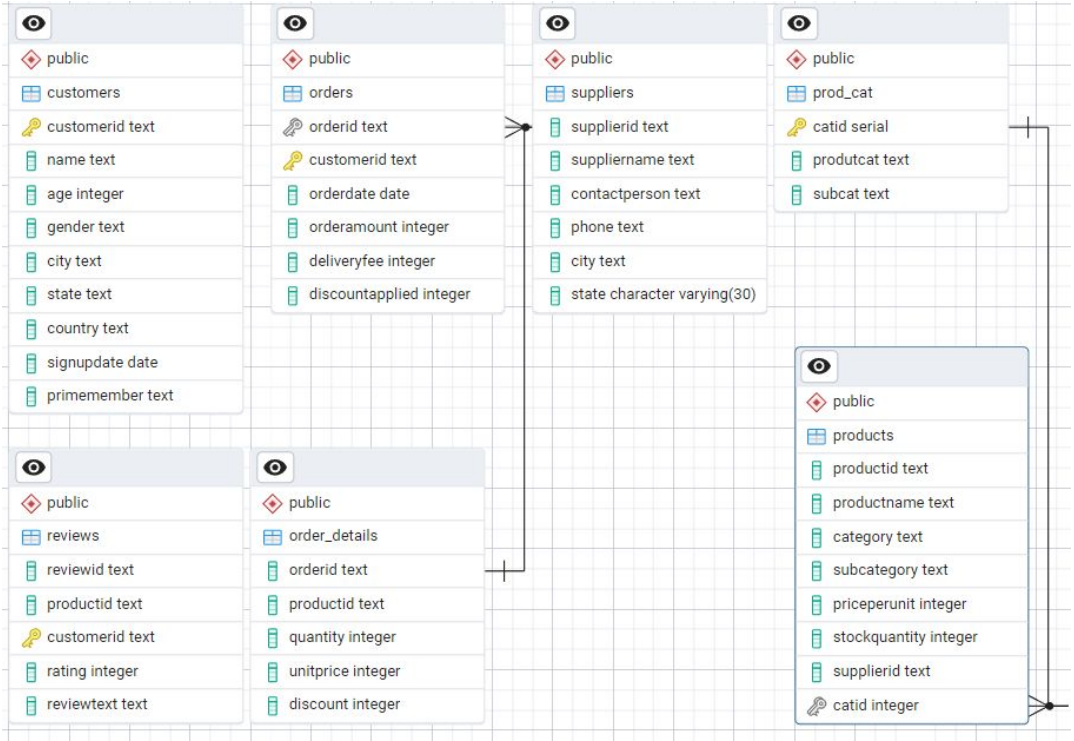
Amazon Fresh Analytics

Analyst : Prasanna Kumar

Task 1:

Create an **ER diagram** for the Amazon Fresh database to understand the relationships between tables



Task 2: Identify the **primary keys** and **foreign keys** for each table and describe their relationships.

```
-- TASK 2
Create table customers(CustomerID text primary key, Name text, Age int, Gender text,
City text, State text,  Country text, SignupDate date,  PrimeMember text)

Select * from customers

create table order_details(
OrderID text ,ProductID text, Quantity int,    UnitPrice int,Discount int)

Select * from order_details

create table orders(OrderID text primary key,CustomerID text,OrderDate date,OrderAmount int,
DeliveryFee int,    DiscountApplied int)

Select * from orders

create table products (ProductID text, ProductName text,   Category text,  SubCategory text,
PricePerUnit int,   StockQuantity int,  SupplierID text)

select * from products

create table reviews(ReviewID text,    ProductID text, CustomerID text,
```

Task 3

Retrieve all customers from a specific city.

```
-- TASK 3 Write a query to:
--          Retrieve all customers from a specific city.

Select * from customers
Select name,city,state,country from customers
where city ='Lake Thomas';
```

utput Messages Notifications

name text	city text	state text	country text
Bobby Morales	Lake Thomas	Ohio	India

Fetch all products under the "Fruits" category

```
43  select productname, category, subcategory from products where category like 'Fruits
44  --Data Definition Language (DDL) and Constraints
45  -- TASK 4  Write DDL statements to recreate the Customers table with the following
46  -- CustomerID as the primary key. (customers, orders, reviews)
47  select * from reviews
48  ALTER TABLE orders DROP CONSTRAINT orders_pkey;
49
50  select customerid, count(*)
51  from reviews
```

Data Output Messages Notifications

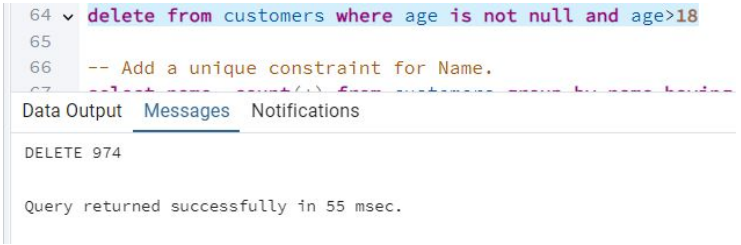
	productname text	category text	subcategory text
1	Continue Fruit	Fruits	Sub-Fruits-1
2	Sit Fruit	Fruits	Sub-Fruits-1
3	Marriage Fruit	Fruits	Sub-Fruits-1
4	Herself Fruit	Fruits	Sub-Fruits-1
5	Early Fruit	Fruits	Sub-Fruits-1
6	Report Fruit	Fruits	Sub-Fruits-1
7	Whatever Fruit	Fruits	Sub-Fruits-1
8	Service Fruit	Fruits	Sub-Fruits-1

Task 4

`CustomerID` as the primary key.

```
alter table reviews
add primary key (customerid);
-- Ensure Age cannot be null and must be greater than 18.
delete from customers where age is not null and age>18
```

Ensure `Age` cannot be null and must be greater than 18.



Add a unique constraint for `Name`.

Task 5

Insert 3 new rows into the `Products` table using `INSERT` statements.

```
74  Insert into products(productid) values(''),(''),('')
75   Select * from productS
76  -- Task 6: Update the stock quantity of a product where Productl
77
78  update products
79  set quantity = 259+1
80  where productid='e9282403-e234-4e35-a711-50acb03bbecc';
```

Data Output | Messages | Notifications

```
INSERT 0 3

Query returned successfully in 77 msec.
```

Task 6

Update the stock quantity of a product where `ProductID` matches a specific ID.

```
78  update products
79  set stockquantity = stockquantity+1
80  where productid='e9282403-e234-4e35-a711-50acb03bbecc';
81  Select * from products where ProductID= 'e9282403-e234-4e35-a711-50acb03bbecc'
82
```

Data Output | Messages | Notifications

SQL

	productid text	productname text	category text	subcategory text	priceperunit integer	stockquantity integer
1	e9282403-e234-4e35-a711-50acb03bbecc	Serve Snack	Snacks	Sub-Snacks-1	905	261

Task 7

Delete a supplier from the `Suppliers` table where their city matches a specific value.

```
83  --Task 7: Delete a supplier from the Suppliers table where their city matches a specific value.
84  delete from suppliers
85  where supplierid='158ae598-5c95-4dd7-b714-1f24332ddf9c'
86  Select * from suppliers
87  Select * from suppliers where supplierid='158ae598-5c95-4dd7-b714-1f24332ddf9c'
88
```

Data Output | Messages | Notifications

```
DELETE 0
```

Task 8

Add a CHECK constraint to ensure that ratings in the Reviews table are between 1 and 5.

```
94  alter table reviews
95  add constraint rating CHECK(rating >=1 and rating<=5);
96
```

Data Output | Messages | Notifications

```
ALTER TABLE

Query returned successfully in 66 msec.
```

Add a DEFAULT constraint for the PrimeMember column in the Customers table (default value: "No").

```
108 --Add a DEFAULT constraint for the PrimeMember
109 SELECT * FROM CUSTOMERS
110
111 alter table customers
112 alter column primemember set default 'NO';
113 --Clauses and Aggregations
```

Data Output | Messages | Notifications

```
ALTER TABLE

Query returned successfully in 76 msec.
```

Task 9

```sql
--Task 9: Write queries using:
--WHERE clause to find orders placed after 2024-01-01.
select * from orders
where orderdate > '2024-01-01'

--HAVING clause to list products with average ratings greater than 4.
select * FROM REVIEWS

select
products.productid AS productid,
products.productname as productname, reviews.rating
from products
full join reviews
    on products.productid=reviews.productid
group by
products.productid, products.productname, reviews.rating
Having avg(reviews.rating)>4 ;
--GROUP BY and ORDER BY clauses to rank products by total sales.
select
    products.productname,
    order_details.productid,
    order_details.quantity,
    rank() over(order by products.productname) as Rank_P
from
    order_details
full join
    products
on
    order_details.productid=products.productid
group by
    products.productname,
    order_details.productid,
    order_details.quantity
```

Task 10

```sql
--Task 10: Write a transaction to:
--Deduct stock from the Products table when a product is sold.
--Insert a new row in the OrderDetails table for the sale.
--Rollback the transaction if the stock is insufficient.
--Commit changes otherwise.

do $$
begin

    if exists (
        select 1
        from Products
        where ProductID = '2aa28375-c563-41b5-aa33-8e2c2e0f4db9'
          and StockQuantity >= 100
    ) then

        update Products
        set StockQuantity = StockQuantity - 5
        where ProductID = '2aa28375-c563-41b5-aa33-8e2c2e0f4db9';


        insert into Order_Details (OrderID, ProductID, Quantity, UnitPrice)
        values ('qqq', '2aa28375-c563-41b5-aa33-8e2c2e0f4db9', 5, 20.00);

        raise notice 'Transaction completed successfully.';
    else

        raise exception 'Insufficient stock for ProductID %', '2aa28375-c563-41b5-aa33-8e2c2e0f4db9';
    end if;

end $$;
```

Task 11

Join the `Orders` and `OrderDetails` tables to calculate total revenue per order.

```sql
select
    orders.orderid,
    orders.customerid,
    orders.orderdate,
    orders.orderamount,
    orders.deliveryfee,
    orders.discountapplied,
    order_details.quantity,
    order_details.unitprice,
    order_details.discount
from orders
full join
    order_details
on
    orders.orderid=order_details.orderid
where
    orders.orderid is not null and order_details.orderid is not null
```

Data Output | Messages | Notifications

Showing rows: 1 to 1000 | Page No: 1

	orderid text	customerid text	orderdate date	orderamount integer	deliveryfee integer	discountapplied integer	quantity integer	unitprice integer	discount integer
1	f844106e-171e-45d4-8e65-f258fd4a29c6	3dd0a404-b5e1-4cd4-bf73-8ed08430cf...	2025-01-01	8859	358	49	3	413	22
2	708e8320-884b-4b68-83f9-56f42b0ae211	7c718c61-9dfb-4ae9-9f8b-f02360d6ae18	2025-01-01	9448	97	91	3	454	91
3	321a0667-7362-4eaf-8ccc-3923a0d92c9a	22b7be10-8fd9-4c3f-9dc2-60b7cf97e02d	2025-01-01	8343	270	4	1	732	65
4	30edbd97-c4b2-486b-8d22-0fc6341fb37c	7b9d18c7-887a-45af-af8f-72bcac749070	2025-01-01	1029	856	7	8	559	24
5	7deaf872-f942-4f79-bb88-3f7717e24665	123fad12-537b-4a5c-a19b-f329c25927...	2025-01-01	4935	238	83	4	417	63
6	a30f5cf3-d906-4e02-8569-30a7d94ef16e	255d05dd-46b6-4f47-ba35-2cb8adff852f	2025-01-01	8970	157	76	7	30	32

Join the `Orders` and `OrderDetails` tables to calculate total revenue per order.

```sql
select
customerid,
count(orderid) as totalorders
from orders
where orderdate between '2025-01-01' AND '2025-12-31'
group by customerid
order by totalorders ASC;
```

Data Output | Messages | Notifications

	customerid [PK] text	totalorders bigint
1	585eabe0-e19b-4ac5-b64c-109926e48...	1
2	777552bc-17a4-4220-8537-225866121...	1
3	c3239e92-fe1b-4300-a854-8e4ab70119...	1
4	1dad9f63-053b-42e1-8b62-7f4281af6a...	1
5	3d1a9c3c-e7f9-45a9-a3a6-b19d3d8e4a...	1
6	2cc25564-f1a2-40c2-b4a4-fae24f2377...	1

Task 12

```
--Separate product categories and subcategories into a new table.

create table prod_cat(
catid serial primary key,
produtcat text,
subcat text
)
insert into prod_cat (produtcat, subcat)
select distinct category,subcategory
from products;

delete from prod_cat
where produtcat is null;
```

```
--Create foreign keys to maintain relationships.

alter table products
add column catid int;

alter table products
add constraint fk
foreign key (catid) references prod_cat(catid);

update Products
set catid = prod_cat.catid
from  prod_cat
where Products.Category = prod_cat.produtcat
  and Products.Subcategory = prod_cat.subcat;
```

Task 13

```sql
select
    products.productid,
    products.productname,
    sum(order_details.quantity * order_details.unitprice) as SalesRevenue
from
    products
join
    order_details
on
    products.productid = order_details.productid
group by
    products.productid, products.productname
order by
    SalesRevenue desc
limit 3;
```

Output Messages Notifications

SQL

productid text	productname text	salesrevenue bigint
1034fbb7-fdce-49e2-9230-98af72d7fa15	Capital Snack	110875
6d26d138-47e1-4082-a20f-8ea8424ac57e	Society Vegetable	103430
227f5964-63b2-4923-9a65-73ac9e7e8b39	Fish Dair	100727

```sql
--Find customers who haven't placed any orders yet.

select customers.customerid,customers.name,orders.orderid,orders.orderamount
from customers
left join orders
on customers.customerid = orders.customerid
where orders.customerid is  null
```

utput Messages Notifications

SQL

customerid text	name text	orderid text	orderamount integer
aa	ramuu	[null]	[null]
44189b67-cf70-4b7c-8315-499f872a588e	Jeremy Duran	[null]	[null]

Task 14

```sql
--Which cities have the highest concentration of Prime members?
select customers.city, count(*) as prime_count
from customers
where primemember='Yes'
group by customers.city
order by prime_count DESC
limit 10;
```

utput Messages Notifications

city text	prime_count bigint
Howardshire	1
South Kevin	1
Port Travisberg	1
Lake Thomas	1
Jenniferville	1
Wagnerburgh	1
Port Mariaburgh	1
Kimberlyfort	1
Wattschester	1
Port Karenfort	1

```sql
--What are the top 3 most frequently ordered categories?
with f_order as(
select order_details.quantity,products.category
from order_details
join products
on order_details.productid=products.productid)
select category, sum (quantity) as h_quantity
from f_order
group by category
order by h_quantity DESC
limit 3;
```

utput Messages Notifications

category text	h_quantity bigint
Meat	11108
Fruits	9936
Snacks	9651