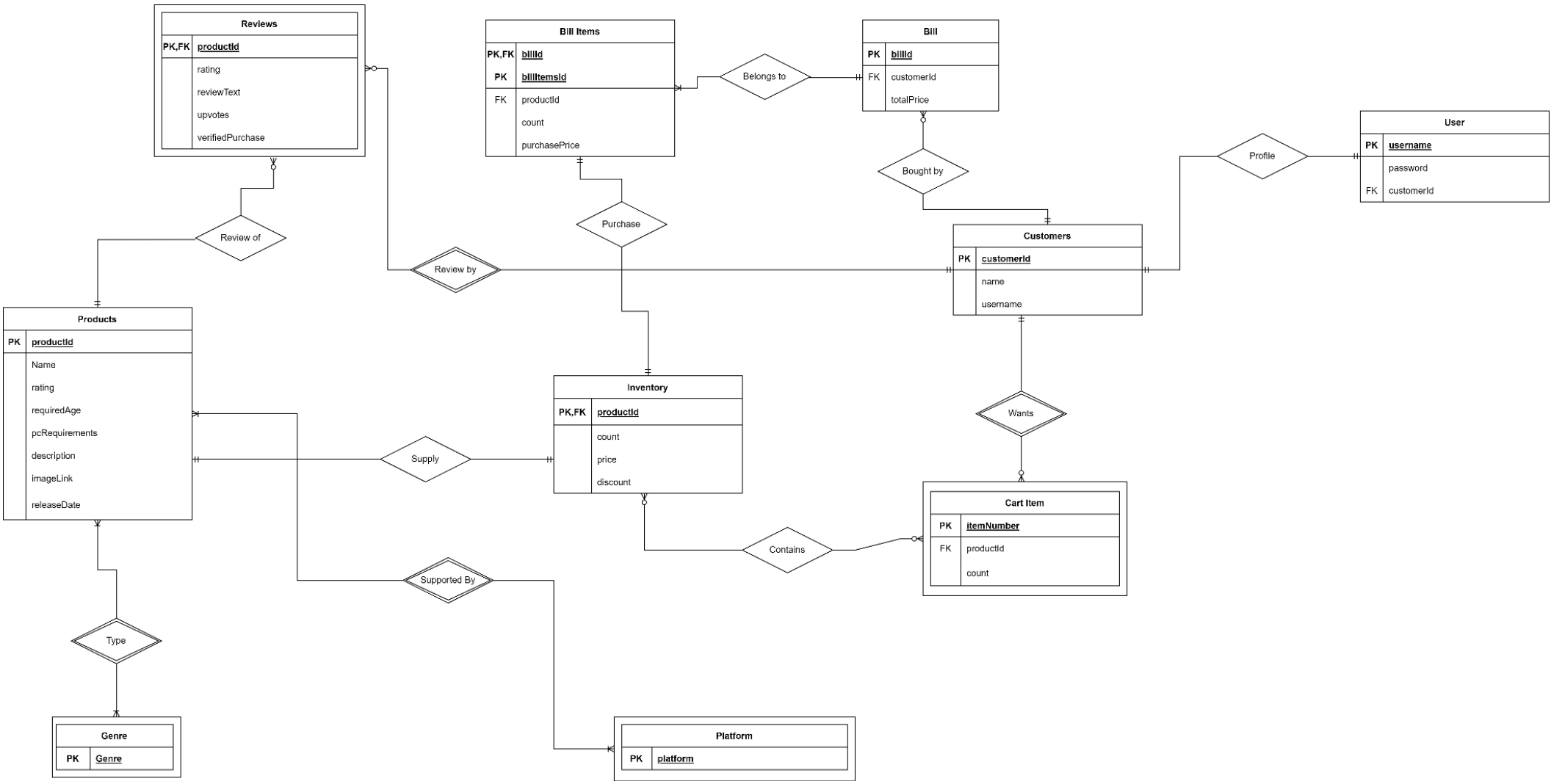
Reviews
PK,FK productId
rating
reviewText
upvotes
verifiedPurchase
Bill Items
PK,FK billId
PK billItemsId
FK productId
count
purchasePrice
Belongs to
Bill
PK billId
FK customerId
totalPrice
Bought by
User
PK username
password
FK customerId
Profile
Review of
Purchase
Review by
Customers
PK customerId
name
username
Products
PK productId
Name
rating
requiredAge
pcRequirements
description
imageLink
releaseDate
Inventory
PK,FK productId
count
price
discount
Supply
Wants
Cart Item
PK itemNumber
FK productId
count
Contains
Supported By
Type
Genre
PK Genre
Platform
PK platform

Assumptions:

We will assume that productId, itemNumber, customerId, uniqueId, billId, billItemsId, and username are unique. We will assume that each purchase consists of one customer that buys one to many games. We will assume that a product can have 0 to many reviews. We will also assume that customers can make 0 to many reviews. We will assume that each review references exactly one product and is written by exactly one customer. We will assume that every bill contains at least one bill item but can contain more and that each bill belongs to only one customer. We will assume that each bill item only contains one product. We will assume that every product in our inventory has exactly one set of product details and that every product in our products table has exactly one inventory entry. We will assume that every game is supported on one to many platforms and that every platform supports one to many games. We will assume that every game has at least one genre but may have more. We will assume that each genre has at least one game associated with it. We will assume that customers can make 0 to many purchases. We will assume that a cart contains 0 to many products and that a product can be in 0 to many cart items. We will assume that each cart item is associated with exactly one user and one product. We will assume that each customer exists as exactly one user. We have two separate entities for user data and user log-in information for password privacy purposes. We have two separate entities for products and inventory to track the supply of products separately from the product's existence and to speed up queries by caching.

Relationships:

1. Type
    a. Identifies the genre of each game
    b. Connects Genre and Products
    c. Many to many relationship
2. Supported by
    a. Identifies which platforms each game is supported by
    b. Connects Platform and Products
    c. Many to many relationship

3. Supply
    a. Connects detailed information about products with the price and available number
    b. Connects Products and Inventory
    c. One to one relationship
4. Contains
    a. Identifies which products are in a user’s cart, which allows the items in a customer’s cart to be displayed
    b. Connects Inventory to Cart Item
    c. Many to many relationship
5. Wants
    a. Identifies the customer associated with a cart item, which allows a customer’s cart to be displayed
    b. Connects Customers to Cart Item
    c. One to many relationship
6. User
    a. Identifies the log-in information for a specific user and connects that information with their profile
    b. Connects Customers to User
    c. One to one relationship
7. Bought by
    a. Identifies the customer that made a specific purchase, which allows
    b. Connects Customers to Bill
    c. One to many relationship
8. Belongs to
    a. Identifies the specific items that are bought in a bill, which allows a detailed summary of the purchase to be created
    b. Connects Bill to Bill Items
    c. One to many relationship
9. Review by

    a. Identifies which customer made a review, which allows determination of whether the customer has actually purchased the product for the verifiedReview attribute
    b. Connects Customers to Reviews
    c. One to many relationship
10. Review of
    a. Identifies the product that a review is referencing, which allows the review to be displayed on the product page
    b. Connects Products to Reviews
    c. One to many relationship
11. Purchase
    a. Identifies which items in the inventory need to have their counts adjusted based on which items are present in a bill
    b. Connects Inventory to Bill Items
    c. One to one relationship

Relational Schema:

Inventory(productId: INT [PK] [FK to Products.productId], supply: INT, price: REAL, discount: REAL)

Customers(customerId: INT [PK], name: VARCHAR(255), username: VARCHAR(255))

Purchases(purchaseId: INT [PK], customerId: INT [FK to Customers.customerId, productId: INT [FK to Products.productId], purchasePrice)

Bill(billId: INT [PK], customerId: INT [FK to Customers.customerId], totalPrice: REAL)

BillItems(billId: INT [PK] [FK to Bill.billId], billItemId: INT [PK], productId: INT [FK to Products.productId], count: INT, purchasePrice: REAL)

Products(productId: INT [PK], name: VARCHAR(255) [PK], rating: INT, requiredAge: INT, pcRequirements: VARCHAR(255), description: VARCHAR(1000), imageLink: VARCHAR(500), releaseDate: DATE)

Platform(platform: VARCHAR(255)[PK], productId: INT [PK][FK to Products.productId])

Reviews(uniqueId: INT [PK], purchaseId: INT [FK to Purchases.purchaseId], rating: INT, reviewText: VARCHAR(2000), upvotes: INT, verifiedPurchase: BOOLEAN)

User(username: VARCHAR(30) [PK], password: VARCHAR(30), userId: INT [FK to Customers.customerId])

Genre(genre: VARCHAR(255) [PK], productID: INT [PK] [FK to Products.productId])

CartItem(itemNumber: INT [PK], productId: INT [FK to Products.productId], count: INT)

SupportedBy(platform: VARCHAR(255) [PK] [FK to Platform.genre], productID: INT [PK] [FK to Products.productId], name: VARCHAR(255) [PK], rating: INT, requiredAge: INT, pcRequirements: VARCHAR(255), description: VARCHAR(1000), imageLink: VARCHAR(500), releaseDate: DATE)

GameType(genre: VARCHAR(255) [PK] [FK to Genre.genre], productID: INT [PK] [FK to Products.productId], name: VARCHAR(255) [PK], rating: INT, requiredAge: INT, pcRequirements: VARCHAR(255), description: VARCHAR(1000), imageLink: VARCHAR(500), releaseDate: DATE)

CartContents(itemNumber: INT [PK] [FK to CartItem.itemNumber], productId: INT [PK] [FK to Products.productId], supply: INT, price: REAL, discount: REAL, count: INT)