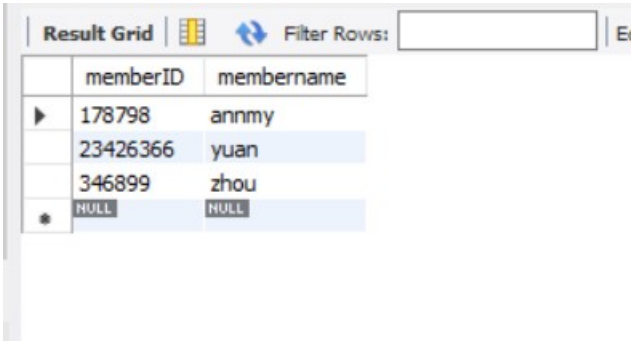
1. Display a list of all members (name and ID) sorted alphabetically by name.

```
SELECT *
FROM member
ORDER BY membername;
```



Result Grid | Filter Rows:

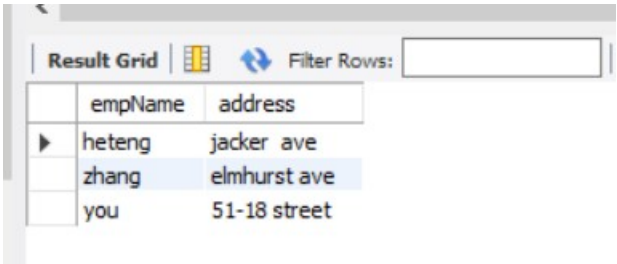
memberID	membername
178798	annmy
23426366	yuan
346899	zhou
NULL	NULL

2. For each item bought at least 12 at a time, list the item id, number bought, and time and date of the transaction

```
SELECT itemID, time, date, quantity
FROM  Item natural JOIN LineItem
WHERE quantity >=12;
```

3. list the name of each employee and the address of the store at which they work.

```
select empName, address
from Employee join Store on Employee.storeID=Store.storeID;
```

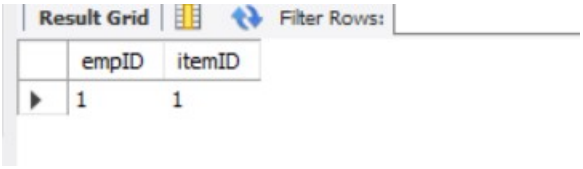


Result Grid | Filter Rows:

empName	address
heteng	jacker ave
zhang	elmhurst ave
you	51-18 street

4. List the employee id of every employee trained in the department of the item with item id 1.

```
select empID ,  itemID

from  Training natural join  Item

where itemId="1"
```



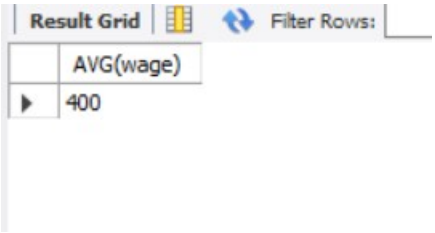
empID	itemID
1	1

5. For each item, list the item name along with each address at which it is sold.

```
SELECT itemName, address

From Item  NATURAL JOIN Store
```

6. Display the average wage at each store, sorted in descending order.

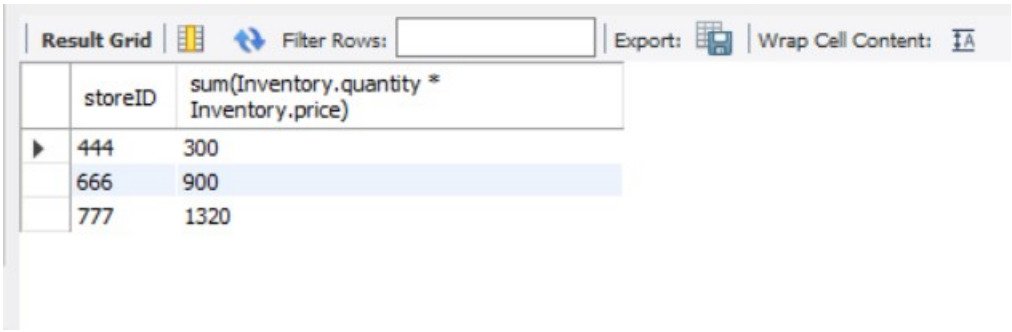
```
SELECT AVG(wage)

FROM  Store NATURAL JOIN Employee

ORDER BY wage DESC;
```



AVG(wage)
400

7. For each store, list the store id and the total cost of all products in the store. For example, if the store has 2 of a product costing $1 and three of a product costing $2, the total cost is $8.

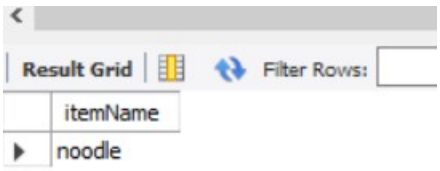
```
select Store.storeID, sum(Inventory.quantity *
Inventory.price)

from Inventory join Store on
Store.storeId=Inventory.storeID

 group by storeID
```



storeID	sum(Inventory.quantity * Inventory.price)
444	300
666	900
777	1320

8. List the name of each item sold somewhere for less than $20. Do not list any item more than once.(9.28)

```
SELECT  distinct itemName

From  Item  NATURAL JOIN  Inventory

 Where price<=20;
```



itemName
noodle

9. List the id of each store with more than ten non-managerial employees. (=employee worker)

```
select storeID, count(*)as nomanage
 from Employee
where  isManager=0
group by storeID
having count(*)>10;
```

10. For each store at which member 1 has at least one transaction, list the store id and the total number of transactions member 1 has made at that store.(member1, so just one transition)

```
select  storeID, count(*) as numTransaction from
Employee  NATURAL JOIN  Transaction
      where memberID=1
       group by  storeID, memberID;
```