

4. Create a table **SALES** with columns SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_BIRTH, SALARY, SALES_AMOUNT and COMMISSION. Minimum age for joining the company must be 18 Yrs. Default value for Commission should be 0. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL constraints on the table. Insert 10 records with data except commission. Manager of Manager can be NULL. Write the following SQL queries:
- Display the details of Sales Persons whose salary is more than Average salary in the company.
 - Update commission as 20% of Sales Amount.
 - Display SNO, SNAME, MANAGER_NAME, SALARY, COMMISSION, MANAGER_SALARY of the sales persons getting sum of salary and commission more than salary of manager. (Self-join)
 - Display the records of employees who finished the service of 10 years.

4.1 Creating the table and describing its structure

```
CREATE TABLE SALES
(
    SNO VARCHAR(5) PRIMARY KEY,
    SNAME VARCHAR(15) NOT NULL,
    MANAGER_NAME VARCHAR(15),
    JOIN_DATE DATE NOT NULL,
    DATE_OF_BIRTH DATE NOT NULL,
    SALARY NUMBER(10, 2) NOT NULL,
    SALES_AMOUNT NUMBER(10,2) NOT NULL,
    COMMISSION NUMBER(8, 2) DEFAULT 0,
    CHECK (JOIN_DATE-DATE_OF_BIRTH >= 18)
);
```

Output

Table created.

Description of SALES table.

DESCRIBE SALES; or DESC SALES;

Name	Null?	Type
-----	-----	-----
SNO	NOT NULL	VARCHAR2(5)
SNAME	NOT NULL	VARCHAR2(15)
MANAGER_NAME		VARCHAR2(15)
JOIN_DATE	NOT NULL	DATE
DATE_OF_BIRTH	NOT NULL	DATE
SALARY	NOT NULL	NUMBER(10,2)
SALES_AMOUNT	NOT NULL	NUMBER(10,2)
COMMISSION		NUMBER(8,2)

4.2 Loading data into the table

```

INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0001', 'TARUN', 'ALBERT', '02-JAN-2022', '10-MAR-1995', 25000, 1000);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0002', 'ASHRAY', 'KUMAR', '01-JAN-2010', '18-OCT-1980', 100000, 10000);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0003', 'NAYANA', 'ASHRAY', '01-NOV-2021', '21-SEP-2000', 22500, 1250);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0004', 'ALBERT', 'ASHRAY', '12-MAY-2011', '19-AUG-1984', 90000, 7700);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0005', 'ABHINAY', 'ASHRAY', '07-MAR-2020', '01-JUN-2000', 20000, 900);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0006', 'ADVAITH', 'ALBERT', '22-FEB-2021', '11-JUL-1998', 23000, 1200);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0007', 'ISHA', 'ASHRAY', '30-APR-2016', '01-OCT-1987', 57000, 4200);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0008', 'KARAN', 'ASHRAY', '02-MAR-2017', '16-SEP-1994', 37500, 3200);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0009', 'RISHAN', 'ALBERT', '18-JUN-2018', '11-SEP-1999', 32000, 2200);
INSERT INTO SALES (SNO, SNAME, MANAGER_NAME, JOIN_DATE, DATE_OF_BIRTH, SALARY,
SALES_AMOUNT) VALUES
('S0010', 'VARSHITH', 'ASHRAY', '12-JAN-2019', '31-DEC-1005', 25000, 1000);

```

Output

10 rows created.

Displaying the values inserted into the SALES table.

```

SELECT *
FROM SALES;

```

SNO	SNAME	MANAGER_NAME	JOIN_DATE	DATE_OF_B	SALARY	SALES_AMOUNT	COMMISSION
S0001	TARUN	ALBERT	02-JAN-22	10-MAR-95	25000	1000	0
S0002	ASHRAY	KUMAR	01-JAN-10	18-OCT-80	100000	10000	0
S0003	NAYANA	ASHRAY	01-NOV-21	21-SEP-00	22500	1250	0
S0004	ALBERT	ASHRAY	12-MAY-11	19-AUG-84	90000	7700	0
S0005	ABHINAY	ASHRAY	07-MAR-20	01-JUN-00	20000	900	0
S0006	ADVAITH	ALBERT	22-FEB-21	11-JUL-98	23000	1200	0
S0007	ISHA	ASHRAY	30-APR-16	01-OCT-87	57000	4200	0
S0008	KARAN	ASHRAY	02-MAR-17	16-SEP-94	37500	3200	0
S0009	RISHAN	ALBERT	18-JUN-18	11-SEP-99	32000	2200	0
S0010	VARSHITH	ASHRAY	12-JAN-19	31-DEC-05	25000	1000	0

10 rows selected.

4.3 Queries

4.3.1 Display the details of Sales Persons whose salary is more than Average salary in the company.

```
SELECT *
FROM SALES
WHERE SALARY >
(
    SELECT AVG(SALARY)
    FROM SALES
);
```

Output

SNO	SNAME	MANAGER_NAME	JOIN_DATE	DATE_OF_B	SALARY	SALES_AMOUNT	COMMISSION
S0002	ASHRAY	KUMAR	01-JAN-10	18-OCT-80	100000	10000	0
S0004	ALBERT	ASHRAY	12-MAY-11	19-AUG-84	90000	7700	0
S0007	ISHA	ASHRAY	30-APR-16	01-OCT-87	57000	4200	0

4.3.2 Update commission as 20% of Sales Amount.

```
UPDATE SALES
SET COMMISSION = SALES_AMOUNT*0.20;
```

Output

10 rows updated.

SNO	SNAME	MANAGER_NAME	JOIN_DATE	DATE_OF_B	SALARY	SALES_AMOUNT	COMMISSION
S0001	TARUN	ALBERT	02-JAN-22	10-MAR-95	25000	1000	200
S0002	ASHRAY	KUMAR	01-JAN-10	18-OCT-80	100000	10000	2000
S0003	NAYANA	ASHRAY	01-NOV-21	21-SEP-00	22500	1250	250
S0004	ALBERT	ASHRAY	12-MAY-11	19-AUG-84	90000	7700	1540
S0005	ABHINAY	ASHRAY	07-MAR-20	01-JUN-00	20000	900	180
S0006	ADVAITH	ALBERT	22-FEB-21	11-JUL-98	23000	1200	240
S0007	ISHA	ASHRAY	30-APR-16	01-OCT-87	57000	4200	840
S0008	KARAN	ASHRAY	02-MAR-17	16-SEP-94	37500	3200	640
S0009	RISHAN	ALBERT	18-JUN-18	11-SEP-99	32000	2200	440
S0010	VARSHITH	ASHRAY	12-JAN-19	31-DEC-05	25000	1000	200

4.3.3 Display SNO, SNAME, MANAGER_NAME, SALARY, COMMISSION, MANAGER_SALARY of the sales persons getting sum of salary and commission more than salary of manager. (Self-join)

```
SELECT S1.SNO, S1.SNAME, S1.MANAGER_NAME, S1.SALARY, S1.COMMISSION,
SUM(S1.SALARY+S1.COMMISSION) AS NET_SALARY
FROM SALES S1, SALES S2
WHERE S1.SNAME=S2.MANAGER_NAME
GROUP BY (S1.SNO, S1.SNAME, S1.MANAGER_NAME, S1.SALARY, S1.COMMISSION)
HAVING SUM(S1.SALARY+S1.COMMISSION)>SUM(S2.SALARY);
```

Output

SNO	SNAME	MANAGER_NAME	SALARY	COMMISSION	NET_SALARY
S0002	ASHRAY	KUMAR	100000	2000	612000
S0004	ALBERT	ASHRAY	90000	1540	274620

