SHEET NO - 04

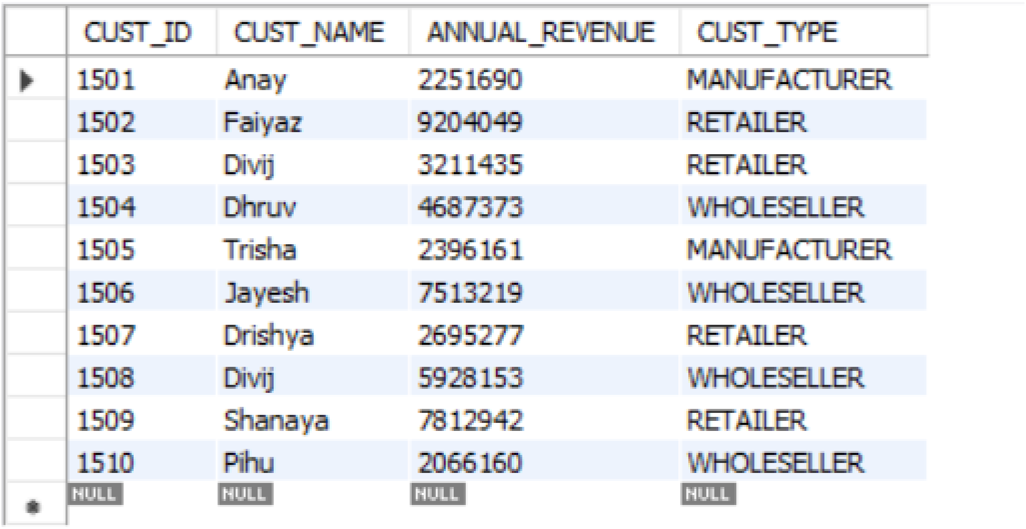
Name : Raksha Pahariya
Roll No : 2021CSB029
Subject : DBMS Lab

Assignment No: 8

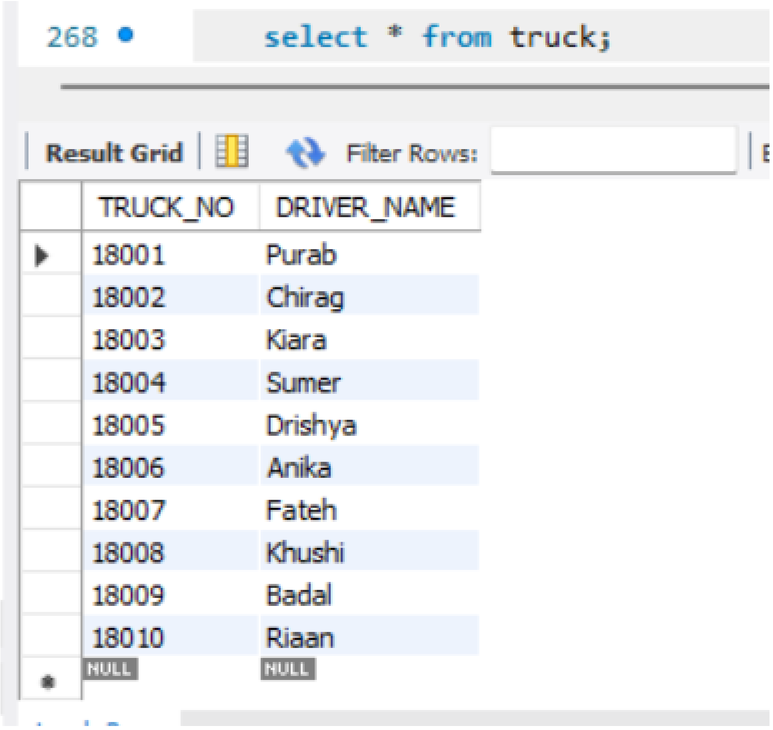
Write the SQL commands to create the following relational schemas:

```
-- Write the SQL commands to create the following relational
schemas:
-- Customer ( cust_id, cust_name, annual_revenue, cust_type )
-- Cust_id must be between 100 and 10,000.
-- Cust_type must be ‘MANUFACTURER’, ‘WHOLESALER’, or
‘RETAILER’.
-- Shipment ( shipment_no, cust_id, weight, truck_no,
destination, ship_date )
-- Foreign keys: cust_id references customer on delete cascade,
truck_no references truck
-- on delete set null, destination references city.
-- Weight must be under 1000.
-- Truck ( truck_no, driver_name )
-- City ( city_name, population)
```

```
CREATE TABLE CUSTOMER(
    CUST_ID NUMERIC(5) PRIMARY KEY CHECK(
        CUST_ID >= 100
        AND CUST_ID < 10000
    ),
    CUST_NAME VARCHAR(20),
    ANNUAL_REVENUE NUMERIC(8),
    CUST_TYPE VARCHAR(15) CHECK(
        CUST_TYPE IN ('MANUFACTURER', 'WHOLESELLER', 'RETAILER')
    )
);

CREATE TABLE TRUCK(
    TRUCK_NO NUMERIC(5) PRIMARY KEY,
    DRIVER_NAME VARCHAR(20)
);

CREATE TABLE CITY(
    CITY_NAME VARCHAR(20) PRIMARY KEY,
    POPULATION NUMERIC(8)
);

CREATE TABLE shipment(
    shipment_no varchar(6),
    cust_id numeric(6) REFERENCES customer(cust_id) ON DELETE
CASCADE,
    weight numeric(4) CHECK(weight < 1000),
    truck_no numeric(5) REFERENCES truck(truck_no) ON DELETE
    SET
        NULL,
        destination varchar(20) REFERENCES city(city_name),
        ship_date date,
        PRIMARY KEY(shipment_no, cust_id)
);
```

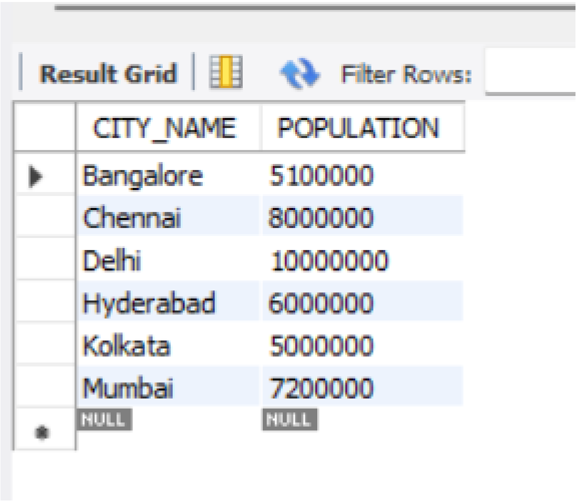


	CUST_ID	CUST_NAME	ANNUAL_REVENUE	CUST_TYPE
▶	1501	Anay	2251690	MANUFACTURER
	1502	Faiyaz	9204049	RETAILER
	1503	Divij	3211435	RETAILER
	1504	Dhruv	4687373	WHOLESELLER
	1505	Trisha	2396161	MANUFACTURER
	1506	Jayesh	7513219	WHOLESELLER
	1507	Drishya	2695277	RETAILER
	1508	Divij	5928153	WHOLESELLER
	1509	Shanaya	7812942	RETAILER
	1510	Pihu	2066160	WHOLESELLER
*	NULL	NULL	NULL	NULL

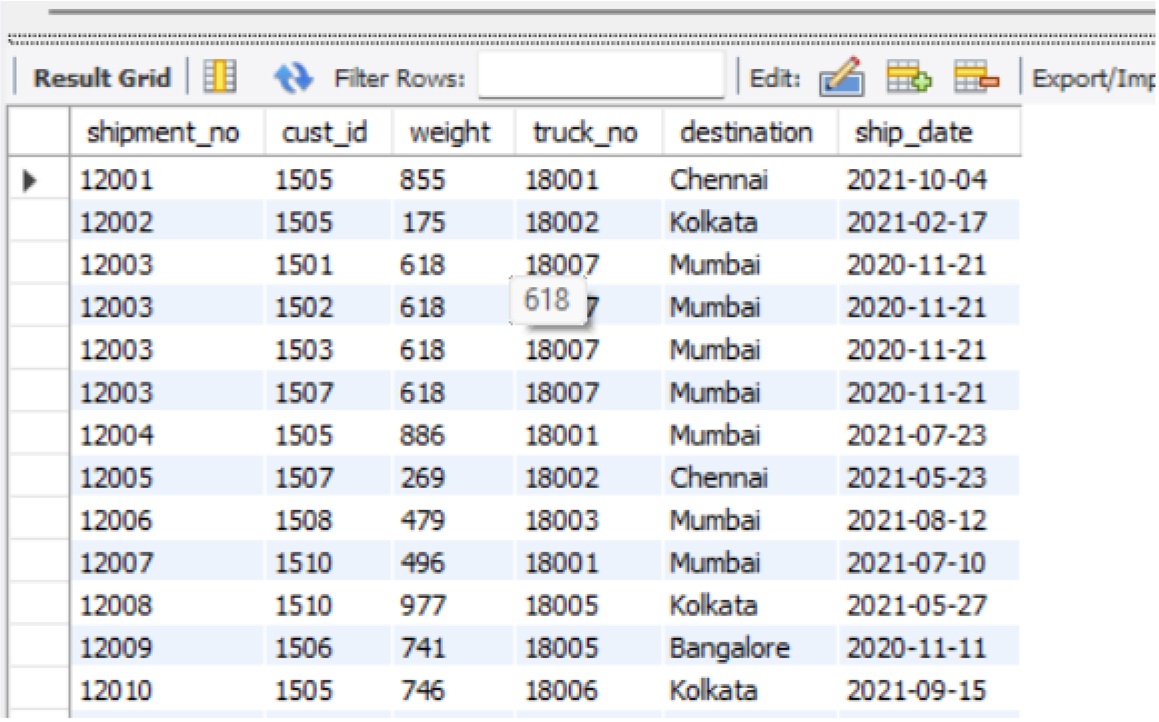


```sql
select * from truck;
```

TRUCK_NO	DRIVER_NAME
18001	Purab
18002	Chirag
18003	Kiara
18004	Sumer
18005	Drishya
18006	Anika
18007	Fateh
18008	Khushi
18009	Badal
18010	Riaan
NULL	NULL

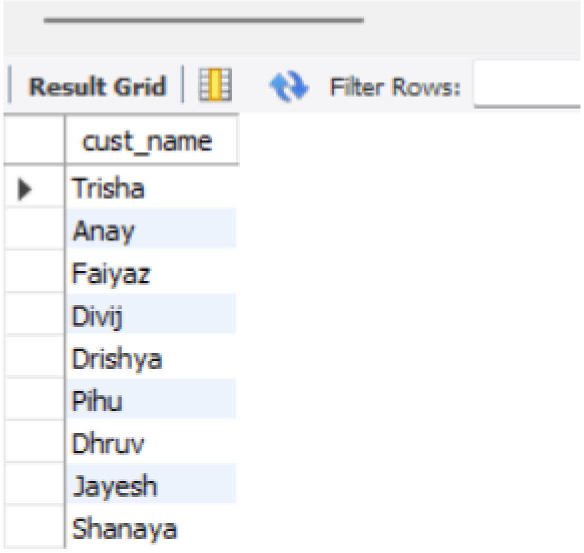


CITY_NAME	POPULATION
Bangalore	5100000
Chennai	8000000
Delhi	10000000
Hyderabad	6000000
Kolkata	5000000
Mumbai	7200000
NULL	NULL



shipment_no	cust_id	weight	truck_no	destination	ship_date
12001	1505	855	18001	Chennai	2021-10-04
12002	1505	175	18002	Kolkata	2021-02-17
12003	1501	618	18007	Mumbai	2020-11-21
12003	1502	618	7	Mumbai	2020-11-21
12003	1503	618	18007	Mumbai	2020-11-21
12003	1507	618	18007	Mumbai	2020-11-21
12004	1505	886	18001	Mumbai	2021-07-23
12005	1507	269	18002	Chennai	2021-05-23
12006	1508	479	18003	Mumbai	2021-08-12
12007	1510	496	18001	Mumbai	2021-07-10
12008	1510	977	18005	Kolkata	2021-05-27
12009	1506	741	18005	Bangalore	2020-11-11
12010	1505	746	18006	Kolkata	2021-09-15

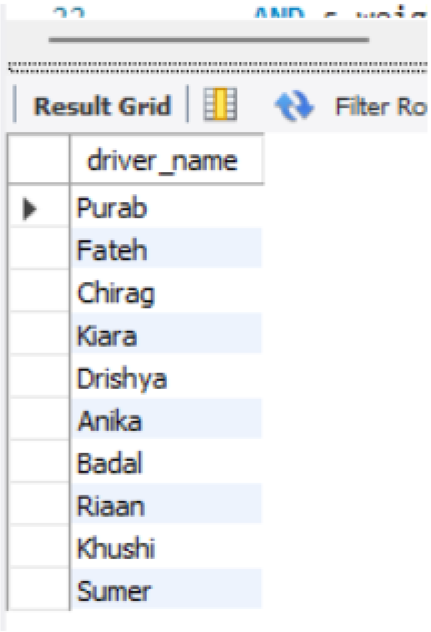
```
SELECT
    DISTINCT cust_name
FROM
    shipment s,
    customer c
WHERE
    s.cust_id = c.cust_id
    AND destination IN ('Chennai', 'Kolkata', 'Mumbai');
```



cust_name
Trisha
Anay
Faiyaz
Divij
Drishya
Pihu
Dhruv
Jayesh
Shanaya

2. List the names of the driver who have delivered shipments weighing over 200 pounds
query:

```
SELECT
DISTINCT driver_name
FROM
shipment s,
truck t
WHERE
s.truck_no = t.truck_no
AND s.weight > 200;
```

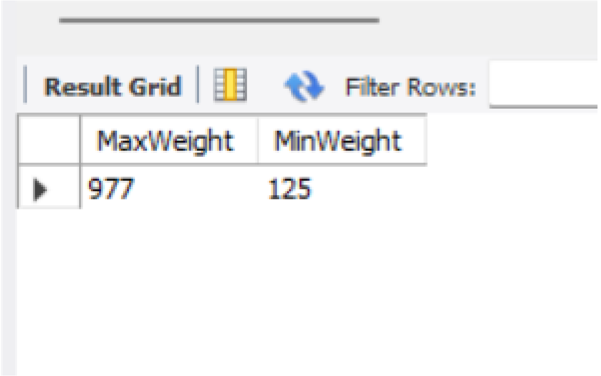


	driver_name
▶	Purab
	Fateh
	Chirag
	Kiara
	Drishya
	Anika
	Badal
	Riaan
	Khushi
	Sumer

3. Retrieve the maximum and minimum weights of the shipments. name the output asMax_Weight and Min_Weight respectively.

query:

```
SELECT
max(weight) AS MaxWeight,
min(weight) AS MinWeight
FROM
Shipment;
```

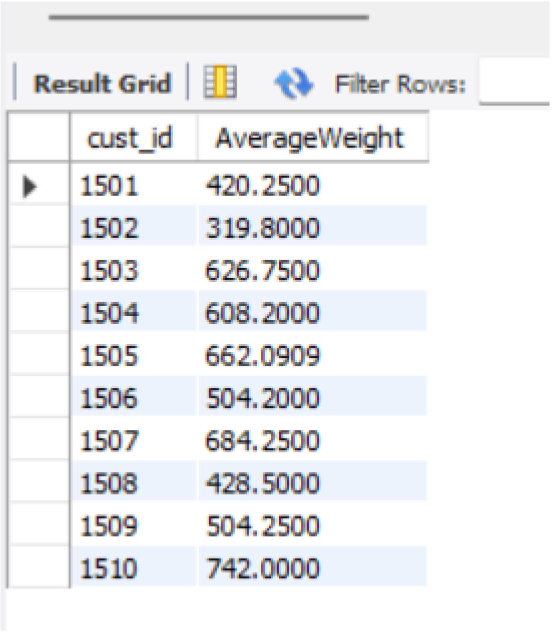


	MaxWeight	MinWeight
▶	977	125

4. For each customer, what is the average weight of package sent by the customer?

query:

```
SELECT
cust_id,
avg(weight) AS AverageWeight
FROM
shipment
GROUP BY
cust_id
ORDER BY
Cust_id;
```



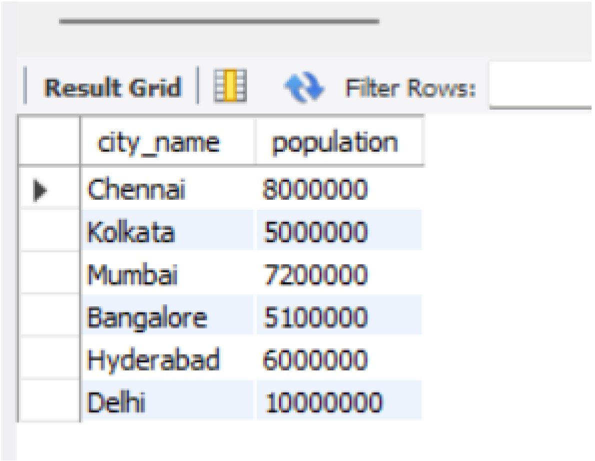
cust_id	AverageWeight
1501	420.2500
1502	319.8000
1503	626.7500
1504	608.2000
1505	662.0909
1506	504.2000
1507	684.2500
1508	428.5000
1509	504.2500
1510	742.0000

5. List the names and populations of cities that have received a shipment weighing over 100 pounds.

query:

```
SELECT
city_name,
population
FROM
shipment s,
city c
```

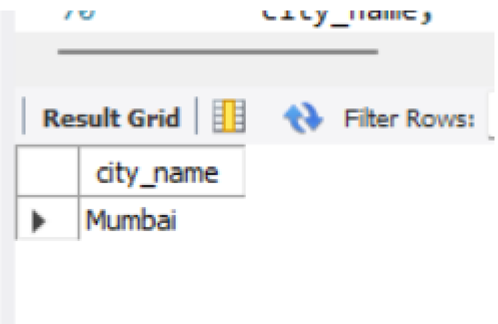
```
WHERE
s.destination = c.city_name
AND weight > 100
GROUP BY
City_name;
```



city_name	population
Chennai	8000000
Kolkata	5000000
Mumbai	7200000
Bangalore	5100000
Hyderabad	6000000
Delhi	10000000

6. List cities that have received shipments from every customer.
query:

```
SELECT
city_name
FROM
shipment s,
city c
WHERE
s.destination = c.city_name
GROUP BY
city_name
HAVING
count(DISTINCT cust_id) = (
SELECT
count(*)
FROM
customer
);
```



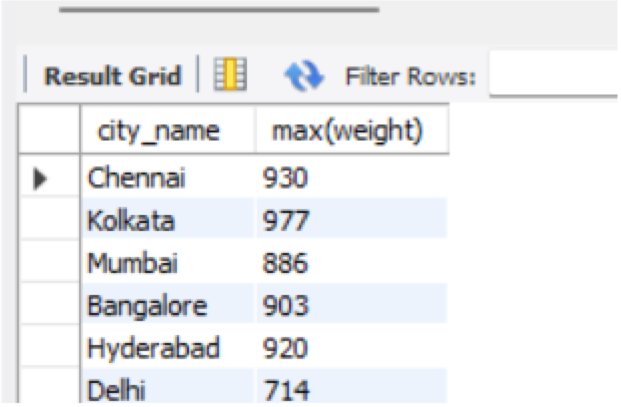
```
city_name;
```

	city_name
▶	Mumbai

7. For each city, what is the maximum weight of a package sent to that city?

query:

```
SELECT
city_name,
max(weight)
FROM
city c,
shipment s
WHERE
c.city_name = s.destination
GROUP BY
City_name;
```

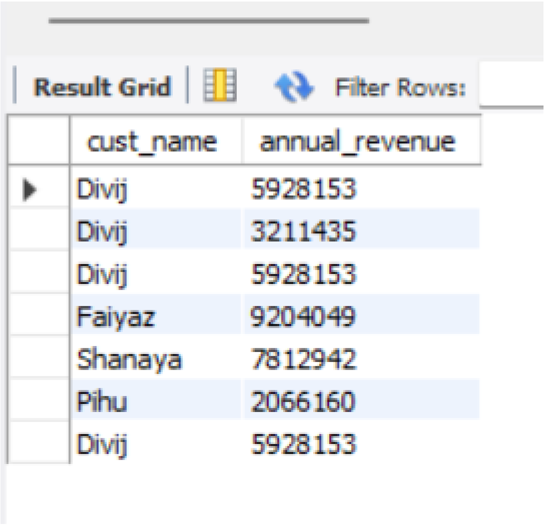


	city_name	max(weight)
▶	Chennai	930
	Kolkata	977
	Mumbai	886
	Bangalore	903
	Hyderabad	920
	Delhi	714

8. List the name and annual revenue of customers whose shipments have been delivered by truck driver 'Kiara'.

query:

```
SELECT
cust_name,
annual_revenue
FROM
customer c,
shipment s,
truck t
WHERE
c.cust_id = s.cust_id
AND s.truck_no = t.truck_no
AND driver_name = 'Kiara';
```



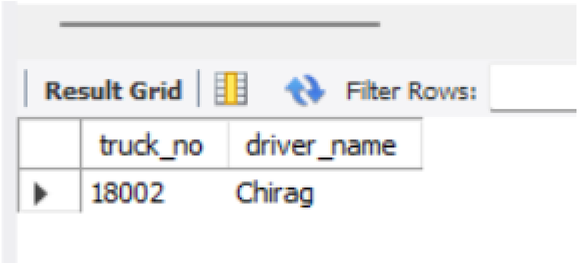
cust_name	annual_revenue
Divij	5928153
Divij	3211435
Divij	5928153
Faiyaz	9204049
Shanaya	7812942
Pihu	2066160
Divij	5928153

9. List drivers who have delivered shipments to every city.

query:

```
SELECT
t.truck_no,
t.driver_name
FROM
shipment s,
truck t
WHERE
s.truck_no = t.truck_no
```

```
GROUP BY
t.truck_no
HAVING
count(DISTINCT(destination)) = (
SELECT
count(*)
FROM
city
);
```

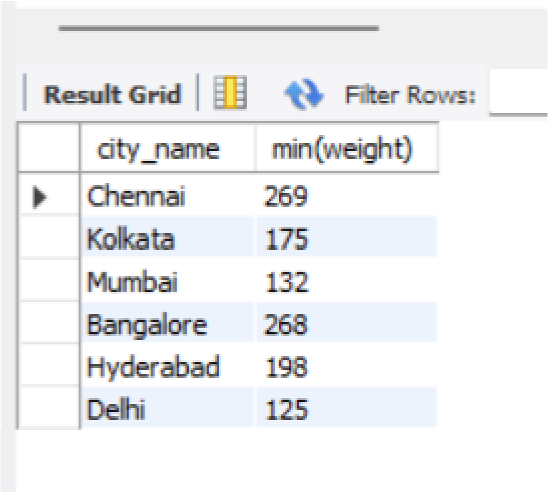


truck_no	driver_name
18002	Chirag

10. For each city, with a population of over 1 million, what is the minimum weight of a package sent to that city.

query:

```
SELECT
city_name,
min(weight)
FROM
shipment,
city
WHERE
destination = city_name
AND population > 1000000
GROUP BY
city_name;
```



city_name	min(weight)
Chennai	269
Kolkata	175
Mumbai	132
Bangalore	268
Hyderabad	198
Delhi	125

Assignment No: 9

Write SQL commands to create the following tables as well as to insert sufficient number of values in the tables:

EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
EMPNO must be between 7000 and 8000.
ENAME must not exceed 10 characters.
JOB must be in (‘Clerk’,‘Salesman’,‘Manager’,‘Analyst’,‘President’).
MGR is the manager’s EMPNO.
COMM must be under 1500 and defaults to 0.

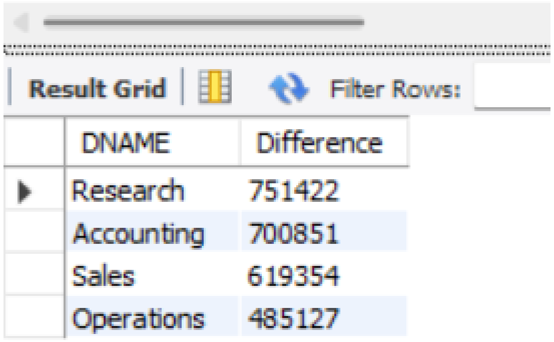
```
CREATE TABLE EMP(
        EMPNO numeric(10) PRIMARY KEY CHECK (
        EMPNO BETWEEN 7000
        AND 8000
        ),
        ENAME varchar(10),
        JOB varchar(10) CHECK (
        JOB IN(
        'President',
        'Manager',
        'Clerk',
        'Salesman',
        'Analyst'
        )
        ),
        MGR numeric(10),
        HIREDATE date,
        SAL numeric(10),
        COMM numeric(5) DEFAULT 0 CHECK(COMM < 1500),
        DEPTNO varchar(5) REFERENCES DEPT(DEPTNO)
        );
```

DEPT (DEPTNO, DNAME, LOC)
DEPTNO must start with ‘D’.
DNAME must be ‘Accounting’ or ‘Sales’ or ‘Research’ or ‘Operations’.

```
CREATE TABLE DEPT(
        DEPTNO varchar(5) PRIMARY KEY CHECK (DEPTNO LIKE 'D%'),
        DNAME varchar(10) CHECK (
        DNAME IN ('Accounting', 'Sales', 'Research', 'Operations')
        ),
        LOC varchar(10)
);
```

1. Display the difference between the highest and lowest salaries of each department in descending order. Label the column as “Difference”.
Query:

```
SELECT
DNAME,
max(SAL) - min(SAL) AS Difference
FROM
DEPT d,
EMP e
WHERE
d.DEPTNO = e.DEPTNO
GROUP BY
D.DEPTNO
ORDER BY
Difference DESC;
```

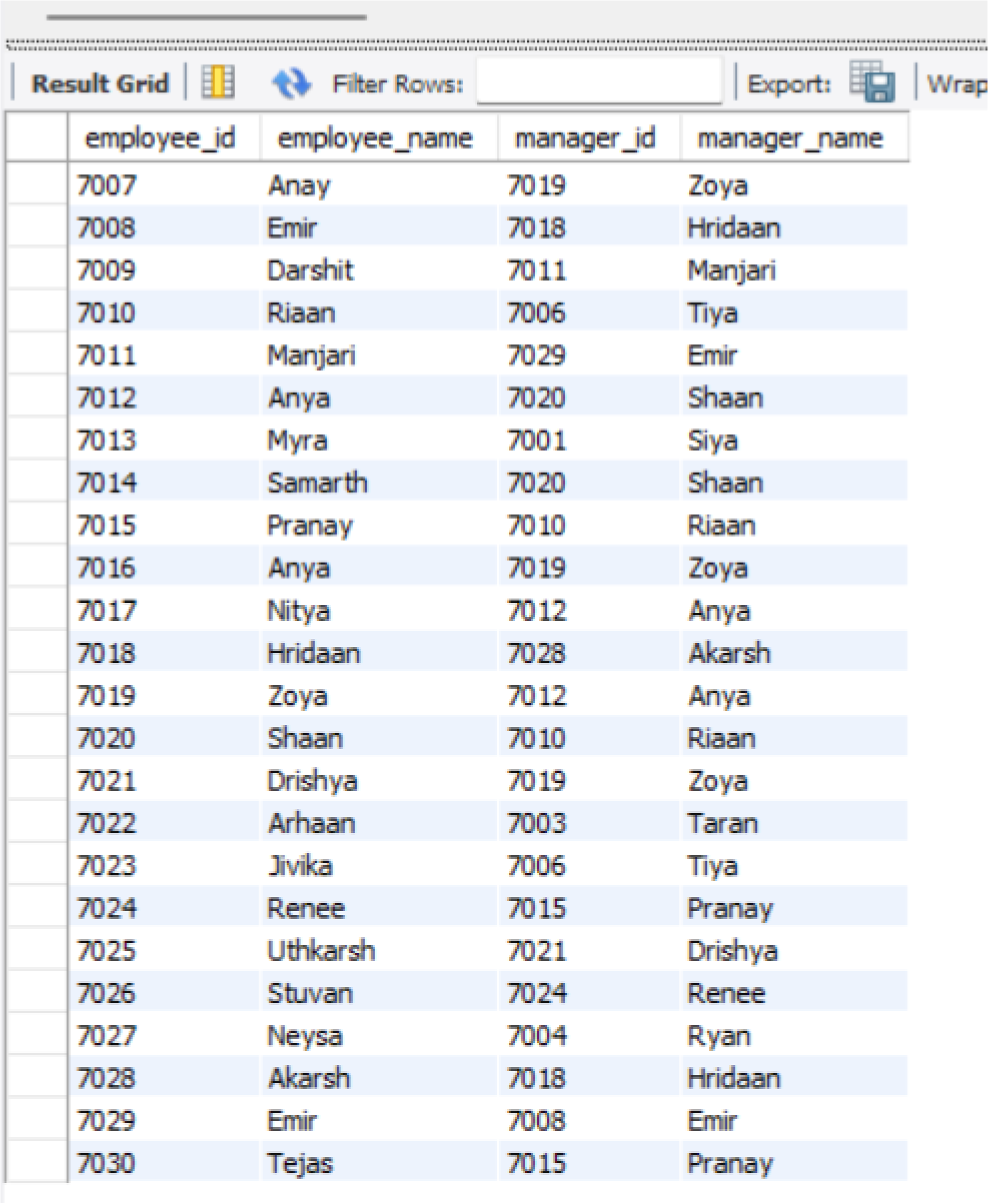


DNAME	Difference
Research	751422
Accounting	700851
Sales	619354
Operations	485127

2. List all the employees’ employee numbers and names along with their immediate managers’ employee numbers and names.

query:

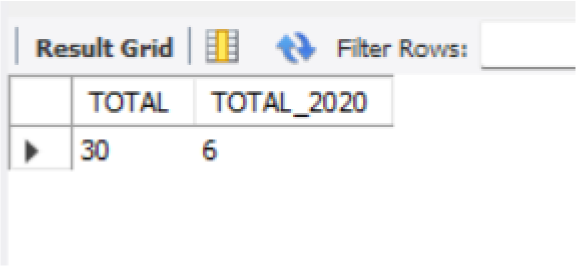
```
SELECT
a.EMPNO AS employee_id,
a.ENAME AS employee_name,
b.EMPNO AS manager_id,
b.ENAME AS manager_name
FROM
EMP a,
EMP b
WHERE
a.MGR = b.EMPNO;
```



employee_id	employee_name	manager_id	manager_name
7007	Anay	7019	Zoya
7008	Emir	7018	Hridaan
7009	Darshit	7011	Manjari
7010	Riaan	7006	Tiya
7011	Manjari	7029	Emir
7012	Anya	7020	Shaan
7013	Myra	7001	Siya
7014	Samarth	7020	Shaan
7015	Pranay	7010	Riaan
7016	Anya	7019	Zoya
7017	Nitya	7012	Anya
7018	Hridaan	7028	Akarsh
7019	Zoya	7012	Anya
7020	Shaan	7010	Riaan
7021	Drishya	7019	Zoya
7022	Arhaan	7003	Taran
7023	Jivika	7006	Tiya
7024	Renee	7015	Pranay
7025	Uthkarsh	7021	Drishya
7026	Stuvan	7024	Renee
7027	Neysa	7004	Ryan
7028	Akarsh	7018	Hridaan
7029	Emir	7008	Emir
7030	Tejas	7015	Pranay

3. Create a query that will display the total number of employees and the total number of employees who were hired only in 2020. Give the column headings as "TOTAL" and "TOTAL_2020" respectively.
query:

```
SELECT
TOTAL,
TOTAL_2020
FROM
(
SELECT
count(*) AS TOTAL
FROM
EMP
) AS hd1,
(
SELECT
count(*) AS TOTAL_2020
FROM
EMP
WHERE
EXTRACT(
YEAR
FROM
HIREDATE
) = 2020
) AS hd2;
```

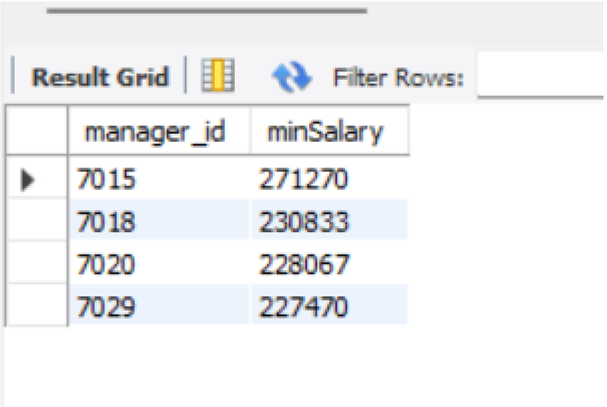


TOTAL	TOTAL_2020
30	6

4. Display the manager number and the salary of the lowest-paid employee under that manager. Exclude anyone

whose manager is not known. Exclude any group where the minimum salary is less than 1000. Sort the output in descending order of salary.
query:

```
SELECT
MGR AS manager_id,
min(SAL) AS minSalary
FROM
EMP
GROUP BY
MGR
HAVING
MGR IS NOT NULL
AND min(SAL) < 300000
ORDER BY
minSalary DESC;
```

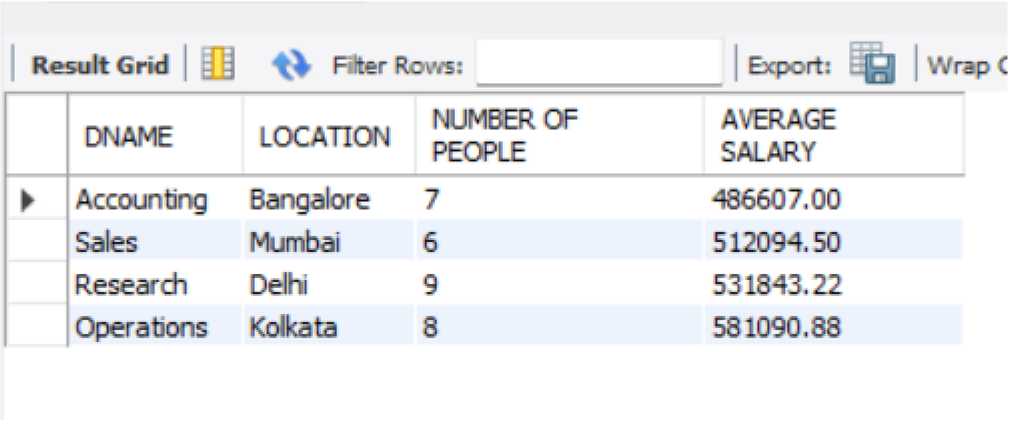


manager_id	minSalary
7015	271270
7018	230833
7020	228067
7029	227470

5. Assume that there are some departments where no employee is assigned. Now, write a query to display the department name, location name, number of employees, and the average salary for all the employees in that department.

Label the columns as “DNAME”, “LOCATION”, “NUMBER OF PEOPLE”, and “AVERAGE SALARY” respectively. Round the average salary to two decimal places. The outcome of the query must include the details of the departments where no employee is assigned and in that case, the “AVERAGE SALARY” for that department is to be displayed as 0 (zero).
query:

```
SELECT
DNAME AS "DNAME",
LOC AS "LOCATION",
count(*) AS "NUMBER OF PEOPLE",
coalesce(round(avg(SAL), 2), 0.0) AS "AVERAGE SALARY"
FROM
DEPT
LEFT JOIN EMP ON (EMP.DEPTNO = DEPT.DEPTNO)
GROUP BY
DNAME,
LOC;
```



DNAME	LOCATION	NUMBER OF PEOPLE	AVERAGE SALARY
Accounting	Bangalore	7	486607.00
Sales	Mumbai	6	512094.50
Research	Delhi	9	531843.22
Operations	Kolkata	8	581090.88