

Database Project

Reuven Chiche 328944517

&

Eyal Seckbach 324863539

Table of Contents

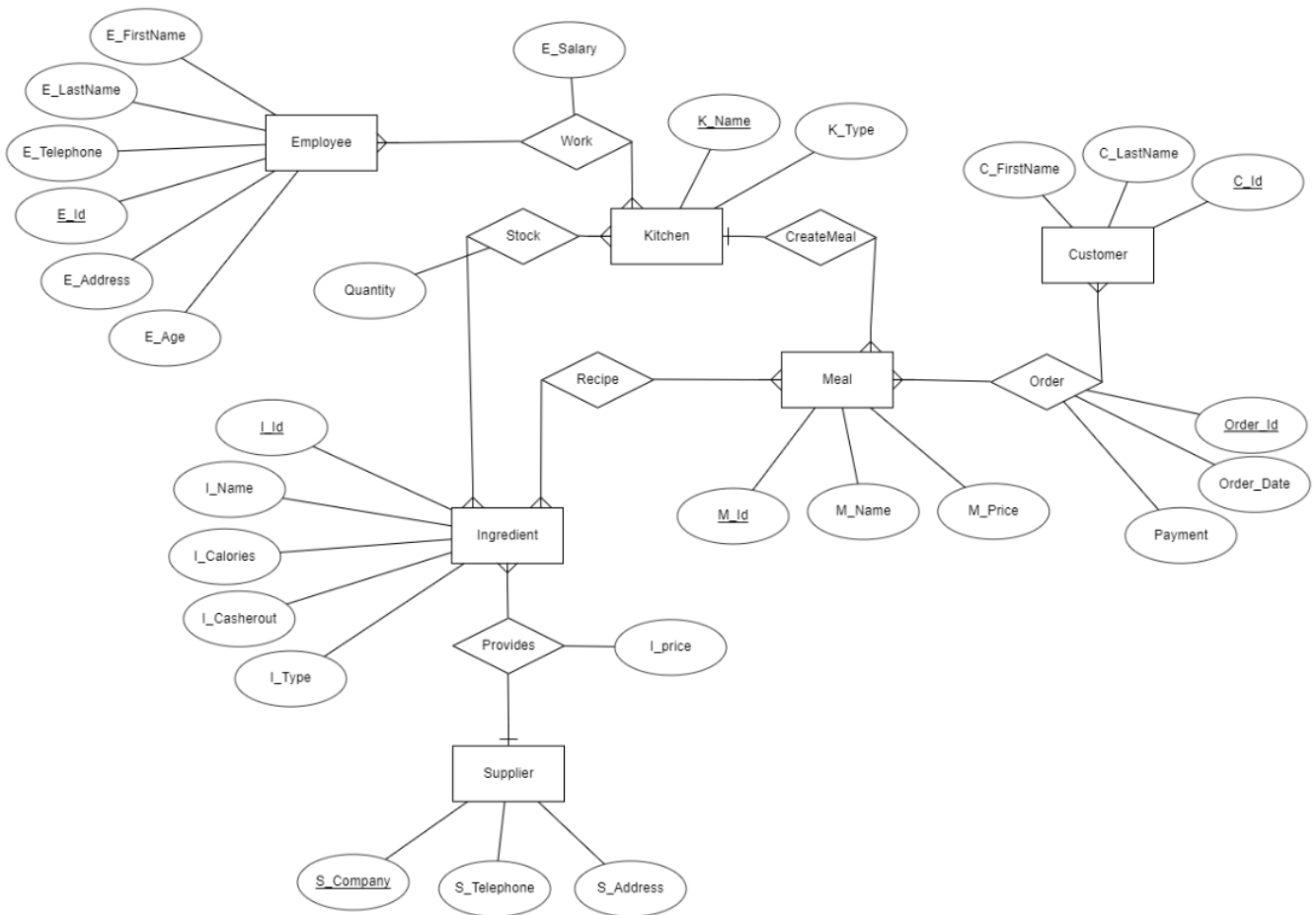
Table of Contents.....	2
Description of the organization.....	3
ERD chart:.....	4
DSD chart:.....	5
Description of the entities:.....	6
Description of the relationships.....	13
Scripts to create tables.....	14
Basic queries:.....	16
Inserting values in the tables:.....	21

Description of the organization

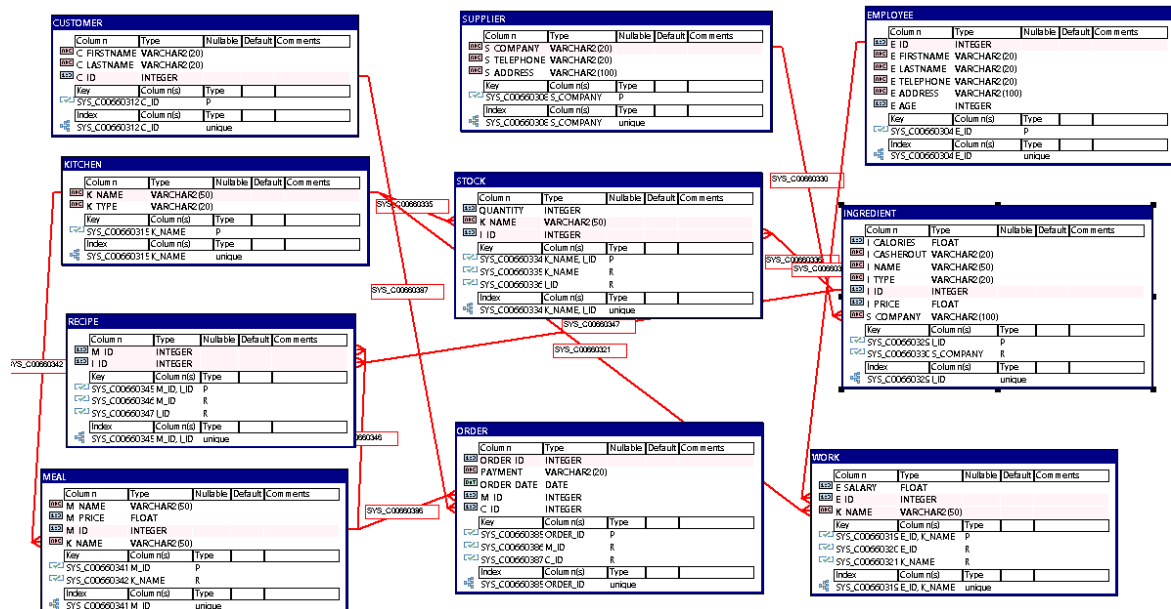
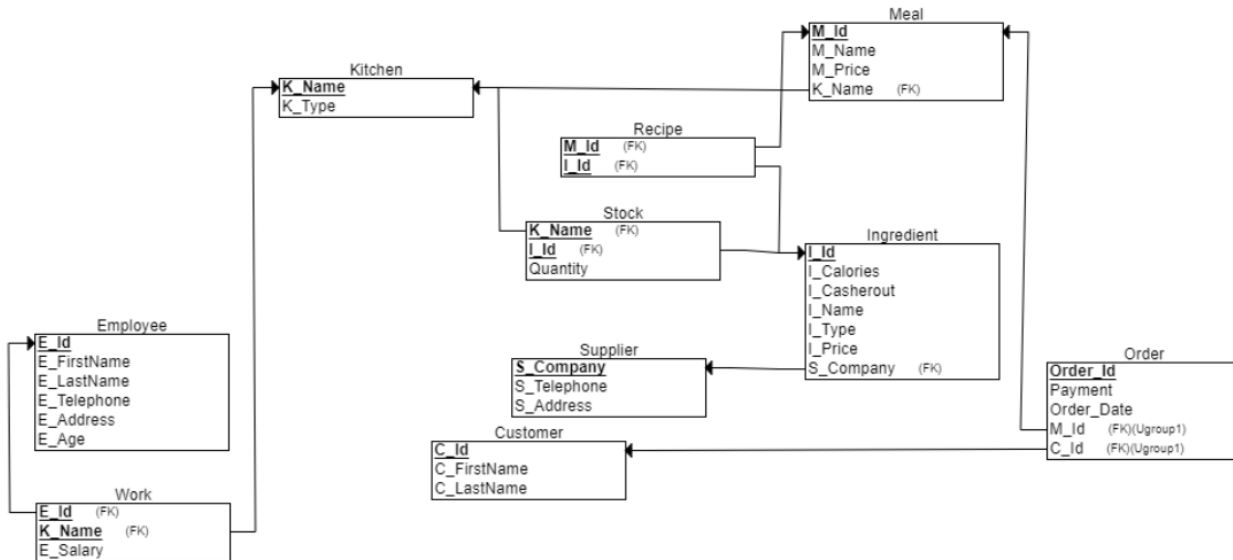
Our cafeteria database project is being developed to streamline the operations of the cafeteria. The database will include information about the inventory of ingredients, suppliers, and their prices. In addition, the database will store employees' information, such as their names, address, telephone and age. and also customers' information such as their orders and payment methods. The aim of the project is to improve efficiency and provide a better customer experience by ensuring that meals are prepared in a timely manner and that the cafeteria is well-stocked with fresh ingredients. Once implemented, the cafeteria database project will be an essential tool for managing the day-to-day operations of the cafeteria.

This cafeteria is a place where customers can come and order meals prepared by the kitchen staff. The kitchen serves a variety of cuisine types and prepares meals using ingredients provided by different supplier companies. Customers can pay for their orders using different payment methods. The cafeteria keeps track of the stock of ingredients and ensures that there is enough quantity available to prepare the meals. The employees working in the kitchen are paid salaries. The cafeteria maintains a customer database and keeps track of the orders made by each customer. Each meal has a unique identifier and the recipe for each meal is recorded, specifying the ingredients used.

ERD chart:



DSD chart:



Description of the entities:

Employee

The "Employee" entity represents the employees of the cafeteria who can work in the cafeteria. The purpose of this entity is to store information about employees of the cafeteria. By storing information about each employee, such as their unique identifier, first and last names, telephone number, address, and age.

E_Id (INT, NOT NULL)	This is the unique identifier for each employee
E_FirstName (VARCHAR, NOT NULL)	This is the first name of the employee
E_LastName (VARCHAR, NOT NULL)	This is the last name of the employee
E_Telephone (VARCHAR, NOT NULL)	This is the telephone number of the employee
E_Address (VARCHAR, NOT NULL)	This is the address of the employee
E_Age (INT, NOT NULL)	This is the age of the employee

Supplier

The "Supplier" entity represents the suppliers who can supply ingredients for the kitchens. The purpose of this entity is to store information such as the supplier's name, telephone number, and address, so that the cafeteria can manage and track the different suppliers that can supply the kitchens.

S_Company (VARCHAR, NOT NULL)	This is the name of the supplier company
S_Telephone (VARCHAR, NOT NULL)	This is the telephone number of the supplier company
S_Address (VARCHAR, NOT NULL)	This is the address of the supplier company

Customer

The "Customer" entity represents the people who place orders for food. The purpose of this entity is to store information about the customers such as their first and last name, and a unique identifier for each customer.

C_FirstName (VARCHAR, NOT NULL)	This is the first name of the customer
C_LastName (VARCHAR, NOT NULL)	This is the last name of the customer
C_Id (INT, NOT NULL)	This is the unique identifier for each customer

Kitchen

The "Kitchen" entity represents the different kitchens in the cafeteria that prepare meals. The purpose of this entity is to store information about each kitchen, such as its name and the type of cuisine it serves, so that the cafeteria can manage and track the different types of meals it offers.

K_Name (VARCHAR, NOT NULL)	This is the name of the kitchen
K_Type (VARCHAR, NOT NULL)	This is the type of cuisine served by the kitchen

Work

The "Work" entity represents the employees who work in the different kitchens in the restaurant. The purpose of this entity is to store information about the employees' salaries, the unique identifier for each employee, and the name of the kitchen where they work, so that the cafeteria can manage employee information and track which employees work in which kitchens.

E_Salary (FLOAT, NOT NULL)	This is the salary of the employee
E_Id (INT, NOT NULL)	This is the unique identifier for each employee
K_Name (VARCHAR, NOT NULL)	This is the name of the kitchen where the employee works

Ingredient

The "Ingredient" entity represents the different ingredients used to prepare the meals. The purpose of this entity is to store information about each ingredient, such as its unique identifier, name, type, calories, casherout certification, price, and the supplier company that provides the ingredient, so that the cafeteria can manage its inventory and track where ingredients come from.

I_Id (INT, NOT NULL)	This is the unique identifier for each ingredient
I_Calories (FLOAT, NOT NULL)	This is the number of calories in the ingredient
I_Casherout (VARCHAR, NOT NULL)	This is the casherout certification of the ingredient
I_Name (VARCHAR, NOT NULL)	This is the name of the ingredient
I_Type (VARCHAR, NOT NULL)	This is the type of the ingredient
I_Price (FLOAT, NOT NULL)	This is the price of the ingredient
S_Company (VARCHAR, NOT NULL)	This is the name of the supplier company that provides the ingredient

Meal

The "Meal" entity represents the different meals that are served in the cafeteria. The purpose of this entity is to store information about each meal, such as its unique identifier, name, price, and the kitchen where it is prepared so that the cafeteria can manage its menu.

M_Name (VARCHAR, NOT NULL)	This is the name of the meal
M_Price (FLOAT, NOT NULL)	This is the price of the meal
M_Id (INT, NOT NULL)	This is the unique identifier for each meal
K_Name (VARCHAR, NOT NULL)	The name of the kitchen where the meal is prepared
M_Name (VARCHAR, NOT NULL)	This is the name of the meal

Order

The "Order" entity represents the orders made by customers for meals. The purpose of this entity is to store information about each order, such as the unique identifier for each order, the payment type for the order, the date the order was made, the meal that was ordered, and the customer who made the order, so that the cafeteria can manage its orders.

Order_Id (INT, NOT NULL)	The unique identifier for each order
Payment (VARCHAR, NOT NULL)	The payment type for the order
Order_Date (DATE, NOT NULL)	The date on which the order was made
M_Id (INT, NOT NULL)	The unique identifier for the meal that was ordered
C_Id (INT, NOT NULL)	The unique identifier for the customer who made the order

Stock

The "Stock" entity represents the stock of ingredients available in the restaurant's different kitchens. The purpose of this entity is to store information about the quantity of each ingredient in stock, the name of the kitchen where the ingredient is used, and the unique identifier for each ingredient so that the cafeteria can manage its inventory.

Quantity (INT, NOT NULL)	The quantity of the ingredient in stock
K_Name (VARCHAR, NOT NULL)	The name of the kitchen where the ingredient is used
I_Id (INT, NOT NULL)	The unique identifier for the ingredient

Recipe

The "Recipe" entity represents the ingredients used to prepare each meal. The purpose of this entity is to store information about the unique identifier for each meal and the unique identifier for each ingredient used in the recipe so that the cafeteria can manage its recipes.

M_Id (INT, NOT NULL)	The unique identifier for the meal
I_Id (INT, NOT NULL)	The unique identifier for the ingredient used in the recipe

Description of the relationships

Create meal

"Create meal" is a many-to-one relationship that signifies that each meal was created by only one kitchen. This relationship helps us keep track of which kitchen was responsible for preparing a particular meal.

Provide

The "Provide" relationship is a many-to-one relationship that tells us that each ingredient is supplied by only one supplier to the kitchens. This relationship is essential in maintaining a record of the suppliers and their provided ingredients for the cafeteria.

Work, Stock, Recipe, Order

These relationships are many-to-many relationships, and have been explained above.

Scripts to create tables

Create

Creating all the entities.

```
CREATE TABLE Employee
(
    E_Id          INT          NOT NULL,
    E_FirstName   VARCHAR NOT NULL,
    E_LastName    VARCHAR NOT NULL,
    E_Telephone   VARCHAR NOT NULL,
    E_Address     VARCHAR NOT NULL,
    E_Age         INT          NOT NULL,
    PRIMARY KEY (E_Id)
);
```

```
CREATE TABLE Supplier
(
    S_Company     VARCHAR NOT NULL,
    S_Telephone   VARCHAR NOT NULL,
    S_Address     VARCHAR NOT NULL,
    PRIMARY KEY (S_Company)
);
```

```
CREATE TABLE Customer
(
    C_FirstName   VARCHAR NOT NULL,
    C_LastName    VARCHAR NOT NULL,
    C_Id          INT          NOT NULL,
    PRIMARY KEY (C_Id)
);
```

```
CREATE TABLE Kitchen
(
    K_Name        VARCHAR NOT NULL,
    K_Type        VARCHAR NOT NULL,
    PRIMARY KEY (K_Name)
);
```

```
CREATE TABLE Work
(
    E_Salary      FLOAT       NOT NULL,
    E_Id          INT          NOT NULL,
    K_Name        VARCHAR NOT NULL,
    PRIMARY KEY (E_Id, K_Name),
    FOREIGN KEY (E_Id) REFERENCES
Employee (E_Id),
    FOREIGN KEY (K_Name) REFERENCES
Kitchen (K_Name)
);
```

```
CREATE TABLE Ingredient
(
    I_Calories    FLOAT       NOT NULL,
    I_Casherout   VARCHAR NOT NULL,
    I_Name        VARCHAR NOT NULL,
    I_Type        VARCHAR NOT NULL,
    I_Id          INT          NOT NULL,
    I_Price       FLOAT       NOT NULL,
    S_Company     VARCHAR NOT NULL,
```

```

PRIMARY KEY (I_Id),

FOREIGN KEY (S_Company)
REFERENCES Supplier (S_Company)
);

```

```

CREATE TABLE Meal
(
    M_Name VARCHAR NOT NULL,
    M_Price FLOAT NOT NULL,
    M_Id INT NOT NULL,
    K_Name VARCHAR NOT NULL,

    PRIMARY KEY (M_Id),

    FOREIGN KEY (K_Name) REFERENCES
Kitchen (K_Name)
);

```

```

CREATE TABLE Order
(
    Order_Id INT NOT NULL,
    Payment VARCHAR NOT NULL,
    Order_Date DATE NOT NULL,
    M_Id INT NOT NULL,
    C_Id INT NOT NULL,

    PRIMARY KEY (Order_Id),

    FOREIGN KEY (M_Id) REFERENCES
Meal (M_Id),

    FOREIGN KEY (C_Id) REFERENCES
Customer (C_Id)
);

```

```

CREATE TABLE Stock
(
    Quantity INT NOT NULL,
    K_Name VARCHAR NOT NULL,
    I_Id INT NOT NULL,

    PRIMARY KEY (K_Name, I_Id),

    FOREIGN KEY (K_Name) REFERENCES
Kitchen (K_Name),

    FOREIGN KEY (I_Id) REFERENCES
Ingredient (I_Id)
);

```

```

CREATE TABLE Recipe
(
    M_Id INT NOT NULL,
    I_Id INT NOT NULL,

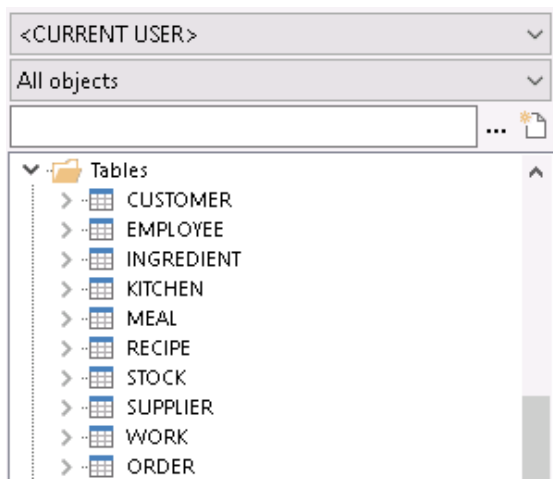
    PRIMARY KEY (M_Id, I_Id),

    FOREIGN KEY (M_Id) REFERENCES
Meal (M_Id),

    FOREIGN KEY (I_Id) REFERENCES
Ingredient (I_Id)
);

```

Results



Basic queries:

Drop

1. Drop the "Order" table:

```
DROP TABLE Order;
```

2. Drop the "Work" table:

```
DROP TABLE Work;
```

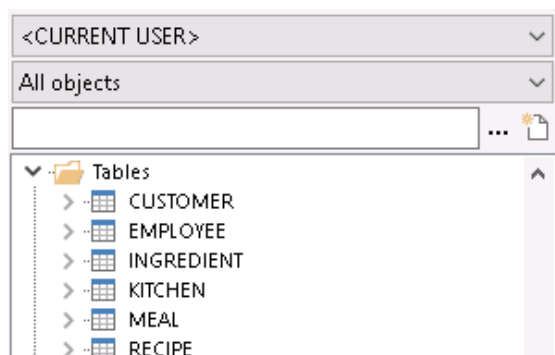
3. Drop the "Kitchen" table:

```
DROP TABLE Supplier;
```

4. Drop the "Employee" table:

```
DROP TABLE Stock;
```

Results



Insert

1. Insert a new customer:

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('John', 'Doe', 15);
```

2. Insert a new meal:

```
INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Chicken Caesar Salad', 14.5, 115, 'Juice Bar');
```

3. Insert a new order:

```
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (999, 'Cash', TO_DATE('2023-03-28', 'YYYY-MM-DD'), 115, 15);
```

4. Insert a new kitchen:

```
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Juice Bar', 'parve');
```

Result

	C_FIRSTNAME	C_LASTNAME	C_ID
▶ 1	John	Doe	15

	M_NAME	M_PRICE	M_ID	K_NAME
▶ 1	Chicken Caesar Salad	14.5	115	Juice Bar

	ORDER_ID	PAYMENT	ORDER_DATE	M_ID	C_ID
▶ 1	999	Cash	28/03/2023	115	15

	K_NAME	K_TYPE
▶ 1	Juice Bar	parve

Delete

1. Delete all orders made by a customer with ID 10:

```
DELETE FROM Order
WHERE C_Id = 10;
```

2. Delete all ingredients that have not been used in any meal:

```
DELETE FROM Ingredient
WHERE I_Id NOT IN (
    SELECT I_Id
    FROM Recipe
);
```

3. Delete all meals that have not been ordered at least once:

```
DELETE FROM Meal
WHERE M_Id NOT IN (
    SELECT M_Id
    FROM Order
);
```

4. Delete all employees who not work:

```
DELETE FROM Employee
WHERE E_Id NOT IN (
    SELECT E_Id
    FROM Work
);
```

Update

1. Update the price of all meals to increase by 10%:

```
UPDATE Meal
SET M_Price = M_Price * 1.1;
```

2. Update the phone number of all employees who are working:

```
UPDATE Employee
SET E_Telephone = '050-1876543'
WHERE E_Id IN (
    SELECT E_Id
    FROM Work
);
```

3. Update the quantity of all ingredients to increase by 50 units:

```
UPDATE Stock
SET Quantity = Quantity + 50;
```

4. Update the salary of all workers to increase by 1000:

```
UPDATE Work
SET E_Salary = E_Salary + 1000;
```

Select

1. Retrieve all the orders made by customers with ID 101:

```
SELECT *
FROM Order
WHERE C_Id = 1;
```

Result:

	ORDER_ID	PAYMENT	ORDER_DATE	M_ID	C_ID
1	1	Cash	01/01/2022	101	1

2. Retrieve the names of all the ingredients used in a specific meal:

```
SELECT I_Name
FROM Ingredient NATURAL JOIN Meal
WHERE M_Id = 101;
```

Result:

	I_NAME	
1	Flour	...
2	Sugar	...
3	Salt	...
4	Olive Oil	...
5	Soy Sauce	...
6	Beef	...
7	Chicken	...
8	Shrimp	...
9	Lemon	...
10	Tomato	...
11	Garlic	...
12	Onion	...
13	Paprika	...
14	Cinnamon	...

3. Retrieve the total amount spent by each customer:

```
SELECT C_FirstName, C_LastName, SUM(M_Price) AS Total_Spent
FROM Order NATURAL JOIN Customer NATURAL JOIN Meal
GROUP BY C_FirstName, C_LastName;
```

Result:

	C_FIRSTNAME		C_LASTNAME		TOTAL_SPENT
1	Marshall	...	Eriksen	...	7.99
2	Jane	...	Doe	...	15.99
3	Phoebe	...	Buffay	...	9.99
4	John	...	Doe	...	10.99
5	Chandler	...	Bing	...	22.99
6	Rachel	...	Green	...	25.99
7	Michael	...	Smith	...	18.99
8	Ross	...	Geller	...	12.99
9	Robin	...	Scherbatsky	...	13.99
10	Monica	...	Geller	...	8.99
11	Ted	...	Mosby	...	16.99
12	Barney	...	Stinson	...	6.99
13	Lily	...	Aldrin	...	24.99
14	Joey	...	Tribbiani	...	11.99

4. Retrieve the most popular meals (in descending order of popularity):

```
SELECT M_Id, COUNT(M_Id) AS Popularity
FROM Order NATURAL JOIN Meal
GROUP BY M_Id
ORDER BY Popularity DESC;
```

Result:

	M_ID	POPULARITY
1	113	1
2	108	1
3	112	1
4	102	1
5	110	1
6	101	1
7	111	1
8	114	1
9	104	1
10	105	1
11	109	1
12	103	1
13	106	1
14	107	1

Inserting values in the tables:

Employee

```
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (1, 'John', 'Doe', '054-1234567', '123 Main St.', 30);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (2, 'Jane', 'Smith', '055-2345678', '456 Oak Ave.', 25);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (3, 'Bob', 'Johnson', '056-3456789', '789 Maple Rd.', 35);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (4, 'Alice', 'Williams', '052-4567890', '246 Elm St.', 28);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (5, 'Mike', 'Brown', '050-5678901', '135 Pine Ave.', 42);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (6, 'Karen', 'Taylor', '058-6789012', '678 Cedar Ln.', 31);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (7, 'David', 'Wilson', '051-7890123', '910 Oak Rd.', 27);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (8, 'Amy', 'Miller', '059-8901234', '345 Elm Ave.', 29);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (9, 'Chris', 'Lee', '056-9012345', '789 Maple St.', 33);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (10, 'Maria', 'Garcia', '052-0123456', '246 Oak Ave.', 26);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (11, 'Tom', 'Anderson', '057-1234567', '123 Pine St.', 38);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (12, 'Emily', 'Clark', '055-2345678', '456 Cedar Ave.', 24);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (13, 'Josh', 'Wright', '058-3456789', '789 Maple Rd.', 30);
INSERT INTO Employee (E_Id, E_FirstName, E_LastName, E_Telephone, E_Address,
E_Age)
VALUES (14, 'Samantha', 'Martin', '050-4567890', '246 Oak St.', 32);
```

Supplier

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('ABC Inc.', '050-1234567', '123 Main St.');
```

S_Company	S_Telephone	S_Address
ABC Inc.	050-1234567	123 Main St.
XYZ Corp.	055-2345678	456 Oak Ave.
123 Co.	056-3456789	789 Maple Rd.
456 Ltd.	052-4567890	246 Elm St.
789 LLC	050-5678901	135 Pine Ave.
DEF Inc.	058-6789012	678 Cedar Ln.
GHI Corp.	051-7890123	910 Oak Rd.
JKL Co.	059-8901234	345 Elm Ave.
MNO Ltd.	056-9012345	789 Maple St.
PQR LLC	052-0123456	246 Oak Ave.
STU Inc.	057-1234567	123 Pine St.
VWX Corp.	055-2345678	456 Cedar Ave.
YZA Co.	058-3456789	789 Maple Rd.
BCD Ltd.	050-4567890	246 Oak St.

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('XYZ Corp.', '055-2345678', '456 Oak Ave.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('123 Co.', '056-3456789', '789 Maple Rd.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('456 Ltd.', '052-4567890', '246 Elm St.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('789 LLC', '050-5678901', '135 Pine Ave.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('DEF Inc.', '058-6789012', '678 Cedar Ln.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('GHI Corp.', '051-7890123', '910 Oak Rd.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('JKL Co.', '059-8901234', '345 Elm Ave.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('MNO Ltd.', '056-9012345', '789 Maple St.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('PQR LLC', '052-0123456', '246 Oak Ave.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('STU Inc.', '057-1234567', '123 Pine St.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('VWX Corp.', '055-2345678', '456 Cedar Ave.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('YZA Co.', '058-3456789', '789 Maple Rd.');
```

```
INSERT INTO Supplier (S_Company, S_Telephone, S_Address)
VALUES ('BCD Ltd.', '050-4567890', '246 Oak St.');
```

Customer

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('John', 'Doe', 1);
```

C_FirstName	C_LastName	C_Id
John	Doe	1
Jane	Doe	2
Michael	Smith	3
Rachel	Green	4
Ross	Geller	5
Monica	Geller	6
Chandler	Bing	7
Phoebe	Buffay	8
Joey	Tribbiani	9

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Jane', 'Doe', 2);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Michael', 'Smith', 3);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Rachel', 'Green', 4);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Ross', 'Geller', 5);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Monica', 'Geller', 6);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Chandler', 'Bing', 7);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Phoebe', 'Buffay', 8);
```

```
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Joey', 'Tribbiani', 9);
```



```

INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Ted', 'Mosby', 10);
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Barney', 'Stinson', 11);
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Lily', 'Aldrin', 12);
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Marshall', 'Eriksen', 13);
INSERT INTO Customer (C_FirstName, C_LastName, C_Id)
VALUES ('Robin', 'Scherbatsky', 14);

```

Kitchen

```

INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Italiano', 'parve');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Sushi House', 'meat');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Burger Joint', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Kebab House', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Wok n Roll', 'meat');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('La Patisserie', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('El Mariachi', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Spice Route', 'meat');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('The Greek', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Aloha Kitchen', 'meat');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Noodle House', 'meat');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Fish n Chips', 'parve');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Waffle House', 'dairy');
INSERT INTO Kitchen (K_Name, K_Type)
VALUES ('Pizza Planet', 'dairy');

```

Work

```

INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (5000, 1, 'Italiano');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (4000, 2, 'Sushi House');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (4500, 3, 'Burger Joint');
INSERT INTO Work (E_Salary, E_Id, K_Name)

```

```

VALUES (3500, 4, 'Kebab House');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (6000, 5, 'Wok n Roll');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (3500, 6, 'La Patisserie');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (4000, 7, 'El Mariachi');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (5000, 8, 'Spice Route');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (4500, 9, 'The Greek');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (5500, 10, 'Aloha Kitchen');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (3500, 11, 'Noodle House');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (4000, 12, 'Fish n Chips');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (3000, 13, 'Waffle House');
INSERT INTO Work (E_Salary, E_Id, K_Name)
VALUES (5500, 14, 'Pizza Planet');

```

Ingredient

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (100.5, 'Rabanout', 'Flour', 'parve', 1, 2.5, 'ABC Inc.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (150.2, 'Badatz', 'Sugar', 'parve', 2, 4.0, 'ABC Inc.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (80.0, 'Rabanout', 'Salt', 'parve', 3, 1.5, 'XYZ Corp.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (200.0, 'LaMehadrine', 'Olive Oil', 'parve', 4, 8.0, 'JKL Co.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (150.0, 'Badatz', 'Soy Sauce', 'parve', 5, 4.5, 'JKL Co.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (250.0, 'Rabanout', 'Beef', 'meat', 6, 12.0, 'XYZ Corp.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (200.0, 'Rabanout', 'Chicken', 'meat', 7, 10.0, '456 Ltd.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (150.0, 'LaMehadrine', 'Shrimp', 'parve', 8, 15.0, '456 Ltd.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (100.0, 'Rabanout', 'Lemon', 'parve', 9, 3.0, 'MNO Ltd.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (80.0, 'LaMehadrine', 'Tomato', 'parve', 10, 2.5, 'MNO Ltd.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (50.0, 'Badatz', 'Garlic', 'parve', 11, 1.5, 'VWX Corp.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (30.0, 'Badatz', 'Onion', 'parve', 12, 1.0, 'VWX Corp.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (120.0, 'Rabanout', 'Paprika', 'parve', 13, 2.0, 'BCD Ltd.');
```

```

INSERT INTO Ingredient (I_Calories, I_Casherout, I_Name, I_Type, I_Id,
I_Price, S_Company)
VALUES (100.0, 'Badatz', 'Cinnamon', 'parve', 14, 3.0, 'BCD Ltd.');
```

Meal

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Hamburger', 10.99, 101, 'Italiano');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Pizza', 15.99, 102, 'Sushi House');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Sushi', 18.99, 103, 'Burger Joint');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Steak', 25.99, 104, 'Kebab House');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Pad Thai', 12.99, 105, 'Wok n Roll');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Tacos', 8.99, 106, 'La Patisserie');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Salmon', 22.99, 107, 'El Mariachi');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Fried Rice', 9.99, 108, 'Spice Route');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Pasta', 11.99, 109, 'The Greek');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Kebab', 16.99, 110, 'Aloha Kitchen');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Soup', 6.99, 111, 'Noodle House');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('BBQ Ribs', 24.99, 112, 'Fish n Chips');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Falafel', 7.99, 113, 'Waffle House');
```

```

INSERT INTO Meal (M_Name, M_Price, M_Id, K_Name)
VALUES ('Pho', 13.99, 114, 'Pizza Planet');
```

Order

```
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (1, 'Cash', TO_DATE('2022-01-01','YYYY-MM-DD'), 101, 1);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (2, 'Credit Card', TO_DATE('2022-01-02','YYYY-MM-DD'), 102, 2);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (3, 'Cash', TO_DATE('2022-01-03','YYYY-MM-DD'), 103, 3);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (4, 'Credit Card', TO_DATE('2022-01-04','YYYY-MM-DD'), 104, 4);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (5, 'Cash', TO_DATE('2022-01-05','YYYY-MM-DD'), 105, 5);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (6, 'Credit Card', TO_DATE('2022-01-06','YYYY-MM-DD'), 106, 6);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (7, 'Cash', TO_DATE('2022-01-07','YYYY-MM-DD'), 107, 7);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (8, 'Credit Card', TO_DATE('2022-01-08','YYYY-MM-DD'), 108, 8);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (9, 'Cash', TO_DATE('2022-01-09','YYYY-MM-DD'), 109, 9);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (10, 'Credit Card', TO_DATE('2022-01-10','YYYY-MM-DD'), 110, 10);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (11, 'Cash', TO_DATE('2022-01-11','YYYY-MM-DD'), 111, 11);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (12, 'Credit Card', TO_DATE('2022-01-12','YYYY-MM-DD'), 112, 12);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (13, 'Cash', TO_DATE('2022-01-13','YYYY-MM-DD'), 113, 13);
INSERT INTO Order (Order_Id, Payment, Order_Date, M_Id, C_Id)
VALUES (14, 'Credit Card', TO_DATE('2022-01-14','YYYY-MM-DD'), 114, 14);
```

Stock

```
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (10, 'Italiano', 1);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (20, 'Sushi House', 2);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (15, 'Burger Joint', 3);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (5, 'Kebab House', 4);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (30, 'Wok n Roll', 5);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (25, 'La Patisserie', 6);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (12, 'El Mariachi', 7);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (8, 'Spice Route', 8);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (18, 'The Greek', 9);
```

```
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (7, 'Aloha Kitchen', 10);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (13, 'Noodle House', 11);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (22, 'Fish n Chips', 12);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (9, 'Waffle House', 13);
INSERT INTO Stock (Quantity, K_Name, I_Id)
VALUES (16, 'Pizza Planet', 14);
```

Recipe

```
INSERT INTO Recipe (M_Id, I_Id)
VALUES (101, 1);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (101, 2);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (101, 3);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (102, 4);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (102, 5);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (102, 6);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (103, 7);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (103, 8);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (103, 9);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (104, 10);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (104, 11);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (104, 12);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (105, 13);
INSERT INTO Recipe (M_Id, I_Id)
VALUES (105, 14);
```

Results

	E_ID	E_FIRSTNAME	E_LASTNAME	E_TELEPHONE	E_ADDRESS	E_AGE
1	1	John	Doe	054-1234567	123 Main St.	30
2	2	Jane	Smith	055-2345678	456 Oak Ave.	25
3	3	Bob	Johnson	056-3456789	789 Maple Rd.	35
4	4	Alice	Williams	052-4567890	246 Elm St.	28
5	5	Mike	Brown	050-5678901	135 Pine Ave.	42
6	6	Karen	Taylor	058-6789012	678 Cedar Ln.	31
7	7	David	Wilson	051-7890123	910 Oak Rd.	27
8	8	Amy	Miller	059-8901234	345 Elm Ave.	29
9	9	Chris	Lee	056-9012345	789 Maple St.	33
10	10	Maria	Garcia	052-0123456	246 Oak Ave.	26
11	11	Tom	Anderson	057-1234567	123 Pine St.	38
12	12	Emily	Clark	055-2345678	456 Cedar Ave.	24
13	13	Josh	Wright	058-3456789	789 Maple Rd.	30
14	14	Samantha	Martin	050-4567890	246 Oak St.	32

	C_FIRSTNAME	C_LASTNAME	C_ID
1	John	Doe	1
2	Jane	Doe	2
3	Michael	Smith	3
4	Rachel	Green	4
5	Ross	Geller	5
6	Monica	Geller	6
7	Chandler	Bing	7
8	Phoebe	Buffay	8
9	Joey	Tribbiani	9
10	Ted	Mosby	10
11	Barney	Stinson	11
12	Lily	Aldrin	12
13	Marshall	Eriksen	13
14	Robin	Scherbatsky	14

	K_NAME	K_TYPE
1	Italiano	parve
2	Sushi House	meat
3	Burger Joint	dairy
4	Kebab House	dairy
5	Wok n Roll	meat
6	La Patisserie	dairy
7	El Mariachi	dairy
8	Spice Route	meat
9	The Greek	dairy
10	Aloha Kitchen	meat
11	Noodle House	meat
12	Fish n Chips	parve
13	Waffle House	dairy
14	Pizza Planet	dairy

	I_CALORIES	I_CASHEROUT	I_NAME	I_TYPE	I_ID	I_PRICE	S_COMPANY
1	100.5	Rabanout	Flour	parve	1	2.5	ABC Inc.
2	150.2	Badatz	Sugar	parve	2	4	ABC Inc.
3	80	Rabanout	Salt	parve	3	1.5	XYZ Corp.
4	200	LaMehadrine	Olive Oil	parve	4	8	JKL Co.
5	150	Badatz	Soy Sauce	parve	5	4.5	JKL Co.
6	250	Rabanout	Beef	meat	6	12	XYZ Corp.
7	200	Rabanout	Chicken	meat	7	10	456 Ltd.
8	150	LaMehadrine	Shrimp	parve	8	15	456 Ltd.
9	100	Rabanout	Lemon	parve	9	3	MNO Ltd.
10	80	LaMehadrine	Tomato	parve	10	2.5	MNO Ltd.
11	50	Badatz	Garlic	parve	11	1.5	VWX Corp.
12	30	Badatz	Onion	parve	12	1	VWX Corp.
13	120	Rabanout	Paprika	parve	13	2	BCD Ltd.
14	100	Badatz	Cinnamon	parve	14	3	BCD Ltd.

	M_NAME	M_PRICE	M_ID	K_NAME
1	Hamburger	10.99	101	Italiano
2	Pizza	15.99	102	Sushi House
3	Sushi	18.99	103	Burger Joint
4	Steak	25.99	104	Kebab House
5	Pad Thai	12.99	105	Wok n Roll
6	Tacos	8.99	106	La Patisserie
7	Salmon	22.99	107	El Mariachi
8	Fried Rice	9.99	108	Spice Route
9	Pasta	11.99	109	The Greek
10	Kebab	16.99	110	Aloha Kitchen
11	Soup	6.99	111	Noodle House
12	BBQ Ribs	24.99	112	Fish n Chips
13	Falafel	7.99	113	Waffle House
14	Pho	13.99	114	Pizza Planet

	M_ID	I_ID
1	101	1
2	101	2
3	101	3
4	102	4
5	102	5
6	102	6
7	103	7
8	103	8
9	103	9
10	104	10
11	104	11
12	104	12
13	105	13
14	105	14

	ORDER_ID	PAYMENT	ORDER_DATE	M_ID	C_ID
1	1	Cash	01/01/2022	101	1
2	2	Credit Card	02/01/2022	102	2
3	3	Cash	03/01/2022	103	3
4	4	Credit Card	04/01/2022	104	4
5	5	Cash	05/01/2022	105	5
6	6	Credit Card	06/01/2022	106	6
7	7	Cash	07/01/2022	107	7
8	8	Credit Card	08/01/2022	108	8
9	9	Cash	09/01/2022	109	9
10	10	Credit Card	10/01/2022	110	10
11	11	Cash	11/01/2022	111	11
12	12	Credit Card	12/01/2022	112	12
13	13	Cash	13/01/2022	113	13
14	14	Credit Card	14/01/2022	114	14

	QUANTITY	K_NAME	I_ID
1	10	Italiano	1
2	20	Sushi House	2
3	15	Burger Joint	3
4	30	Wok n Roll	5
5	25	La Patisserie	6
6	12	El Mariachi	7
7	8	Spice Route	8
8	18	The Greek	9
9	7	Aloha Kitchen	10
10	13	Noodle House	11
11	22	Fish n Chips	12
12	9	Waffle House	13
13	16	Pizza Planet	14
14	5	Kebab House	4

	S_COMPANY	S_TELEPHONE	S_ADDRESS
1	ABC Inc.	050-1234567	123 Main St.
2	XYZ Corp.	055-2345678	456 Oak Ave.
3	456 Ltd.	052-4567890	246 Elm St.
4	789 LLC	050-5678901	135 Pine Ave.
5	DEF Inc.	058-6789012	678 Cedar Ln.
6	GHI Corp.	051-7890123	910 Oak Rd.
7	JKL Co.	059-8901234	345 Elm Ave.
8	MNO Ltd.	056-9012345	789 Maple St.
9	PQR LLC	052-0123456	246 Oak Ave.
10	STU Inc.	057-1234567	123 Pine St.
11	VWX Corp.	055-2345678	456 Cedar Ave.
12	YZA Co.	058-3456789	789 Maple Rd.
13	BCD Ltd.	050-4567890	246 Oak St.
14	123 Co.	056-3456789	789 Maple Rd.

	E_SALARY	E_ID	K_NAME	
1	5000	1	Italiano	...
2	4000	2	Sushi House	...
3	4500	3	Burger Joint	...
4	3500	4	Kebab House	...
5	6000	5	Wok n Roll	...
6	3500	6	La Patisserie	...
7	4000	7	El Mariachi	...
8	5000	8	Spice Route	...
9	4500	9	The Greek	...
10	5500	10	Aloha Kitchen	...
11	3500	11	Noodle House	...
12	4000	12	Fish n Chips	...
13	3000	13	Waffle House	...
14	5500	14	Pizza Planet	...