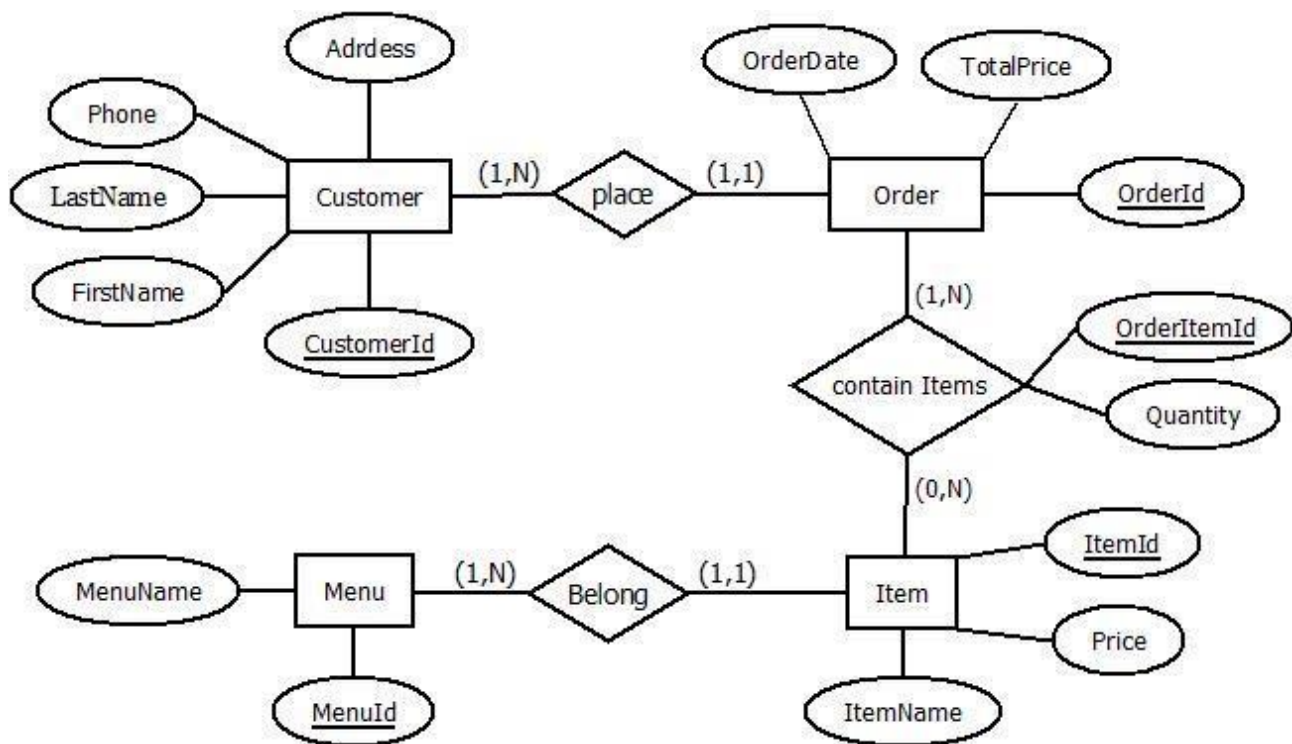


Entity Relationship Diagram (ERD) for Restaurant



Customer				
CustomerId	FirstName	LastName	Phone	Address
1	Anthony	Al Najour	70147805	Beirut
2	Rita	Data	71245685	Batroun
3	Richard	Badran	71456985	Saida
4	David	Salloum	71854632	Tripoli

Order			
OrderId	CustomerId	OrderDate	TotalPrice
1	1	20-03-24	85
2	2	21-03-24	69
3	3	22-03-23	121
4	2	23-03-24	56
5	4	24-03-23	95
6	1	25-03-24	70

OrderItem				
OrderItemId	OrderId	ItemId	Quantity	
1	1	1	1	
2	2	5	2	
3	3	2	1	
4	4	3	3	
5	5	4	1	
6	6	1	2	
7	2	1	1	

Item			
ItemId	ItemName	Price	MenuId
1	Burger	26	1
2	Pizza	19	1
3	Fries	7	2
4	Cake	10	3
5	Juice	4	4

Menu	
MenuId	MenuName
1	Main Course
2	Appetizers
3	Desserts
4	Beverages

Relational Algebra Operations

• Select operation with multiple conditions in one statement:

- **Question:** Retrieve all items with a price greater than \$10 and belong to the "Main course" Menu.

MENU_ITM $\leftarrow (Item \bowtie_{Item.MenuId = Menu.MenuId} Menu)$

R $\leftarrow \sigma_{(Price > 10 \text{ AND } MenuName = 'Main Course')} (MENU_ITM)$

R

ItemId	ItemName	Price	MenuId	MenuId	MenuName
1	Burger	26	1	1	Main Course
2	Pizza	19	1	1	Main Course

• 2 different statements of containing project operation + select operations:

- **Question 1:** Get the first and last name, total price of customers who made orders with a total price greater than \$85

CUST_ORD $\leftarrow (Customer \bowtie_{Customer.CustomerId = Order.CustomerId} Order)$

R $\leftarrow \pi_{(FirstName, LastName, TotalPrice)}(\sigma_{(TotalPrice \geq 85)}(CUST_ORD))$

FirstName	LastName	TotalPrice
Anthony	Al Najour	85
Richard	Badran	121
David	Salloum	95

- **Question 2:** Display the name and price of items with a price greater than 18

R $\leftarrow \pi_{(ItemName, Price)}(\sigma_{(Price > 18)}(Item))$

ItemName	Price
Burger	26
Pizza	19

• **2 join + select operation in one statement:**

- **Question:** Retrieve the names of customers who placed an order contain "Pizza" item.

CUST_ORD \leftarrow (**Customer** \bowtie **Customer.CustomerId = Order.CustomerId** **Order**)
ORD_ITM \leftarrow (**OrderItem** \bowtie **OrderItem.OrderId = CUST_ORD.OrderId** **CUST_ORD**)
CUST_ITM \leftarrow (**Item** \bowtie **Item.ItemId = ORD_ITM.ItemId** **ORD_ITM**)
CUST_PIZZA \leftarrow σ (ItemName = 'Pizza') **CUST_ITM**
R \leftarrow π (FirstName, LastName) (**CUST_PIZZA**)

FirstName	LastName
Richard	Badran

• **2 different statements of union operation:**

- **Question1:** Combine the all order details placed by phone numbers "71245685" and "71854632".

CUST_ORD \leftarrow (**Customer** \bowtie **Customer.CustomerId = Order.CustomerId** **Order**)
CUST1 \leftarrow π (FirstName, LastName, OrderDate, TotalPrice)(σ (Phone = '71245685') (**CUST_ORD**))
CUST2 \leftarrow π (FirstName, LastName, OrderDate, TotalPrice)(σ (Phone = '71854632') (**CUST_ORD**))
R \leftarrow **CUST1** \cup **CUST2**

FirstName	LastName	OrderDate	TotalPrice
David	Salloum	24-03-23	95
Rita	Data	21-03-24	69
Rita	Data	23-03-24	56

- **Question2:** Get All items (ItemName, MenuName) in "Appetizers" and "Desserts" menus.

MENU_ITM \leftarrow (**Item** \bowtie **Item.MenuId = Menu.MenuId** **Menu**)
ITM_APP \leftarrow π (ItemName, MenuName)(σ (MenuName = 'Appetizers') (**MENU_ITM**))
ITM_DES \leftarrow π (ItemName, MenuName)(σ (MenuName = 'Desserts') (**MENU_ITM**))
R \leftarrow **ITM_APP** \cup **ITM_DES**

ItemName	MenuName
Cake	Desserts
Fries	Appetizers

• **2 different statements of intersection operation:**

- **Question1:** Find common item name placed by customers Address "Batroun" and "Beirut".

CUST_ORD ← (Customer ⋈ Customer.CustomerId = Order.CustomerId **Order**)
ORD ← (OrderItem ⋈ OrderItem.OrderId = CUST_ORD.OrderId **CUST_ORD**)
ORD_ITM ← (Item ⋈ Item.ItemId = ORD.OrderId **ORD**)

ITM_BAT ← $\pi_{\text{ItemName}}(\sigma_{\text{Address} = \text{'Batroun'}}(\text{ORD_ITM}))$
ITM_BEI ← $\pi_{\text{ItemName}}(\sigma_{\text{Address} = \text{'Beirut'}}(\text{ORD_ITM}))$
R ← **ITM_BAT** ∩ **ITM_BEI**

ItemName
Burger

- **Question2:** Retrieve the common name of item that were ordered both in the "Appetizers" menu and by customers with addresses Batroun.

CUST_ORD ← (Customer ⋈ Customer.CustomerId = Order.CustomerId **Order**)
BAT_ORD ← $\pi_{\text{OrderId}}(\sigma_{\text{Address} = \text{'Batroun'}}(\text{CUST_ORD}))$
ORD_ITM ← (OrderItem ⋈ OrderItem.OrderId = BAT_ORD.OrderId **BAT_ORD**)
ITM_BAT ← $\pi_{\text{ItemName}}(\text{Item} \times \text{Item.ItemId} = \text{ORD_ITM.ItemId} \text{ORD_ITM})$

MENU_ITM ← (Item ⋈ Item.MenuId = Menu.MenuId **Menu**)
MENU_ITM_APP ← $\sigma_{\text{MenuName} = \text{'Appetizers'}}(\text{MENU_ITM})$
ITM_APP ← $\pi_{\text{ItemName}}(\text{MENU_ITM_APP})$
R ← **ITM_BAT** ∩ **ITM_APP**

ItemName
Fries

• **minus + select operations in one statement:**

- **Question:** Retrieve the name of items from the "Beverages" menu that have not been ordered by customers in "Beirut"

CUST_ORD \leftarrow (**Customer** \bowtie **Customer.CustomerId = Order.CustomerId** **Order**)

BEI_ORD \leftarrow ($\pi_{(OrderId)}(\sigma_{(Address = 'Beirut')}$ (**CUST_ORD**))

CUST_ITM \leftarrow (**OrderItem** \bowtie **ItemOrder.OrderId = BEI_ORD.OrderId** **BEI_ORD**)

ITM_BEI \leftarrow $\pi_{(ItemName)}(Item \bowtie Item.ItemId = CUST_ITM.ItemId$ **CUST_ITM**)

MENU_ITM \leftarrow (**Item** \bowtie **Item.MenuId = Menu.MenuId** **Menu**)

MENU_ITM_BEV \leftarrow $\sigma_{(MenuName = 'Beverages')}$ (**MENU_ITM**)

ITM_BEV \leftarrow $\pi_{(ItemName)}(MENU_ITM_BEV)$

R \leftarrow **ITM_BEV - ITM_BEI**

ItemName
Juice

• **select + union + project operations in one statement:**

- **Question:** Find the First and Last names of the customer who either address "batroun" or "Beirut"

CUST_BEI \leftarrow $\pi_{(FirstName, LastName)}(\sigma_{(Address = 'Beirut')}$ (**CUSTOMER**))

CUST_BAT \leftarrow $\pi_{(FirstName, LastName)}(\sigma_{(Address = 'Batroun')}$ (**CUSTOMER**))

R \leftarrow **CUST_BEI U CUST_BAT**

FirstName	LastName
Anthony	Al Najour
Rita	Data

• **select + union + join + project operations in one statement:**

- **Question:** retrieve the names of customers who have either placed an order contain "Pizza" or "Burger."

CUST_ORD \leftarrow (**Customer** \bowtie **Customer.CustomerId = Order.CustomerId** **Order**)

ORD_ITM \leftarrow (**OrderItem** \bowtie **OrderItem.OrderId = CUST_ORD.OrderId** **CUST_ORD**)

CUST_ITM \leftarrow (**Item** \bowtie **Item.ItemId = ORD_ITM.ItemId** **ORD_ITM**)

CUST_ORD_PIZ $\leftarrow \pi_{(FirstName, LastName)}(\sigma_{(ItemName = 'Pizza')}(\mathbf{CUST_ITM}))$

CUST_ORD_BUR $\leftarrow \pi_{(FirstName, LastName)}(\sigma_{(ItemName = 'Burger')}(\mathbf{CUST_ITM}))$

R $\leftarrow \mathbf{CUST_ORD_PIZ} \cup \mathbf{CUST_ORD_BUR}$

FirstName	LastName
Anthony	Al Najour
Richard	Badran
Rita	Data

1) **SOL queries containing the below:**

• **Average , Having, Group by**

- **Question:** Retrieve the average total price of orders for customers who have placed more than one order.

```
SELECT c.CustomerId, c.FirstName, c.LastName, AVG(o.TotalPrice) AS
AvgerageTotalPrice
FROM Customer c
JOIN Order o on c.CustomerId = o.CustomerId
GROUP BY c.CustomerId, c.FirstName, c.LastName
HAVING COUNT (o.OrderId) > 1;
```

CustomerId	FirstName	LastName	AvgerageTotalPrice
1	Anthony	Al Najour	77.5
2	Rita	Data	62.5

- **Max , Having , Group by, where**

- **Question:** Find the CustomerId, FirstName, LastName, and maximum TotalPrice for customers who have placed more than one order after "2024-01-01".

```
SELECT c.CustomerId, c.FirstName, c.LastName, MAX(o.TotalPrice) AS  
MaximumTotalPrice  
FROM Customer c  
JOIN Order o on c.CustomerId = o.CustomerId  
WHERE o.OrderDate > '2024-01-01'  
GROUP BY c.CustomerId, c.FirstName, c.LastName  
HAVING COUNT (o.OrderId) > 1;
```

CustomerId	FirstName	LastName	MaximumTotalPrice
1	Anthony	Al Najour	85
2	Rita	Data	69

- **Count , Having , Group by, Order by**

- **Question:** Find the customer Id, name and last name of customers who have placed more than one order and list them in alphabetical order.

```
SELECT DISTINCT c.CustomerId, c.FirstName, c.LastName  
FROM Customer c  
JOIN Order o on c.CustomerId = o.CustomerId  
GROUP BY c.CustomerId, c.FirstName, c.LastName  
HAVING COUNT(o.OrderId) > 1  
ORDER BY c.FirstName, c.LastName;
```

CustomerId	FirstName	LastName
1	Anthony	Al Najour
2	Rita	Data