu", "Mumbai", 400044,"Maharashtra",3500,200,150,"Good");
```

## TABLES: -

### CLIENT\_MASTER: -

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	ClientNO	NAME	CITY	PINCODE	STATE	BALDUE	
▶	C00001	Ivan bayross	Mumbai	400054	Maharashtra	15000.00	
	C00002	Mamta muzumdar	Madras	780001	Tamil nadu	0.00	
	C00003	Chhaya bankar	Mumbai	400057	Maharashtra	5000.00	
	C00004	Ashwini joshi	Bangalore	560001	Karnataka	0.00	
	C00005	Hansel colaco	Mumbai	400060	Maharashtra	2000.00	
	C00006	Deepak sharma	Mangalore	560050	Karnataka	0.00	

### PRODUCT\_MASTER: -

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IA

	PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERL_VL	SELLPRICE	COSTPRICE
▶	P00001	T-Shirt	5.00	Piece	200	50	350.00	250.00
	P0345	Shirt	6.00	Piece	150	50	500.00	350.00
	P06734	Cotton jeans	5.00	Piece	100	20	600.00	450.00
	P07865	Jeans	5.00	Piece	100	20	750.00	500.00
	P07868	Trousers	2.00	Piece	150	50	850.00	550.00
	P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
	P07965	Denim jeans	4.00	Piece	100	40	350.00	250.00

### SALESMAN\_MASTER: -

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IA

	SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTGET	YTDSALES	REMARKS
▶	S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000	100	50.00	Good
	S00001	Omkar	65	Nariman	Mumbai	400001	Maharashtra	3000	200	100.00	Good
	S00001	Raj	P-7	Bandra	Mumbai	400032	Maharashtra	3000	200	100.00	Good
	S00001	Ashish	A/5	Jihu	Mumbai	400044	Maharashtra	3500	200	150.00	Good

## Experiment 2: - DML (Data Manipulation Language) command with constraints

**Objective:** - To understand the concept of different DML commands.

### SQL PROGRAM: -

```
use dbms_lab;
```

```
select Name from CLIENT_MASTER;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Name				
▶	Ivan bayross			
	Mamta muzumdar			
	Chhaya bankar			
	Ashwini joshi			
	Hansel colaco			
	Deepak sharma			

```
select * from CLIENT_MASTER;
```

Result Grid

Filter Rows:

Export:

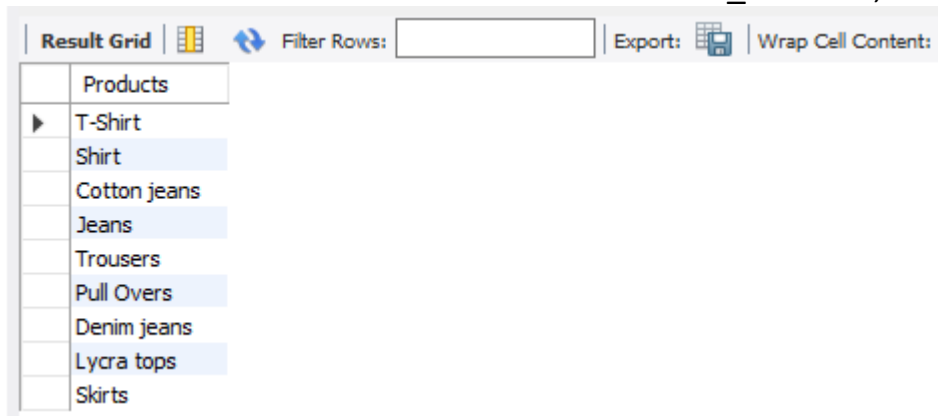
Wrap Cell Content:

	ClientNO	NAME	CITY	PINCODE	STATE	BALDUE
▶	C00001	Ivan bayross	Mumbai	400054	Maharashtra	15000.00
	C00002	Mamta muzumdar	Madras	780001	Tamil nadu	0.00
	C00003	Chhaya bankar	Mumbai	400057	Maharashtra	5000.00
	C00004	Ashwini joshi	Bangalore	560001	Karnataka	0.00
	C00005	Hansel colaco	Mumbai	400060	Maharashtra	2000.00
	C00006	Deepak sharma	Mangalore	560050	Karnataka	0.00

```
select Name, CITY , STATE from CLIENT_MASTER;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Name		CITY	STATE	
▶	Ivan bayross	Mumbai	Maharashtra	
	Mamta muzumdar	Madras	Tamil nadu	
	Chhaya bankar	Mumbai	Maharashtra	
	Ashwini joshi	Bangalore	Karnataka	
	Hansel colaco	Mumbai	Maharashtra	
	Deepak sharma	Mangalore	Karnataka	

select DESCRIPTION as Products from PRODUCT\_MASTER;



The screenshot shows a database query result grid. The header row is labeled 'Products'. The data rows list various clothing items: T-Shirt, Shirt, Cotton jeans, Jeans, Trousers, Pull Overs, Denim jeans, Lycra tops, and Skirts. The grid includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' option.

Products
T-Shirt
Shirt
Cotton jeans
Jeans
Trousers
Pull Overs
Denim jeans
Lycra tops
Skirts

select Name from CLIENT\_MASTER where CITY= "Mumbai";



The screenshot shows a database query result grid. The header row is labeled 'Name'. The data rows list three names: Ivan bayross, Chhaya bankar, and Hansel colaco. The grid includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' option.

Name
Ivan bayross
Chhaya bankar
Hansel colaco

select SALESMANNAME from SALESMAN\_MASTER where SALAMT=3000;



The screenshot shows a database query result grid. The header row is labeled 'SALESMANNAME'. The data rows list three names: Aman, Omkar, and Raj. The grid includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' option.

SALESMANNAME
Aman
Omkar
Raj

update CLIENT\_MASTER set CITY="Bangalore" where ClientNO ="C00005";

update CLIENT\_MASTER set BALDUE=1000 where ClientNO ="C00001";

update PRODUCT\_MASTER set CostPrice=1000 where DESCRIPTION  
="Trousers";

update SALESMAN\_MASTER set CITY="Pune";

delete from SALESMAN\_MASTER where SALAMT=3500;

delete from PRODUCT\_MASTER where QTYONHAND=100;

delete from CLIENT\_MASTER where STATE="Tamil nadu";

alter table CLIENT\_MASTER add column Telephone int(10);

alter table PRODUCT\_MASTER MODIFY COLUMN SELLPRICE decimal (10,2);

Result Grid     Filter Rows: <input type="text"/>   Export:    Wrap Cell Content:							
	ClientNO	NAME	CITY	PINCODE	STATE	BALDUE	Telephone
▶	C00001	Ivan bayross	Mumbai	400054	Maharashtra	1000.00	NULL
	C00002	Mamta muzumdar	Madras	780001	Tamil nadu	0.00	NULL
	C00003	Chhaya bankar	Mumbai	400057	Maharashtra	5000.00	NULL
	C00004	Ashwini joshi	Bangalore	560001	Karnataka	0.00	NULL
	C00005	Hansel colaco	Bangalore	400060	Maharashtra	2000.00	NULL
	C00006	Deepak sharma	Mangalore	560050	Karnataka	0.00	NULL

drop table CLIENT\_MASTER;

alter table SALESMAN\_MASTER rename sman\_mast;

select \* from PRODUCT\_MASTER;

Result Grid     Filter Rows: <input type="text"/>   Export:    Wrap Cell Content:								
	PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNITMEASURE	QTYONHAND	REORDERL_VL	SELLPRICE	COSTPRICE
▶	P00001	T-Shirt	5.00	Piece	200	50	350.00	250.00
	P0345	Shirt	6.00	Piece	150	50	500.00	350.00
	P07868	Trousers	2.00	Piece	150	50	850.00	1000.00
	P07885	Pull Overs	2.50	Piece	80	30	700.00	450.00
	P07975	Lycra tops	5.00	Piece	70	30	300.00	175.00
	P08865	Skirts	5.00	Piece	75	30	450.00	300.00

Select \* from sman\_mast;

Result Grid											Filter Rows:	Export:	Wrap Cell Content:
	SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTGET	YTDSALES	REMARKS		
▶	S00001	Aman	A/14	Worli	Pune	400002	Maharashtra	3000	100	50.00	Good		
	S00001	Omkar	65	Nariman	Pune	400001	Maharashtra	3000	200	100.00	Good		
	S00001	Raj	P-7	Bandra	Pune	400032	Maharashtra	3000	200	100.00	Good		

**Objective:** - To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key

```
use dbms_lab;
```

```
insert into CLIENT_MASTER_1 VALUES
('C00001','Ivan bayross',null,null,'MUMBAI',400054,'MAHARASHTRA',15000),
('C00002','MAMTA MAZUMDAR',NULL,NULL,'MADRAS',780001,'TAMIL
NADU',0),
('C00003','CHHAYA
BANKAR',null,null,'MUMBAI',400057,'MAHARASHTRA',5000),
('C00004','ASHWINI JOSHI',null,null,'BANGALORE',560001,'KARNATAKA',0),
('C00005','HANSEL
COLACO',null,NULL,'MUMBAI',400060,'MAHARASHTRA',2000),
('C00006','DEEPAK
SHARMA',NULL,NULL,'MANGALORE',560050,'KARNATAKA',0);
```

```
select * from CLIENT MASTER 1;
```

[illegible]



```
create TABLE PRODUCT_MASTER_1(PRODUCTNO varchar(6) PRIMARY KEY,
check(PRODUCTNO like "P%"),DESCRIPTION varchar(15) NOT
NULL,PROFITPERCENT DECIMAL(4,2) NOT NULL,UNIT_MEASURE varchar(10) not
null,qtyonhand int(8) not null,reorderl_vl int(8) not null,sell_price decimal(8,2)
not null,cost_price decimal(8,2) not null);
```

```
insert into PRODUCT_MASTER_1 values
('P00001','TSHIRT',5,'PIECE',200,50,350,250),
('P0345','SHIRTS',6,'PIECE',150,50,500,350),
('P06734','COTTON JEANS',5,'PIECE',100,20,600,450),
('P07865','JEANS',5,'PIECE',100,20,750,500),
('P07868','TROUSERS',2,'PIECE',150,50,850,550),
('P07885','PULL OVERS',2.5,'PIECE',80,30,700,450),
('P07965','DENIM JEANS',4,'PIECE',100,40,350,250),
('P07975','LYCRA TOPS',5,'PIECE',70,30,300,175),
('P08865','SKIRTS',5,'PIECE',75,30,450,300);
```

```
select * from PRODUCT_MASTER_1;
```

Result Grid								
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:		
	PRODUCTNO	DESCRIPTION	PROFITPERCENT	UNIT_MEASURE	qtyonhand	reorderl_vl	sell_price	cost_price
▶	P00001	TSHIRT	5.00	PIECE	200	50	350.00	250.00
	P0345	SHIRTS	6.00	PIECE	150	50	500.00	350.00
	P06734	COTTON JEANS	5.00	PIECE	100	20	600.00	450.00
	P07865	JEANS	5.00	PIECE	100	20	750.00	500.00
	P07868	TROUSERS	2.00	PIECE	150	50	850.00	550.00
	P07885	PULL OVERS	2.50	PIECE	80	30	700.00	450.00
	P07965	DENIM JEANS	4.00	PIECE	100	40	350.00	250.00
	P07975	LYCRA TOPS	5.00	PIECE	70	30	300.00	175.00
	P08865	SKIRTS	5.00	PIECE	75	30	450.00	300.00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
create table SALESMAN_MASTER_1(SALESMANNO varchar(6) primary key,
check (SALESMANNO like "S%"),SALESMANNAME varchar(20) NOT
NULL,ADDRESS_1 varchar(20) not null,ADDRESS_2 varchar(30),CITY
varchar(20),PINCODE int(8),STATE varchar(20),SALAMT real(8,2)
check(SALAMT>0),TGTTTOGET decimal(6,2) not null
check(TGTTTOGET>=0),YTDSALES double(6,2) not null,REMARKS varchar(60));
```

```
insert into SALESMAN_MASTER_1 values
```

```
select * from SALESMAN_MASTER_1;
```

[illegible]

## Experiment 4: - DDL (Data Definition Language) commands with Data Constraints

**Objective:** - To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key

### SQL PROGRAM: -

```
-- EXP-4 31/01/23
use dbms_lab;
create table AUTHOR(Author_ID varchar(5) primary key, Lastname text(15) NOT
NULL, Firstname text(15) NOT NULL, Email text(40), City text(15), Country
varchar(15));

create table BOOK(Book_ID varchar(5) primary key , check(Book_ID like 'B%'),
Book_Title text NOT NULL, Copies int(2));

create table AUTHOR_LIST(Author_ID varchar(5) references
AUTHOR(Author_ID), Book_ID varchar(5) references BOOK(Book_ID), Role
text(15));

insert into AUTHOR values
("A01","Sarah", "Buchman", "sarahbuchman7@gmail.com","Bronx","USA"),
("A2","Hallie","Hull","hallie@gmail.com","San Francisco","USA"),
("A3","Srushti","Deshmukh","srushtijd7@gmail.com","Bopal","India"),
("A4","Morgan","Housel","morgan00@gmail.com","Alexandria","USA");

insert into BOOK values
("B01","Book1","5"),
("B02","Book2","9"),
("B03","Book3","3"),
("B04","Book4","10");

insert into AUTHOR_LIST values
("A01","B01","Author"),
("A02","B02","Co-Author"),
("A03","B03","Author"),
```

```
("A04","B04","Reviewer");
```

```
alter table AUTHOR_LIST add column publisher text(30);
```

```
select * from AUTHOR;
```

	Author_ID	Lastname	Firstname	Email	City	Country
▶	A01	Sarah	Buchman	sarahbuchman7@gmail.com	Bronx	USA
	A2	Hallie	Hull	hallie@gmail.com	San Francisco	USA
	A3	Srushti	Deshmukh	srushtjd7@gmail.com	Bopal	India
	A4	Morgan	Housel	morgan00@gmail.com	Alexandria	USA
*	NULL	NULL	NULL	NULL	NULL	NULL

```
select * from BOOK;
```

	Book_ID	Book_Title	Copies
▶	B01	Book1	5
	B02	Book2	9
	B03	Book3	3
	B04	Book4	10
*	NULL	NULL	NULL

```
select * from AUTHOR_LIST;
```

	Author_ID	Book_ID	Role	publisher
▶	A01	B01	Author	NULL
	A02	B02	Co-Author	NULL
	A03	B03	Author	NULL
	A04	B04	Reviewer	NULL

## Experiment 5, 6: - Use of Inbuilt functions and relational algebra operation

**Objective:** - To understand the use of inbuilt function and relational algebra with sql query.

**Problem:** -

1. Consider the following table structure and attempt.

Supplier-(scode,sname,scity,turnover)

Part-(pcode,weigh,color,cost,sellingprice)

Supplier\_Part-(scode,pcode,qty)

a) Create tables

```
use dbms_lab;
-- ex 5 and 6 07/02/2023

CREATE TABLE Supplier (scode INT PRIMARY KEY, sname VARCHAR(50), scity VARCHAR(50),turnover INT);

CREATE TABLE Part (pcode INT PRIMARY KEY, weight INT, color VARCHAR(50), cost INT, sellingprice INT);

CREATE TABLE Supplier_Part (scode INT, pcode INT, qty INT, FOREIGN KEY (scode) REFERENCES Supplier(scode), FOREIGN KEY (pcode) REFERENCES Part(pcode));
```

b) Populate the table.

```
INSERT INTO Supplier (scode, sname, scity, turnover) VALUES
(1, 'Supplier1', 'Ahmedabad', 75000000),
(2, 'Supplier2', 'Surat', 45000000),
(3, 'Supplier3', 'Rajkot', 64200000),
```

```
(4, 'Supplier4', 'Ahmedabad', 43000000),
(5, 'Supplier5', 'Gandhinagar', 71000000);
```

```
INSERT INTO Part (pcode, weight, color, cost, sellingprice) VALUES
(1, 20, 'Red', 8, 35),
(2, 30, 'Blue', 14, 30),
(3, 40, 'Green', 15, 30),
(4, 25, 'Balck', 7, 30),
(5, 35, 'White', 5, 29);
```

```
INSERT INTO Supplier_Part (scode, pcode, qty) VALUES
(1, 1, 548), (2, 2, 249), (3, 3, 359), (4, 4, 410), (5, 5, 250);
```

## 2. Write appropriate SQL Statement for the following:

### 1. Get the supplier number and part number in ascending order of supplier number.

```
select scode, pcode from Supplier_Part order by scode asc;
```

scode	pcode
1	1
2	2
3	3
4	4
5	5

### 2. Get the details of supplier who operate from Bombay with turnover 50.

```
select * from Supplier where scity = 'Bombay' and turnover = 50;
```

	scode	sname	scity	turnover
*	NULL	NULL	NULL	NULL

```
select * from Supplier where scity = 'Surat' and turnover = 45000000;
```

	scode	sname	scity	turnover
▶	2	Supplier2	Surat	45000000
*	NULL	NULL	NULL	NULL

3. Get the total number of suppliers.

```
select count(*) from Supplier;
```

	count(*)
▶	5

4. Get the part number weighing between 25 and 35.

```
select pcode from Part where weight between 25 and 35;
```

	pcode
▶	2
	4
	5
*	NULL

5. Get the supplier number whose turnover is null.

```
select scode from Supplier where turnover is null;
```

	scode
*	NULL

```
select scode from Supplier where turnover = '75000000';
```

	scode
▶	1
*	NULL

6. Get the part number that cost 20, 30 or 40 rupees.

```
select pcode from Part where cost in (8, 15, 20);
```

	pcode
▶	1
	3
*	NULL

7. Get the total quantity of part 2 that is supplied.

```
select sum(qty) from Supplier_Part where pcode = 2;
```

	sum(qty)
▶	249

8. Get the name of supplier who supply part 2.

```
select Supplier.sname from Supplier_Part inner join Supplier on  
Supplier_Part.scode = Supplier.scode where Supplier_Part.scode = 2;
```

	sname
▶	Supplier2

9. Get the part number whose cost is greater than the average cost.

```
select pcode from Part where cost > (select avg(cost) from Part);
```

	pcode
▶	2
	3
*	NULL

10. Get the supplier number and turnover in descending order of turnover.

```
select scode, turnover from Supplier order by turnover desc;
```

	scode	turnover
▶	1	75000000
	5	71000000
	3	64200000
	2	45000000
	4	43000000
*	NULL	NULL



## Experiment 7, 8: - Nested SQL queries or Subqueries

**Objective:** - To understand the use SQL Subquery.

**Problem:** - 1. Create the two tables (EMP and DEPT)

```
CREATE TABLE DEPT (
  DEPTNO INT PRIMARY KEY,
  DNAME VARCHAR(20) NOT NULL,
  LOC VARCHAR(20) NOT NULL
);
```

```
INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES
(10, 'ACCOUNTING', 'NEW YORK'),
(20, 'RESEARCH', 'DALLAS'),
(30, 'SALES', 'CHICAGO'),
(40, 'OPERATIONS', 'BOSTON');
```

```
SELECT * FROM DEPT;
```

	DEPTNO	DNAME	LOC
▶	10	ACCOUNTING	NEW YORK
	20	RESEARCH	DALLAS
	30	SALES	CHICAGO
	40	OPERATIONS	BOSTON
✱	NULL	NULL	NULL

```
CREATE TABLE EMP (EMPNO INT PRIMARY KEY,
  ENAME VARCHAR(20) NOT NULL,
  JOB VARCHAR(20) NOT NULL,
  MGR INT,
  HIREDATE DATE NOT NULL,
  SAL DECIMAL(10,2) NOT NULL,
  COMM DECIMAL(10,2),
  DEPTNO INT,
  FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO));
```

```
INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM,
  DEPTNO) VALUES
```

```
(7369, 'SMITH', 'CLERK', 7902, '1980/12/17', 500, 800, 20),
(7499, 'ALLEN', 'SALESMAN', 7698, '1981/02/20', 1600, 300, 30),
(7521, 'WARD', 'SALESMAN', 7698, '1981/02/22', 1250, 500, 30),
(7566, 'JONES', 'MANAGER', 7839, '1981/04/02', 2975, NULL, 20),
(7654, 'MARTIN', 'SALESMAN', 7698, '1981/09/28', 1250, 1400, 30),
(7698, 'BLAKE', 'MANAGER', 7839, '1981/05/01', 2850, NULL, 30),
```

```
select * FROM EMP;
```

[illegible]

- ```
SELECT * FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'BLAKE');
```

[illegible]

```
SELECT * FROM EMP WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'BLAKE');
```

[illegible]

```
SELECT * FROM EMP WHERE SAL = (SELECT SAL FROM EMP WHERE ENAME =
'FORD') OR SAL = (SELECT SAL FROM EMP WHERE ENAME = 'SMITH') ORDER
BY ENAME DESC;
```

[illegible]

```
SELECT * FROM EMP WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME =
'MILLER') OR SAL > (SELECT SAL FROM EMP WHERE ENAME = 'ALLEN');
```

[illegible]

5. Find the highest paid employee of sales department.

```
SELECT * FROM EMP WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES') ORDER BY SAL DESC LIMIT 1;
```

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|---------|------|------------|---------|------|--------|
| ▶ | 7698  | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30     |
| • | NULL  | NULL  | NULL    | NULL | NULL       | NULL    | NULL | NULL   |

6. List the employees who are senior to most recently hired employee working under king.

```
SELECT * FROM EMP E1 WHERE HIREDATE < (SELECT MAX(HIREDATE) FROM EMP WHERE JOB = E1.JOB AND MGR = (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));
```

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|---------|------|------------|---------|------|--------|
| ▶ | 7566  | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL | 20     |
|   | 7698  | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30     |
| * | NULL  | NULL  | NULL    | NULL | NULL       | NULL    | NULL | NULL   |

7. List the names of the emps who are getting the highest sal dept wise.

```
SELECT ENAME, SAL FROM EMP E WHERE E.SAL IN(SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO);
```

|   | ENAME | SAL     |
|---|-------|---------|
| ▶ | BLAKE | 2850.00 |
|   | SCOTT | 3000.00 |
|   | KING  | 5000.00 |
|   | FORD  | 3000.00 |

8. List the emps whose sal is equal to the average of max and minimum

```
SELECT * FROM EMP WHERE SAL = (SELECT (MAX(SAL) + MIN(SAL))/2 FROM EMP);
```

**9. List the emps who joined in the company on the same date.**

**10. Find out the emps who joined in the company before their managers.**

[illegible]

## Experiment 9: - Group by & having clause

**Objective:** To understand the use of group by and having clause.

**Problem:** - Write the SQL Queries for the following queries (use EMP and DEPT table of Exp 8).

1. List the Deptno where there are no emps.

```
SELECT D.DEPTNO FROM DEPT D WHERE DEPTNO NOT IN (SELECT DISTINCT DEPTNO FROM EMP);
```

|   | DEPTNO |
|---|--------|
| ▶ | 40     |
| * | NULL   |

2. List the No.of emp's and Avg salary within each department for each job.

```
SELECT DEPTNO, JOB, COUNT(*) AS NUM_EMPLOYEES, AVG(SAL) AS AVG_SALARY FROM EMP GROUP BY DEPTNO, JOB;
```

|   | DEPTNO | JOB       | NUM_EMPLOYEES | AVG_SALARY  |
|---|--------|-----------|---------------|-------------|
| ▶ | 20     | CLERK     | 2             | 800.000000  |
|   | 30     | SALESMAN  | 4             | 1400.000000 |
|   | 20     | MANAGER   | 1             | 2975.000000 |
|   | 30     | MANAGER   | 1             | 2850.000000 |
|   | 10     | MANAGER   | 1             | 2450.000000 |
|   | 20     | ANALYST   | 2             | 3000.000000 |
|   | 10     | PRESIDENT | 1             | 5000.000000 |
|   | 30     | CLERK     | 1             | 950.000000  |
|   | 10     | CLERK     | 1             | 1300.000000 |

3. Find the maximum average salary drawn for each job except for 'President'.

```
SELECT JOB, MAX(AVG_SAL) AS MAX_AVG_SALARY FROM (SELECT JOB, AVG(SAL) AS AVG_SAL FROM EMP WHERE JOB != 'President' GROUP BY JOB, DEPTNO) AS JOB_AVG_SALARY GROUP BY JOB;
```

|   | JOB      | MAX_AVG_SALARY |
|---|----------|----------------|
| ▶ | CLERK    | 1300.000000    |
|   | SALESMAN | 1400.000000    |
|   | MANAGER  | 2975.000000    |
|   | ANALYST  | 3000.000000    |

4. List the department details where at least two emps are working.

**SELECT \* FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING COUNT(\*) >= 2);**

|   | DEPTNO | DNAME      | LOC      |
|---|--------|------------|----------|
| ▶ | 10     | ACCOUNTING | NEW YORK |
|   | 20     | RESEARCH   | DALLAS   |
|   | 30     | SALES      | CHICAGO  |
| * | NULL   | NULL       | NULL     |

5. List the no. of emps in each department where the no. is more than 3.

**SELECT DEPTNO, COUNT(\*) AS NUM\_EMPLOYEES FROM EMP GROUP BY DEPTNO HAVING COUNT(\*) > 3;**

|   | DEPTNO | NUM_EMPLOYEES |
|---|--------|---------------|
| ▶ | 20     | 5             |
|   | 30     | 6             |

6. List the names of the emps who are getting the highest sal dept wise.

**SELECT D.DEPTNO, E.ENAME, E.SAL FROM EMP E INNER JOIN DEPT D ON E.DEPTNO = D.DEPTNO WHERE (E.DEPTNO, E.SAL) IN (SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTNO) ORDER BY DEPTNO;**

|   | DEPTNO | ENAME | SAL     |
|---|--------|-------|---------|
| ▶ | 10     | KING  | 5000.00 |
|   | 20     | SCOTT | 3000.00 |
|   | 20     | FORD  | 3000.00 |
|   | 30     | BLAKE | 2850.00 |

7. List the Deptno and their average salaries for dept with the average salary less than the averages for all departments.

```
SELECT DEPTNO, AVG(SAL) AS AVG_SALARY FROM EMP GROUP BY DEPTNO  
HAVING AVG(SAL) < (SELECT AVG(SAL) FROM EMP);
```

|   | DEPTNO | AVG_SALARY |
|---|--------|------------|
| ▶ | 30     | 1566.66667 |



## Experiment 10: - (Joins in SQL) To execute and verify the SQL commands using Join.

**Objective:** - SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.

**Problem:** - Refer Experiment 7 & 8 and execute the same questions by using join.

1. List the details of the emps whose Salaries more than the employee BLAKE.

```
SELECT E1.* FROM EMP E1 JOIN EMP E2 ON E1.SAL > E2.SAL AND E2.ENAME = "BLAKE";
```

|   | EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|-----------|------|------------|---------|------|--------|
| ▶ | 7566  | JONES | MANAGER   | 7839 | 1981-04-02 | 2975.00 | NULL | 20     |
|   | 7788  | SCOTT | ANALYST   | 7566 | 1982-12-09 | 3000.00 | NULL | 20     |
|   | 7839  | KING  | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL | 10     |
|   | 7902  | FORD  | ANALYST   | 7566 | 1981-12-03 | 3000.00 | NULL | 20     |

2. List the emps whose Jobs are same as ALLEN.

```
SELECT E1.* FROM EMP E1 JOIN EMP E2 ON E1.JOB = E2.JOB AND E2.ENAME = "ALLEN";
```

|   | EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL     | COMM    | DEPTNO |
|---|-------|--------|----------|------|------------|---------|---------|--------|
| ▶ | 7499  | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00  | 30     |
|   | 7521  | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250.00 | 500.00  | 30     |
|   | 7654  | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250.00 | 1400.00 | 30     |
|   | 7844  | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500.00 | 0.00    | 30     |

3. List the Emps whose Sal is same as FORD or SMITH in desc order of Names.

**SELECT e.\* FROM EMP e JOIN (SELECT SAL FROM EMP WHERE ENAME IN ('FORD', 'SMITH')) s ON e.SAL = s.SAL ORDER BY e.ENAME DESC;**

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM   | DEPTNO |
|---|-------|-------|---------|------|------------|---------|--------|--------|
| ▶ | 7369  | SMITH | CLERK   | 7902 | 1980-12-17 | 500.00  | 800.00 | 20     |
|   | 7788  | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000.00 | NULL   | 20     |
|   | 7902  | FORD  | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL   | 20     |

**4. List the emps Whose Jobs are same as MILLER or Sal is more than ALLEN.**

**SELECT E1.EMPNO, E1.ENAME, E1.JOB, E1.MGR, E1.HIREDATE, E1.SAL, E1.COMM, E1.DEPTNO FROM EMP E1 JOIN EMP E2 ON (E1.JOB =E2.JOB AND E2.ENAME = "MILLER") OR (E1.SAL > E2.SAL AND E2.ENAME = "ALLEN");**

|   | EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL     | COMM   | DEPTNO |
|---|-------|--------|-----------|------|------------|---------|--------|--------|
| ▶ | 7369  | SMITH  | CLERK     | 7902 | 1980-12-17 | 500.00  | 800.00 | 20     |
|   | 7566  | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975.00 | NULL   | 20     |
|   | 7698  | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850.00 | NULL   | 30     |
|   | 7782  | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450.00 | NULL   | 10     |
|   | 7788  | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000.00 | NULL   | 20     |
|   | 7839  | KING   | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL   | 10     |
|   | 7876  | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100.00 | NULL   | 20     |
|   | 7900  | JAMES  | CLERK     | 7698 | 1981-12-03 | 950.00  | NULL   | 30     |
|   | 7902  | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000.00 | NULL   | 20     |
|   | 7934  | MILLER | CLERK     | 7782 | 1982-01-23 | 1300.00 | NULL   | 10     |

**5. Find the highest paid employee of sales department.**

**SELECT e.\* FROM EMP e INNER JOIN DEPT d ON e.DEPTNO = d.DEPTNO WHERE d.DNAME = 'SALES' ORDER BY e.SAL DESC LIMIT 1;**

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|---------|------|------------|---------|------|--------|
| ▶ | 7698  | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30     |

**6. List the employees who are senior to most recently hired employee working under king.**

```
SELECT E1.* FROM EMP E1 JOIN (SELECT JOB, MAX(HIREDATE) AS
MAX_HIREDATE FROM EMP WHERE MGR = (SELECT EMPNO FROM EMP
WHERE ENAME = 'KING') GROUP BY JOB) E2 ON E1.JOB = E2.JOB AND
E1.HIREDATE < E2.MAX_HIREDATE;
```

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|---------|------|------------|---------|------|--------|
| ▶ | 7566  | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL | 20     |
|   | 7698  | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30     |

7. List the names of the emps who are getting the highest sal dept wise.

```
SELECT DNAME, ENAME, SAL FROM EMP E1 JOIN DEPT D1 ON E1.DEPTNO =
D1.DEPTNO WHERE SAL = (SELECT MAX(SAL) FROM EMP E2 WHERE
E1.DEPTNO = E2.DEPTNO);
```

|   | DNAME      | ENAME | SAL     |
|---|------------|-------|---------|
| ▶ | ACCOUNTING | KING  | 5000.00 |
|   | RESEARCH   | SCOTT | 3000.00 |
|   | RESEARCH   | FORD  | 3000.00 |
|   | SALES      | BLAKE | 2850.00 |

s

8. List the emps whose sal is equal to the average of max and minimum

```
SELECT e.* FROM EMP e JOIN (SELECT AVG(SAL) AS AVG_SAL FROM (SELECT
MAX(SAL) AS SAL FROM EMP UNION SELECT MIN(SAL) AS SAL FROM EMP) s) a
ON e.SAL = a.AVG_SAL;
```

|  | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|--|-------|-------|-----|-----|----------|-----|------|--------|
|--|-------|-------|-----|-----|----------|-----|------|--------|

9. List the emps who joined in the company on the same date.

```
SELECT E1.EMPNO, E1.ENAME, E1.JOB, E1.MGR, E1.HIREDATE, E1.SAL,
E1.COMM, E1.DEPTNO FROM EMP E1 JOIN EMP E2 ON E1.HIREDATE =
E2.HIREDATE AND E1.EMPNO <> E2.EMPNO ORDER BY E1.HIREDATE;
```

|   | EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL     | COMM | DEPTNO |
|---|-------|-------|---------|------|------------|---------|------|--------|
| ▶ | 7902  | FORD  | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL | 20     |
|   | 7900  | JAMES | CLERK   | 7698 | 1981-12-03 | 950.00  | NULL | 30     |

10. Find out the emps who joined in the company before their managers.

```
SELECT E1.EMPNO, E1.ENAME, E1.JOB, E1.MGR, E1.HIREDATE, E1.SAL,  
E1.COMM, E1.DEPTNO FROM EMP E1 JOIN EMP E2 ON E1.MGR = E2.EMPNO  
AND E1.HIREDATE < E2.HIREDATE ORDER BY E1.HIREDATE;
```

|   | EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL     | COMM   | DEPTNO |
|---|-------|-------|----------|------|------------|---------|--------|--------|
| ▶ | 7369  | SMITH | CLERK    | 7902 | 1980-12-17 | 500.00  | 800.00 | 20     |
|   | 7499  | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30     |
|   | 7521  | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250.00 | 500.00 | 30     |
|   | 7566  | JONES | MANAGER  | 7839 | 1981-04-02 | 2975.00 | NULL   | 20     |
|   | 7698  | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850.00 | NULL   | 30     |
|   | 7782  | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450.00 | NULL   | 10     |