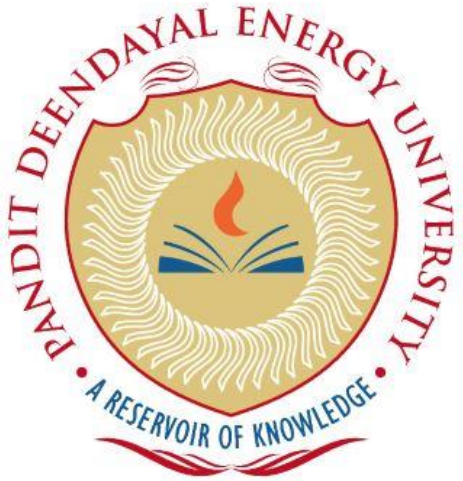
PANDIT DEENDAYAL ENERGY UNIVERSITY

SCHOOL OF TECHNOLOGY



Course: Database Management Systems

Course Code: 20CP208P

LAB MANUAL

B.Tech. (Computer Science and Engineering)

Semester 4

Submitted To:

Dr. Hargeet Kaur

Submitted By:

HARSH SHAH

21BCP359

G11 Batch

INDEX

S No.	List of Practical	Date	Sign
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

EXPERIMENT 1

Date: 17th January 2023

TITLE: DDL (Data Definition Language) commands

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

1. CREATE THE TABLES DESCRIBED BELOW:

Table name: CLIENT_MASTER

Column name	Data type	Size
CLIENTNO	Varchar	6
NAME	Varchar	20
ADDRESS 1	Varchar	30
ADDRESS 2	Varchar	30
CITY	Varchar	15
PINCODE	Integer	
STATE	Varchar	15
BALDUE	decimal	10.2

Query:

```
CREATE TABLE CLIENT_MASTER (
        CLIENTNO VARCHAR (6),
        NAME VARCHAR (20),
        ADDRESS1 VARCHAR (30),
        ADDRESS2 VARCHAR (30),
        CITY VARCHAR (15),
        PINCODE INTEGER,
        STATE VARCHAR (15),
```

```
    BALDUE DECIMAL (10,2)
    );
DESC CLIENT_MASTER;
```

Output:

Field	Type	Null	Key	Default	Extra
CLIENTNO	varchar(6)	YES		NULL	
NAME	varchar(20)	YES		NULL	
ADDRESS1	varchar(30)	YES		NULL	
ADDRESS2	varchar(30)	YES		NULL	
CITY	varchar(15)	YES		NULL	
PINCODE	int	YES		NULL	
STATE	varchar(15)	YES		NULL	
BALDUE	decimal(10,2)	YES		NULL	

Table name: PRODUCT_MASTER

Column name	Data type	Size
PRODUCTNO	Varchar	6
DESCRIPTION	Varchar	15
PROFITPERCENT	Decimal	4.2
UNIT MEASURE	Varchar	10
QTYONHAND	Integer	
REORDERL VL	Integer	
SELLPRICE	Decimal	8.2
COSTPRICE	Decimal	8.2

Query:

```
CREATE TABLE PRODUCT_MASTER(
    PRODUCTNO VARCHAR (6),
```

```
    DESCRIPTION VARCHAR(15),
    PROFIT_PERCENT DECIMAL (4,2),
    UNIT_MEASURE VARCHAR (10),
    QTY_ON_HAND INTEGER,
    REORDERL_VL INTEGER,
    SELLPRICE DECIMAL(8,2),
    COST_PRICE DECIMAL(8,2)
);
DESC PRODUCT_MASTER;
```

Output:

Field	Type	Null	Key	Default	Extra
PRODUCTNO	varchar(6)	YES		NULL	
DESCRIPTION	varchar(15)	YES		NULL	
PROFIT_PERCENT	decimal(4,2)	YES		NULL	
UNIT_MEASURE	varchar(10)	YES		NULL	
QTY_ON_HAND	int	YES		NULL	
REORDERL_VL	int	YES		NULL	
SELLPRICE	decimal(8,2)	YES		NULL	
COST_PRICE	decimal(8,2)	YES		NULL	

Table name: SALESMAN_MASTER

Column name	**Data type**	**Size**
SALESMANNO	Varchar	6
SALESMANNAME	Varchar	20
ADDRESS 1	Varchar	30
ADDRESS 2	Varchar	30
CITY	Varchar	20
PINCODE	Integer	
STATE	Varchar	20

SALAMT	Real	
TGTTOGET	Decimal	
YTDSALES	Double	6.2
REMARKS	Varchar	60

Query:

```
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)
);
DESC SALESMAN_MASTER;
```

Output:

Field	Type	Null	Key	Default	Extra
SALESMANNO	varchar(6)	YES		NULL	
SALESMANNAME	varchar(20)	YES		NULL	
ADDRESS1	varchar(30)	YES		NULL	
ADDRESS2	varchar(30)	YES		NULL	
CITY	varchar(20)	YES		NULL	
PINCODE	int	YES		NULL	
STATE	varchar(20)	YES		NULL	
SALAMT	double	YES		NULL	
TGTTOGET	decimal(10,0)	YES		NULL	
YTDSALES	double(6,2)	YES		NULL	
REMARKS	varchar(60)	YES		NULL	

2. INSERT DATA INTO THE RESPECTIVE TABLES:

a. Data for CLIENT_MASTER table:

Client no	Name	City	Pincode	State	BalDue
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

Query:

```
INSERT INTO CLIENT_MASTER VALUES("C00001", "Ivan bayross", NULL, ", "Mumbai", 400054, "Maharashtra", 15000);

INSERT INTO CLIENT_MASTER(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES("C00002", "Mamta muzumdar", "Madras", 780001, "Tamil nadu", 0);

INSERT INTO CLIENT_MASTER(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00003', 'Chhaya bankar', 'Mumbai', 400057, 'Maharashtra', 5000);

INSERT INTO CLIENT_MASTER(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00004', 'Ashwini joshi', 'Bangalore', 560001, 'Karnataka', 0);

INSERT INTO CLIENT_MASTER(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00005', 'Hansel colaco', 'Mumbai', 400060, 'Maharashtra', 2000);

INSERT INTO CLIENT_MASTER(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00006', 'Deepak sharma', 'Mangalore', 560050, 'Karnataka', 0);

SELECT * FROM CLIENT_MASTER;
```

Output:

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C00001	Ivan bayross	NULL		Mumbai	400054	Maharashtra	15000.00
C00002	Mamta muzumdar	NULL	NULL	Madras	780001	Tamil nadu	0.00
C00003	Chhaya bankar	NULL	NULL	Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini joshi	NULL	NULL	Bangalore	560001	Karnataka	0.00
C00005	Hansel colaco	NULL	NULL	Mumbai	400060	Maharashtra	2000.00
C00006	Deepak sharma	NULL	NULL	Mangalore	560050	Karnataka	0.00

b. Data for PRODUCT_MASTER table:

ProductNo	Description	Profit percent	Unit measure	Qtyonhand	RecorderLvl	SellPrice	CostPrice
P00001	T-Shirt	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

Query:

```
INSERT INTO PRODUCT_MASTER VALUES('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250);

INSERT INTO PRODUCT_MASTER VALUES('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350);

INSERT INTO PRODUCT_MASTER VALUES('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450);

INSERT INTO PRODUCT_MASTER VALUES('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500);

INSERT INTO PRODUCT_MASTER VALUES('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550);

INSERT INTO PRODUCT_MASTER VALUES('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450);

INSERT INTO PRODUCT_MASTER VALUES('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250);
```

INSERT INTO PRODUCT_MASTER VALUES('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175);

INSERT INTO PRODUCT_MASTER VALUES('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);

SELECT * FROM PRODUCT_MASTER;

Output:

PRODUCTNO	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASURE	QTY_ON_HAND	REORDERL_VL	SELLPRICE	COST_PRICE
P00001	T-Shirt	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

c. Data for SALESMAN_MASTER table:

SalesmanNo	Name	Address1	Address2	City	Pincode	State
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra

SalesmanNo	SalAmt	TgtToGet	YtdSales	Remarks
S00001	3000	100	50	Good
S00002	3000	200	100	Good
S00003	3000	200	100	Good
S00004	3500	200	150	Good

Query:

```
INSERT INTO SALESMAN_MASTER VALUES('S00001', 'Aman', 'A/14', 'Worli',
'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good');

INSERT INTO SALESMAN_MASTER VALUES('S00002', 'OMKAR', '65', 'NARIMAN',
'Mumbai', 400002, 'Maharashtra',3000, 200, 100, 'Good');

INSERT INTO SALESMAN_MASTER VALUES('S00003', 'RAJ', 'P-7', 'BANDRA',
'Mumbai', 400032, 'Maharashtra',3000, 200, 100, 'Good');

INSERT INTO SALESMAN_MASTER      VALUES('S00004', 'ASHISH', 'A/5', 'JUHU',
'Mumbai', 400044, 'Maharashtra',3000, 200, 150, 'Good');

SELECT * FROM SALESMAN_MASTER;
```

Output:

SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	OMKAR	65	NARIMAN	Mumbai	400002	Maharashtra
S00003	RAJ	P-7	BANDRA	Mumbai	400032	Maharashtra
S00004	ASHISH	A/5	JUHU	Mumbai	400044	Maharashtra

SALAMT	TGTTOGET	YTDSALES	REMARKS
3000	100	50.00	Good
3000	200	100.00	Good
3000	200	100.00	Good
3000	200	150.00	Good

EXPERIMENT 2

Date: 20th January 2023

TITLE: DML (Data Manipulation Language) commands

Objective: To understand the concept of different DML commands.

1. EXERCISE ON RETRIEVING RECORDS FROM A TABLE.

a. Find out the names of all the clients.

- SELECT NAME FROM CLIENT_MASTER;

NAME
Ivan bayross
Mamta muzumdar
Chhaya bankar
Ashwini joshi
Hansel colaco
Deepak sharma

b. Retrieve the entire contents of the Client_Master table.

- SELECT NAME FROM CLIENT_MASTER;

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C00001	Ivan bayross	NULL		Mumbai	400054	Maharashtra	15000.00
C00002	Mamta muzumdar	NULL	NULL	Madras	780001	Tamil nadu	0.00
C00003	Chhaya bankar	NULL	NULL	Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini joshi	NULL	NULL	Bangalore	560001	Karnataka	0.00
C00005	Hansel colaco	NULL	NULL	Mumbai	400060	Maharashtra	2000.00
C00006	Deepak sharma	NULL	NULL	Mangalore	560050	Karnataka	0.00

c. Retrieve the list of names, city and the state of all the clients.

- SELECT NAME, CITY, STATE FROM CLIENT_MASTER;

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

d. List the various products available from the Product_Master table.

- SELECT DESCRIPTION FROM PRODUCT_MASTER;

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

e. List all the clients who are located in Mumbai.

- SELECT NAME FROM CLIENT_MASTER WHERE CITY="MUMBAI";

NAME
Ivan bayross
Chhaya bankar
Hansel colaco

f. Find the names of salesman who have a salary equal to Rs.3000.

- SELECT SALESMANNAME FROM SALESMAN_MASTER WHERE SALAMT="3000";

NAME
Ivan bayross
Chhaya bankar
Hansel colaco

2. EXERCISE ON UPDATING RECORDS IN A TABLE

a. Change the city of ClientNo 'C00005' to 'Bangalore'.

- UPDATE CLIENT_MASTER SET CITY = "BANGALORE" WHERE CLIENTNO = "C00005";
- SELECT CLIENTNO, CITY FROM CLIENT_MASTER WHERE CLIENTNO = "C00005";

CLIENTNO	CITY
C00005	BANGALORE

b. Change the BalDue of ClientNo 'C00001' to Rs.1000.

- UPDATE CLIENT_MASTER SET BALDUE = "1000" WHERE CLIENTNO = "C00001";
- SELECT CLIENTNO, BALDUE FROM CLIENT_MASTER WHERE CLIENTNO = "C00001";

CLIENTNO	BALDUE
C00001	1000.00

c. Change the cost price of 'Trousers' to Rs.950.00.

- UPDATE PRODUCT_MASTER SET COST_PRICE = "950" WHERE DESCRIPTION = "TROUSERS";
- SELECT DESCRIPTION, COST_PRICE FROM PRODUCT_MASTER WHERE DESCRIPTION = "TROUSERS";

DESCRIPTION	COST_PRICE
Trousers	950.00

d. Change the city of the salesman to Pune.

- UPDATE SALESMAN_MASTER SET CITY = "PUNE";
- SELECT * FROM SALESMAN_MASTER;

SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTDSALES	REMARKS
S00001	Aman	A/14	Worli	PUNE	400002	Maharashtra	3000	100	50.00	Good
S00002	OMKAR	65	NARIMAN	PUNE	400002	Maharashtra	3000	200	100.00	Good
S00003	RAJ	P-7	BANDRA	PUNE	400032	Maharashtra	3000	200	100.00	Good
S00004	ASHISH	A/5	JUHU	PUNE	400044	Maharashtra	3000	200	150.00	Good

3. EXERCISE ON DELETING RECORDS IN A TABLE

a. Delete all salesman from the Salesman_Master whose salaries are equal to Rs.3500.

- DELETE FROM SALESMAN_MASTER WHERE SALAMT=3500;
- SELECT * FROM SALESMAN_MASTER;

SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTDSALES	REMARKS
S00001	Aman	A/14	Worli	PUNE	400002	Maharashtra	3000	100	50.00	Good
S00002	OMKAR	65	NARIMAN	PUNE	400002	Maharashtra	3000	200	100.00	Good
S00003	RAJ	P-7	BANDRA	PUNE	400032	Maharashtra	3000	200	100.00	Good
S00004	ASHISH	A/5	JUHU	PUNE	400044	Maharashtra	3000	200	150.00	Good

b. Delete all products from Product_Master where the quantity on hand is equal to 100.

- DELETE FROM PRODUCT_MASTER WHERE QTY_ON_HAND=100;
- SELECT * FROM PRODUCT_MASTER;

PRODUCTNO	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASURE	QTY_ON_HAND	REORDERL_VL	SELLPRICE	COST_PRICE
P00001	T-Shirt	5.00	Piece	200	50	350.00	250.00
P0345	Shirts	6.00	Piece	150	50	500.00	350.00
P07868	Trousers	2.00	Piece	150	50	850.00	950.00
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

c. Delete from Client_Master where the column state holds the value 'Tamil Nadu'.

- DELETE FROM CLIENT_MASTER WHERE STATE="TAMIL NADU";
- SELECT * FROM CLIENT_MASTER;

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C00001	Ivan bayross	NULL		Mumbai	400054	Maharashtra	1000.00
C00003	Chhaya bankar	NULL	NULL	Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini joshi	NULL	NULL	Bangalore	560001	Karnataka	0.00
C00005	Hansel colaco	NULL	NULL	BANGALORE	400060	Maharashtra	2000.00
C00006	Deepak sharma	NULL	NULL	Mangalore	560050	Karnataka	0.00

4. EXERCISE ON ALTERING THE TABLE STRUCTURE

a. Add a column called 'Telephone' of data type integer to the Client_Master table.

- ALTER TABLE CLIENT_MASTER ADD TELEPHONE INTEGER;
- DESC CLIENT_MASTER;

Field	Type	Null	Key	Default	Extra
CLIENTNO	varchar(6)	YES		NULL	
NAME	varchar(20)	YES		NULL	
ADDRESS1	varchar(30)	YES		NULL	
ADDRESS2	varchar(30)	YES		NULL	
CITY	varchar(15)	YES		NULL	
PINCODE	int	YES		NULL	
STATE	varchar(15)	YES		NULL	
BALDUE	decimal(10,2)	YES		NULL	
TELEPHONE	int	YES		NULL	

b. Change the size of SellPrice column in Product _Master to (10, 2).

- ALTER TABLE PRODUCT_MASTER MODIFY SELLPRICE DECIMAL(10,2);
- DESC PRODUCT_MASTER;

Field	Type	Null	Key	Default	Extra
PRODUCTNO	varchar(6)	YES		NULL	
DESCRIPTION	varchar(15)	YES		NULL	
PROFIT_PERCENT	decimal(4,2)	YES		NULL	
UNIT_MEASURE	varchar(10)	YES		NULL	
QTY_ON_HAND	int	YES		NULL	
REORDERL_VL	int	YES		NULL	
SELLPRICE	decimal(10,2)	YES		NULL	
COST_PRICE	decimal(8,2)	YES		NULL	

5. EXERCISE ON DELETING THE TABLE STRUCTURE ALONG WITH THE DATA

a. Destroy the table Client_Master along with its data.

- DROP TABLE CLIENT_MASTER;
- SHOW TABLES;

Tables_in_dmbs_lab
product_master
salesman_master

6. EXERCISE ON RENAMING THE TABLE

a. Change the name of the Salesman_Master to sman_mast.

- ALTER TABLE SALESMAN_MASTER RENAME TO SMAN_MAST;
- SHOW TABLES;

Tables_in_dmbs_lab
product_master
sman_mast

EXPERIMENT 3

Date: 27th January 2023

TITLE: 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

CREATE THE TABLES DESCRIBED BELOW:

a. Table name: CLIENT_MASTER_1

Description: use to store client information

Column name	data type	Size	Constraints
CLIENTNO	Varchar	6	Primary key / first letter must start with 'C'
NAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	
ADDRESS 2	Varchar	30	
CITY	Varchar	15	
PINCODE	Integer	8	
STATE	Varchar	15	
BALDUE	Decimal	10,2	

Query:

```
CREATE TABLE CLIENT_MASTER_1(
    CLIENTNO VARCHAR (6),
    NAME VARCHAR(20) NOT NULL,
    ADDRESS1 VARCHAR (30),
    ADDRESS2 VARCHAR (30),
    CITY VARCHAR (15),
    PINCODE INTEGER(8),
    STATE VARCHAR (15),
    BALDUE DECIMAL (10,2),
   PRIMARY KEY (CLIENTNO),
   CHECK (CLIENTNO LIKE 'C%')
);
```

DESC CLIENT_MASTER_1;

Field	Type	Null	Key	Default	Extra
CLIENTNO	varchar(6)	NO	PRI	NULL	
NAME	varchar(20)	NO		NULL	
ADDRESS1	varchar(30)	YES		NULL	
ADDRESS2	varchar(30)	YES		NULL	
CITY	varchar(15)	YES		NULL	
PINCODE	int	YES		NULL	
STATE	varchar(15)	YES		NULL	
BALDUE	decimal(10,2)	YES		NULL	

b. Table Name: PRODUCT_MASTER_1

Description: used to store product information

Column name	data type	Size	Attributes
PRODUCTNO	Varchar	6	Primary Key/ first letter must start with 'P'
DESCRIPTION	Varchar	15	Not Null
PROFITPERCENT	Decimal	4,2	Not Null
UNIT MEASURE	Varchar	10	Not Null
QTYONHAND	Integer	8	Not Null
REORDERL VL	Integer	8	Not Null
SELLPRICE	Decimal	8,2	Not Null
COSTPRICE	Decimal	8,2	Not Null

Query:

```
CREATE TABLE PRODUCT_MASTER_1(
    PRODUCTNO VARCHAR (6),
    DESCRIPTION VARCHAR(15) NOT NULL,
    PROFIT_PERCENT DECIMAL (4,2) NOT NULL,
    UNIT_MEASURE VARCHAR (10) NOT NULL,
    QTY_ON_HAND INTEGER(8) NOT NULL,
    REORDERL_VL INTEGER(8) NOT NULL,
```

```
    SELLPRICE DECIMAL(8,2) NOT NULL,
    COST_PRICE DECIMAL(8,2) NOT NULL,
    PRIMARY KEY (PRODUCTNO),
    CHECK (PRODUCTNO LIKE 'P%')
);
DESC PRODUCT_MASTER_1;
```

Field	Type	Null	Key	Default	Extra
PRODUCTNO	varchar(6)	NO	PRI	NULL	
DESCRIPTION	varchar(15)	NO		NULL	
PROFIT_PERCENT	decimal(4,2)	NO		NULL	
UNIT_MEASURE	varchar(10)	NO		NULL	
QTY_ON_HAND	int	NO		NULL	
REORDERL_VL	int	NO		NULL	
SELLPRICE	decimal(8,2)	NO		NULL	
COST_PRICE	decimal(8,2)	NO		NULL	

c. Table Name: SALESMAN_MASTER _1

Description: used to store salesman information working for the company

Column name	data type	Size	Attributes
SALESMANNO	Varchar	6	Primary Key/ first letter must start with 'S'
SALESMANNAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	Not Null
ADDRESS 2	Varchar	30	
CITY	Varchar	20	
PINCODE	Integer	8	
STATE	Varchar	20	
SALAMT	Real	8,2	Not Null , Cannot be 0
TGTTOGET	Decimal	6,2	Not Null , Cannot be 0
YTDSALES	Double	6,2	Not Null
REMARKS	Varchar	60	

Query:

```
CREATE TABLE SALESMAN_MASTER_1(
    SALESMANNO VARCHAR (6),
    SALESMANNAME VARCHAR(20) NOT NULL,
    ADDRESS1 VARCHAR(30) NOT NULL,
    ADDRESS2 VARCHAR(30),
    CITY VARCHAR (20),
    PINCODE INTEGER(8),
    STATE VARCHAR (20),
    SALAMT REAL(8,2) NOT NULL CHECK(SALAMT!=0),
    TGTTOGET DECIMAL(6,2) NOT NULL CHECK(TGTTOGET!=0),
    YTDSALES DOUBLE(6,2) NOT NULL,
    REMARKS VARCHAR (60),
    PRIMARY KEY (SALESMANNO),
    CHECK (SALESMANNO LIKE 'S%')
);
DESC SALESMAN_MASTER_1;
```

Field	Type	Null	Key	Default	Extra
SALESMANNO	varchar(6)	NO	PRI	NULL	
SALESMANNAME	varchar(20)	NO		NULL	
ADDRESS1	varchar(30)	NO		NULL	
ADDRESS2	varchar(30)	YES		NULL	
CITY	varchar(20)	YES		NULL	
PINCODE	int	YES		NULL	
STATE	varchar(20)	YES		NULL	
SALAMT	double(8,2)	NO		NULL	
TGTTOGET	decimal(6,2)	NO		NULL	
YTDSALES	double(6,2)	NO		NULL	
REMARKS	varchar(60)	YES		NULL	

REINSERT THE DATA IN THESE TWO TABLES BASED UPON LAB 1

a. Table name: CLIENT_MASTER_1

INSERT INTO CLIENT_MASTER_1 VALUES("C00001", "Ivan bayross", NULL, ", "Mumbai", 400054, "Maharashtra", 15000);

INSERT INTO CLIENT_MASTER_1(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES("C00002", "Mamta muzumdar", "Madras", 780001, "Tamil nadu", 0);

INSERT INTO CLIENT_MASTER_1(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00003', 'Chhaya bankar', 'Mumbai', 400057, 'Maharashtra', 5000);

INSERT INTO CLIENT_MASTER_1(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00004', 'Ashwini joshi', 'Bangalore', 560001, 'Karnataka', 0);

INSERT INTO CLIENT_MASTER_1(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00005', 'Hansel colaco', 'Mumbai', 400060, 'Maharashtra', 2000);

INSERT INTO CLIENT_MASTER_1(CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES('C00006', 'Deepak sharma', 'Mangalore', 560050, 'Karnataka', 0);

b. Table Name: PRODUCT_MASTER_1

INSERT INTO PRODUCT_MASTER_1 VALUES('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250);

INSERT INTO PRODUCT_MASTER_1 VALUES('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350);

INSERT INTO PRODUCT_MASTER_1 VALUES('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450);

INSERT INTO PRODUCT_MASTER_1 VALUES('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500);

INSERT INTO PRODUCT_MASTER_1 VALUES('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550);

INSERT INTO PRODUCT_MASTER_1 VALUES('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450);

INSERT INTO PRODUCT_MASTER_1 VALUES('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250);

INSERT INTO PRODUCT_MASTER_1 VALUES('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175);

INSERT INTO PRODUCT_MASTER_1 VALUES('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);

c. Table Name: SALESMAN_MASTER _1

INSERT INTO SALESMAN_MASTER_1 VALUES('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good');

INSERT INTO SALESMAN_MASTER_1 VALUES('S00002', 'OMKAR', '65', 'NARIMAN', 'Mumbai', 400002, 'Maharashtra',3000, 200, 100, 'Good');

INSERT INTO SALESMAN_MASTER_1 VALUES('S00003', 'RAJ', 'P-7', 'BANDRA', 'Mumbai', 400032, 'Maharashtra',3000, 200, 100, 'Good');

INSERT INTO SALESMAN_MASTER_1 VALUES('S00004', 'ASHISH', 'A/5', 'JUHU', 'Mumbai', 400044, 'Maharashtra',3000, 200, 150, 'Good');

d. Display the contents of each table.

SELECT * FROM CLIENT_MASTER_1;

CLIENTNO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BALDUE
C00001	Ivan bayross	NULL		Mumbai	400054	Maharashtra	15000.00
C00002	Mamta muzumdar	NULL	NULL	Madras	780001	Tamil nadu	0.00
C00003	Chhaya bankar	NULL	NULL	Mumbai	400057	Maharashtra	5000.00
C00004	Ashwini joshi	NULL	NULL	Bangalore	560001	Karnataka	0.00
C00005	Hansel colaco	NULL	NULL	Mumbai	400060	Maharashtra	2000.00
C00006	Deepak sharma	NULL	NULL	Mangalore	560050	Karnataka	0.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM PRODUCT_MASTER_1;

PRODUCTNO	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASURE	QTY_ON_HAND	REORDERL_VL	SELLPRICE	COST_PRICE
P00001	T-Shirt	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

SELECT * FROM SALESMAN_MASTER_1;

SALESMANNO	SALESMANNAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SALAMT	TGTTOGET	YTDSALES	REMARKS
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra	3000.00	100.00	50.00	Good
S00002	OMKAR	65	NARIMAN	Mumbai	400002	Maharashtra	3000.00	200.00	100.00	Good
S00003	RAJ	P-7	BANDRA	Mumbai	400032	Maharashtra	3000.00	200.00	100.00	Good
S00004	ASHISH	A/5	JUHU	Mumbai	400044	Maharashtra	3000.00	200.00	150.00	Good
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

EXPERIMENT 4

Date: 3rd February 2023

TITLE: 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, Foreign Key and constraints.

1. **Create table AUTHOR = {Author_ID, Lastname, Firstname, Email, City, Country}**

   Constraints:

   - **Author_ID – text data type, 5 characters, primary key**
   - **Lastname – text data type, 15 characters, not null**
   - **Firstname – text data type, 15 characters, not null**
   - **Email – text data type, 40 characters,**
   - **City – text data type, 15 characters,**
   - **Country – text data type, 15 characters**

   Query:

```
CREATE TABLE AUTHOR (
    AUTHOR_ID VARCHAR(5) PRIMARY KEY,
   LASTNAME VARCHAR(15),
   FIRSTNAME VARCHAR(15),
   EMAIL VARCHAR(40),
   CITY VARCHAR(15),
   COUNTRY VARCHAR(15)
);
DESC AUTHOR;
```

Field	Type	Null	Key	Default	Extra
AUTHOR_ID	varchar(5)	NO	PRI	NULL	
LASTNAME	varchar(15)	YES		NULL	
FIRSTNAME	varchar(15)	YES		NULL	
EMAIL	varchar(40)	YES		NULL	
CITY	varchar(15)	YES		NULL	
COUNTRY	varchar(15)	YES		NULL	

2. **Create Table BOOK = {Book_ID, Book_Title, Copies}**

Constraints:

- **Book_ID – text data type, 5 characters Primary Key Start With Character B**
- **Book_Title - Text data Type Not Null**
- **Copies- No.of copies Data Type int always greater the 2**

Query:

```
CREATE TABLE BOOK (
 BOOK_ID VARCHAR(5) PRIMARY KEY CHECK(BOOK_ID LIKE "B%"),
 BOOK_TITLE VARCHAR(50),
 COPIES INT CHECK(COPIES>2)
);
DESC BOOK;
```

Field	Type	Null	Key	Default	Extra
BOOK_ID	varchar(5)	NO	PRI	NULL	
BOOK_TITLE	varchar(50)	YES		NULL	
COPIES	int	YES		NULL	

3. **Create table AUTHOR_LIST = {Author_ID , Book_ID , Role}**

Constraints:

- **Author_ID – text data type, 5 characters, referenced by Author_ID from AUTHOR table**
- **Book_ID – text data type, 5 characters, referenced by Book_ID from BOOK table**
- **Role – text data type, 15 characters**
- **and primary key is: Author_ID, Book_ID**

Query:

```
CREATE TABLE AUTHOR_LIST (
   AUTHOR_ID VARCHAR(5),
   BOOK_ID VARCHAR(5),
   ROLE VARCHAR(15),
   PRIMARY KEY (AUTHOR_ID, BOOK_ID),
   FOREIGN KEY (AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID),
```

```
    FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID)
);
DESC AUTHOR_LIST;
```

Field	Type	Null	Key	Default	Extra
AUTHOR_ID	varchar(5)	NO	PRI	NULL	
BOOK_ID	varchar(5)	NO	PRI	NULL	
ROLE	varchar(15)	YES		NULL	

4. Add four records in each tables AUTHOR, BOOK, BOOK_LIST.

A. AUTHOR Table

```
INSERT INTO AUTHOR VALUES("A0001", "Tripathi", "Amish",
"amish.tripathi@gmail.com", "London", "United Kingdom");
INSERT INTO AUTHOR VALUES("A0002", "Christie", "Agatha",
"agatha.christie@gmail.com", "London", "United Kingdom");
INSERT INTO AUTHOR VALUES("A0003", "Doyle", "Arthur Cona",
"ac.doyle@gmail.com", "London", "United Kingdom");
INSERT INTO AUTHOR VALUES("A0004", "Verne", "Jules",
"jules.verne@gmail.com", "Paris", "France");
SELECT * FROM AUTHOR;
```

AUTHOR_ID	LASTNAME	FIRSTNAME	EMAIL	CITY	COUNTRY
A0001	Tripathi	Amish	amish.tripathi@gmail.com	London	United Kingdom
A0002	Christie	Agatha	agatha.christie@gmail.com	London	United Kingdom
A0003	Doyle	Arthur Cona	ac.doyle@gmail.com	London	United Kingdom
A0004	Verne	Jules	jules.verne@gmail.com	Paris	France
NULL	NULL	NULL	NULL	NULL	NULL

B. BOOK Table

```
INSERT INTO BOOK VALUES("B0001", "War of Lanka", 200);
INSERT INTO BOOK VALUES("B0002", "The Immortals of Meluha", 500);
INSERT INTO BOOK VALUES("B0003", "Murder on the Orient Express", 300);
INSERT INTO BOOK VALUES("B0004", "The Adventues of Sherlock Holmes", 400);
SELECT * FROM BOOK;
```

BOOK_ID	BOOK_TITLE	COPIES
B0001	War of Lanka	200
B0002	The Immortals of Meluha	500
B0003	Murder on the Orient Express	300
B0004	The Adventues of Sherlock Holmes	400
NULL	NULL	NULL

C. AUTHOR_LIST Table

INSERT INTO AUTHOR_LIST VALUES("A0001", "B0001", "Writer");

INSERT INTO AUTHOR_LIST VALUES("A0001", "B0002", "Author");

INSERT INTO AUTHOR_LIST VALUES("A0002", "B0003", "Author");

INSERT INTO AUTHOR_LIST VALUES("A0003", "B0004", "Author");

SELECT * FROM AUTHOR_LIST;

AUTHOR_ID	BOOK_ID	ROLE
A0001	B0001	Writer
A0001	B0002	Author
A0002	B0003	Author
A0003	B0004	Author
NULL	NULL	NULL

5. Alter structure of table AUTHOR_LIST add the field Publisher data type of 30 Character.

ALTER TABLE AUTHOR_LIST ADD PUBLISHER VARCHAR(30);

Field	Type	Null	Key	Default	Extra
AUTHOR_ID	varchar(5)	NO	PRI	NULL	
BOOK_ID	varchar(5)	NO	PRI	NULL	
ROLE	varchar(15)	YES		NULL	
PUBLISHER	varchar(30)	YES		NULL	

EXPERIMENT 5-6

Date: 17th February 2023

TITLE: Use of Inbuilt functions and relational algebra operation

OBJECTIVE: To understand the use of inbuilt function and relational algebra with SQL query.

1. **Consider the given Table Structures and**

   a) **Create Table**

SUPPLIER - (SCODE, SNAME, SCITY, TURNOVER)

```
CREATE TABLE SUPPLIER (
   SCODE VARCHAR(5) PRIMARY KEY,
   SNAME VARCHAR(30),
   SCITY VARCHAR(20),
   TURNOVER INTEGER
);
```

Field	Type	Null	Key	Default	Extra
SCODE	varchar(5)	NO	PRI	NULL	
SNAME	varchar(30)	YES		NULL	
SCITY	varchar(20)	YES		NULL	
TURNOVER	int	YES		NULL	

PART - (PCODE, WEIGH, COLOR, COST, SELLINGPRICE)

```
CREATE TABLE PART (
   PCODE VARCHAR(5) PRIMARY KEY,
   WEIGH DECIMAL(3,2),
   COLOR VARCHAR(10),
   COST INTEGER,
   SELLINGPRICE INTEGER
);
```

Field	Type	Null	Key	Default	Extra
PCODE	varchar(5)	NO	PRI	NULL	
WEIGH	decimal(3,2)	YES		NULL	
COLOR	varchar(10)	YES		NULL	
COST	int	YES		NULL	
SELLINGPRICE	int	YES		NULL	

SUPPLIER_PART - (SCODE, PCODE, QTY)

```
CREATE TABLE SUPPLIER_PART(
   SCODE VARCHAR(5),
   PCODE VARCHAR(5),
   QTY INTEGER,
   FOREIGN KEY (SCODE) REFERENCES SUPPLIER(SCODE),
   FOREIGN KEY (PCODE) REFERENCES PART(PCODE)
);
```

Field	Type	Null	Key	Default	Extra
SCODE	varchar(5)	YES	MUL	NULL	
PCODE	varchar(5)	YES	MUL	NULL	
QTY	int	YES		NULL	

b) Populate the tables.

```
INSERT INTO SUPPLIER VALUES('S01','TOM','BOMBAY',50);
INSERT INTO SUPPLIER VALUES('S02','TONY','NEW YORK',NULL);
INSERT INTO SUPPLIER VALUES('S03','PETER','CHENNAI',80);
INSERT INTO SUPPLIER VALUES('S04','JACK','AHEMDABAD',120);
```

SCODE	SNAME	SCITY	TURNOVER
s01	Tom	Bombay	50
s02	Tony	New York	NULL
s03	Peter	Chennai	80
s04	Jack	Ahemdabad	120
NULL	NULL	NULL	NULL

```
INSERT INTO PART VALUES("P01", 28, "RED", 30, 1000);
INSERT INTO PART VALUES("P02", 30, "BLUE",20, 800);
INSERT INTO PART VALUES("P03", 32, "PURPLE", 40, 100);
INSERT INTO PART VALUES("P04", 40, "ORANGE", 70, 700);
```

PCODE	WEIGH	COLOR	COST	SELLINGPRICE
p01	28	Red	30	1000
p02	30	Blue	20	800
p03	32	Purple	40	100
p04	40	Orange	70	700
NULL	NULL	NULL	NULL	NULL

INSERT INTO SUPPLIER_PART VALUES('S01',"P01",50);

INSERT INTO SUPPLIER_PART VALUES('S02','P02',150);

INSERT INTO SUPPLIER_PART VALUES('S03','P03',30);

INSERT INTO SUPPLIER_PART VALUES('S04','P04',100);

SCODE	PCODE	QTY
s01	p01	50
s02	p02	150
s03	p03	30
s04	p04	100

2. Write appropriate SQL Statement for the following:

a) Get the supplier number and part number in ascending order of supplier number.

$$\prod_{SCODE,PCODE}(\sigma_{SUPPLIER.SCODE=PART.PCODE}((SUPPLIER) \bowtie (PART)))$$

SELECT SCODE, PCODE FROM SUPPLIER, PART ORDER BY SUPPLIER.SCODE;

SCODE	PCODE
s01	p01
s02	p02
s03	p03
s04	p04

b) Get the details of supplier who operate from Bombay with turnover 50.

$$\prod_{SNAME}(\sigma_{SCITY = "BOMBAY" \wedge TURNOVER = 50}(SUPPLIER))$$

SELECT SNAME FROM SUPPLIER WHERE (SCITY = "BOMBAY" AND TURNOVER = 50);

SNAME
Tom

c) Get the total number of suppliers.

$$\prod_{COUNT(SCODE)}(\sigma(SUPPLIER))$$

SELECT COUNT(SCODE) AS TOTAL_NO_OF_SUPPLIER FROM SUPPLIER;

TOTAL_NO_OF_SUPPLIER
4

d) Get the part number weighing between 25 and 35.

$$\prod_{\text{PCODE}} (\sigma_{\text{WEIGH>25} \wedge \textit{WEIGH<35}} (\text{PART}))$$

SELECT PCODE AS PART FROM PART WHERE (WEIGH BETWEEN 25 AND 35);

PART
p01
p02
p03

e) Get the supplier number whose turnover is null.

$$\prod_{\text{SCODE}} (\sigma_{\text{TURNOVER IS NULL}} (\text{SUPPLIER}))$$

SELECT SCODE AS SUPPLIER_NUMBER FROM SUPPLIER WHERE TURNOVER IS NULL;

SUPPLIER_NUMBER
s02

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

$$\prod_{\text{PCODE}} (\sigma_{\text{COST IN (20, 30, 40)}} (\text{PART}))$$

SELECT PCODE FROM PART WHERE COST IN (20, 30, 40);

PCODE
p01
p02
p03
NULL

g) Get the total quantity of part 2 that is supplied.

$$\prod_{\text{SUM(QTY)}} (\sigma_{\text{PCODE='2'}} (\text{SUPPLIER\_PART}))$$

SELECT SUM(QTY) AS TOTAL_QUANTITY FROM SUPPLIER_PART WHERE PCODE = "2";

TOTAL_QUANTITY
NULL

h) Get the name of supplier who supply part 2.

$$\Pi_{SNAME}(\sigma_{PCODE='2'}(SUPPLIER \bowtie SUPPLIER\_PART))$$

SELECT SNAME FROM SUPPLIER WHERE SCODE IN (SELECT SCODE FROM SUPPLIER_PART WHERE PCODE = '2');

SNAME
Tony

i) Get the part number whose cost is greater than the average cost.

$$\Pi_{PCODE}(\sigma_{COST > (\Pi_{AVG(COST)}(PART))})$$

SELECT PCODE FROM PART WHERE COST > (SELECT AVG(COST) FROM PART);

PCODE
p04
NULL

j) Get the supplier number and turnover in descending order of turnover.

$$\Pi_{SNAME, TURNOVER}(SUPPLIER)$$

SELECT SNAME, TURNOVER FROM SUPPLIER ORDER BY TURNOVER DESC;

SNAME	TURNOVER
Jack	120
Peter	80
Tom	50
Tony	NULL

EXPERIMENT 7-8

Date: 24th March 2023

TITLE: Nested SQL queries or Subqueries

OBJECTIVE: To understand the use SQL Subquery

1. Create Tables (EMP and DEPT)

```
CREATE TABLE DEPT (
    DEPTNO INTEGER PRIMARY KEY,
   DNAME VARCHAR(20),
   LOC VARCHAR(20)
);
```

Field	Type	Null	Key	Default	Extra
DEPTNO	int	NO	PRI	NULL	
DNAME	varchar(20)	YES		NULL	
LOC	varchar(20)	YES		NULL	

```
CREATE TABLE EMP (
    EMPNO INTEGER PRIMARY KEY,
   EMPNAME VARCHAR(20),
   JOB VARCHAR(20),
   MGR INTEGER,
   HIREDATE DATE,
   SAL INTEGER,
   COMM INTEGER,
   DEPTNO INTEGER,
   FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO)
);
```

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
EMPNAME	varchar(20)	YES		NULL	
JOB	varchar(20)	YES		NULL	
MGR	int	YES		NULL	
HIREDATE	date	YES		NULL	
SAL	int	YES		NULL	
COMM	int	YES		NULL	
DEPTNO	int	YES	MUL	NULL	

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

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

INSERT INTO EMP VALUES(7369 ,'SMITH','CLERK',7902,'1980-12-17',800,NULL,20);
INSERT INTO EMP VALUES(7499 ,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);
INSERT INTO EMP VALUES(7521 ,'WARD','SALESMAN',7698,' 1981-02-22',1250,500,30);
INSERT INTO EMP VALUES(7566, 'JONES','MANAGER',7839,' 1981-04-02',2975 ,NULL,20);
INSERT INTO EMP VALUES(7654, 'MARTIN','SALESMAN',7698 ,'1981-09-28',1250,1400,30);
INSERT INTO EMP VALUES(7698, 'BLAKE','MANAGER',7839,' 1981-05-01',2850,NULL,30);
INSERT INTO EMP VALUES(7782, 'CLARK','MANAGER',7839 ,'1981-06-09' ,2450 ,NULL,10);
INSERT INTO EMP VALUES(7788, 'SCOTT','ANALYST', 7566 ,'1987-04-19' ,3000 ,NULL,20);
INSERT INTO EMP VALUES(7839, 'KING','PRESIDENT',NULL, '1981-11-17', 5000,NULL,10);
INSERT INTO EMP VALUES(7844, 'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);
INSERT INTO EMP VALUES(7876, 'ADAMS','CLERK',7788,'1987-05-23',1100 ,NULL,20);
INSERT INTO EMP VALUES(7900, 'JAMES','CLERK',7698, '1981-12-03', 950 ,NULL,30);
INSERT INTO EMP VALUES(7902, 'FORD','ANALYST',7566, '1981-12-03',3000 ,NULL,20);
INSERT INTO EMP VALUES(7934, 'MILLER','CLERK',7782, '1982-01-23',1300 ,NULL,10);

EMPNO	EMPNAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800	NULL	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
7782	CLARK	MANAGER	7839	1981-06-09	2450	NULL	10
7788	SCOTT	ANALYST	7566	1987-04-19	3000	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Write the Nested Queries for the following queries.

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

SELECT * FROM EMP WHERE SAL >

(SELECT SAL FROM EMP WHERE EMPNAME = "BLAKE");

EMPNO	EMPNAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7788	SCOTT	ANALYST	7566	1987-04-19	3000	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

SELECT EMPNAME FROM EMP WHERE JOB =

(SELECT JOB FROM EMP WHERE EMPNAME = "ALLEN");

EMPNAME
ALLEN
WARD
MARTIN
TURNER

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

SELECT EMPNAME FROM EMP WHERE SAL IN

(SELECT SAL FROM EMP WHERE EMPNAME IN ('FORD', 'SMITH')) ORDER BY EMPNAME DESC;

EMPNAME
SMITH
SCOTT
FORD

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

SELECT EMPNAME FROM EMP WHERE SAL >

(SELECT SAL FROM EMP WHERE EMPNAME = "ALLEN")

OR JOB = (SELECT JOB FROM EMP WHERE EMPNAME = "MILLER");

EMPNAME
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
ADAMS
JAMES
FORD
MILLER

e. Find the highest paid employee of sales department.

```
SELECT EMPNAME FROM EMP WHERE SAL =
(SELECT MAX(SAL) FROM EMP WHERE DEPTNO IN
(SELECT DEPTNO FROM DEPT WHERE DNAME = "SALES")
AND DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'SALES'));
```

EMPNAME	MAX(SAL)
ALLEN	1600
WARD	1250
MARTIN	1250
BLAKE	2850
TURNER	1500
JAMES	950

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

```
SELECT EMPNAME FROM EMP WHERE HIREDATE <
(SELECT MAX(HIREDATE) FROM EMP WHERE MGR IN
(SELECT EMPNO FROM EMP WHERE EMPNAME = "KING"));
```

	EMPNAME
▶	SMITH
	ALLEN
	WARD
	JONES
	BLAKE

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

```
SELECT EMPNAME, DEPTNO FROM EMP,
(SELECT MAX(SAL) AS M, DEPTNO AS D FROM EMP GROUP BY DEPTNO) as MD
WHERE SAL = MD.M AND DEPTNO = MD.D;
```

	EMPNAME	DEPTNO
1	BLAKE	30
2	SCOTT	20
3	KING	10
4	FORD	20

h. 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);
```

EMPNO	EMPNAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO

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

SELECT * FROM EMP AS E WHERE HIREDATE IN

(SELECT HIREDATE FROM EMP WHERE E.EMPNO <> EMPNO);

EMPNO	EMPNAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7900	JAMES	CLERK	7698	1981-12-03	950	<null>	30
7902	FORD	ANALYST	7566	1981-12-03	3000	<null>	20

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

SELECT EMPNAME FROM EMP E WHERE HIREDATE <

(SELECT HIREDATE FROM EMP WHERE EMPNO = E.MGR);

EMPNAME
SMITH
ALLEN
WARD
JONES
BLAKE
CLARK

EXPERIMENT 9

Date: 31st March 2023

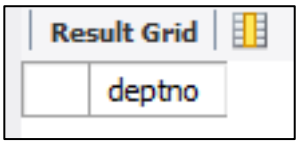
TITLE: Group by & Having clause

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

Write SQL Queries for the following queries (use EMP and DEPT table of previous experiment).

1. List the Deptno where there are no emps.

```
SELECT deptno FROM Emp GROUP BY deptno
HAVING COUNT(empno) = 0;
```



Result Grid

	deptno

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

```
SELECT COUNT(empno) AS TotalEmp, AVG(sal) AS AvgSal,
deptno, job FROM Emp GROUP BY deptno, job;
```

	TotalEmp	AvgSal	deptno	job
▶	1	2850.0000	20	NULL
	2	950.0000	20	CLERK
	4	1400.0000	30	SALESMAN
	1	2975.0000	20	MANAGER
	1	2850.0000	30	MANAGER
	1	2450.0000	10	MANAGER
	2	3000.0000	20	ANALYST
	1	5000.0000	10	PRESIDENT
	1	950.0000	30	CLERK
	1	1300.0000	10	CLERK

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

```
SELECT MAX(sal) AS MaxAvgSal FROM Emp WHERE sal IN
(SELECT AVG(sal) AS AvgSal FROM Emp WHERE
job != "President" GROUP BY job);
```

	MaxAvgSal
▶	3000

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(empno) >= 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 COUNT(empno) AS TotalEmp, deptno FROM Emp
GROUP BY deptno HAVING COUNT(empno) > 3;
```

TotalEmp	deptno
6	20
6	30

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

```
SELECT ename, deptno, sal FROM Emp
WHERE sal IN (SELECT MAX(sal) FROM
Emp GROUP BY deptno);
```

ename	deptno	sal
XYZ	20	2850
BLAKE	30	2850
SCOTT	20	3000
KING	10	5000
FORD	20	3000

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) FROM Emp
GROUP BY deptno HAVING AVG(sal) <
(SELECT AVG(sal) FROM Emp);
```

deptno	avg(sal)
30	1566.6667

EXPERIMENT 10

Date: 7th April 2023

TITLE: Joins in SQL

AIM: 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.

Write SQL Queries using Join for the following queries.

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

   SELECT e1.* FROM Emp e1 INNER JOIN Emp e2

   ON e2.ename = "Blake" WHERE e1.sal > e2.sal;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	1981-04-02	2975	NULL	20
7788	SCOTT	ANALYST	7566	1987-04-19	3000	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000	NULL	10
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

   SELECT e1.ename FROM Emp e1 INNER JOIN Emp e2

   ON e1.job = e2.job WHERE e2.ename = "Allen";

ename
ALLEN
WARD
MARTIN
TURNER

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

   SELECT e1.ename FROM Emp e1 INNER JOIN Emp e2

   ON e1.sal = e2.sal WHERE e1.ename = "Ford"

   OR e2.ename = "Smith" ORDER BY e1.ename DESC;

ename
SMITH
SCOTT
FORD

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

```
SELECT e1.ename FROM Emp e1 INNER JOIN Emp e2
ON e1.job = e2.job WHERE e2.ename = "Miller"
UNION SELECT e3.ename FROM Emp e3 INNER JOIN
Emp e4 ON e3.sal > e4.sal WHERE e4.ename = "Allen";
```

ename
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
ADAMS
JAMES
FORD
MILLER

5. Find the highest paid employee of sales department.

```
SELECT e2.ename, e2.sal FROM Emp e2 INNER JOIN
(SELECT MAX(sal) AS M, e1.deptno AS D
FROM Emp e1 INNER JOIN Dept d1 ON
e1.deptno = d1.deptno WHERE d1.dname = "Sales" GROUP BY D)
AS tab1 ON e2.deptno = tab1.D WHERE e2.sal  = tab1.M;
```

ename	sal
BLAKE	2850

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

```
SELECT e1.ename FROM Emp e1 INNER JOIN
(SELECT MAX(e2.hiredate) AS MaxH FROM Emp e2
INNER JOIN Emp e3 ON e2.mgr =
e3.empno WHERE e3.ename = "King")
AS tab ON e1.hiredate < tab.MaxH;
```

ename
SMITH
ALLEN
WARD
JONES
BLAKE

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

```
SELECT e1.ename, e1.sal, e1.deptno FROM Emp e1
INNER JOIN (SELECT MAX(e2.sal) AS MaxSal, e2.deptno
FROM Emp e2 GROUP BY e2.deptno) AS tab
ON e1.sal = tab.MaxSal;
```

ename	sal	deptno
XYZ	2850	20
BLAKE	2850	30
SCOTT	3000	20
KING	5000	10
FORD	3000	20

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

```
SELECT e1.* FROM Emp e1 INNER JOIN
(SELECT (MAX(sal) + MIN(sal) ) / 2 AS S
FROM Emp) AS e2 ON e1.sal = e2.S;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
SELECT e1.* FROM Emp e1 INNER JOIN
Emp e2 ON e1.hiredate = e2.hiredate
WHERE e1.ename != e2.ename;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7900	JAMES	CLERK	7698	1981-12-03	950	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
SELECT e1.ename FROM Emp e1 INNER
JOIN Emp e2 ON e1.mgr = e2.empno
WHERE e1.hiredate < e2.hiredate;
```

ename
SMITH
ALLEN
WARD
JONES
BLAKE
CLARK