Group 8 Video Game Store Database

Database Specification: Purpose, Business Problems Addressed and Business Rules

Anqi Guan, Chuhan Xu, Lin Ye, Xinyu Qiu, Yuwen Mai

Database Purpose:

The purpose of this database is to maintain the data used to support customer game purchases, and analyze store operational l performance. It will be used by store employees only.

Business Problems Addressed:

- Enable the video game store to create detailed sales reports, breaking down sales trends by game, platform and publisher to optimize sales strategies.
- Provide data for inventory management to ensure that supply meets demand across different games and platforms, preventing overstocking or stockouts.
- Offer insights for targeted promotions by analyzing customer preferences and purchase histories, allowing for personalized marketing campaigns.
- Enhance customer service by using sales and preference data to provide recommendations, improving customer satisfaction and loyalty.

Business Rules:

- Each Order is placed by one Customer.
- Each Order is processed by one Employee.
- Each Order will have one Payment record.
- Each Order may have one or more Games.
- A Customer may have one or more Orders.
- An Employee may process one or more Orders.
- A Payment record is related to one Order.
- A Game may appear in one or more Orders.
- Each Game is developed by one Developer.
- Each Game is published by one Publisher.

- Each Game is categorized under one Genre.
- Each Game is available on one or more Platforms.
- Each Game can have one Inventory record.
- A Developer can develop one or more Games.
- A Publisher can publish one or more Games.
- A Genre can encompass one or more Games.
- A Platform can support one or more Games.
- An Inventory record will have one Game.
- A Customer will have one Preference record.
- Each Preference record will have one Customer.
- Each Preference record will have one Genre and one Publisher.

Design Requirements (Credit to Professor Simon Wang):

- Use Crow's Foot Notation.
- Specify the primary key fields in each table by specifying PK beside the fields.
- Draw a line between the fields of each table to show the relationships between each table.
- Specify which table is on the one side of the relationship by placing a one next to the field where the line starts.
- Specify which table is on the many sides of the relationship by placing a crow's feet symbol next to the field where the line ends.

Design Decisions:

Entity Name	**Why Entity Included**	**How Entity is Related to Other Entities**
Customer	In the Video Games Store Management Database, the customer entity is included to collect and manage comprehensive information about the individuals who purchase video games. This entity is crucial for tracking customer preferences, purchase history, and interactions with the store. By maintaining detailed records on customers, such as birthdate, gender, street address, the database facilitates personalized marketing, enhances customer service, and enables analysis of buying patterns to optimize inventory management.	As the main entity in the database, the customer's primary key, Customer ID, relates it to Order and Preference so that important information about personalized marketing and customer service may be gained.

Preference	The Preference entity is a crucial component of a video games database designed to enhance the personalized gaming experience for users. It serves to capture and understand the gaming preferences of individual customers, encompassing their favored genres, preferred publishers. By collecting these preferences, the database can facilitate targeted recommendations, improve customer satisfaction, and drive engagement by suggesting games that align with users' tastes.	The Preference entity is intricately linked to the Customer, Publisher, and Genre entities through various relationships. It connects to the Customer entity via a one-to-one relationship, allowing a single customer to express one preference. With the Publisher and Genre entities, it forms many-to-one relationships through foreign keys, Publisher ID and Genre ID, respectively. This setup acknowledges that customers can favor one publisher and genre, enabling the database to tailor recommendations and marketing efforts based on individual gaming tastes.
Employee	The Employee entity is included to manage information about the individuals who work within the store. This entity is crucial for tracking employee details such as their names. By maintaining comprehensive records on employees, the database supports the store's operational workflow and customer service.	It has a one-to-many relationship with the Order entity, indicating that each employee can manage multiple orders, enabling tracking of sales transactions managed by each employee.
Payment	The Payment entity is included to accurately record and manage all financial transactions associated with the purchases made by customers. It captures essential details such as the total price of orders, the current status of each payment (to track if a payment is pending, completed, or refunded), and the type of payment used (like credit card, cash, or online payment platforms). This inclusion is crucial for customer service, and strategic business analysis.	The Payment entity is directly related to the order entity through the OrderID as a foreign key, establishing a one-to-one relationship. This linkage ensures that every payment is matched to its corresponding order, enabling detailed financial tracking, reporting, and analysis. It facilitates the reconciliation of financial data, supports the auditing process, and enhances the understanding of purchasing behaviors and revenue streams.
Order	The Order entity is crucial within the Video Game Store Management Database for recording and managing customer purchase details. It captures essential information such as the customer's	The Order entity is directly related to the Game entity through an associative entity OrderedDetails due to the many-to-many

	identification, the employee managing the sale, and the order date. This entity facilitates sales transaction tracking, providing insights into customer buying trends and store performance. It plays a vital role in sales analysis, inventory control, and understanding customer preferences, which are critical for enhancing store operations and customer service strategies.	relationships. Many orders may be created and could contain multiple games. The Order entity also relates to the Employee entity and Customer entity with many to one relationship, as one customer and one employee can create many orders while one order only belongs to one customer and employee.
OrderDetails	The OrderedDetails entity is included in the database to track the specifics of each item within an order, such as the quantity of each game purchased and the total price for those items. This detailed tracking enables accurate inventory management, financial reporting, and insights into customer preferences and sales trends.	The OrderDetails entity has a many-to-one relationship with both the Order and Game entities, meaning that a single order and a single game can be associated with multiple records in OrderDetails, while each OrderDetails record is linked to only one order and one game.
Game	The Game entity in the database design is crucial for several reasons. Firstly, it serves as a fundamental building block for organizing and managing information related to individual games within the system. By storing attributes such as GameID, GenreID, PublisherID, DeveloperID, GameName, Price, and Release Year, the game entity provides a comprehensive representation of each game's unique characteristics. Additionally, by centralizing game-related data in a dedicated entity, it facilitates efficient querying, reporting, and maintenance tasks, enhancing the overall functionality and usability of the database.	The Game entity is directly related to the Order entity through an associate entity OrderedDetails due to many-to-many relationship. Many orders may be created and could contain multiple games. The Game entity is also directly related to the Platform entity through an associate entity GamePlatform due to many-to-many relationship. As many games can be published on many platforms, and many platforms can provide many games. The Game entity has many to one relationship to the entity Developer and Publisher through GameID, as one developer and one publisher can develop many games. The Game entity also has one to one relationship with entity Inventory through Game ID because it specifies the record of how many

		copies of each game currently available in the store.
Genre	The Genre entity is included to categorize video games based on their genre type. This entity is crucial for aiding customers in finding games of their preferred genre easily. By maintaining a list of genres and associating games with them, the database enhances the browsing experience for customers, and supports targeted marketing strategies based on genre preferences.	Genre maintains a one-to-many relationship with the Preference entity through Genre ID, indicating that a single genre can garner preferences from multiple customers, while each customer can exhibit preferences for only one genre.
Inventory	The Inventory entity in the database is crucial for managing the stock levels and availability of games. By storing attributes such as InventoryID, GameID, and Quantity, the Inventory entity provides a comprehensive representation of the quantity of each game available in stock. This entity allows for efficient tracking of inventory levels, enabling businesses to monitor stock availability, manage replenishment processes, and fulfill customer orders promptly.	The Inventory entity has one to one relationship with entity Game through Game ID because it specifies the record of how many copies of each game currently available in the store.
Platform	The Platform entity is essential in a video games database to represent the various game platforms on which games can be played. It helps in understanding market trends, such as the popularity of specific platforms and the distribution of games across different platforms. The inclusion of the Platform entity allows for detailed analytics on platform-specific game performance, aiding publishers and developers in strategic decision-making regarding game releases and updates.	The Platform entity is directly related to the Game entity through an associate entity GamePlatform due to many-to-many relationship. Many games may be created and maintained on the platform.
GamePlatform	The GamePlatform entity is included to manage the relationship between video games and the platforms they are available on. This entity is crucial for aiding customers in finding games compatible with their gaming platform of choice. By maintaining records of which games are available on which platforms, the database enhances the browsing experience for customers, and supports targeted marketing strategies based on platform preferences.	The GamePlatform entity relates to the Platform entity and Game entity with many-to-one relationship as it ties GameID to Game entity and PlatformID to Platform entity.
Publisher	The Publisher entity is included to store information about the companies or individuals who publish	Publisher entity exhibits a one-to-many relationship with

	video games. This entity is crucial for tracking the publishers associated with each game and for understanding the market dynamics and relationships between publishers and customers. By maintaining records on publishers, including their names, the database facilitates market analysis, and personalized recommendations based on publisher preferences.	both the Game and Preference entities, highlighting that a single publisher can release multiple games, and each game can be associated with a single publisher. Additionally, the same publisher can be favored by multiple customers, as expressed in their preferences.
Developer	The Developer entity is included to store information about the developers responsible for creating specific games. This entity is crucial for facilitating a better understanding of the developer's impact on the quality, popularity, and consumer preferences of games.	Developer entity has a one-to-many relationship with the Game entity, signifying that a single developer can be responsible for multiple games. Each game is tied to one Developer, which allows for aggregation and analysis of sales by developer, aiding in the understanding of market trends and developer performance.