

# AN SQL PROJECT FOR “IKEA FURNITURE”



Hej!

## TABLE OF CONTENTS

<b>1. Introduction</b>	<b>2</b>
<b>2. Data sources</b>	<b>2</b>
<b>3. Entities and Attributes</b>	<b>3</b>
<b>4. Business Rules</b>	<b>5</b>
<b>5. Enhanced Entity Relationship Diagram</b>	<b>7</b>
<b>6. The Relational Model</b>	<b>9</b>
<b>7. Data Normalization</b>	<b>12</b>
<b>8. Creation of Tables and Data</b>	<b>15</b>
<b>9. Research Questions</b>	<b>23</b>
<b>10. Conclusion</b>	<b>26</b>
<b>11. Appendix</b>	<b>27</b>

## 1. Introduction

**Name:** IKEA Furniture

**Business Sector:** Retail, E-commerce

**Description:**

IKEA is a Swedish furniture retail company that designs and sells high-quality and affordable ready-to-assemble furniture and home appliances.

The main purpose of our database is to identify the functionalities offered by the e-commerce platform of the furniture company IKEA. This involves day-to-day operations such as user, inventory, product, logistics, and payment management.

**Why did we pick this organization?**

IKEA has a very unique business model, providing a very family-friendly shopping experience, sustainable products, do-it-yourself assembly of products, and also has delicious food. The commerce that goes behind such a successful and revolutionary business model was very intriguing to us.

Ecommerce marketplaces have been on the rise around the world since the early 1900s and continue to be the most user-friendly way to shop. IKEA alone experienced a 30% hike in its sales after the introduction of its e-commerce platform.

Thus, we choose the e-commerce platform of the furniture retail giant IKEA as our database model.

## 2. Data sources

- Web scraping of the IKEA website for Products, Product\_Category, Product\_Inventory, and Discount entity information.  
Google search to obtain the Store information of various IKEA stores.
- Entity data for Customers, Payment\_Details, Orders, and Order\_Items cannot be found online. We created simulated data for these entities.

### 3. Entities and Attributes

<b>STORE</b> <ul style="list-style-type: none"> <li>• <b>Store_ID</b></li> <li>• Store_Name</li> <li>• Store_Address1</li> <li>• Store_Address2</li> <li>• City</li> <li>• State</li> <li>• Postal_code</li> <li>• Country</li> <li>• Phone_Number</li> </ul>	<b>PRODUCT</b> <ul style="list-style-type: none"> <li>• <b>Product_ID</b></li> <li>• Product_Name</li> <li>• Price</li> <li>• Size (L, B, H)</li> <li>• Sellable_Online</li> <li>• {Available_In_Colours}</li> <li>• Warranty</li> </ul>	<b>PRODUCT_CATEGORY</b> <ul style="list-style-type: none"> <li>• <b>Product_Category_ID</b></li> <li>• Product_Category_Name</li> </ul>
<b>CUSTOMER</b> <ul style="list-style-type: none"> <li>• <b>Customer_ID</b></li> <li>• First_Name</li> <li>• Last_Name</li> <li>• Email</li> <li>• Street_Address1</li> <li>• Street_Address2</li> <li>• City</li> <li>• State</li> <li>• Postal_code</li> <li>• Country</li> <li>• Phone_Number</li> </ul>	<b>ORDERS</b> <ul style="list-style-type: none"> <li>• <b>Order_ID</b></li> <li>• Order_Quantity</li> <li>• Total_Price</li> <li>• Status</li> <li>• Delivery_Option</li> </ul>	<b>ORDER_ITEMS</b> <ul style="list-style-type: none"> <li>• Order_ID</li> <li>• Product_ID</li> <li>• Quantity</li> </ul>
<b>INVENTORY</b> <ul style="list-style-type: none"> <li>• <b>Inventory_ID</b></li> <li>• Product_Count</li> <li>• Inventory_Value</li> </ul>	<b>DISCOUNT</b> <ul style="list-style-type: none"> <li>• <b>Discount_ID</b></li> <li>• Created_At</li> <li>• Name</li> <li>• Active</li> <li>• Discount_Percentage</li> <li>• Deleted_At</li> </ul>	<b>PAYMENT</b> <ul style="list-style-type: none"> <li>• <b>Payment_ID</b></li> <li>• Mode_Of_Payment</li> <li>• Status</li> </ul>

Fig 3.1 : List of Entities and Attributes

## Description of the entities and attributes:

### 1. Store: Information about each IKEA store

- a. Store\_ID : An ID to uniquely define the IKEA Store
- b. Store Name: Name of the IKEA store
- c. Store Address, City, State, Country, Postal\_Code: Location of each IKEA Store
- d. Phone Number: Phone number of the IKEA Store

### 2. Product: Information about each Product in IKEA

- a. Product\_ID: An ID to uniquely define the IKEA Product
- b. Product Name: Swedish Name of the product
- c. Price: Price of each product
- d. Size: A composite attribute containing dimensions such as: length, breadth, height of the product
- e. Sellable Online: A boolean attribute to know if the product is sellable online. It takes the value of 1 if it is sellable online and 0 if it is not sellable online
- f. Available\_In\_Colours: As we know, IKEA products are available in multiple colours for each product. Available\_In\_Colours is a multivalued attribute to show what all are the available colours for a product.
- g. Warranty: Warranty period of each product

### 3. Product Category: Each Product belongs to a category. Product Category Stores information about various product categories in IKEA

- a. Product\_Category\_ID: An ID to uniquely define each product category
- b. Product\_Category\_Name: Names of each Product category such as Bed, and Cookware.



Sofas & Seating



Beds & Mattresses



Bookcases & Shelving



TV & Media Furniture

### 4. Customer: Information about customers shopping in the e-commerce platform of IKEA

- a. Customer\_ID - An ID to uniquely identify each customer
- b. First\_Name
- c. Last\_Name
- d. Email
- e. Street\_Address1
- f. Street\_Address2
- g. City
- h. State
- i. Postal\_code
- j. Country
- k. Phone\_Number

Fig 3.2 : Featured Product categories

**5. Order: Information on customer orders**

- a. Order\_ID - An ID to uniquely identify each order
- b. Order\_Quantity - The Quantity per order
- c. Total\_Price - The sum total price of all items in the Order
- d. Status - Takes values "Paid" or "Pending"
- e. Delivery\_Option - Takes values "Pickup" or "Delivery"

**6. Inventory: Each IKEA store contains inventory of products. Information of inventory of products of each store is present in the Inventory entity**

- a. Inventory\_ID - An ID to uniquely identify each Inventory
- b. Product\_Count - Number of Products in each inventory
- c. Inventory\_Value - \$value of the inventory

**7. Discount: IKEA offers Discounts during the holidays and these discounts can be applied by the customers while purchasing a product**

- a. Discount\_ID - An ID to uniquely define each type of discount
- b. Created\_At - The timestamp when the discount was started
- c. Name
- d. Active - Takes Value "Y" or "N"
- e. Discount\_Percentage - % of discount
- f. Deleted\_At - The timestamp when discount was removed

**8. Payment: Payment details of the order made by the customer**

- a. Payment\_ID - An ID to uniquely define each Payment ID
- b. Mode\_Of\_Payment - Debit Card, Credit Card or Cash on delivery
- c. Status - check for whether the payment is completed or pending.

**9. Order Details: Each Order detail consists of products ordered in an order and their quantities.**

- a. Order\_ID - An ID for individual orders.
- b. Product\_ID - A unique ID for each product
- c. Quantity - Number of order items per Order.

## 4. Business Rules

**Rationale - Explaining the business rules**

**1. Store - Product:**

Each store must have one or many products.

Each product may be in one or many stores.

**Rationale:** Each store should have at least one product and can have a maximum of any number of products. Each product can be present in a minimum or zero stores and maximum of any number of stores

**2. Product - Product Category:**

Each product category must have one or many products.

Each product must be associated with only one product category.

**Rationale:** Each product category should have a minimum of one product and a maximum of any number of products. Each product should be associated with only one product category.

**3. Product - Discount:**

Each product may have one or many discounts.

Each Discount may be applied to one or many products.

**Rationale:** Each product can have a minimum of zero discounts and a maximum of any number of discounts. Each discount can be applied to a minimum of zero products and a maximum of any number of products

**4. Product - Inventory:**

Each product may be present in one or many inventories.

Each Inventory must have one or more Products.

**Rationale:** Each inventory must have at least one product in it. However, each product may or may not be present in one or more inventories.

**5. Customer - Order:**

Each Customer may have one or more orders.

Each order must be associated with one customer only.

**Rationale:** Each Customer may have one or more orders associated with them. However, each Order must be associated with one customer only. This is done to avoid duplication.

**6. Order - Order Items:**

Each order must have one or many order items.

Each order item must belong to only one order.

**Rationale:** Each order must have one or more order items, and each order item must belong to one orderID only.

**7. Inventory - Store:**

Each Inventory must belong to one Store.

Each Store may have many inventories.

**Rationale:** Each inventory must be associated with one Store only. However, each store may have multiple inventories.

**8. Payment - Order:**

Each Payment must be associated with one Order.

Each order may be associated with one payment.

**Rationale:** Each payment must be associated with one Order only. However, each order may be associated with one payment.

## 9. Product - Order Items:

Each product may be a part of one or more order items.

Each order item must have only one product.

**Rationale:** Each product may be associated with one or more order items. However, each order item must include one product only.

## 5. Enhanced Entity Relationship Diagram

An **Enhanced Entity Relationship (EER) Diagram (Fig 5.2)** is a type of flowchart that illustrates how “entities” relate to each other within the system. They depict the interconnectedness of entities, relationships and their attributes. We converted the above business rules into an entity-relationship diagram.

### Subtype - Supertype:

<b>Supertype:</b> Customer	<b>Subtypes:</b> Member, Guest
----------------------------	--------------------------------

There are two kinds of customers in the IKEA store:

- Members:** Are registered with IKEA and have a membership account. The registered members' profiles including their full name, address, phone number are stored in the IKEA database. Registered members have an **optional attribute** called **email** in the database.

IKEA Family members can get 15% off in the IKEA restaurant in May. Take a day off from cooking!

Fig 5.1: A promotional offer for Members

- Guest:** Are not registered with IKEA. They do not possess a membership account. No data about the Guest except email is stored in the IKEA database. **Email is not an optional attribute** for Guests.

**Constraint - Disjoint:** An instance of customer can only be a Member or a Guest

**Subtype Discriminator** - We will be adding another attribute in the Customer Entity called “Customer Type”. Customer Type is a simple attribute with alternative values to indicate the possible subtypes Member or Guest. It will take the values “M” for Member and “G” for Guest. Table 5.1 illustrates the changes made in the transformation of supertype to subtypes

Supertype	Subtypes
Customer	– Member and Guest subtype entities are added to Customer supertype entity – Customer entity has a new attribute called “CustomerType” to indicate the possible subtypes (Member/Guest)

Table 5.1: Transformation of Supertype - Subtypes



**EER Diagram for IKEA Furniture Company**

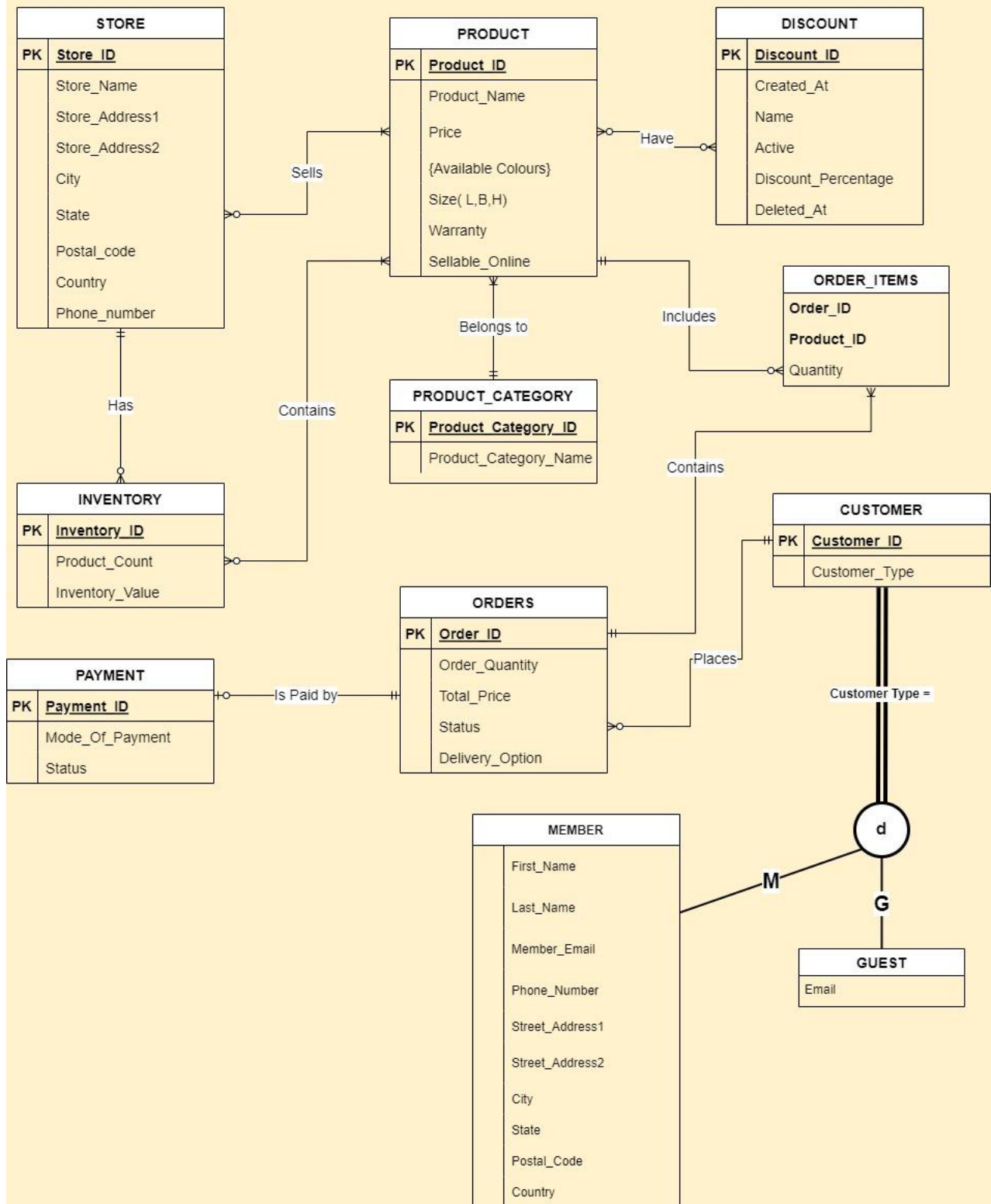


Fig: 5.2: Enhanced Entity-Relationship Diagram

## 6. The Relational Model

A relation is a named, two-dimensional table of data. A table consists of rows and columns. The requirements for a table to qualify as a relation and the entity-attributes in our data that does not qualify the requirements can be seen in Table 6.1

Requirements	Entity - Attribute not satisfying requirements
It must have a unique name	Every entity has a unique name. No changes are made.
Every attribute value must be atomic (not multivalued, not composite)	<b>Multivalued attribute:</b> Product -> {Available_In_Colours} <b>Composite attribute:</b> Product -> Size (L, B, H)
Every row must be unique (can't have two rows with exactly the same values for all their fields).	Member and Guest tables have a possibility to have the same values in all fields. Thus we can introduce MemberID, GuestID in the Member and Guest relations respectively.
Attributes (columns) in tables must have unique names.	All columns have unique names.
The order of the columns must be irrelevant.	Order of the columns is irrelevant.
The order of the rows must be irrelevant.	Order of the rows is irrelevant.

Table 6.1: Entity-attributes in the data that requires changes

Our goal now is to transform our Enhanced Entity Relationship Diagram (EERD) into relations.

### Key Fields:

**Primary Keys:** Primary keys are unique to the relation.

**Foreign keys:** Foreign keys are identifiers that enable a **dependent relation (on the many side of a relationship)** to refer to its parent relation (on the one side of the relationship)

Table 6.2 shows the Primary keys for each of the relations and foreign keys if the relation is dependent.

SNO	Entity	Primary Key	Dependent Relation?	Foreign Key
1	Store	Store_ID	No	
2	Inventory	Inventory_ID	Yes	Store_ID
3	Payment	Payment_ID	No	
4	Product	Product_ID	Yes	Product_Category_ID
5	Discount	Discount_ID	No	
6	Guest	Guest_ID	No	
7	Member	Member_ID	No	
8	Customer	Customer_ID	No	
9	Order	Order_ID	Yes	Customer_ID
10	Order Items	Order_Items_ID	Yes	Product_ID, Order_ID
11	Product Category	Product_Category_ID	No	

Table 6.2: Key fields in the relational model

#### Entities:

##### a. Simple Attribute

E-R attributes map directly onto the relation.

##### b. Composite Attribute

An attribute composed of many other attributes is called a composite attribute.

We use only their simple, component attributes in the relational model.

Table 6.3 shows the conversion of the Composite attribute size containing the attributes Length, Breath and Height into simple attributes.

Unsatisfied requirement	Before Transformation	Steps taken to transform	New attributes after Transformation
Composite attribute	Product -> Size (L, B, H)	Separate into simple attributes	Product -> Length Product -> Breadth Product -> Height

Table 6.3: Transformation to Relational Model - Entities

### c. Multivalued Attribute

- Product table has one multivalued attribute “Available\_In\_Colours”
- Each Product must be available in one or many colours
- Each colour may be applicable to one or more products
- Each order item must have only one (Product, Colour combination)
- Each (Product, Colour combination) may be applied to one or more order items
- Thus, a new relation is created called “Available\_Colours” with attributes Colour\_ID and Colour\_Name
- Multivalued attribute “Available\_In\_Colours” becomes a separate relation “Product\_Colour” with foreign key on Product\_ID and Colour\_ID.

Figure 6.1 shows the transformation of multivalued attribute Available\_In\_Colours.

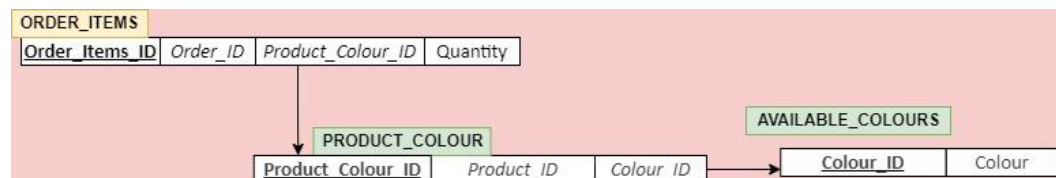


Fig 6.1: Transformation of Multivalued attribute “Available\_In\_Colours”

### Binary Relationships:

Unsatisfied requirement	Before Transformation	Steps taken to transform	New attributes after Transformation
Many-to-Many relationships	List of many - to- many relationships: a. Store-Product b. Inventory-Product c. Product-Discout	Create a new relation with the primary keys of the two entities as its primary key	New relations created: a. Store-Product(StoreID, ProductID) b. Inventory-Product(Invento ryID, ProductID) c. Product-Discout (ProductID, DiscountID)
One-to-One relationships	List-of one-to-one relationships: a. Orders-Payment	Primary key on mandatory side becomes a foreign key on optional side	Primary key Order_ID on mandatory side Orders becomes a foreign key Order_ID on optional side Payment  Payment -> OrderID
One-to-Many relationships	List of one-to-many relationships: a. Store - Inventory b. ProductCategory - Product c. Customer-Orders	Primary key on the one side becomes a foreign key on the many side	a. Inventory -> StoreID b. ProductID -> ProductCategoryID c. Orders -> CustomerID

Table 6.4: Transformation to Relational Model - Binary Relations

## Supertype/Subtype Relations:

Rule	Steps taken to transform:
One relation for supertype and for each subtype	Member and Guest subtypes are in separate relations
Supertype attributes (including identifier and subtype discriminator) go into supertype relation	Customer_ID, Customer_Type belong to Customer relation
Subtype attributes go into each subtype; primary key of supertype relation also becomes primary key of subtype relation	Member subtype attributes go into Member relation Customer_ID which is the primary key of supertype relation also becomes primary key Member_ID of Member relation  Guest subtype attributes go into Guest relation Customer_ID which is the primary key of supertype relation also becomes primary key Guest_ID of Guest relation
1:1 relationship established between supertype and each subtype, with supertype as primary table	1:1 relationship established between supertype Customer and each subtype(Member and Guest), with supertype Customer as primary table

Table 6.5: Transformation of Supertype/Subtype relations.

## 7. Data Normalization

Data normalization is a tool used to validate and improve a logical design so that it satisfies certain constraints that avoid unnecessary duplication of data. It is the process of decomposing relations with anomalies to produce smaller, well-structured relations. Well-structured relations contains minimal data redundancy and allows users to insert, delete, and update rows without causing data inconsistencies.

Our Goal is to avoid anomalies such as:

➤ **Insertion Anomaly:** Adding new rows forces user to create duplicate data.

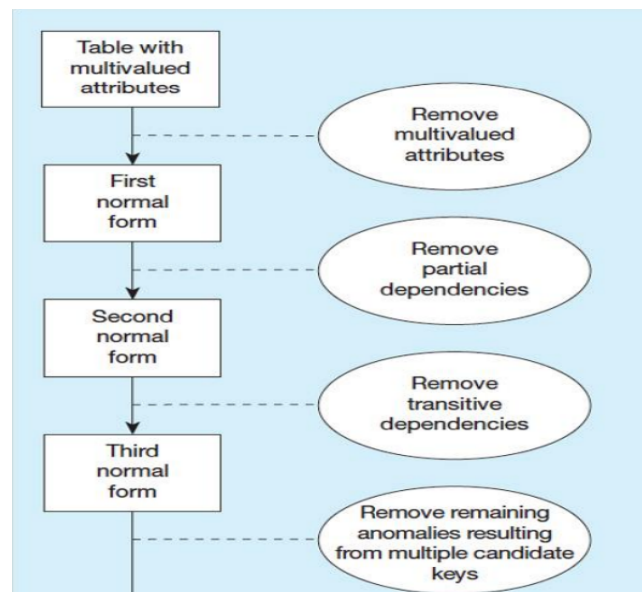


Fig 7.1: Steps in Normalization

➤ **Deletion Anomaly:** Deleting rows may cause a loss of data that would be needed for other future rows

➤ **Modification Anomaly:** Changing data in a row forces changes to other rows because of duplication

Data Normalization allows to solve table anomalies. Fig 7.1 shows the various steps in the normalization process.

#### First Normal Form:

Steps	Changes Made (if any)
No multivalued attributes	Done
Every attribute value is atomic	Done

#### Dependencies:

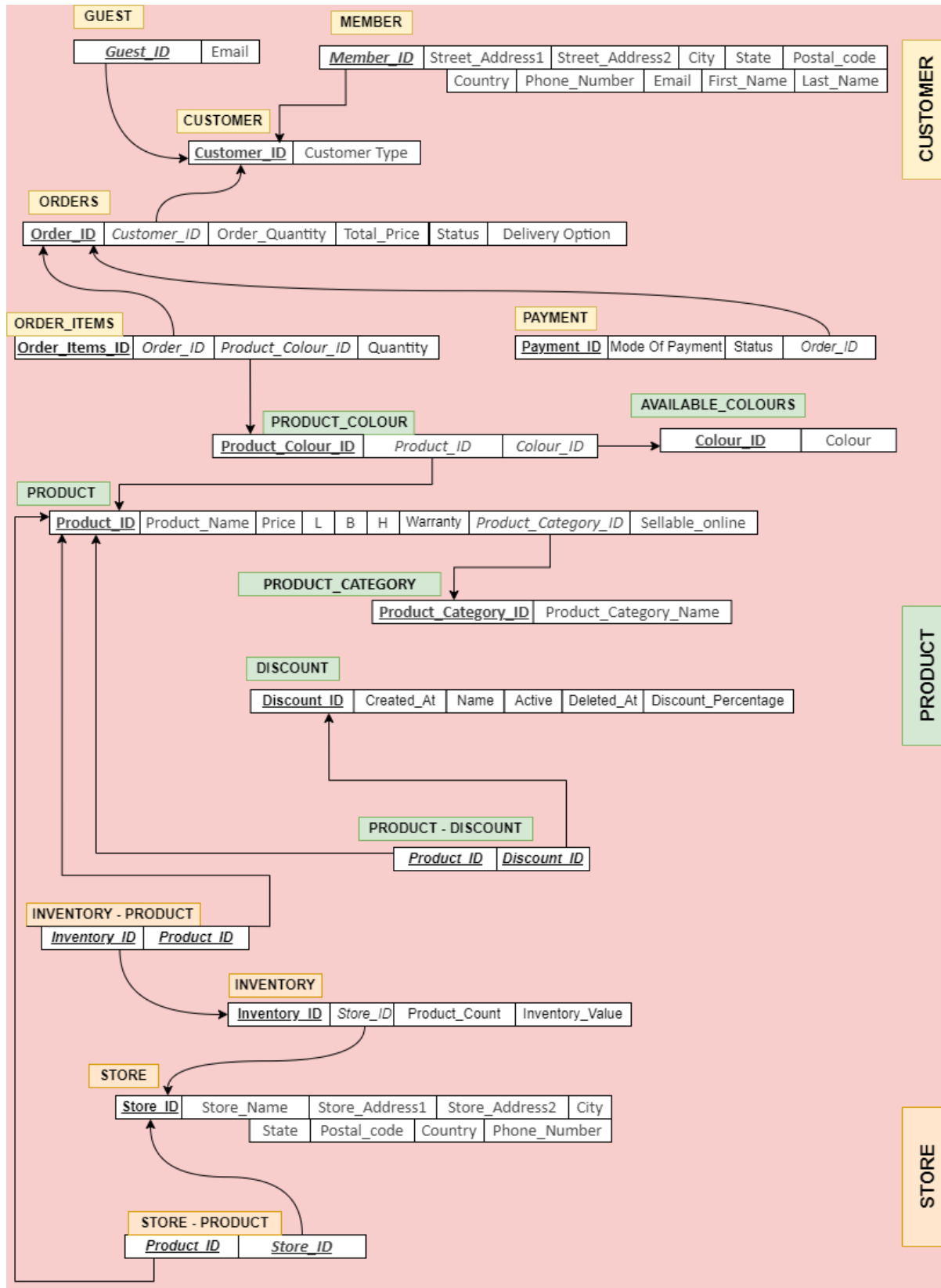
<b>Full Dependency:</b> A nonkey attribute is functionally dependent on all the parts of the candidate key.	All attributes are fully dependent on all parts of their entity candidate key
<b>Partial Dependency:</b> A nonkey attribute is functionally dependent on some, but not all, of the parts of the candidate key.	No Partial dependencies
<b>Transitive Dependency :</b> A nonkey attribute is functionally dependent on attribute(s) which are not part of the candidate key	No Transitive Dependencies

#### Second Normal Form:

Steps	Changes Made (if any)
Be in 1NF	Done
No Partial Dependencies	Done

#### Third Normal Form:

Steps	Changes Made (if any)
Be in 2NF	Done
No Transitive Dependencies	Done



Fig

7.2: 3NF Relational model

## 8. Creation of Tables and Data

### Schema Creation:

#### #Schema Related

```
CREATE SCHEMA IF NOT EXISTS ikea;  
USE ikea;
```

### Creation of tables:

#### #Discount

```
CREATE TABLE Discount  
( Discount_ID VARCHAR(10) NOT NULL,  
  Created_At datetime,  
  Name VARCHAR(50),  
  Active char CHECK(Active IN ('Y' , 'N')),  
  Discount_Percentage INT,  
  Deleted_At datetime,  
  
  CONSTRAINT Discount_PK PRIMARY KEY (Discount_ID) );
```

#### #Customer

```
CREATE TABLE Customer  
(Customer_ID VARCHAR(10) NOT NULL,  
  Type_Of_Customer VARCHAR(10) CHECK(Type_Of_Customer IN ('Member' , 'Guest')),  
  
  CONSTRAINT Customer_PK PRIMARY KEY(Customer_ID) );
```

#### #Available Colours

```
CREATE TABLE Available_Colours  
(Colour_ID VARCHAR(10) NOT NULL,  
  Colour VARCHAR(20),  
  
  CONSTRAINT Colour_PK PRIMARY KEY(Colour_ID));
```

#### #Guest

```
CREATE TABLE Guest  
(Guest_ID VARCHAR(10) NOT NULL,  
  Email VARCHAR(20) NOT NULL,  
  
  CONSTRAINT Guest_PK PRIMARY KEY(Guest_ID),  
  CONSTRAINT Guest_FK FOREIGN KEY(Guest_ID) REFERENCES Customer (Customer_ID));
```



#### #Member

```
CREATE TABLE Member
(Member_ID VARCHAR(10) NOT NULL,
First_Name VARCHAR(20),
Last_Name VARCHAR(20),
Email VARCHAR (20),
Street_Address1 VARCHAR (50),
Street_Address2 VARCHAR (50),
City VARCHAR(20),
State VARCHAR(20),
Postal_Code INT,
Country VARCHAR(20),
Phone_Number VARCHAR(15),

CONSTRAINT Member_PK PRIMARY KEY(Member_ID),
CONSTRAINT Member_FK FOREIGN KEY(Member_ID) REFERENCES Customer (Customer_ID));
```

#### #Store

```
CREATE TABLE Store
(Store_ID VARCHAR(5) NOT NULL,
Store_Name VARCHAR(25) ,
Store_Address1 VARCHAR(30) ,
Store_Address2 VARCHAR(30) ,
City VARCHAR(20) ,
State CHAR(2) ,
Postal_Code VARCHAR(10) ,
Country VARCHAR(10) ,
Phone_Number VARCHAR(15) ,

CONSTRAINT Store_PK PRIMARY KEY (Store_ID));
```

#### #Inventory

```
CREATE TABLE Inventory
( Inventory_ID VARCHAR(10) NOT NULL,
Store_ID VARCHAR(5) NOT NULL,
Product_Count INT,
Inventory_Value INT,

CONSTRAINT Inventory_PK PRIMARY KEY (Inventory_ID),
CONSTRAINT Inventory_FK FOREIGN KEY(Store_ID) REFERENCES Store (Store_ID) );
```

#### #Order

```
CREATE TABLE Orders
(Order_ID INT NOT NULL,
Customer_ID VARCHAR(10) NOT NULL,
Order_Quantity INT,
Total_Price INT,
```

```

Status VARCHAR(20),
Delivery_Option VARCHAR(20),
CONSTRAINT Order_PK PRIMARY KEY(Order_ID),
CONSTRAINT Order_FK1 FOREIGN KEY(Customer_ID) REFERENCES Customer(Customer_ID));

#Payment
CREATE TABLE Payment
(Payment_ID VARCHAR(10) NOT NULL,
Mode_Of_Payment VARCHAR(15),
Status VARCHAR(10),
Order_ID INT NOT NULL,

CONSTRAINT Payment_PK PRIMARY KEY (Payment_ID),
CONSTRAINT Payment_FK FOREIGN KEY(Order_ID) REFERENCES Orders (Order_Id) );

#Product Category
CREATE TABLE Product_Category
( Product_Category_ID VARCHAR(6) NOT NULL,
Product_Category_Name VARCHAR(20) NOT NULL,
CONSTRAINT Product_CategoryPK PRIMARY KEY (Product_Category_ID) );

#Product
CREATE TABLE Product
( Product_ID VARCHAR(6) NOT NULL,
Product_Name VARCHAR(25),
Price DECIMAL(6,2),
Warranty VARCHAR(20),
Sellable_Online BOOLEAN,
L INT,
B INT,
H INT,
Product_Category_ID VARCHAR(6),

CONSTRAINT Product_PK PRIMARY KEY (Product_ID),
CONSTRAINT Product_FK FOREIGN KEY (Product_Category_ID) REFERENCES
Product_Category(Product_Category_ID) );

#Inventory Product
CREATE TABLE Inventory_Product
(Inventory_ID VARCHAR(10) NOT NULL,
Product_ID VARCHAR(6) NOT NULL,
CONSTRAINT Inventory_Product_PK PRIMARY KEY (Inventory_ID, Product_ID),
CONSTRAINT Inventory_Product_FK1 FOREIGN KEY (Inventory_ID) REFERENCES
Inventory(Inventory_ID),
CONSTRAINT Inventory_Product_FK2 FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID));

#Store Product
CREATE TABLE Store_Product

```

```
(Store_ID          VARCHAR(5)  NOT NULL,
Product_ID         VARCHAR(6)  NOT NULL,
CONSTRAINT Store_Product_PK PRIMARY KEY (Store_ID, Product_ID),
CONSTRAINT Store_Product_FK1 FOREIGN KEY (Store_ID) REFERENCES Store(Store_ID),
CONSTRAINT Store_Product_FK2 FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID));
```

#### #Product Colour

CREATE TABLE **Product\_Colour**

```
(Product_Colour_ID VARCHAR(6)  NOT NULL,
Product_ID          VARCHAR(12) NOT NULL,
Colour_ID           VARCHAR(10) NOT NULL,
CONSTRAINT Product_Colour_PK PRIMARY KEY (Product_Colour_ID),
CONSTRAINT Product_Colour_FK1 FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID),
CONSTRAINT Product_Colour_FK2 FOREIGN KEY (Colour_ID) REFERENCES
Available_Colours(Colour_ID));
```

#### #Order Items

CREATE TABLE **Order\_Items**

```
(Order_Items_ID INT NOT NULL,
Order_ID INT,
Product_Colour_ID VARCHAR(20),
Quantity INT,
```

```
CONSTRAINT Order_Items_PK PRIMARY KEY (Order_Items_ID),
CONSTRAINT Order_Items_FK FOREIGN KEY(Order_ID) REFERENCES Orders (Order_ID),
CONSTRAINT Order_Items_FK2 FOREIGN KEY(Product_Colour_ID) REFERENCES Product_Colour
(Product_Colour_ID));
```

#### #Product Discount

CREATE TABLE **Product\_Discount**

```
(Product_ID          VARCHAR(12) NOT NULL,
Discount_ID          VARCHAR(12) NOT NULL,
CONSTRAINT Product_Discount_PK PRIMARY KEY ( Product_ID, Discount_ID),
CONSTRAINT Product_Discount_FK1 FOREIGN KEY (Product_ID) REFERENCES Product (Product_ID),
CONSTRAINT Product_Discountt_FK2 FOREIGN KEY (Discount_ID) REFERENCES Discount
(Discount_ID));
```

#### Insert Queries:

##### #Discount Insert Statements

```
INSERT INTO Discount ( Discount_ID, Created_At, Name, Active, Discount_Percentage, Deleted_At )
VALUES ('D001', '2021-07-01 1:00:00', 'Fourth of July', 'N', 20, '2021-07-21 10:00:00'),
('D002', '2021-11-20 10:00:00', 'Black Friday', 'N', 15, '2021-11-30 10:00:00'),
('D003', '2022-05-01 9:00:00', 'Mothers Day', 'Y', 10, NULL);
```

##### #Customer Insert Statements

```
INSERT INTO Customer (Customer_ID, Type_Of_Customer) VALUES
('M_009', 'Member'), ('G_0081', 'Guest'), ('M_102', 'Member'), ('G_010', 'Guest'),
```

```
('G_128' , 'Guest'), ('M_087' , 'Member');
```

#### #Guest Insert Statements

```
INSERT INTO Guest (Guest_ID, Email) VALUES
```

```
('G_0081' , 'sjaiswal@gmail.com'), ('G_010' , 'achheda@gmail.com'), ('G_128' , 'bhotta@yahoo.com');
```

#### #Member Insert Statements

```
INSERT INTO Member (Member_ID, First_Name, Last_Name, Email, Street_Address1, Street_Address2, City, State, Postal_code, Country, Phone_Number) VALUES
```

```
('M_009', 'Chandni', 'Shah', 'cshah@sfsu.edu', '490 W Capitol Ave', 'Apt 245', 'San Jose', 'CA', '95435', 'USA', '6697023451'),
```

```
('M_102', 'Uma', 'Rajagopalan', 'uraj@yahoo.com', '978 Crescent Village', 'Unit 567', 'New York City', 'NY', '76540', 'USA', '9796532183'),
```

```
('M_087', 'Sudharsanan', 'Sadagopan', NULL, '782 First Streeet', 'Suite 750', 'Philadelphia', 'PA', '71270', 'USA', '4097621345');
```

#### #Available Colour Insert Statements

```
INSERT INTO Available_Colours (Colour_ID, Colour) VALUES
```

```
('C1' , 'White'), ('C2' , 'Red'), ('C3' , 'Blue'), ('C4' , 'Black');
```

#### #Order Insert Statements

```
INSERT INTO Orders (Order_ID, Customer_ID, Order_Quantity, Total_Price, Status, Delivery_Option) VALUES (10107, 'M_009', 3, 95.7, 'Paid', 'Pickup'),
```

```
(10121, 'G_0081' , 4, 151.35, 'Pending', 'Delivery'),
```

```
(10134, 'M_102' , 1, 94.74, 'Paid', 'Delivery'),
```

```
(10145, 'G_010' , 5, 53.26, 'Paid', 'Pickup'),
```

```
(10159, 'G_128' , 9, 100, 'Pending', 'Delivery'),
```

```
(10168, 'M_087' , 6, 196.66, 'Paid', 'Pickup'),
```

```
(10170, 'M_009', 4, 53, 'Paid', 'Delivery'),
```

```
('10172', 'M_102', 2, 34.99, 'Paid', 'Pickup');
```

#### #Payment Insert Statements

```
INSERT INTO Payment (Payment_ID, Mode_Of_Payment, Status, Order_ID) VALUES
```

```
('P001', 'CreditCard', 'Completed', 10107), ('P002', 'DebitCard', 'Completed', 10121),
```

```
('P003', 'CreditCard', 'Completed', 10134), ('P004', 'CashOnDelivery', 'Pending', 10121);
```

#### #Store Insert Statements

```
INSERT INTO Store (Store_ID, Store_Name, Store_Address1, Store_Address2, City, State, Postal_Code, Country ,Phone_number ) VALUES
```

```
('S01', 'Ikea San Francisco', '4400 Shellmound St' , 'Emeryville' , 'San Francisco', 'CA' , 94608, 'USA', '(657) 888-4531'),
```

```
('S02' , 'Ikea Los Angeles', '20700 S Avalon Blvd' , NULL, 'Carson', 'CA' , 90746, 'USA', '(657) 888-4532'),
```

```
('S03', 'Ikea Palo Alto', '1700 E Bayshore Rd', 'East Palo Alto', 'Palo Alto', 'CA', 94303, 'USA', '(657) 888-4533'),
```

```
('S04' , 'Ikea Seattle' , '601 Short West' , '41st St' , 'Seattle', 'WA' , 98057, 'USA' , '(657) 888-4534'),
```

```
('S05' , 'Ikea Tampa', '1103 Norstar', '22nd St', 'Tampa', 'FL', 33605, 'USA' , '(657) 888-4535');
```

#### #Product Category Insert Statements

```
INSERT INTO Product_Category (Product_Category_ID, Product_Category_Name)
VALUES('PC10','Bed'), ('PC20','Outdoor Furniture'), ('PC30','Cookware');
```

#### #Inventory Insert Statements

```
INSERT INTO Inventory (Inventory_ID, Store_ID, Product_Count, Inventory_Value) VALUES
('I001','S01',200,20000), ('I002','S02',100,10000), ('I003','S03',4000,400000);
```

#### #Product Insert Statements

```
INSERT INTO Product(Product_ID, Product_Name, Price, Warranty,
Sellable_Online, L,B,H, Product_Category_ID) VALUES
('P401' , 'FREKVEN', 265 , '2 months' , TRUE, 51, 25, 51, 'PC20'),
('P402' , 'NORDVIKEN', 995, '3 months', FALSE, 10, 25, 70 , 'PC20'),
('P403' , 'PLATSA' , 1105, '6 months', TRUE, 140, 244, 163, 'PC10'),
('P404', 'INGOLF', 345 , '2 months', TRUE, 109, 87, 76, 'PC10'),
('P405', 'TOMMARYD', 754, '3 months', TRUE, 34, 26, 32, 'PC30'),
('P406', 'JANINGE', 19, '0 months', FALSE, 10, 5, 10, 'PC30');
```

#### #Product-Color Insert Statements

```
INSERT INTO Product_Colour(Product_Colour_ID, Product_ID, Colour_ID) VALUES
('PC1', 'P401', 'C1'), ('PC2', 'P401', 'C2'), ('PC3', 'P402', 'C3'), ('PC4', 'P402', 'C4'), ('PC5', 'P403', 'C1'),
('PC6', 'P403', 'C2'), ('PC7', 'P404', 'C3'), ('PC8', 'P404', 'C4'), ('PC9', 'P405', 'C1'), ('PC10', 'P405', 'C2'),
('PC11', 'P406', 'C3'), ('PC12', 'P406', 'C4'), ('PC13', 'P406', 'C1');
```

#### #Order Items Insert Statements

```
INSERT INTO Order_Items (Order_Items_ID, Order_ID, Product_Colour_ID, Quantity) VALUES
(101, 10107, 'PC1', 1), (102, 10121, 'PC4', 4), (103, 10107, 'PC6', 1), (104, 10145, 'PC7', 5),
(105, 10159, 'PC10', 9), (106, 10107, 'PC13', 1);
```

#### #Store\_Product Insert Statements

```
INSERT INTO Store_Product (Store_ID, Product_ID) VALUES
('S01', 'P401'), ('S01', 'P402'), ('S05', 'P403'), ('S04', 'P404'), ('S01', 'P405'), ('S01', 'P406'), ('S02', 'P404'),
('S02', 'P405'), ('S02', 'P406'), ('S03', 'P403'), ('S03', 'P404'), ('S03', 'P405');
```

#### #Inventory-Product Insert Statements

```
INSERT INTO Inventory_Product( Inventory_ID, Product_ID) VALUES
('I001', 'P401'), ('I003', 'P402'), ('I001', 'P403'), ('I002', 'P405');
```

#### #Product-Discount Insert Statements

```
INSERT INTO Product_Discount ( Product_ID, Discount_ID) VALUES
('P401', 'D001'), ('P401', 'D002'), ('P403', 'D003'), ('P404', 'D003'), ('P403', 'D002');
```

## Data Screenshots:

**select \* from Discount;**

Discount_ID	Created_At	Name	Active	Discount_Percentage	Deleted_At
D001	2021-07-01 01:00:00	Fourth of July	N	20	2021-07-21 10:00:00
D002	2021-11-20 10:00:00	Black Friday	N	15	2021-11-30 10:00:00
D003	2022-05-01 09:00:00	Mothers Day	Y	10	

**select \* from Inventory;**

Inventory_ID	Store_ID	Product_Count	Inventory_Value
I001	S01	200	20000
I002	S02	100	10000
I003	S03	4000	400000

**select \* from Guest;**

Guest_ID	Email
G_0081	sjaiswal@gmail.com
G_010	achheda@gmail.com
G_128	bhotta@yahoo.com

**select \* from Customer;**

Customer_ID	Type_Of_Customer
G_0081	Guest
G_010	Guest
G_128	Guest
M_009	Member
M_087	Member
M_102	Member

**select \* from Member;**

Member_ID	First_Name	Last_Name	Email	Street_Address1	Street_Address2	City	State	Postal_Code	Country	Phone_Number
M_009	Chandni	Shah	cshah@sfsu.edu	490 W Capitol Ave	Apt 245	San Jose	CA	95435	USA	6697023451
M_087	Sudharsanan	Sadagopan		782 First Street	Suite 750	Philadelphia	PA	71270	USA	4097621345
M_102	Uma	Rajagopalan	uraj@yahoo.com	978 Crescent Village	Unit 567	New York City	NY	76540	USA	9796532183

**select \* from Available\_Colours ;**

Colour_ID	Colour
C1	White
C2	Red
C3	Blue
C4	Black

**select \* from Orders;**

Order_ID	Customer_ID	Order_Quantity	Total_Price	Status	Delivery_Option
10107	M_009	3	96	Paid	Pickup
10121	G_0081	4	151	Pending	Delivery
10134	M_102	1	95	Paid	Delivery
10145	G_010	5	53	Paid	Pickup
10159	G_128	9	100	Pending	Delivery
10168	M_087	6	197	Paid	Pickup
10170	M_009	4	53	Paid	Delivery
10172	M_102	2	35	Paid	Pickup

**select \* from Inventory\_product;**

Inventory_ID	Product_ID
I001	P401
I003	P402
I001	P403
I002	P405

**select \* from Order\_items;**

Order_Items_ID	Order_ID	Product_Colour_ID	Quantity
101	10107	PC1	1
102	10121	PC4	4
103	10107	PC6	1
104	10145	PC7	5
105	10159	PC10	9
106	10107	PC13	1

**select \* from Payment;**

Payment_ID	Mode_Of_Payment	Status	Order_ID
P001	CreditCard	Completed	10107
P002	DebitCard	Completed	10121
P003	CreditCard	Completed	10134
P004	CashOnDelivery	Pending	10121

**select \* from Product\_category;**

Product_Category_ID	Product_Category_Name
PC10	Bed
PC20	Outdoor Furniture
PC30	Cookware

**select \* from Product;**

Product_ID	Product_Name	Price	Warranty	Sellable_Online	L	B	H	Product_Category_ID
P401	FREKVENS	265.00	2 months	1	51	25	51	PC20
P402	NORDVIKEN	995.00	3 months	0	10	25	70	PC20
P403	PLATSA	1105.00	6 months	1	140	244	163	PC10
P404	INGOLF	345.00	2 months	1	109	87	76	PC10
P405	TOMMARYD	754.00	3 months	1	34	26	32	PC30
P406	JANINGE	19.00	0 months	0	10	5	10	PC30

**select \* from Store;**

Store_ID	Store_Name	Store_Address1	Store_Address2	City	State	Postal_Code	Country	Phone_Number
S01	Ikea San Francisco	4400 Shellmound St	Emeryville	San Francisco	CA	94608	USA	(657) 888-4531
S02	Ikea Los Angeles	20700 S Avalon Blvd	NULL	Carson	CA	90746	USA	(657) 888-4532
S03	Ikea Palo Alto	1700 E Bayshore Rd	East Palo Alto	Palo Alto	CA	94303	USA	(657) 888-4533
S04	Ikea Seattle	601 Short West	41st St	Seattle	WA	98057	USA	(657) 888-4534
S05	Ikea Tampa	1103 Norstar	22nd St	Tampa	FL	33605	USA	(657) 888-4535

**select \* from Product\_colour;**

Product_Colour_ID	Product_ID	Colour_ID
PC11	P406	C3
PC12	P406	C4
PC13	P406	C1
PC2	P401	C2
PC3	P402	C3
PC4	P402	C4
PC5	P403	C1
PC6	P403	C2
PC7	P404	C3
PC8	P404	C4
PC9	P405	C1

**select \* from Product\_discount;**

Product_ID	Discount_ID
P401	D001
P401	D002
P403	D002
P403	D003
P404	D003

**select \* from Store\_product;**

Store_ID	Product_ID
S01	P401
S01	P402
S03	P403
S05	P403

Table 8.1: Data screenshots

## 9. Research Questions

### Q1 . Customers Who Purchased Products worth more than \$100

```
SELECT c.Customer_ID,  
c.Type_Of_Customer,  
g.email as Email,  
#mem.email as Member_Email,  
o.Total_Price as Order_Total  
FROM (Customer c, Orders o)  
INNER JOIN  
Guest AS g  
ON c.Customer_ID = g.Guest_ID  
  
WHERE c.Customer_ID = o.Customer_ID  
AND o.Total_Price >= 100  
  
UNION  
  
SELECT c.Customer_ID,  
c.Type_Of_Customer,  
#g.email as Email,  
mem.email as Member_Email,  
o.Total_Price as Order_Total  
FROM (Customer c, Orders o)  
INNER JOIN  
Member AS mem  
ON c.Customer_ID = mem.Member_ID  
WHERE c.Customer_ID = o.Customer_ID  
AND o.Total_Price >= 100
```

**Purpose of the question:** The objective behind this question was to find out Ikea's most esteemed customers who have made maximum sales so far. Through this question, we will also be able to segregate what type of customers (whether Guests or Members) have made the highest sale so far.

**Output :**

Customer_ID	Type_Of_Customer	Email	Order_Total
G_0081	Guest	sjaiswal@gmail.com	151
G_128	Guest	bhotta@yahoo.com	100
M_087	Member	NULL	197

**Findings:** 3 customers made orders more than \$100.

### Q2 : Type of Customer Generating most revenue ?

```
SELECT COUNT(Type_Of_Customer) As 'Number of Customers' ,  
Type_Of_Customer , SUM(Total_Price) As 'Sum of Total Sales'  
FROM Orders  
JOIN Customer  
ON Orders.Customer_ID = Customer.Customer_ID  
GROUP BY Type_Of_Customer  
ORDER BY Total_Price;
```

**Purpose of the question**

This question is in continuation to Q1, as now we aim to find out what type of customer is generating the highest revenue for IKEA.

**Output :**

Number of Customers	Type_Of_Customer	Sum of Total Sales
5	Member	476
3	Guest	304

**Findings:** From the output, it can be interpreted that most revenue is generated by Members of Ikea.



**Q3: What is the most commonly used payment method?**

```
SELECT COUNT(Mode_Of_Payment) As Count_Mode_Of_Payment,  
Mode_Of_Payment  
FROM Payment  
GROUP BY Mode_Of_Payment  
ORDER BY COUNT(Mode_Of_Payment) DESC  
LIMIT 1 ;
```

**Purpose of the question :**

This question will help in knowing what is the preferred mode of Payment so that Ikea could focus its marketing strategy and give additional discounts on that mode of payment to attract more sales.

**Output :**

Count_Mode_Of_Payment	Mode_Of_Payment
2	CreditCard

**Findings:** Credit card is the most used payment method.

**Q4: Give the details of the most sold product**

```
CREATE VIEW Sold_Products AS  
SELECT oi.Quantity,p.Product_Name, p.Product_ID,  
pcol.Product_Colour_ID,pc.Product_Category_Name,ac.Colour  
FROM Product p, Product_Category pc, Order_Items oi,  
Available_Colours ac,Product_Colour pcol  
WHERE oi.Product_Colour_ID = pcol.Product_Colour_ID  
AND pcol.Product_ID = p.Product_ID  
AND pcol.Colour_ID = ac.Colour_ID  
AND pc.Product_Category_ID = p.Product_Category_ID;  
  
select * from Sold_Products  
where Quantity = (select max(Quantity) from Sold_Products);
```

**Purpose of the question :**

To elicit information about the features of the most sold product and the product category to which it belongs.

**Output :**

Quantity	Product_Name	Product_ID	Product_Colour_ID	Product_Category_Name	Colour
9	TOMMARYD	P405	PC10	Cookware	Red

**Findings:** TOMMARYD is the most sold Product which falls under the Cookware category and is red.

#### Q5: Availability of Inventory for the most sold Product

```
SELECT s.Store_Name, i.Product_Count, p.Product_ID, p.Product_Name
FROM Inventory i, Product p, Inventory_Product ip, Store s
WHERE ip.Inventory_ID = i.Inventory_ID
AND ip.Product_ID = p.Product_ID
AND i.Store_ID = s.Store_ID
AND p.Product_ID IN (select Product_ID from Sold_Products
where Quantity = (select max(Quantity) from Sold_Products));
```

##### Purpose of the question

To ascertain whether Ikea had adequate inventory for the most sold product.

\*For this question we used the View (Sold\_products) created in Q4.

##### Output :

Store_Name	Product_Count	Product_ID	Product_Name
Ikea Los Angeles	100	P405	TOMMARYD

**Findings:** Ikea Los Angeles has inventory for TOMMARYD which is the most sold product

#### Q6: Products having discounts greater than 10% and their before and after prices

```
SELECT p.Product_Name, d.Discount_Percentage as 'Discount %',
Product_Category_Name as Product_Type, p.price as Original_Price,
ROUND((p.price - (p.price*d.Discount_Percentage/100)),2) as Discounted_Price
FROM Product p, Discount d, Product_Discount pd, Product_Category pc
WHERE p.Product_ID = pd.Product_ID
AND d.Discount_ID = pd.Discount_ID
AND p.Product_Category_ID = pc.Product_Category_ID
AND p.price IS NOT NULL
AND d.Discount_Percentage > 10
ORDER BY d.Discount_Percentage DESC;
```

##### Purpose of the question:

To display the product that has an ongoing discount of more than 10% and provide the before and after the price of the products.

##### Output :

Product_Name	Discount %	Product_Type	Original_Price	Discounted_Price
FREKVENS	20	Outdoor Furniture	265.00	212.00
FREKVENS	15	Outdoor Furniture	265.00	225.25
PLATSA	15	Bed	1105.00	939.25

**Findings:** Highest discount of 20% is for the product FREKVENS which is an Outdoor Furniture.

## 10. Conclusion

### Benefits:

The project was a great learning experience as it not only prepared us for the real world but also gave us a sneak peek at the database structure of IKEA. Through this project, we were efficiently able to put our knowledge of Data Management into action. The whole experience of building and designing a database with real-life applicability, not only refined our conceptual understanding but also boosted our problem-solving skills. This project taught us

### Eventual Challenges:

- As we are aware that a Retail group like Ikea has many entities in its database, the foremost and the biggest challenge that we faced was to decide when to draw a line that the number of proposed entities was sufficient for the project. We had various entities in our mind but to prioritize the top 9 entities was a challenge that we faced.
- Another challenge was to decide how each entity will interact with the other. Though, the business rules helped in defining the relationship among the entities.
- Due to the complex nature of the business, coming up with the entity relationships and working on the normalization of the various attributes was a big challenge.

### Future Scope:

There is a humongous scope for expanding this project :

- Plan to increase the number of entities and attributes associated with each of them.
- Redefine business rules for newly-added entities.
- Plan to provide discounts on the Orders instead of giving discounts only to products.
- Plan to incorporate the details and information about the Employees of Ikea.
- Plan on expanding the database for the products returned to Ikea.

## 11. Appendix

### List of Figures

SNO	Figure Name	Page Number
3.1	List of Entities and Attributes	3
3.2	Featured Product Categories	4
5.1	A promotional offer for Members	7
5.2	Enhanced Entity-Relationship Diagram	8
6.1	Transformation of Multivalued attribute "Available_In_Colours"	11
7.1	Steps in Normalization	12
7.2	3NF Relational model	14

### List of Tables

SNO	Table Name	Page Number
5.1	Transformation of Subtype-Supertype	7
6.1	Entity-attributes in the data that requires changes	9
6.2	Key fields in the relational model	10
6.3	Transformation to Relational Model - Entities	10
6.4	Transformation to Relational Model - Binary Relations	11
6.5	Transformation of Supertype/Subtype relations.	12
8.1	Data screenshots	21-22