

*ASSIG NM EN T - 3*

CREATE TABLE Customer ( cust\_id INT PRIMARY KEY, cust\_name VARCHAR(255), city VARCHAR(255),

grade INT

);

CREATE TABLE Salesman ( salesman\_id INT PRIMARY KEY, salesman\_name VARCHAR(255), city VARCHAR(255),

commission DECIMAL(5,2)

);

CREATE TABLE Orders ( ord\_no INT PRIMARY KEY, ord\_date DATE, purch\_amt DECIMAL(8,2), cust\_id INT,

salesman\_id INT,

FOREIGN KEY (cust\_id) REFERENCES Customer(cust\_id),

FOREIGN KEY (salesman\_id) REFERENCES Salesman(salesman\_id)

);

INSERT INTO Customer VALUES

(1, 'Customer1', 'City1', 200),

(2, 'Customer2', 'City2', 300),

(3, 'Customer3', 'City3', 250);

INSERT INTO Salesman VALUES

(101, 'Salesman1', 'City1', 0.15),

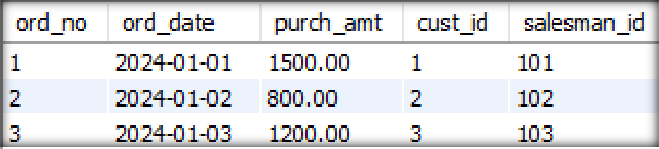
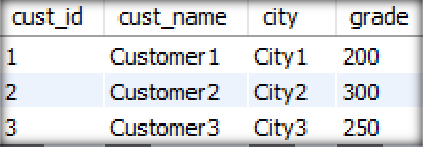
(102, 'Salesman2', 'City2', 0.12),

(103, 'Salesman3', 'City3', 0.18);

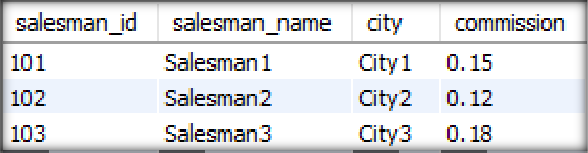
INSERT INTO Orders VALUES (1, '2024-01-01', 1500, 1, 101),

(2, '2024-01-02', 800, 2, 102),

(3, '2024-01-03', 1200, 3, 103);



:: TABLES ::

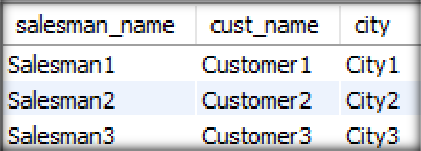


QUESTION – 1:

Write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name, and city.

Query

select Salesman.salesman\_name,Customer.cust\_name,Customer.city from Salesman inner join Customer on Customer.city=Salesman.city;

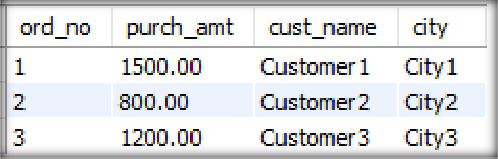


QUESTION – 2:

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.

Query

select ord\_no,purch\_amt,cust\_name,city from Orders inner join Customer on Orders.cust\_id=Customer.cust\_id where Orders.purch\_amt between 500 and 2000;

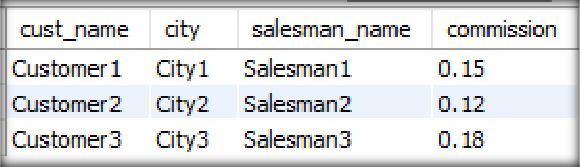


QUESTION – 3:

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

Query

select Customer.cust\_name,Customer.city,Salesman.salesman\_name,Salesman.commission from Salesman inner join Customer on Customer.city=Salesman.city;

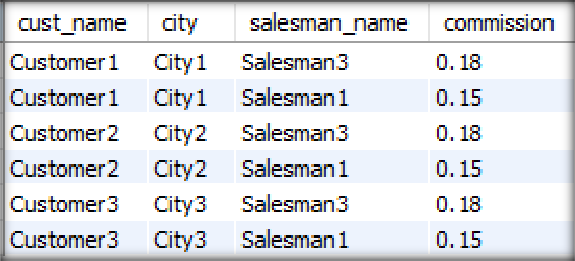


QUESTION – 4:

Write a SQL query to find salespeople who received commissions of more than 0.12 percent from the company. Return Customer Name, customer city, Salesman, commission.

Query

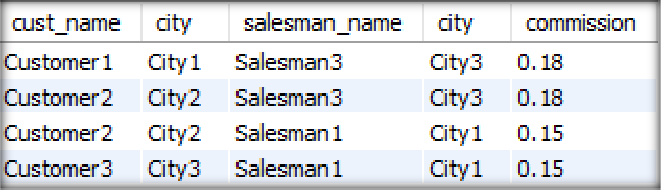
select Customer.cust\_name,Customer.city,Salesman.salesman\_name,Salesman.commission from Salesman inner join Customer where Salesman.commission>0.12;



QUESTION – 5:

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. Query

select Customer.cust\_name,Customer.city,Salesman.salesman\_name,Salesman.city,Salesman.commission from Salesman inner join Customer on Customer.city!=Salesman.city where Salesman.commission>0.12;



QUESTION – 6:

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

Query

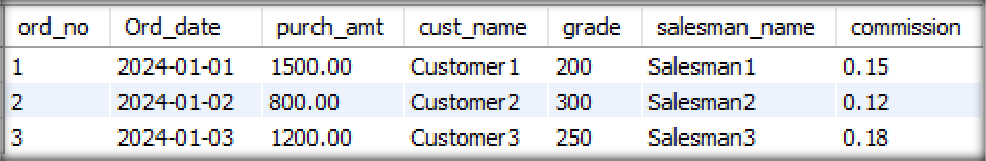
select

Orders.ord\_no,Orders.Ord\_date,Orders.purch\_amt,Customer.cust\_name,Customer.grade,Salesman.salesma n\_name,Salesman.commission

from Orders

inner join Customer on Orders.cust\_id=Customer.cust\_id

inner join Salesman on Orders.salesman\_id=Salesman.salesman\_id;



QUESTION – 7:

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.

Query

SELECT

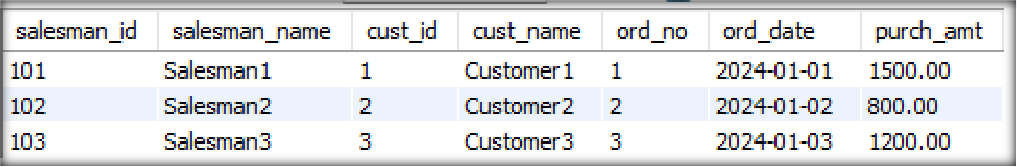
salesman.salesman\_id, salesman.salesman\_name, customer.cust\_id,

customer.cust\_name, orders.ord\_no,

orders.ord\_date, orders.purch\_amt

FROM orders

JOIN salesman ON orders.salesman\_id = salesman.salesman\_id JOIN customer ON orders.cust\_id = customer.cust\_id;



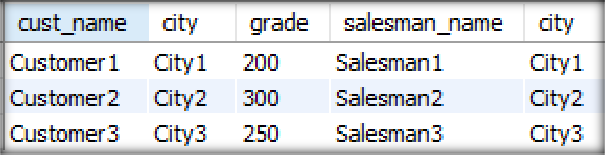
QUESTION – 8:

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

Query

select Customer.cust\_name,Customer.city,Customer.grade,Salesman.salesman\_name, Salesman.city from Customer

inner join Salesman on Customer.city=Salesman.city order by Customer.cust\_id asc;



QUESTION – 9:

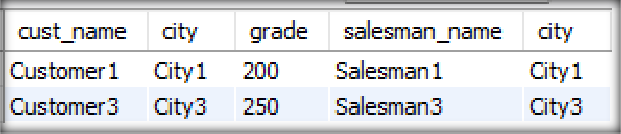
Write a SQL query to find those customers with a grade less than 300. Return cust\_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer\_id.

Query

select Customer.cust\_name, Customer.city,Customer.grade,Salesman.salesman\_name,Salesman.city from Customer

inner join Salesman on Customer.city=Salesman.city where Customer.grade<300

Order by Customer.cust\_id asc;

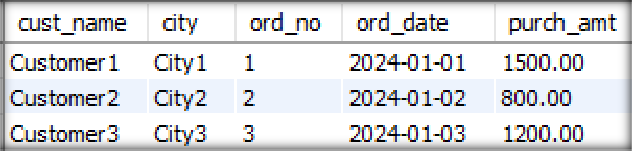


QUESTION – 10:

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 of the existing customers have placed an order or not.

Query

SELECT C.cust\_name, C.city, O.ord\_no, O.ord\_date, O.purch\_amt FROM Customer C LEFT JOIN Orders O ON C.cust\_id = O.cust\_id ORDER BY O.ord\_date ASC;



*ASSIG NM ENT - 4*

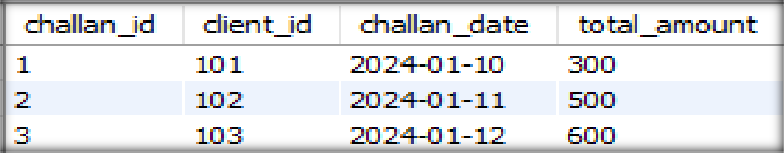


QUESTION - 1

**Retrieve all records from CHALLAN\_HEADER**

Query

SELECT \* FROM CHALLAN\_HEADER;

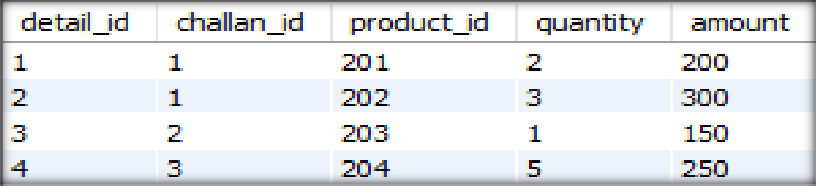


QUESTION - 2

Retrieve all records from CHALLAN\_DETAILS

Query

SELECT \* FROM CHALLAN\_DETAILS;



QUESTION - 3

Retrieve details of a specific challan along with associated products

Query

select CH.challan\_id,CH.client\_id,

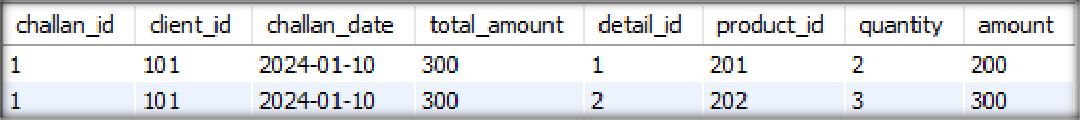
ch.challan\_date, ch.total\_amount, cd.detail\_id,cd.product\_id,

cd.quantity,cd.amount

from

CHALLAN\_HEADER as CH

Join CHALLAN\_DETAILS AS CD on CH.challan\_id = CD.challan\_id WHERE CD.challan\_id = 1;

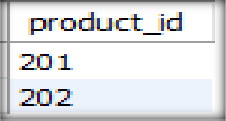


QUESTION - 4

Retrieve details of products in a specific challan

Query

select product\_id from challan\_details where challan\_id=1;



QUESTION - 5

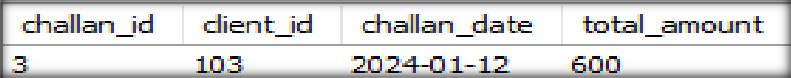
Retrieve all Challan Headers with total amount greater than 500.00

Query

select \*

from CHALLAN\_HEADER

where total\_amount>500;



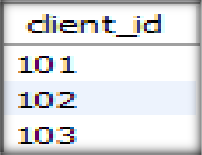
QUESTION - 6

Retrieve all distinct clients who have made a challan

Query

select distinct client\_id from CHALLAN\_HEADER GROUP BY

client\_id ;



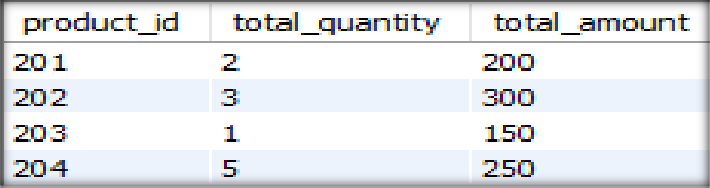
QUESTION - 7

Retrieve the total quantity and amount for each product across all challans:

Query

select product\_id, sum(quantity) as total\_quantity, sum(amount) as total\_amount from challan\_details

group by product\_id;



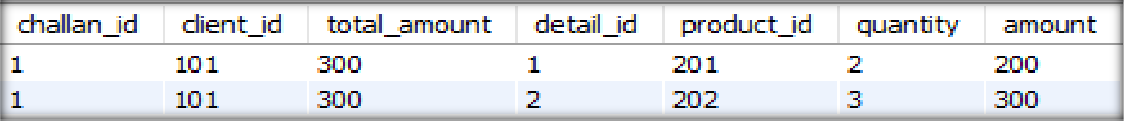
QUESTION - 8

Retrieve the challan details along with the client information for a specific challan

Query

select ch.challan\_id,ch.client\_id,ch.total\_amount,cd.detail\_id,cd.product\_id,cd.quantity,cd.amount from challan\_header ch

join challan\_details cd on ch.challan\_id=cd.challan\_id where ch.challan\_id = 1 ;



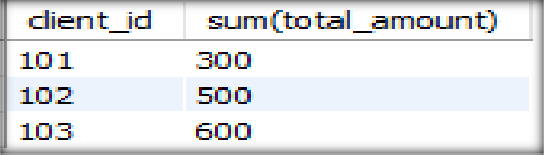
QUESTION - 9

Retrieve the total amount for each client based on all their challans

Query

select distinct client\_id,sum(total\_amount) from challan\_header

group by client\_id ;



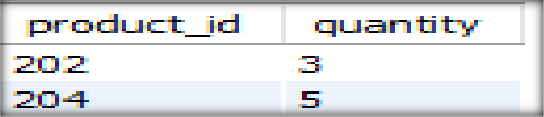
QUESTION - 10

Retrieve the products with quantity greater than 2 in any challan

Query

select product\_id, quantity from challan\_details

where quantity > 2 ;



QUESTION - 11

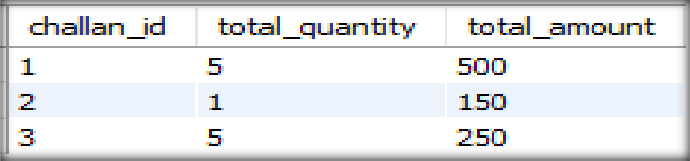
Retrieve all challans along with the total quantity and amount for each challan

Query

select challan\_id,

sum(quantity) as total\_quantity, sum(amount) as total\_amount from challan\_details

group by challan\_id;



QUESTION - 12

Retrieve the details of the latest challan made by each client

Query

select max(challan\_id),client\_id,max(challan\_date) from challan\_header group by client\_id;

