**National University of Computer and Emerging Sciences**

****

**Group Members**

|  |  |
| --- | --- |
| **Name** | **Roll Number** |
| Muhammad Hanzala | 22F-3896 |
| Talha Abdullah Bangyal | 22F-3194 |

**Course Instructor**

Ms. Areeba Waseem

**Project Name**

Duarcade – "Level Up. Play Bold. Duarcade."

Table of Contents

[Deliverable 1 Report: Game Store Database Design 2](#_Toc197641441)

[1. Objective 2](#_Toc197641442)

[2. Schema Design Overview 2](#_Toc197641443)

[2.1. Users Table 2](#_Toc197641444)

[2.2. Games Table 3](#_Toc197641445)

[2.3. GameImages Table 3](#_Toc197641446)

[2.4. Inventory Table 3](#_Toc197641447)

[2.5. Orders Table 4](#_Toc197641448)

[2.6. OrderItems Table 4](#_Toc197641449)

[2.7. Payments Table 4](#_Toc197641450)

[3. Entity Relationships 4](#_Toc197641451)

[4. Normalization 5](#_Toc197641452)

[5. SQL Enhancements 5](#_Toc197641453)

[5.1. Indexes 5](#_Toc197641454)

[5.2. Views 5](#_Toc197641455)

[5.3. Trigger 5](#_Toc197641456)

[6. Summary 6](#_Toc197641457)

# Deliverable 1 Report: Game Store Database Design

## 1. Objective

The objective of this delivery is to design a normalized and relational database schema for an online game store. The database supports functionalities including user management, product listing, order processing, payment tracking, and inventory management.

## 2. Schema Design Overview

### 2.1. Users Table

Manages user account information, including admin and customer profiles.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| user\_id | INT (PK) | Unique identifier |
| username | VARCHAR(50) | Unique username |
| email | VARCHAR(100) | User's email |
| password | VARCHAR(255) | Hashed password |
| role | ENUM | 'customer' or 'admin' |
| first\_name | VARCHAR(50) | User's first name |
| last\_name | VARCHAR(50) | User's last name |
| profile\_image | VARCHAR(255) | Path/URL to profile image |

### 2.2. Games Table

Stores information about games available in the store.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| game\_id | INT (PK) | Unique identifier |
| title | VARCHAR(100) | Game title |
| description | TEXT | Game description |
| actual\_price | DECIMAL(10,2) | Original price |
| discounted\_price | DECIMAL(10,2) | Discounted price |
| genre | VARCHAR(50) | Game genre |
| platform | VARCHAR(50) | Game platform |
| cover\_image | VARCHAR(255) | Cover image URL or path |

### 2.3. GameImages Table

Stores multiple images per game.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| image\_id | INT (PK) | Unique identifier |
| game\_id | INT (FK) | References Games(game\_id) |
| image\_url | VARCHAR(255) | Image URL or path |

### 2.4. Inventory Table

Tracks stock quantities of each game.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| game\_id | INT (PK, FK) | Game reference |
| stock\_quantity | INT | Current stock level |

### 2.5. Orders Table

Records on user orders.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| order\_id | INT (PK) | Unique identifier |
| user\_id | INT (FK) | References Users(user\_id) |
| order\_date | DATETIME | Date of order |
| total\_amount | DECIMAL(10,2) | Total order amount |
| status | ENUM | Order status |

### 2.6. OrderItems Table

Line items are within each order.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| order\_item\_id | INT (PK) | Unique identifier |
| order\_id | INT (FK) | References Orders(order\_id) |
| game\_id | INT (FK) | References Games(game\_id) |
| quantity | INT | Quantity ordered |
| price\_at\_purchase | DECIMAL(10,2) | Price at the time of purchase |

### 2.7. Payments Table

Tracks payment status for orders.

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| payment\_id | INT (PK) | Unique identifier |
| order\_id | INT (FK) | References Orders(order\_id) |
| payment\_date | DATETIME | Date of payment |
| payment\_status | ENUM | Payment state |
| payment\_method | VARCHAR(50) | Payment method used |

## 3. Entity Relationships

* One User → Many Orders
* One Order → Many OrderItems
* One Game → Many OrderItems
* One Order → One Payment
* One Game → One Inventory
* One Game → Many GameImages

## 4. Normalization

The schema is normalized up to Third Normal Form (3NF):

* 1NF: All attributes are atomic.
* 2NF: Partial dependencies removed using OrderItems.
* 3NF: All non-key attributes depend only on the primary key.

## 5. SQL Enhancements

### 5.1. Indexes

CREATE INDEX idx\_user\_email ON Users(email);

CREATE INDEX idx\_order\_user ON Orders(user\_id);

CREATE INDEX idx\_payment\_order ON Payments(order\_id);

### 5.2. Views

CREATE VIEW OrderSummary AS

SELECT

o.order\_id,

CONCAT(u.first\_name, ' ', u.last\_name) AS customer\_name,

o.order\_date,

o.total\_amount,

o.status,

p.payment\_status,

p.payment\_method

FROM Orders o

JOIN Users u ON o.user\_id = u.user\_id

LEFT JOIN Payments p ON o.order\_id = p.order\_id;

### 5.3. Trigger

DELIMITER $$

CREATE TRIGGER trg\_update\_inventory\_after\_order

AFTER INSERT ON OrderItems

FOR EACH ROW

BEGIN

UPDATE Inventory

SET stock\_quantity = stock\_quantity - NEW.quantity

WHERE game\_id = NEW.game\_id;

END$$

DELIMITER ;

## 6. Summary

This scheme provides a scalable and well-structured foundation for an online game store application, ensuring data integrity, flexibility in reporting, and efficient query performance through normalization, indexing, and automation via triggers.