**Sheet No : 4**

**Assignment No : 08**

1. **Creation of Tables:**
2. **Creation of Customer table:**

**Query:**

CREATE TABLE IF NOT EXISTS Customer (

cust\_id INT UNSIGNED,

cust\_name VARCHAR(50),

annual\_revenue INT UNSIGNED,

cust\_type ENUM('MANUFACTURER', 'WHOLESALER', 'RETAILER'),

PRIMARY KEY (cust\_id),

CHECK (cust\_id BETWEEN 100 AND 10000)

);

**Inserting values:**

INSERT INTO Customer (cust\_id, cust\_name, annual\_revenue, cust\_type)

VALUES

(1001, 'Aritra Bandyopdhyay', 500000, 'MANUFACTURER'),

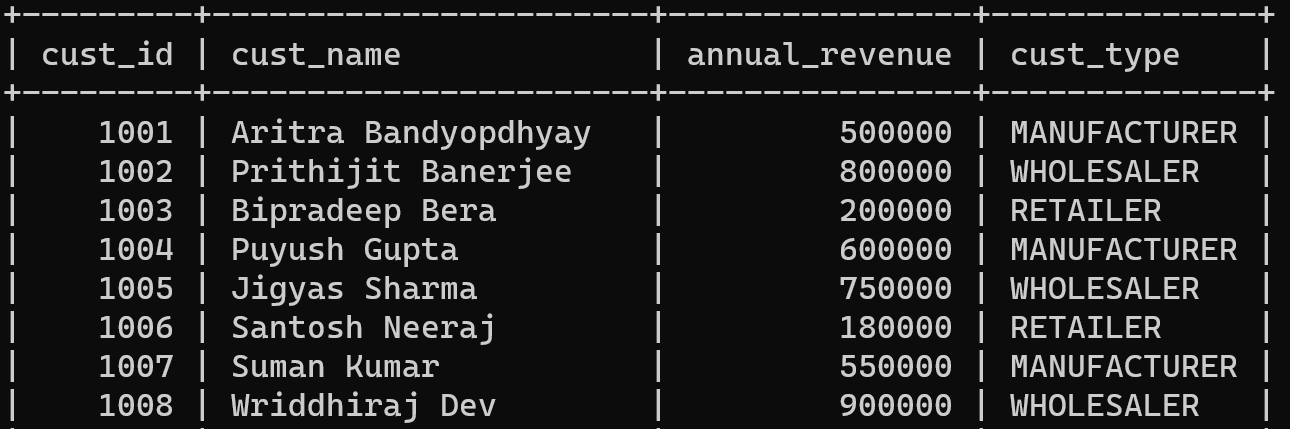
(1002, 'Prithijit Banerjee', 800000, 'WHOLESALER'),

(1003, 'Bipradeep Bera', 200000, 'RETAILER'),

(1004, 'Puyush Gupta', 600000, 'MANUFACTURER'),

(1005, 'Jigyas Sharma', 750000, 'WHOLESALER');

**Snapshot of the table:**



1. **Creation of Truck table:**

**Query:**

CREATE TABLE IF NOT EXISTS Truck (

truck\_no CHAR(10),

driver\_name VARCHAR(50),

CHECK(truck\_no LIKE 'T%'),

PRIMARY KEY (truck\_no)

);

**Inserting values:**

INSERT INTO Truck (truck\_no, driver\_name)

VALUES

('TNO0000001','Prashanjit Basu'),

('TNO0000010','Tanis Ahamad'),

('TNO0000011','Shubham Ghosh'),

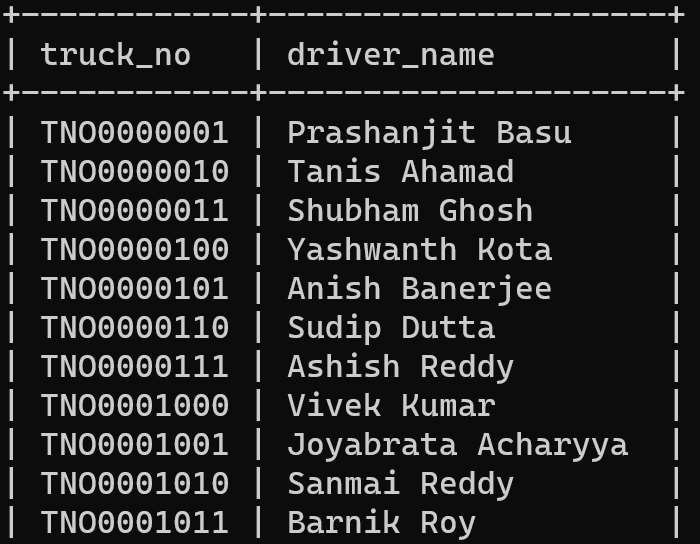
('TNO0000100','Yashwanth Kota'),

('TNO0000101','Anish Banerjee'),

('TNO0000110','Sudip Dutta'),

('TNO0000111','Ashish Reddy');

**Snapshot of the table:**



1. **Creation of City table:**

**Query:**

CREATE TABLE IF NOT EXISTS City (

city\_name VARCHAR(50),

population INT UNSIGNED,

PRIMARY KEY (city\_name)

);

**Inserting values:**

INSERT INTO City (city\_name, population)

VALUES

('Mumbai', 12442373),

('Delhi', 11007835),

('Bangalore', 8443675),

('Kolkata', 4949498),

('Chennai', 4646732),

('Hyderabad', 6759413),

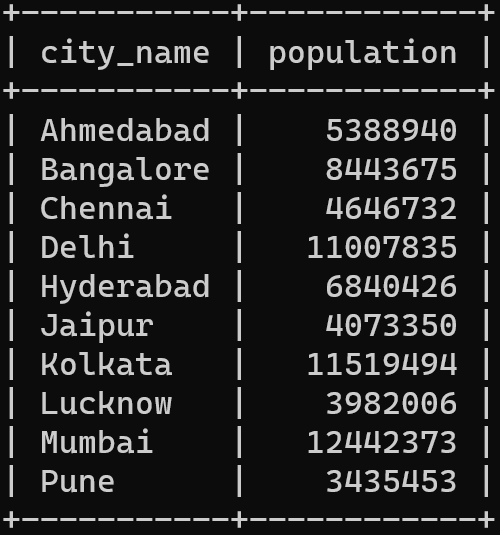
('Pune', 3124453),

('Ahmedabad', 5577940),

('Jaipur', 3073350),

('Lucknow', 3382000);

**Snapshot of the table:**



1. **Creation of Shipment table:**

**Query:**

CREATE TABLE IF NOT EXISTS Shipment (

shipment\_no CHAR(10),

cust\_id INT UNSIGNED,

weight DECIMAL(5, 2),

truck\_no CHAR(10),

destination VARCHAR(50),

ship\_date DATE,

CHECK (weight >= 0 AND weight < 1000),

CHECK (shipment\_no LIKE 'S%'),

PRIMARY KEY (shipment\_no, cust\_id),

FOREIGN KEY (cust\_id) REFERENCES Customer(cust\_id) ON DELETE CASCADE,

FOREIGN KEY (truck\_no) REFERENCES Truck(truck\_no) ON DELETE SET NULL,

FOREIGN KEY (destination) REFERENCES City(city\_name)

);

**Inserting values:**

INSERT INTO Shipment (shipment\_no, cust\_id, weight, truck\_no, destination, ship\_date)

VALUES

('SNO2709268', 4004, 230.39, 'TNO0010000', 'Jaipur', '2023-06-02'),

('SNO2709268', 3003, 665.34, 'TNO0001010', 'Hyderabad', '2023-05-15'),

('SNO2709268', 2002, 319.97, 'TNO0010110', 'Jaipur', '2023-05-09'),

('SNO4761770', 3001, 969.62, 'TNO0001010', 'Pune', '2023-06-20'),

('SNO4761770', 3004, 947.81, 'TNO0011000', 'Ahmedabad', '2023-05-26'),

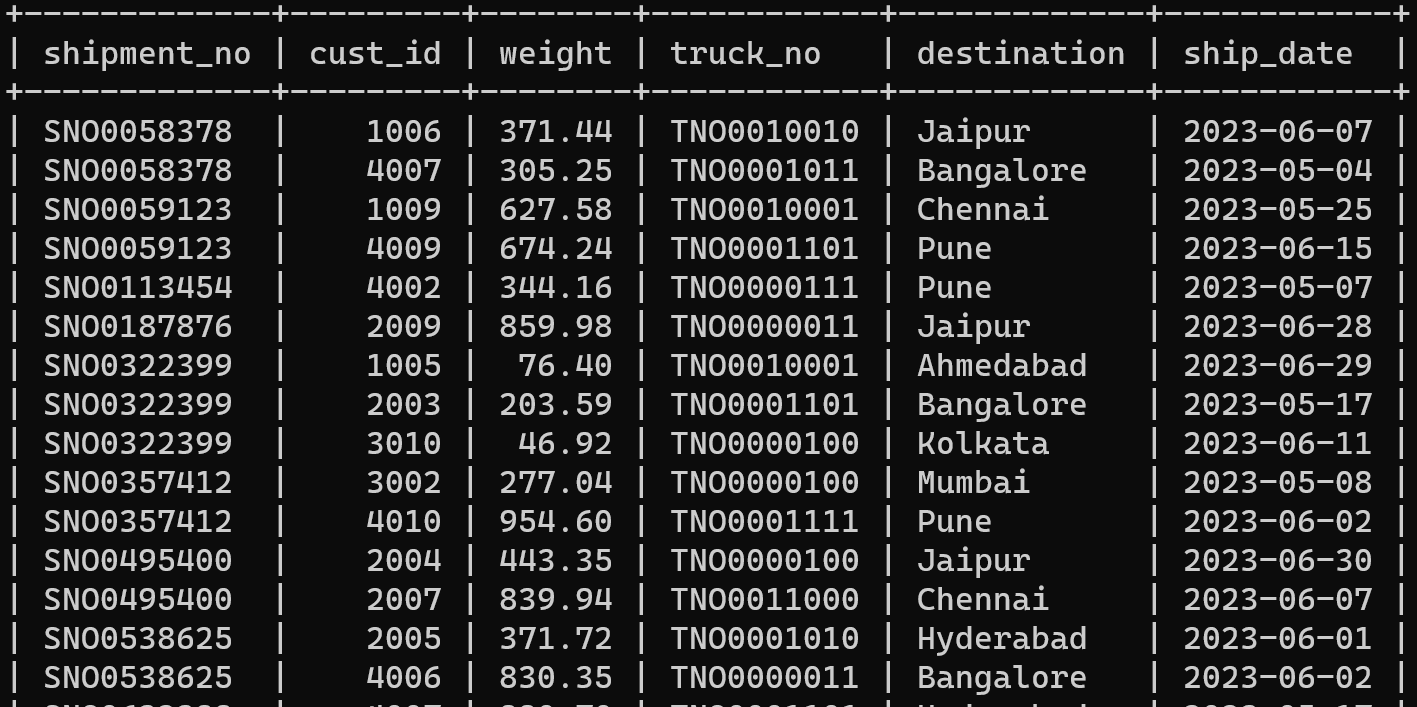
('SNO0187876', 2009, 859.98, 'TNO0000011', 'Jaipur', '2023-06-28'),

('SNO4156545', 2001, 703.03, 'TNO0010111', 'Chennai', '2023-06-27'),

('SNO5060006', 2006, 777.32, 'TNO0000101', 'Kolkata', '2023-05-10'),

('SNO5060006', 1003, 250.69, 'TNO0011000', 'Jaipur', '2023-06-28');

**Snapshot of the table:**



1. **Queries and their solutions:**
2. **Give names of customer who have sent packages (shipments) to Kolkata, Chennai and Mumbai.**

**Query:**

select distinct cust\_name  
from shipment s, customer c  
where s.cust\_id = c.cust\_id  
and destination in (‘Kolkata’, ‘Chennai’, ‘Mumbai’);

**Output:**

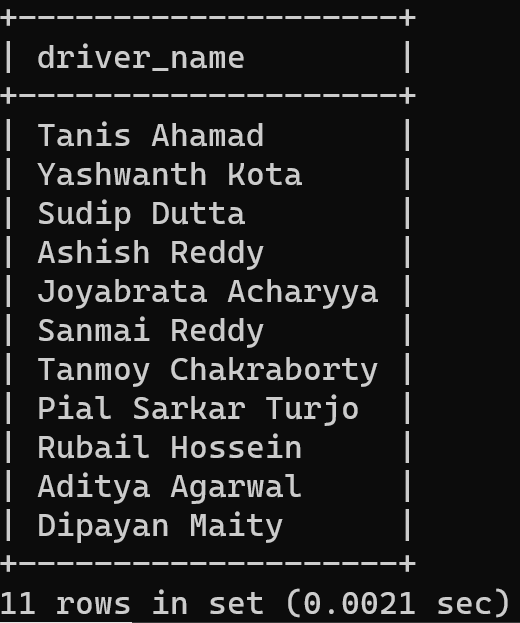


1. **List the names of the driver who have delivered shipments weighing over 900 pounds.**

**Query:**

select distinct driver\_name  
from shipment s, truck t  
where s.truck\_no = t.truck\_no  
and s.weight > 900;

**Output:**

****

1. **Retrieve the maximum and minimum weights of the shipments. Rename the output as Max\_Weight and Min\_Weight respectively.**

**Query:**

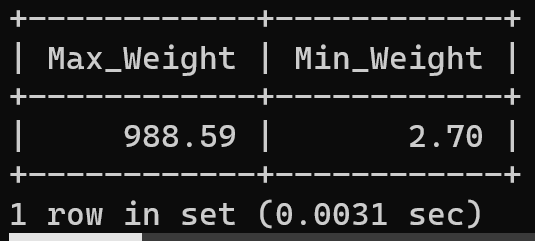
select

max(weight) as Max\_Weight,

min(weight) as Min\_Weight

from shipment;

**Output:**



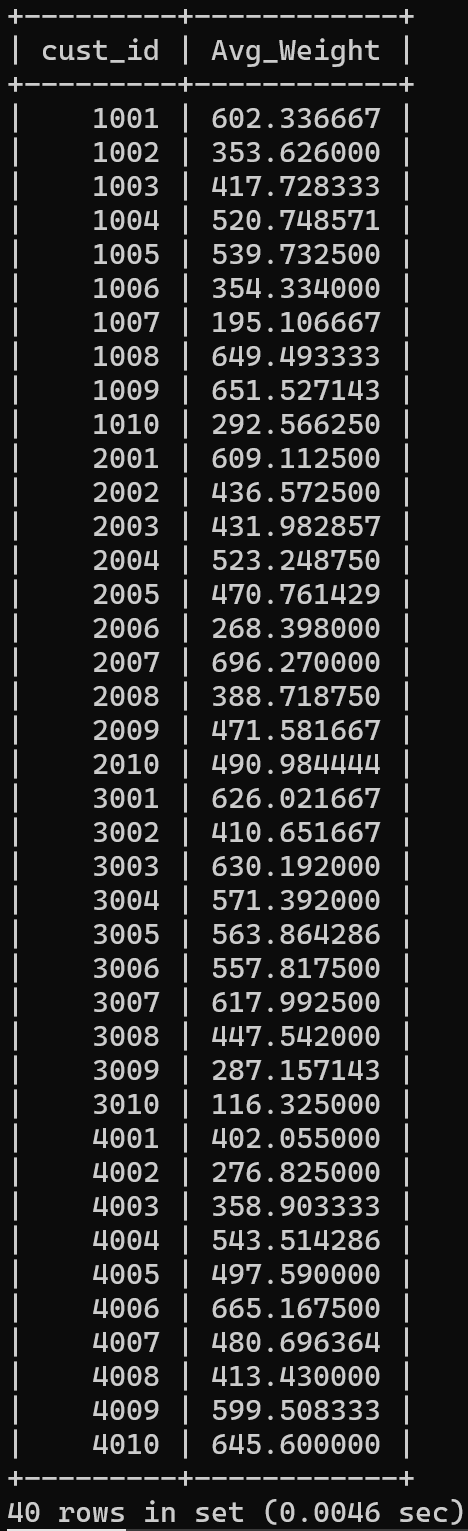
1. **For each customer, what is the average weight of package sent by the customer?**

**Query:**

select cust\_id, avg(weight) as Avg\_Weight  
from shipment  
group by cust\_id

Order by cust\_id;

**Output:**



1. **List the names and populations of cities that have received a shipment weighing over 900 pounds.**

**Query:**

select distinct s.destination as city, c.population as population

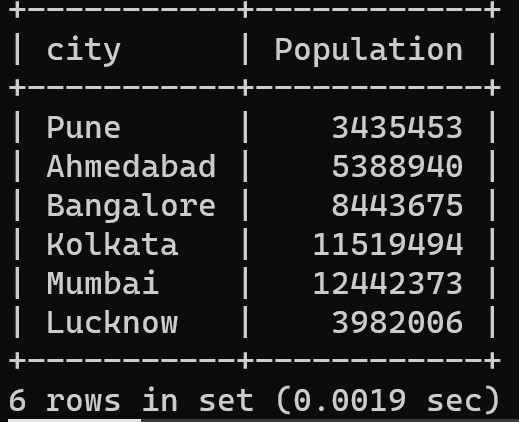
from shipment s, (select city\_name, population from city) c

where s.destination = c.city\_name

and s.weight > 900

group by s.destination;

**Output:**



1. **List cities that have received shipments from every customer.**

**Query:**

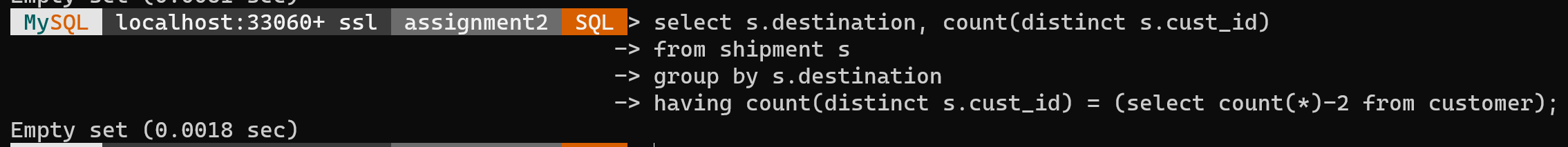
select s.destination, count(distinct s.cust\_id)

from shipment s

group by s.destination

having count(distinct s.cust\_id) = (select count(\*)-2 from customer);

**Output:**



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

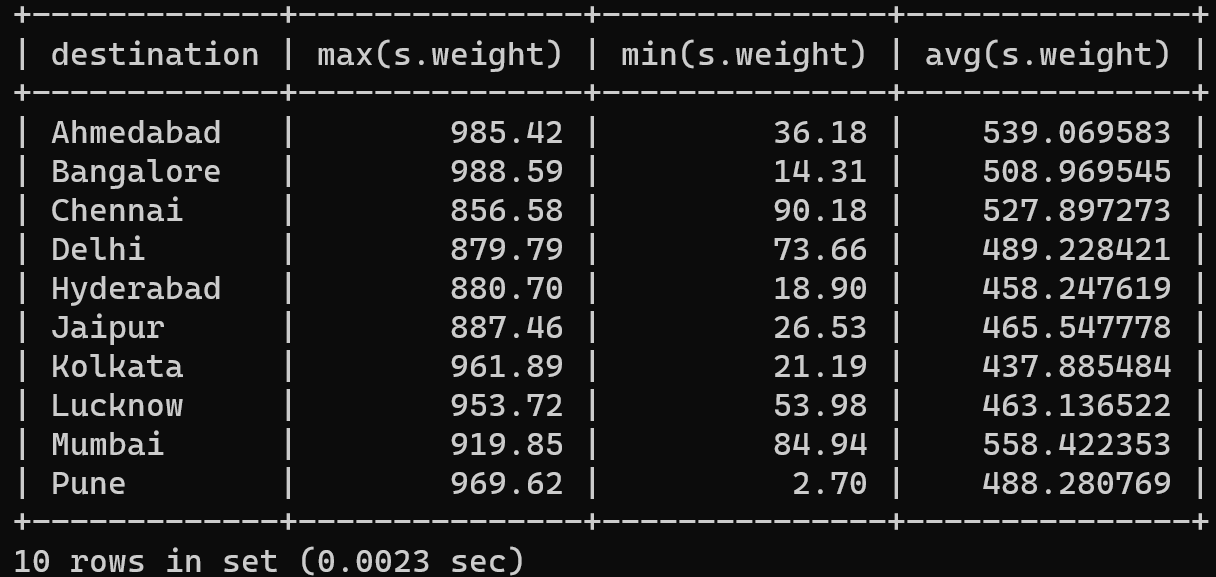
**Query:**

select s.destination, max(s.weight), min(s.weight), avg(s.weight)

from shipment s

group by s.destination;

**Output:**



1. **List the name and annual revenue of customers whose shipments have been delivered by truck driver ‘Yashwanth Kota’.**

**Query:**

select distinct c.cust\_id, c.cust\_name, c.annual\_revenue

from customer c

where c.cust\_id in (

select distinct s.cust\_id

from shipment s

where s.truck\_no in (

select t.truck\_no

from truck t

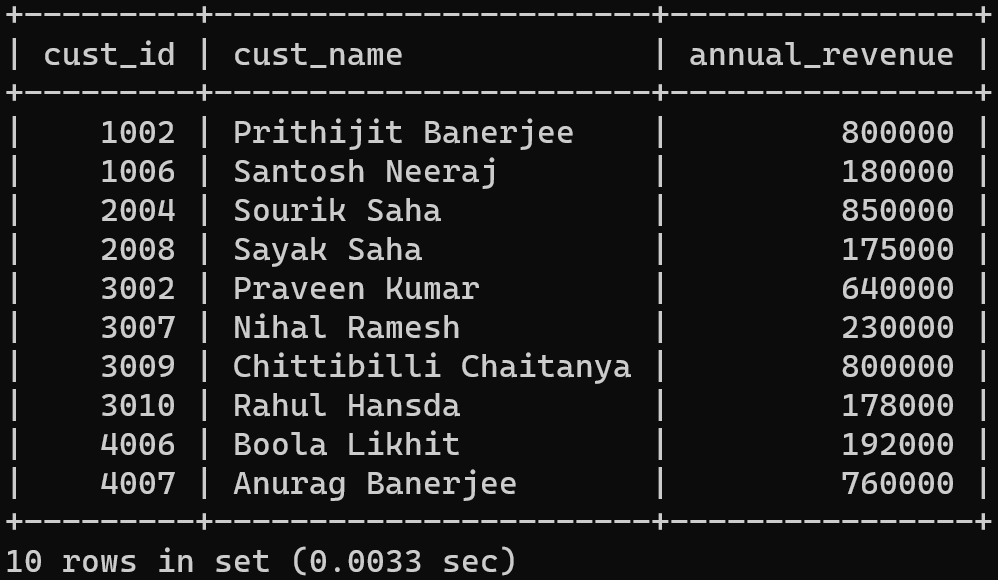
where t.driver\_name = 'yashwanth kota'

)

)

order by c.cust\_id;

**Output:**



1. **List drivers who have delivered shipments to every city.**

**Query:**

select t.truck\_no, t.driver\_name

from truck t

where exists (

select 1

from shipment s

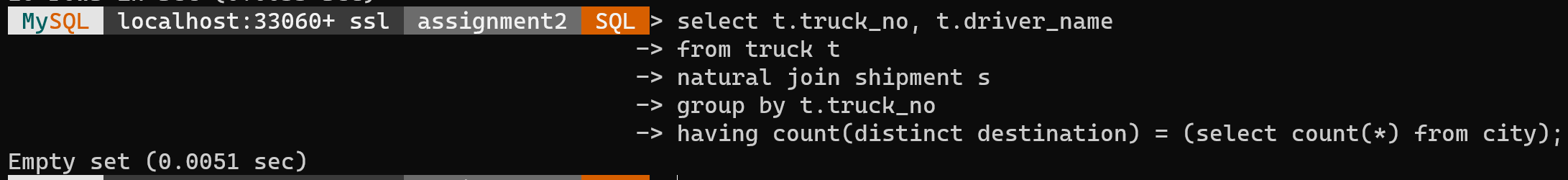
where s.truck\_no = t.truck\_no

group by s.truck\_no

having count(distinct s.destination) = (select count(\*) from city)

);

**Output:**



1. **For each city, with population 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

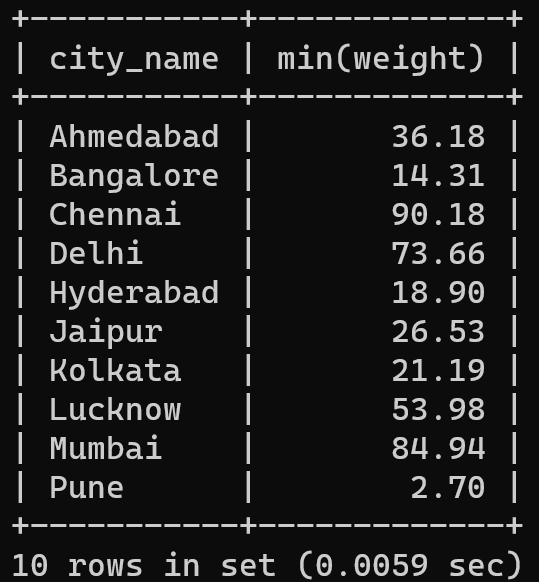
destination = city\_name

and

population > 1000000;

group by city\_name;

**Output:**



**Assignment No : 09**

1. **Creation of Tables:**
2. **Creation of EMP Table:**

**Query:**

CREATE TABLE IF NOT EXISTS dept (

deptno CHAR(5) PRIMARY KEY CHECK (deptno LIKE 'D%'),

dname ENUM('Accounting', 'Sales', 'Research', 'Operations'),

loc VARCHAR(20)

);

**Inserting values:**

INSERT INTO dept (deptno, dname, loc) VALUES

('D0001', 'Accounting', 'Lucknow'),

('D0012', 'Sales', 'Pune'),

('D0013', 'Research', 'Bangalore'),

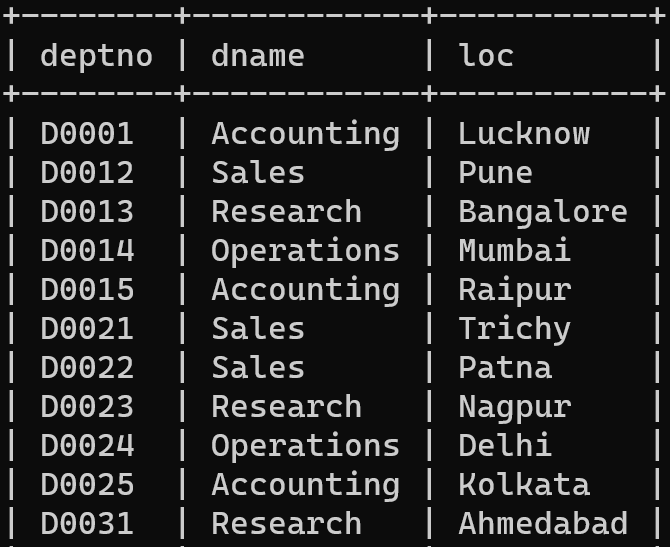
('D0014', 'Operations', 'Mumbai'),

('D0015', 'Accounting', 'Raipur'),

('D0021', 'Sales', 'Trichy'),

('D0023', 'Research', 'Nagpur');

**Snapshot of the table:**



1. **Creation of DEPT Table:**

**Query:**

CREATE TABLE IF NOT EXISTS emp (

empno INT PRIMARY KEY CHECK (empno BETWEEN 7000 AND 8000),

ename VARCHAR(10),

job ENUM('Clerk', 'Salesman', 'Manager', 'Analyst', 'President'),

mgr INT REFERENCES emp(empno),

hiredate DATE,

sal DECIMAL(10, 0),

comm DECIMAL(4, 0) DEFAULT 0 CHECK (comm < 1500),

deptno CHAR(5) REFERENCES dept(deptno)

);

**Inserting values:**

INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) VALUES

(7444, 'Anish', 'Manager', 7439, '2019-07-06', 4327, 1333, 'D0034'),

(7890, 'Subhra', 'Manager', 7157, '2021-11-01', 9026, 1378, 'D0025'),

(7784, 'Achuth', 'Manager', 7585, '2022-02-19', 3454, 469, 'D0022'),

(7182, 'Devika', 'Manager', 7137, '2020-04-23', 9091, 49, 'D0031'),

(7681, 'Rishabh', 'Manager', 7046, '2023-05-04', 4388, 31, 'D0031'),

(7013, 'Sudip', 'Manager', 7881, '2020-11-10', 4300, 461, 'D0024'),

(7617, 'Ashish', 'Manager', 7182, '2021-04-13', 6923, 387, 'D0015');

**Snapshot of the table:**

****

1. **Queries and their solutions:**
2. **Display the difference between highest and lowest salary of each department in descending order. Label the column as “Difference”.**

**Query:**

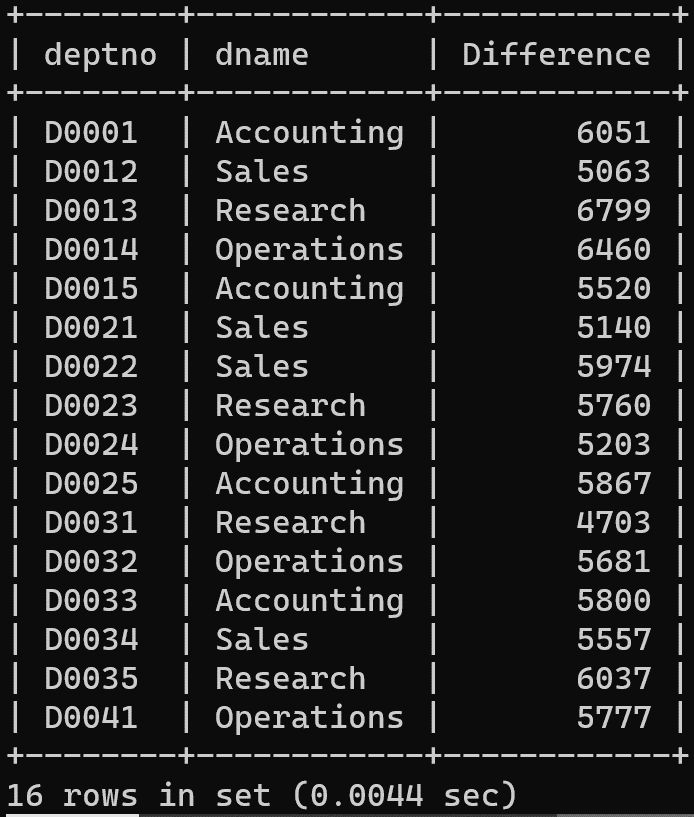
select d.deptno, d.dname,

(select max(sal) from emp where emp.deptno = d.deptno) -

(select min(sal) from emp where emp.deptno = d.deptno) as difference

from dept d;

**Output:**



1. **List all the employees’ employee number and name along with their immediate managers’ employee number and name.**

**Query:**

select e1.empno, e1.ename, e1.mgr,

(select e2.empno from emp e2 where e2.empno = e1.mgr) as mgr\_no,

(select e2.ename from emp e2 where e2.empno = e1.mgr) as mgr\_name

from emp e1

where e1.mgr is not null

having mgr\_no is not null and mgr\_name is not null;

**Output:**

****





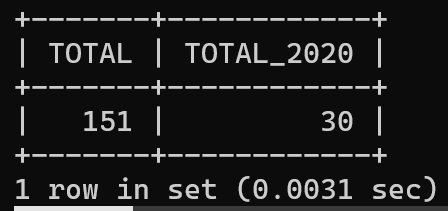
1. **Create a query that will display the total number of employees and the total number of employees who were hired only in 2020. Give the column headings as “TOTAL” and “TOTAL\_2020” respectively.**

**Query:**

select count(\*) as TOTAL, count(case when year(hiredate) = 2020 then 1 else null end) as TOTAL\_2020

from emp;

**Output:**



1. **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 salaries less than 1000. Sort the output in descending order of salary.**

**Query:**

select e1.empno as "manager number",

(select min(e2.sal) from emp e2 where e2.mgr = e1.empno) as min\_sal

from emp e1

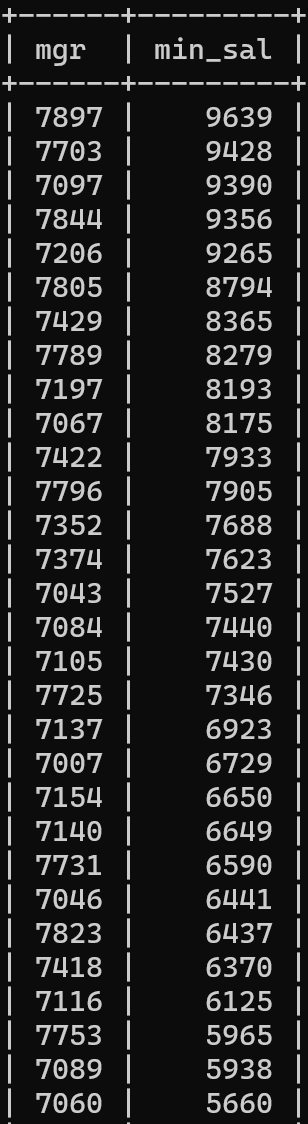
where e1.empno in (select distinct mgr from emp where mgr is not null)

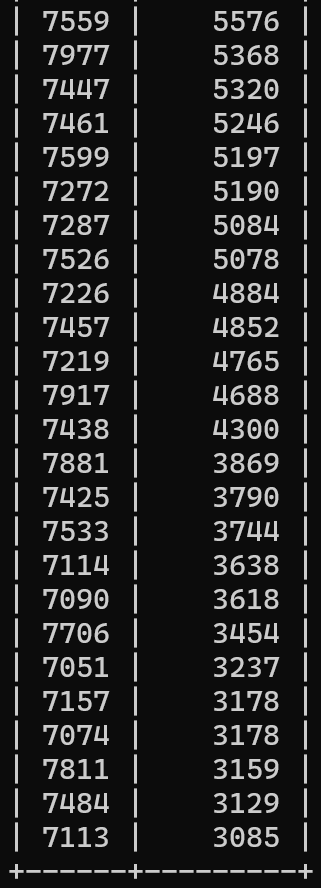
group by e1.empno

having (select min(e2.sal) from emp e2 where e2.mgr = e1.empno) >= 1000

order by min\_sal desc;

**Output:**





1. **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 averge 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 d.dname as "DNAME",

d.loc as "LOCATION",

(select count(e.empno) from emp e where e.deptno = d.deptno) as "NUMBER OF PEOPLE",

(select round(avg(e.sal), 2) from emp e where e.deptno = d.deptno) as "AVERAGE SALARY"

from dept d

union all

select d.dname as "DNAME",

d.loc as "LOCATION",

0 as "NUMBER OF PEOPLE",

0.00 as "AVERAGE SALARY"

from dept d

where d.deptno not in (select distinct deptno from emp)

order by "DNAME";

**Output:**

