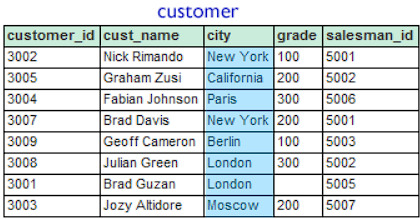
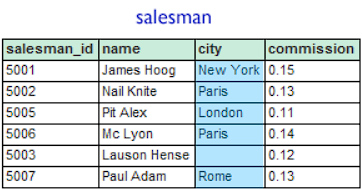
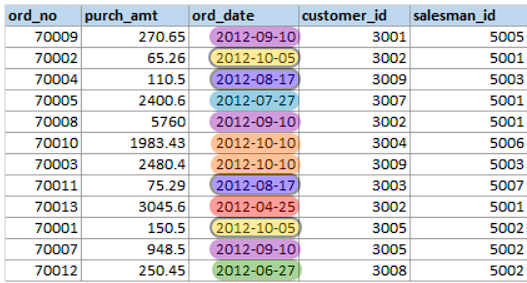
**Clauses, Joins and Subqueries**



ORDERS



Refer the datasets given above to answer the following questions:

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

SELECT

c.cust\_name AS Customer\_Name,

c.city AS Customer\_City,

s.name AS Salesman,

s.commission AS Commission

FROM

customer c

JOIN

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

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

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

JOIN

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

ORDER BY

c.customer\_id ASC;

1. From the following tables 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.

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

JOIN

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

WHERE

c.city <> s.city

AND s.commission > 12;

1. From the following tables write a SQL query to count the number of customers with grades above the average in New York City. Return grade and count.

SELECT

c.grade AS Grade,

COUNT(\*) AS Count

FROM

customer c

WHERE

c.city = 'New York'

AND c.grade > (SELECT AVG(grade) FROM customer WHERE city = 'New York')

GROUP BY

c.grade;

1. Write a query to find the sums of the amounts from the **orders table**, grouped by date, and eliminate all dates where the sum was not at least 1000.00 above the maximum order amount for that date.

SELECT

ord\_date,

SUM(purch\_amt) AS Total\_Amount

FROM

ORDERS

GROUP BY

ord\_date

HAVING

SUM(purch\_amt) >= 1000.00 + (SELECT MAX(purch\_amt) FROM ORDERS o2 WHERE o2.ord\_date = ORDERS.ord\_date);

1. From the following tables write a SQL query to find the salespeople who deal with customers with more than one order. Return salesman\_id, name, city and commission.

SELECT DISTINCT

s.salesman\_id,

s.name,

s.city,

s.commission

FROM

salesman s

JOIN

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

JOIN

ORDERS o ON c.customer\_id = o.customer\_id

GROUP BY

s.salesman\_id, s.name, s.city, s.commission

HAVING

COUNT(o.ord\_no) > 1;

1. Refer the following table for the dataset: [SQL Week 2 dataset](https://docs.google.com/spreadsheets/d/1pFA6B656WdfdYLl1OHRwv2adE6hLSeLTFN30K8gHxwM/edit?usp=sharing)  
   Get a list of **cust\_city**, *sum* of **opening\_amt**, *average* of **receive\_amt** and *maximum* **payment\_amt** from customer table with following conditions-

a. Grade of customer table must be 2

b. *Average* of **receive\_amt** for each group of **cust\_city** must be more than *500*

SELECT

CUST\_CITY,

SUM(OPENING\_AMT) AS Total\_Opening\_Amt,

AVG(RECEIVE\_AMT) AS Avg\_Receive\_Amt,

MAX(PAYMENT\_AMT) AS Max\_Payment\_Amt

FROM

dataset

WHERE

GRADE = 2

GROUP BY

CUST\_CITY

HAVING

AVG(RECEIVE\_AMT) > 500;

1. From the table “orders”, arrange the entries in ascending order of their **ord\_no**. Then return all the entries which have **purch\_amt** > Rs. 500.

SELECT \*

FROM ORDERS

WHERE purch\_amt > 500

ORDER BY ord\_no ASC;