

# SQL Practice 2

## Multiple tables Joins Nested Queries

salesman

<u>salesman_id</u>	<u>name</u>	<u>city</u>	<u>commission</u>
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

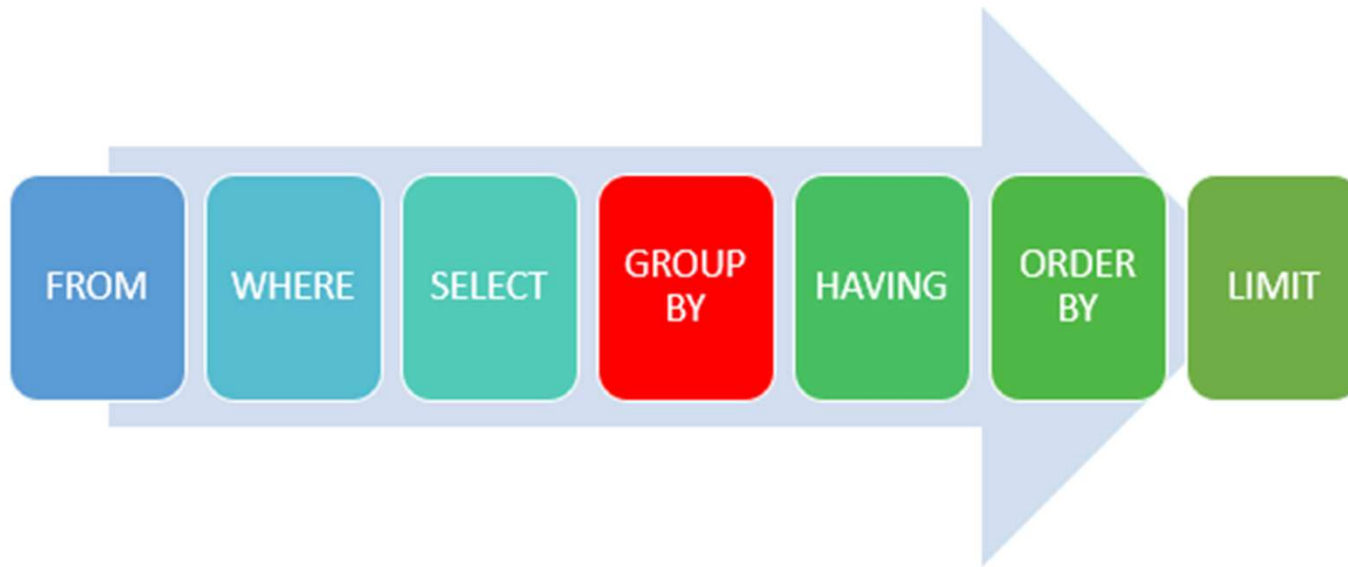
customer

<u>customer_id</u>	<u>customer_name</u>	<u>city</u>	<u>grade</u>	<u>salesman_id</u>
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

<u>order no</u>	<u>purch amt</u>	<u>order date</u>	<u>customer id</u>	<u>salesman id</u>
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



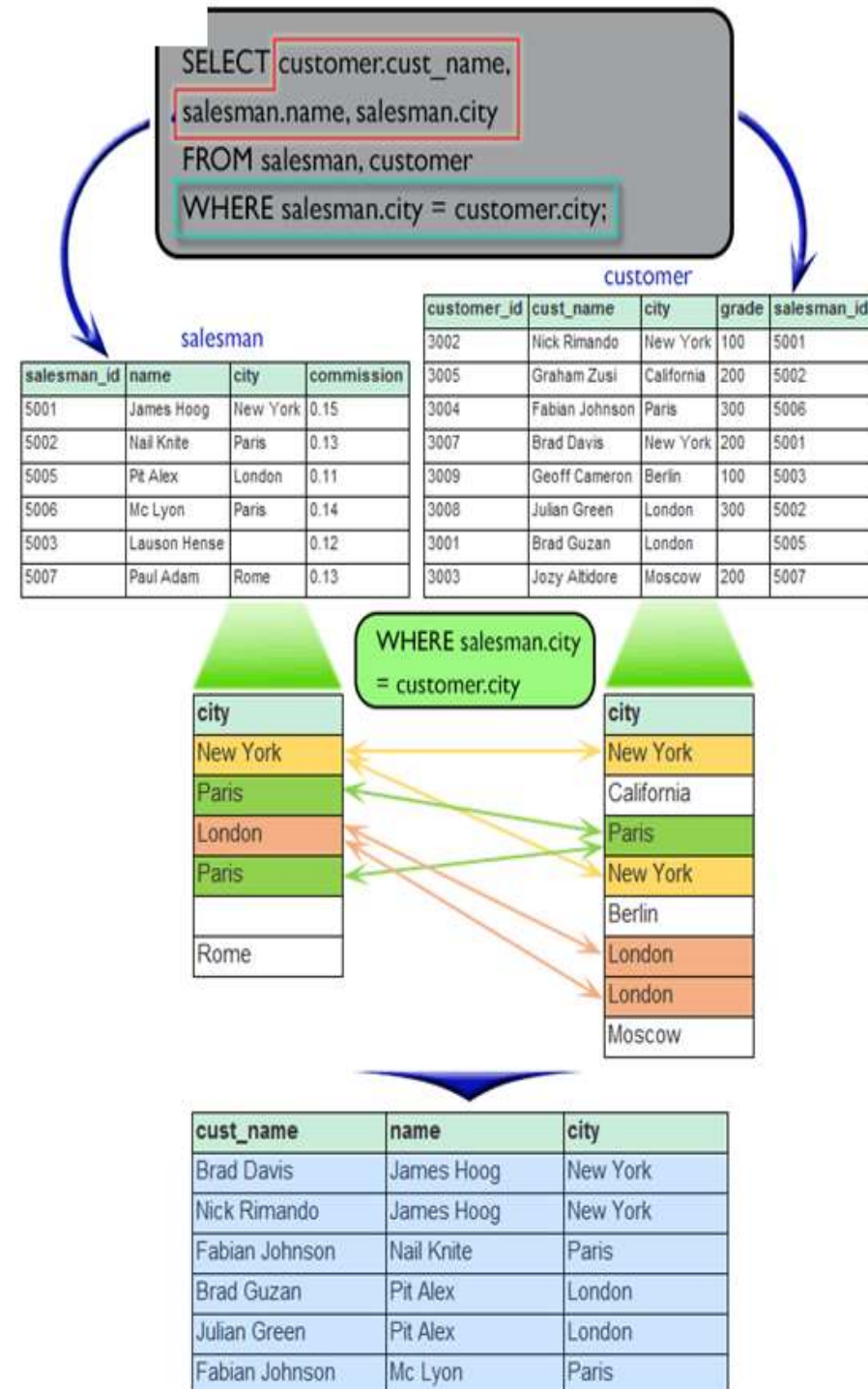
# Query 1

- 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

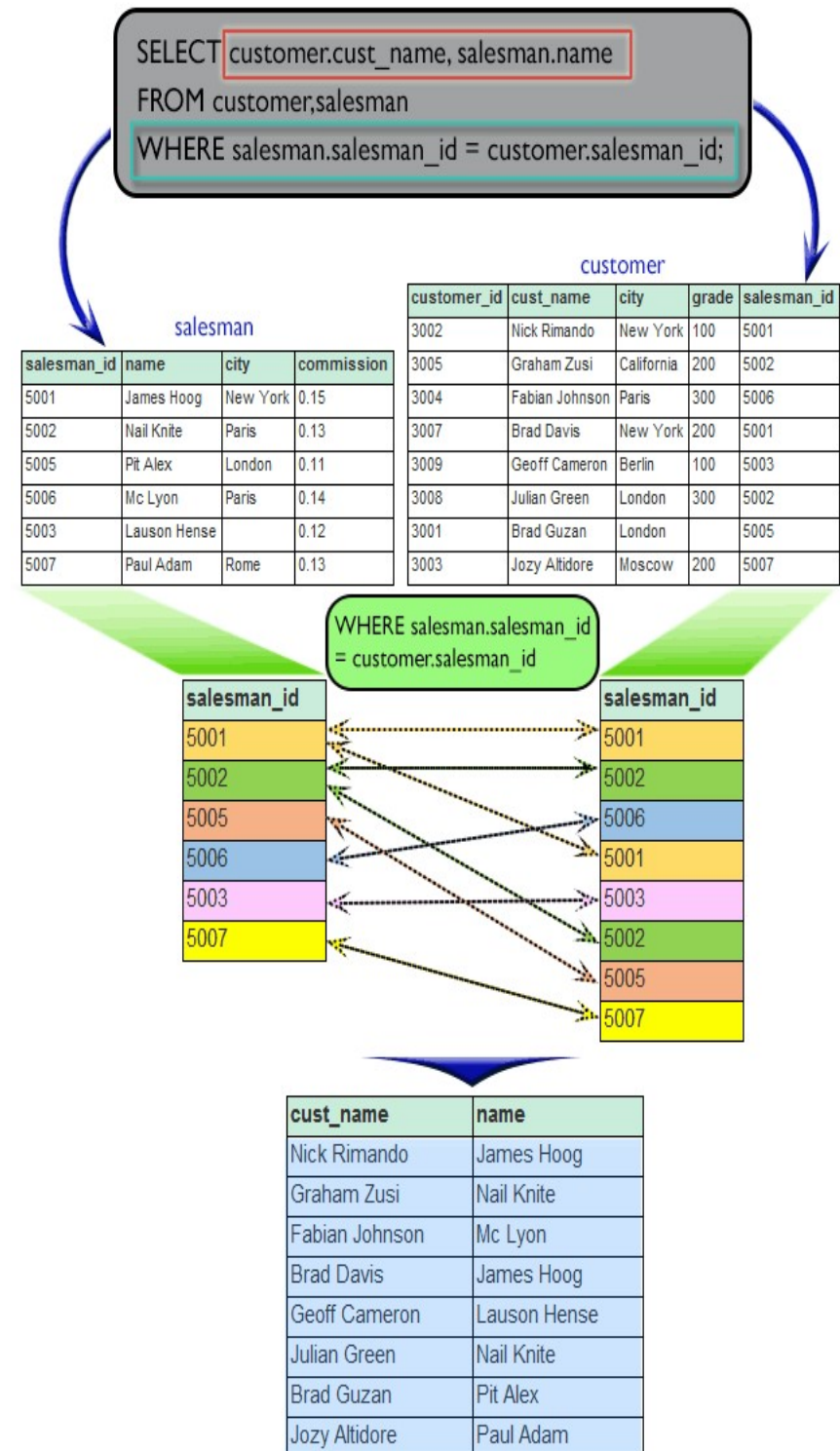
- Find the names of all customers along with the salesmen who works for them.

<u>cust_name</u>	<u>name</u>
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;

```





# Query 3

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

```
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;
```

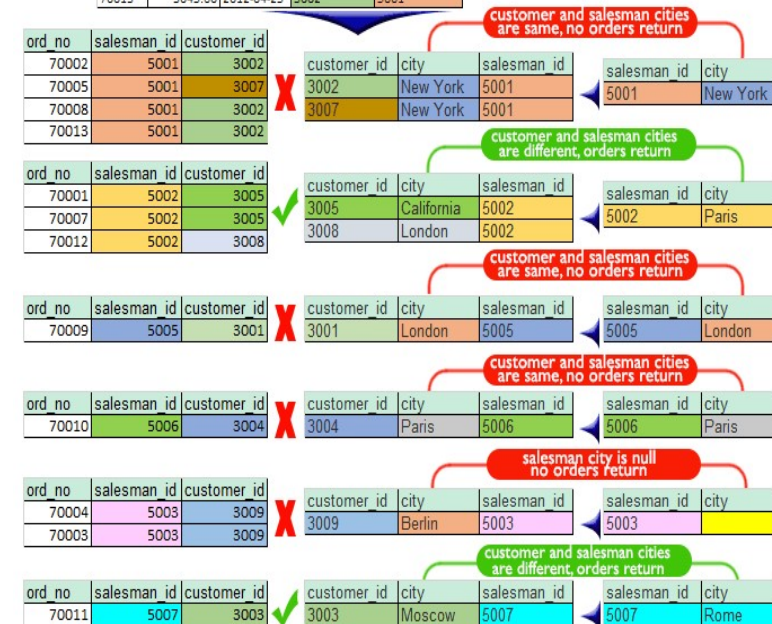
customer					salesman			
customer_id	cust_name	city	grade	salesman_id	salesman_id	name	city	commission
3002	Nick Rimando	New York	100	5001	5001	James Hoog	New York	0.15
3005	Graham Zusi	California	200	5002	5002	Nail Krite	Paris	0.13
3004	Fabian Johnson	Paris	300	5006	5005	Pit Alex	London	0.11
3007	Brad Davis	New York	200	5001	5006	Mc Lyon	Paris	0.14
3009	Geoff Cameron	Berlin	100	5003	5003	Lauson Hense	Yellow	0.12
3008	Julian Green	London	300	5002	5007	Paul Adam	Rome	0.13
3001	Brad Guzan	London	500	5005				
3003	Jozy Altidore	Moscow	200	5007				

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

orders

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;
```



ord_no	cust_name	customer_id	salesman_id
70001	Graham Zusi	3005	5002
70007	Graham Zusi	3005	5002
70012	Julian Green	3008	5002
70011	Jozy Altidore	3003	5007

## 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');
```

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

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
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

inner query

```
SELECT salesman_id  
FROM salesman  
WHERE name='Paul Adam'
```

outer query

```
SELECT * FROM orders  
WHERE salesman_id = 5007
```

orders

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

salesman\_id=5007

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');
```

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

inner query

```
SELECT AVG(purch_amt)  
FROM orders  
WHERE ord_date = '10/10/2012'
```

ord\_date = '10-10-2012'

orders

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

AVG(purch\_amt)  
= 2231.915

outer query

```
SELECT * FROM orders  
WHERE purch_amt > 2231.915
```

orders

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

purch\_amt  
> 2231.915

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



## 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');
```

# 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
    )
);
    
```

```

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));
    
```

2nd inner query

```

SELECT MAX(commission)
FROM salesman
    
```

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
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

← MAX(commission)

1st inner query

```

SELECT salesman_id FROM salesman
WHERE commission = ( result of 2nd inner query )
    
```

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
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

→ salesman\_id 5001

outer query

```

SELECT ord_no, purch_amt,
ord_date, salesman_id FROM orders
WHERE salesman_id IN( result of 1st inner )
    
```

orders

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	110.50	2012-08-17	3009	5003
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
70011	75.29	2012-08-17	3003	5007
70013	3045.60	2012-04-25	3002	5001
70001	150.50	2012-10-05	3005	5002
70007	948.50	2012-09-10	3005	5002
70012	250.45	2012-06-27	3008	5002

ord_no	purch_amt	ord_date	salesman_id
70002	65.26	2012-10-05	5001
70005	2400.6	2012-07-27	5001
70008	5760	2012-09-10	5001
70013	3045.6	2012-04-25	5001

# Query 8 (using subquery)

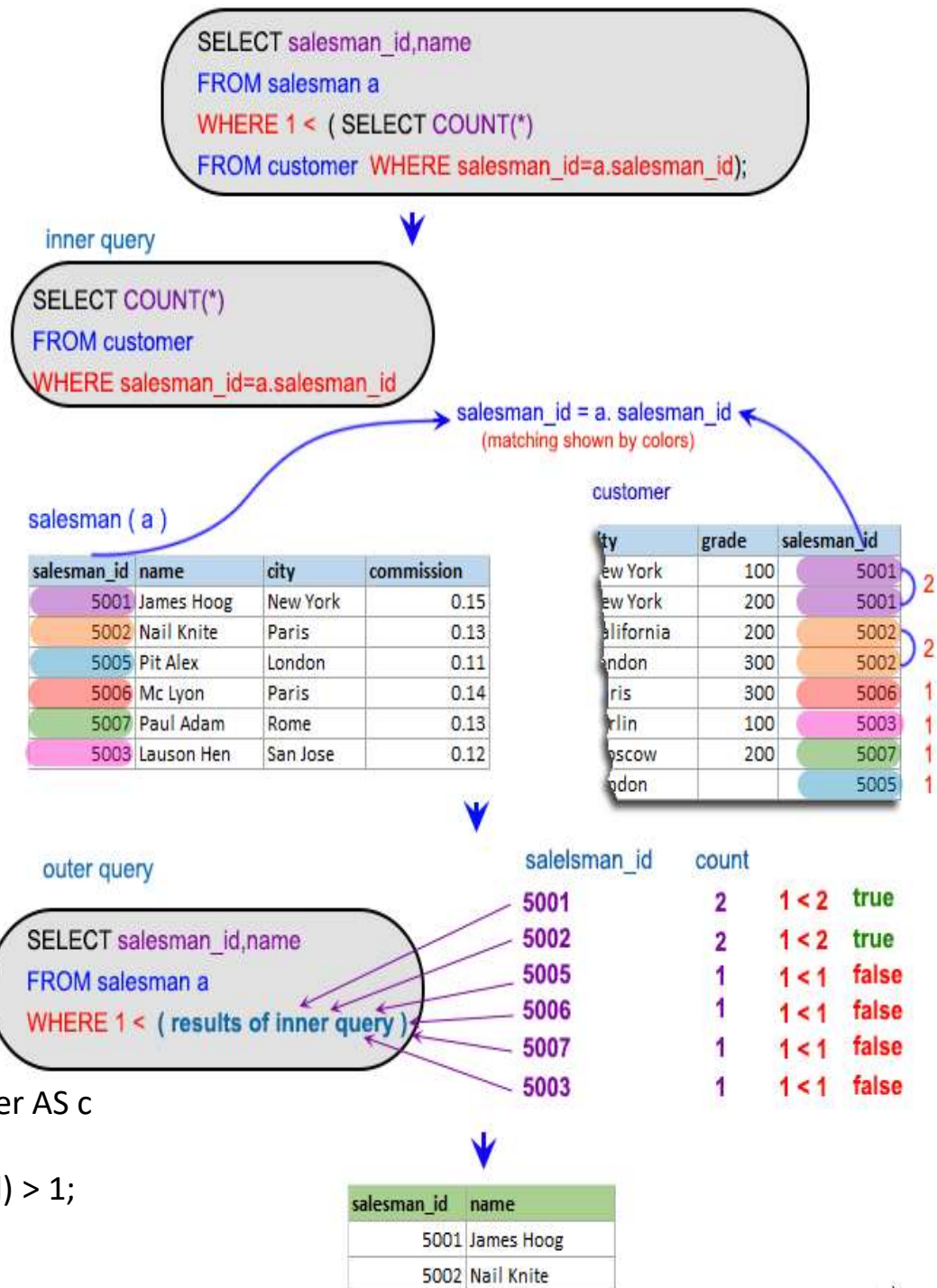
Find the name and ids of all salesmen who had more than one customer.

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;
```



# 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));
  
```

```

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));
  
```

2nd inner query

```

SELECT * FROM customer b, customer a
WHERE b.salesman_id=a.salesman_id
AND b.cust_name<>a.cust_name
  
```

customer ( a )

customer_id	cust_name	...	salesman_id
3002	Nick Rimando	...	5001
3007	Brad Davis	...	5001
3005	Graham Zusi	...	5002
3008	Julian Green	...	5002
3004	Fabian Johnson	...	5006
3009	Geoff Cameron	...	5003
3003	Jozy Altidor	...	5007
3001	Brad Guzan	...	5005

customer ( b )

customer_id	cust_name	...	salesman_id
3002	Nick Rimando	...	5001
3007	Brad Davis	...	5001
3005	Graham Zusi	...	5002
3008	Julian Green	...	5002
3004	Fabian Johnson	...	5006
3009	Geoff Cameron	...	5003
3003	Jozy Altidor	...	5007
3001	Brad Guzan	...	5005

customer_id	cust_name	...	salesman_id
3002	Nick Rimando	...	5001

customer_id	cust_name	...	salesman_id	
3002	Nick Rimando	...	5001	false
3007	Brad Davis	...	5001	true

and vice versa

customer_id	cust_name	...	salesman_id
3005	Graham Zusi	...	5002

customer_id	cust_name	...	salesman_id	
3005	Graham Zusi	...	5002	false
3008	Julian Green	...	5002	true

and vice versa

remaining all matching are false

2nd inner query returns

customer_id	cust_name	city	grade	salesman_id	customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001	3007	Brad Davis	New York	200	5001
3007	Brad Davis	New York	200	5001	3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002	3008	Julian Green	London	300	5002
3008	Julian Green	London	300	5002	3005	Graham Zusi	California	200	5002

1st inner query

```

SELECT DISTINCT salesman_id
FROM customer a
WHERE NOT EXISTS ( result of query )
  
```

salesman_id
5003
5005
5007
5006

except those  
DISTINCT salesman\_id

outer query

```

SELECT * FROM salesman
WHERE salesman_id IN ( 5006, 5003,
5007, 5005 ) results of the inner query
  
```

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
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

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



## Query 9: Equivalent Queries

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

```
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;
```

<u>salesman_id</u>	<u>name</u>	<u>city</u>	<u>commission</u>
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 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);
```



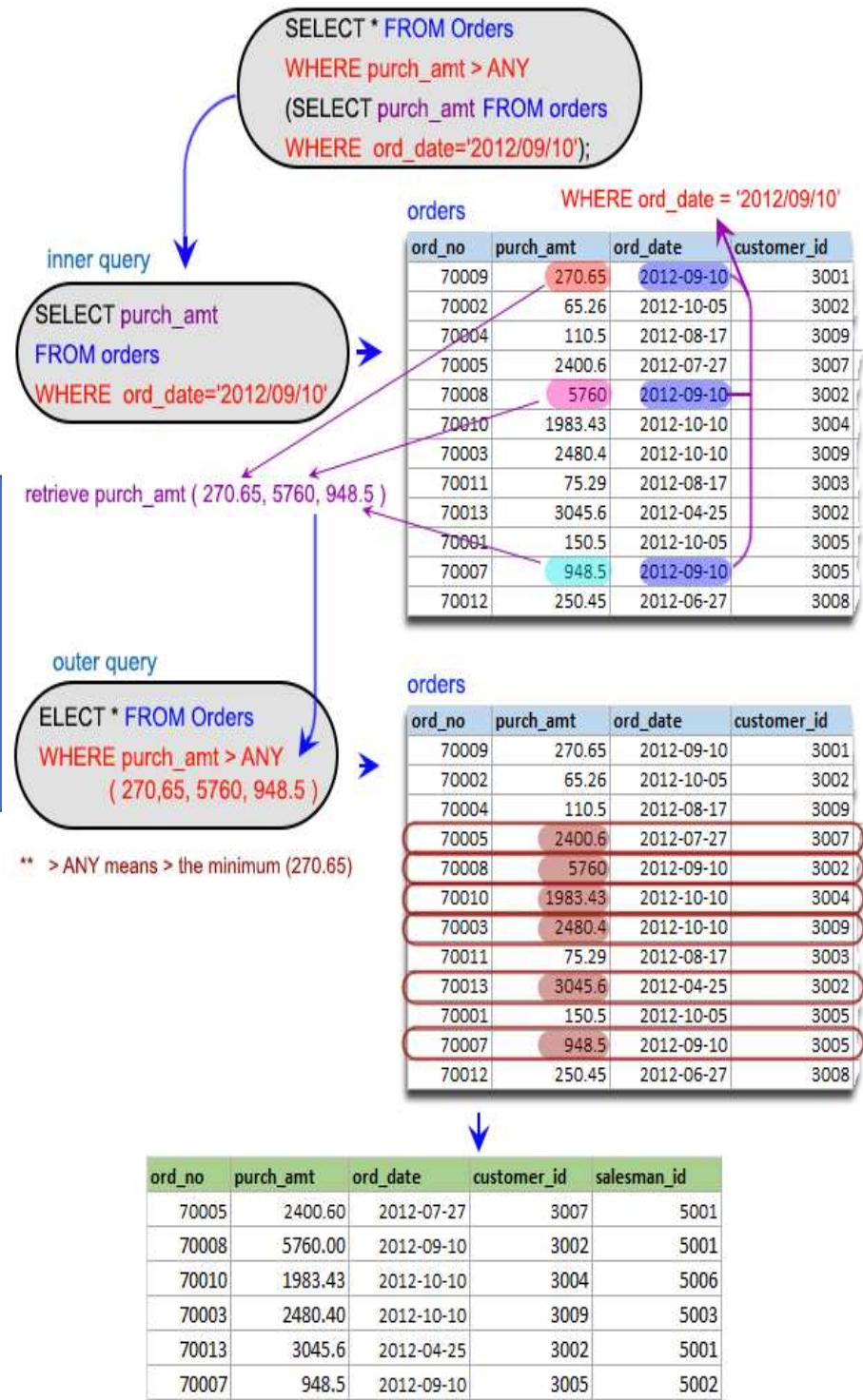
# Query 10 (using subquery)

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');
  
```



# 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');
```

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

inner query

```
SELECT grade FROM customer  
WHERE city='New York'
```

customer

WHERE city = 'New York'

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

return grade 100,200

outer query

```
SELECT * FROM customer  
WHERE grade > ALL ( 100, 200 )
```

\*\* > ALL means > the maximum

customer

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

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