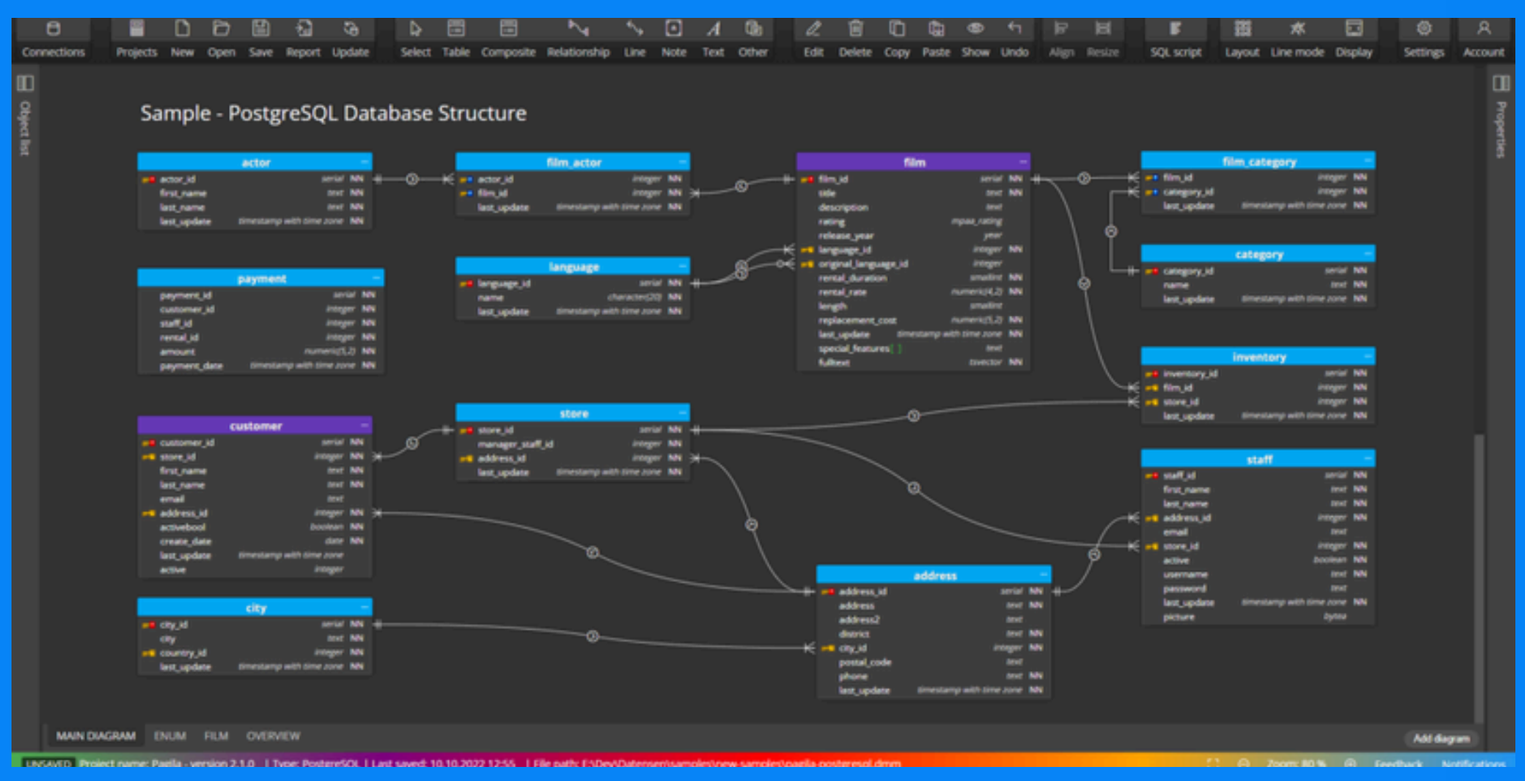
SQL Practice Sheet



SQL Practice 1

One table, Aggregation, Group By
24 Queries

Link: https://www.w3resource.com/sql-exercises/

salesman

salesman_id	name	city	commission
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
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

customer

customer_id	customer_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moncow	200	5007

order

order_no	*purch_amt*	*order_date*	*customer_id*	*salesman_id*
70001	*150.5*	*2016-10-05*	*3005*	*5002*
70009	*270.65*	*2016-09-10*	*3001*	
70002	*65.26*	*2016-10-05*	*3002*	*5001*
70004	*110.5*	*2016-08-17*	*3009*	
70007	*948.5*	*2016-09-10*	*3005*	*5002*
70005	*2400.6*	*2016-07-27*	*3007*	*5001*
70008	*5760*	*2016-09-10*	*3002*	*5001*
70010	*1983.43*	*2016-10-10*	*3004*	*5006*
70003	*2480.4*	*2016-10-10*	*3009*	
70012	*250.45*	*2016-06-27*	*3008*	*5002*
70011	*75.29*	*2016-08-17*	*3003*	*5007*

Query 1

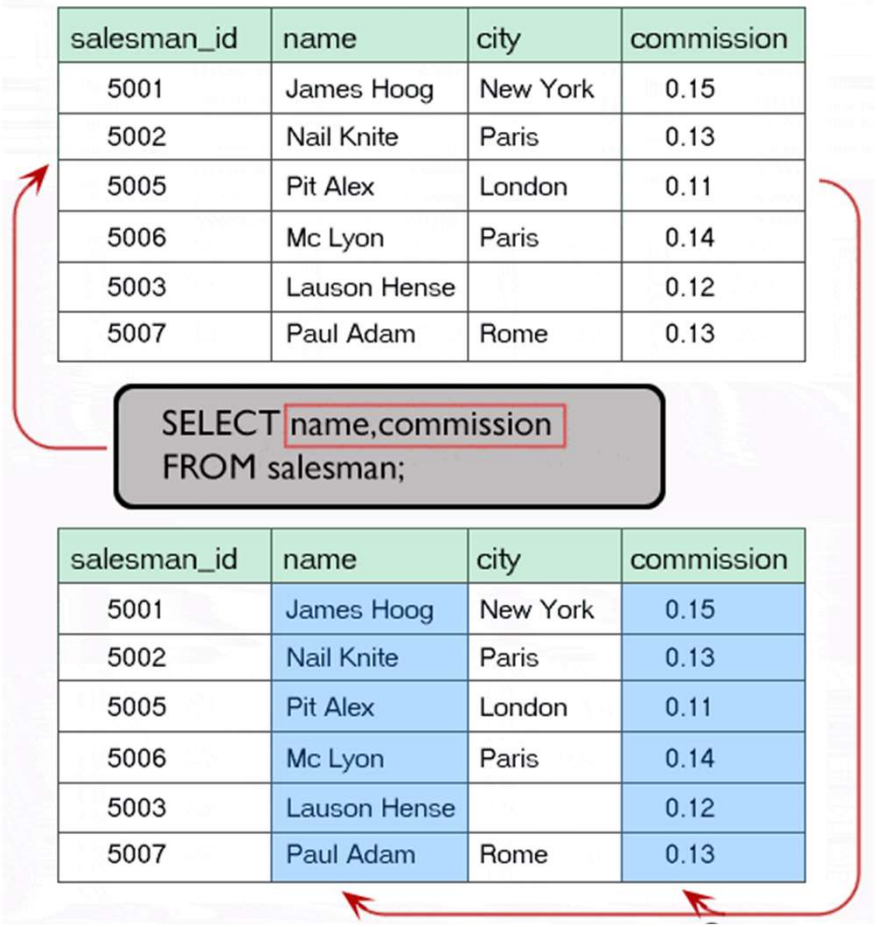
- Display name and commission of all the salesmen.

Query 1

- Display name and commission for all the salesmen.

name	commission
James Hoog	0.15
Nail Knite	0.13
Pit Alex	0.11
Mc Lyon	0.14
Paul Adam	0.13
Lauson Hen	0.12

SELECT name, commission
FROM salesman;



Query 2

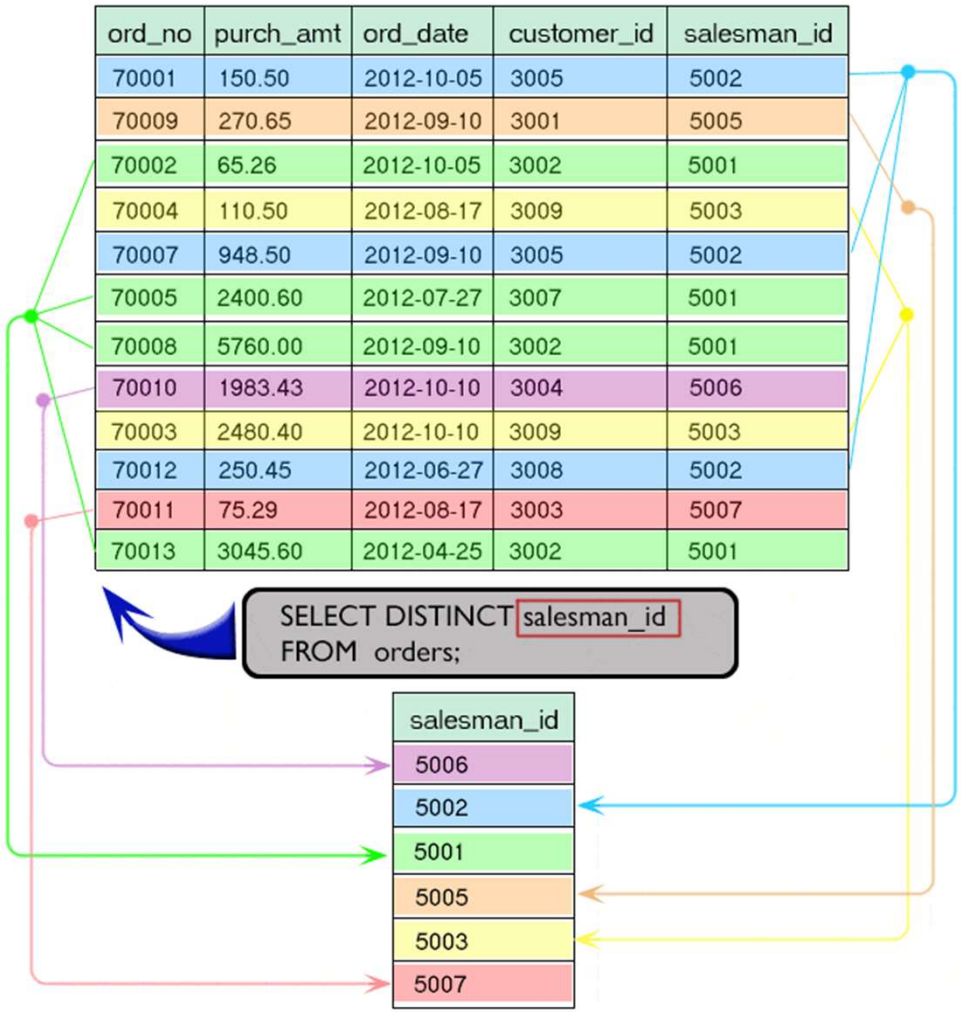
- Retrieve salesman id of all salesmen from orders table without any repeats.

Query 2

- Retrieve salesman id of all salesmen from orders table without any repeats.

salesman_id
5002
5003
5006
5001
5005
5007

SELECT DISTINCT salesman_id
FROM orders;



Query 3

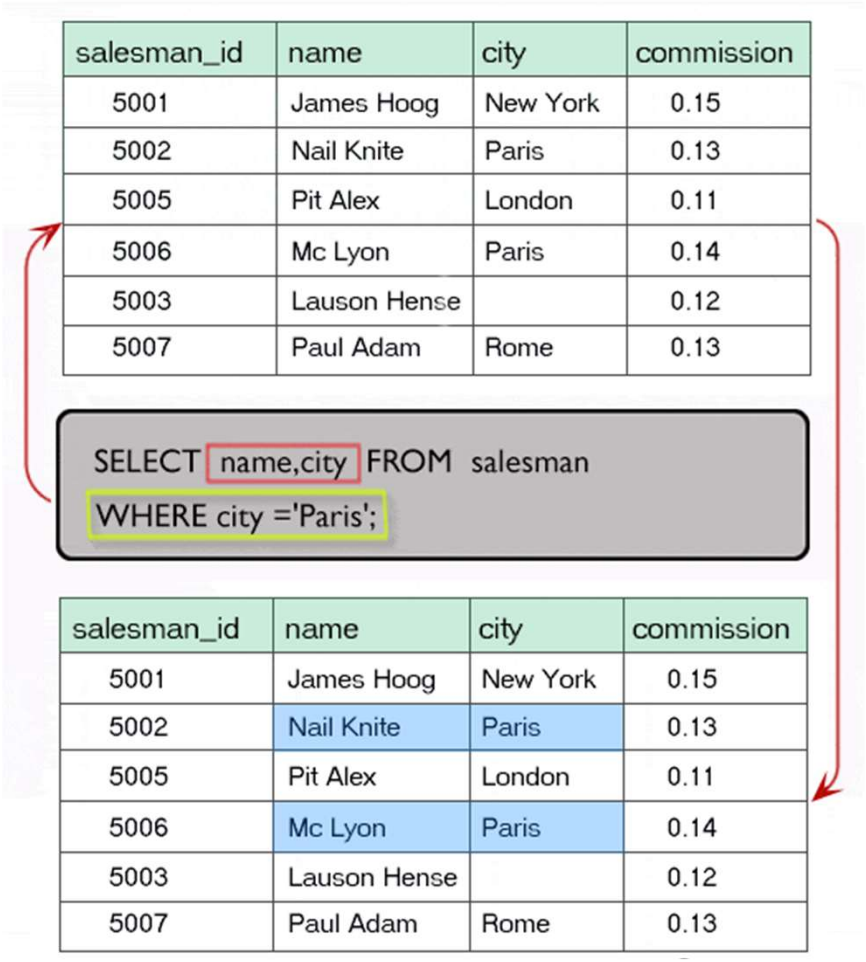
- Display names and city of salesman, who belongs to the city of Paris.

Query 3

- Display names and city of salesman, who belongs to the city of Paris.

name	city
Nail Knite	Paris
Mc Lyon	Paris

SELECT name,city
FROM salesman
WHERE city='Paris';

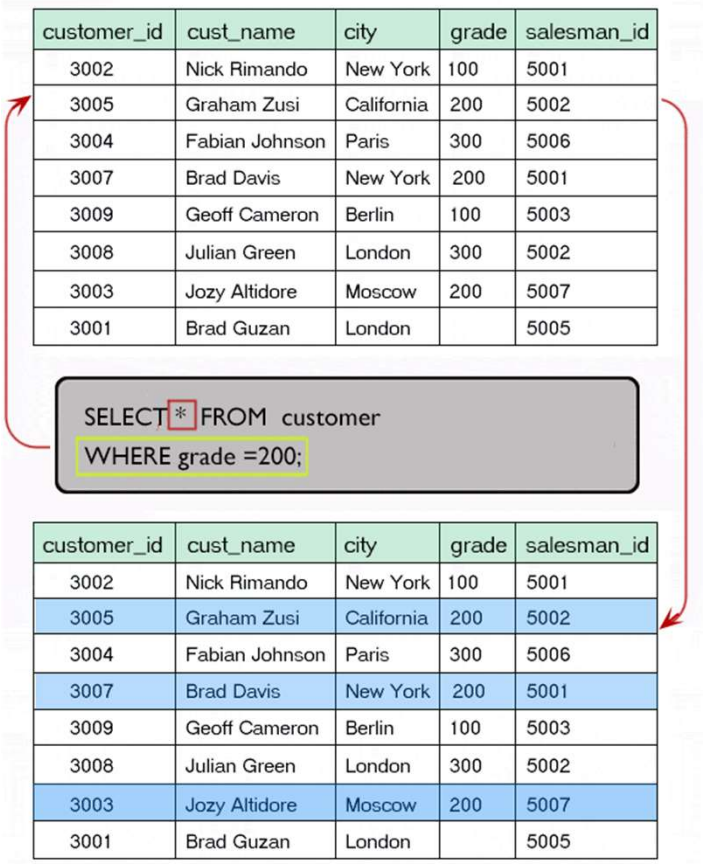


Query 4

- Display all the information for those customers with a grade of 200.

customer_id	cust_name	city	grade	salesman_id
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3003	Jozy Altidor	Moscow	200	5007

SELECT *
FROM customer
WHERE grade = 200;

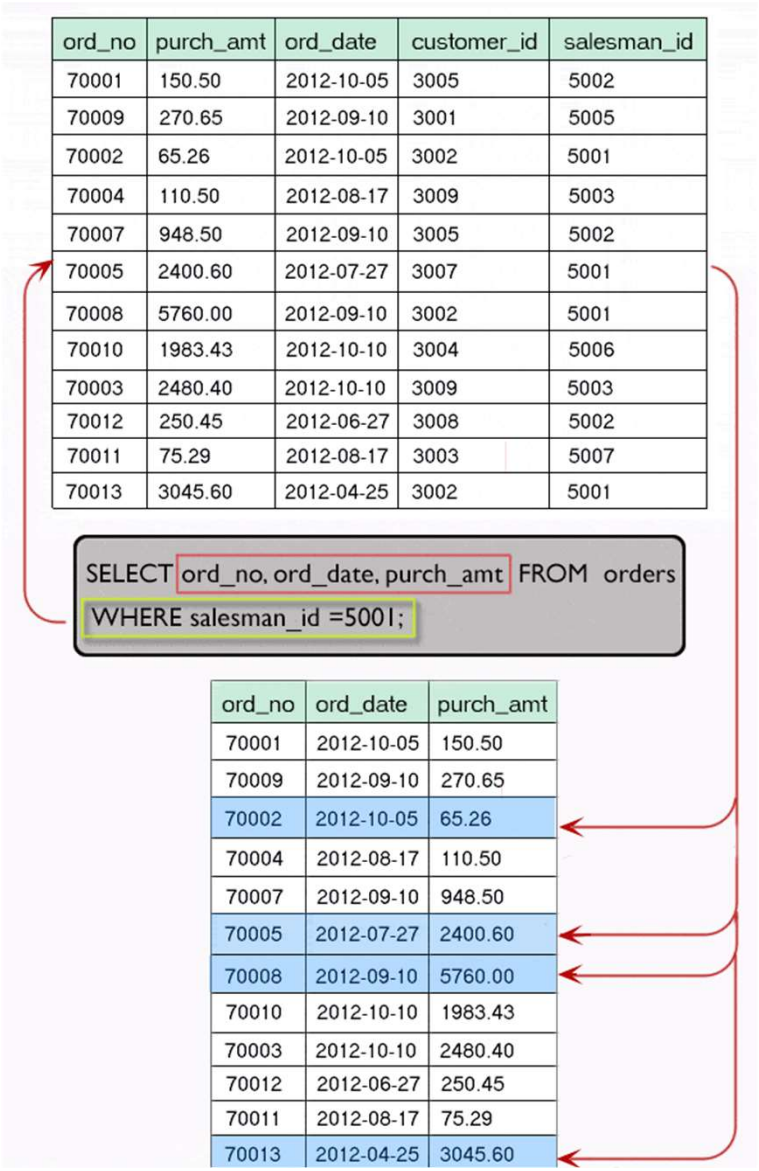


Query 5

- Display the order number, order date and the purchase amount for order(s) which will be delivered by the salesman with ID 5001.

ord_no	ord_date	purch_amt
70002	2012-10-05	65.26
70005	2012-07-27	2400.60
70008	2012-09-10	5760.00
70013	2012-04-25	3045.60

SELECT ord_no, ord_date, purch_amt
FROM orders
WHERE salesman_id = 5001;



Query 6 (table: nobel_win)

- Show the winner of the 1971 prize for Literature.

winner
Pablo Neruda

```
SELECT winner
FROM nobel_win
WHERE year = 1971
AND subject = 'Literature';
```

Query 7

- Show all the details of the winners with first name Louis.

year	subject	winner	country	category
1970	Physics	Louis Neel	France	Scientist

```
SELECT *
FROM nobel_win
WHERE winner LIKE 'Louis%';
```

Query 8

- Show all the winners in Physics for 1970 together with the winner of Economics for 1971.

year	subject	winner	country	category
1970	Physics	Hannes Alfven	Sweden	Scientist
1970	Physics	Louis Neel	France	Scientist
1971	Economics	Simon Kuznets	Russia	Economist

```
SELECT *
FROM nobel_win
WHERE (subject = 'Physics' AND year = 1970)
UNION
(SELECT *
FROM nobel_win
WHERE (subject = 'Economics' AND year = 1971)
);
```

Query 9

- Show all the winners of Nobel prize in the year 1970 except the subject Physiology and Economics.

year	subject	winner	country	category
1970	Physics	Hannes Alfven	Sweden	Scientist
1970	Physics	Louis Neel	France	Scientist
1970	Chemistry	Luis Federico Leloir	France	Scientist
1970	Literature	Aleksandr Solzhenitsyn	Russia	Linguist

```
SELECT *
FROM nobel_win
WHERE year = 1970
AND subject NOT IN ('Physiology','Economics');
```

Query 10

- Find all the details of the Nobel winners for the subject not started with the letter 'P' and arranged the list as the most recent comes first, then by name in order.

year	subject	winner	country	category
1994	Literature	Kenzaburo Oe	Japan	Linguist
1994	Economics	Reinhard Selten	Germany	Economist
1987	Chemistry	Donald J. Cram	USA	Scientist
1987	Chemistry	Jean-Marie Lehn	France	Scientist
1987	Literature	Joseph Brodsky	Russia	Linguist
1987	Economics	Robert Solow	USA	Economist
1971	Chemistry	Gerhard Herzberg	Germany	Scientist
1971	Literature	Pablo Neruda	Chile	Linguist
1971	Economics	Simon Kuznets	Russia	Economist
1970	Literature	Aleksandr Solzhenitsyn	Russia	Linguist
1970	Chemistry	Luis Federico Leloir	France	Scientist
1970	Economics	Paul Samuelson	USA	Economist

```
SELECT *
FROM nobel_win
WHERE subject NOT LIKE 'P%'
ORDER BY year DESC, winner;
```

Query 11 (table: item_mast)

- Find the name and price of the cheapest item(s).

pro_name	pro_price
ZIP drive	250.00
Mouse	250.00

```
SELECT pro_name, pro_price
FROM   item_mast
WHERE pro_price =  (SELECT MIN(pro_price)
                    FROM item_mast);
```

Query 12 (table: customer)

- Display all the customers, who are either belongs to the city New York or not had a grade above 100.

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3009	Geoff Cameron	Berlin	100	5003

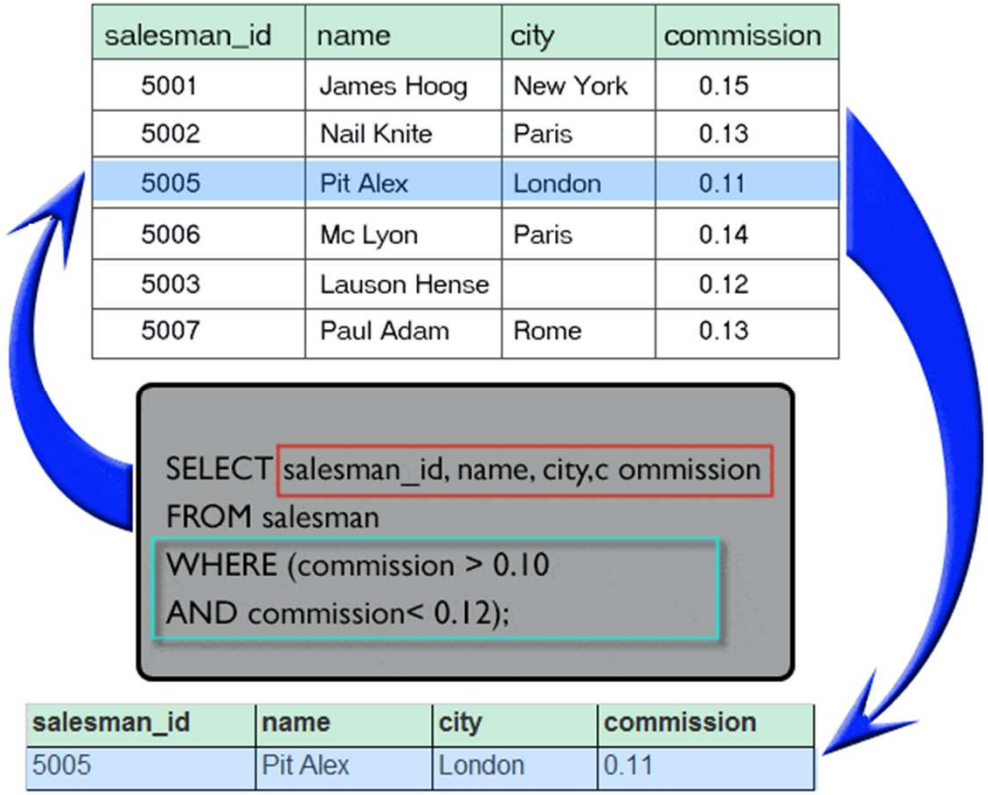
```
SELECT *
FROM customer
WHERE city = 'New York' OR NOT grade > 100;
```

Query 13 (table: salesman)

- Find those salesmen with all information who gets the commission within a range of 0.12 and 0.14.

```
SELECT salesman_id, name, city, commission
FROM salesman
WHERE (commission > 0.10 AND commission < 0.12);
```

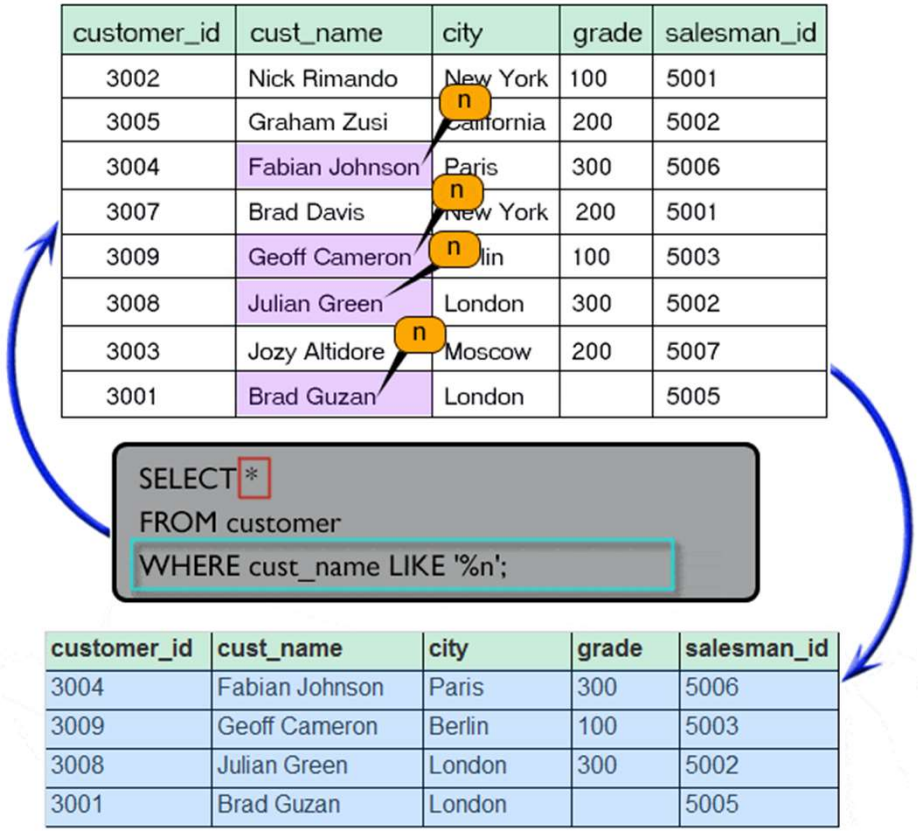
```
SELECT salesman_id, name, city, commission
FROM salesman
WHERE commission between 0.10 AND 0.12;
```



Query 14 (table: customer)

- Find all those customers with all information whose names are ending with the letter 'n'.

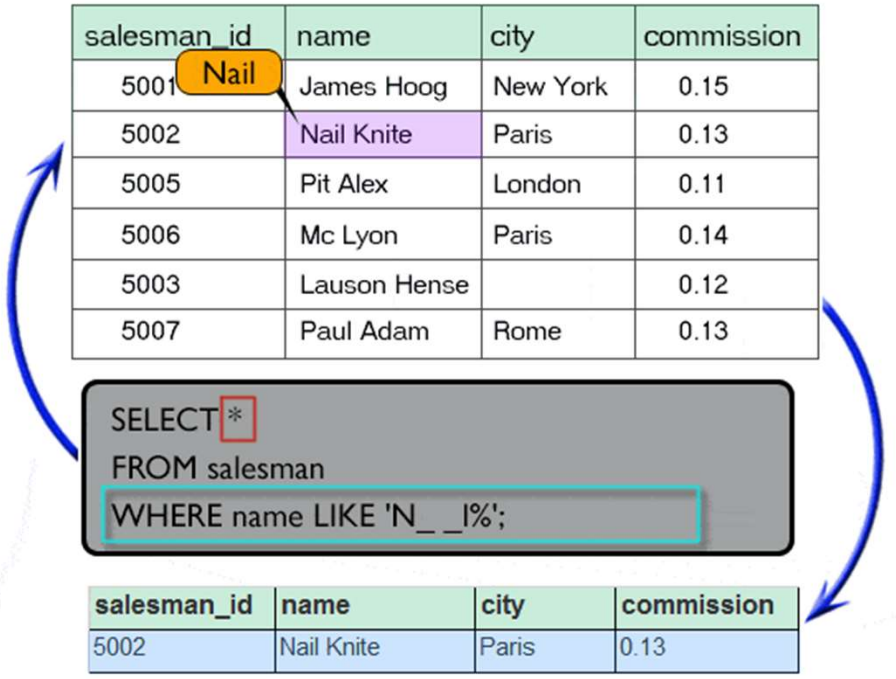
```
SELECT *
FROM customer
WHERE cust_name LIKE '%n';
```



Query 15 (table: salesmen)

- Find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.

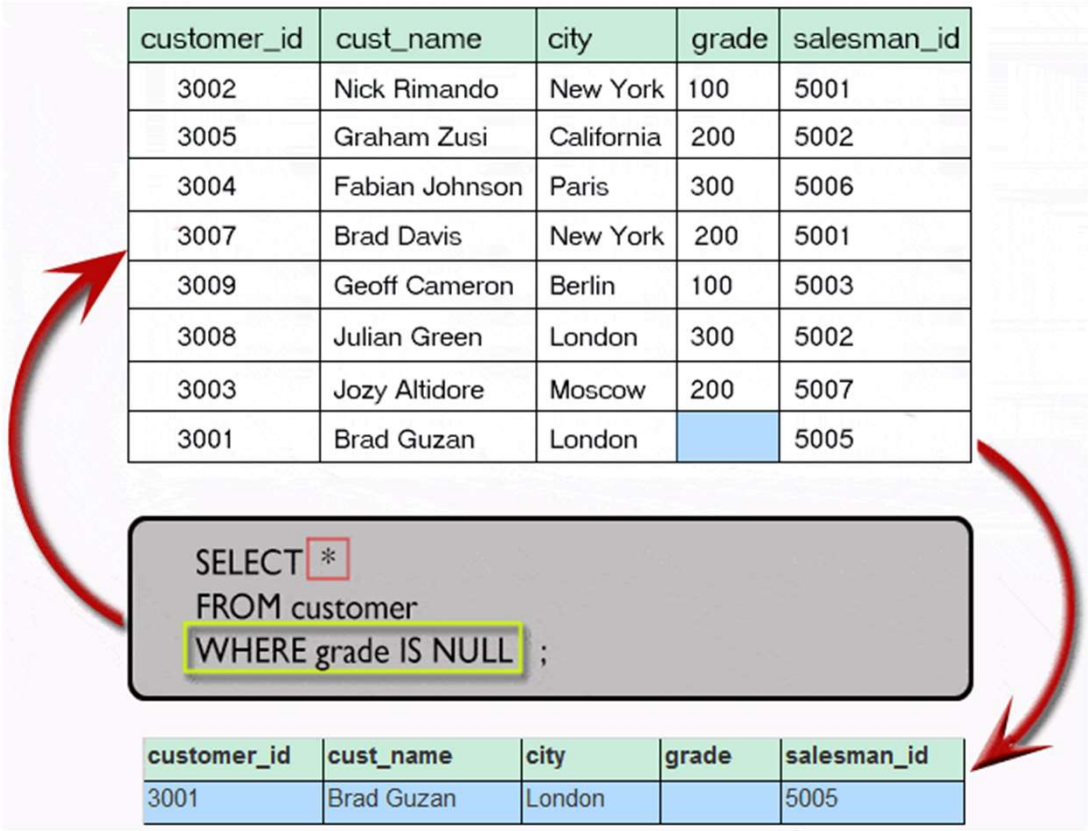
SELECT *
FROM salesman
WHERE name **LIKE** 'N__l%';



Query 16 (table: customer)

- Find that customer with all information who does not get any grade except NULL.

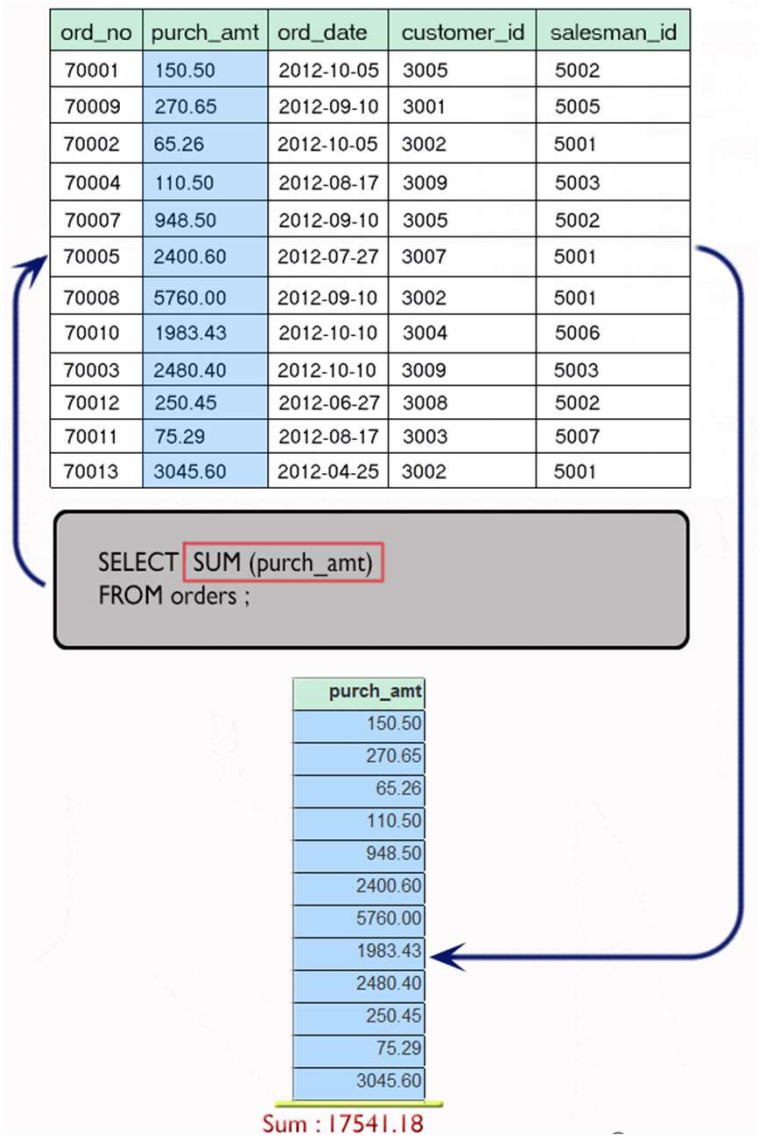
```
SELECT *
FROM customer
WHERE grade IS NULL;
```



Query 17 (table: orders)

- Find the total purchase amount of all orders.

SELECT SUM (purch_amt)
FROM orders;

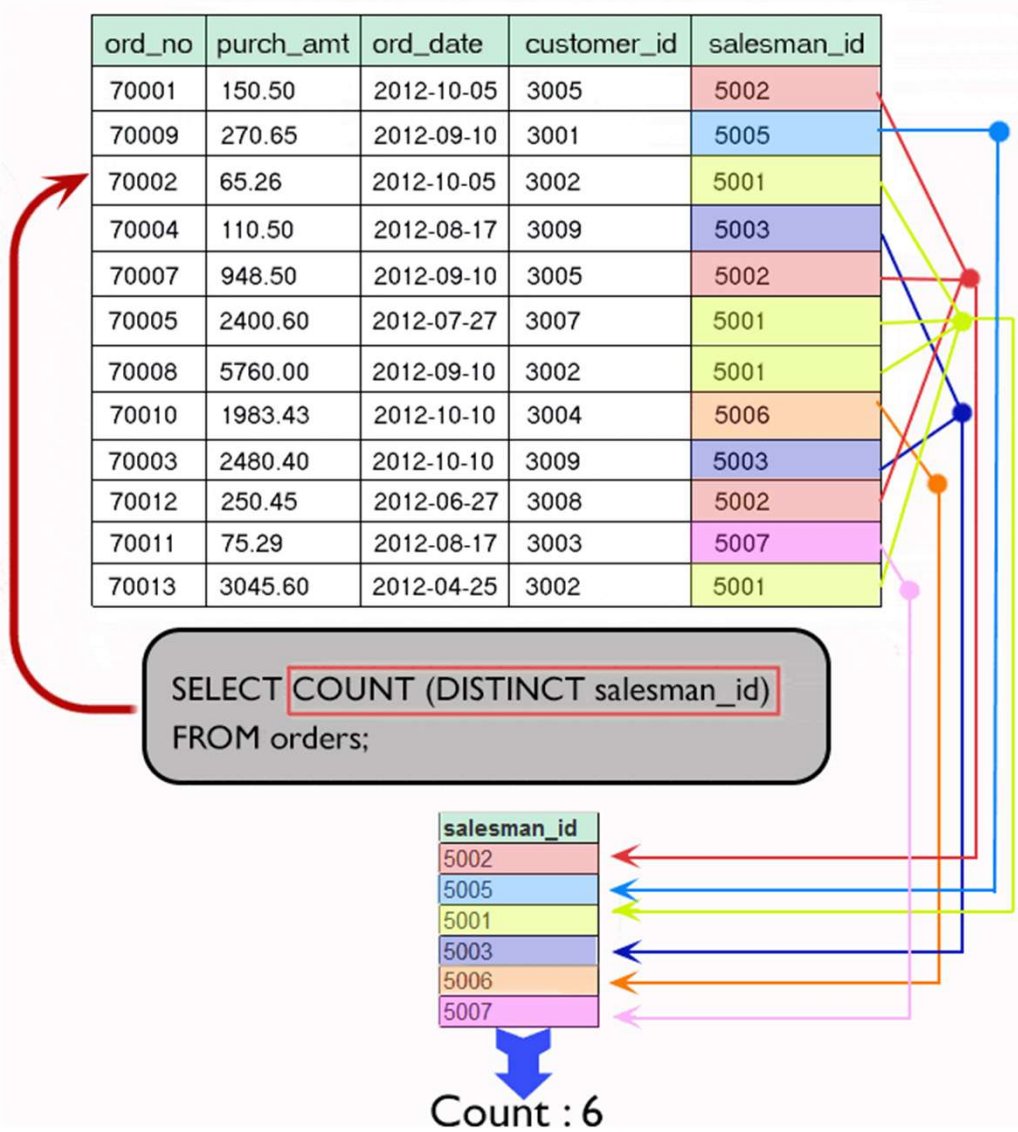


Query 18 (table: orders)

- Find the number of salesman currently listing for all of their customers.

SELECT COUNT (salesman_id)
FROM orders;

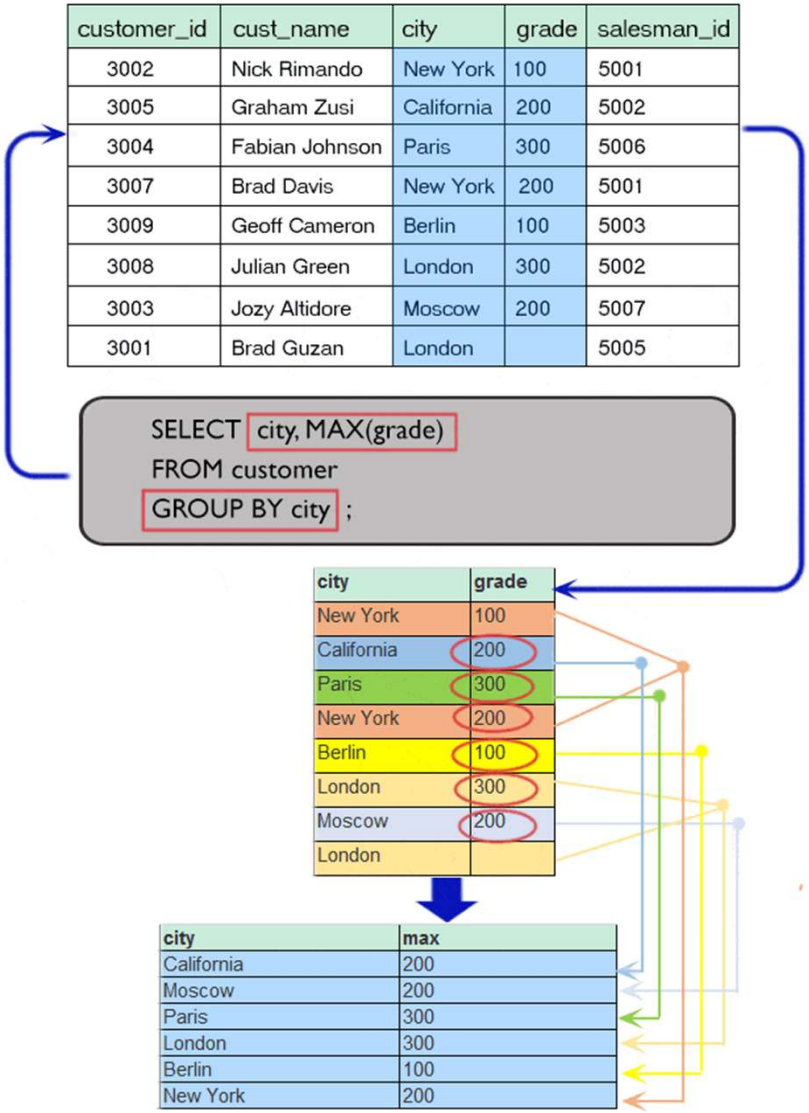
SELECT COUNT (**DISTINCT** salesman_id)
FROM orders;



Query 19 (table: customer)

- Find the highest grade for each of the cities of the customers.

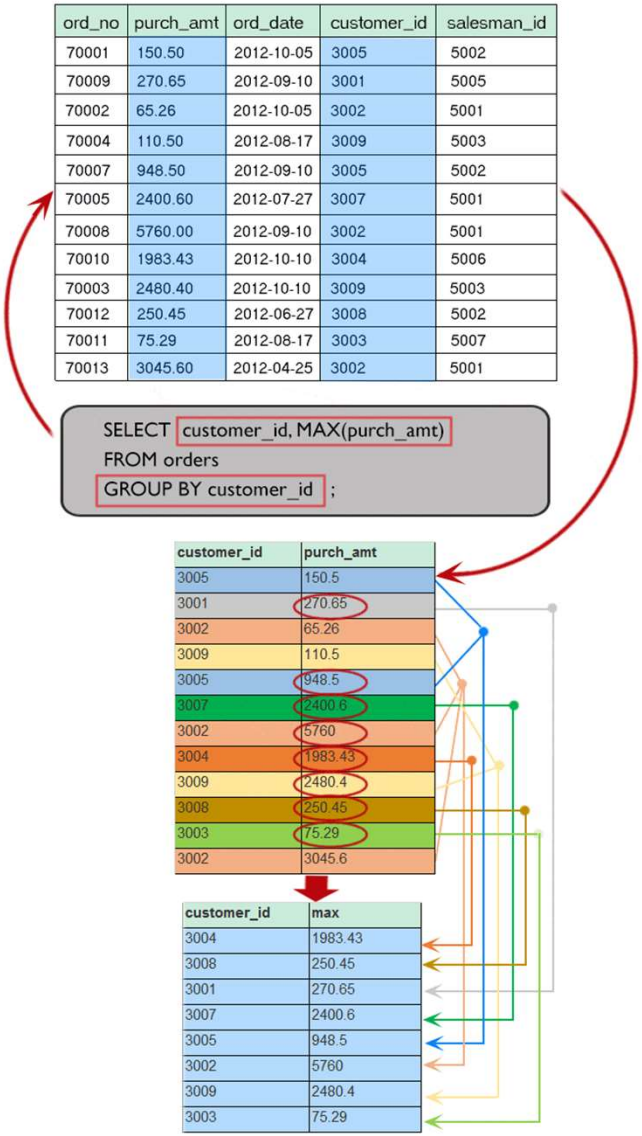
```
SELECT city, MAX(grade)
FROM customer
GROUP BY city;
```



Query 20 (table: orders)

- Find the highest purchase amount ordered by the each customer with their ID and highest purchase amount.

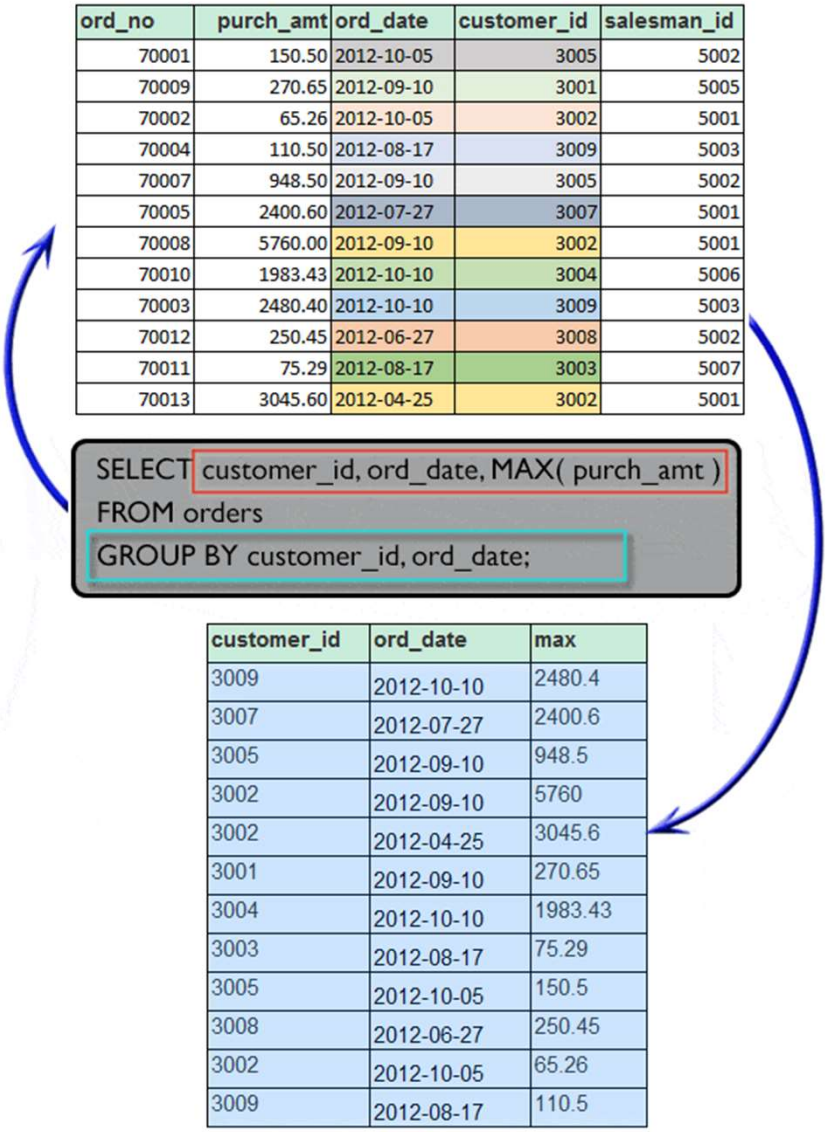
SELECT customer_id, **MAX**(purch_amt)
FROM orders
GROUP BY customer_id;



Query 21 (table: orders)

- Find the highest purchase amount ordered by the each customer on a particular date with their ID, order date and highest purchase amount.

SELECT customer_id, ord_date, **MAX**(purch_amt)
FROM orders
GROUP BY customer_id, ord_date;



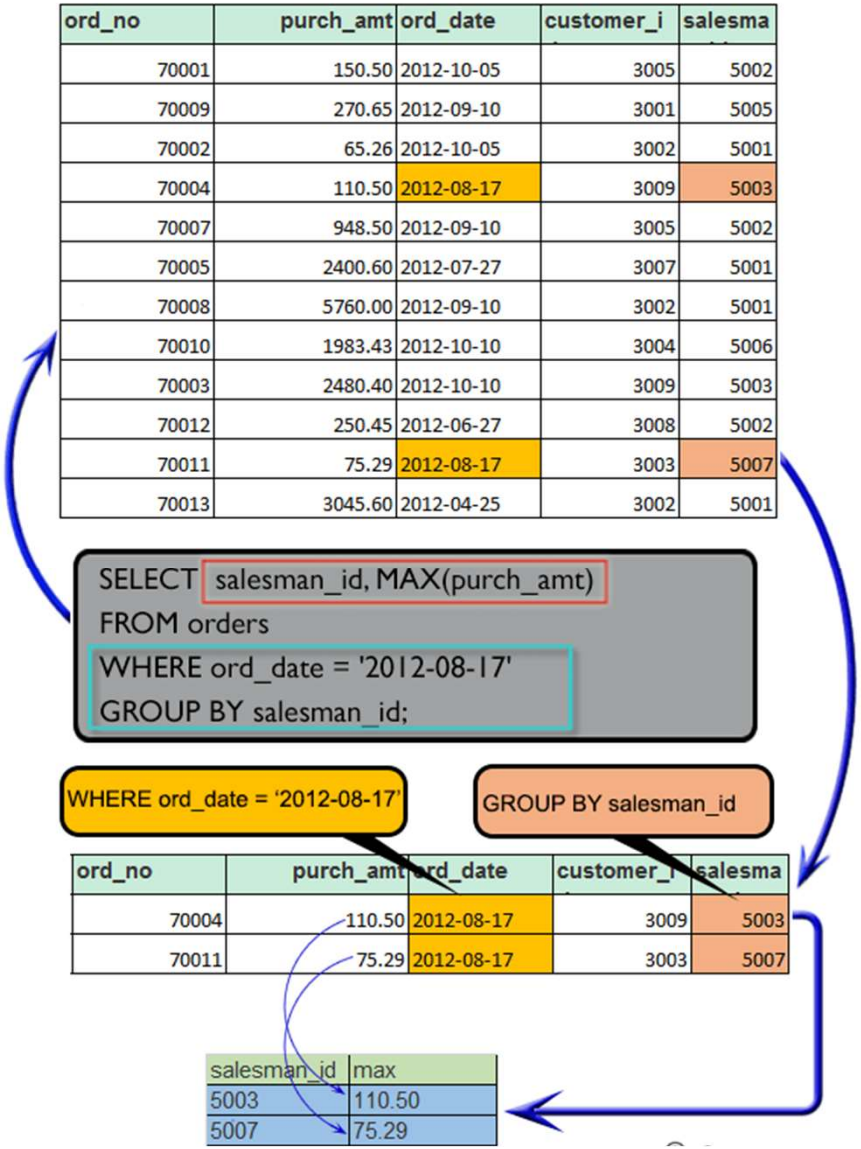
ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001

customer_id	ord_date	max
3009	2012-10-10	2480.4
3007	2012-07-27	2400.6
3005	2012-09-10	948.5
3002	2012-09-10	5760
3002	2012-04-25	3045.6
3001	2012-09-10	270.65
3004	2012-10-10	1983.43
3003	2012-08-17	75.29
3005	2012-10-05	150.5
3008	2012-06-27	250.45
3002	2012-10-05	65.26
3009	2012-08-17	110.5

Query 22 (table: orders)

- Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

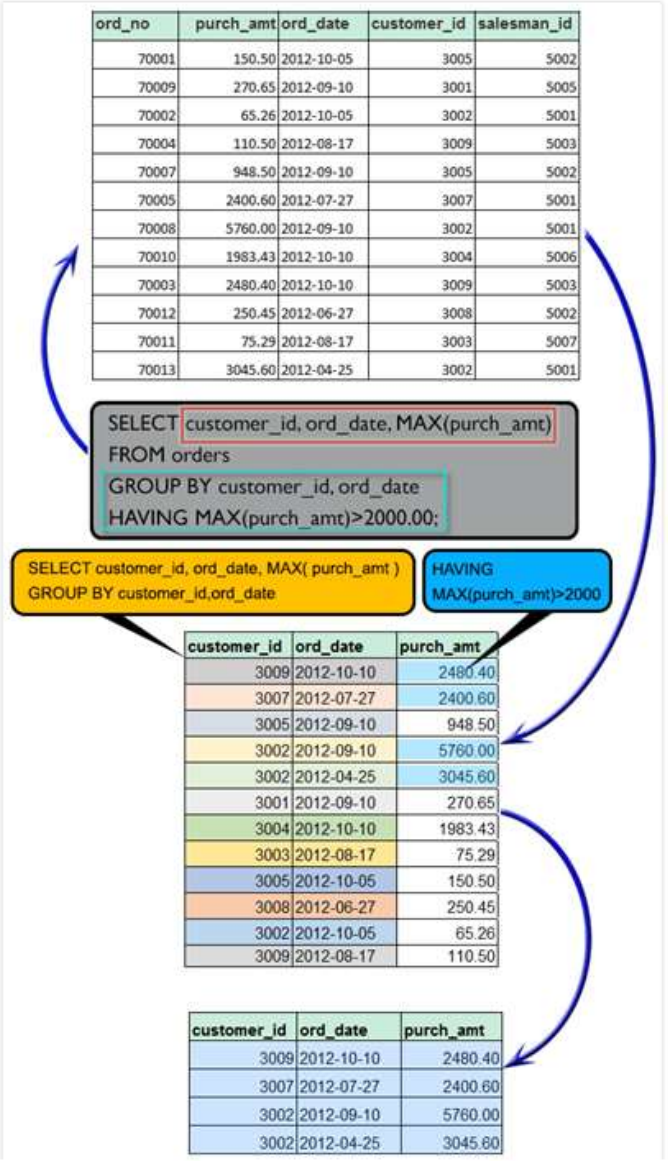
```
SELECT salesman_id, MAX(purch_amt)
FROM orders
WHERE ord_date = '2012-08-17'
GROUP BY salesman_id;
```



Query 23 (table: orders)

- Find the highest purchase amount with their customer ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.

SELECT customer_id, ord_date, **MAX**(purch_amt)
FROM orders
GROUP BY customer_id, ord_date
HAVING MAX(purch_amt) > 2000.00;



ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001

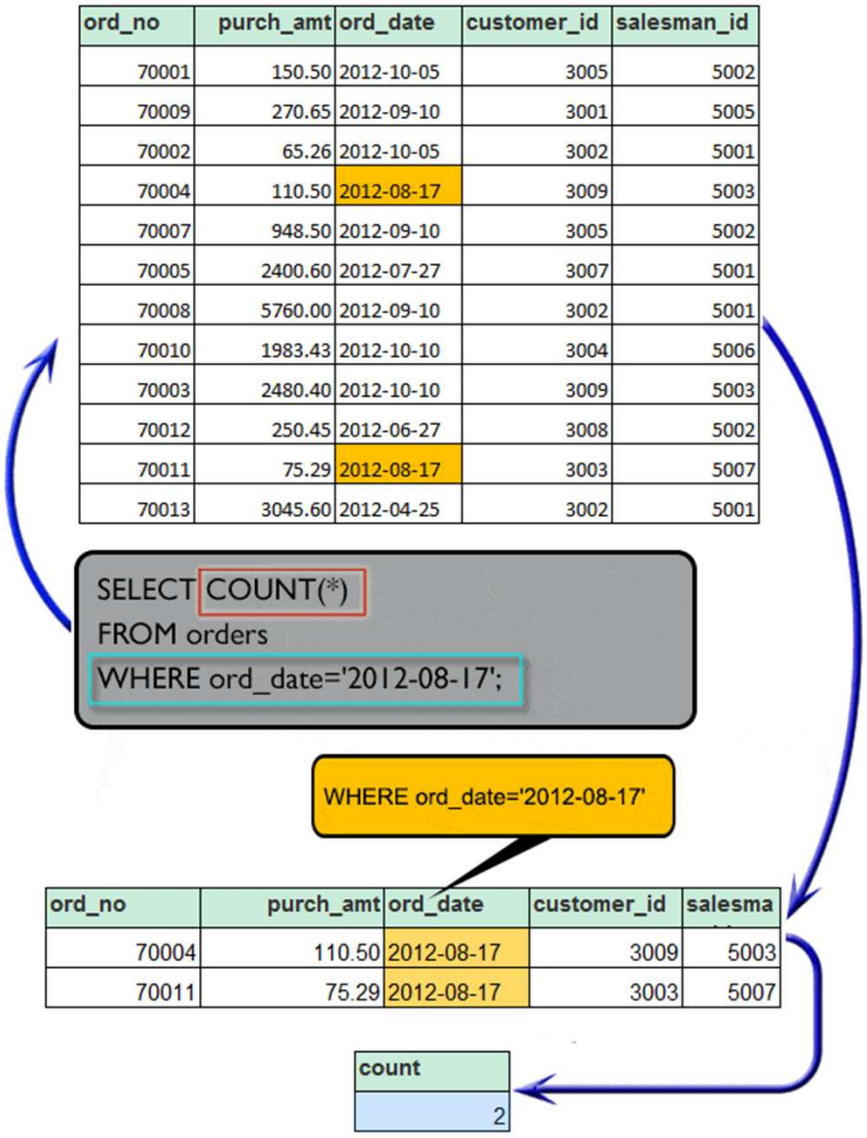
customer_id	ord_date	purch_amt
3009	2012-10-10	2480.40
3007	2012-07-27	2400.60
3005	2012-09-10	948.50
3002	2012-09-10	5760.00
3002	2012-04-25	3045.60
3001	2012-09-10	270.65
3004	2012-10-10	1983.43
3003	2012-08-17	75.29
3005	2012-10-05	150.50
3008	2012-06-27	250.45
3002	2012-10-05	65.26
3009	2012-08-17	110.50

customer_id	ord_date	purch_amt
3009	2012-10-10	2480.40
3007	2012-07-27	2400.60
3002	2012-09-10	5760.00
3002	2012-04-25	3045.60

Query 24 (table: orders)

- Write a SQL statement that counts all orders for a date August 17th, 2012.

```
SELECT COUNT(*)
FROM orders
WHERE ord_date = '2012-08-17';
```



ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.50	2012-08-17	3009	5003
70007	948.50	2012-09-10	3005	5002
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001

ord_no	purch_amt	ord_date	customer_id	salesma
70004	110.50	2012-08-17	3009	5003
70011	75.29	2012-08-17	3003	5007

count
2

SQL Practice 2
Multiple tables Joins Nested Queries

Link: https://www.w3resource.com/sql-exercises/

salesman

salesman_id	name	city	commission
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
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

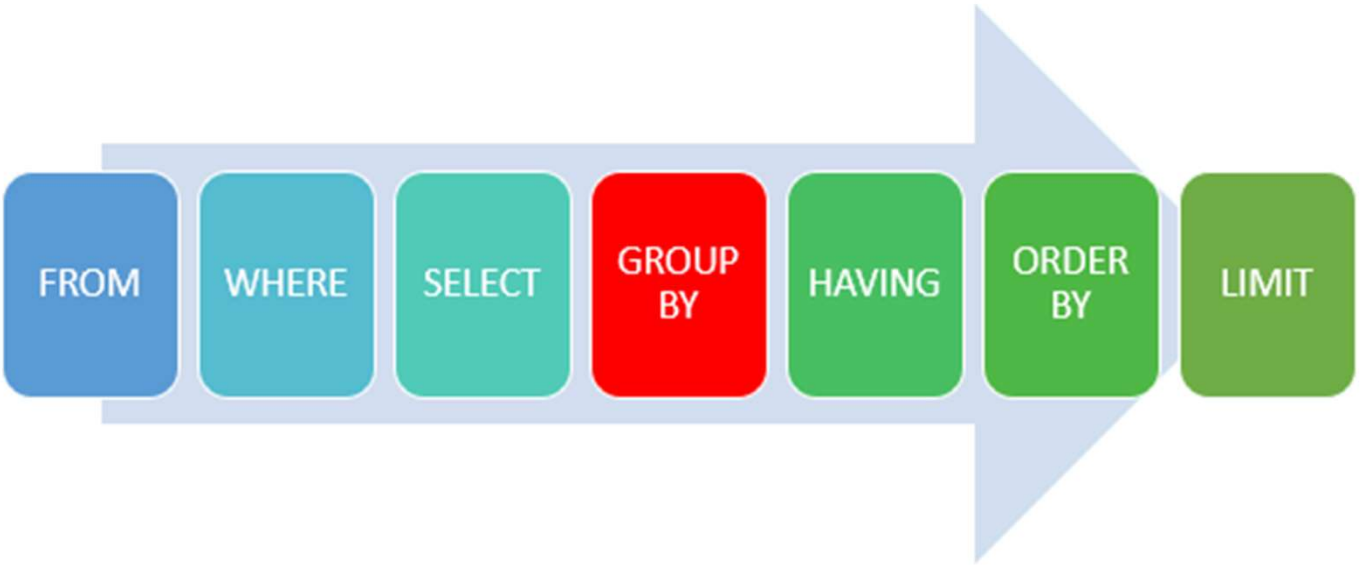
customer

customer_id	customer_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moncow	200	5007

order

order_no	*purch_amt*	*order_date*	*customer_id*	*salesman_id*
70001	*150.5*	*2016-10-05*	*3005*	*5002*
70009	*270.65*	*2016-09-10*	*3001*	
70002	*65.26*	*2016-10-05*	*3002*	*5001*
70004	*110.5*	*2016-08-17*	*3009*	
70007	*948.5*	*2016-09-10*	*3005*	*5002*
70005	*2400.6*	*2016-07-27*	*3007*	*5001*
70008	*5760*	*2016-09-10*	*3002*	*5001*
70010	*1983.43*	*2016-10-10*	*3004*	*5006*
70003	*2480.4*	*2016-10-10*	*3009*	
70012	*250.45*	*2016-06-27*	*3008*	*5002*
70011	*75.29*	*2016-08-17*	*3003*	*5007*

Order of SQL Statement

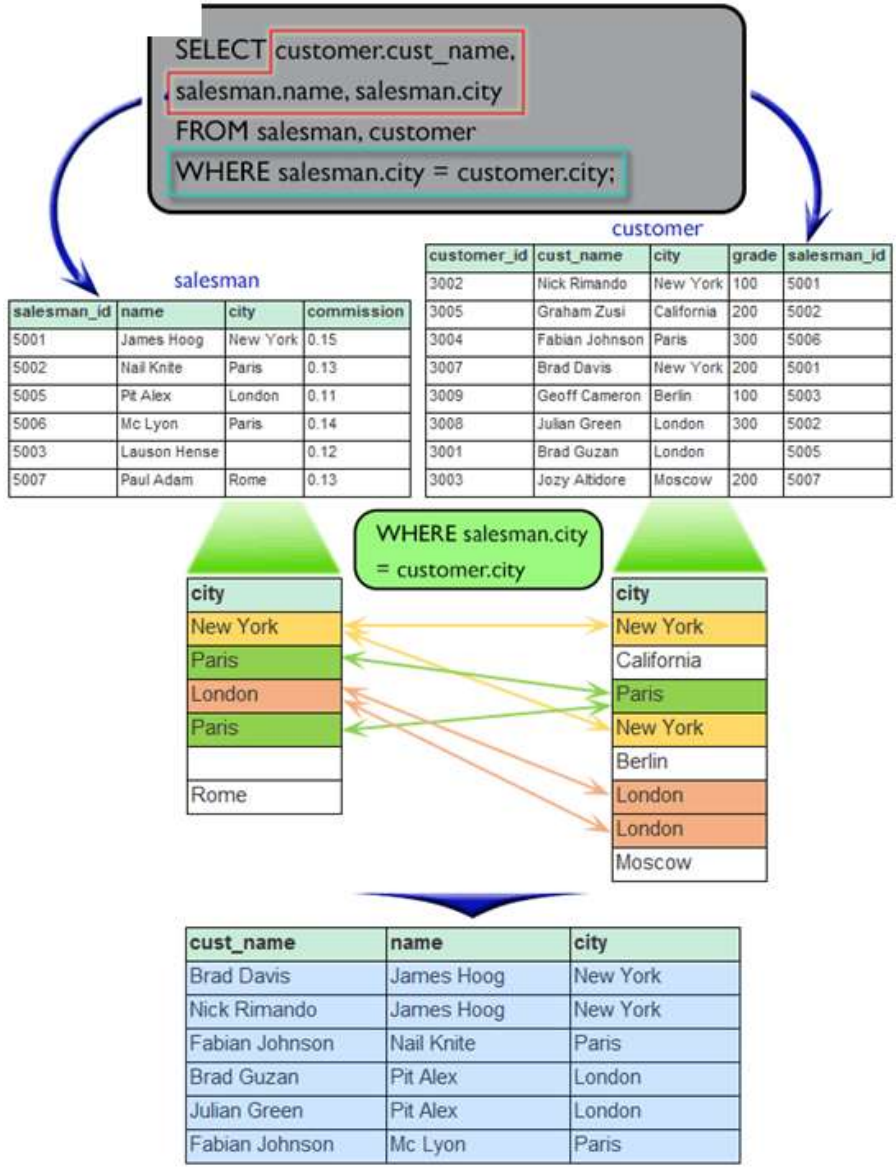


Query 1

- Find the name and city of those customers and salesmen who lives in the same city.

cust_name	name	city
Nick Rimando	James Hoog	New York
Brad Davis	James Hoog	New York
Julian Green	Pit Alex	London
Fabian Johnson	Mc Lyon	Paris
Fabian Johnson	Nail Knite	Paris
Brad Guzan	Pit Alex	London

SELECT C.cust_name S.name S.city
FROM salesman **AS** S customer **AS** C
WHERE S.city = C.city

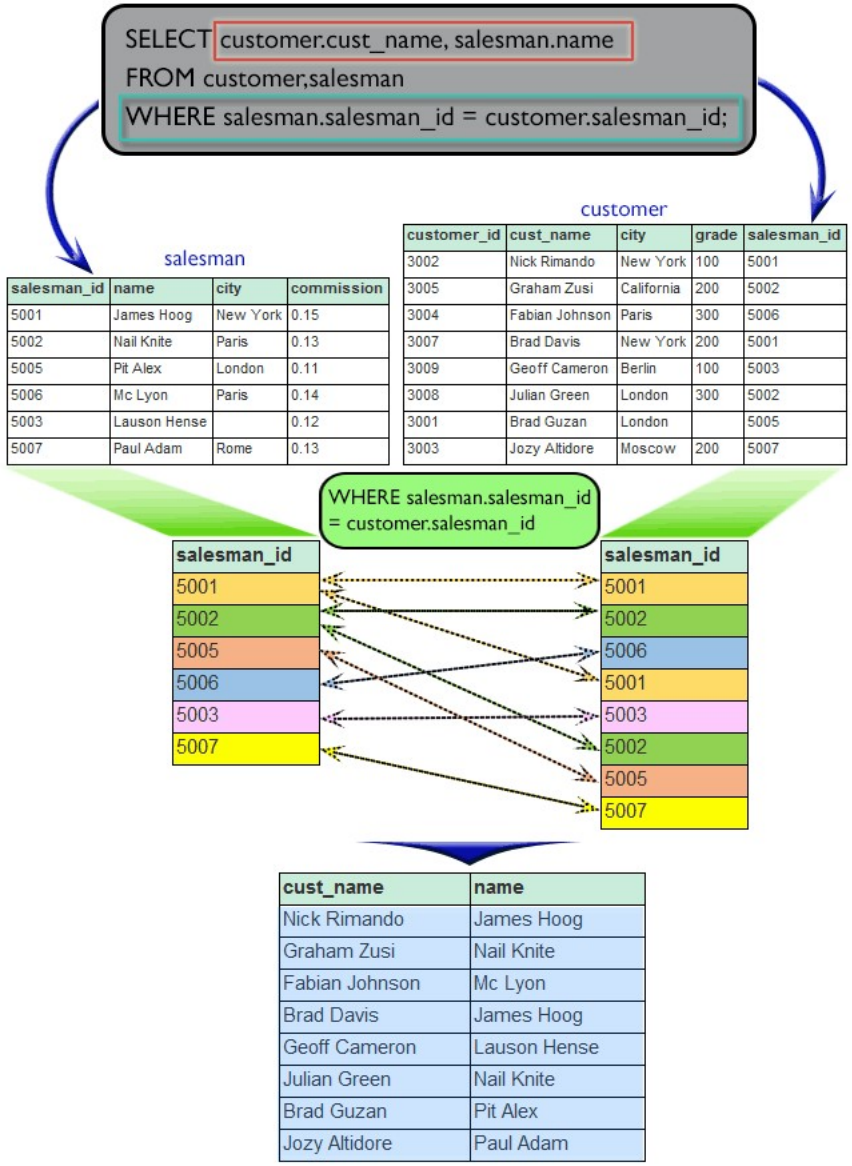


Query 2

- Find the names of all customers along with the salesmen who works for them.

cust_name	name
Nick Rimando	James Hoog
Brad Davis	James Hoog
Graham Zusi	Nail Knite
Julian Green	Nail Knite
Fabian Johnson	Mc Lyon
Geoff Cameron	Lauson Hen
Jozy Altidor	Paul Adam
Brad Guzan	Pit Alex

SELECT customer.cust_name salesman.name
FROM customer salesman
WHERE salesman.salesman_id = customer.salesman_id;



salesman_id	name	city	commission
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
5003	Lauson Hense		0.12
5007	Paul Adam	Rome	0.13

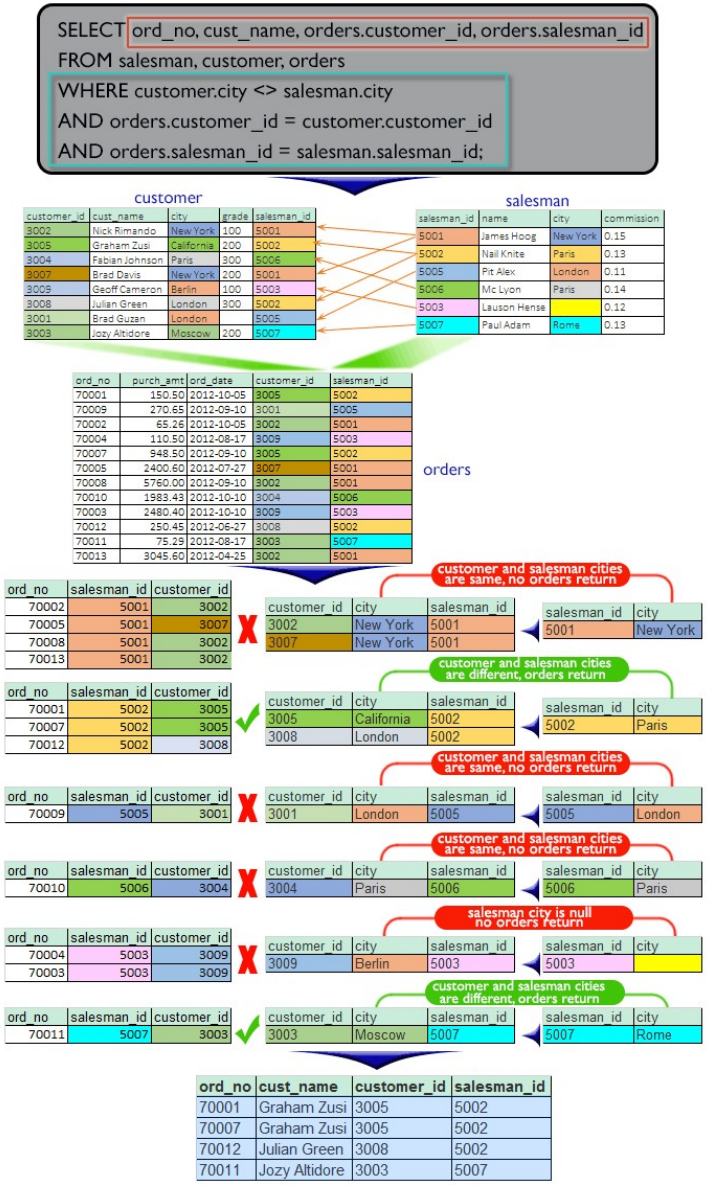
customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3004	Fabian Johnson	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Cameron	Berlin	100	5003
3008	Julian Green	London	300	5002
3001	Brad Guzan	London		5005
3003	Jozy Altidore	Moscow	200	5007

Query 3

- Display all those orders by the customers not located in the same cities where their salesmen live.

ord_no	cust_name	customer_id	salesman_id
70004	Geoff Cameron	3009	5003
70003	Geoff Cameron	3009	5003
70011	Jozy Altidor	3003	5007
70001	Graham Zusi	3005	5002
70007	Graham Zusi	3005	5002
70012	Julian Green	3008	5002

SELECT ord_no cust_name orders.customer_id orders.salesman_id
FROM salesman customer orders
WHERE customer.city **<>** salesman.city
AND orders.customer_id **=** customer.customer_id
AND orders.salesman_id **=** salesman.salesman_id;



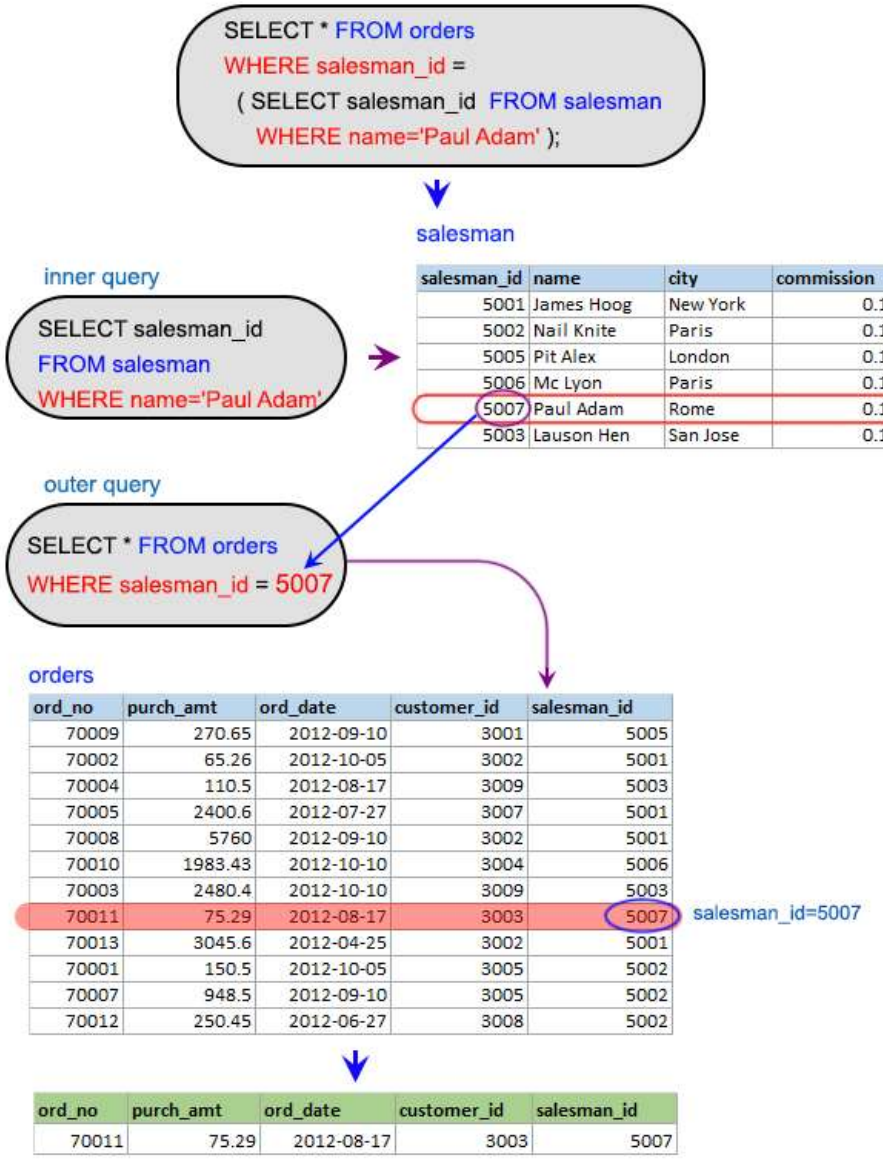
Query 4 (using subquery)

Display all the orders issued by the salesman 'Paul Adam' from the orders table.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70011	75.29	2012-08-17	3003	5007

```
SELECT *
FROM orders
WHERE salesman_id =
   (SELECT salesman_id
    FROM salesman
    WHERE name = 'Paul Adam');
```

- Can we make this query unnested? If yes how?



salesman_id	name	city	commission
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

ord_no	purch_amt	ord_date	customer_id	salesman_id
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
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
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001
70001	150.5	2012-10-05	3005	5002
70007	948.5	2012-09-10	3005	5002
70012	250.45	2012-06-27	3008	5002

ord_no	purch_amt	ord_date	customer_id	salesman_id
70011	75.29	2012-08-17	3003	5007

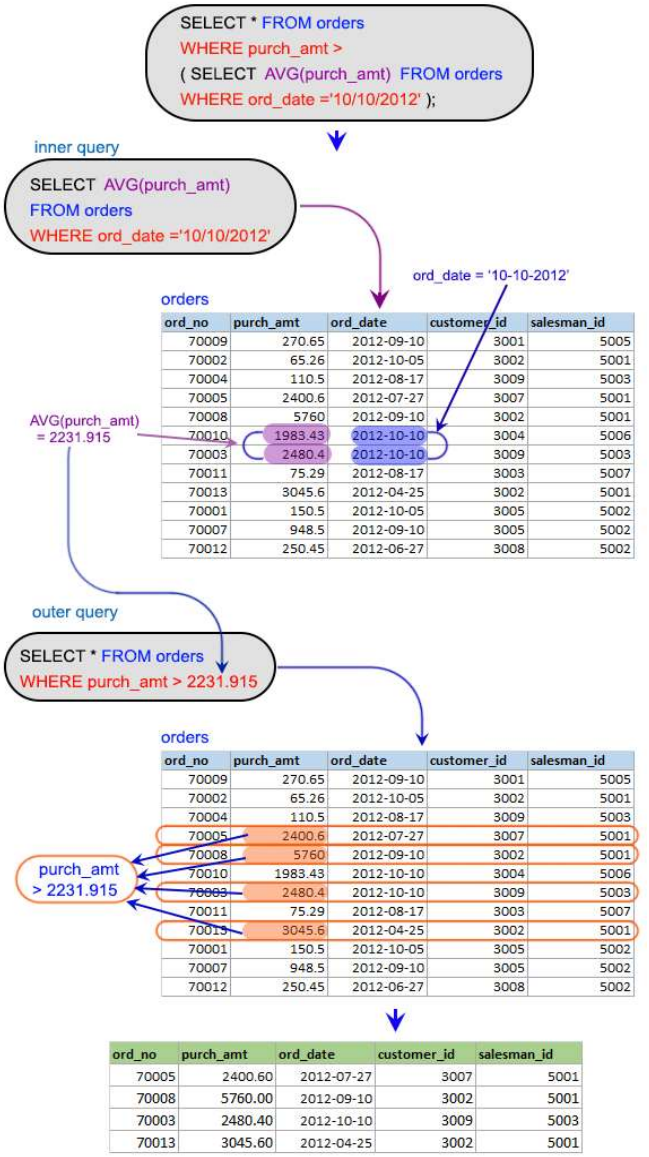
Query 5 (using subquery)

Display all the orders which values are greater than the average order value for 10th October 2012.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70003	2480.40	2012-10-10	3009	5003
70013	3045.60	2012-04-25	3002	5001

```
SELECT *
FROM orders
WHERE purch_amt >
   (SELECT  AVG(purch_amt)
    FROM orders
    WHERE ord_date = '2012-10-10');
```

- Can we make this query unnested? If yes how?



Query 6 (using subquery)

Find all orders attributed to salesmen in Paris.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70007	948.50	2012-09-10	3005	5002
70012	250.45	2012-06-27	3008	5002
70010	1983.43	2012-10-10	3004	5006

```
SELECT *
FROM orders
WHERE salesman_id IN
   (SELECT salesman_id
    FROM salesman
    WHERE city ='Paris');
```

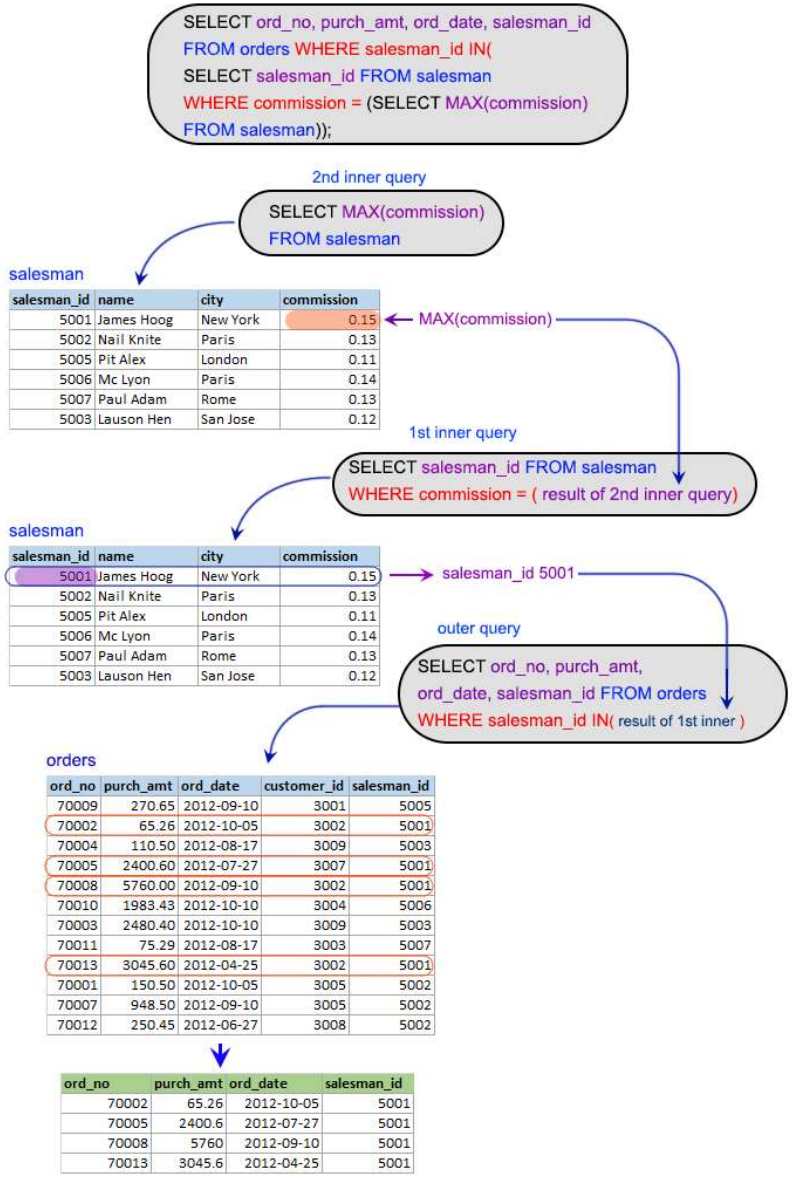
- Can we make this query unnested? If yes how?

Query 7 (using subquery)

Extract the data from the orders table for the salesman who earned the maximum commission.

ord_no	purch_amt	ord_date	salesman_id
70002	65.26	2012-10-05	5001
70005	2400.60	2012-07-27	5001
70008	5760.00	2012-09-10	5001
70013	3045.60	2012-04-25	5001

```
SELECT ord_no, purch_amt, ord_date, salesman_id
FROM orders
WHERE salesman_id IN (
        SELECT salesman_id
        FROM salesman
        WHERE commission = (
                SELECT MAX(commission)
                FROM salesman)
        );
```



Query 8 (using subquery)

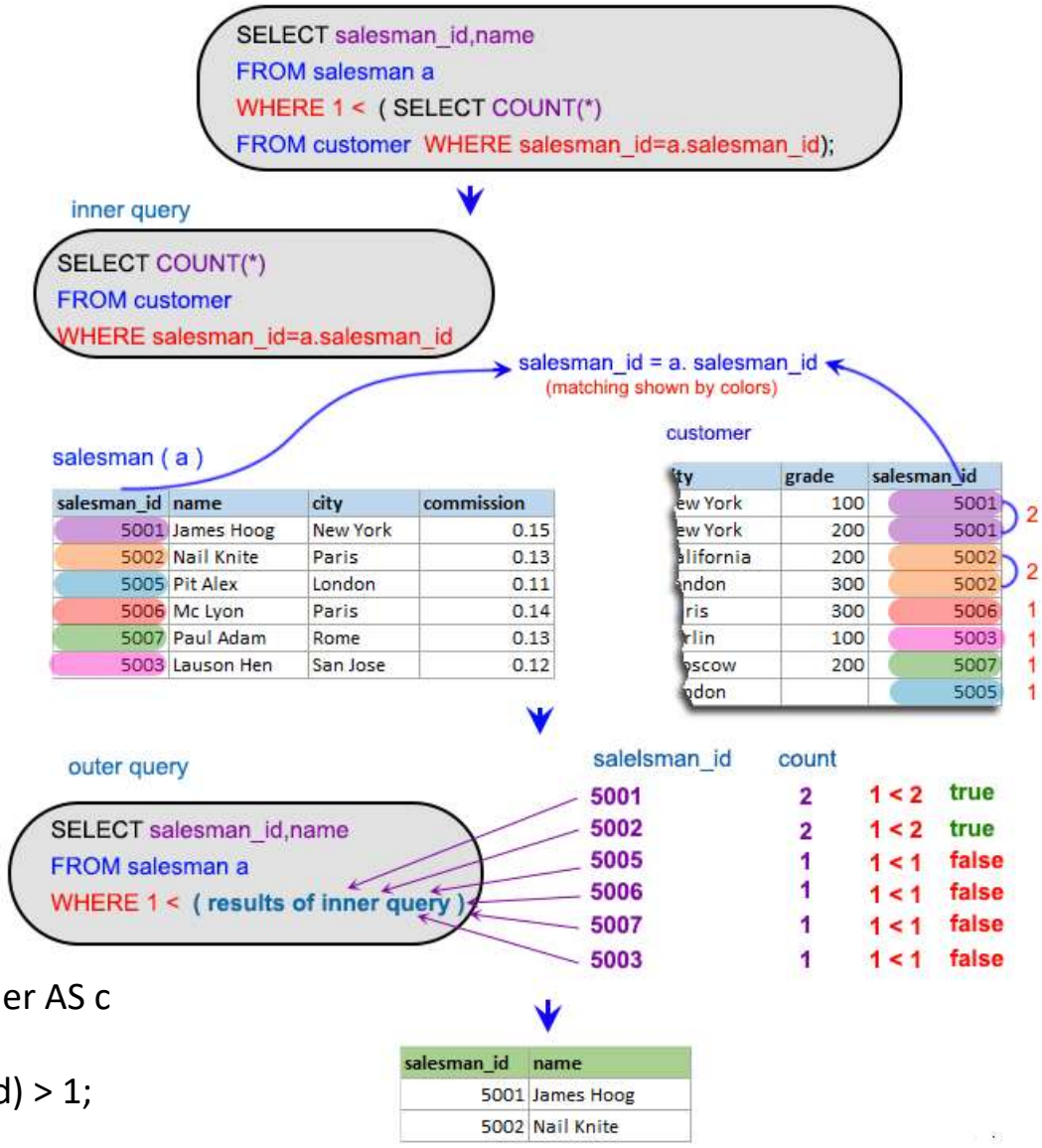
Find the name and ids of all salesmen who had more than one customer.

salesman_id	name
5001	James Hoog
5002	Nail Knite

```
SELECT salesman_id, name
FROM salesman AS a
WHERE 1 <
   (SELECT COUNT(*)
    FROM customer AS c
    WHERE c.salesman_id = a.salesman_id);
```

- Can we make this query unnested? If yes how?

```
SELECT c.salesman_id, s.name  FROM salesman AS s, customer AS c
where s.salesman_id = c.salesman_id
group by c.salesman_id, s.name  Having count(c.salesman_id) > 1;
```



salesman_id	name	city	commission
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

ty	grade	salesman_id	
ew York	100	5001	2
ew York	200	5001	
alifornia	200	5002	2
ndon	300	5002	
ris	300	5006	1
rlin	100	5003	1
scow	200	5007	1
ndon		5005	1

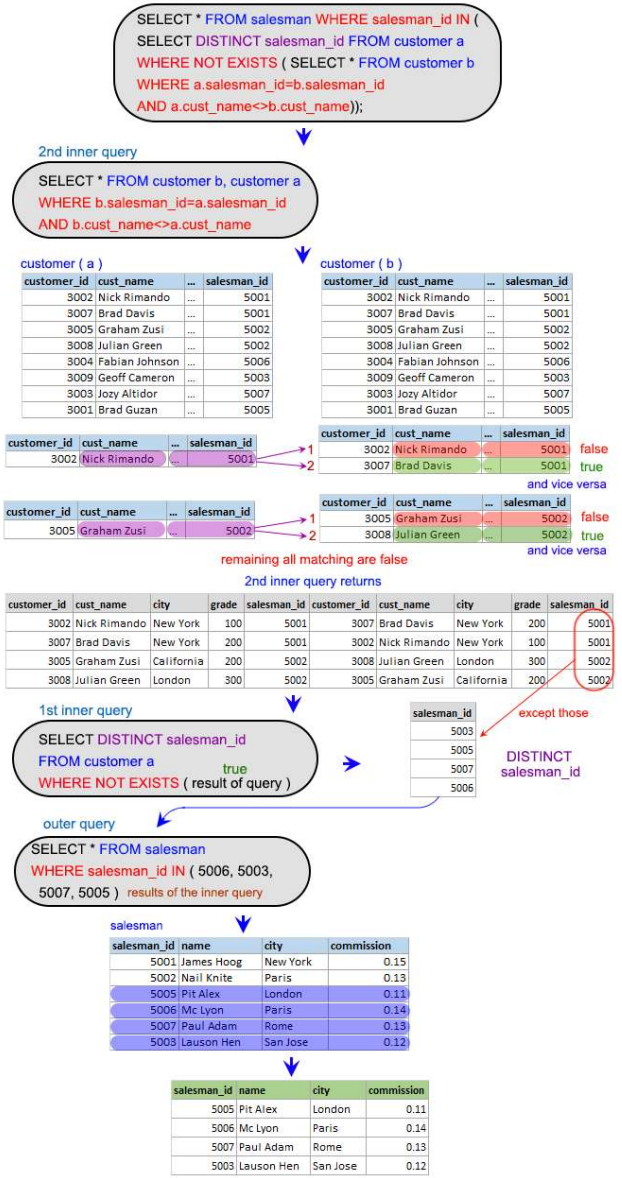
salesman_id	name
5001	James Hoog
5002	Nail Knite

Query 9 (using subquery)

Write a query to find all the salesmen who worked for only one customer.

salesman_id	name	city	commission
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

```
SELECT *
FROM salesman
WHERE salesman_id IN (
  SELECT DISTINCT salesman_id
  FROM customer a
  WHERE NOT EXISTS (
    SELECT * FROM customer b
    WHERE a.salesman_id = b.salesman_id
    AND a.cust_name <> b.cust_name));
```



Query 9: Equivalent Queries

Write a query to find all the salesmen who worked for only one customer.

salesman_id	name	city	commission
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

```
SELECT c.salesman_id, s.name, s.city, s.commission
FROM salesman s, customer c
where s.salesman_id = c.salesman_id
group by c.salesman_id, s.name
Having count(c.salesman_id) = 1;
```

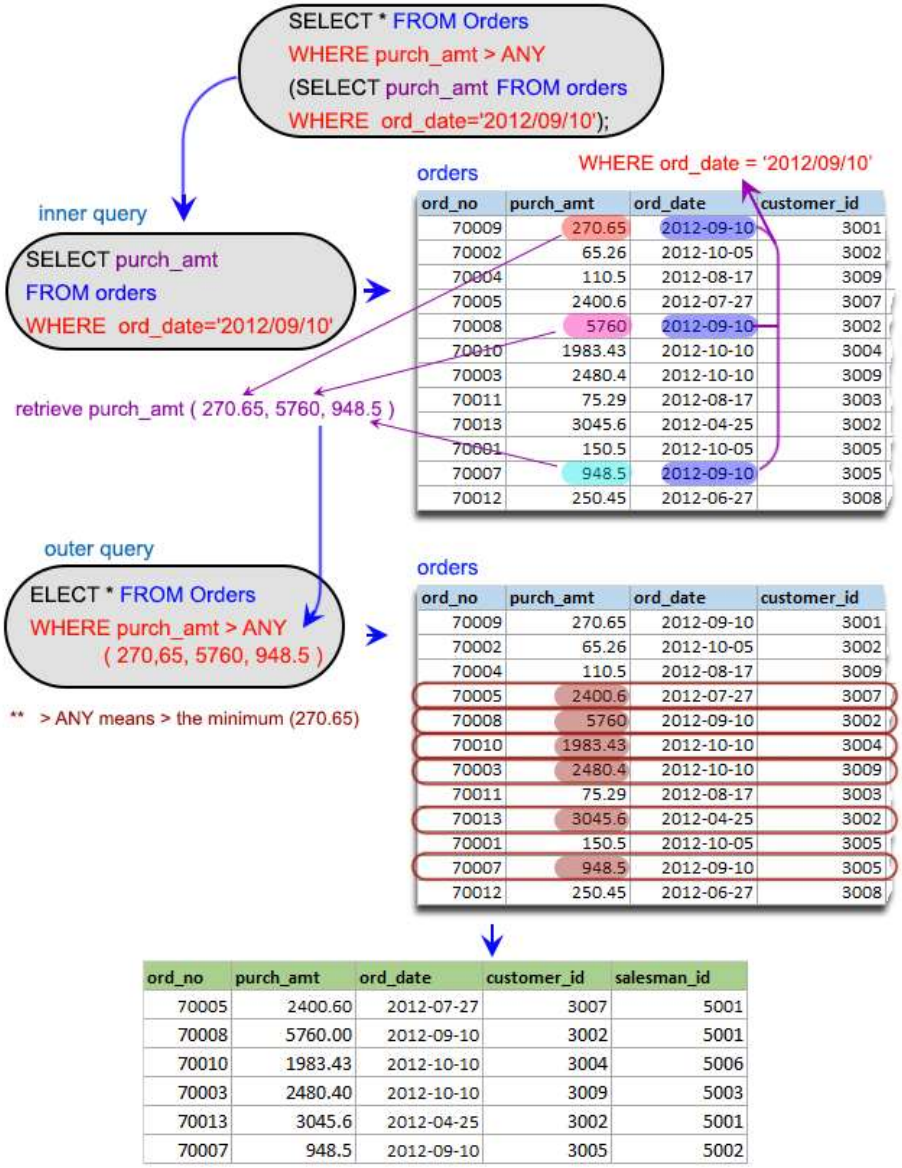
```
SELECT *
FROM salesman
WHERE salesman_id NOT IN (
                    SELECT a.salesman_id
                    FROM customer a, customer b
                    WHERE a.salesman_id = b.salesman_id
                    AND a.cust_name <> b.cust_name);
```

Query 10 (using subquery)

Display all the orders that had amounts that were greater than **at least one of the orders** from September 10th 2012.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70013	3045.60	2012-04-25	3002	5001
70007	948.50	2012-09-10	3005	5002

```
SELECT *
FROM Orders
WHERE purch_amt > ANY
                  (SELECT purch_amt
                   FROM orders
                   WHERE  ord_date = '2012-09-10');
```



ord_no	purch_amt	ord_date	customer_id
70009	270.65	2012-09-10	3001
70002	65.26	2012-10-05	3002
70004	110.5	2012-08-17	3009
70005	2400.6	2012-07-27	3007
70008	5760	2012-09-10	3002
70010	1983.43	2012-10-10	3004
70003	2480.4	2012-10-10	3009
70011	75.29	2012-08-17	3003
70013	3045.6	2012-04-25	3002
70001	150.5	2012-10-05	3005
70007	948.5	2012-09-10	3005
70012	250.45	2012-06-27	3008

ord_no	purch_amt	ord_date	customer_id
70009	270.65	2012-09-10	3001
70002	65.26	2012-10-05	3002
70004	110.5	2012-08-17	3009
70005	2400.6	2012-07-27	3007
70008	5760	2012-09-10	3002
70010	1983.43	2012-10-10	3004
70003	2480.4	2012-10-10	3009
70011	75.29	2012-08-17	3003
70013	3045.6	2012-04-25	3002
70001	150.5	2012-10-05	3005
70007	948.5	2012-09-10	3005
70012	250.45	2012-06-27	3008

ord_no	purch_amt	ord_date	customer_id	salesman_id
70005	2400.60	2012-07-27	3007	5001
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.40	2012-10-10	3009	5003
70013	3045.6	2012-04-25	3002	5001
70007	948.5	2012-09-10	3005	5002

Query 11 (using subquery)

display only those customers whose grade are, in fact, higher than every customer in New York.

customer_id	cust_name	city	grade	salesman_id
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006

```
SELECT *
FROM customer
WHERE grade > ALL
                (SELECT grade
                 FROM customer
                 WHERE city = 'NewYork');
```