FRESHENS FRESH FOOD STUDIO

MGMT 582 Management of Organisational Data Group Project
Session 1- Group 8

Chethan Manjunath
Madhulika Chilla
Mahima Ashok Kriplani
Ricardo Nunez Magana
Ujjwal Garg

Table of Contents

1. CLIENT BACKGROUND AND CURRENT STATE 3

2. INTRODUCTION - PROJECT OBJECTIVE 3

3. DATASET DESCRIPTION 3

4. ENTITY RELATIONSHIP DIAGRAM 4

5. RELATIONAL DATA MODEL 5

6. NORMALISATION 5

7. DATA WAREHOUSE 8

8. DATA MODEL AND DESIGN CHOICES 8

1. CLIENT BACKGROUND AND CURRENT STATE

Located in the Cordova Recreation Centre, Freshens Fresh Food Studio is Purdue's **'healthy fresh casual'** concept offering which serves prepared-to-order food inspired by fresh ingredients. They are centred around creating a culture of health and wellness. Their signature fresh blended smoothies are extremely popular amongst Purdue students.

Freshens does not have a centralised database system to keep track of their data. Information regarding vendors, customer orders, employees and supplier orders are difficult to access. Some of their information is stored in spreadsheets, while others can only be found in physical copies. This rudimentary approach has caused Freshens to struggle to conveniently access the data they need.

2. INTRODUCTION - PROJECT OBJECTIVE

The objective of the project is to:

- Determine the flow of data within databases used by Freshens.
- Find answers to crucial business problems by querying data.
- Recommend key practices and changes for an effective system for Freshens.

3. DATASET DESCRIPTION

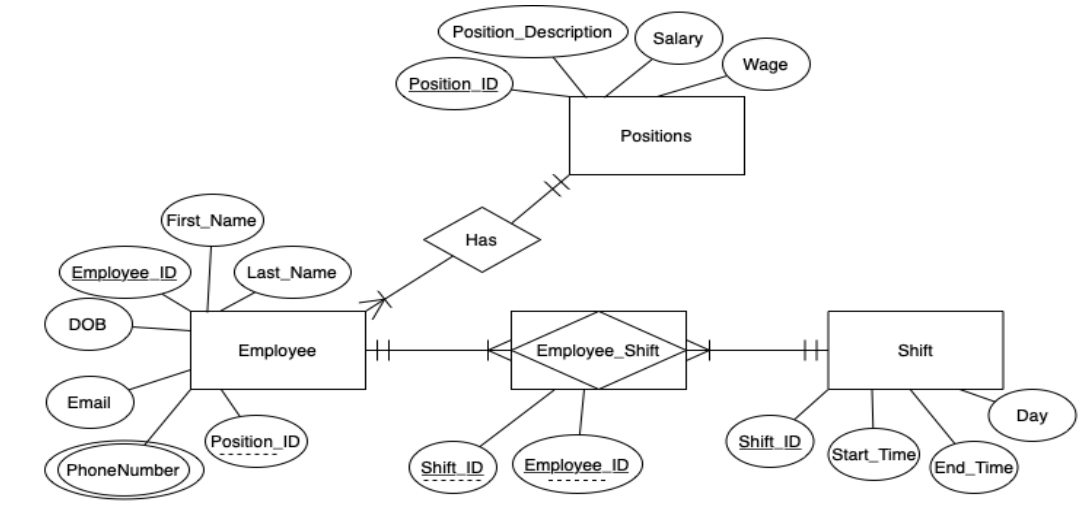
Freshens food studio has over 50 employees where an employee can have only one position, but each position can be held by multiple employees. Employees can work on multiple shifts and each shift consists of multiple employees. Every employee has a First Name, Last Name, DOB, Employee Id, Email, Position Id, Email, Phone number. Employee Id is a unique identifier. Each employee can have more than one phone number. Attributes for employee position are Position Id, Position description, Salary, and hourly wage of employees. Employee shift details are stored in a separate table which comprises Shift id, Start time, End time, Day.

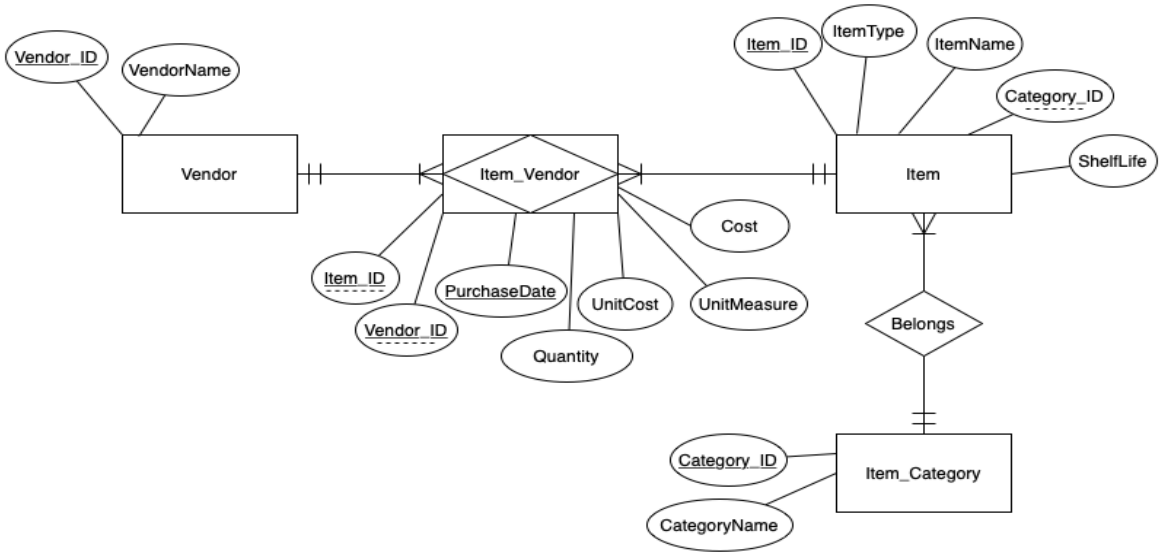
Items can be purchased from multiple vendors and a single vendor can sell multiple items. Vendor has Vendor Id and name as its attributes where vendor id is the primary key. Inventory items can be described by Item Id, Item type, Item Name, Category Id, and Shelf life where item id is a unique identifier. Items have date of purchase from the vendor, unit cost of that specific item with respect to the vendor, quantity of item ordered from that vendor, total cost, and unit measure. Every item is mapped to a particular category and each category has one or more items. Each category can be distinctively identified by Category Id.

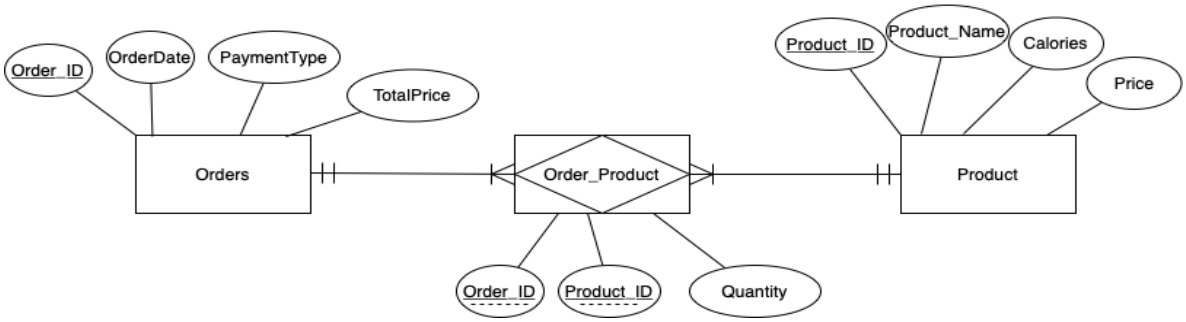
Each order consists of multiple products that are available on the menu and each product can be present in more than one order placed by the customer. Order date, Order Id, Payment type,

Total price are the attributes for orders table where order id is the primary key. Product tables can be defined by Product id, Product name, Calories, and Price with Product Id as the primary key.

4. ENTITY RELATIONSHIP DIAGRAM







5. RELATIONAL DATA MODEL

EMPLOYEE (Employee_ID, First_Name, Last_Name, DOB, Email, Position_ID)

Phone (Employee_ID, PhoneNumber)

POSITIONS (Position_ID, Position_Description, Salary, Wage)

SHIFT (Shift_ID, Start_Time, End_Time, Day)

Employee_Shift (Shift_ID, Employee_ID)

ORDER (Order_ID, OrderDate, PaymentType, TotalPrice)

OrderProduct (Order_ID, Product_ID, Quantity)

PRODUCT (Product_ID, Product_Name, Calories, Price)

VENDOR (Vendor_ID, VendorName)

ITEM (Item_ID, ItemType, ItemName, ShelfLife, Category_ID)

Iteam_Category (Category_ID, CategoryName)

Item_Vendor (Item_ID, Vendor_ID, PurchaseDate, Quantity, UnitCost, UnitMeasure, Cost)

6. NORMALISATION

We have created 8 entities, 3 associative entities and 2 relationships.

1. The Employees table was in 0 NF due to Phone_number as a multi-valued attribute.

Employee_ID	First_Name	Last_Name	DOB	Email	Phone_Number	Position_Description	Salary	Wage
EMP1	NANCY	SMITH	2/5/1964	NANCYSMITH@GMAIL.COM	8430817145	MANAGER I	80000	
EMP2	DAVID	JOHNSON	12/12/1950	DAVIDJOHNSON@GMAIL.COM	1813318082, 7652253598	MANAGER I	80000	
EMP3	ELLEN	GARCIA	5/12/1980	ELLENGARCIA@GMAIL.COM	4569694854	GENERAL I		13

It was converted to 1 NF by creating another table for Employee_ID, Phone_Number.

Employee_ID	Phone_Number
EMP1	8430817145
EMP2	1813318082
EMP2	7652253598
EMP3	4569694854

Further, it was in 2NF since there were *no partial dependencies* due to the presence of a single primary key.

The table was *not in 3 NF* due to a transitive dependency, since Employee_ID -> Position_Description -> Salary, Wage.

It was converted to 3 NF by mapping each Position_Description to a Position_ID and *creating a separate table to store the corresponding Salary and Wage*.

Position_ID	Position_Description	Salary	Wage
P1	MANAGER I	80000	
P7	GENERAL I		13

The final tables after normalisation are as below:

Employees

Employee_ID	First_Name	Last_Name	DOB	Email	Position_ID
EMP1	NANCY	SMITH	2/5/1964	NANCYSMITH@GMAIL.COM	P1
EMP2	DAVID	JOHNSON	12/12/1950	DAVIDJOHNSON@GMAIL.COM	P1
EMP3	ELLEN	GARCIA	5/12/1980	ELLENGARCIA@GMAIL.COM	P7

Phone

Employee_ID	Phone_Number
EMP1	8430817145
EMP2	1813318082
EMP2	7652253598
EMP3	4569694854

Position

Position_ID	Position_Description	Salary	Wage
P1	MANAGER I	80000	
P7	GENERAL I		13

2. The Shift table was already in 2NF. It was not in 3NF due to the presence of a transitive dependency, since Employee_ID -> Shift_ID -> Start_Time, End_Time, Day.

Employee_ID	Shift_ID	Start_Time	End_Time	Day
EMP1	S1	8AM	10AM	Mon
EMP1	S2	10AM	12PM	Mon
EMP1	S3	12PM	2PM	Mon
EMP9	S1	8AM	10AM	Mon
EMP9	S2	10AM	12PM	Mon
EMP9	S3	12PM	2PM	Mon

We converted it to 3NF by creating two separate tables.

The final tables after normalisation are as below:

Employee_ID	Shift_ID
EMP1	S1
EMP1	S2
EMP1	S3
EMP9	S1
EMP9	S2
EMP9	S3

Shift_ID	Start_Time	End_Time	Day
S1	8AM	10AM	Mon
S2	10AM	12PM	Mon
S3	12PM	2PM	Mon

3. The Orders table was *already in 1NF* due to no composite or multivalued attributes. However, it was *not in 2NF* since there were partial dependencies.

Order_ID	Product_ID	OrderDate	PaymentType	TotalPrice	Quantity	Product_Name	Calories	Price
2000	P23	5/9/2022	Cash	39.95	5	Chicken Caprese	940	7.99
2001	P31	5/29/2022	Cash	14.97	3	Bangin Berry Smoothie	375	4.99
2002	P32	4/7/2022	Cash	14.97	3	Caribbean Craze Smoothie	335	4.99

Functional dependencies:

Order_ID, Product_ID -> Quantity

Order_ID -> OrderDate, PaymentType, TotalPrice (partial dependency)

Product_ID -> Product_Name, Calories, Price (partial dependency)

Three tables were created to convert it to 2NF. Finally, *it was also in 3NF due to no transitive dependency.*

The final tables after normalisation are as below:

Order_ID	OrderDate	PaymentType	TotalPrice
2000	5/9/2022	Cash	39.95
2001	5/29/2022	Cash	14.97
2002	4/7/2022	Cash	14.97

Product_ID	Product_Name	Calories	Price
P23	Chicken Caprese	940	7.99
P31	Bangin Berry Smoothie	375	4.99
P32	Caribbean Craze Smoothie	335	4.99

Order_ID	Product_ID	Quantity
2000	P23	5
2001	P31	3
2002	P32	3

4. The Items and Vendor information table was in 1NF. However, *it was not in 2NF due to partial dependencies.*

Item_ID	Vendor_ID	PurchaseDate	Unit Measure	Quantity	UnitCost	Cost	VendorName	ItemType	ItemName	Category_ID	ShelfLife
112935	1001	1/21/2022	Bottles	6	4	24	SYSCO FOOD	Direct Sell	PEACE TEA SNOWBERRY	C1	134
150002	1001	2/18/2022	Bottles	12	3	36	SYSCO FOOD	Direct Sell	VITAMIN WATER ZERO SI	C1	134
150018	1001	2/20/2022	Packets	30	2	60	SYSCO FOOD	Raw Material	WHITE GUMMY BEAR	C9	100

Functional dependencies:

Item_ID, Vendor_ID, PurchaseDate-> Quantity, UnitCost, UnitMeasure, Cost

Vendor_ID-> VendorName (partial dependency)

Item_ID-> ItemType, ItemName, Category_ID, ShelfLife (partial dependency)

Three tables were created to convert it to 2NF. Finally, *it was also in 3NF due to no transitive dependency.*

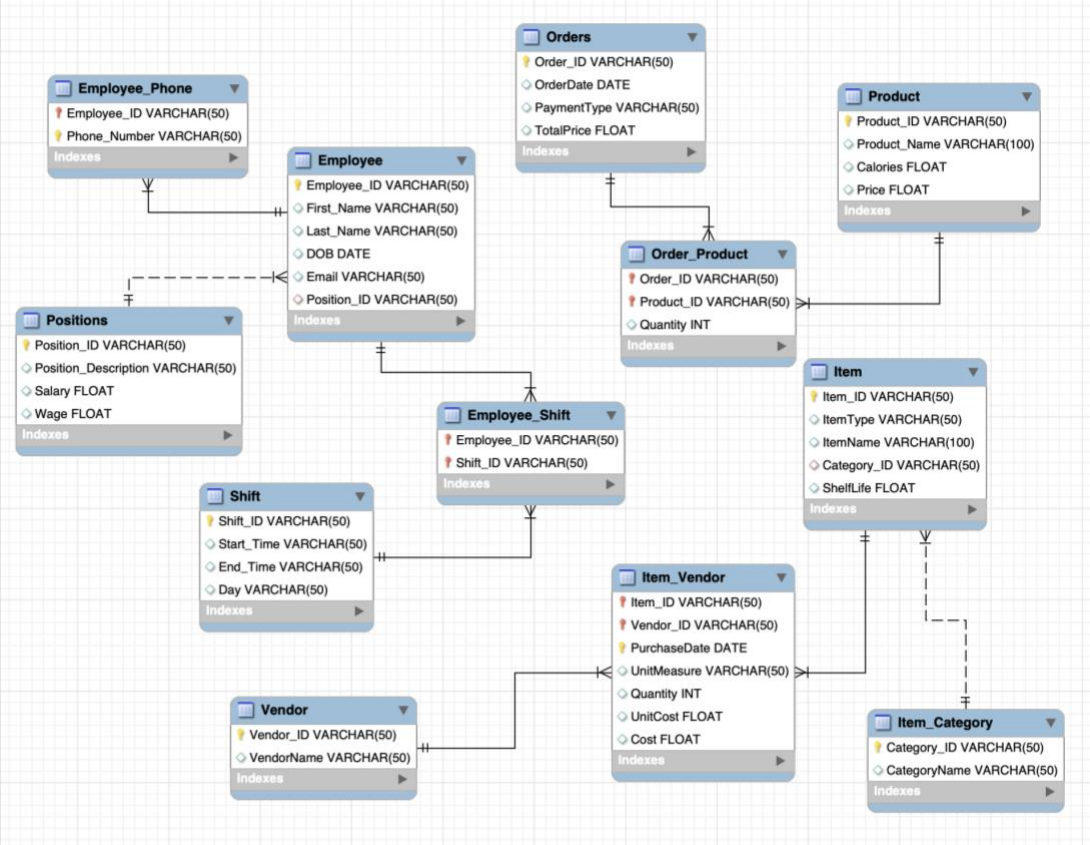
The final tables after normalisation are as below:

Item_ID	Vendor_ID	PurchaseDate	Unit Measure	Quantity	UnitCost	Cost
112935	1001	1/21/2022	Bottles	6	4	24
150002	1001	2/18/2022	Bottles	12	3	36
150018	1001	2/20/2022	Packets	30	2	60

Vendor_ID	VendorName
1001	SYSCO FOOD

Item_ID	ItemType	ItemName	Category_ID	ShelfLife
112935	Direct Sell	PEACE TEA SNOWBERRY	C1	134
150002	Direct Sell	VITAMIN WATER ZERO SHINE	C1	134
150018	Raw Material	WHITE GUMMY BEAR	C9	100

7. DATA WAREHOUSE



8. DATA MODEL AND DESIGN CHOICES

Information at Freshens Food Studio is maintained in the form of spreadsheets and physical copies. To ensure a smooth flow of information and easy accessibility to answers of crucial business problems, designing a data model for Freshens is imperative. The first step towards doing so was to get in touch with the business and figure out the business rules. The relationship cardinality was decided based on these rules. For example: one vendor supplies many items, also an item can be purchased from many vendors. This led us to create an associative entity between the two due to the presence of a many-to-many relationship. Finally, we created an entity relationship diagram after doing a thorough requirement analysis for Freshens. The ER diagram consists of information that can be broadly classified into 3 parts:

1. employee, position, shift
2. Item, vendor

3. Order, product

All the tables were normalised and finally converted into a 3NF form before adding it to the data warehouse. For example: Since Position description had a transitive dependency on employee ID, we created a Position ID corresponding to each Position description and a different table called ‘Position’ after separating it from the employee table. Finally, the schema was created in MySQL workbench.