PRACTICAL 1

```
create table salesman(
    -> salesnam_id int primary key,
    -> name varchar(50),
    -> city varchar(50),
    -> commission decimal(4,2));

create table customer(
    -> customer_id int primary key,
    -> customer_name varchar(50),
    -> city varchar(50),
    -> grade int,
    -> salesnam_id int,
    -> foreign key(salesnam_id) references salesman(salesnam_id));

create table orders(
    -> order_no int primary key,
    -> purch_amt decimal(10,2),
    -> order_date date,
    -> customer_id int,
    -> salesnam_id int,
    -> foreign key (customer_id)references customer(customer_id),
    -> foreign key (salesnam_id)references salesman(salesnam_id));

insert into salesman(salesnam_id, name,city, commission) values
    -> (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 adan','rome',0.13);
```

PRACTICAL 1

```
INSERT INTO customer (customer_id, customer_name, city, grade, salesnam_id) VALUES
    -> (3002, 'Nick Rimando', 'New York', 100, 5001),
    -> (3005, 'Graham zusi' , 'california' , 200, 5002),
    -> (3001, 'Brad Guzan', 'london', null,null),
    -> (3004, 'Fabian John', 'Paris', 300, 5006),
    -> (3007, 'Brad Davis', 'New York', 200, 5001),
    -> (3009, 'Geoff Camero', 'Berlin', 100,null ),
    -> (3003, 'Joey Altidore', 'Moscow', 200, 5007),
    -> (3008, 'Julian Green', 'London', 300, 5002);
```

```
INSERT INTO orders (order_no, purch_amt, order_date, customer_id, salesnam_id) VALUES
    -> (70001, 150.50, '2016-10-05', 3005, 5002),
    -> (70009, 270.65, '2016-09-10', 3001, null),
    -> (70002, 65.26, '2016-10-05', 3002, 5001),
    -> (70004, 110.5, '2016-08-17', 3009, null),
    -> (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, null),
    -> (70012, 250.45,'2016-06-27',3008,5002),
    -> (70011, 75.29,'2016-08-17', 3003, 5007);
```

1. Display name and commission for all the salesmen.

```
SELECT name, commission FROM salesman;
```

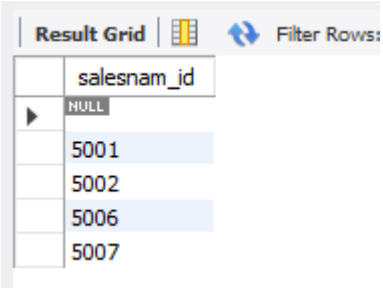
Result Grid | Filter Rows:

name	commission
james hoog	0.15
nail knite	0.13
lauson hen	0.12
pit alex	0.11
mc lyon	0.14
paul adan	0.13

PRACTICAL 1

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

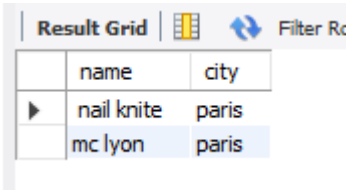
SELECT DISTINCT salesnam_id FROM orders;



salesnam_id
NULL
5001
5002
5006
5007

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

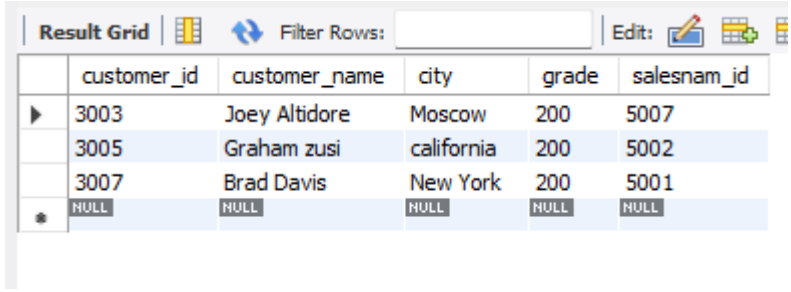
SELECT name, city FROM salesman WHERE city = 'Paris';



name	city
nail knite	paris
mc lyon	paris

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

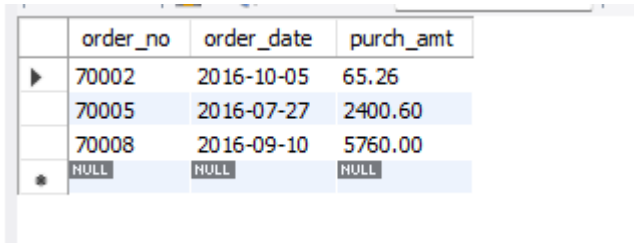
SELECT * FROM customer WHERE grade = 200;



customer_id	customer_name	city	grade	salesnam_id
3003	Joey Altidore	Moscow	200	5007
3005	Graham zusi	california	200	5002
3007	Brad Davis	New York	200	5001
NULL	NULL	NULL	NULL	NULL

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

SELECT order_no, order_date, purch_amt from orders where salesnam_id=5001;

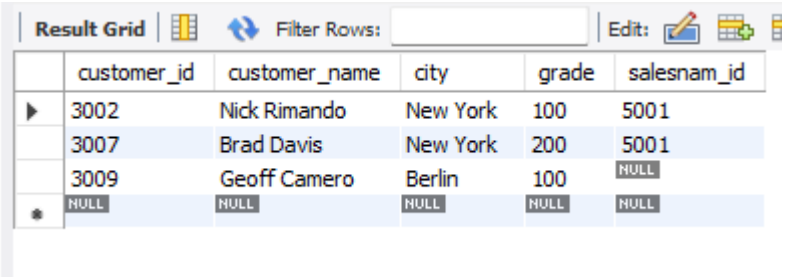


order_no	order_date	purch_amt
70002	2016-10-05	65.26
70005	2016-07-27	2400.60
70008	2016-09-10	5760.00
NULL	NULL	NULL

PRACTICAL 1

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

SELECT * FROM customer WHERE city = 'New York' OR grade <= 100;

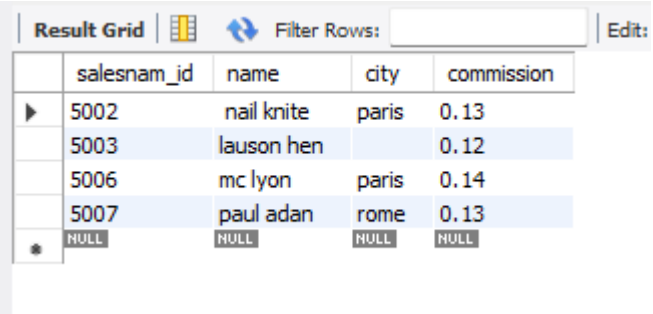


Result Grid | Filter Rows: | Edit:

customer_id	customer_name	city	grade	salesnam_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	NULL
NULL	NULL	NULL	NULL	NULL

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

SELECT * FROM salesman WHERE commission BETWEEN 0.12 AND 0.14;

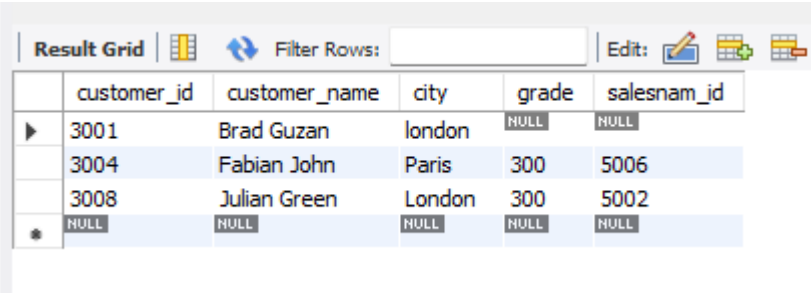


Result Grid | Filter Rows: | Edit:

salesnam_id	name	city	commission
5002	nail knite	paris	0.13
5003	lauson hen		0.12
5006	mc lyon	paris	0.14
5007	paul adan	rome	0.13
NULL	NULL	NULL	NULL

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

SELECT * FROM customer WHERE customer_name LIKE '%n';



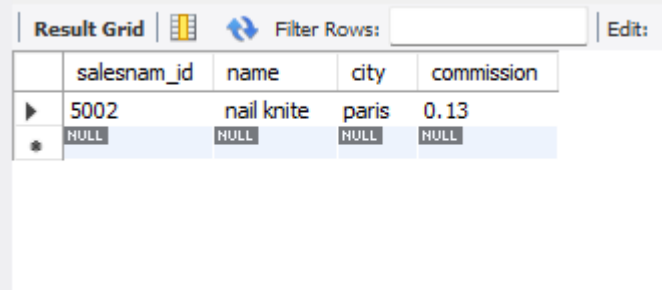
Result Grid | Filter Rows: | Edit:

customer_id	customer_name	city	grade	salesnam_id
3001	Brad Guzan	london	NULL	NULL
3004	Fabian John	Paris	300	5006
3008	Julian Green	London	300	5002
NULL	NULL	NULL	NULL	NULL

PRACTICAL 1

9. 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%';

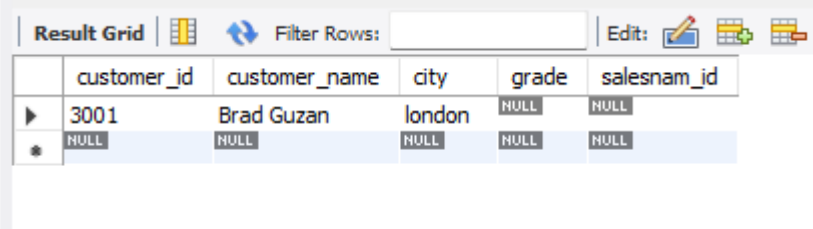


Result Grid | Filter Rows: | Edit:

salesnam_id	name	city	commission
5002	nail knite	paris	0.13
NULL	NULL	NULL	NULL

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

SELECT * FROM customer WHERE grade IS NULL;



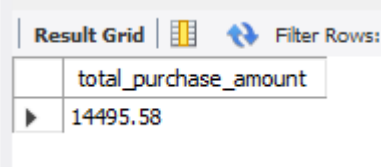
Result Grid | Filter Rows: | Edit:

customer_id	customer_name	city	grade	salesnam_id
3001	Brad Guzan	london	NULL	NULL
NULL	NULL	NULL	NULL	NULL

11. Find the total purchase amount of all orders.

SELECT SUM(purch_amt) AS total_purchase_amount

FROM orders;

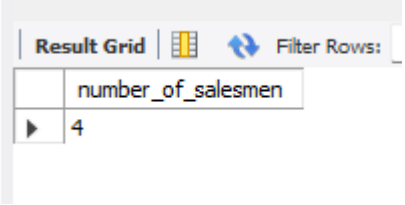


Result Grid | Filter Rows:

total_purchase_amount
14495.58

12. Find the number of salesmen currently listing for all of their customers.

SELECT COUNT(DISTINCT salesman_id) AS number_of_salesmen FROM customer;



Result Grid | Filter Rows:

number_of_salesmen
4

PRACTICAL 1

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

SELECT city, MAX(grade) AS highest_grade

FROM customer

GROUP BY city;

city	highest_grade
Berlin	100
california	200
london	300
Moscow	200
New York	200
Paris	300

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

SELECT customer_id, MAX(purch_amt) AS highest_purchase_amount

FROM order

GROUP BY customer_id;

Result Grid | Filter Rows: | Expc

customer_id	highest_purchase_amount
3001	270.65
3002	5760.00
3003	75.29
3004	1983.43
3005	948.50
3007	2400.60
3008	250.45
3009	2480.40

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

SELECT customer_id, order_date, MAX(purch_amt) AS highest_purchase_amount

FROM order

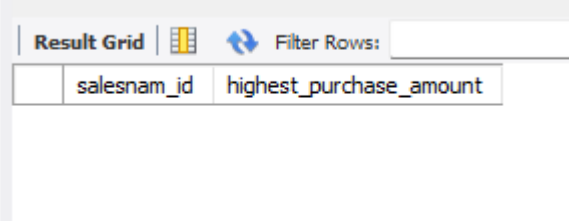
GROUP BY customer_id, order_date;

PRACTICAL 1

	customer_id	order_date	highest_purchase_amount
▶	3001	2016-09-10	270.65
	3002	2016-09-10	5760.00
	3002	2016-10-05	65.26
	3003	2016-08-17	75.29
	3004	2016-10-10	1983.43
	3005	2016-09-10	948.50
	3005	2016-10-05	150.50
	3007	2016-07-27	2400.60
	3008	2016-06-27	250.45
	3009	2016-08-17	110.50
	3009	2016-10-10	2480.40

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

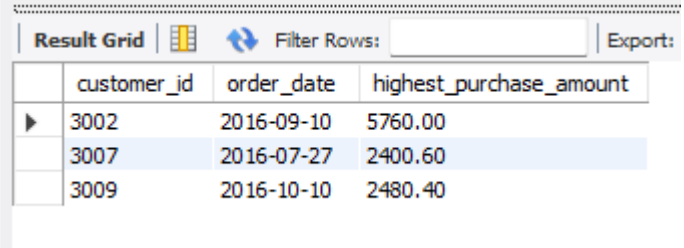
```
SELECT salesnam_id, MAX(purch_amt) AS highest_purchase_amount
FROM orders
WHERE order_date = '2012-08-17'
GROUP BY salesnam_id;
```



17. 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,
   order_date,
   highest_purchase_amount
FROM (
   SELECT
      customer_id,
      order_date,
      MAX(purch_amt) AS highest_purchase_amount
   FROM
      order
   GROUP BY
```

```
    customer_id,
    order_date
) AS daily_max
WHERE
  highest_purchase_amount > 2000;
```



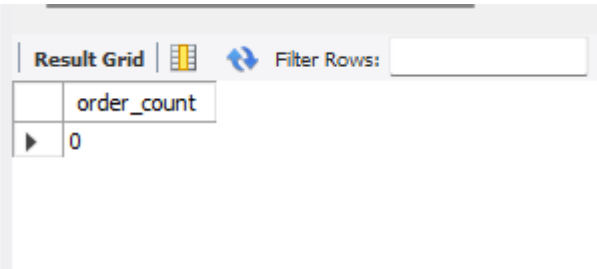
Result Grid | Filter Rows: | Export:

customer_id	order_date	highest_purchase_amount
3002	2016-09-10	5760.00
3007	2016-07-27	2400.60
3009	2016-10-10	2480.40

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

sql

```
SELECT COUNT(order_no) AS order_count
FROM orders
WHERE order_date = '2012-08-17';
```



Result Grid | Filter Rows:

order_count
0