**Description of the Project and its Scope:**

This café serves foods and different kinds of beverage. The café would like to keep track of its sales and to be able to have a record of its customers.

The scope of the project consists of the following sections:

* Items
* Manager
* Employee
* Customer

The item code and its price should be recorded. Moreover, it is important to keep track of how many times each item is ordered.

The manger keeps information of managers and manages its employees.

This café has three type of employees: cashier, chef, and waiter. Cashiers deal with money and bills. Chef prepares food. And a waiter takes customers’ orders.

**Business Rules:**

1. Each item can either be food or beverage.
2. All items have similar attributes, like (name, product type code, description, and item number). Items also have unique attributes; for example, food (contains nuts, and its price) and drinks (flavor, and price).
3. Manager manages many employees and an employee doesn’t have more than one manager
4. Each customer can have many orders, but each order can be taken only by one waiter.
5. Each order can contain many items and each item can be in many orders.
6. Each bill can belong to many orders, but each order can’t have more than one bill.

**Entities and Attributes:**

**Entity #1: Manager**

The manager entity keeps information about café management team.

1. Attribute #1 (PK): MAN\_ID

* Description: MAN\_ID is our primary key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: MAN\_FNAM

* Description: MAN\_FNAME contains manager first name.
* Data Type: char (40)
* Sample value: Edward, John, etc.

1. Attribute #3: MAN\_LNAM

* Description: MAN\_LNAME contains manager last name.
* Data Type: char (40)
* Sample value:  Smith, etc.

1. Attribute #4: MAN\_ADDRESS\_ST

* Description: MAN\_ADDRESS\_ST contains manager street’s name.
* Data Type: char (80)
* Sample value: Beauregard ST, etc.

1. Attribute #5: MAN\_ADDRESS\_CITY

* Description: MAN\_ADDRESS\_CITY contains manager address.
* Data Type: char (20)
* Sample value: Alexandria, etc.

1. Attribute #6: MAN\_ADDRESS\_STATE

* Description: MAN\_ADDRESS\_STATE contains manager address.
* Data Type: char (20)
* Sample value: Virginia, etc.

1. Attribute #7: MAN\_HOURLY\_PAY

* Description: MAN\_HOURLY\_PAY contains information about manager hourly pay.
* Data Type: float
* Sample value: 12 dollars, etc.

1. Attribute #8: MAN\_CONTACT

* Description: MAN\_CONTACT contains information about manager contact.
* Data Type: char
* Sample value: Phone number, etc.

**Entity #2: EMPLOYEE**

The employee entity keeps information about employees. Since not all employee share common attributes, there is a need for specialization hierarchy.

1. Attribute #1 (PK): EMP\_ID

* Description: EMP\_ID is our primary key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: EMP\_FNAM

* Description: EMP\_FNAME contains employee first name.
* Data Type: char (40)

Sample value: Edward, John, etc.

1. Attribute #3: EMP\_LNAM

* Description: EMP\_LNAME contains employee last name.
* Data Type: char (40)
* Sample value:  Smith, etc.

1. Attribute #4: EMP\_ADDRESS\_ST

* Description: EMP\_ADDRESS\_ST contains employee street’s name.
* Data Type: char (80)
* Sample value: Beauregard ST, etc.

1. Attribute #5: EMP\_ADDRESS\_CITY

* Description: EMP\_ADDRESS\_CITY contains employee address.
* Data Type: char (20)
* Sample value: Alexandria, etc.

1. Attribute #6: EMP\_ADDRESS\_STATE

* Description: EMP\_ADDRESS\_STATE contains employee address.
* Data Type: char (20)
* Sample value: Virginia, etc.

1. Attribute #7: EMP\_CONTACT

* Description: EMP\_CONTACT contains information about employee contact.
* Data Type: char
* Sample value: Phone number

1. Attribute #8 (FK): MAN\_ID

* Description: MAN\_ID contains information about manager ID. It is the foreign key.
* Data Type: INT
* Sample value:  100, etc.

1. Attribute #9: EMP\_JOB\_CODE

* Description: EMP\_JOB\_CODE contains job code for employees.
* Data Type: char (40)
* Sample value:  CH, CA, WA, etc.

**Entity #3: CHEF**

The chef entity is a sub-entity and will include unique attributes that describe chefs.

1. Attribute #1 (PK) (FK): EMP\_ID

* Description: EMP\_ID is our primary key and foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: CHEF\_STYLE

* Description: CHEF\_STYLE contains information about style of each chef.
* Data Type: char

Sample value: Italian, Mexican, etc.

1. Attribute #3: CHEF\_HOURLY\_PAY

* Description: CHEF\_HOURLY\_PAY contains information about chef hourly payment.
* Data Type: float

Sample value: 12 dollars, etc.

1. Attribute #4: CHEF\_JOB\_DESC

* Description: CHEF\_JOB\_DESC contains information about job description for chefs.
* Data Type: char

Sample value: Chef

**Entity #4: CASHIER**

The cashier entity is a sub-entity and will include unique attributes that describe cashiers.

1. Attribute #1 (PK) (FK): EMP\_ID

* Description: EMP\_ID is our primary key and foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: CASHIER\_HOURLY\_PAY

* Description: CASHIER\_HOURLY\_PAY contains information about cashier hourly payment.
* Data Type: float

Sample value: 12 dollars, etc.

1. Attribute #3: CASHIER\_JOB\_DESC

* Description: CASHIER\_JOB\_DESC contains information about his or her job.
* Data Type: char

Sample value: Cashier

**Entity #5: WAITER**

The waiter entity is a sub-entity and will include unique attributes that describe waiters.

1. Attribute #1 (PK) (FK): EMP\_ID

* Description: EMP\_ID is our primary key and foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: WAITER\_HOURLY\_PAY

* Description: WAITER\_HOURLY\_PAY contains information about waiter hourly payment.
* Data Type: float

Sample value: 12 dollars, etc.

1. Attribute #3: WAITER\_JOB\_DESC

* Description: WAITER\_JOB\_DESC contains information about job description for waiters.
* Data Type: char
* Sample value: Waiters

**Entity #6: BILL**

The BILL entity keeps information about customer bill.

1. Attribute #1 (PK): BILL\_ID

* Description: BILL\_ID is our primary key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: BILL\_DESC

- Description: BILL\_DESC contains information about customer bill.

* Data Type: char

Sample value: customer’s special request

1. Attribute #3(FK): CUS\_ID

* Description: CUS\_ID is our foreign key that connects customer to bill.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #4 (FK): EMP\_ID

* Description: EMP\_ID is our foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

**Entity #7: CUSTOMER**

The CUSTOMER entity keeps information about customer.

1. Attribute #1 (PK): CUS\_ID

* Description: CUS\_ID is our primary key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: CUS\_FNAM

* Description: CUS\_FNAME contains customer first name.
* Data Type: char (40)

Sample value: Edward, John, etc.

1. Attribute #3: CUS\_LNAM

* Description: CUS\_LNAME contains customer last name.
* Data Type: char (40)
* Sample value:  Smith, etc.

1. Attribute #4: CUS\_ADDRESS\_ST

* Description: CUS\_ADDRESS\_ST contains customer street’s name.
* Data Type: char (80)
* Sample value: Beauregard ST, etc.

1. Attribute #5: CUS\_ADDRESS\_CITY

* Description: CUS\_ADDRESS\_CITY contains customer address.
* Data Type: char (20)
* Sample value: Alexandria, etc.

1. Attribute #6: CUS\_ADDRESS\_STATE

* Description: CUS\_ADDRESS\_STATE contains customer address.
* Data Type: char (20)
* Sample value: Virginia, etc.

1. Attribute #7: CUS\_CONTACT

* Description: EMP\_CONTACT contains information about customer contact.
* Data Type: char

Sample value: Phone number

**Entity #8: ORDER**

The ORDER entity will include unique attributes that describe customer order.

1. Attribute #1 (PK): OR\_ID

* Description: OR\_ID is our primary key
* Data Type: int

- Sample value: 1, 100, 200, etc.

1. Attribute #2: OR\_DESC

* Description: OR\_DESC contains information about what customer has ordered.
* Data Type: char
* Sample value: description

1. Attribute #3 (FK): EMP\_ID

* Description: EMP\_ID is our foreign key that connects employee to order.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #4 (FK): CUS\_ID

* Description: CUS\_ID is our foreign key that connects customer to order.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #5 (FK): BILL\_ID

* Description: BILL\_ID is our foreign key that connects bills to order.
* Data Type: int
* Sample value: 1, 100, 200, etc.

**Entity #9: ITEM**

The item entity keeps information about items. Since not all items share common attributes, there is a need for specialization hierarchy.

1. Attribute #1 (PK): ITEM\_ID

* Description: ITEM\_ID is our primary key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

2. Attribute #2: ITEM\_NAM

* Description: ITEM\_NAME contains item name.
* Data Type: char (40)

Sample value: Bread, etc.

3.Attribute #3: ITEM\_DESC

* Description: ITEM\_DESC contains descriptions about each item.
* Data Type: char (40)

Sample value:  description, etc.

4.Attribute #4: ITEM\_CODE

* Description: ITEM\_DESC contains code of food and drinks.
* Data Type: char (40)

Sample value:  FO, DR

**Entity #10: FOOD**

The food entity is a sub-entity and will include unique attributes that describe food.

1. Attribute #1 (PK) (FK): ITEM\_ID

* Description: ITEM\_ID is our primary key and foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: FOOD\_CONTAIN\_NUTS

* Description: FOOD\_ CONTAIN\_NUTS contains descriptions about food.
* Data Type: char (40)
* Sample value:  FALSE, TRUE

1. Attribute #3: FOOD\_PRICE

* Description: FOOD\_ PRICE contains price.
* Data Type: float
* Sample value:  12 dollars, etc.

**Entity #11: DRINK**

The drink entity is a sub-entity and will include unique attributes that describe drink.

1. Attribute #1 (PK) (FK): ITEM\_ID

* Description: ITEM\_ID is our primary key and foreign key.
* Data Type: int
* Sample value: 1, 100, 200, etc.

1. Attribute #2: DR\_FLAVOR

* Description: DR\_FLAVOR contains descriptions about drink.
* Data Type: char (40)
* Sample value:  FLAVOR

1. Attribute #3: DR\_PRICE

* Description: DR\_ PRICE contains price.
* Data Type: float
* Sample value:  5 dollars, etc.

**Entity #12:** **ORDER\_ITEM**

Since there is a M:N relationships between ORDER and ITEM, there is a need for a bridge. The ORDER\_ITEM is a bridge. It has a composite primary key.

1. Attribute #1 (PK)(FK): ITEM\_ID
   * + Description: ITEM\_ID is our primary key and foreign key.
     + Data Type: int
     + Sample value: 1, 100, 200, etc.
2. Attribute #2 (PK)(FK): OR\_ID

* Description: OR\_ID is our primary key and foreign key.
* Data Type: int

- Sample value: 1, 100, 200, etc.

1. Attribute #3 OI\_QUANTITY

* Description: OI\_QUANTITY is the number of each item.
* Data Type: int

- Sample value: 1, 3 etc.

**Entities and Relationships:**

**Relationship: Manager manages employees**

**Relationship type:** 1:M

Explanation: business rules state, “a manager can manage many employee and an employee doesn’t have more than one manager.”

**Relationship participation:** mandatory on the both side of the relationship

**Cardinality:**

* the cardinality of MANAGER is (1, 1)
* the cardinality of EMPLOYEE is (1, M)

**Relationship strength:** Weak

Explanation: The relationship is weak because the primary key of the EMPLOYEE entity does not contain the primary key of the MANAGER entity.

**Relationship:** EMPLOYEE “is a” CHEF

**Relationship type:** 1:1

Explanation: Sine the EMPLOYEE entity is a super-entity and the CHEF entity is a sub-entity, there is a 1:1 relationship between them.

**Relationship participation:** NA

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the CHEF entity contains the primary key of the EMPLOYEE entity.

**Relationship:** EMPLOYEE “is a” WAITER

**Relationship type:** 1:1

Explanation: Sine the EMPLOYEE entity is a super-entity and the WAITER entity is a sub-entity, there is a 1:1 relationship between them.

**Relationship participation:** NA

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the WAITER entity contains the primary key of the EMPLOYEE entity.

**Relationship:** EMPLOYEE “is a” CASHIER

**Relationship type:** 1:1

Explanation: Sine the EMPLOYEE entity is a super-entity and the CASHIER entity is a sub-entity, there is a 1:1 relationship between them.

**Relationship participation:** NA

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the CASHIER entity contains the primary key of the EMPLOYEE entity.

**Relationship:** ITEM “is a” FOOD

**Relationship type:** 1:1

Explanation: Sine the ITEM entity is a super-entity and the FOOD entity is a sub-entity, there is a 1:1 relationship between them.

**Relationship participation:** NA

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the FOOD entity contains the primary key of the ITEM entity.

**Relationship:** ITEM “is a” DRINK

**Relationship type:** 1:1

Explanation: Sine the ITEM entity is a super-entity and the DRINK entity is a sub-entity, there is a 1:1 relationship between them.

**Relationship participation:** NA

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the DRINK entity contains the primary key of the ITEM entity.

**Relationship: WAITER TAKES ORDER**

**Relationship type:** 1:M

Explanation: business rules state, “Each cashier can take many orders, but each order can have been taken only by one waiter.”

**Relationship participation:** mandatory on the both side of the relationship

**Cardinality:**

* the cardinality of WAITER is (1, 1)
* the cardinality of ORDER is (1, M)

**Relationship strength:** Weak

Explanation: The relationship is weak because the primary key of the ORDER entity does not contain the primary key of the WAITER entity.

**Relationship: CUSTOMER PLACES ORDER**

**Relationship type:** 1:M

Explanation: business rules state, “Each customer can place many orders, but each order can belong to only one customer.”

**Relationship participation:** mandatory on the both side of the relationship

**Cardinality:**

* the cardinality of CUSTOMER is (1, 1)
* the cardinality of ORDER is (1, M)

**Relationship strength:** Weak

Explanation: The relationship is weak because the primary key of the ORDER entity does not contain the primary key of the CUSTOMER entity.

**Relationship: CUSTOMER PAYS BILL**

**Relationship type:** 1:M

Explanation: business rules state, “Each customer can have many bills, but each bill belongs to only one customer.”

**Relationship participation:** mandatory on the both side of the relationship

**Cardinality:**

* the cardinality of CUSTOMER is (1, 1)
* the cardinality of BILL is (1, M)

**Relationship strength:** Weak

Explanation: The relationship is weak because the primary key of the BILL entity does not contain the primary key of the CUSTOMER entity.

**Relationship: ORDER HAS BILL**

**Relationship type:** 1:M

Explanation: business rules state, “Each order can have one bill, but each bill can include many orders.”

**Relationship participation:** mandatory on the both side of the relationship

**Cardinality:**

* the cardinality of BILL is (1, 1)
* the cardinality of ORDER is (1, M)

**Relationship strength:** Weak

Explanation: The relationship is weak because the primary key of the ORDER entity does not contain the primary key of the BILL entity.

**Relationship:** ORDER includes ORDER\_ITEM

**Relationship type:** 1:M

Explanation: business rules state, “Each order could have many items …” There is a M:N relationship between ITEM and ORDER so the bridge is needed. When a bridge is created, it’s linked with two 1:M relationships with the original entities. The ORDER entity is on the “one” side of the relationship and the ORDER\_ITEM entity is on the “many” side of the relationship.

**Relationship participation:** mandatory on the “one” side of the relationship and mandatory on the “many” side of the relationship.

**Cardinality:**

* the cardinality of ORDER is (1, 1)
* the cardinality of ORDER\_ITEM is (1, N)

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the ORDER\_ITEM entity contains the primary key of the ORDER entity.

**Relationship:** ITEM being requested ORDER\_ITEM

**Relationship type:** 1:M

Explanation: business rules state, “Each order could have many items …” There is a M:N relationship between ITEM and ORDER so the bridge is needed. When a bridge is created, it’s linked with two 1:M relationships with the original entities. The ITEM entity is on the “one” side of the relationship and the ORDER\_ITEM entity is on the “many” side of the relationship.

**Relationship participation:** mandatory on the “one” side of the relationship and mandatory on the “many” side of the relationship.

**Cardinality:**

* the cardinality of ITEM is (1, 1)
* the cardinality of ORDER\_ITEM is (1, N)

**Relationship strength:** Strong

Explanation: The relationship is strong because the primary key of the ORDER\_ITEM entity contains the primary key of the ITEM entity.

**Entity Relationship Diagram (ERD):**

A screenshot of a video game

Description generated with high confidence