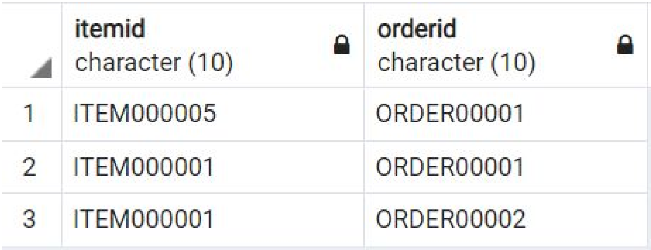
Lab 6

Relational Algebra & SQL Queries

1. Find the ordered items which are in stock.

```
select s.itemid,orderQuantity.orderid from stock as s
join (select oc.itemid,oc.quantity,o.orderid from orders as o
join ordercontains as oc
on oc.orderid= o.orderid) as orderQuantity
on orderQuantity.itemid = s.itemid
where orderQuantity.quantity<=s.quantity;
```

$$\pi_{\text{item.id,orderid}}(\text{stock}_{<\text{orderQuantity.itemid=stock.itemid AND orderQuantity.quantity<=stock.quantity}>}\bowtie(\text{orderQuantity},(\sigma_{(\text{ordercontains.itemid,ordercontains.quantity})}(\text{orders}))_{<\text{ordercontains.orderid=orders.orderid}>}\bowtie\text{ordercontains}))$$

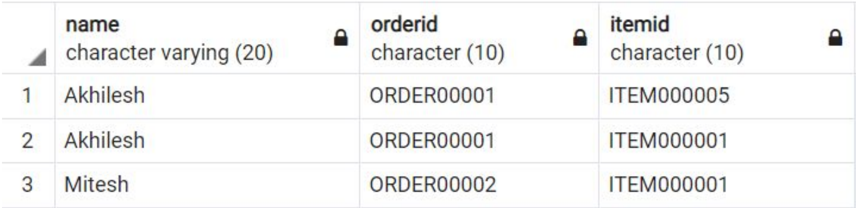


	itemid character (10)	orderid character (10)
1	ITEM000005	ORDER00001
2	ITEM000001	ORDER00001
3	ITEM000001	ORDER00002

2. Find the customers whose orders are in stock.

```
Select s.itemid,orderQuantity.orderid,orderQuantity.name from stock as s
JOIN (select oc.itemid,oc.quantity,o.orderid,c.name from orders as o
JOIN ordercontains as oc
on oc.orderid= o.orderid
JOIN customer as c
on c.customerid= o.customerid
) as orderQuantity
on orderQuantity.itemid = s.itemid
where orderQuantity.quantity<=s.quantity;
```

$$\pi_{\text{stock.item.id,orderQuantity.orderid,orderQuantity.name}}(\text{stock}_{<\text{orderQuantity.itemid=stock.itemid AND orderQuantity.quantity<=stock.quantity}>} \bowtie (\text{orderQuantity},(\sigma_{(\text{ordercontains.itemid,ordercontains.quantity, orders.orderid,customer.name})}(\text{orders}))_{<(\text{ordercontains.orderid=orders.orderid}),(\text{orders.customerid=customer.customerid})>} \bowtie \text{ordercontains} \bowtie \text{customer}))$$



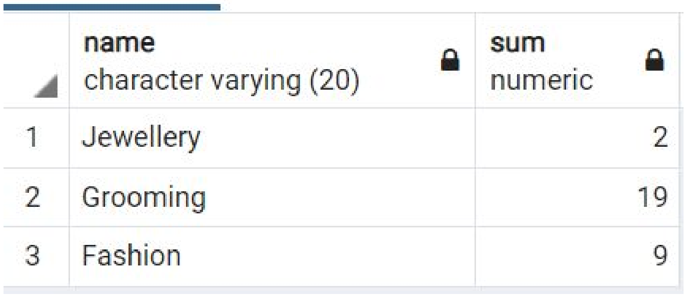
	name character varying (20)	orderid character (10)	itemid character (10)
1	Akhilesh	ORDER00001	ITEM000005
2	Akhilesh	ORDER00001	ITEM000001
3	Mitesh	ORDER00002	ITEM000001

3. Display the category names with their ordered quantities from all the orders placed.

```
Select name,sum(result.quantity) from
(Select itemcategory.name,ordercontains.quantity from itemcategory
JOIN item on item.categoryid = itemcategory.categoryid
JOIN ordercontains on item.itemid = ordercontains.itemid
JOIN orders on ordercontains.orderid = orders.orderid
) as result
GROUP BY name
```

r1 = $\pi(_{\text{itemcategory.name,ordercontains.quantity)}}$itemcategory $\bowtie_{\text{<items.categoryid=itemcategory.categoryid>)}}$item$\bowtie_{\text{<items.itemsid=ordercontains.itemid>)}}$ ordercontains$\bowtie_{\text{<ordercontains.itemsid=orders.itemid>)}}$orders

result=$\pi_{\text{(name,}}\mathcal{F}_{\text{SUM(quantity))}}$(r1)

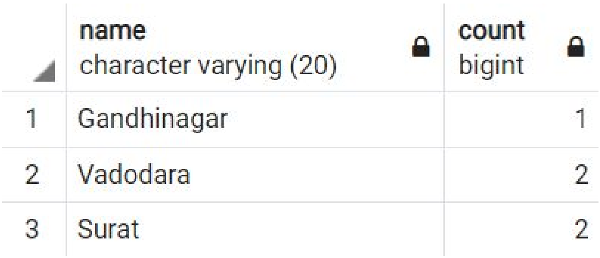


	name character varying (20)	sum numeric
1	Jewellery	2
2	Grooming	19
3	Fashion	9

4. Show the number of warehouses present in each city.

Select city.name,count(*) from warehouse
JOIN city on city.cityid=warehouse.cityid
GROUP BY city.name

$\pi_{city.name,}\mathcal{F}_{count(*)}(warehouse \bowtie_{<warehouse.cityid=city.cityid>} city)$

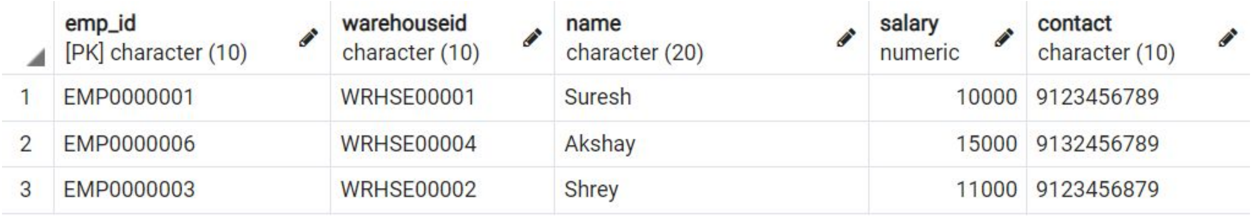


	name character varying (20)	count bigint
1	Gandhinagar	1
2	Vadodara	2
3	Surat	2

5. List all the managers of different warehouses.

Select emp.* from employee as emp
JOIN warehouse as w
on w.managerid = emp.emp_id

$\pi_{emp,*}(employee \bowtie_{<warehouse.managerid=employee.employeeid>} warehouse)$



	emp_id [PK] character (10)	warehouseid character (10)	name character (20)	salary numeric	contact character (10)
1	EMP0000001	WRHSE00001	Suresh	10000	9123456789
2	EMP0000006	WRHSE00004	Akshay	15000	9132456789
3	EMP0000003	WRHSE00002	Shrey	11000	9123456879

6. In which warehouses the employee count is below the average count.

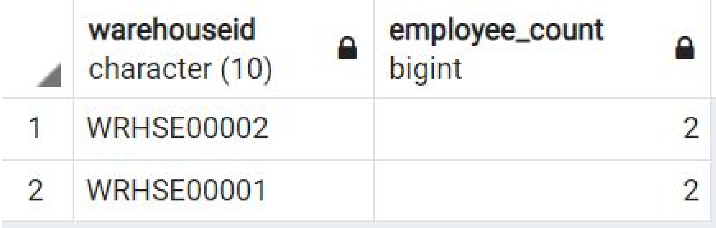
```
Select warehouseid ,count(*) as Employee_count from employee as e
GROUP BY e.warehouseid having count(*) >
(select Trunc(avg(ware_count.count) ,2)from
(select warehouseid ,count(*) as count
from employee as e
GROUP BY e.warehouseid) as ware_count)
```

$r1 = {}_{warehouseid,}\mathcal{F}_{count(*)\text{->}count}$ (employee)

$r2 = \mathcal{F}_{AVG(count)}$ (r1)

$r3 = \sigma_{(r1..count>r2.AVG(count))}$

result = $\pi_{(warehouseid,count)}(r3)$

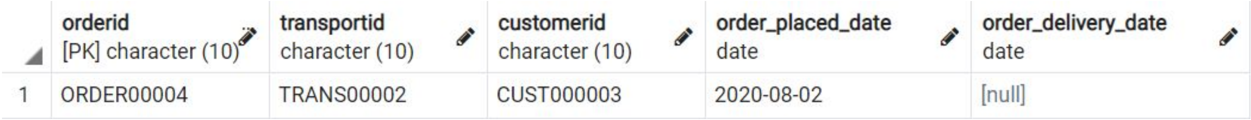


	warehouseid character (10)	employee_count bigint
1	WRHSE00002	2
2	WRHSE00001	2

7. List all the pending orders.

```
Select * from orders where order_delivery_date is null
```

$\sigma_{(order\_delivery\_date\ =\ NULL)}(orders)$



	orderid [PK] character (10)	transportid character (10)	customerid character (10)	order_placed_date date	order_delivery_date date
1	ORDER00004	TRANS00002	CUST000003	2020-08-02	[null]

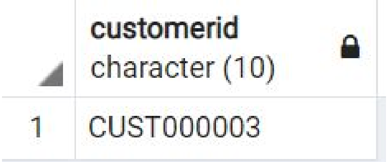
8. List the customerid with the highest number of orders placed.

```
Select customerid from
( select customerid ,count(*) from orders
GROUP BY customerid ) as maxcount
where count in (select max(count) from (select customerid ,count(*) from orders
GROUP BY customerid )as maxc)
```

$r1 = {}_{customerid,}\mathcal{F}_{count(*)\text{->}count}$ (orders)

$r2 = \mathcal{F}_{max(count)}$ (r1)

result = $\pi_{(customerid,count)}\sigma_{(r1.count=r2.max(count))}$



	customerid character (10)
1	CUST000003

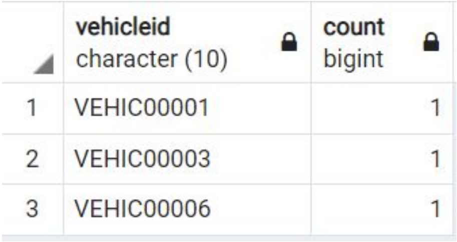
9. Name the vehicle no/id which is having the highest number of orders to transport.

```
Select vehicleid,count from
(select vehicleid, count(*) from transport
GROUP BY vehicleid) as maxcount
where count in (select max(count) from (select vehicleid, count(*) from
transport GROUP BY vehicleid)  as maxc)
```

$r1 = {}_{vehicleid,}\mathcal{F}_{count(*)\text{->}count}$ (transport)

$r2 = \mathcal{F}_{max(count)}$ (r1)

result = $\pi_{(vehicleid,count)}\sigma_{(r1.count=r2.max(count))}$



	vehicleid character (10)	count bigint
1	VEHIC00001	1
2	VEHIC00003	1
3	VEHIC00006	1