CS4347 Database Systems
Final Project Deliverable 1

MMO Virtual Item Marketplace

Group Members: Alexis Kaufman, Amila Haque, Brandon Maweu, Justin Jeirles, Kashvi Asokan, Brian Pham, Quintin Mitchell-Reyes, Ilan Ali, Shahreen Iqbal

Description

Title: MMO Virtual Item Marketplace

GitHub repository details: https://github.com/BNP200000/4347-005-MMO_Marketplace.git

Team Members:

- Alexis Kaufman: will be the frontend lead, whose duties will include setting up the development environment using React TSX, working on the frontend elements of the project, and onboarding other frontend team members through the project setup.
- Amila Haque: will act alongside Kashvi as a project manager, helping schedule meetings and delegating tasks throughout the project. Amila will also work in the database portion of the backend but has skills that they will apply to the front end, such as having familiarity with React and art.
- Brandon Maweu: will be assisting in the backend development for the project.
- Justin Jeirles will be working on backend development and assisting with front-end development when needed.
- Kashvi Asokan: will be one of the project managers responsible for setting deadlines and assigning tasks, and will step in wherever else is needed.
- Brian Pham: will be working on the backend development for the project.
- Quintin Mitchell-Reyes: will assist with frontend development and implementing game universe-related ideas into the marketplace, and will create simple art to display to the user.
- Ilan Ali will work on backend development and assist in implementing ideas into the marketplace.
- Shahreen Iqbal: will work on the front end and assist with the back end along with any graphics required for the project.
- Everybody can assist with the database code itself when needed.

Our Motivation

Our team formed due to a mutual enjoyment of video games and game development. As video game players ourselves, we wanted to try a hand at implementing functions from one. Since we have an interest in this topic, we expect this to show in our final product, as we believe it is a pleasure to work on something you are passionate about. We appreciate the flexibility of the professor in choosing our topic. The final product can be considered for use as a component of a new/existing game.

Project Timeline and Delegation of Tasks

Our team meets regularly to decide how to delegate tasks and discuss the implementation of our ideas. Our first meeting after Deliverable 1 was released was on September 29th, in which we determined which group members would work on specific tasks, and set a team deadline of October 6th to do research for the background section. All members of the team contributed to the research, while Amila and Ilan were assigned to work on the EER diagram. Quintin worked on the schema diagram, and Alexis, Brandon, and Brian were assigned to work on creating and populating the databases. Justin and Shahreen were assigned to work on the query execution, and Kashvi worked on the description, introduction, and references. We then met on October 8th to go over the specifics of the database so everyone could start working on their assigned parts.

Introduction

This project centers on building a virtual item marketplace for an MMO (massively multiplayer online) game. There are many games out there that have their own marketplaces, and we wanted to make one of our own to create a secure and user-friendly platform that would enhance the user's experience. While other marketplaces may face some challenges with protecting players' information because of how their marketplace was developed and stored within the game itself, our marketplace is completely separate from the game itself and thus adds an extra layer of security. This means that even if the main game is hacked, the information stored in the marketplace database is completely safe. We also aim to make implementation easier, which makes it faster for other games to implement this marketplace into their own game without much hassle. Although this means that our marketplace won't come with features that would be unique to a specific game, we are providing the base database and basic features that come with an MMO marketplace, making it easier for games to simply add or edit a few things to make it cater to their game specifically.

Background and Related Work

[10 POINTS] 2. Background and Related Work: You can use any style for comparison: Paragraph-based, tables, etc. Please include properly cited references in IEEE paper referencing format. (You may see a referencing example in the sample IEEE paper in URL: https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf)

FIFA Marketplace (2008) [1]

- The FIFA Marketplace is a good example of an in-game marketplace. This is where players of the game FIFA can buy or sell items available in the game, such as player cards or consumables, using real money. There are a few different filters that players can use to sort through items, such as the Rarity filter or the Price Range. Players can also

search through different categories to browse different items. In addition, FIFA Marketplace has a search filter that allows the player to find exactly what they're looking for. We will be implementing filters into our database as well, but they will differ from the FIFA Marketplace filters as we will tailor the filters to our specific content. While our database will also allow players to trade items for currency, we will not be using real-world currency and instead will be using an in-game currency that players earn in some way.

Second Life Marketplace (2010) [2]

- Second Life Marketplace is a virtual marketplace where users can buy and sell virtual goods such as clothing, buildings, vehicles, and even services. This marketplace complements the game Second Life which is a 3D online space where players can explore, socialize, and create their own digital items and environments. The marketplace is extensive, and users can create their own items for sale. The Marketplace is a key part of the game's economy, allowing users to buy, sell, and trade virtual goods, ranging from clothing and avatars to buildings, vehicles, and even services like scripting or custom designs. The marketplace also has its own currency, which is reflected in the game, called the Linden Dollar (L).

Unity Asset Store (November 2010) [3]

- The Unity Asset Store is a good example of a digital marketplace that facilitates the selling and purchasing of assets for game developers or other creative professionals. It provides a platform where users can access a wide variety of assets from 3D models, VFX, and even entire game systems which can accelerate development workflows. The site lets users list and sell assets that other users can directly import into their Unity development environment. The site itself uses a structured database to maintain detailed information about each asset and the store contains important metadata regarding titles, descriptions, tags, and system requirements. In addition, it provides a useful search and filtering option where users can search for assets by using keywords, pricing, and price range. Lastly, it also manages transactions, licensing, and user access rights because when users purchase an asset it grants the user a license to download and use that asset for example. The Unity Asset store is very similar to what we are attempting to do with the MMO marketplace, like keeping track of transactions and allowing users to list their own items/assets. One of the biggest differences between our MMO marketplace and the Unity Asset store is that the Unity Asset store requires less concurrency control since users are buying digital rights to use an asset while in the MMO marketplace, you're buying digital items. It is more similar to a virtual economy since users would be buying a specific item hence the need for locking mechanisms to prevent two users from buying the same item at the same time.

Steam Community Market (December 12, 2012) [4]

- The Steam Community Market is a virtual marketplace within the Steam platform – a digital game distribution service and store developed by Valve – and is one of the largest

digital distribution platforms for PC gaming, facilitating the trading of in-game items. It uses a digital currency, "Funds", stored in a user's "Steam Wallet" and purchased using real-life money. Within the Steam Community Market, users can purchase any item they wish for any game. One distinct similarity that our MMO Marketplace can draw from this example is that, like the Steam Community Market, our database also exists as an external, virtual market for users to purchase MMO-related items – that can be filtered into categories – and translated back into the game. What makes the MMO Marketplace different from the Community Market is that the MMO Marketplace is essentially a virtual market for one game; The Steam Community Market is a multi-game virtual market, meaning that users can purchase any items that exist within that particular game.

Eorzea Database (FFXIV, The Lodestone) (August 2013) [5]

- The Eorzea Database is a user-friendly way of accessing most of FFXIV's data on items, raids, quests, etc. I'll focus on items here. Like in our database brainstorming, items are split into several categories, including arms/weapons, armor, accessories, materials, and more. Every type of item we listed has some sort of parallel in the Eorzea Database, which could be used to justify our item tuples. However, this public database only lists types of items, not specific instances or player inventories.
- Every item has information on it, much of which depends on what type of item it is. For example, the terrible magic staff pictured above lists its damage, cast time, stat modifiers, which classes can equip it, how much it can be upgraded, crafting and repair information, and sell prices for NPC vendors and players. Although it can be sold by players on the game's marketboard, only NPC vendors and quests are listed as sources of items on this website. This still may be comparable to any search functions or relationships between items and sellers in our database.
- In comparison to the staff, a fish item would have little of the same info. Instead, it has a description used for the fisher class, a note if it's able to be displayed in a player's home, and a list of crafting recipes it's used in.
- Overall, this source might be best compared to our item entity, its attributes, and how it's related to other entities such as characters, classes, and maybe transactions.

BitSkins (July 21st, 2015) [6]

- BitSkins is a marketplace that allows players from Counter-Strike: Global Offensive (CS) and Dota 2 to buy, sell, and trade in-game items for real-world currency. Because BitSkins involves items from MMOs, many of the attributes and filtered views of its database can apply to ours as well. BitSkins also showcases a transparent pricing system, enabling users to see the current market value of items, which aids in making informed buying and selling decisions. In virtual item trading, where prices can fluctuate based on demand and rarity, this transparent pricing can be a functionality that we implement in our marketplace. However, our virtual marketplace differs in that we would only be using our in-game monetary system, so the security measures that BitSkins needs to employ for real-world transactions aren't as necessary for our database. Furthermore, unlike

BitSkins, which mainly focuses on a general trade of skins for all users, our marketplace will incorporate class-specific items, so our database will reflect how certain items are restricted to specific player classes.

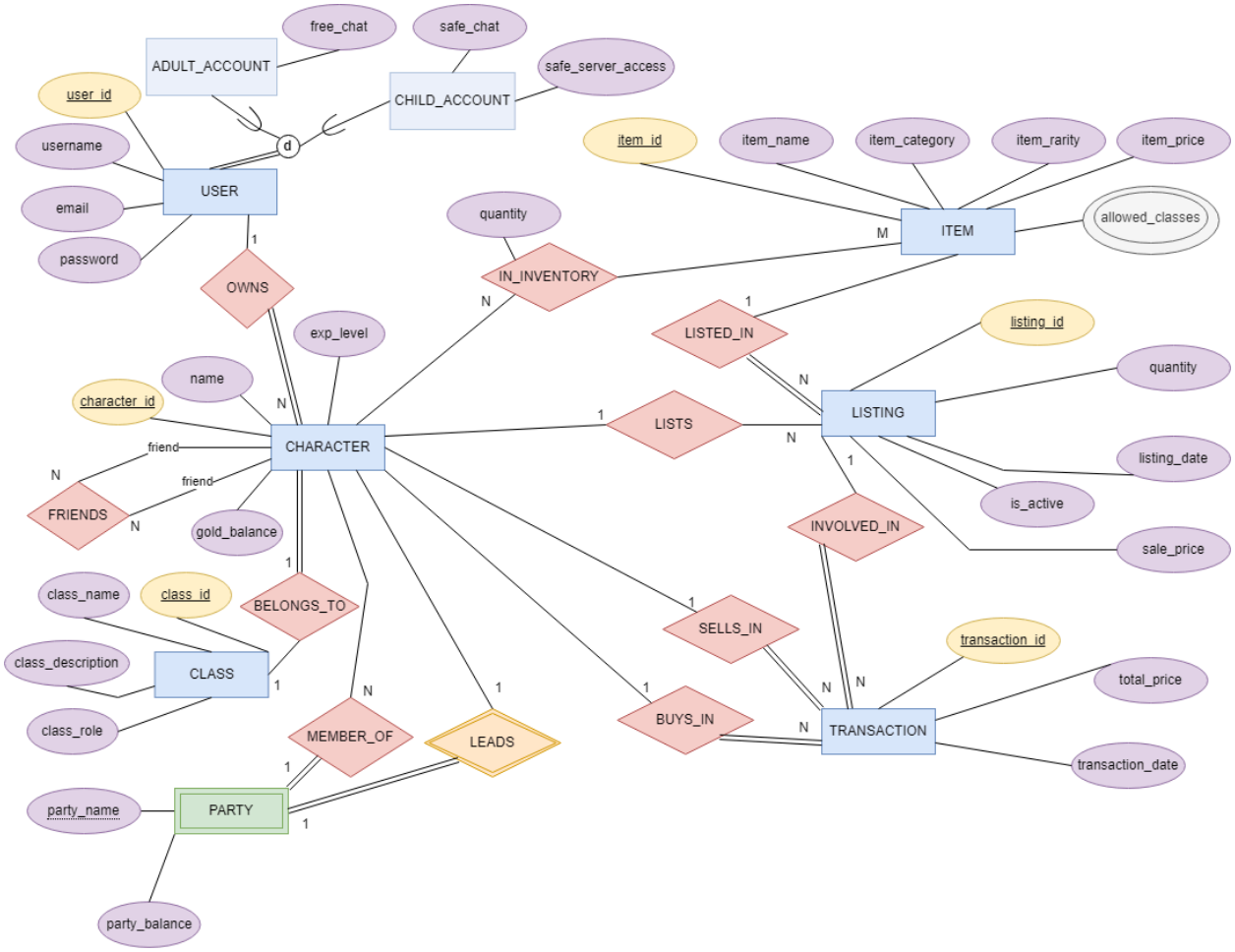
CryptoPunks (2017) [7]

- CryptoPunks is an example of a system that has digital collectibles, specifically on the blockchain. While our project has a simpler scope and does not involve NFTs, this site lets users buy, sell, and trade digital assets in the form of 24x24 pixel art characters. Furthermore, each character has many attributes, such that characters can have special eyes, facial hair, mouth, ears, and more. Then, the rarity of any given character is measured by the combination and frequency of these attributes; in the image, one is much rarer than the other due to its many rare traits. The system thusly allows players to determine the uniqueness and value of each "Punk", so the database schema must not only accommodate ownership and trade history, but also it must calculate and compare "Total Rarity Scores". These characters all come together to form a digital marketplace similar to the one we are designing, such that different items (or in this case, characters), will have special values and not all be worth the same amount of currency.

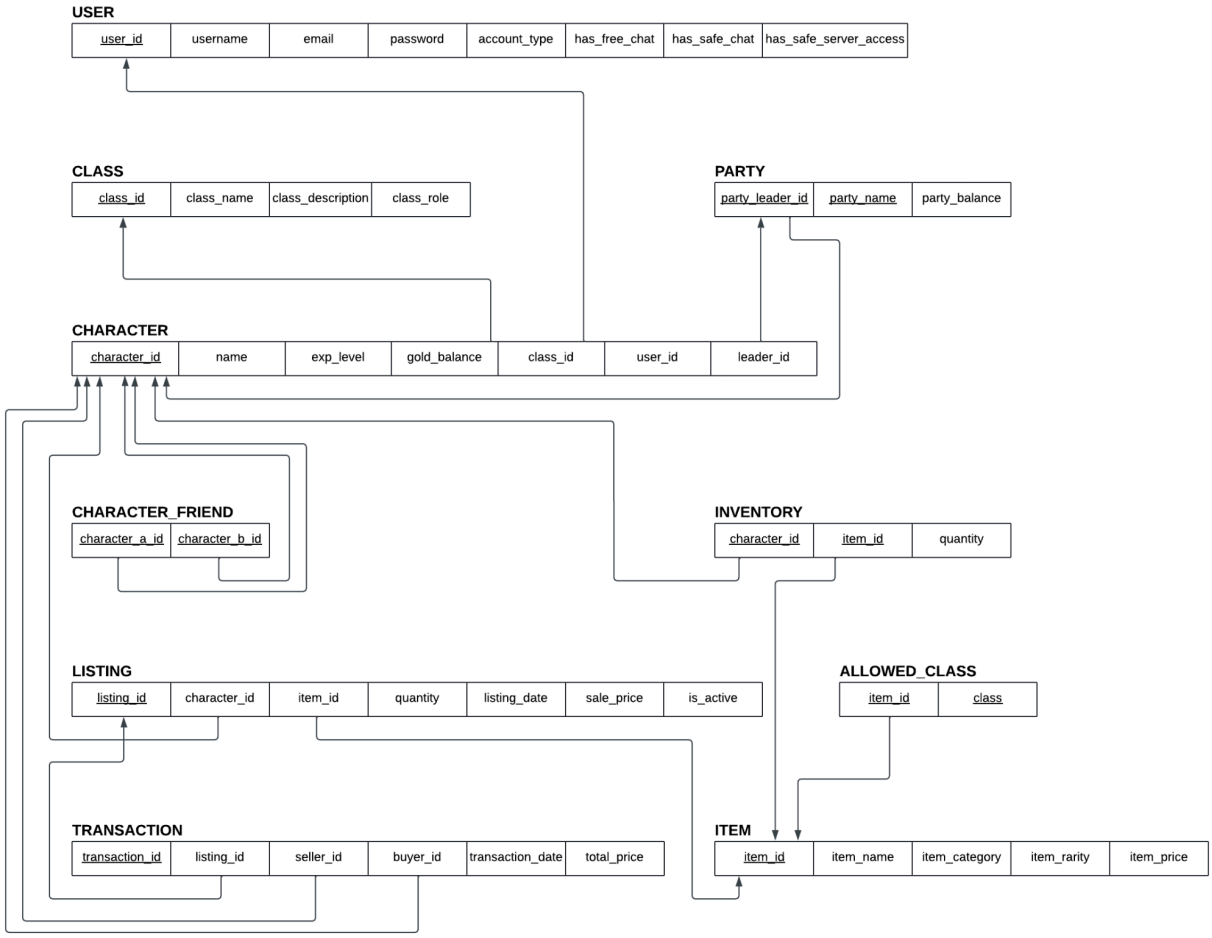
Runescape Marketplace (June 7, 2021) [8]

- This source is a good example of an MMO marketplace. Despite the game being released in 2001, the marketplace is a relatively new addition to the game and is a replacement for several older features in the game. Our database is similar to the marketplace because our goal is to create a shop where players can buy or sell items. The items will be categorized and have different attributes. The marketplace in the game is very similar to what we plan to create because it sells several things like consumables, armor/costumes, animations, pets, etc. The marketplace uses in-game called oddments. There are several seasonal currencies too, but that will not be something we add to our database shop.

EER Conceptual Data Model Design



Relational Data Model Design



Create your Database and Populate

Database Creation Statement

```
CREATE TABLE
    IF NOT EXISTS "USER" (
        user_id VARCHAR(25) PRIMARY KEY,
        username VARCHAR(25) NOT NULL,
        password VARCHAR(25) NOT NULL,
        email VARCHAR(25),
        account_type VARCHAR(25),
        has_free_chat BOOLEAN,
        has_safe_chat BOOLEAN,
        has_safe_server_access BOOLEAN
    );

CREATE TABLE
    IF NOT EXISTS "CLASS" (
        class_id VARCHAR(25) PRIMARY KEY,
        class_name VARCHAR(25),
        class_description VARCHAR(25),
        class_role VARCHAR(25)
    );

CREATE TABLE
    IF NOT EXISTS "CHARACTER" (
        character_id VARCHAR(25) PRIMARY KEY,
        exp_level INT,
        character_name VARCHAR(25),
        gold_balance INT,
        owner_id VARCHAR(25) NOT NULL,
        character_class VARCHAR(25),
        leader_id VARCHAR(25),
        FOREIGN KEY (owner_id) REFERENCES "USER" (user_id) ON DELETE CASCADE ON UPDATE CASCADE, -- "OWNS" relationship, 1 to N
        FOREIGN KEY (character_class) REFERENCES "CLASS" (class_id) ON DELETE CASCADE ON UPDATE CASCADE -- "BELONGS TO" relationship, 1 to 1
        -- NOTE; The foreign key constraint for "leader_id" is located in "insert_tables.sql" to avoid circular dependency. It relies on "PARTY"'s party_leader.
    );

CREATE TABLE
    IF NOT EXISTS "CHARACTER_FRIEND" (
        character_a_id VARCHAR(25),
        character_b_id VARCHAR(25),
        PRIMARY KEY (character_a_id, character_b_id),
        FOREIGN KEY (character_a_id) REFERENCES "CHARACTER" (character_id),
        FOREIGN KEY (character_b_id) REFERENCES "CHARACTER" (character_id)
    );
```

```sql
CREATE TABLE
    IF NOT EXISTS "PARTY" (
        party_name VARCHAR(50) UNIQUE,
        party_leader VARCHAR(25) UNIQUE,
        party_balance INT,
        PRIMARY KEY (party_name, party_leader),
        FOREIGN KEY (party_leader) REFERENCES "CHARACTER" (character_id) ON DELETE CASCADE ON UPDATE CASCADE
    );

CREATE TABLE
    IF NOT EXISTS "ITEM" (
        item_id SERIAL PRIMARY KEY,
        item_name VARCHAR(50) NOT NULL,
        item_category VARCHAR(25),
        item_rarity VARCHAR(25),
        item_price NUMERIC(10, 0),
        allowed_classes TEXT[]
    );

CREATE TABLE
    IF NOT EXISTS "IN_INVENTORY" (
        character_id VARCHAR(25) NOT NULL,
        item_id SERIAL NOT NULL,
        quantity int,
        FOREIGN KEY (character_id) REFERENCES "CHARACTER" (character_id) ON DELETE CASCADE,
        FOREIGN KEY (item_id) REFERENCES "ITEM" (item_id) ON DELETE CASCADE
    );

CREATE TABLE
    IF NOT EXISTS "LISTING" (
        listing_id SERIAL PRIMARY KEY,
        character_id VARCHAR(25) NOT NULL,
        item_id SERIAL NOT NULL,
        quantity int NOT NULL,
        listing_date DATE NOT NULL,
        is_active BOOLEAN,
        sale_price NUMERIC(10, 0),
        FOREIGN KEY (character_id) REFERENCES "CHARACTER" (character_id),
        FOREIGN KEY (item_id) REFERENCES "ITEM" (item_id)
    );

CREATE TABLE
    IF NOT EXISTS "TRANSACTION" (
        transaction_id CHAR(10) PRIMARY KEY,
        listing_id SERIAL,
        seller_id VARCHAR(25) NOT NULL,
        buyer_id VARCHAR(25) NOT NULL,
        total_price INT,
        transaction_date DATE,
        FOREIGN KEY (seller_id) REFERENCES "CHARACTER" (character_id) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (buyer_id) REFERENCES "CHARACTER" (character_id) ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (listing_id) REFERENCES "LISTING" (listing_id) ON DELETE CASCADE ON UPDATE CASCADE
    );
```

Database Population Statement

```
INSERT INTO "USER" (user_id, username, password, email, account_type, has_free_chat, has_safe_chat, has_safe_server_access)
VALUES
    ('1x2y3z456', 'dragonlord', 'pass123', 'dragonlord@example.com', 'adult', TRUE, TRUE, TRUE),
    ('2y3z4a567', 'skywalker', 'star123', 'skywalker@example.com', 'child', FALSE, TRUE, FALSE),
    ('3z4a5b678', 'ironfist', 'punch789', 'ironfist@example.com', 'adult', TRUE, TRUE, TRUE),
    ('4a5b6c789', 'shadowblade', 'shadow456', 'shadowblade@example.com', 'child', FALSE, TRUE, FALSE),
    ('5b6c7d890', 'stormrider', 'rider567', 'stormrider@example.com', 'adult', TRUE, TRUE, TRUE),
    ('6c7d8e901', 'firemage', 'fire321', 'firemage@example.com', 'child', FALSE, TRUE, FALSE),
    ('7d8e9f012', 'windrunner', 'wind111', 'windrunner@example.com', 'adult', TRUE, TRUE, TRUE),
    ('8e9f0g123', 'earthwarden', 'earth222', 'earthwarden@example.com', 'child', FALSE, TRUE, FALSE),
    ('9f0g1h234', 'waterseer', 'water333', 'waterseer@example.com', 'adult', TRUE, TRUE, TRUE),
    ('0g1h2i345', 'lightbringer', 'light444', 'lightbringer@example.com', 'child', FALSE, TRUE, FALSE);


INSERT INTO
    "CLASS" (class_id, class_name, class_description, class_role)
VALUES
    ('1m2n3o456', 'Warrior', 'Frontline fighter', 'Tank'),
    ('2n3o4p567', 'Rogue', 'Stealthy assassin', 'DPS'),
    ('3o4p5q678', 'Mage', 'Master of magic', 'DPS'),
    ('4p5q6r789', 'Cleric', 'Healer of the wounded', 'Healer'),
    ('5q6r7s890', 'Paladin', 'Holy warrior', 'Tank'),
    ('6r7s8t901', 'Ranger', 'Expert marksman', 'DPS'),
    ('7s8t9u012', 'Druid', 'Nature spellcaster', 'Healer'),
    ('8t9u0v123', 'Monk', 'Martial artist', 'DPS'),
    ('9u0v1w234', 'Bard', 'Musical supporter', 'Support'),
    ('0v1w2x345', 'Necromancer', 'Summoner of the dead', 'DPS');

INSERT INTO
    "CHARACTER" (character_id, exp_level, character_name, gold_balance, owner_id, character_class, leader_id)
VALUES
    ('1a2b3c456', 5, 'Alduin', 300, '1x2y3z456', '1m2n3o456', NULL),
    ('2b3c4d567', 3, 'Lyria', 150, '2y3z4a567', '2n3o4p567', '1a2b3c456'),
    ('3c4d5e678', 7, 'Gorath', 500, '3z4a5b678', '3o4p5q678', '1a2b3c456'),
    ('4d5e6f789', 6, 'Serana', 250, '4a5b6c789', '1m2n3o456', NULL),
    ('5e6f7g890', 10, 'Theron', 800, '5b6c7d890', '2n3o4p567', '1a2b3c456'),
    ('6f7g8h901', 4, 'Miraak', 200, '6c7d8e901', '3o4p5q678', '2b3c4d567'),
    ('7g8h9i012', 12, 'Valdrin', 1000, '7d8e9f012', '4p5q6r789', '3c4d5e678'),
    ('8h9i0j123', 2, 'Nalya', 100, '8e9f0g123', '1m2n3o456', '2b3c4d567'),
    ('9i0j1k234', 9, 'Druin', 750, '9f0g1h234', '4p5q6r789', '3c4d5e678'),
    ('0j1k2l345', 8, 'Kael', 600, '0g1h2i345', '5q6r7s890', '1a2b3c456');
```

```
INSERT INTO
    "PARTY" (party_name, party_leader, party_balance)
VALUES
    ('Lyrical', '1a2b3c456', 53422),
    ('Arcane', '4d5e6f789', 3234223),
    ('Dragon Slayer', '2b3c4d567', 1000),
    ('Witch Hunter', '3c4d5e678', 54223),
    ('Tuba Gang', '7g8h9i012', 546),
    ('Stroopwafel', '0j1k2l345', 564),
    ('Birds of Prey', '9i0j1k234', 32),
    ('The Fallen', '6f7g8h901', 7869),
    ('Masked Fools', '5e6f7g890', 435345),
    ('Asgard', '8h9i0j123', 34523);

ALTER TABLE "CHARACTER"
ADD CONSTRAINT FK_leader_id
FOREIGN KEY (leader_id) REFERENCES "PARTY" (party_leader);

INSERT INTO
    "CHARACTER_FRIEND" (character_a_id, character_b_id)
VALUES
    ('1a2b3c456', '2b3c4d567'),
    ('1a2b3c456', '3c4d5e678'),
    ('2b3c4d567', '6f7g8h901'),
    ('3c4d5e678', '9i0j1k234'),
    ('4d5e6f789', '1a2b3c456'),
    ('5e6f7g890', '2b3c4d567'),
    ('6f7g8h901', '7g8h9i012'),
    ('7g8h9i012', '3c4d5e678'),
    ('8h9i0j123', '9i0j1k234'),
    ('0j1k2l345', '5e6f7g890');
```

```sql
INSERT INTO "ITEM" (item_name, item_category, item_rarity, item_price, allowed_classes) VALUES
('Health Potion', 'Consumable', 'Common', 50, ARRAY[
    '1m2n3o456', '2n3o4p567', '3o4p5q678', '4p5q6r789', '5q6r7s890',
    '6r7s8t901', '7s8t9u012', '8t9u0v123', '9u0v1w234', '0v1w2x345'
]),
('Greater Health Potion', 'Consumable', 'Uncommon', 100, ARRAY[
    '1m2n3o456', '2n3o4p567', '3o4p5q678', '4p5q6r789', '5q6r7s890',
    '6r7s8t901', '7s8t9u012', '8t9u0v123', '9u0v1w234', '0v1w2x345'
]),
('Mana Potion', 'Consumable', 'Common', 50, ARRAY[
    '1m2n3o456', '2n3o4p567', '3o4p5q678', '4p5q6r789', '5q6r7s890',
    '6r7s8t901', '7s8t9u012', '8t9u0v123', '9u0v1w234', '0v1w2x345'
]),
('Greater Mana Potion', 'Consumable', 'Uncommon', 100, ARRAY[
    '1m2n3o456', '2n3o4p567', '3o4p5q678', '4p5q6r789', '5q6r7s890',
    '6r7s8t901', '7s8t9u012', '8t9u0v123', '9u0v1w234', '0v1w2x345'
]),
('Steel Dagger', 'Weapon', 'Rare', 600, ARRAY['2n3o4p567', '9u0v1w234', '6r7s8t901']),
('Iron Sword', 'Weapon', 'Uncommon', 300, ARRAY['1m2n3o456', '5q6r7s890', '6r7s8t901']),
('Wooden Staff', 'Weapon', 'Uncommon', 200, ARRAY['3o4p5q678', '0v1w2x345', '7s8t9u012']),
('Cloak of Shadows', 'Armor', 'Rare', 550, ARRAY['2n3o4p567', '6r7s8t901', '9u0v1w234']),
('Iron Helm', 'Armor', 'Uncommon', 250, ARRAY['8t9u0v123', '5q6r7s890']),
('Amulet of Druidic Power', 'Accessory', 'Rare', 300, ARRAY['7s8t9u012', '6r7s8t901']),
('Robe of Arcane Mysteries', 'Armor', 'Uncommon', 200, ARRAY['3o4p5q678', '7s8t9u012', '8t9u0v123']),
('Ring of the Necromancer', 'Accessory', 'Epic', 600, ARRAY['0v1w2x345']),
('Bow of the Silent Hunter', 'Weapon', 'Rare', 250, ARRAY['6r7s8t901', '2n3o4p567']),
('Divine Shield', 'Shield', 'Legendary', 1200, ARRAY['4p5q6r789', '5q6r7s890', '1m2n3o456']),
('Crown of the Bard King', 'Headgear', 'Legendary', 1000, ARRAY['9u0v1w234']),
('Warhammer of the Paladin King', 'Weapon', 'Legendary', 950, ARRAY['5q6r7s890']);
```

```
INSERT INTO "LISTING" (character_id, quantity, listing_date, is_active, sale_price) VALUES
('1a2b3c456', 10, '2024-09-15', TRUE, 150),
('2b3c4d567', 5, '2024-08-23', TRUE, 350),
('3c4d5e678', 1, '2024-10-01', FALSE, 1000),
('4d5e6f789', 2, '2024-09-10', TRUE, 1200),
('5e6f7g890', 3, '2024-08-30', FALSE, 500),
('6f7g8h901', 7, '2024-09-25', TRUE, 250),
('7g8h9i012', 4, '2024-09-02', TRUE, 200),
('8h9i0j123', 12, '2024-07-20', FALSE, 300),
('9i0j1k234', 6, '2024-08-29', TRUE, 950),
('0j1k2l345', 1, '2024-09-18', FALSE, 50),
('1a2b3c456', 4, '2024-08-15', TRUE, 400),
('2b3c4d567', 6, '2024-07-28', FALSE, 180),
('3c4d5e678', 8, '2024-09-30', TRUE, 600),
('4d5e6f789', 9, '2024-08-25', TRUE, 1000),
('5e6f7g890', 10, '2024-09-01', FALSE, 800),
('6f7g8h901', 11, '2024-09-12', TRUE, 50);

INSERT INTO "IN_INVENTORY" (character_id, quantity) VALUES
('1a2b3c456', 32),
('2b3c4d567', 15),
('3c4d5e678', 40),
('4d5e6f789', 22),
('5e6f7g890', 12),
('6f7g8h901', 47),
('7g8h9i012', 29),
('8h9i0j123', 18),
('9i0j1k234', 35),
('0j1k2l345', 41),
('1a2b3c456', 27),
('2b3c4d567', 14),
('3c4d5e678', 38),
('4d5e6f789', 46),
('5e6f7g890', 9),
('6f7g8h901', 44);

INSERT INTO
    "TRANSACTION" (transaction_id, seller_id, buyer_id, transaction_date, total_price)
VALUES
    ('741966', '1a2b3c456', '2b3c4d567', '2024-10-22', 53422),
    ('703221', '2b3c4d567', '1a2b3c456', '2024-10-28', 3234223),
    ('703814', '3c4d5e678', '0j1k2l345', '2024-11-04', 1000),
    ('772759', '0j1k2l345', '5e6f7g890', '2024-11-06', 54223),
    ('744093', '2b3c4d567', '5e6f7g890', '2024-11-12', 546),
    ('733026', '3c4d5e678', '5e6f7g890', '2024-11-23', 564),
    ('773521', '7g8h9i012', '6f7g8h901', '2024-11-30', 32),
    ('781428', '8h9i0j123', '0j1k2l345', '2024-12-14', 7869),
    ('761722', '6f7g8h901', '5e6f7g890', '2024-12-21', 435345),
    ('716844', '5e6f7g890', '7g8h9i012', '2024-12-24', 34523);
```

Resulting Tables

1. USER

```
1  select * from "USER";
```

Data Output | Messages | Notifications

	user_id [PK] character varying (25)	username character varying (25)	password character varying (25)	email character varying (25)	account_type character varying (25)	has_free_chat boolean	has_safe_chat boolean	has_safe_server_access boolean
1	1x2y3z456	dragonlord	pass123	dragonlord@example.com	adult	true	true	true
2	2y3z4a567	skywalker	star123	skywalker@example.com	child	false	true	false
3	3z4a5b678	ironfist	punch789	ironfist@example.com	adult	true	true	true
4	4a5b6c789	shadowblade	shadow456	shadowblade@example.c...	child	false	true	false
5	5b6c7d890	stormrider	rider567	stormrider@example.com	adult	true	true	true
6	6c7d8e901	firemage	fire321	firemage@example.com	child	false	true	false
7	7d8e9f012	windrunner	wind111	windrunner@example.com	adult	true	true	true
8	8e9f0g123	earthwarden	earth222	earthwarden@example.co...	child	false	true	false
9	9f0g1h234	waterseer	water333	waterseer@example.com	adult	true	true	true
10	0g1h2i345	lightbringer	light444	lightbringer@example.com	child	false	true	false

2. CHARACTER

Query | Query History

```
1  select * from "CHARACTER";
```

Data Output | Messages | Notifications

	character_id [PK] character varying (25)	exp_level integer	character_name character varying (25)	gold_balance integer	owner_id character varying (25)	character_class character varying (25)	leader_id character varying (25)
1	1a2b3c456	5	Alduin	300	1x2y3z456	1m2n3o456	[null]
2	2b3c4d567	3	Lyria	150	2y3z4a567	2n3o4p567	1a2b3c456
3	3c4d5e678	7	Gorath	500	3z4a5b678	3o4p5q678	1a2b3c456
4	4d5e6f789	6	Serana	250	4a5b6c789	1m2n3o456	[null]
5	5e6f7g890	10	Theron	800	5b6c7d890	2n3o4p567	1a2b3c456
6	6f7g8h901	4	Miraak	200	6c7d8e901	3o4p5q678	2b3c4d567
7	7g8h9i012	12	Valdrin	1000	7d8e9f012	4p5q6r789	3c4d5e678
8	8h9i0j123	2	Nalya	100	8e9f0g123	1m2n3o456	2b3c4d567
9	9i0j1k234	9	Druin	750	9f0g1h234	4p5q6r789	3c4d5e678
10	0j1k2l345	8	Kael	600	0g1h2i345	5q6r7s890	1a2b3c456

3. CLASS

Query | Query History

```
1  select * from "CLASS";
```

Data Output | Messages | Notifications

	class_id [PK] character varying (25)	class_name character varying (25)	class_description character varying (25)	class_role character varying (25)
1	1m2n3o456	Warrior	Frontline fighter	Tank
2	2n3o4p567	Rogue	Stealthy assassin	DPS
3	3o4p5q678	Mage	Master of magic	DPS
4	4p5q6r789	Cleric	Healer of the wounded	Healer
5	5q6r7s890	Paladin	Holy warrior	Tank
6	6r7s8t901	Ranger	Expert marksman	DPS
7	7s8t9u012	Druid	Nature spellcaster	Healer
8	8t9u0v123	Monk	Martial artist	DPS
9	9u0v1w234	Bard	Musical supporter	Support
10	0v1w2x345	Necromancer	Summoner of the dead	DPS

4. PARTY

Query | Query History

```
1  select * from "PARTY";
```

Data Output | Messages | Notifications

	party_name [PK] character varying (50)	party_leader [PK] character varying (25)	party_balance integer
1	Lyrical	1a2b3c456	53422
2	Arcane	4d5e6f789	3234223
3	Dragon Slayer	2b3c4d567	1000
4	Witch Hunter	3c4d5e678	54223
5	Tuba Gang	7g8h9i012	546
6	Stroopwafel	0j1k2l345	564
7	Birds of Prey	9i0j1k234	32
8	The Fallen	6f7g8h901	7869
9	Masked Fools	5e6f7g890	435345
10	Asgard	8h9i0j123	34523

5. CHARACTER_FRIEND

Query | Query History

```
1  select * from "CHARACTER_FRIEND";
```

Data Output | Messages | Notifications

	character_a_id [PK] character varying (25)	character_b_id [PK] character varying (25)
1	1a2b3c456	2b3c4d567
2	1a2b3c456	3c4d5e678
3	2b3c4d567	6f7g8h901
4	3c4d5e678	9i0j1k234
5	4d5e6f789	1a2b3c456
6	5e6f7g890	2b3c4d567
7	6f7g8h901	7g8h9i012
8	7g8h9i012	3c4d5e678
9	8h9i0j123	9i0j1k234
10	0j1k2l345	5e6f7g890

6. ITEM

Query | Query History

```
1  select * from "ITEM";
```

Data Output | Messages | Notifications

	item_id [PK] integer	item_name character varying (50)	item_category character varying (25)	item_rarity character varying (25)	item_price numeric (10)	allowed_classes text[]
1	1	Health Potion	Consumable	Common	50	{1m2n3o456,2n3o4p567,3o4p5q678,4p5q6r789,5q6r7s890,6r7s8t901,7s8t9u012,8t9u0v123,9u0v1w234,0v1w2x3...
2	2	Greater Health Potion	Consumable	Uncommon	100	{1m2n3o456,2n3o4p567,3o4p5q678,4p5q6r789,5q6r7s890,6r7s8t901,7s8t9u012,8t9u0v123,9u0v1w234,0v1w2x3...
3	3	Mana Potion	Consumable	Common	50	{1m2n3o456,2n3o4p567,3o4p5q678,4p5q6r789,5q6r7s890,6r7s8t901,7s8t9u012,8t9u0v123,9u0v1w234,0v1w2x3...
4	4	Greater Mana Potion	Consumable	Uncommon	100	{1m2n3o456,2n3o4p567,3o4p5q678,4p5q6r789,5q6r7s890,6r7s8t901,7s8t9u012,8t9u0v123,9u0v1w234,0v1w2x3...
5	5	Steel Dagger	Weapon	Rare	600	{2n3o4p567,9u0v1w234,6r7s8t901}
6	6	Iron Sword	Weapon	Uncommon	300	{1m2n3o456,5q6r7s890,6r7s8t901}
7	7	Wooden Staff	Weapon	Uncommon	200	{3o4p5q678,0v1w2x345,7s8t9u012}
8	8	Cloak of Shadows	Armor	Rare	550	{2n3o4p567,6r7s8t901,9u0v1w234}
9	9	Iron Helm	Armor	Uncommon	250	{8t9u0v123,5q6r7s890}
10	10	Amulet of Druidic Power	Accessory	Rare	300	{7s8t9u012,6r7s8t901}
11	11	Robe of Arcane Mysteries	Armor	Uncommon	200	{3o4p5q678,7s8t9u012,8t9u0v123}
12	12	Ring of the Necromancer	Accessory	Epic	600	{0v1w2x345}
13	13	Bow of the Silent Hunter	Weapon	Rare	250	{6r7s8t901,2n3o4p567}
14	14	Divine Shield	Shield	Legendary	1200	{4p5q6r789,5q6r7s890,1m2n3o456}
15	15	Crown of the Bard King	Headgear	Legendary	1000	{9u0v1w234}
16	16	Warhammer of the Paladin King	Weapon	Legendary	950	{5q6r7s890}

7. LISTING

Query | Query History

```
1  select * from "LISTING";
```

Data Output | Messages | Notifications

	listing_id [PK] integer	character_id character varying (25)	item_id integer	quantity integer	listing_date date	is_active boolean	sale_price numeric (10)
1	1	1a2b3c456	1	10	2024-09-15	true	150
2	2	2b3c4d567	2	5	2024-08-23	true	350
3	3	3c4d5e678	3	1	2024-10-01	false	1000
4	4	4d5e6f789	4	2	2024-09-10	true	1200
5	5	5e6f7g890	5	3	2024-08-30	false	500
6	6	6f7g8h901	6	7	2024-09-25	true	250
7	7	7g8h9i012	7	4	2024-09-02	true	200
8	8	8h9i0j123	8	12	2024-07-20	false	300
9	9	9i0j1k234	9	6	2024-08-29	true	950
10	10	0j1k2l345	10	1	2024-09-18	false	50
11	11	1a2b3c456	11	4	2024-08-15	true	400
12	12	2b3c4d567	12	6	2024-07-28	false	180
13	13	3c4d5e678	13	8	2024-09-30	true	600
14	14	4d5e6f789	14	9	2024-08-25	true	1000
15	15	5e6f7g890	15	10	2024-09-01	false	800
16	16	6f7g8h901	16	11	2024-09-12	true	50

8. IN_INVENTORY

Query | Query History

```
1  select * from "IN_INVENTORY";
```

Data Output | Messages | Notifications

	character_id character varying (25)	item_id integer	quantity integer
1	1a2b3c456	1	32
2	2b3c4d567	2	15
3	3c4d5e678	3	40
4	4d5e6f789	4	22
5	5e6f7g890	5	12
6	6f7g8h901	6	47
7	7g8h9i012	7	29
8	8h9i0j123	8	18
9	9i0j1k234	9	35
10	0j1k2l345	10	41
11	1a2b3c456	11	27
12	2b3c4d567	12	14
13	3c4d5e678	13	38
14	4d5e6f789	14	46
15	5e6f7g890	15	9
16	6f7g8h901	16	44

9. TRANSACTION

Query | Query History

```
1  select * from "TRANSACTION";
```

Data Output | Messages | Notifications

	transaction_id [PK] character (10)	listing_id integer	seller_id character varying (25)	buyer_id character varying (25)	total_price integer	transaction_date date
1	741966	1	1a2b3c456	2b3c4d567	53422	2024-10-22
2	703221	2	2b3c4d567	1a2b3c456	3234223	2024-10-28
3	703814	3	3c4d5e678	0j1k2l345	1000	2024-11-04
4	772759	4	0j1k2l345	5e6f7g890	54223	2024-11-06
5	744093	5	2b3c4d567	5e6f7g890	546	2024-11-12
6	733026	6	3c4d5e678	5e6f7g890	564	2024-11-23
7	773521	7	7g8h9i012	6f7g8h901	32	2024-11-30
8	781428	8	8h9i0j123	0j1k2l345	7869	2024-12-14
9	761722	9	6f7g8h901	5e6f7g890	435345	2024-12-21
10	716844	10	5e6f7g890	7g8h9i012	34523	2024-12-24

Database Query Execution

USERS Table

1. Display

Query | Query History | Scratch Pad

```
SELECT * FROM "USER";
```

Data Output | Messages | Notifications

	user_id [PK] character varying (25)	username character varying (25)	password character varying (25)	email character varying (25)	account_type character varying (25)	has_free_chat boolean	has_safe_ch boolean
1	1x2y3z456	dragonlord	pass123	dragonlord@example.com	adult	true	true
2	2y3z4a567	skywalker	star123	skywalker@example.com	child	false	true
3	3z4a5b678	ironfist	punch789	ironfist@example.com	adult	true	true
4	4a5b6c789	shadowblade	shadow456	shadowblade@example.c...	child	false	true
5	5b6c7d890	stormrider	rider567	stormrider@example.com	adult	true	true
6	6c7d8e901	firemage	fire321	firemage@example.com	child	false	true
7	7d8e9f012	windrunner	wind111	windrunner@example.com	adult	true	true
8	8e9f0g123	earthwarden	earth222	earthwarden@example.co...	child	false	true
9	9f0g1h234	waterseer	water333	waterseer@example.com	adult	true	true
10	0g1h2i345	lightbringer	light444	lightbringer@example.com	child	false	true

2. Insert

Query | Query History | Scratch Pad

```
INSERT INTO "USER" (user_id, username, password, email, account_type, has_free_c
VALUES
    ('1h2i3j456', 'nightstalker', 'dark999', 'nightstalker@example.com', 'adult'

SELECT * FROM "USER"
```

Data Output | Messages | Notifications

	user_id [PK] character varying (25)	username character varying (25)	password character varying (25)	email character varying (25)	account_type character varying (25)	has_free_chat boolean	has_safe boolean
3	3z4a5b678	ironfist	punch789	ironfist@example.com	adult	true	true
4	4a5b6c789	shadowblade	shadow456	shadowblade@example.c...	child	false	true
5	5b6c7d890	stormrider	rider567	stormrider@example.com	adult	true	true
6	6c7d8e901	firemage	fire321	firemage@example.com	child	false	true
7	7d8e9f012	windrunner	wind111	windrunner@example.com	adult	true	true
8	8e9f0g123	earthwarden	earth222	earthwarden@example.co...	child	false	true
9	9f0g1h234	waterseer	water333	waterseer@example.com	adult	true	true
10	0g1h2i345	lightbringer	light444	lightbringer@example.com	child	false	true
11	1h2i3j456	nightstalker	dark999	nightstalker@example.com	adult	true	true

3. Delete

Query | Query History | Scratch Pad

```
DELETE FROM "USER"
WHERE user_id = '1h2i3j456';

SELECT * FROM "USER"
```

Data Output | Messages | Notifications

	user_id [PK] character varying (25)	username character varying (25)	password character varying (25)	email character varying (25)	account_type character varying (25)	has_free_chat boolean	has_safe boolean
2	2y3z4a567	skywalker	star123	skywalker@example.com	child	false	true
3	3z4a5b678	ironfist	punch789	ironfist@example.com	adult	true	true
4	4a5b6c789	shadowblade	shadow456	shadowblade@example.c…	child	false	true
5	5b6c7d890	stormrider	rider567	stormrider@example.com	adult	true	true
6	6c7d8e901	firemage	fire321	firemage@example.com	child	false	true
7	7d8e9f012	windrunner	wind111	windrunner@example.com	adult	true	true
8	8e9f0g123	earthwarden	earth222	earthwarden@example.co…	child	false	true
9	9f0g1h234	waterseer	water333	waterseer@example.com	adult	true	true
10	0g1h2i345	lightbringer	light444	lightbringer@example.com	child	false	true

4. Update

Query | Query History | Scratch Pad

```
UPDATE "USER"
SET email = 'newemail@example.com', account_type = 'child'
WHERE user_id = '1h2i3j456';

SELECT * FROM "USER"
```

Data Output | Messages | Notifications

	user_id [PK] character varying (25)	username character varying (25)	password character varying (25)	email character varying (25)	account_type character varying (25)	has_free_chat boolean	has_safe boolean
3	3z4a5b678	ironfist	punch789	ironfist@example.com	adult	true	true
4	4a5b6c789	shadowblade	shadow456	shadowblade@example.c…	child	false	true
5	5b6c7d890	stormrider	rider567	stormrider@example.com	adult	true	true
6	6c7d8e901	firemage	fire321	firemage@example.com	child	false	true
7	7d8e9f012	windrunner	wind111	windrunner@example.com	adult	true	true
8	8e9f0g123	earthwarden	earth222	earthwarden@example.co…	child	false	true
9	9f0g1h234	waterseer	water333	waterseer@example.com	adult	true	true
10	0g1h2i345	lightbringer	light444	lightbringer@example.com	child	false	true
11	1h2i3j456	nightstalker	dark999	newemail@example.com	child	true	true

CLASS Table:

1. Display

Query | Query History

```
1  SELECT * FROM "CLASS";
```

Data Output | Messages | Notifications

	class_id [PK] character varying (25)	class_name character varying (25)	class_description character varying (25)	class_role character varying (25)
1	1m2n3o456	Warrior	Frontline fighter	Tank
2	2n3o4p567	Rogue	Stealthy assassin	DPS
3	3o4p5q678	Mage	Master of magic	DPS
4	4p5q6r789	Cleric	Healer of the wounded	Healer
5	5q6r7s890	Paladin	Holy warrior	Tank
6	6r7s8t901	Ranger	Expert marksman	DPS
7	7s8t9u012	Druid	Nature spellcaster	Healer
8	8t9u0v123	Monk	Martial artist	DPS
9	9u0v1w234	Bard	Musical supporter	Support
10	0v1w2x345	Necromancer	Summoner of the dead	DPS

2. Insert

Query | Query History

```
1  INSERT INTO "CLASS" (class_id, class_name, class_description, class_role)
2  VALUES ('1a2b3c456', 'Warlock', 'User of dark powers', 'DPS');
3
4  SELECT * FROM "CLASS";
```

Data Output | Messages | Notifications

	class_id [PK] character varying (25)	class_name character varying (25)	class_description character varying (25)	class_role character varying (25)
1	1m2n3o456	Warrior	Frontline fighter	Tank
2	2n3o4p567	Rogue	Stealthy assassin	DPS
3	3o4p5q678	Mage	Master of magic	DPS
4	4p5q6r789	Cleric	Healer of the wounded	Healer
5	5q6r7s890	Paladin	Holy warrior	Tank
6	6r7s8t901	Ranger	Expert marksman	DPS
7	7s8t9u012	Druid	Nature spellcaster	Healer
8	8t9u0v123	Monk	Martial artist	DPS
9	9u0v1w234	Bard	Musical supporter	Support
10	0v1w2x345	Necromancer	Summoner of the dead	DPS
11	1a2b3c456	Warlock	User of dark powers	DPS

3. Delete

Query | Query History

```
1  DELETE FROM "CLASS"
2  WHERE class_name = 'Warlock';
3
4  SELECT * FROM "CLASS";
```

Data Output | Messages | Notifications

	class_id [PK] character varying (25)	class_name character varying (25)	class_description character varying (25)	class_role character varying (25)
1	1m2n3o456	Warrior	Frontline fighter	Tank
2	2n3o4p567	Rogue	Stealthy assassin	DPS
3	3o4p5q678	Mage	Master of magic	DPS
4	4p5q6r789	Cleric	Healer of the wounded	Healer
5	5q6r7s890	Paladin	Holy warrior	Tank
6	6r7s8t901	Ranger	Expert marksman	DPS
7	7s8t9u012	Druid	Nature spellcaster	Healer
8	8t9u0v123	Monk	Martial artist	DPS
9	9u0v1w234	Bard	Musical supporter	Support
10	0v1w2x345	Necromancer	Summoner of the dead	DPS

4. Update

Query | Query History

```
1  UPDATE "CLASS"
2  SET class_description = 'Master of elemental magic'
3  WHERE class_name = 'Mage';
4
```

Data Output | Messages | Notifications

	class_id [PK] character varying (25)	class_name character varying (25)	class_description character varying (25)	class_role character varying (25)
1	1m2n3o456	Warrior	Frontline fighter	Tank
2	2n3o4p567	Rogue	Stealthy assassin	DPS
3	4p5q6r789	Cleric	Healer of the wounded	Healer
4	5q6r7s890	Paladin	Holy warrior	Tank
5	6r7s8t901	Ranger	Expert marksman	DPS
6	7s8t9u012	Druid	Nature spellcaster	Healer
7	8t9u0v123	Monk	Martial artist	DPS
8	9u0v1w234	Bard	Musical supporter	Support
9	0v1w2x345	Necromancer	Summoner of the dead	DPS
10	3o4p5q678	Mage	Master of elemental magic	DPS

CHARACTER Table:

1. Display + Insert:

Query | Query History | Scratch Pad

```
INSERT INTO "CHARACTER" (character_id, exp_level, character_name, gold_balance,
VALUES ('1k2l3m456', 4, 'Sylas', 400, '0g1h2i345', '1m2n3o456', NULL);
```

Data Output | Messages | Notifications

	character_id [PK] character varying (25)	exp_level integer	character_name character varying (25)	gold_balance integer	owner_id character varying (25)	character_class character varying (25)	leader_id character varying (25)
1	2b3c4d567	3	Lyria	150	2y3z4a567	2n3o4p567	1a2b3c456
2	3c4d5e678	7	Gorath	500	3z4a5b678	3o4p5q678	1a2b3c456
3	4d5e6f789	6	Serana	250	4a5b6c789	1m2n3o456	[null]
4	5e6f7g890	10	Theron	800	5b6c7d890	2n3o4p567	1a2b3c456
5	6f7g8h901	4	Miraak	200	6c7d8e901	3o4p5q678	2b3c4d567
6	7g8h9i012	12	Valdrin	1000	7d8e9f012	4p5q6r789	3c4d5e678
7	8h9i0j123	2	Nalya	100	8e9f0g123	1m2n3o456	2b3c4d567
8	9i0j1k234	9	Druin	750	9f0g1h234	4p5q6r789	3c4d5e678
9	0j1k2l345	8	Kael	600	0g1h2i345	5q6r7s890	1a2b3c456
10	1a2b3c456	5	Alduin	350	1x2y3z456	1m2n3o456	[null]
11	1k2l3m456	4	Sylas	400	0g1h2i345	1m2n3o456	[null]

2. DELETE

Query | Query History | Scratch Pad

```
DELETE FROM "CHARACTER"
WHERE character_name = 'Sylas';

SELECT * FROM "CHARACTER";
```

Data Output | Messages | Notifications

	character_id [PK] character varying (25)	exp_level integer	character_name character varying (25)	gold_balance integer	owner_id character varying (25)	character_class character varying (25)	leader_id character varying (25)
1	2b3c4d567	3	Lyria	150	2y3z4a567	2n3o4p567	1a2b3c456
2	3c4d5e678	7	Gorath	500	3z4a5b678	3o4p5q678	1a2b3c456
3	4d5e6f789	6	Serana	250	4a5b6c789	1m2n3o456	[null]
4	5e6f7g890	10	Theron	800	5b6c7d890	2n3o4p567	1a2b3c456
5	6f7g8h901	4	Miraak	200	6c7d8e901	3o4p5q678	2b3c4d567
6	7g8h9i012	12	Valdrin	1000	7d8e9f012	4p5q6r789	3c4d5e678
7	8h9i0j123	2	Nalya	100	8e9f0g123	1m2n3o456	2b3c4d567
8	9i0j1k234	9	Druin	750	9f0g1h234	4p5q6r789	3c4d5e678
9	0j1k2l345	8	Kael	600	0g1h2i345	5q6r7s890	1a2b3c456
10	1a2b3c456	5	Alduin	350	1x2y3z456	1m2n3o456	[null]

3. UPDATE

Query | Query History | Scratch Pad

```
UPDATE "CHARACTER"
SET gold_balance = gold_balance + 50
WHERE gold_balance > 500;
```

Data Output | Messages | Notifications

	character_id [PK] character varying (25)	exp_level integer	character_name character varying (25)	gold_balance integer	owner_id character varying (25)	character_class character varying (25)	leader_id character varying (25)
1	2b3c4d567	3	Lyria	150	2y3z4a567	2n3o4p567	1a2b3c456
2	3c4d5e678	7	Gorath	500	3z4a5b678	3o4p5q678	1a2b3c456
3	4d5e6f789	6	Serana	250	4a5b6c789	1m2n3o456	[null]
4	6f7g8h901	4	Miraak	200	6c7d8e901	3o4p5q678	2b3c4d567
5	8h9i0j123	2	Nalya	100	8e9f0g123	1m2n3o456	2b3c4d567
6	1a2b3c456	5	Alduin	350	1x2y3z456	1m2n3o456	[null]
7	5e6f7g890	10	Theron	850	5b6c7d890	2n3o4p567	1a2b3c456
8	7g8h9i012	12	Valdrin	1050	7d8e9f012	4p5q6r789	3c4d5e678
9	9i0j1k234	9	Druin	800	9f0g1h234	4p5q6r789	3c4d5e678
10	0j1k2l345	8	Kael	650	0g1h2i345	5q6r7s890	1a2b3c456

PARTY Table

1. DISPLAY

Query | Query History

```
SELECT *
FROM "PARTY"
WHERE party_balance > 5000;
```

Data Output | Messages | Notifications

	party_name [PK] character varying (50)	party_leader [PK] character varying (25)	party_balance integer
1	Lyrical	1a2b3c456	53422
2	Arcane	4d5e6f789	3234223
3	Witch Hunter	3c4d5e678	54223
4	The Fallen	6f7g8h901	7869
5	Masked Fools	5e6f7g890	435345
6	Asgard	8h9i0j123	34523

2. Insert

Query | Query History

```
1 INSERT INTO "PARTY" (party_name, party_leader, party_balance)
2 VALUES ('Warriors of Light', '1k2l3m456', 15000);
```

Data Output | Messages | Notifications

	party_name [PK] character varying (50)	party_leader [PK] character varying (25)	party_balance integer
1	Lyrical	1a2b3c456	53422
2	Arcane	4d5e6f789	3234223
3	Dragon Slayer	2b3c4d567	1000
4	Witch Hunter	3c4d5e678	54223
5	Tuba Gang	7g8h9i012	546
6	Stroopwafel	0j1k2l345	564
7	Birds of Prey	9i0j1k234	32
8	The Fallen	6f7g8h901	7869
9	Masked Fools	5e6f7g890	435345
10	Asgard	8h9i0j123	34523
11	Warriors of Light	1k2l3m456	15000

3. **DELETE**

Query Query History

```
1 DELETE FROM "PARTY"
2 WHERE party_name = 'Mystic Alliance';
3
```

Data Output Messages Notifications

SQL

	party_name [PK] character varying (50)	party_leader [PK] character varying (25)	party_balance integer
1	Lyrical	1a2b3c456	53422
2	Arcane	4d5e6f789	3234223
3	Dragon Slayer	2b3c4d567	1000
4	Witch Hunter	3c4d5e678	54223
5	Tuba Gang	7g8h9i012	546
6	Stroopwafel	0j1k2l345	564
7	Birds of Prey	9i0j1k234	32
8	The Fallen	6f7g8h901	7869
9	Masked Fools	5e6f7g890	435345
10	Asgard	8h9i0j123	34523
11	Warriors of Light	1k2l3m456	15000

4. **UPDATE**

```
UPDATE "PARTY"
SET party_balance = party_balance - 20;
```

Data Output | Messages | Notifications

	party_name [PK] character varying (50)	party_leader [PK] character varying (25)	party_balance integer
1	Lyrical	1a2b3c456	53402
2	Arcane	4d5e6f789	3234203
3	Dragon Slayer	2b3c4d567	980
4	Witch Hunter	3c4d5e678	54203
5	Tuba Gang	7g8h9i012	526
6	Stroopwafel	0j1k2l345	544
7	Birds of Prey	9i0j1k234	12
8	The Fallen	6f7g8h901	7849
9	Masked Fools	5e6f7g890	435325
10	Asgard	8h9i0j123	34503
11	Warriors of Light	1k2l3m456	14980

CHARACTER_FRIEND Table

1. **DISPLAY**

Query | Query History

```
SELECT *
FROM "CHARACTER_FRIEND"
WHERE character_a_id = '1a2b3c456';
```

Data Output | Messages | Notifications

	character_a_id [PK] character varying (25)	character_b_id [PK] character varying (25)
1	1a2b3c456	2b3c4d567
2	1a2b3c456	3c4d5e678

2. **INSERT**

Query Query History

```
1 INSERT INTO "CHARACTER_FRIEND" (character_a_id, character_b_id)
2 VALUES ('1a2b3c456', '4d5e6f789');
3
```

Data Output Messages Notifications

	character_a_id [PK] character varying (25)	character_b_id [PK] character varying (25)
1	2b3c4d567	6f7g8h901
2	3c4d5e678	9i0j1k234
3	4d5e6f789	1a2b3c456
4	5e6f7g890	2b3c4d567
5	6f7g8h901	7g8h9i012
6	7g8h9i012	3c4d5e678
7	8h9i0j123	9i0j1k234
8	0j1k2l345	5e6f7g890
9	1a2b3c456	2b3c4d567
10	1a2b3c456	3c4d5e678
11	1a2b3c456	4d5e6f789

3. DELETE

Query Query History

```
1 DELETE FROM "CHARACTER_FRIEND"
2 WHERE character_a_id = '1a2b3c456' AND character_b_id = '2b3c4d567
3
```

Data Output Messages Notifications

Graph Visualiser

	character_a_id [PK] character varying (25)	character_b_id [PK] character varying (25)
1	2b3c4d567	6f7g8h901
2	3c4d5e678	9i0j1k234
3	4d5e6f789	1a2b3c456
4	5e6f7g890	2b3c4d567
5	6f7g8h901	7g8h9i012
6	7g8h9i012	3c4d5e678
7	8h9i0j123	9i0j1k234
8	0j1k2l345	5e6f7g890
9	1a2b3c456	3c4d5e678
10	1a2b3c456	4d5e6f789

4. **UPDATE**

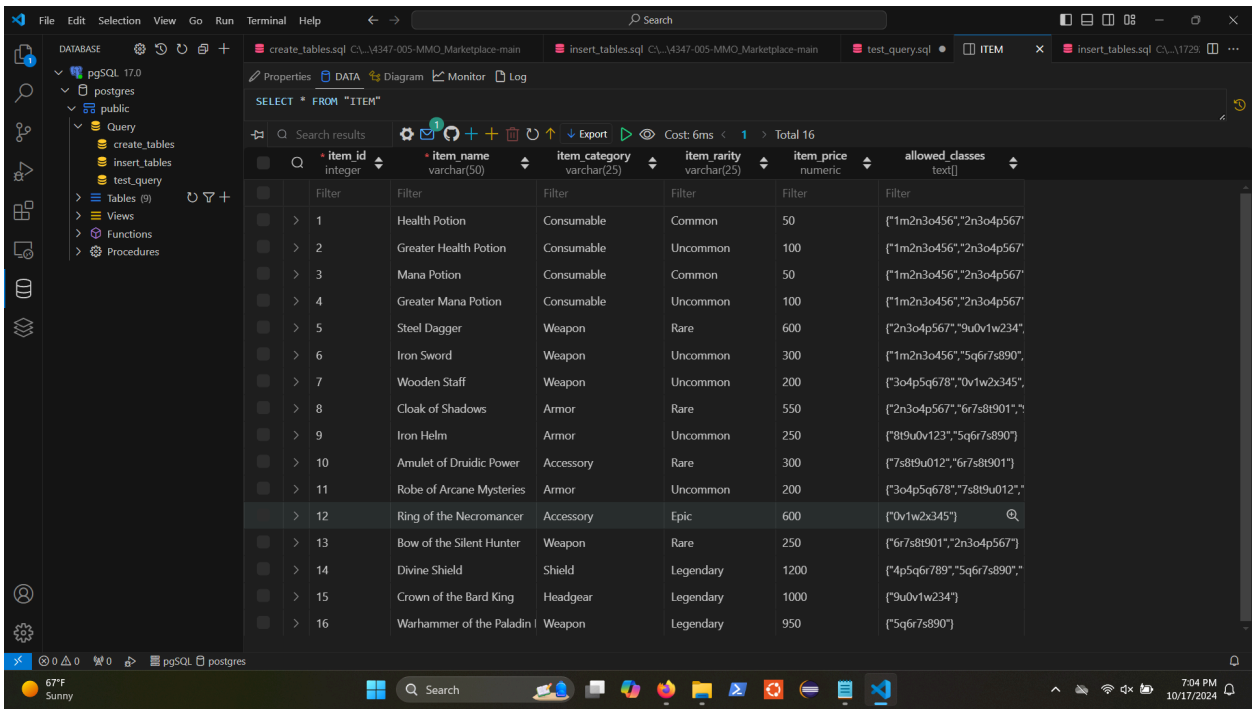
```sql
UPDATE "CHARACTER_FRIEND"
SET character_b_id = '9i0j1k234'
WHERE character_a_id = '1a2b3c456' AND character_b_id = '2b3c4d567';
```

Data Output | Messages | Notifications

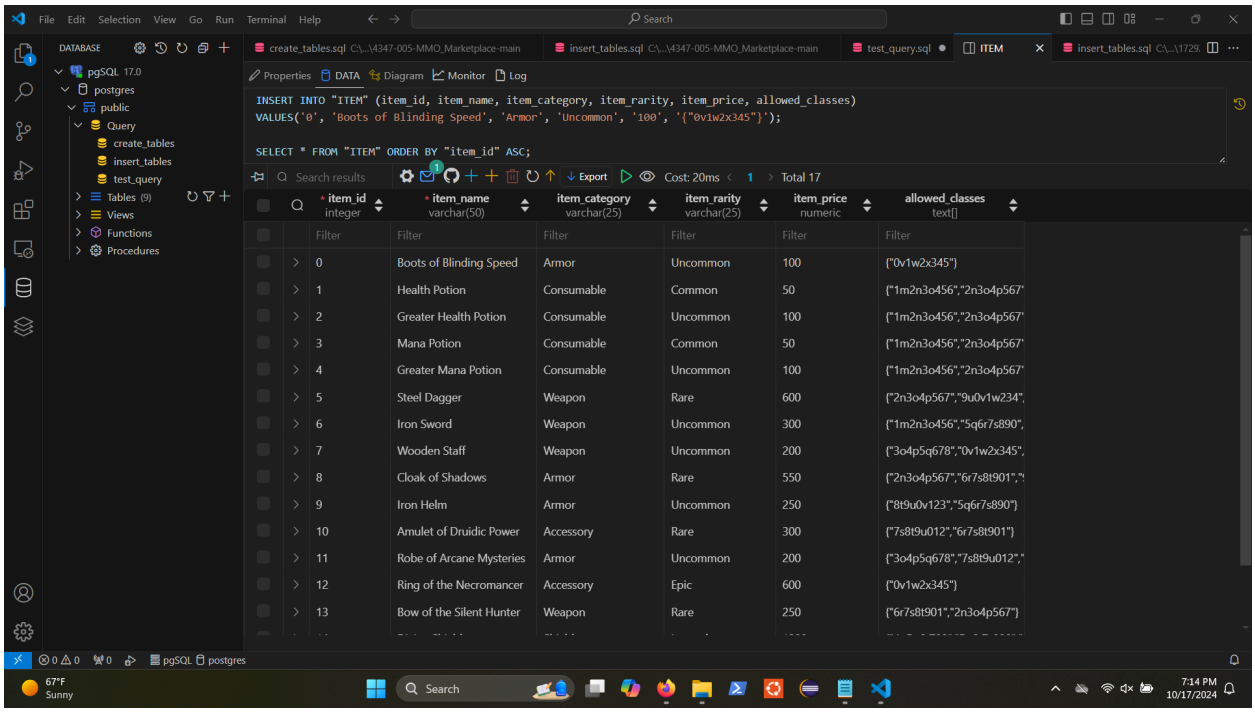
	character_a_id [PK] character varying (25)	character_b_id [PK] character varying (25)
1	2b3c4d567	6f7g8h901
2	3c4d5e678	9i0j1k234
3	4d5e6f789	1a2b3c456
4	5e6f7g890	2b3c4d567
5	6f7g8h901	7g8h9i012
6	7g8h9i012	3c4d5e678
7	8h9i0j123	9i0j1k234
8	0j1k2l345	5e6f7g890
9	1a2b3c456	3c4d5e678
10	1a2b3c456	4d5e6f789

ITEM Table

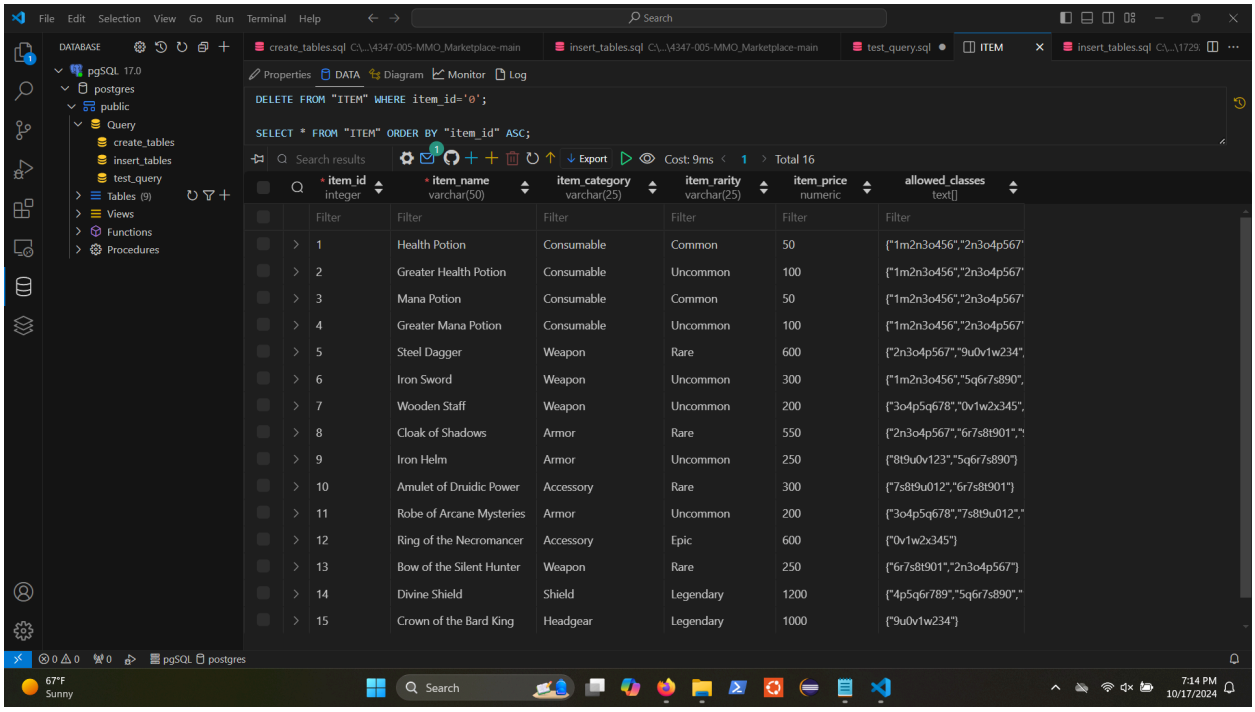
1. DISPLAY



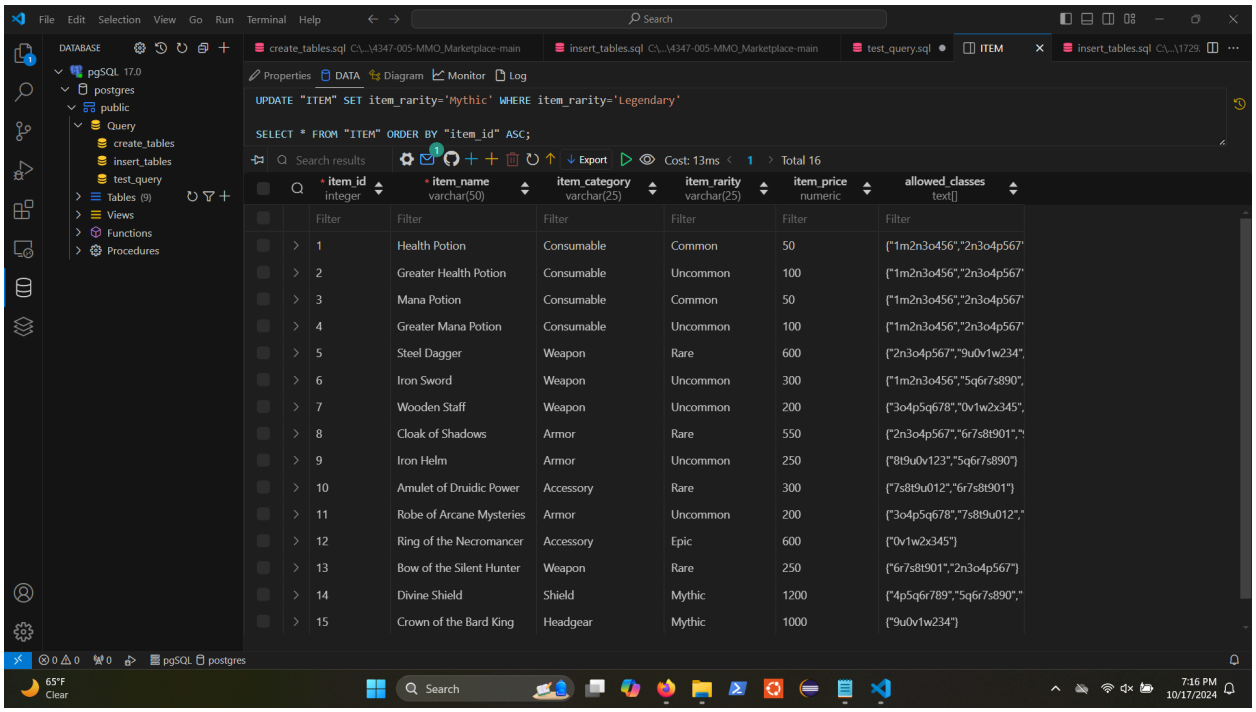
2. INSERT



3. DELETE

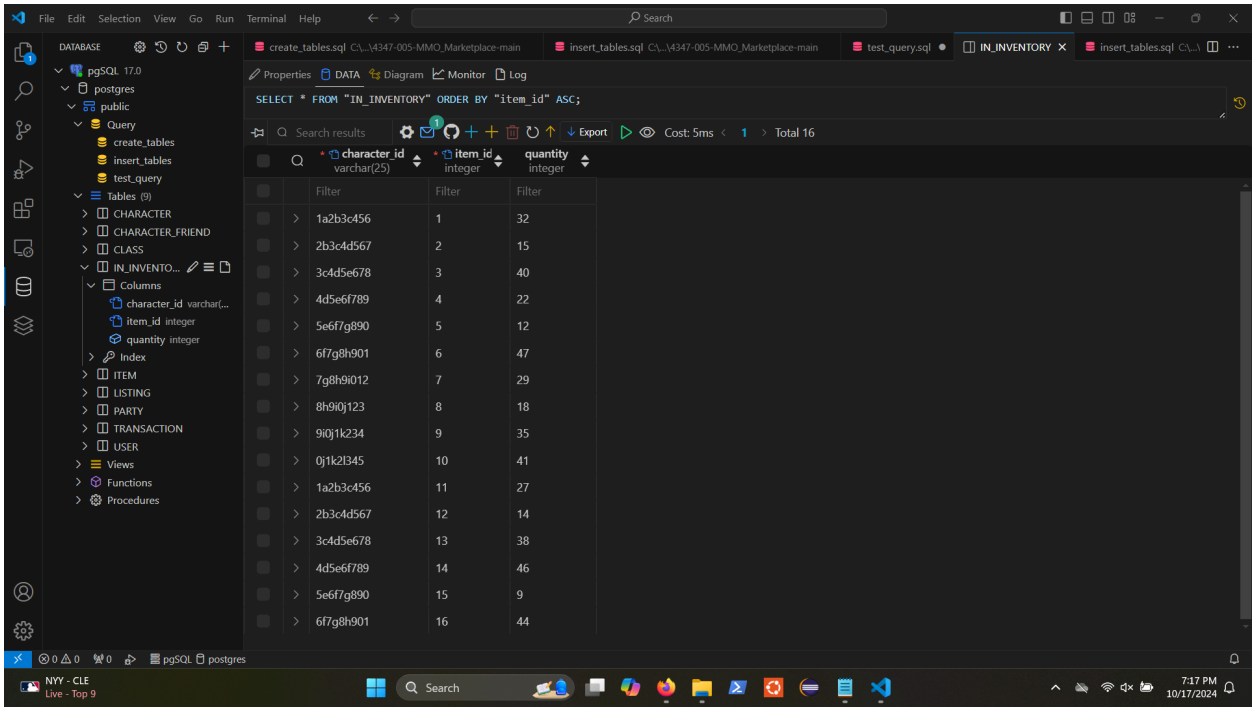


4. UPDATE

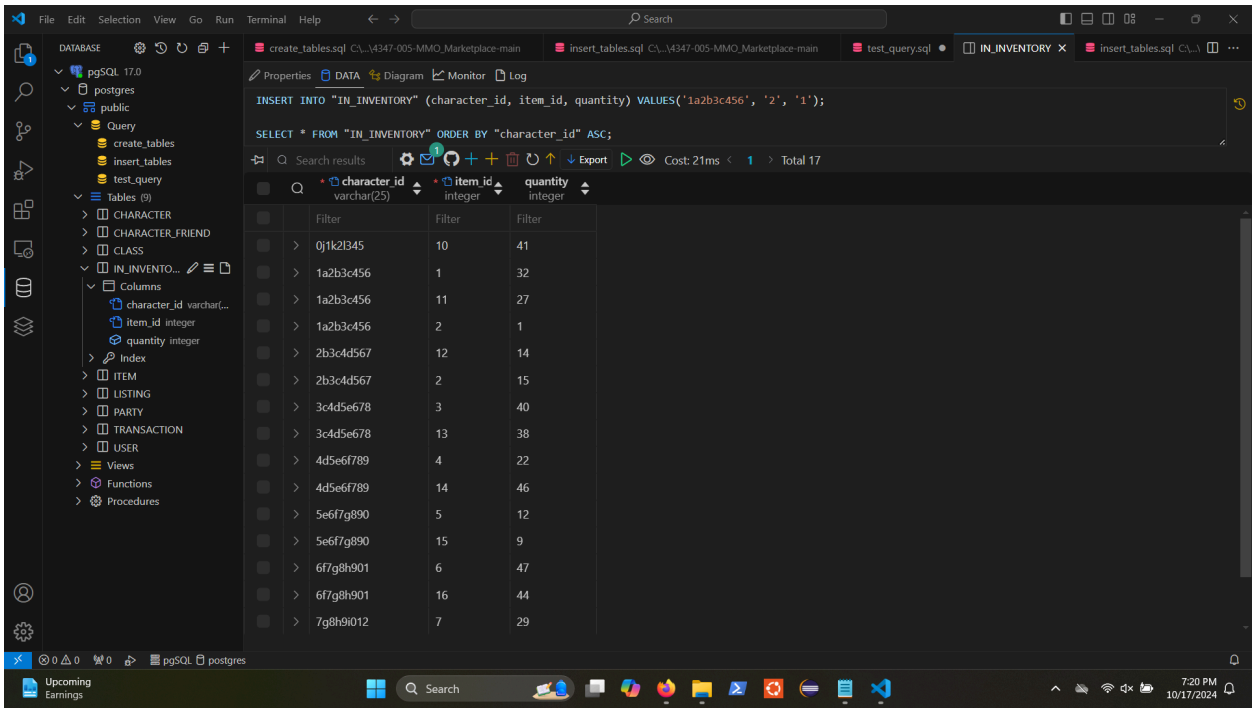


IN_INVENTORY Table

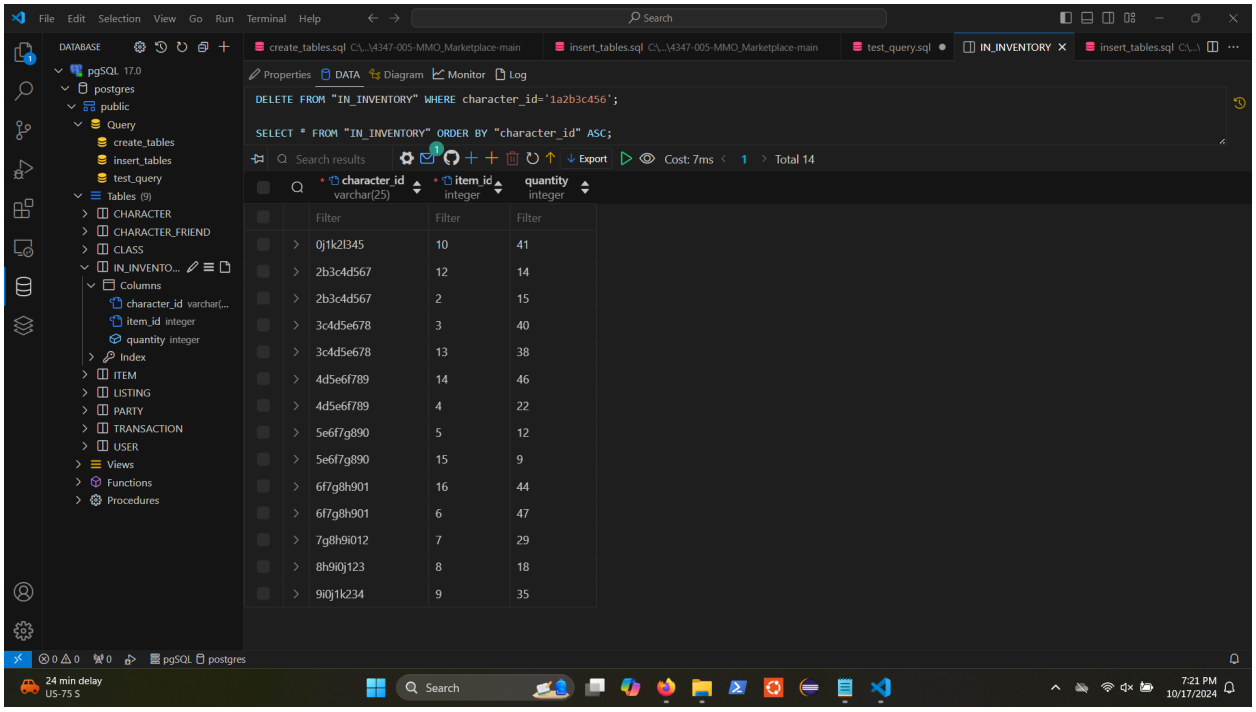
1. DISPLAY



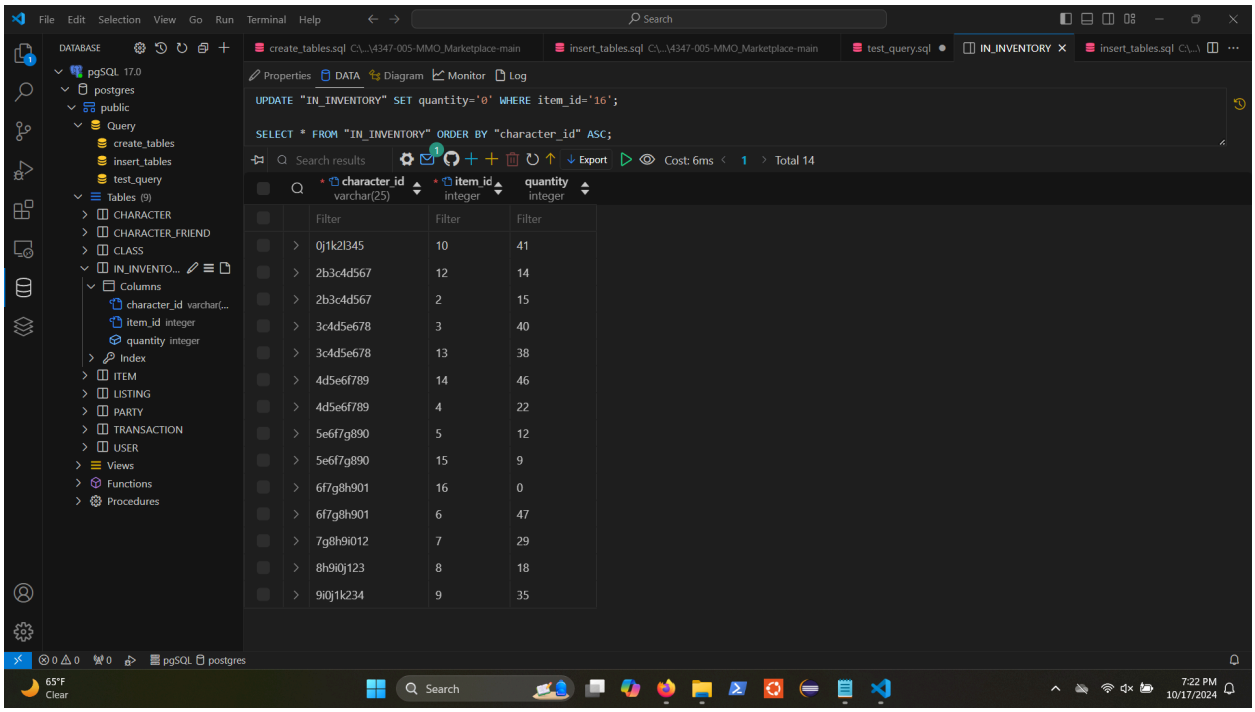
2. INSERT



3. DELETE

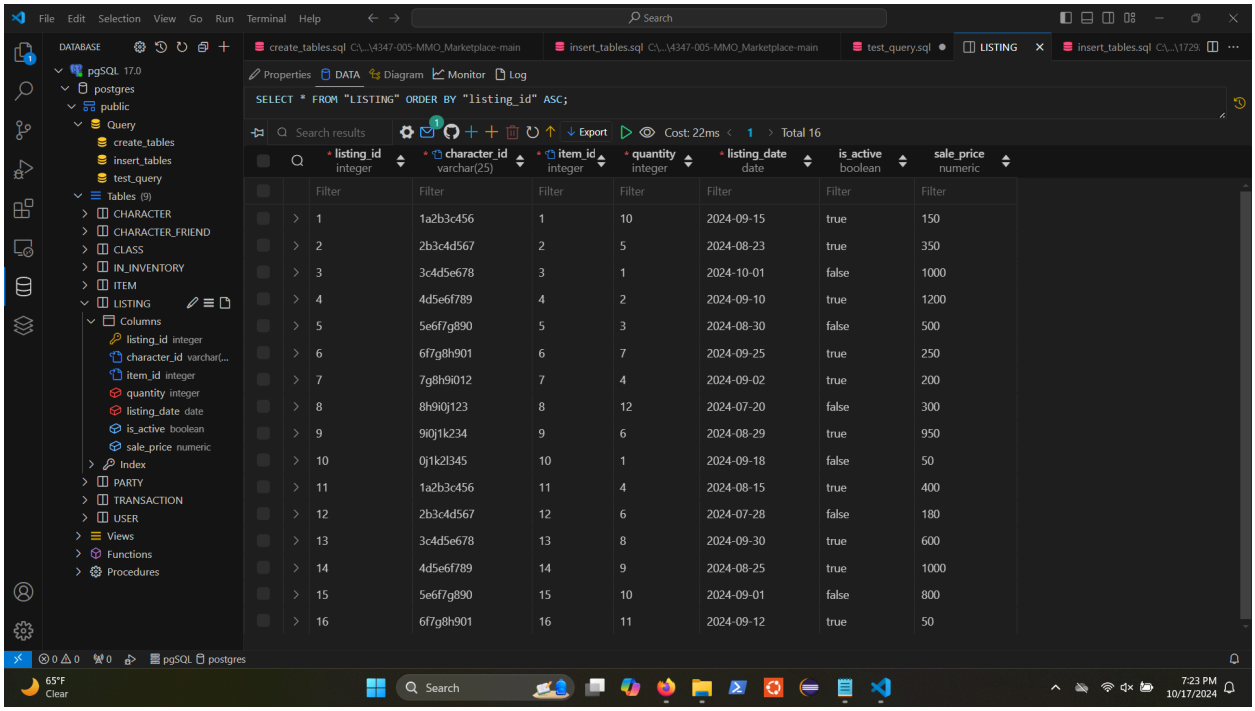


4. **UPDATE**

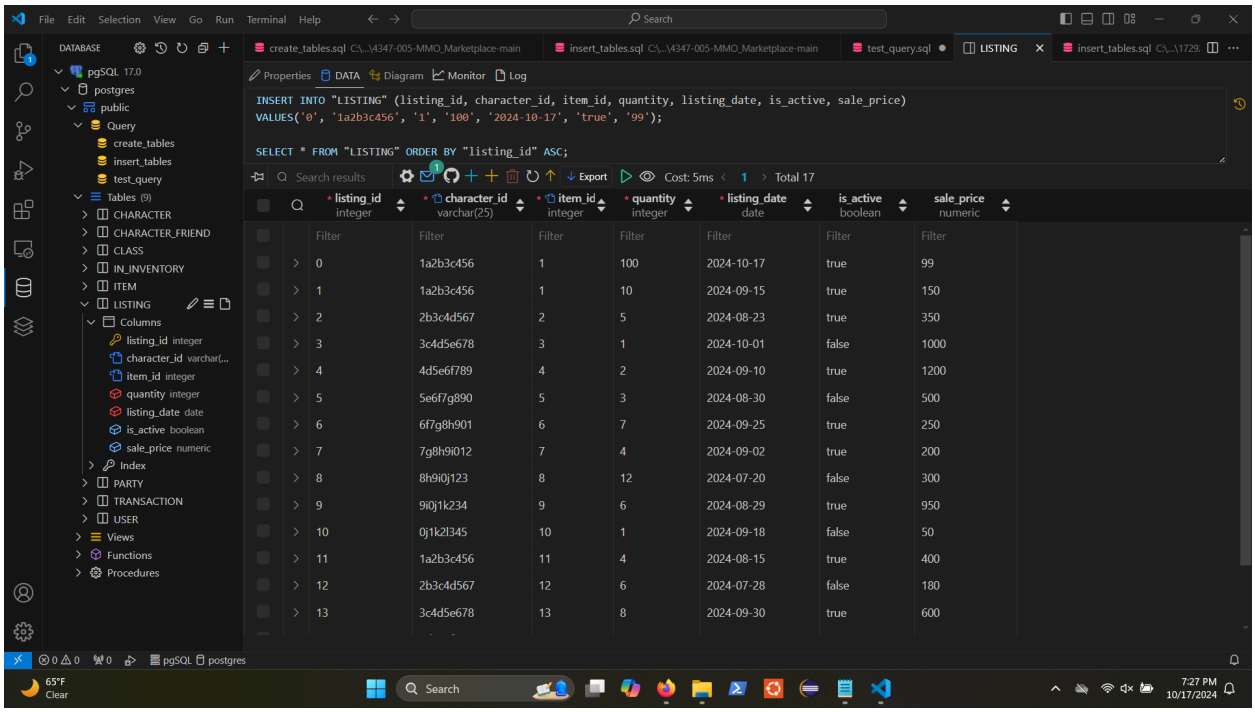


LISTING Table

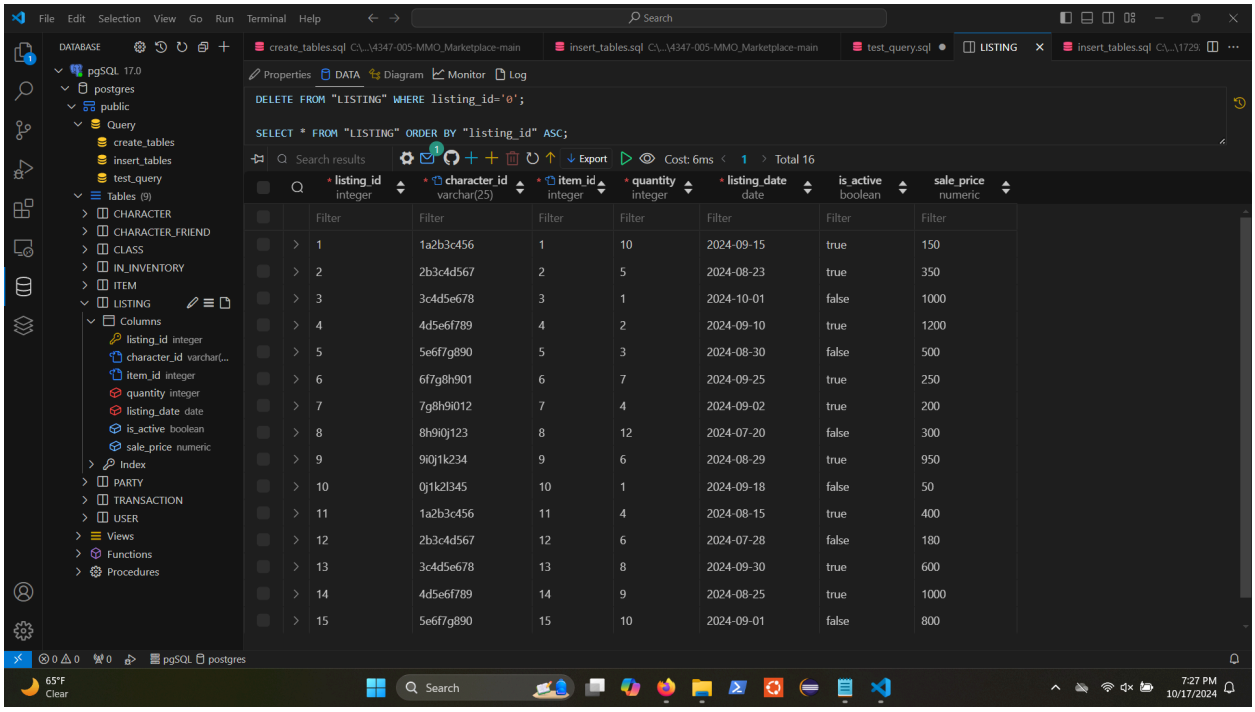
1. **DISPLAY**



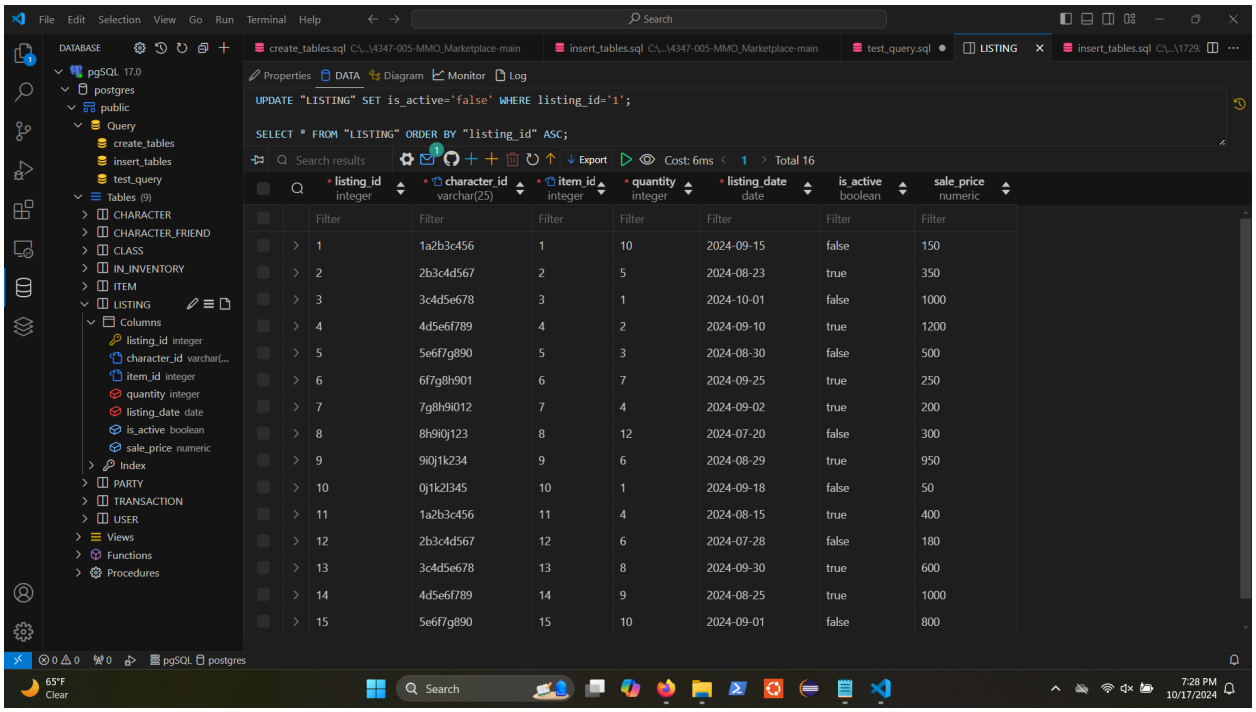
2. INSERT



3. DELETE

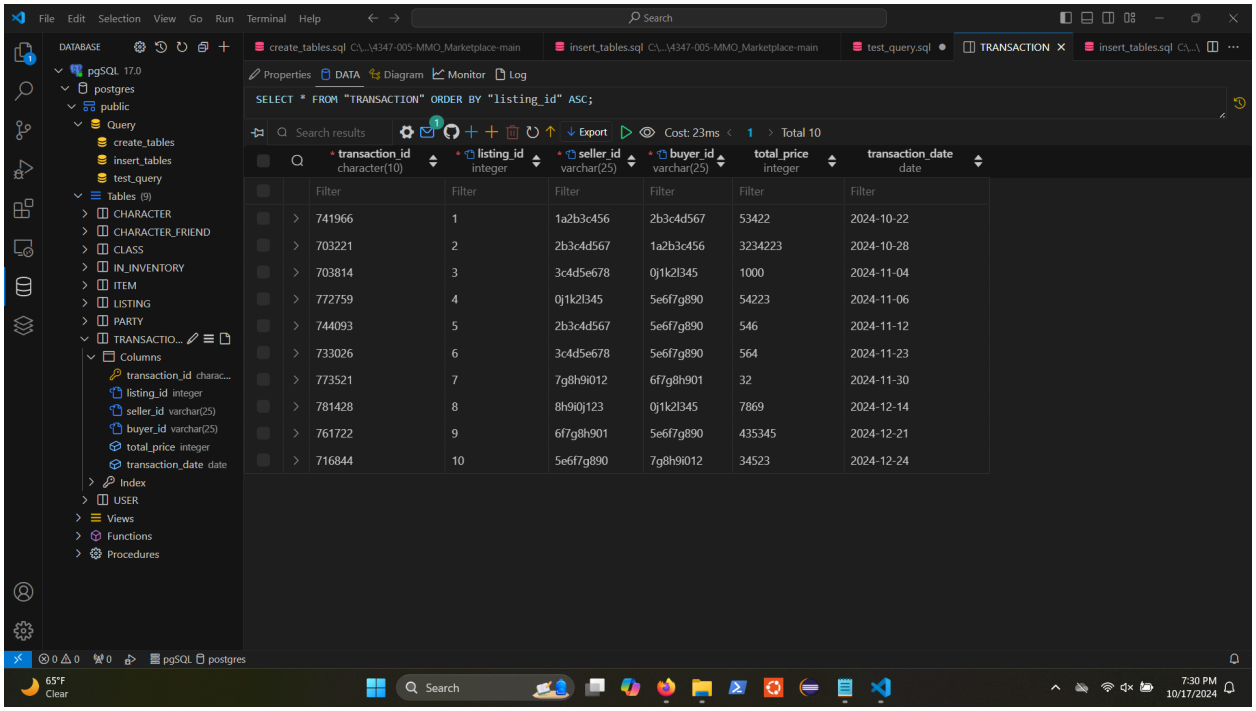


4. UPDATE

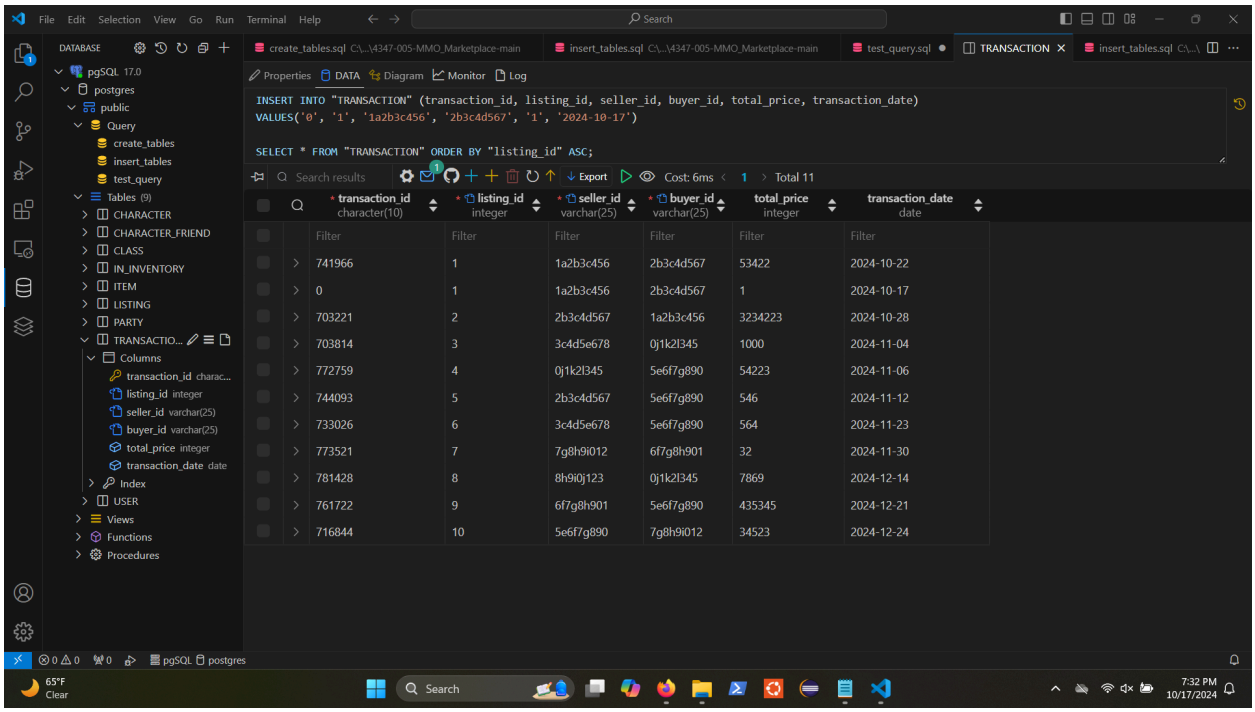


TRANSACTION Table

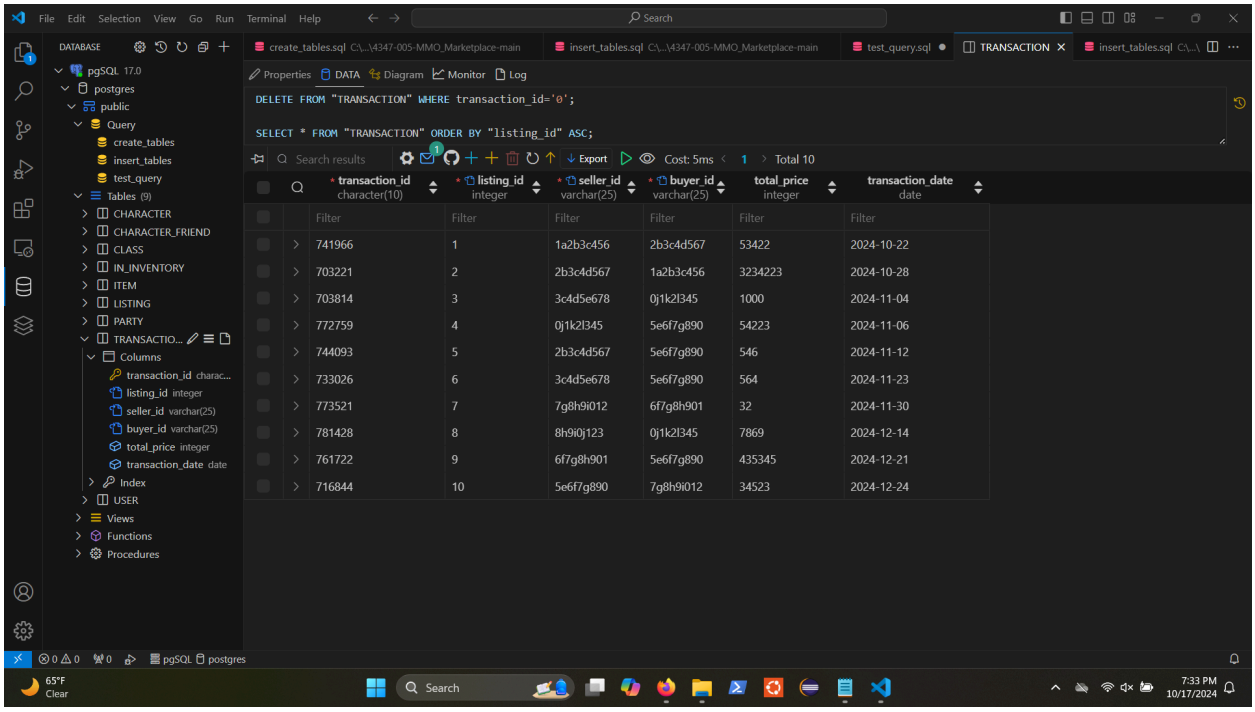
1. DISPLAY



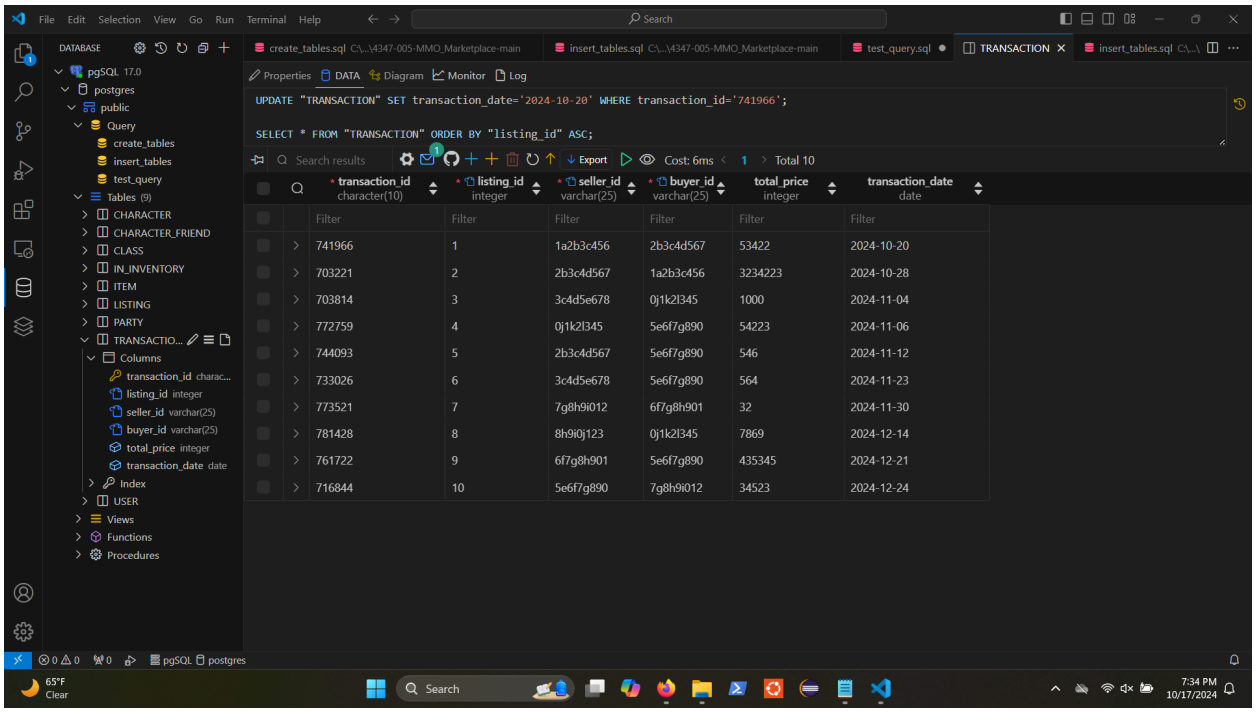
2. INSERT



3. DELETE



4. UPDATE


create_tables.sql
insert_tables.sql
test_query.sql
TRANSACTION
Properties
DATA
Diagram
Monitor
Log
UPDATE "TRANSACTION" SET transaction_date='2024-10-20' WHERE transaction_id='741966';
SELECT * FROM "TRANSACTION" ORDER BY "listing_id" ASC;
Export
Cost: 6ms
Total 10
transaction_id character(10)
listing_id integer
seller_id varchar(25)
buyer_id varchar(25)
total_price integer
transaction_date date
741966 1 1a2b3c456 2b3c4d567 53422 2024-10-20
703221 2 2b3c4d567 1a2b3c456 3234223 2024-10-28
703814 3 3c4d5e678 0j1k2l345 1000 2024-11-04
772759 4 0j1k2l345 5e6f7g890 54223 2024-11-06
744093 5 2b3c4d567 5e6f7g890 546 2024-11-12
733026 6 3c4d5e678 5e6f7g890 564 2024-11-23
773521 7 7g8h9i012 6f7g8h901 32 2024-11-30
781428 8 8h9i0j123 0j1k2l345 7869 2024-12-14
761722 9 6f7g8h901 5e6f7g890 435345 2024-12-21
716844 10 5e6f7g890 7g8h9i012 34523 2024-12-24

References

[1] "FIFA Marketplace," FIFA Collect, https://collect.fifa.com/marketplace (accessed Oct. 8, 2024).

[2] "Official site: Second life - virtual worlds, virtual reality, VR, avatars, free 3D chat," Second Life Marketplace, https://marketplace.secondlife.com/en-US (accessed Oct. 5, 2024).

[3] "Unity Asset Store," Unity Asset Store 2024 version, https://assetstore.unity.com/ (accessed Oct. 4, 2024).

[4] "Steam community :: Steam community market," Steam Community :: Steam Community Market, https://steamcommunity.com/market/ (accessed Oct. 6, 2024).

[5] S. E. Inc., "Eorzea database," FINAL FANTASY XIV, The Lodestone, https://na.finalfantasyxiv.com/lodestone/playguide/db/ (accessed Oct. 6, 2024).

[6] BitSkins, "DOTA2 market: Buy & Sell DOTA2 skins," BitSkins, https://bitskins.com/market/dota2 (accessed Oct. 5, 2024).

[7] Larvalabs, "Larvalabs/cryptopunks: Collectible 8-bit characters on the Ethereum blockchain.," GitHub, https://github.com/larvalabs/cryptopunks (accessed Oct. 6, 2024).

[8] "Marketplace," RuneScape Wiki, https://runescape.wiki/w/Marketplace (accessed Oct. 5, 2024).