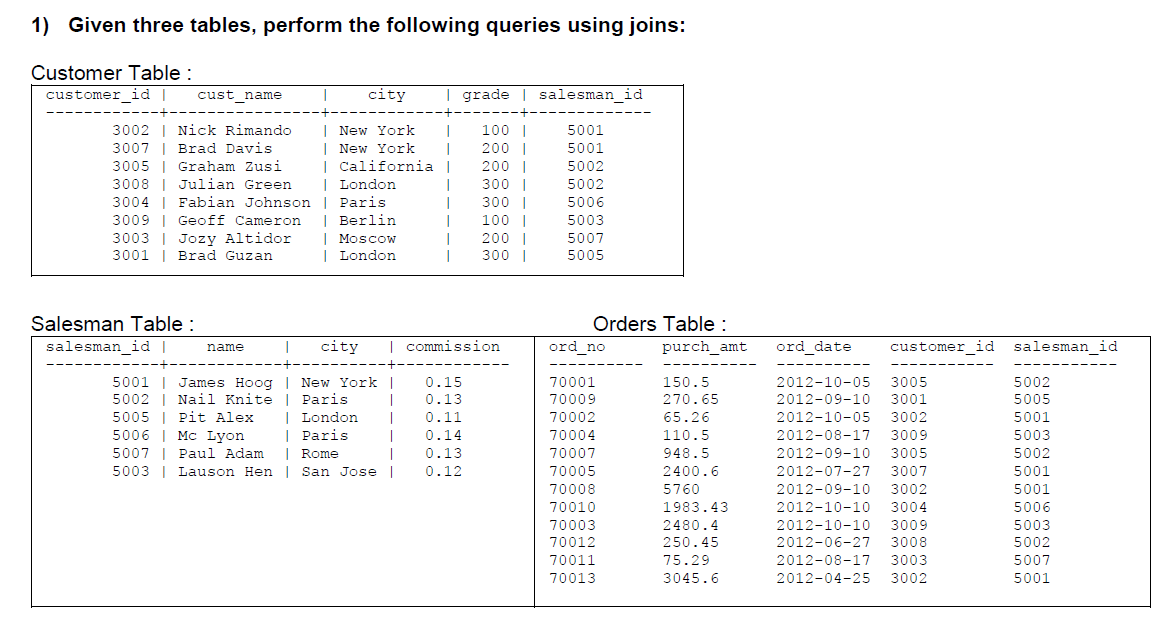


|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | CSLR 51 : DBMS LAB-3 | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Roll no. : 106119100Name : Rajneesh PandeySection : CSE-B |  | | | |
|  | | |  | | |  | | |

PROBLEM 1



**ER Diagram :**

Diagram

Description automatically generated

CREATE DATABASE salesDB;

CREATE TABLE salesman (

  `salesman\_id` INT NOT NULL,

  `name` VARCHAR(100) NOT NULL,

  `city` VARCHAR(45) NOT NULL,

  `commission` DECIMAL(5,3) NOT NULL,

  PRIMARY KEY (`salesman\_id`)

  );

CREATE TABLE customer (

  `customer\_id` INT NOT NULL,

  `cust\_name` VARCHAR(100) NOT NULL,

  `city` VARCHAR(45) NOT NULL,

  `grade` INT NOT NULL,

  `salesman\_id` INT NOT NULL,

  PRIMARY KEY (`customer\_id`),

  INDEX `salesman\_id\_idx` (`salesman\_id` ASC) VISIBLE,

  CONSTRAINT `salesman\_id`

    FOREIGN KEY (`salesman\_id`)

    REFERENCES `salesDB`.`salesman` (`salesman\_id`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION

    );

CREATE TABLE orders(

  `ord\_no` INT NOT NULL,

  `purch\_amt` DECIMAL(8,3) NOT NULL,

  `ord\_date` DATE NOT NULL,

  `customer\_id` INT NOT NULL,

  `salesman\_id` INT NOT NULL,

    PRIMARY KEY (`ord\_no`),

    FOREIGN KEY (`customer\_id`)

    REFERENCES `salesDB`.`customer` (`customer\_id`),

    FOREIGN KEY (`salesman\_id`)

    REFERENCES `salesDB`.`salesman` (`salesman\_id`)

    );

INSERT INTO salesman (`salesman\_id`, `name`, `city`, `commission`)

VALUES

        ('5001', 'James Hoog', 'New York', '0.15'),

        ('5002', 'Nail Knite', 'Paris', '0.13'),

        ('5005', 'Pit Alex', 'London', '0.11'),

        ('5006', 'Mc Lyon', 'Paris', '0.14'),

        ('5007', 'Paul Adam', 'Rome', '0.13'),

        ('5003', 'Lauson Hen', 'San Jose', '0.12');

INSERT INTO customer (`customer\_id`, `cust\_name`, `city`, `grade`, `salesman\_id`)

VALUES

        ('3002', 'Nick Rimando', 'New York', '100', '5001'),

        ('3007', 'Brad Davis', 'New York', '200', '5001'),

        ('3005', 'Graham Zusi', 'California', '200', '5002'),

        ('3008', 'Julian Green', 'London', '300', '5002'),

        ('3004', 'Fabian Johnson', 'Paris', '300', '5006'),

        ('3009', 'Geoff Cameroon', 'Berlin', '100', '5003'),

        ('3003', 'Jozy Altidor', 'Moscow', '200', '5007'),

        ('3001', 'Brad Guzan', 'London', '200', '5005');

INSERT INTO orders (`ord\_no`, `purch\_amt`, `ord\_date`, `customer\_id`, `salesman\_id`)

VALUES

        ('70001', '150.5', '2012-10-05', '3005', '5002'),

        ('70009', '270.65', '2012-09-10', '3001', '5005'),

        ('70002', '65.26', '2012-10-05', '3002', '5001'),

        ('70004', '110.5', '2012-08-17', '3009', '5003'),

        ('70007', '948.5', '2012-09-10', '3005', '5002'),

        ('70005', '2400.6', '2012-07-27', '3007', '5001'),

        ('70008', '5760', '2012-09-10', '3002', '5001'),

        ('70010', '1983.43', '2012-10-10', '3004', '5006'),

        ('70003', '2480.4', '2012-10-10', '3009', '5003'),

        ('70012', '250.45', '2012-06-27', '3008', '5002'),

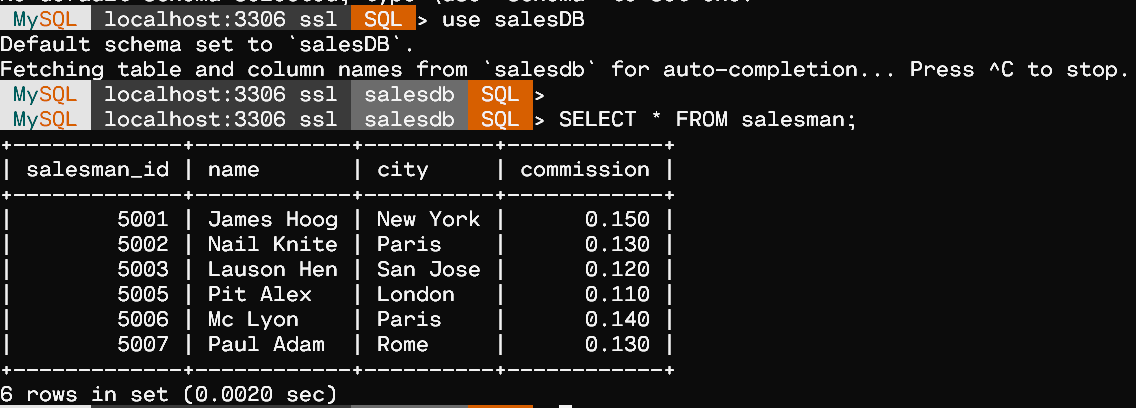
        ('70011', '75.29', '2012-08-17', '3003', '5007'),

        ('70013', '3045.6', '2012-04-25', '3002', '5001');

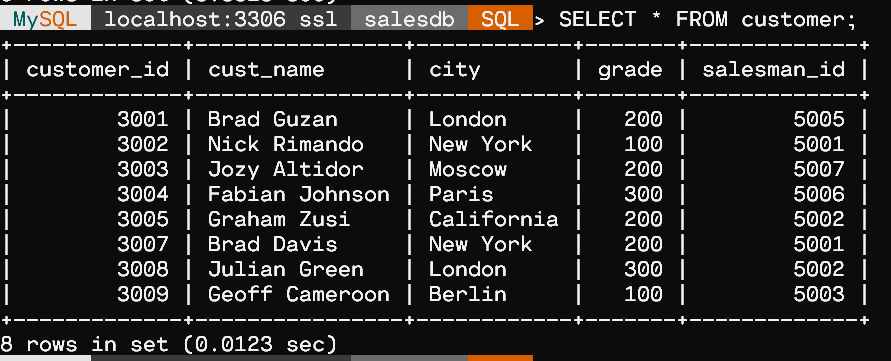
**Query :-**

Show Tables :

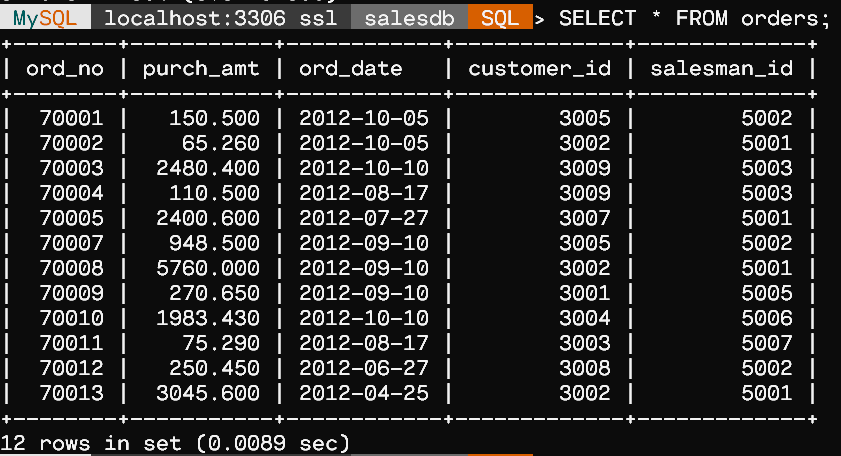
SELECT \* FROM salesman;



SELECT \* FROM customer;



SELECT \* FROM orders;



a) Write a SQL query to find those salespersons who received a commission from the company more than 12%.Return Customer Name,customercity, Salesman, commission (Use Inner join)

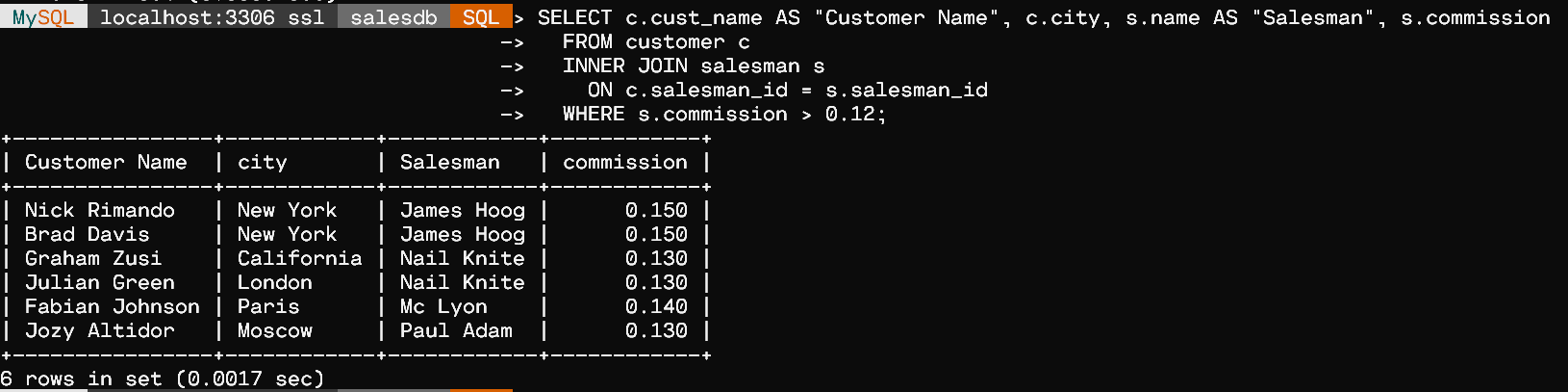
SELECT c.cust\_name AS "Customer Name", c.city, s.name AS "Salesman", s.commission

  FROM customer c

  INNER JOIN salesman s

    ON c.salesman\_id = s.salesman\_id

  WHERE s.commission > 0.12;



b) Write a SQL statement to make a report with customer name, city,order number, order date,and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders. (Use left outer join)

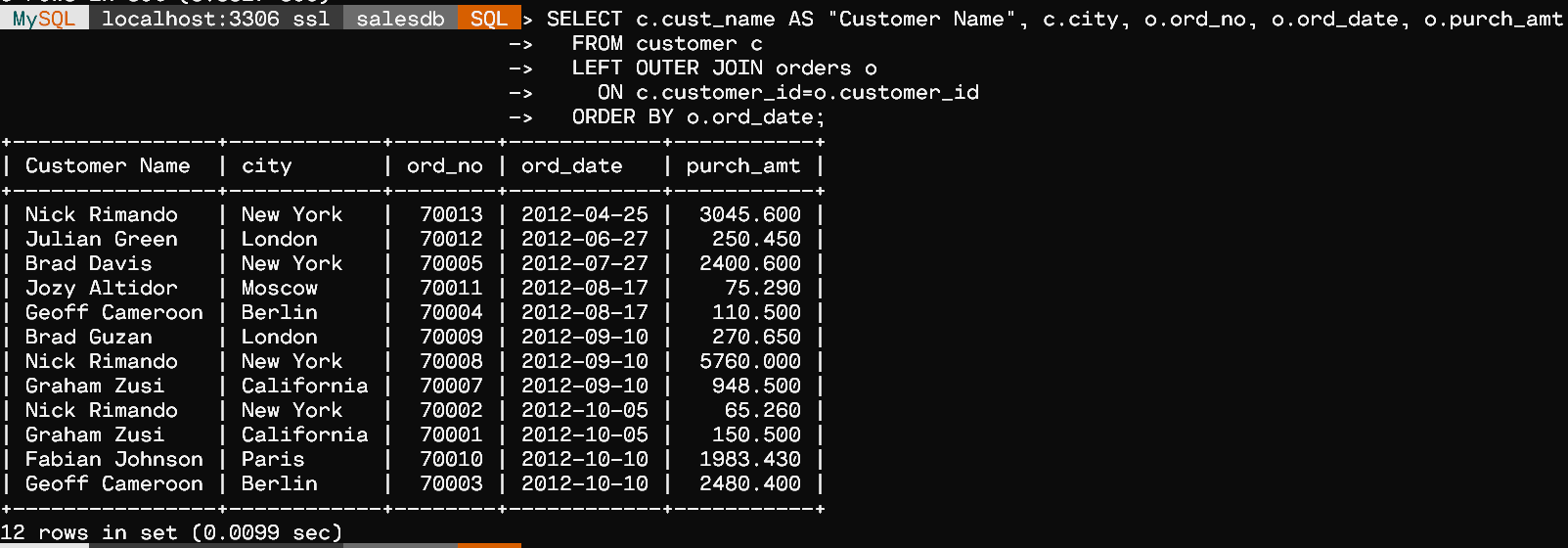
SELECT c.cust\_name AS "Customer Name", c.city, o.ord\_no, o.ord\_date, o.purch\_amt

  FROM customer c

  LEFT OUTER JOIN orders o

    ON c.customer\_id=o.customer\_id

  ORDER BY o.ord\_date;



c) Write a SQL statement to make a report with customer name, city,order number, order date, order amount salesman name and commission tofind that either any of the existing customers have placed no orde or placed one or more orders by their salesman or by own.

SELECT c.cust\_name AS "Customer Name",c.city,o.ord\_no,o.ord\_date,o.purch\_amt,s.name AS "Salesman",

       s.commission

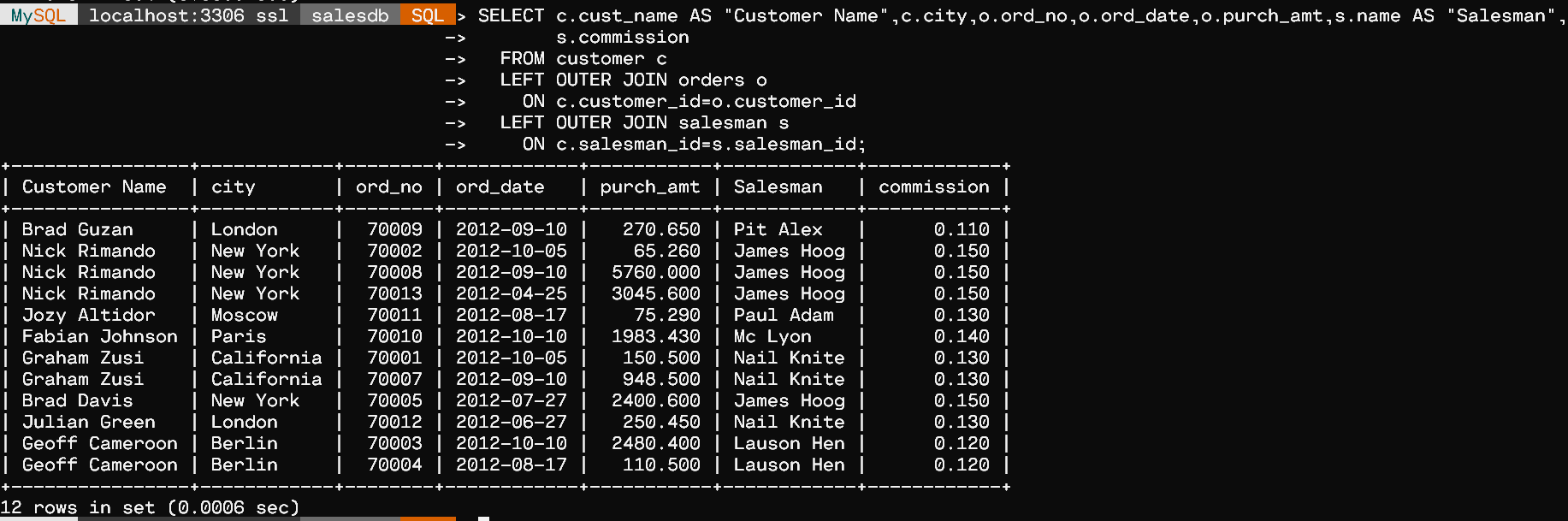
  FROM customer c

  LEFT OUTER JOIN orders o

    ON c.customer\_id=o.customer\_id

  LEFT OUTER JOIN salesman s

    ON c.salesman\_id=s.salesman\_id;



d) Write a SQL statement to make a list in ascending order for the salesmenwho works either for one or more customer or not yet join under and of the customers. (Use Right outer join)

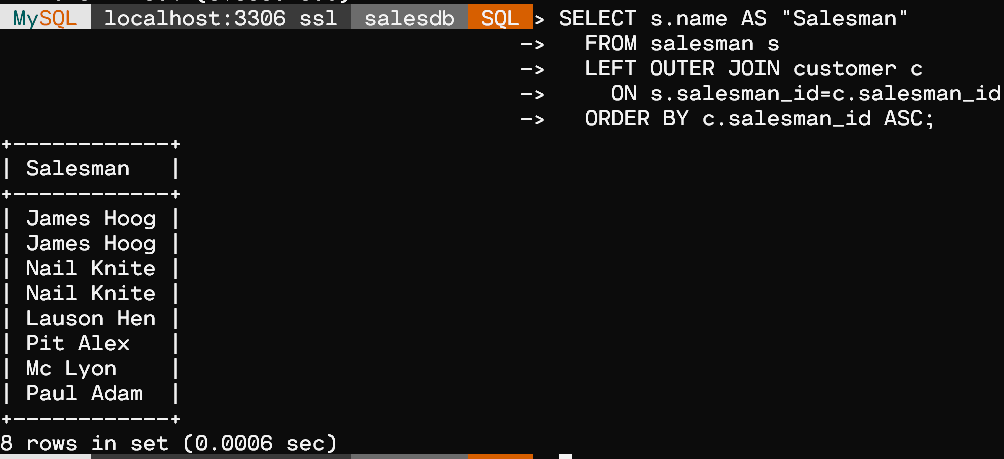
SELECT s.name AS "Salesman"

  FROM salesman s

  RIGHT OUTER JOIN customer c

    ON s.salesman\_id=c.salesman\_id

  ORDER BY c.salesman\_id ASC;

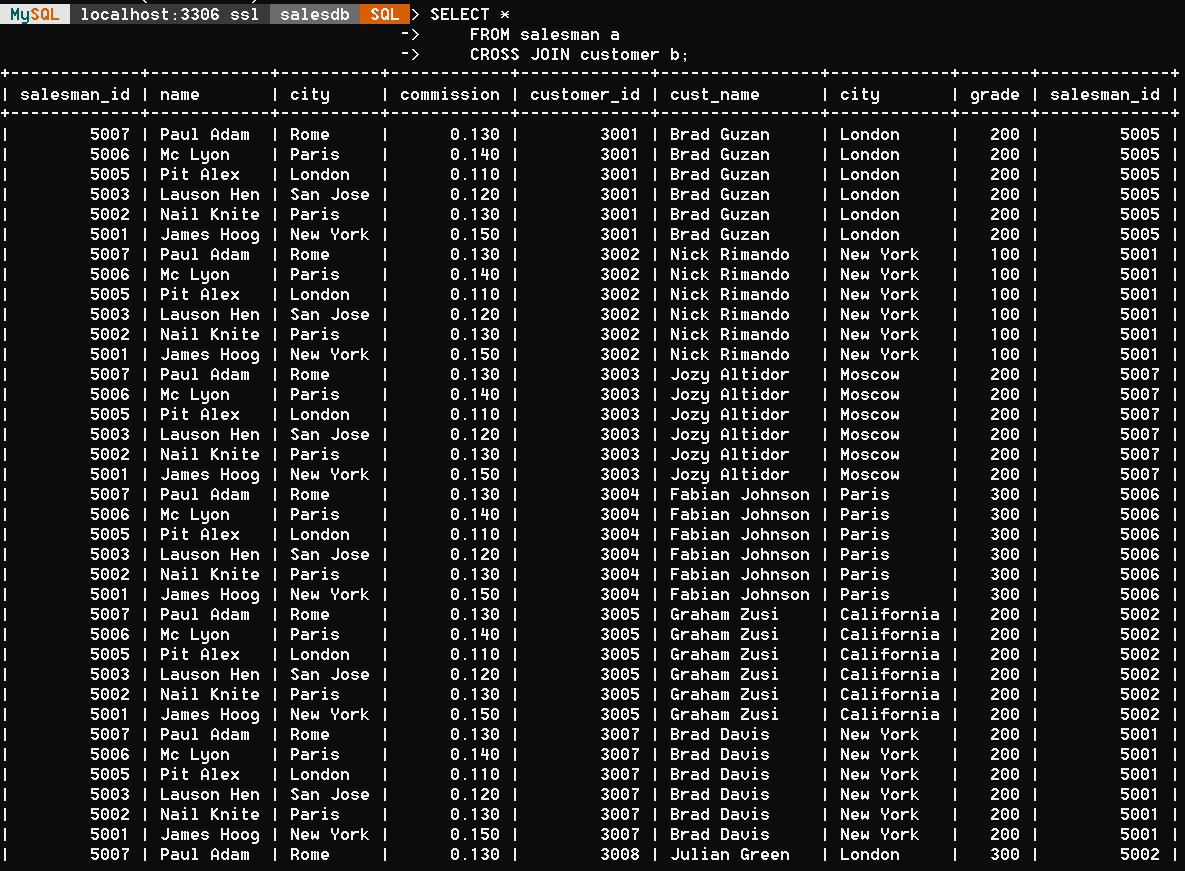


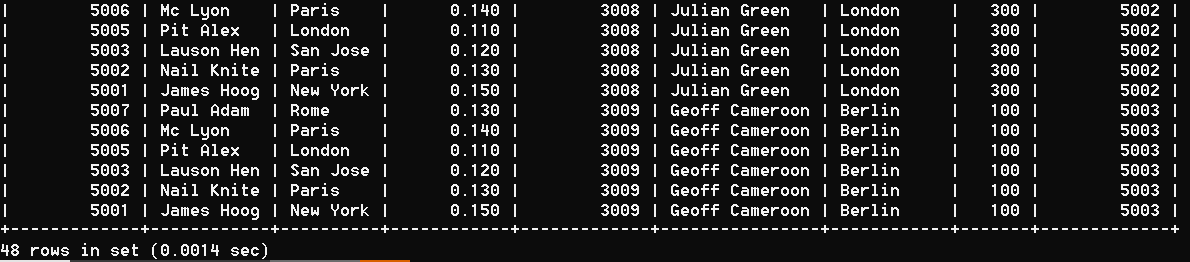
e) Write a SQL query to combine each row of salesman table with each row of customer table. (Use cross join)

SELECT \*

    FROM salesman a

    CROSS JOIN customer b;



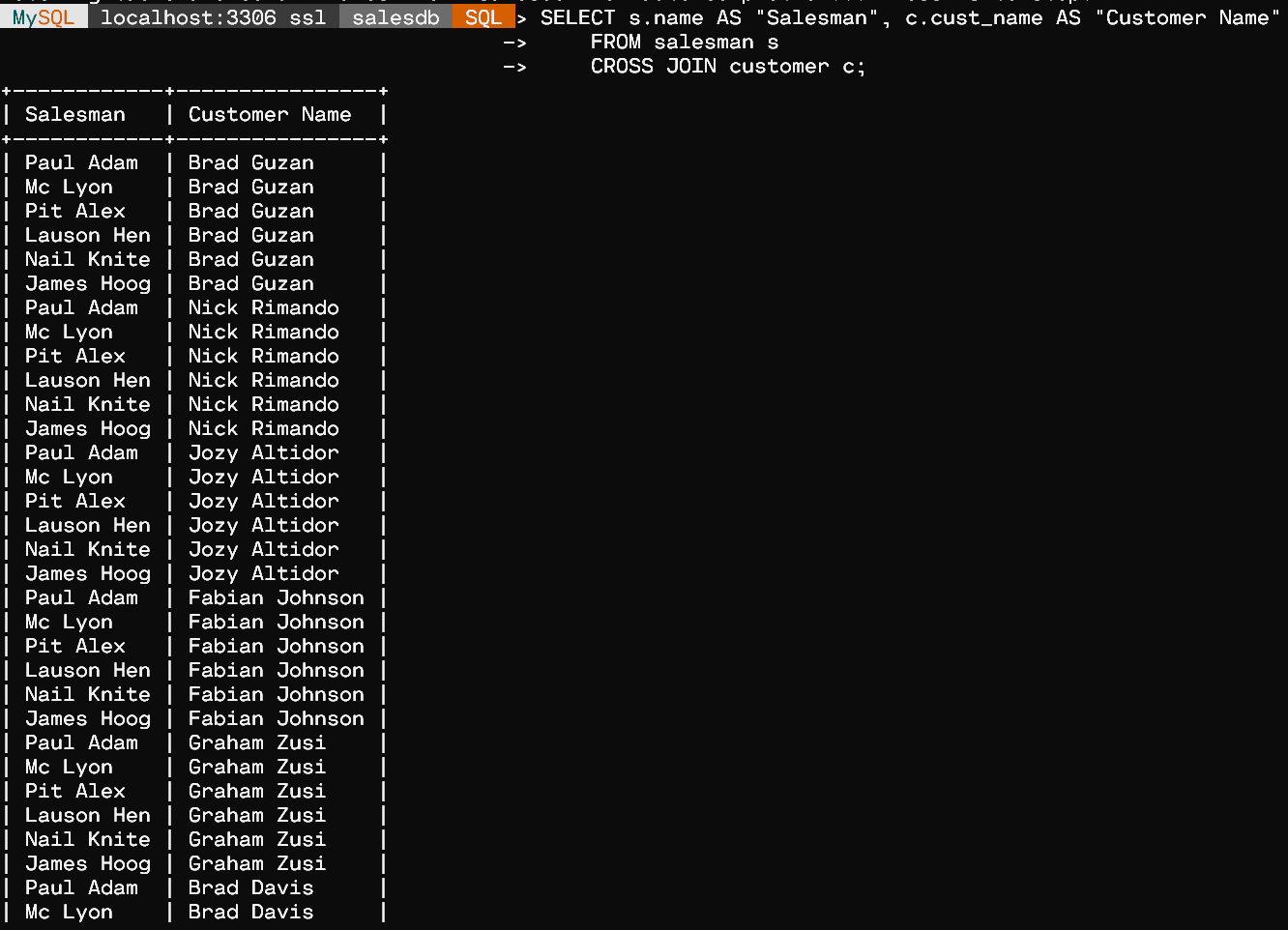
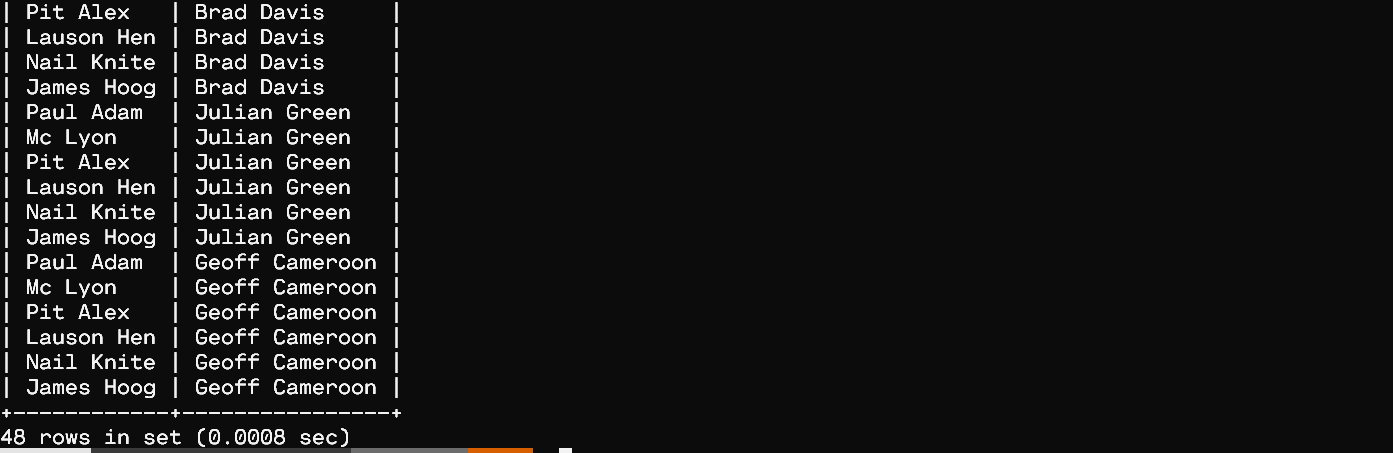


f) Write a SQL statement to make a cartesian product between salesman and customer i.e.each salesman will appear for all customer and vice versa for that salesman who belongs to a city.(Use cross join)

SELECT s.name AS "Salesman", c.cust\_name AS "Customer Name"

    FROM salesman s

    CROSS JOIN customer c;

g) Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer.The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier. (Use left outer join and right outer join)

SELECT s.name AS "Salesman"

  FROM salesman s

  LEFT OUTER JOIN customer c

    ON s.salesman\_id=c.salesman\_id

  RIGHT OUTER JOIN orders o

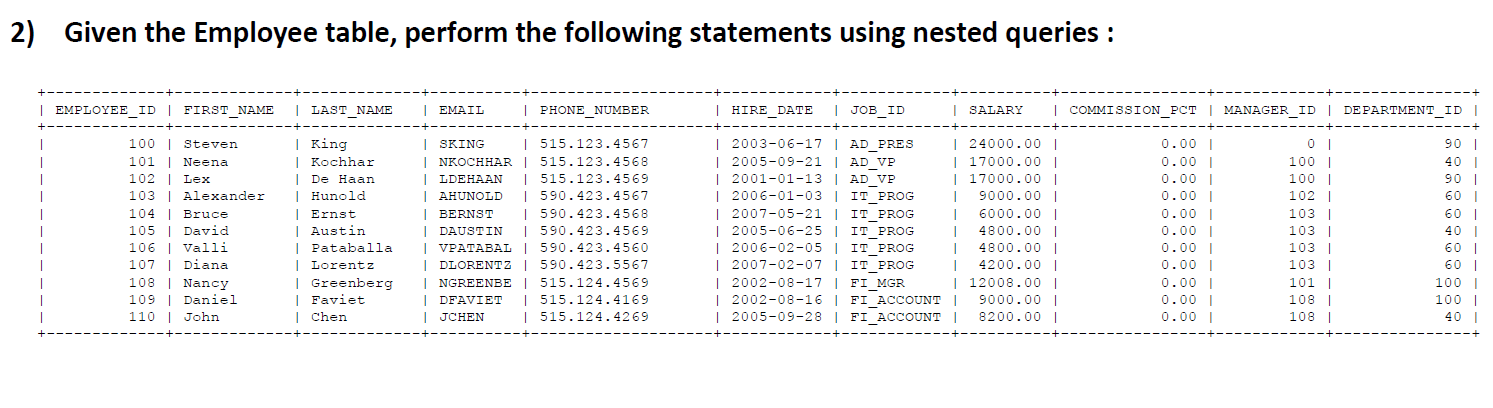
    ON c.customer\_id=o.customer\_id

  WHERE o.purch\_amt >= 2000

    AND grade IS NOT NULL;



PROBLEM 2



**ER Diagram :**

**Graphical user interface, text, application

Description automatically generated**

CREATE DATABASE employeeDB;

CREATE TABLE `employee`(

  `employee\_id` INT NOT NULL,

  `first\_name` VARCHAR(100) NOT NULL,

  `last\_name` VARCHAR(100) NOT NULL,

  `email` VARCHAR(100) NOT NULL,

  `phone\_number` VARCHAR(45) NOT NULL,

  `hire\_date` DATE NOT NULL,

  `job\_id` VARCHAR(45) NOT NULL,

  `salary` DECIMAL(12,3) NOT NULL,

  `commission\_pct` DECIMAL(5,3) NOT NULL,

  `manager\_id` INT NOT NULL,

  `department\_id` INT NOT NULL,

  PRIMARY KEY (`employee\_id`)

  );

INSERT INTO employee

(`employee\_id`, `first\_name`, `last\_name`, `email`, `phone\_number`, `hire\_date`, `job\_id`, `salary`, `commission\_pct`, `manager\_id`, `department\_id`)

VALUES

        ('100', 'Steven', 'King', 'SKING', '515.123.4567', '2003-06-17', 'AD\_PRES', 24000.00, '0.00', '0', '90'),

        ('101', 'Neena', 'Kochhar', 'NKOCHHAR ', '515.123.4568 ', ' 2005-09-21 ', 'AD\_VP', 17000.00, '0.00', '100', '40'),

        ('102', 'Lex', 'De Haan', 'LDEHAAN', '515.123.4569', '2001-01-13 ', 'AD\_VP', 17000.00, '0.00', '100', '90'),

        ('103', 'Alexander', 'Hunold', 'AHUNOLD', '590.423.4567', '2006-01-03', 'IT\_PROG', 9000.00, '0.00', '102', '60'),

        ('104', 'Bruce', 'Ernst', 'BERNST', '590.423.4568 ', '2007-05-21', 'IT\_PROG', 6000.00, '0.00', '103', '60'),

        ('105', 'David', 'Austin', 'DAUSTIN', '590.423.4569', '2005-06-25', 'IT\_PROG', 4800.00, '0.00', '103', '40'),

        ('106', 'Valli', 'Pataballa', 'VPATABAL', '590.423.4560', '2006-02-05', 'IT\_PROG', 4800.00, '0.00', '103', '60'),

        ('107', 'Diana', 'Lorentz', 'DLORENTZ', '590.423.5567', '2007-02-07', 'IT\_PROG', 4200.00, '0.00', '103', '60'),

        ('108', 'Nancy', 'Greenberg', 'NGREENBE', '515.124.4569', '2002-08-17', 'FI\_MGR', 12008.00, '0.00', '101', '100'),

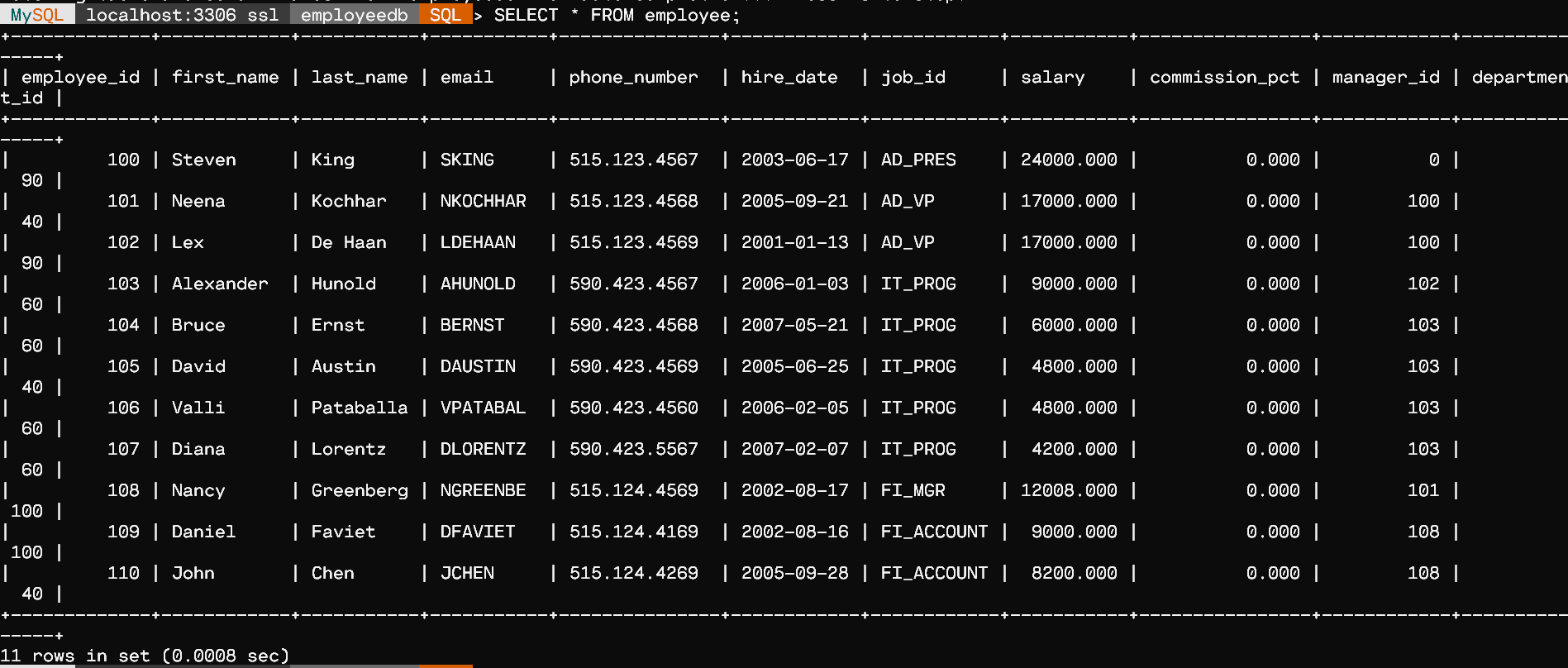
        ('109', 'Daniel', 'Faviet', 'DFAVIET', '515.124.4169', '2002-08-16', 'FI\_ACCOUNT', 9000.00, '0.00', '108', '100'),

        ('110', 'John', 'Chen', 'JCHEN', '515.124.4269', '2005-09-28', 'FI\_ACCOUNT', 8200.00, '0.00', '108', '40');

-- Query :-

-- Show Tables :

    SELECT \* FROM employee;



a) Write a query to display the employee name (first name and last name) and department for all employees for any existence of those employees whose salary is more than 3700.

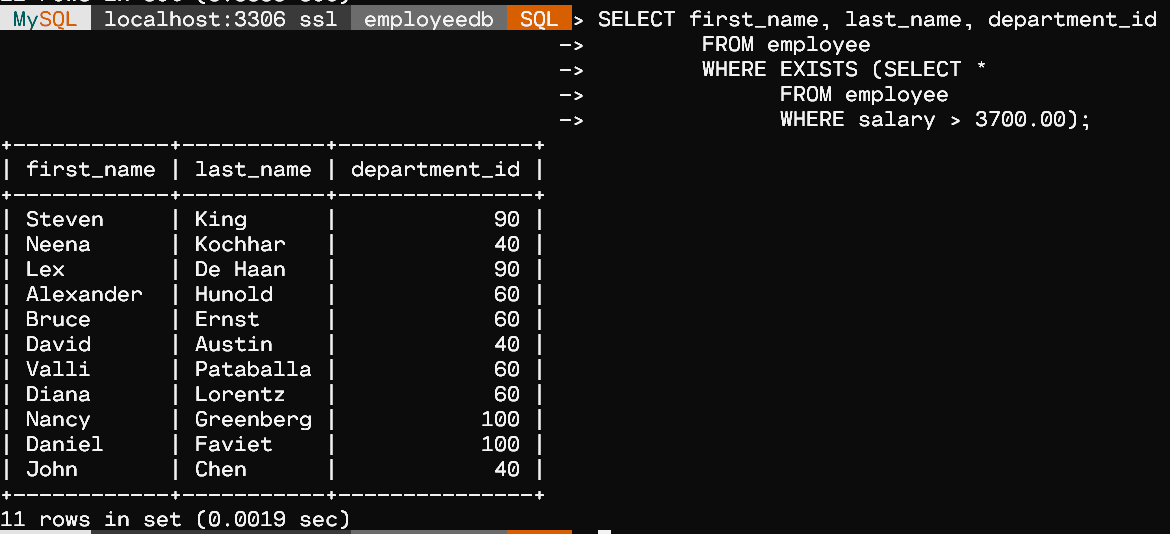
    SELECT first\_name, last\_name, department\_id

        FROM employee

        WHERE EXISTS (SELECT \*

              FROM employee

              WHERE salary > 3700.00);



b) Write a query to display the department id and the total salary for those departments which contains at least one employee.

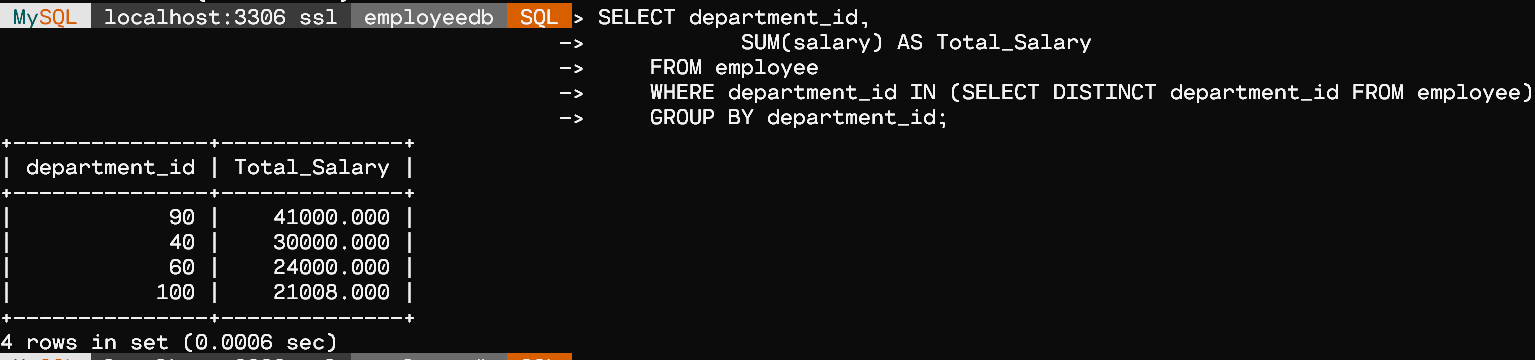
    SELECT department\_id,

           SUM(salary) AS Total\_Salary

    FROM employee

    WHERE department\_id IN (SELECT DISTINCT department\_id FROM employee)

    GROUP BY department\_id;



c) Write a subquery that returns a set of rows to find all departments that do actually have one or more employees assigned to them.

    SELECT DISTINCT department\_id FROM employee;



d) Write a query in SQL to display the first and last name, salary, and department ID for all those employees who earn more than the average salary and arrange the list in descending order on salary.

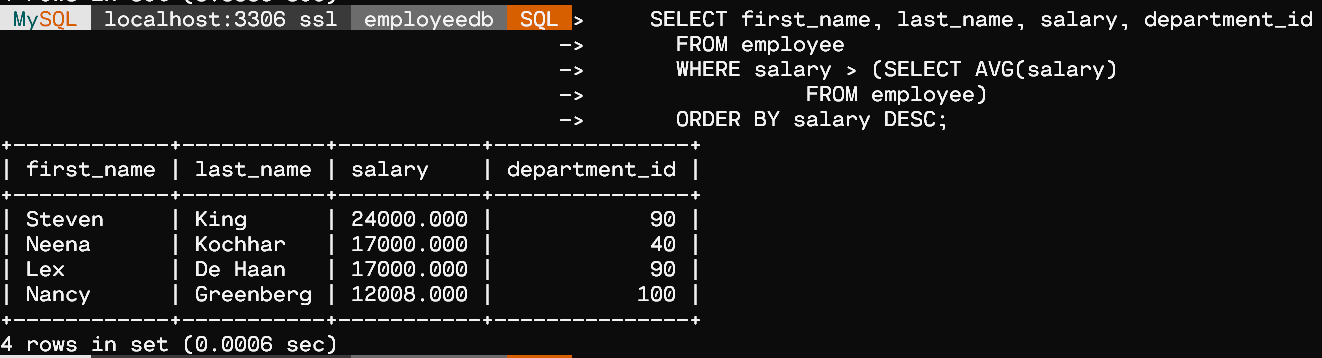
    SELECT first\_name, last\_name, salary, department\_id

      FROM employee

      WHERE salary > (SELECT AVG(salary)

                FROM employee)

      ORDER BY salary DESC;



e) Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn more than the maximum salary of a department which ID is 40.

      SELECT first\_name, last\_name, salary, department\_id

          FROM employee

          WHERE salary > (SELECT MAX(salary)

                    FROM employee

                            WHERE department\_id = 40);

