

Table of Contents

Title	1
Author(s)	2
Database Design Assignment Part A	3
A.1. Case Study Overview	3
A.2. Database Functions	2
A.3. Data Requirements	2
A.4. Business Rules.....	2

Designing a Database to Support Online Gaming

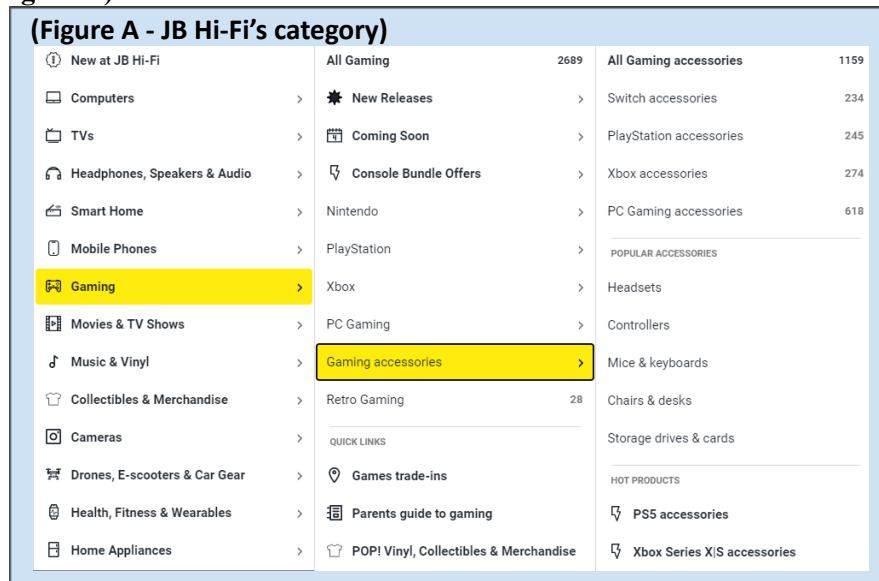
Author(s)

- 1. Student ID and Name: 13922277 Sarrah Barodawala**
- 2. Student ID and Name: 14284917 Jeffrey Cheng**
- 3. Student ID and Name: 13890280 Abina Kirubaharan**

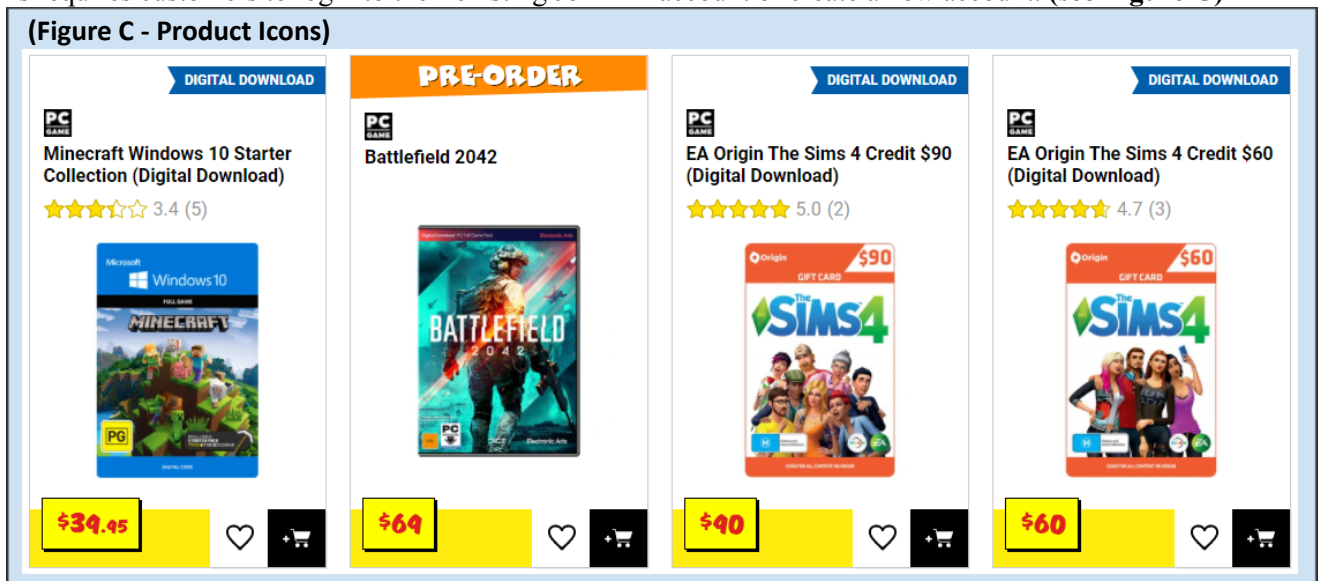
Database Design Assignment Part A

A.1. Case Study Overview

- JB Hi-Fi, (JB Hi-Fi Group Pty Ltd), is one of Australia's largest home entertainment retailers which has a website (jbhifi.com.au). The system separates their products into a list of categories under the Products tab, the list includes: New at JB-Hi Fi, Computers, TBS, Headphones, Speakers & Audio, Smart Home, Mobile Phones, Gaming, Movies & TV shows, Music & Vinyl, Collectibles & Merchandise, Cameras, Drones, E-scooters & Car Gear, Health, Fitness & Wearables, Home Appliances. Each category is then separated into more subcategories which makes it easier for customers to search for their product. *The focus of this assignment is purchasing video games* (see figure A)



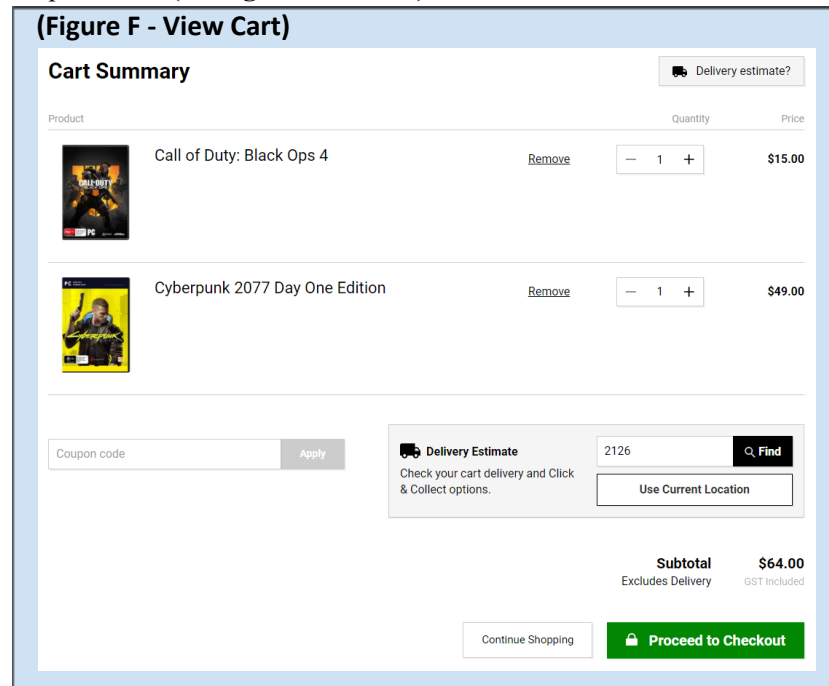
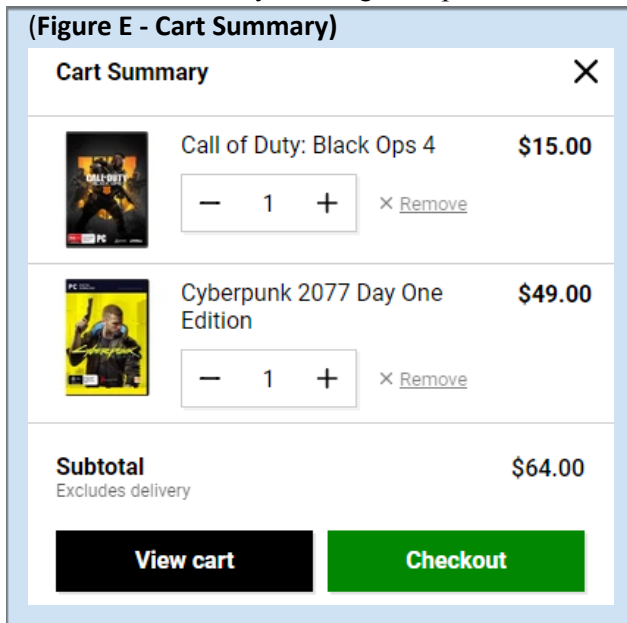
- Upon selecting the correct category/ subcategory, the website then redirects the customer to a page filled with products that are classified in that specific category. Customers can then sort the order of products based on availability, brand, price, new to old etc. Each product contains an image that reflects the product, price, product name, reviews, review rating (out of 5 stars) and the type of product (e.g. digital copy, pre order). Customers are able to click on the product to further examine the specific product in detail or add the product to their cart by clicking the cart icon. They can also add specific products to their Wish List by clicking the heart icon, however this requires customers to login to their existing Jb Hi-Fi account or create a new account. (see Figure C)



- Once the customer selects a specific product, the website then redirects them to its respective product page which contains more product images and details alongside with a comprehensive product description and customer reviews. A series of 'FREQUENTLY BOUGHT TOGETHER' products are also recommended on the product page. Customers are also given the option to add that specific product to their Wish List (which requires an account) or add product to cart. A phone number is also provided for customers who would like to 'price match' and 'ask for a JB Deal'. The system can also help customers to check for stocks at various stores by simply entering their postcode. (see figure D)



- After adding their desired item to the cart, customers are then able to review their cart before checking out. A cart summary will be generated which enables customers to review their selected products, change the quantity of each individual product or remove certain products. By clicking on 'View cart', the system will then redirect the customer to a more comprehensive version of the cart summary in which customers are able to get a delivery estimate by entering their postcode and/or apply a coupon code. (see figures E and F)



- Upon checkout, customers are given the option to log in to their existing JB Hi-Fi account (or create an account) or check out as a guest. Customers will be required to enter their contact information which is their email, alongside with their delivery details. Within delivery details, customers are required to choose between delivery, click and collect and collection point. If customers select click and collect or collection point, they will be required to select a store/ collection point by entering their postcode along with their pick up details (first and last name and their phone number). If delivery is selected, customers are then required to enter their first and last name, phone number, address and company (optional) upon which the system displays a few delivery methods, each with an estimated delivery date and cost. According to the terms of sale provided in their website, customers are encouraged to create an account on the JB Hi-Fi Website unless they utilise the 'Guest Checkout' feature. Logging in or creating an account will speed up the checkout process since their personal details (first and last name, email, phone number and address) will be linked and stored in their account. (see figure G, H, I, J)

(Figure G - Check out as guest)

Secure checkout

[Cart](#) > [Customer and Shipping Information](#) > [Payment](#)

Already have an account?

Log in

Check out as a guest

Contact information

Email

Enter a valid email

☐ Keep me up to date on exclusive JB deals and offers

Delivery details

Country/region
Australia

☐ Save this information for next time

1 Call of Duty: Black Ops 4 \$15.00

1 Cyberpunk 2077 Day One Edition \$49.00

Coupon code

Subtotal \$64.00

Shipping \$7.99

Order total AUD **\$71.99**
Includes \$5.81 GST

(Figure H - Delivery)

Delivery

Click and Collect

Collection Point

First name

Last name

Enter a first name

Enter a last name

+61 Mobile phone

Enter a valid mobile phone

Start typing your address

Company (optional)

< Return to cart

Proceed To Payment

(Figure I - Click and Collect)

Delivery

Click and Collect

Collection Point

Postcode

2000

Chatswood Westfield

In stock

SHOP 500, WESTFIELD CHATSWOOD, CHATSWOOD NSW 2067

8.19km away

Item(s) ready for pick up tomorrow during store opening hours at Chatswood Westfield (8.19km away).

Pickup details

First name

Last name

+61 Mobile phone

(Figure J - Collection Point)

Delivery

Click and Collect

Collection Point

Collection point location

383 Kent Street Parcel Locker, 383 Kent Street, SYDNEY NSW

Australia Post eParcel

Items delivered with tracking by Friday 3rd September.

\$3.98

Pickup details

First name

Last name

+61 Mobile phone

< Return to cart

Proceed To Payment

- In order to create an account, customers must enter their first and last name and personal email. They must also create a password that contains at least 6 characters for their account. Their email address and password will be used to login to their newly created/ existing accounts. (see figures K, L)

(Figure K - Create an account)

Create an account

- ✓ Checkout faster with saved addresses
- ✓ All your online orders and in-store digital receipts in one place
- ✓ Save products to your Wish List
- ✓ Access exclusive JB deals

First name

Last name

Email address

Password

Show

Must be at least 6 characters.

By selecting 'Create account' you agree to our [terms of sale](#), [terms of use](#) and how we handle your personal information as outlined in our [privacy policy](#).

Create account

(Figure L - Log in)

Store Finder

JB HI-FI

Search products

Help & Support

Wish List

My Account

My Cart

Products

Brands

Deals & Sales

Services

Gift Cards

Log in

Email address

Password

Show

[Forgot password?](#)

Log in

Faster checkout, access your tax invoices, in-store digital receipts and get exclusive JB deals!

Create an account

- After reviewing their cart and entering their details, the system will then redirect the customer to the payment page where they will be asked to select a payment method which ranges from credit or debit card, PayPal, afterpay, zip and latitude pay. If a customer chooses a credit or debit card, they will be required to fill in their card details which include their card number, name on card, expiration date and security code. If a customer chooses PayPal, afterpay, zip or latitude pay, the system will redirect them to the respective website after clicking 'Complete Order'. Customers are still able to change any of their personal details and delivery method, they can also redeem their gift card by simply entering their gift card number and pin. Customers can also use another address as their billing address by selecting 'Use a different billing address' and input a different address in the payment page. Given that the card details are valid, upon filling in all the details required, customers can click 'Pay Now' to finalise payment. (see figures M, N)

(Figure M - Payment Page)

Gift card

Payment

Total payable: \$71.99

☒ Credit or debit card

☐ **PayPal** Pay Now or Pay in 4

☐ **afterpay**

☐ **zip**

☐ **LATITUDEPAY**

Billing address

Select the address that matches your card or payment method.

☒ Same as shipping address
 ☐ Use a different billing address

By placing your order and providing the personal information requested, you confirm that you agree and accept our [website terms of sale](#) and [privacy policy](#).

(Figure N - Other Payment method)

Payment

Total payable: \$71.99

☐ Credit or debit card

☒ **PayPal** Pay Now or Pay in 4

Pay now or pay later in 4 interest-free instalments. After clicking "Complete order", you will be redirected to PayPal to complete your purchase securely.

☐ **afterpay**

☐ **zip**

☐ **LATITUDEPAY**

- Finally, a confirmation email will be sent to their personal email address and customers can also review their purchases by logging in and clicking on 'my account'. Both the confirmation email and purchase history will provide customers with a unique purchase order number, date of purchase, fulfillment type, order status and the total.

A.2. Database Functions

The database of JB-Hi-Fi's company website (the system) should effectively and efficiently support the company's core business functions and overall goal through the services the website provides to the customers (the end users). The database should provide indexing functionality by assisting users search or browse through their available products and produce an accurate list of products based on what type of data the customer has searched for/requested.

The database should be able to filter through all the data available on JB Hi-Fi's products and generate search results matching what the user/customer has requested from the system. The database should also correctly update the user's cart as necessary when they select a specific item and add it to the cart, and also update and display the total cost of items which is owed by the user to JB Hi-Fi (using the displayed prices on the website). It should update the total cost payable for the user when they select a particular delivery option type.

It should also perform the functionality of a completed business transaction by directing the user to a new page for the user to enter their payment details, including keeping the user's personal details (such as name, address, contact number and email address) and credit card details. The database must quickly compare the user's inputted details with those of every other past user in the database to determine the legitimacy of the purchaser.

A.3. Data Requirements

User: The user is the customer account through which purchases from JB Hi-Fi are made. It is uniquely identified by UserID and retains data about the user's personal details such as name, phone number, address and email address.

- **User attributes:** UserID, UserEmail, UserFirstName, UserLastName, UserPhoneNum, UserAddressNum, UserAddressStreet, UserAddressState, UserAddressState, UserAddressPostCode

Order: The order refers to the specific combination of products that the customer/user has chosen for themselves. It is uniquely identified by OrderID and will contain details about the total price payable for the user and the quantity of products ordered.

- **Order attributes:** OrderID, OrderTotalPrice, OrderQuantity,

Product: The product is the item available for purchase from JB Hi-Fi. It is uniquely identified as ProductID and has data about the product's name and price for purchase.

- **Product attributes:** ProductID, ProductName, ProductPrice

Delivery: This refers to the type of delivery option selected by the user as part of their order. It is uniquely identified by DeliveryID and has attributes which define the details of the delivery type such as the date and time of expected arrival as well as the cost for selecting that particular delivery option.

- **Delivery attributes:** DeliveryID, DeliveryDate, DeliveryTime, DeliveryPrice

Payment: The payment is the final transaction between the user and JB Hi-Fi which involves the transferral of money for the products the user has ordered. It is uniquely identified by PaymentID and retains data about the details of the purchase.

- **Payment attributes:** PaymentID, PaymentCardNum, PaymentCCV, PaymentExpiryDate, PaymentCardFirstName, PaymentCardLastName

A.4. Business Rules

BR1: A customer will have a unique Customer ID

BR2: Every customer needs to provide his/her personal information including name, home address, email address and contact phone number at the beginning of a transaction.

BR3: An order can have any number of items but must have at least one

BR4: It is mandatory for a customer to provide their credit card details to finish the transaction

BR5: Customers may only select one delivery method.

BR6: An item can be in many orders

BR7: One address can receive one or many orders.

BR8: A customer can make one or many orders.

BR9: A customer may not create a new account with an already used email address – they must first log in

BR10: One customer can have one or many delivery addresses

Database Design Assignment Part B

B.1. Overview of the case study in at most five sentences (From Part A)

JB Hi-Fi's website (jbhifi.com.au) separates their products into a list of categories and subcategories to assist customers to navigate their webpage easily. Upon selecting the desired category/ subcategory, the system then displays a table of matching products. Subsequently, customers can add the matching products into their shopping cart and a cart summary will be generated for review. As the customer proceeds to checkout, a detailed check out page will be generated with the option for customers to log in to their existing JB Hi-Fi accounts or continue as guests. Once the customers fill out all the required details, the system will generate a payment page for customers to finalise their transaction.

B.2. Revised Business Rules and Assumptions

Below are the revised business rules and assumptions for the database which were used to identify all the relevant entities and cardinality relationships. An accurate ERD was constructed in section B.3. in respect of this.

CUSTOMER ENTITY:

- **BR1:** A customer will have a unique Customer ID
- **BR2:** Every customer needs to provide his/her personal information including name, home address, email address and contact phone number at the beginning of a transaction.
- **BR3:** Every customer can make many orders but need not make one.
- **BR4:** Each order can only be linked to one customer
- **BR5:** It is mandatory for a customer to provide their credit card details to finish the transaction
- **BR6:** Customers may only select one delivery method.

ORDER ENTITY:

- **BR7:** An order can not exist without a customer.
- **BR8:** Each order is identified by a unique ID, the date it was made and the form of delivery
- **BR9:** A click-n-collect order can only be collected from one store
- **BR10:** A CNC order must include a date to be picked up from a store
- **BR11:** A shipping address must be provided for the order to be delivered
- **BR12:** A delivery must include a date when it will be arriving to the customer
- **BR13:** An order can have any number of products but must have at least one
- **BR14:** Each order creates one payment.
- **BR15:** Each order generates one bill

PRODUCT ENTITY:

- **BR16:** Each product is uniquely identified by an ID and has a type, name, price
- **BR17:** An product can be in none or many orders

- **BR18:** A product can be included more than once in an order
- **BR19:** A product can be supplied by any number of warehouses to more than one store but each product must be supplied by at least one warehouse to a store
- **BR20:** Each product belongs to one category.
- **BR21:** A product is sold by at least one store.
- **BR22:** A product can be included in the same store more than once

PAYMENT ENTITY:

- **BR23:** Each payment has a unique ID
- **BR24:** Each payment stores credit card details, payment date, and the total cost
- **BR25:** Each payment is linked to one order.

BILL ENTITY:

- **BR26:** Each bill has a unique ID and date.
- **BR27:** Each bill is linked to one order.

STORE ENTITY:

- **BR28:** Each store has a unique ID, address and phone number.
- **BR29:** Each store can supply any number of products from more than one warehouse, but must be supplied with at least one product.
- **BR30:** Each store can have many click-n-collect deliveries for customers to pick up.

WAREHOUSE ENTITY:

- **BR31:** Each warehouse has a unique ID, address and phone number.
- **BR32:** Each warehouse can supply many products to any number of stores, but need not supply any products

CATEGORIES ENTITY:

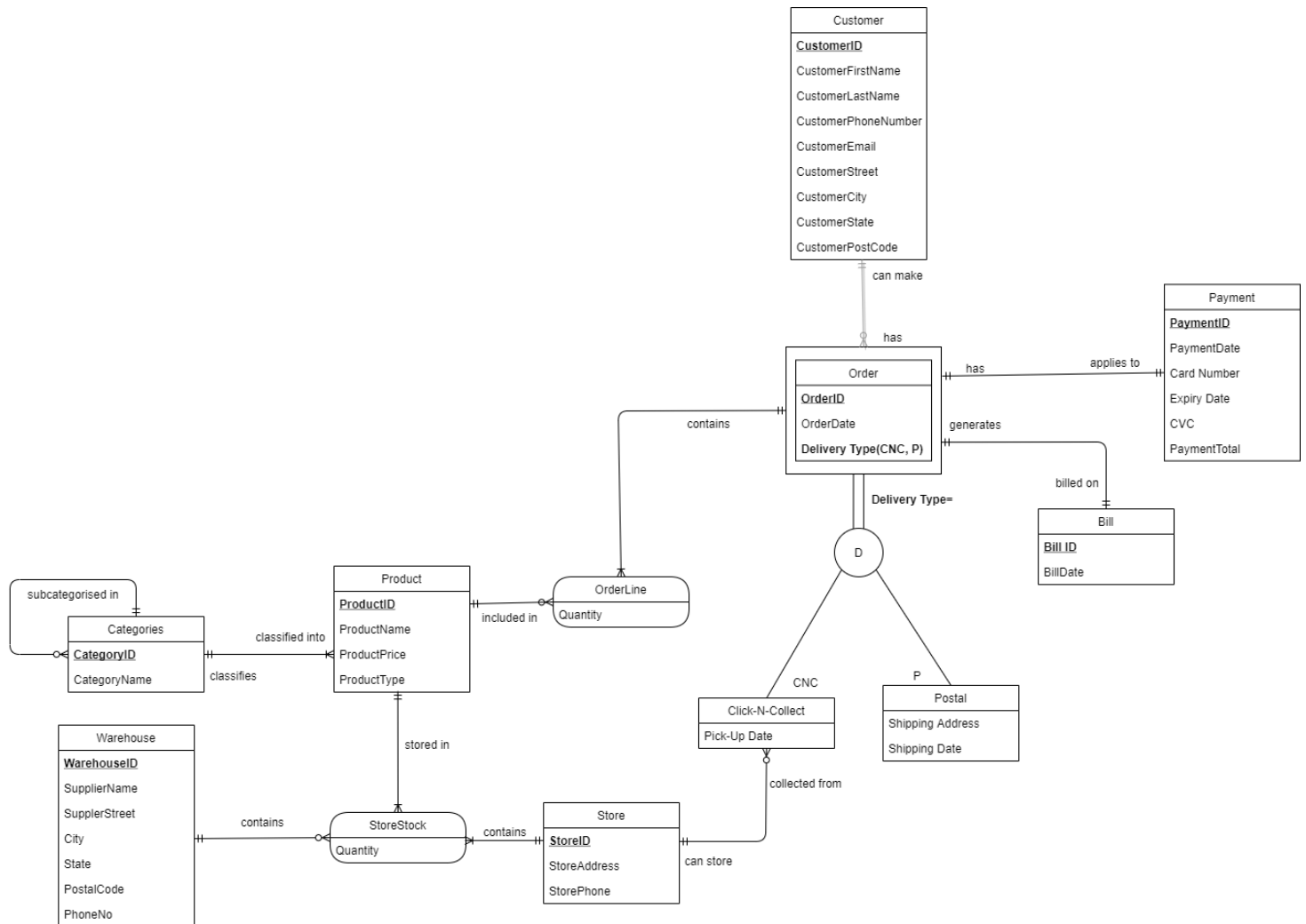
- **BR33:** Each category has a name and unique ID.
- **BR34:** Each category can have many sub-categories.
- **BR35:** A category can hold many products but must have at least one.

Assumptions

- **Assumption1:** Customer will utilise the guest checkout.
- **Assumption2:** The shipping address and the address provided by the customer is the same
- **Assumption3:** The system will only provide free standard delivery

- **Assumption4:** Each products will fall under only one category, you will not find a product with two different category types
- **Assumption5:** A product is not sold in bulk and must be quantified with the number of stock of a specific product within a store
- **Assumption6:** Customers will be purchasing a physical product. (Not a digital download)

B.3. ERD



B.4. Justifications of the ERD based on the business rules and/or assumptions

Entity - Customer - *Strong Entity*

- Business rules related to entity CUSTOMER:
 - **BR1:** A customer will have a unique Customer ID
 - **BR2:** Every customer needs to provide his/her personal information including name, home address, email address and contact phone number at the beginning of a transaction.
 - **BR5:** It is mandatory for a customer to provide their credit card details to finish the transaction

The PK of CUSTOMER is: CustomerID

Assumption1: Customer will utilise the guest checkout.

- Business rules related to the relationship between CUSTOMER and ORDER and its cardinalities:
 - **BR3:** Every customer can make many orders but need not make one (cardinality 0 to many)
 - **BR4:** Each order can only be linked to one customer (cardinality 1 to 1)
 - **BR6:** Customers may only select one delivery method.

The use of the double lines for the cardinality shows that the Customer entity is a strong entity for Order.

Entity - Order - *Weak Entity/Supertype*

- Business rules related to entity ORDER:
 - **BR7:** An order can not exist without a customer.

Based on this Business rule, we have made Order a weak entity as it cannot exist without a customer making one

- **BR8:** Each order is identified by a unique ID, the date it was made and the form of delivery

The PK of ORDER is: OrderID and CustomerID (Composite Primary Keys), OrderID is the partial key and CustomerID is the strong primary key

The order entity is also declared as a supertype, as there can be two forms of delivery. The entity follows the Total Specialisation Rule as all possible subtypes are included in the ERD (this is shown by the double line). The D in the middle circle shows that an Order can either be a Delivery or a Click-N-Collect, but not both.

Subtype - Click-N-Collect

- **BR9:** A click-n-collect order can only be collected from one store (cardinality 1 to 1)
- **BR10:** A CNC order must include a date to be picked up from a store

The FK of Click-N-Collect is: StoreID

Subtype- Delivery

- **BR11:** A shipping address must be provided for the order to be delivered
- **BR12:** A delivery must include a date when it will be arriving to the customer

Assumption2: The shipping address and the address provided by the customer is the same

Assumption3: The system will only provide free standard delivery.

- Business rules related to the relationship between ORDER and PRODUCT and its cardinalities:

- **BR13:** An order can have many products but must have at least one. (cardinality 1 to many)
- **Note:** As you can see there exists a many-to-many relationship between ORDER and PRODUCT, which is converted to an associative entity. (OrderLine)

The PK of ORDERLINE is: OrderID and ProductID (Composite Primary Keys)

- Business rules related to the relationship between ORDER and PAYMENT and its cardinalities:
 - **BR14:** Each order creates one payment. (cardinality 1 to 1).
- Business rules related to the relationship between ORDER and BILL and its cardinalities:
 - **BR15:** Each order generates one bill. (cardinality 1 to 1).

Entity - Product

- Business rules related to entity PRODUCT:
 - **BR16:** Each product is uniquely identified by an ID and has a type, name, price

The PK of PRODUCT is: ProductID
The FK of PRODUCT is: CategoryID
- Business rules related to the relationship between PRODUCT and CATEGORIES and its cardinalities:
 - **BR20:** Each product belongs to one category. (cardinality 1 to 1).
 - **Assumption4:** Each products will fall under only one category, you will not find a product with two different category types
- Business rules related to the relationship between PRODUCT and STORE and WAREHOUSE and its cardinalities:
 - **BR21:** A product is sold by at least one store. (cardinality 1 to many)
 - **BR22:** A product can be included in the same store more than once
 - **BR19:** A product can be supplied by any number of warehouses to more than one store but each product must be supplied by at least one warehouse to a store (cardinality 1 to many)
 - **Note:** As you can see there is a many-to-many relationship between PRODUCT, STORE and WAREHOUSE. This is also a ternary relationship and is converted to an associative entity (StoreStock)

The PK of STORESTOCK: ProductID, StoreID and WarehouseID (Composite Primary Key)
Assumption5: A product is not sold in bulk and must be quantified with the number of stock of a specific product within a store
- Business rules related to the relationship between PRODUCT and ORDER and its cardinalities:
 - **BR17:** A product can be in none or many orders (cardinality 0 to many)
 - **BR18:** A product can be included more than once in an order
 - **Note:** As you can see there exists a many-to-many relationship between ORDER and PRODUCT, which is converted to an associative entity. (OrderLine)

The PK of ORDERLINE is: OrderID and ProductID (Composite Primary Keys)

Entity - Categories

- Business rules related to entity CATEGORIES:
 - **BR33:** Each category has a name and unique ID.
The PK of CATEGORIES is: CategoryID
The FK of CATEGORIES is: CategoryID
- Business rules related to the relationship of CATEGORIES and its cardinalities:
 - **BR34:** Each category can have many sub-categories (cardinality 0 to many).
 - **Note:** As a category can be subcategorized into another category, it means it has a unary relationship, as shown with in the ERD
- Business rules related to the relationship between CATEGORIES and PRODUCT and its cardinalities:
 - **BR35:** A category can hold many products, but must have at least one (cardinality 1 to many)

Entity - Store

- Business rules related to entity STORE:
 - **BR28:** Each store has a unique ID, address and phone number.
The PK of STORE is: StoreID
- Business rules related to the relationship between PRODUCT and STORE and WAREHOUSE and its cardinalities:
 - **BR29:** Each store can supply any number of products from more than one warehouse, but must be supplied with at least one product. (cardinality 1 to many)
 - **Note:** As you can see there is a many-to-many and ternary relationship between PRODUCT, STORE and WAREHOUSE, and that is converted to an associative entity. (StoreStock)
The PK of STORESTOCK: ProductID, StoreID and WarehouseID
- Business rules related to the relationship between STORE and SUBTYPE(CLICK-N-COLLECT) and its cardinalities:
 - **BR30:** Each store can have many click-n-collect deliveries for customers to pick up (cardinality 0 to many)

Entity - Payment

- Business rules related to entity PAYMENT:
 - **BR23:** Each payment has a unique ID.
 - **BR24:** Each payment stores credit card details, payment date, and the total cost
The PK of PAYMENT is: PaymentID
The FKs of PAYMENT are: OrderID and CustomerID
As Order is a weak entity, it will carry on its partial key(OrderID) and strong entities primary key(CustomerID)
- Business rules related to the relationship between PAYMENT and ORDER and its cardinalities:

- **BR25:** Each payment is linked to one order. (cardinality 1 to 1)

Entity - Bill

- Business rules related to entity BILL:
 - **BR26:** Each bill has a unique ID and date.
The PK of BILL is: BillID,
The FKs of BILL are: OrderID and CustomerID
 As Order is a weak entity, it will carry on its partial key(OrderID) and strong entities primary key(CustomerID)
- Business rules related to the relationship between BILL and ORDER and its cardinalities:
 - **BR27:** Each bill is linked to one order. (cardinality 1 to 1)

Entity - Warehouse

- Business rules related to entity WAREHOUSE :
 - **BR31:** Each warehouse has a unique ID, name and address.
The PK of WAREHOUSE is: WarehouseID
Assumption6: Customers will be purchasing a physical product. (Not a digital download)
- Business rules related to the relationship between PRODUCT and STORE and WAREHOUSE and its cardinalities:
 - **BR32:** Each warehouse can supply many products to any number of stores, but need not supply any products (0 to many)
 - **Note:** As you can see there is a many-to-many and ternary relationship between PRODUCT, STORE and WAREHOUSE, and that is converted to an associative entity. (StoreStock)
The PK of STORESTOCK is: ProductID, StoreID and WarehouseID

Database Design Assignment Part C

C.1. Revised Business rules and assumptions (From part B)

Below are the revised business rules and assumptions for the database which were used to identify all the relevant entities and cardinality relationships. An accurate ERD was constructed in section B.3. in respect of this.

CUSTOMER ENTITY:

- **BR1:** A customer will have a unique Customer ID
- **BR2:** Every customer needs to provide his/her personal information including name, home address, email address and contact phone number at the beginning of a transaction.
- **BR3:** Every customer can make many orders but need not make one.
- **BR4:** Each order can only be linked to one customer
- **BR5:** It is mandatory for a customer to provide their credit card details to finish the transaction
- **BR6:** Customers may only select one delivery method.

ORDER ENTITY:

- **BR7:** An order can not exist without a customer.
- **BR8:** Each order is identified by a unique ID, the date it was made and the form of delivery
- **BR9:** A click-n-collect order can only be collected from one store
- **BR10:** A CNC order must include a date to be picked up from a store
- **BR11:** A shipping address must be provided for the order to be delivered
- **BR12:** A delivery must include a date when it will be arriving to the customer
- **BR13:** An order can have any number of products but must have at least one
- **BR14:** Each order creates one payment.
- **BR15:** Each order generates one bill

PRODUCT ENTITY:

- **BR16:** Each product is uniquely identified by an ID and has a type, name, price
- **BR17:** An product can be in none or many orders
- **BR18:** A product can be included more than once in an order
- **BR19:** A product can be supplied by any number of warehouses to more than one store but each product must be supplied by at least one warehouse to a store
- **BR20:** Each product belongs to one category.
- **BR21:** A product is sold by at least one store.
- **BR22:** A product can be included in the same store more than once

PAYMENT ENTITY:

- **BR23:** Each payment has a unique ID

- **BR24:** Each payment stores credit card details, payment date, and the total cost
- **BR25:** Each payment is linked to one order.

BILL ENTITY:

- **BR26:** Each bill has a unique ID and date.
- **BR27:** Each bill is linked to one order.

STORE ENTITY:

- **BR28:** Each store has a unique ID, address and phone number.
- **BR29:** Each store can supply any number of products from more than one warehouse, but must be supplied with at least one product.
- **BR30:** Each store can have many click-n-collect deliveries for customers to pick up.

WAREHOUSE ENTITY:

- **BR31:** Each warehouse has a unique ID, address and phone number.
- **BR32:** Each warehouse can supply many products to any number of stores, but need not supply any products

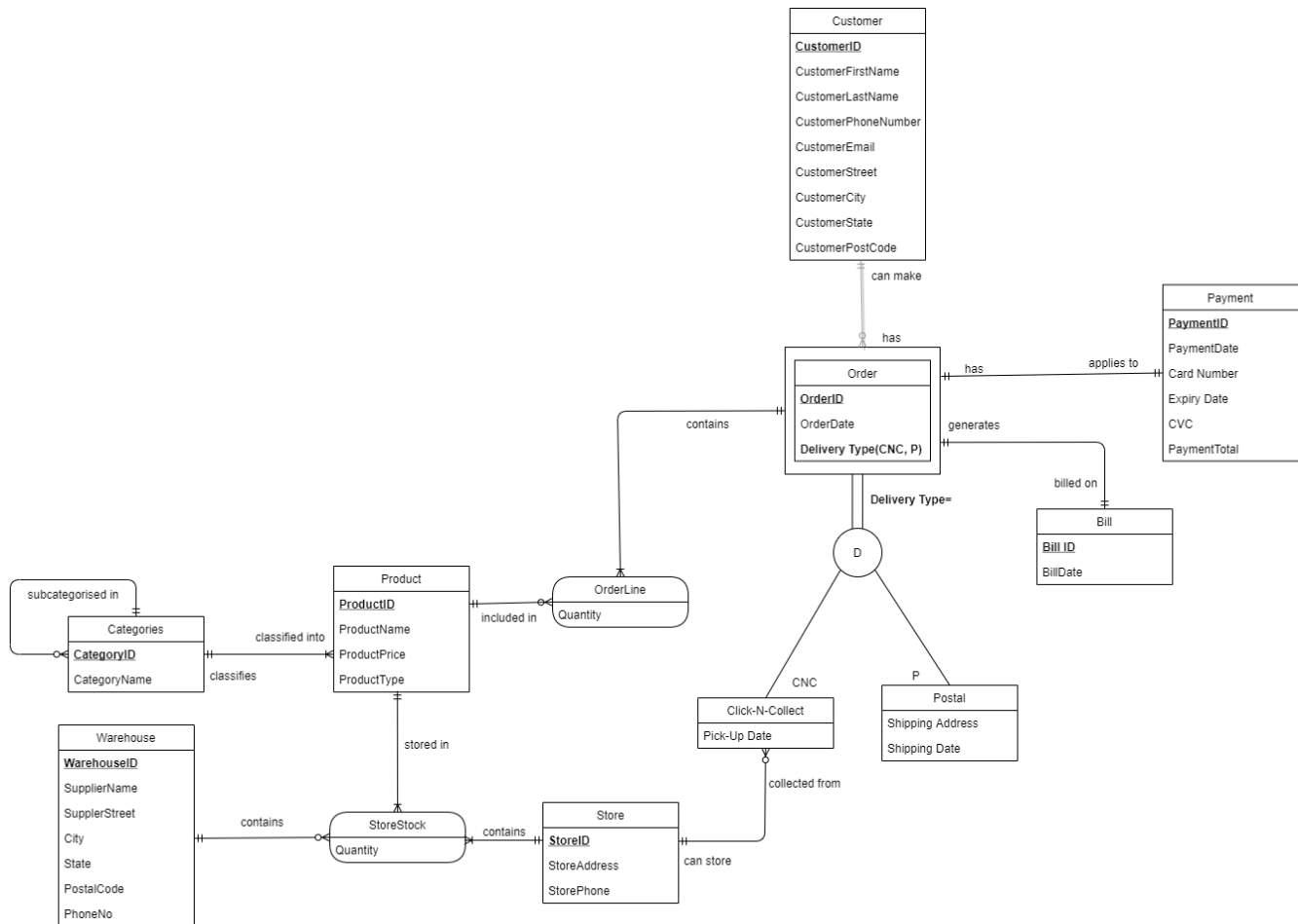
CATEGORIES ENTITY:

- **BR33:** Each category has a name and unique ID.
- **BR34:** Each category can have many sub-categories.
- **BR35:** A category can hold many products but must have at least one.

Assumptions

- **Assumption1:** Customer will utilise the guest checkout.
- **Assumption2:** The shipping address and the address provided by the customer is the same
- **Assumption3:** The system will only provide free standard delivery
- **Assumption4:** Each products will fall under only one category, you will not find a product with two different category types
- **Assumption5:** A product is not sold in bulk and must be quantified with the number of stock of a specific product within a store
- **Assumption6:** Customers will be purchasing a physical product. (Not a digital download)

C.2. Revised ERD of Part B



C.3. Relations

Customer(CustomerID, CustomerFirstName, CustomerLastName, CustomerPhoneNum, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerPostalCode)

Order(OrderID, CustomerID*, OrderDate, DeliveryType)

FK(CustomerID) references(Customer)

Click-N-Collect(OrderID*, CustomerID*, StoreID*, Pick-Up Date)

FK (OrderID) references (Order)

FK (CustomerID) references (Customer)

FK(StoreID) references(Store)

Postal(OrderID*, CustomerID*, Shipping Address, Shipping Date)

FK (OrderID) references (Order)

FK (CustomerID) references (Customer)

Payment(**PaymentID**, OrderID*, CustomerID*, PaymentDate, CardNumber, ExpiryDate, CVC, Total)
FK(OrderID) references (Order)
FK (CustomerID) references (Customer)

Bill(**BillID**, OrderID*, CustomerID*, BillDate)
FK(OrderID) references (Order)
FK(CustomerID) references (Customer)

OrderLine(**OrderID***, **ProductID***, Quantity)
FK(OrderID) references (Order)
FK(ProductID) references (Product)

Product(**ProductID**, CategoryID* ProductName, ProductPrice, ProductType)
FK(CategoryID) references(Category)

Categories(**CategoryID**, SuperCategoryID*, CategoryName)
FK(SuperCategoryID) references (Category)

Store(**StoreID**, StoreAddress, StorePhone)

WareHouse(**WarehouseID**, SupplierName, SupplierStreet, City, State, PostalCode, PhoneNo)

StoreStock(**ProductID***, **StoreID***, **WarehouseID***, Quantity)
FK(ProductID) references (Product)
FK(WarehouseID) references (Warehouse)
FK(StoreID) references (Store)

C.4. List of functional dependencies related to each business rule.

FD1: CustomerID \rightarrow CustomerFirstName, CustomerLastName, CustomerPhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerPostCode
(refer to BR1, BR2, A1)

FD2: OrderID \rightarrow CustomerID
(refer to BR4, BR7)

We will consider “Order” as a weak entity of “Customer” based on the BR7.

FD3: OrderID, CustomerID \rightarrow OrderDate, DeliveryType(CNC, P)
(refer to BR3, BR6, BR8, BR7)

FD4: CustomerID, OrderID \rightarrow StoreID, Pick-Up Date

(refer to BR9, BR10, BR30)

FD5: OrderID, CustomerID → Shipping Address, ShippingDate

(refer to BR11, BR12, A2, A3)

FD6: PaymentID, OrderID, CustomerID → PaymentDate, CardNumber, ExpiryDate, CVC, PaymentTotal

(refer to BR5, BR14)

FD7: PaymentID → PaymentDate, CardNumber, ExpiryDate, CVC, PaymentTotal, OrderID, CustomerID

(refer to BR23, BR24, BR25)

FD8: BillID, OrderID, CustomerID → BillDate

(refer to BR15,

FD9: OrderID, ProductID → Quantity

(refer to BR13, BR17, BR18

FD10: ProductID → ProductName, ProductPrice, ProductType, CategoryID

(refer to BR16, BR20, A4, BR35)

FD11: BillID → OrderID, CustomerID, BillDate

(refer to BR27, BR26)

FD12: CategoryID → CategoryName

(refer to BR33)

FD13: CategoryID → CategoryID

(refer to BR34)

FD14: WarehouseID → WarehousePhoneNumber, WarehouseCity, WarehouseStreet, WarehouseState, WarehousePostCode

(refer BR31)

FD15: WarehouseID, StoreID, ProductID → Quantity

(refer to B19, B21, B22, BR29, BR32, A5)

FD16: StoreID → StorePhoneNumber, StoreStreet, StoreCity, StoreState, StorePostCode

(refer BR28)

C.5. Normalisation (Logical design):

1. Customer(**CustomerID**, CustomerFirstName, CustomerLastName, CustomerPhoneNum, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerPostalCode)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for every PK value and there is a unique identifier for each row.

CustomerID -> CustomerFirstName, CustomerLastName, CustomerPhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerPostCode

- This relation is in 2NF as it is in 1NF, and there are no partial functional dependencies on the key CustomerID. All the attributes are related to Customer.
- This relation is in 3NF as it is in 2NF, and there are no transitive dependencies.

2. Order(**OrderID**, **CustomerID***, OrderDate, DeliveryType)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row

OrderID, CustomerID -> OrderDate, DeliveryType(CNC, P)

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the key OrderID and CustomerID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

3. Click-N-Collect(**OrderID***, **CustomerID***, StoreID*, Pick-Up Date)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row

CustomerID, OrderID -> StoreID, Pick-Up Date

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the key OrderID and CustomerID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

4. Postal(**OrderID***, **CustomerID***, Shipping Address, Shipping Date)

- This relation is not in 1NF as Shipping address is non atomic and can be divided into Street, City, State, Postal Code

OrderID, CustomerID-> Shipping Address, ShippingDate - - - Address needs to be atomic

Postal(**OrderID***, **CustomerID***, Street, City, State, Postal Code, Shipping Date) - - CORRECT FORM of 1NF

P

Postal
Street
City
State
Postal Code
Shipping Date

OrderID, CustomerID -> Street, City, State, Postal Code, ShippingDate - - - CORRECT FD

- Now there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row
- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the key OrderID and CustomerID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

5. Payment(**PaymentID**, **OrderID***, **CustomerID***, PaymentDate, CardNumber, ExpiryDate, CVC, Total)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

PaymentID, OrderID, CustomerID -> PaymentDate, CardNumber, ExpiryDate, CVC, Total

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the keys PaymentID, OrderID and CustomerID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

6. Bill(**BillID**, **OrderID***, **CustomerID***, BillDate)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

BillID, OrderID, CustomerID -> BillDate

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the keys BillID, OrderID and CustomerID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

7. OrderLine(**OrderID***, **ProductID***, Quantity)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

OrderID, ProductID -> Quantity

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the keys OrderID and ProductID.
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

8. Product(**ProductID**, CategoryID* ProductName, ProductPrice, ProductType)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

ProductID -> CategoryID, ProductName, ProductPrice, ProductType

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the key ProductID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

9. Categories(**CategoryID**, CategoryID*, CategoryName)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

CategoryID -> CategoryID, CategoryName*

- This relation is in 2NF as it is in 1NF and there are no partial functional dependencies on the key CategoryID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

10. Store(**StoreID**, StoreAddress, StorePhone)

- This relation is not in 1NF as StoreAddress is an attribute which is not atomic as it can be divided into Street, City, Post Code, State

Store(**StoreID**, StoreStreet, StoreCity, StoreState, StorePostCode) - - CORRECT FORM of 1NF

Store
<u>StoreID</u>
Street
City
State
PostalCode
StorePhone

StoreID → StorePhoneNumber, StoreStreet, StoreCity, StoreState, StorePostCode - - CORRECT FD

- Now there are no atomic or derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row
- This relation is in 2NF as it is in 1NF and there are no partial dependencies on the key StoreID
- This relation is in 3NF as it is in 2NF and there are no transitive dependencies

11. Warehouse(WarehouseID, SupplierName, SupplierStreet, City, State, PostalCode, PhoneNo)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

WarehouseID → WarehousePhoneNumber, WarehouseCity, WarehouseStreet, WarehouseState, WarehousePostCode

- This relation is in 2NF as it is in 1NF, and there are no partial functional dependencies on the key WarehouseID. All the attributes are related to Warehouse.
- This relation is in 3NF as it is in 2NF, and there are no transitive dependencies.

12. StoreStock(ProductID*, StoreID*, WarehouseID*, Quantity)

- This relation is in 1NF as all attributes are atomic and there are no derived attributes. All attributes have atomic values for each PK, and a primary key uniquely identifies each row.

WarehouseID, StoreID, ProductID → Quantity

- This relation is in 2NF as it is in 1NF, and there are no partial functional dependencies on the key ProductID, StoreID and WarehouseID.
- This relation is in 3NF as it is in 2NF, and there are no transitive dependencies.

Database Design Assignment Part D

D.1. Created Database