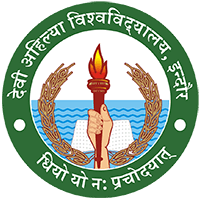
**Institute of Engineering & Technology**

**Devi Ahilya Vishwavidyalaya, Indore (M.P)**

**Department of Computer Science & Engineering**



**DATABASE MANAGEMENT SYSTEM(CER4C4)**

**Lab Assignment-6**

**Submitted To: Submitted By:**

**Mrs. Jyoti Haveliya Mam Tanishq Chauhan (21C4184)**

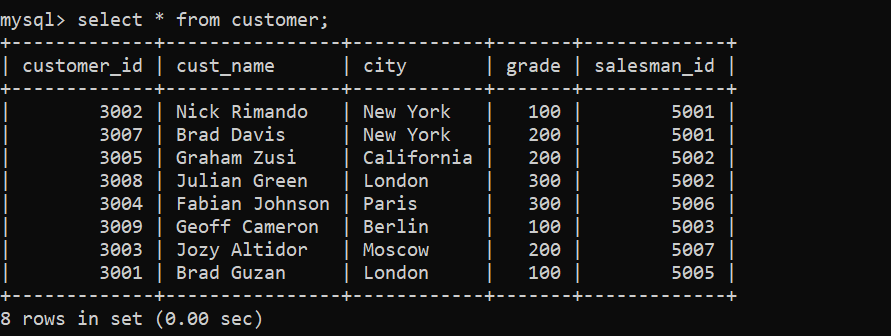
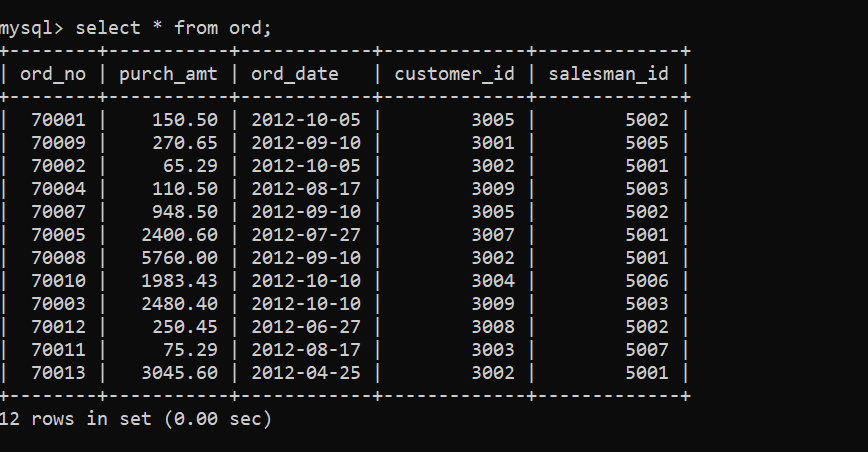
**CS-Dept CS “B” 2nd Year**

**IET-DAVV**

**Lab Assignment-6**

**Create three tables salesman, customer and order with following data respectively:**

****

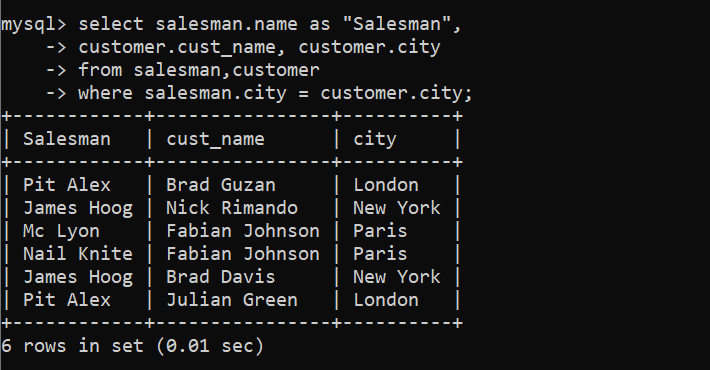
****

**Now solve the following query:-**

1. **WAQ to find the salesperson and customer who belongs to same city. Return Salesman, cust\_name and city.**

**Query:-** select salesman.name as “Salesman”,

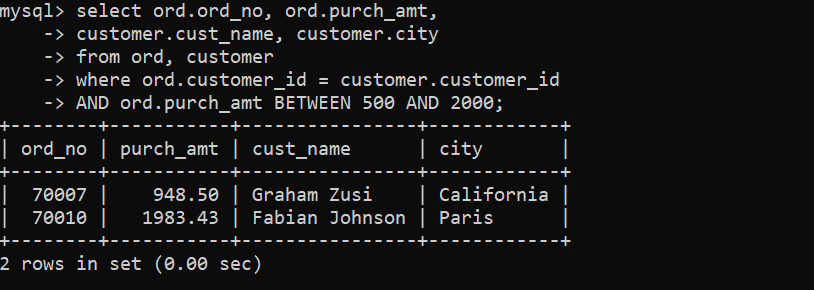
* Customer.cust\_name, customer.city
* From salesman, customer
* Where salesman.city = customer.city;

****

1. **WAQ to find those orders where order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.**

**Query:-** select ord.ord\_no, ord.purch-amt,

* customer.cust\_name, customer.city
* from ord, customer
* where ord.customer\_id = customer.customer\_id
* AND ord.purch\_amt BETWEEN 500 AND 2000;

****

1. **WAQ to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.**

**Query:-** select c.cust\_name as “Customer Name”

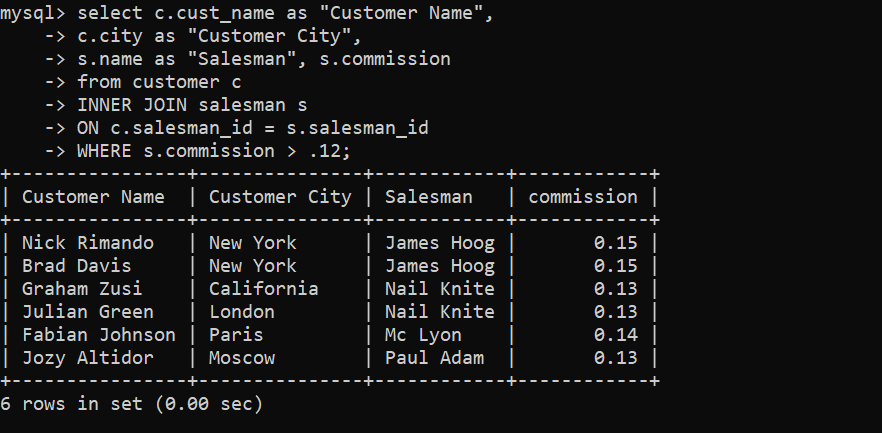
* c.city, s.name as “Salesman”, s.commission
* from customer c
* INNER JOIN salesman s
* ON c.salesman\_id = s.salesman\_id;

****

1. **WAQ to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission.**

**Query:-** select c.cust\_name as “Customer Name”,

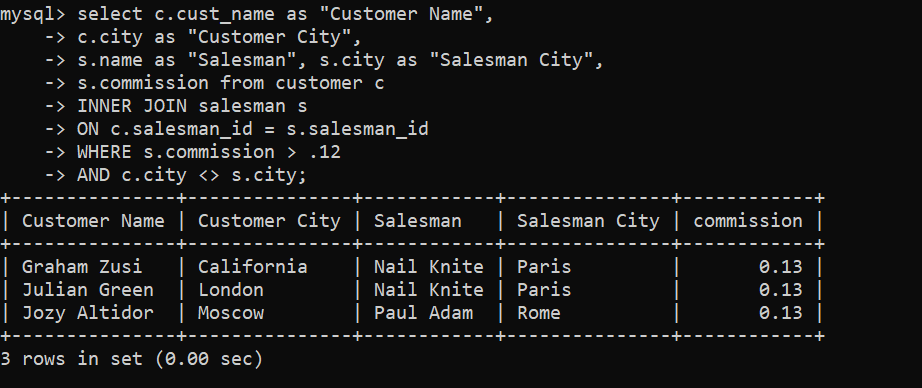
* c.city as “Customer City”,
* s.name as “Salesman”, s.commission
* from customer c
* INNER JOIN salesman s
* ON c.salesman\_id = s.salesman\_id
* WHERE s.commission > .12;

****

1. **WAQ to find those salespersons who do not live in the same city where their customers live and received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, salesman city, commission.**

**Query:-** select c.cust\_name as “Customer Name”,

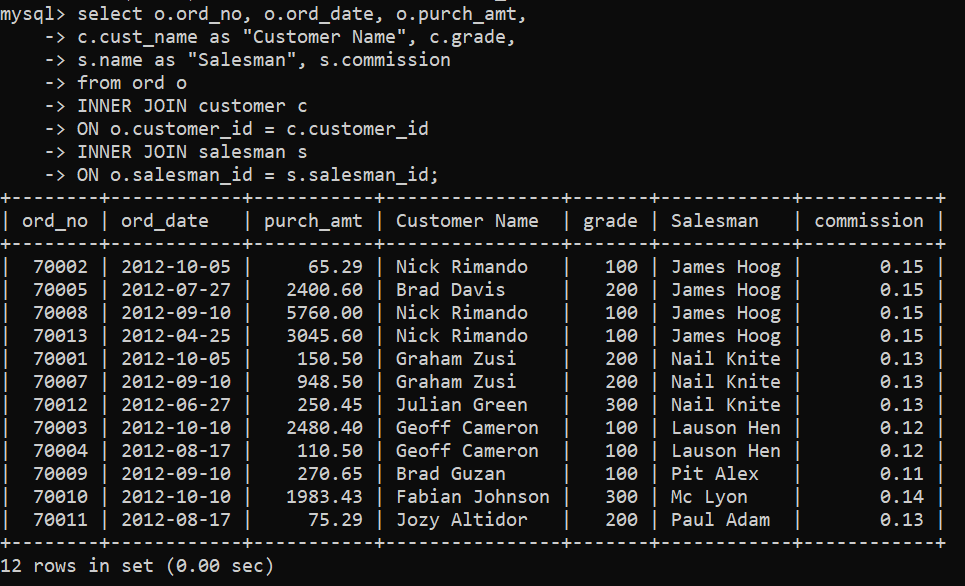
* c.city as “Customer City”,
* s.name as “Salesman”, s.city as “Salesman City”,
* s.commission from customer c
* INNER JOIN salesman s
* ON c.salesman\_id = s.salesman\_id
* WHERE s.commission > .12
* AND c.city <> s.city;

****

1. **WAQ to find the details of an order. Return ord\_no, ord\_date, purch\_amt, Customer Name, grade, Salesman, commission.**

**Query:-** select o.ord\_no, o.ord\_date, o.purch\_amt,

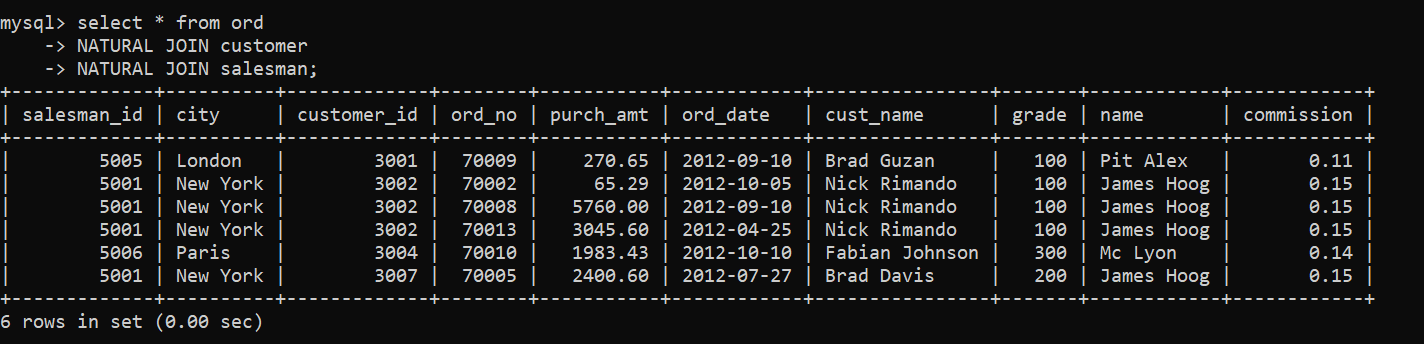
* c.cust\_name as “Customer Name”, c.grade,
* s.name as “Salesman”, s.commission
* from ord o
* INNER JOIN customer c
* ON o.customer\_id = c.customer\_id
* INNER JOIN salesman s
* ON o.salesman\_id = s.salesman\_id;

****

1. **Write a statement to make a join on the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.**

**Query:-** select \* from ord

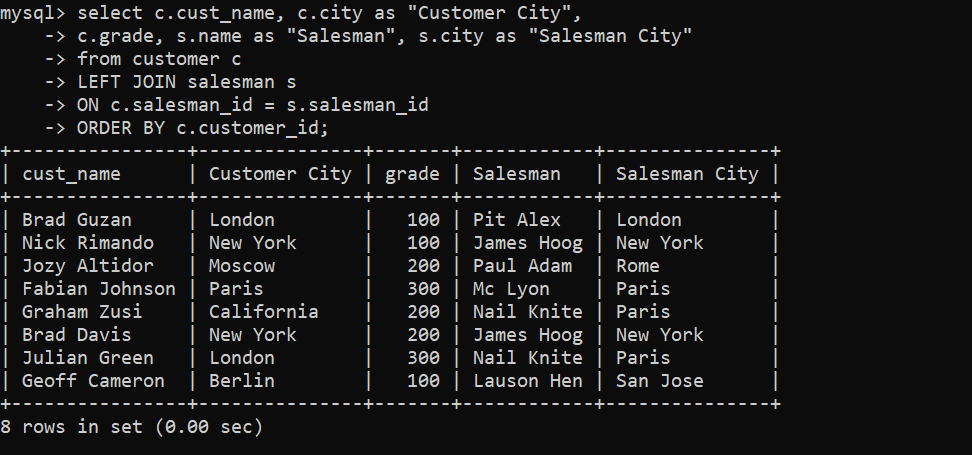
* NATURAL JOIN customer
* NATURAL JOIN salesman;

****

1. **WAQ to display the cust\_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending on customer\_id.**

**Query:-** select c.cust\_name, c.city as “Customer City”,

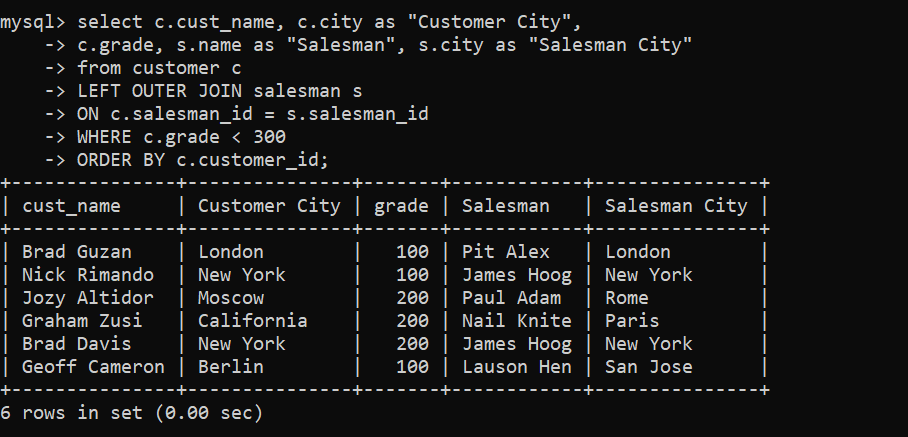
* c.grade, s.name as “Salesman”, s.city as “Salesman City”
* from customer c
* LEFT JOIN salesman s
* ON c.salesman\_id = s.salesman\_id
* ORDER BY c.customer\_id;

****

1. **WAQ to find those customers whose grade less than 300. Return cust\_name, customer city, grade, Salesman, saleman city. The result should be ordered by ascending customer\_id.**

**Query:-** select c.cust\_name, c.city as “Customer City”,

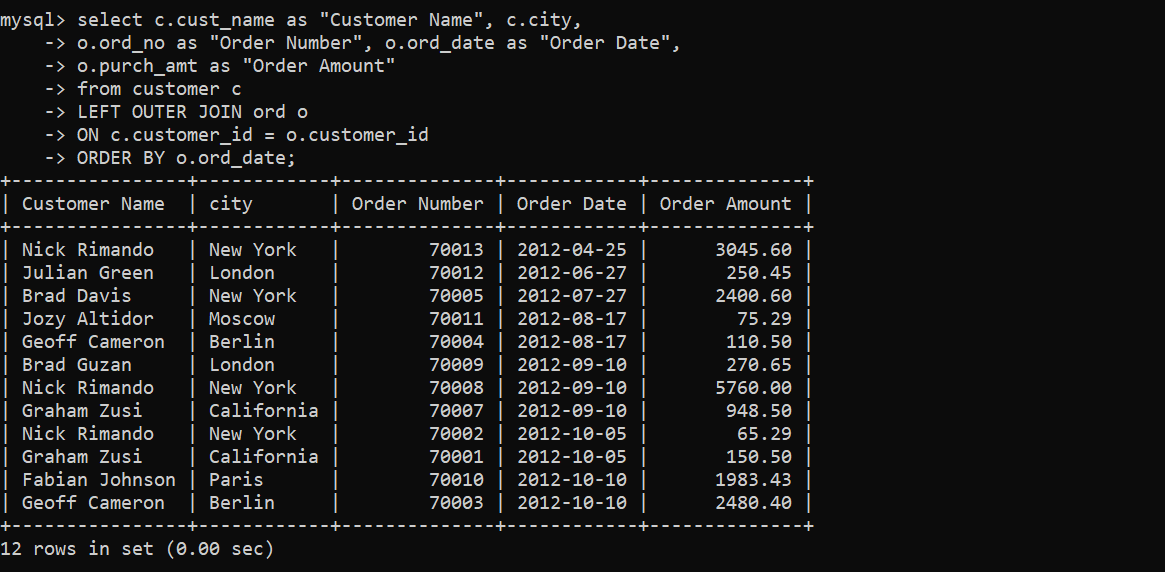
* c.grade, s.name as “Salesman”, s.city as “Salesman city”
* from customer c
* LEFT OUTER JOIN salesman s
* ON c.salesman\_id = s.salesman\_id
* WHERE c.grade < 300
* ORDER BY c.customer\_id;

****

1. **Write a 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.**

**Query:-** select c.cust\_name as “Customer Name”, c.city,

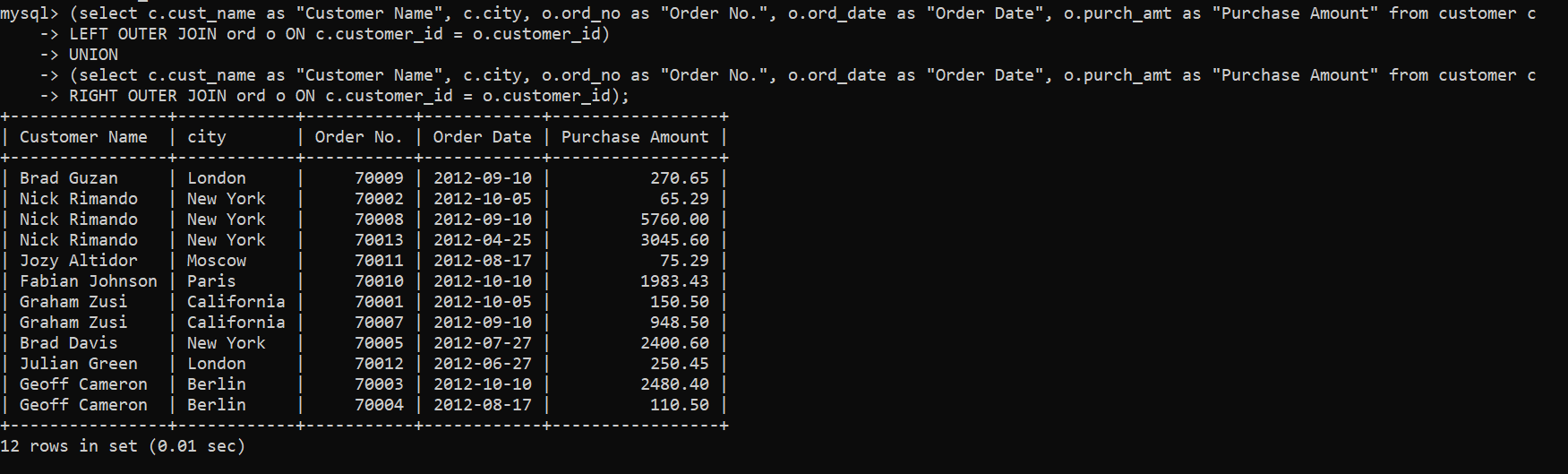
* o.ord\_no as “Order Number”, o.ord\_date as “Order Date”,
* o.purch\_amt as “Order Amount”
* from customer c
* LEFT OUTER JOIN ord o
* ON c.customer\_id = o.customer\_id
* ORDER BY o.ord\_date;

****

1. **Write a statement to make a report with customer name, city, order no. order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who is not on the list.**

**Query:-** (select c.cust\_name as "Customer Name", c.city, o.ord\_no as "Order No.", o.ord\_date as "Order Date", o.purch\_amt as "Purchase Amount" from customer c

* LEFT OUTER JOIN ord o ON c.customer\_id = o.customer\_id)
* UNION
* (select c.cust\_name as "Customer Name", c.city, o.ord\_no as "Order No.", o.ord\_date as "Order Date", o.purch\_amt as "Purchase Amount" from customer c
* RIGHT OUTER JOIN ord o ON c.customer\_id = o.customer\_id);

****