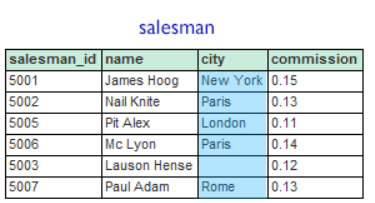
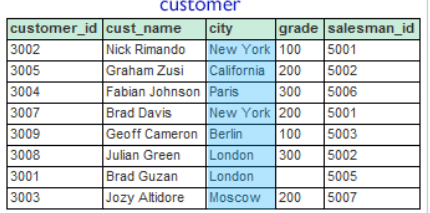
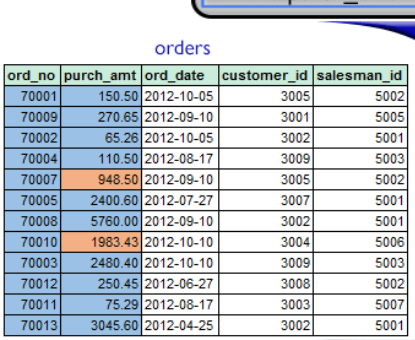
**Assignment 2**

***Sample table*: salesman**

****

***Sample table*: customer**

****

****

1. From the above tables write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city.

**CODE**:-

|  |
| --- |
| SELECT S.NAME AS SALESMAN, C.cust\_name AS CUSTOMER, S.city AS City FROM SALESMAN S INNER JOIN Customer C ON S.city = C.city; |

**OUTPUT**:-

|  |
| --- |
|  |

1. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.

**CODE:-**

|  |
| --- |
| SELECT  O.ORD\_NO,  O.PURCH\_AMT,  C.CUST\_NAME,  C.CITY FROM ORDERS O INNER JOIN CUSTOMER C ON O.CUSTOMER\_ID = C.CUSTOMER\_ID WHERE O.PURCH\_AMT BETWEEN 500 AND 2000; |

**OUTPUT:-**

|  |
| --- |
|  |

1. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.

**CODE:-**

|  |
| --- |
| **SELECT**  **C.CUST\_NAME AS "Customer Name",**  **C.CITY AS "City",**  **S.NAME AS "Salesman",**  **S.COMMISSION AS "Commission"**  **FROM CUSTOMER C INNER JOIN SALESMAN S ON C.SALESMAN\_ID = S.SALESMAN\_ID;** |

**OUTPUT:-**

|  |
| --- |
|  |

1. write a SQL query to find salespeople who received more than 12 percent commissions from the company. Return Customer Name, customer city, Salesman, commission.

**CODE:-**

|  |
| --- |
| SELECT  C.CUST\_NAME AS "Customer Name",  C.CITY AS "Customer City",  S.NAME AS "Salesman",  S.COMMISSION AS "Commission"  FROM CUSTOMER C  INNER JOIN SALESMAN S  ON C.salesman\_id = S.salesman\_id  WHERE S.commission > 0.12; |

**Output:-**

|  |
| --- |
|  |

1. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission.

**CODE:-**

|  |
| --- |
| SELECT  C.CUST\_NAME AS "Customer Name",  C.CITY AS "Customer City",  S.NAME AS "Salesman",  S.CITY AS "Salesman City",  S.COMMISSION AS "Commission"  FROM CUSTOMER C  INNER JOIN SALESMAN S  ON C.salesman\_id = S.salesman\_id  WHERE S.commission > 0.12  AND C.city <> S.city; |

**OUTPUT:-**

|  |
| --- |
|  |

1. write a SQL query to find the details of an order. Return ord\_no, ord\_date, purch\_amt, Customer Name, grade, Salesman, commission.

**CODE:-**

|  |
| --- |
| SELECT  O.ORD\_NO AS "Order Number",  O.ORD\_DATE AS "Order Date",  O.PURCH\_AMT AS "Purchase Amount",  C.CUST\_NAME AS "Customer Name",  C.GRADE AS "Grade",  S.NAME AS "Salesman",  S.COMMISSION AS "Commission"  FROM ORDERS O  INNER JOIN CUSTOMER C  ON O.CUSTOMER\_ID = C.CUSTOMER\_ID  INNER JOIN SALESMAN S  ON C.SALESMAN\_ID = S.SALESMAN\_ID; |

**OUTPUT:-**

|  |
| --- |
|  |

1. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

**CODE:-**

|  |
| --- |
| SELECT  S.SALESMAN\_ID,  S.NAME,  S.CITY AS "Salesman City",  S.COMMISSION,  C.CUSTOMER\_ID,  C.CUST\_NAME,  C.CITY AS "Customer City",  C.GRADE,  O.ORD\_NO,  O.ORD\_DATE,  O.PURCH\_AMT  FROM ORDERS O  INNER JOIN CUSTOMER C  ON O.CUSTOMER\_ID = C.CUSTOMER\_ID  INNER JOIN SALESMAN S  ON C.SALESMAN\_ID = S.SALESMAN\_ID; |

**OUTPUT:-**

|  |
| --- |
|  |

1. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer\_id.

**CODE:-**

|  |
| --- |
| SELECT  C.CUST\_NAME AS "Customer Name",  C.CITY AS "Customer City",  C.GRADE AS "Grade",  S.NAME AS "Salesman",  S.CITY AS "Salesman City"  FROM CUSTOMER C  INNER JOIN SALESMAN S  ON C.SALESMAN\_ID = S.SALESMAN\_ID  ORDER BY C.CUSTOMER\_ID ASC; |

**OUTPUT:-**

|  |
| --- |
|  |

1. write a SQL query to find those customers with a grade less than 300. Return cust\_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer\_id.

**CODE:-**

|  |
| --- |
| SELECT  C.CUST\_NAME AS "Customer Name",  C.CITY AS "Customer City",  C.GRADE AS "Grade",  S.NAME AS "Salesman",  S.CITY AS "Salesman City"  FROM CUSTOMER C  INNER JOIN SALESMAN S  ON C.SALESMAN\_ID = S.SALESMAN\_ID  WHERE C.GRADE < 300  ORDER BY C.CUSTOMER\_ID ASC; |

**OUTPUT:-**

|  |
| --- |
|  |

1. 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 determine whether any existing customers have placed an order.

**CODE:-**

|  |
| --- |
| SELECT  C.CUST\_NAME AS "Customer Name",  C.CITY AS "Customer City",  O.ORD\_NO AS "Order Number",  O.ORD\_DATE AS "Order Date",  O.PURCH\_AMT AS "Order Amount"  FROM CUSTOMER C  INNER JOIN ORDERS O  ON C.CUSTOMER\_ID = O.CUSTOMER\_ID  ORDER BY O.ORD\_DATE ASC; |

**OUTPUT:-**

|  |
| --- |
|  |