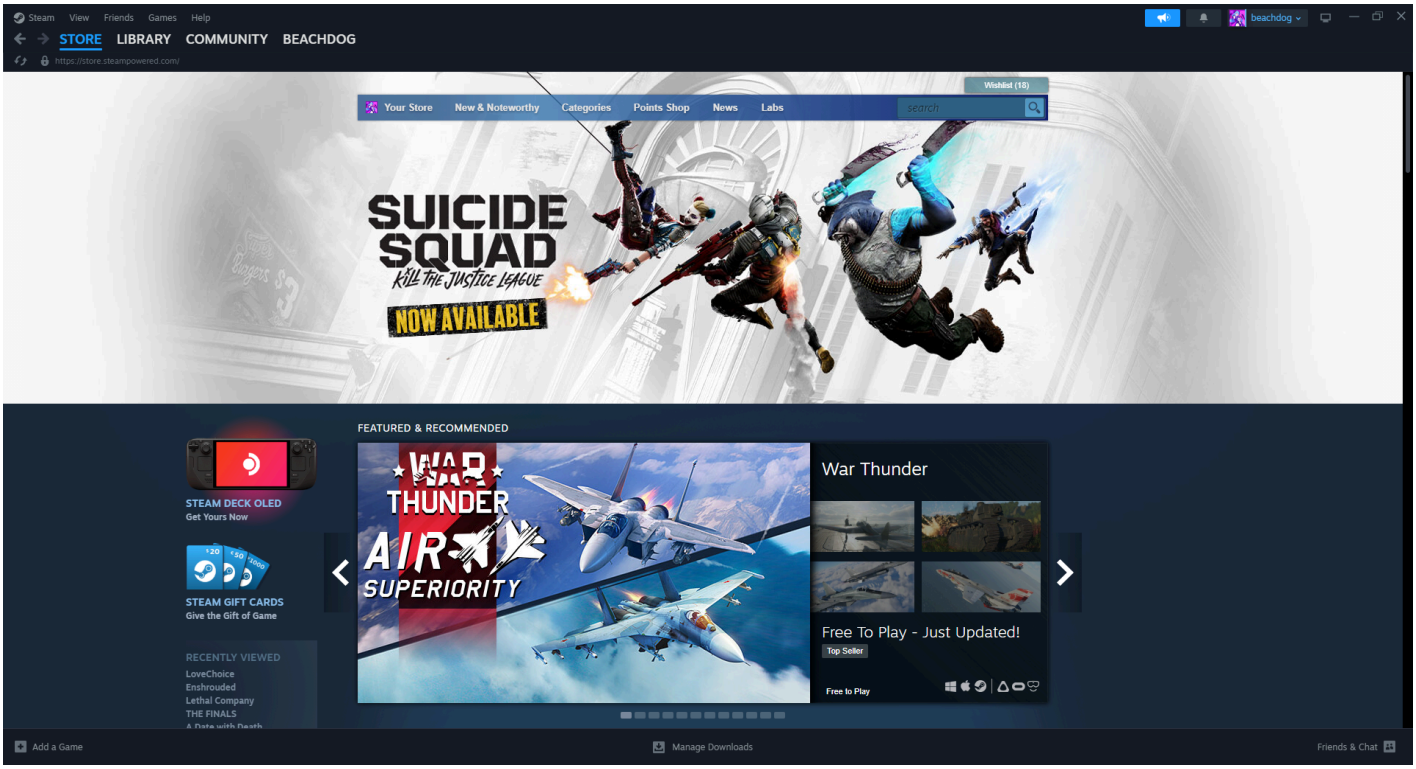
ER Diagram of Valve’s Steam Store

Khristian Jean Jacques

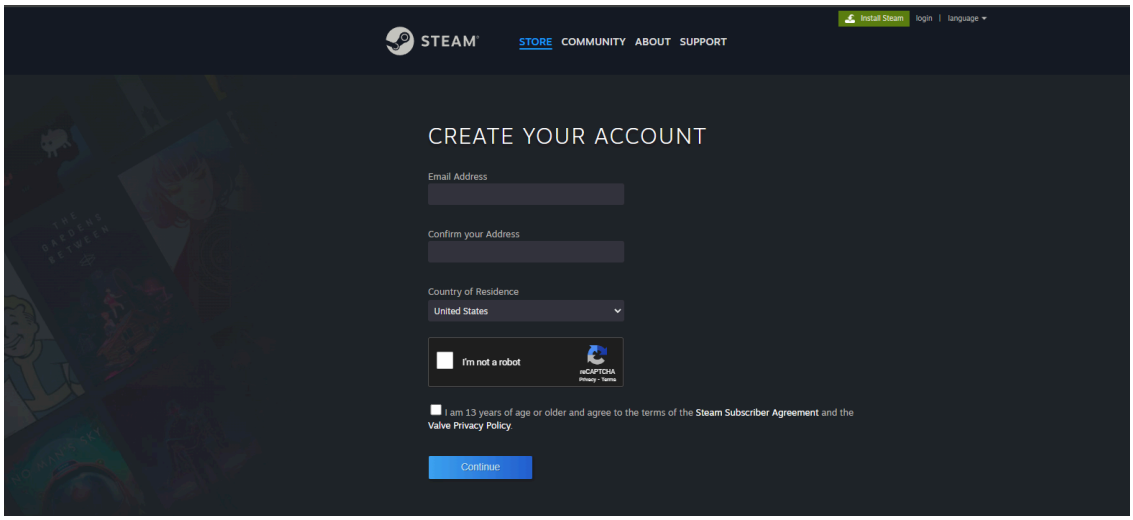
Brianna Pilarte

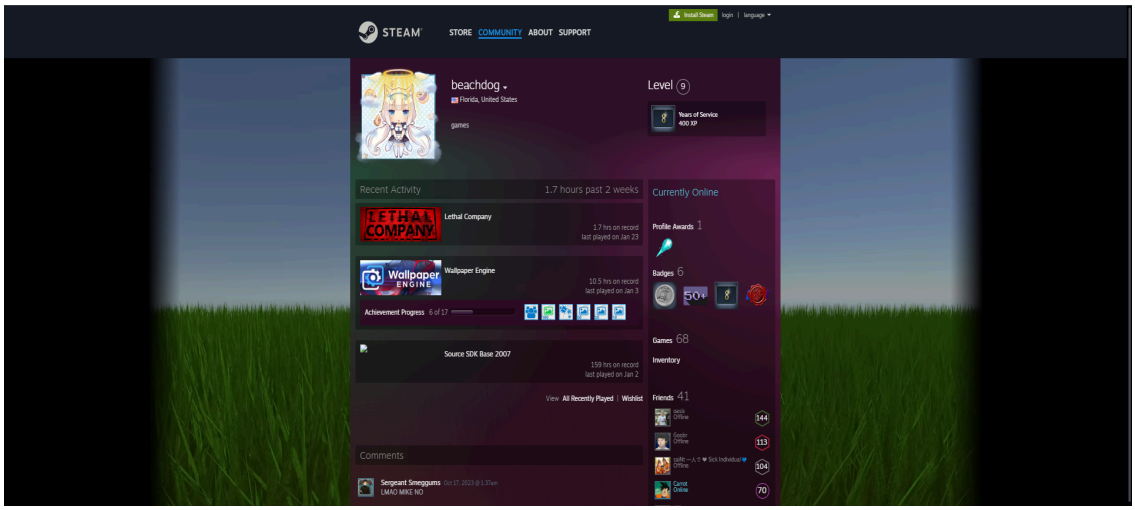
Steam Webpage

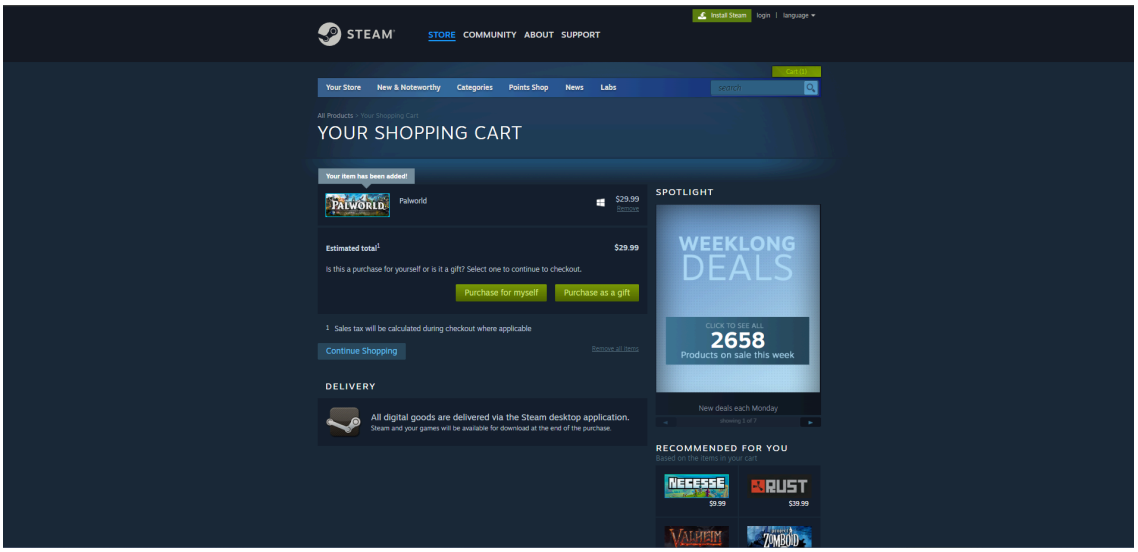


Created in 2003, Steam was launched by the gaming company Valve. Valve successfully created a relationship between consumers and creators/publishers of gaming entertainment before any app store existed. Steam and its community enables millions to register accounts, purchase games, create friendships to message/gift people, join groups and share achievements with each other.

Web Page(s)


STEAM
STORE COMMUNITY ABOUT SUPPORT
Install Steam login | language
CREATE YOUR ACCOUNT
Email Address
Confirm your Address
Country of Residence
United States
I'm not a robot
reCAPTCHA
Privacy - Terms
I am 13 years of age or older and agree to the terms of the Steam Subscriber Agreement and the Valve Privacy Policy.
Continue


STEAM
STORE COMMUNITY ABOUT SUPPORT
beachdog
Level 9
Recent Activity
1.7 hours past 2 weeks
Lethal Company
Wallpaper Engine
Source SDK Base 2007
Currently Online
Profile Awards 1
Badges 6
Games 68
Inventory
Friends 41
Comments


STEAM
STORE COMMUNITY ABOUT SUPPORT
Your Store New & Noteworthy Categories Points Shop News Labs
YOUR SHOPPING CART
Your item has been added!
Palworld
$29.99
Estimated total
$29.99
Is this a purchase for yourself or is it a gift? Select one to continue to checkout.
Purchase for myself
Purchase as a gift
Sales tax will be calculated during checkout where applicable
Continue Shopping
DELIVERY
All digital goods are delivered via the Steam desktop application.
SPOTLIGHT
WEEKLONG DEALS
2658
Products on sale this week
New deals each Monday
RECOMMENDED FOR YOU
$9.99
$39.99

These webpages show creating an account specific to an email address. A steam account requiring username, password, email, and birthdate. A steam account will enable access to the steam store, buying(and gifting) games or programs, a profile, friend interactions(to and from), creating groups, achievements, and an inventory.

Jobs

Frontend Development: Creating an intuitive UI that showcases game titles, descriptions, user reviews, and ratings. The team will use technologies like HTML, CSS, and JavaScript frameworks to ensure responsiveness and interactivity.

Backend Development: Implementing server-side logic to handle user accounts, payment processing, and game library management. This involves setting up databases, secure authentication methods, and server APIs for data retrieval and updates.

Content Management: Working with game developers and publishers to list new titles, update existing game information, and manage promotional content. This includes setting up digital storefronts for individual games, complete with multimedia assets like screenshots and trailers.

User Support and Community Features: Integrating a support system for user inquiries and issues, along with community features such as forums, user profiles, and social sharing options.

Security and Compliance: Ensuring all components comply with digital security standards, data protection laws, and payment processing regulations. This includes encryption of user data, secure transaction processing, and regular security audits.

Rules

Account Ownership: Each user account is owned by a single individual. Co-ownership of accounts is not allowed to ensure security and personalization of content and recommendations.

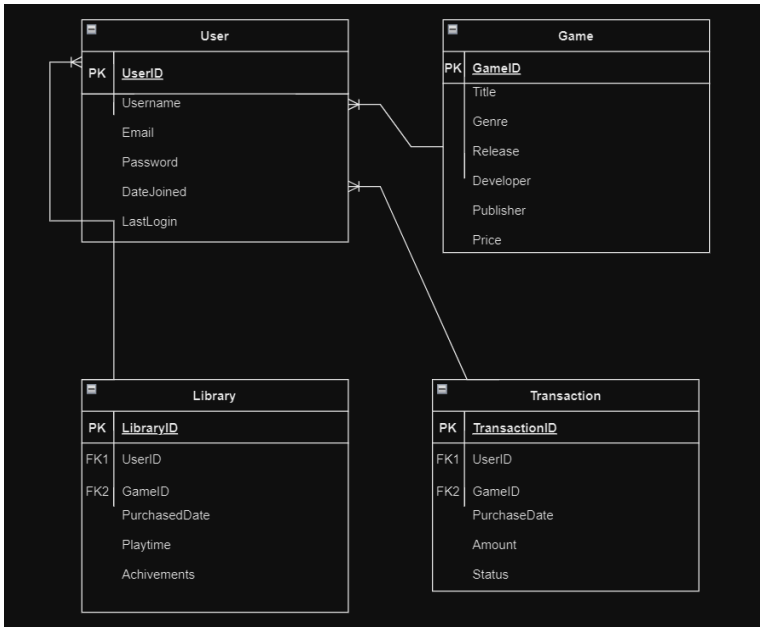
Unique Usernames: Every user must have a unique username to identify their profile, ensuring clear communication and interaction within the community features.

Multiple Payment Methods: Users can link multiple payment methods to their account but must designate one as the primary method for transactions.

Game Ownership: A user can own multiple games, but multiple users cannot own a single copy of a game. This rule is essential for managing digital rights and game licenses.

Review Posting: Users can only post reviews for games they have purchased or played, ensuring the authenticity and relevance of feedback.

These rules help define the relationships and cardinalities in the E-R diagram, such as one-to-many relationships between users and games owned, or the one-to-one relationship between a user and their primary payment method.



Entities:
User:
- Attributes: UserID (Primary Key), Username, Email, Password, DateJoined, LastLogin, etc.
Game:
- Attributes: GameID (Primary Key), Title, Genre, ReleaseDate, Developer, Publisher, Price, etc.
Transaction:
- Attributes: TransactionID (Primary Key), UserID (Foreign Key), GameID (Foreign Key), PurchaseDate, TransactionAmount, Status, etc.
Library:
- Attributes: LibraryID (Primary Key), UserID (Foreign Key), GameID (Foreign Key), PurchaseDate, Playtime, etc.

Relationships:
User purchases Game (1:M):
- One user can make multiple game purchases.
- One game can be purchased by multiple users.
User has Library (1:M):
- One user can have multiple games in their library.
- One game can be in the libraries of multiple users.
Transaction records purchase (M:1):
- Multiple transactions can be associated with one user.

MySQL Script and Schema Diagram

Khristian Jean Jacques

Brianna Pilarte

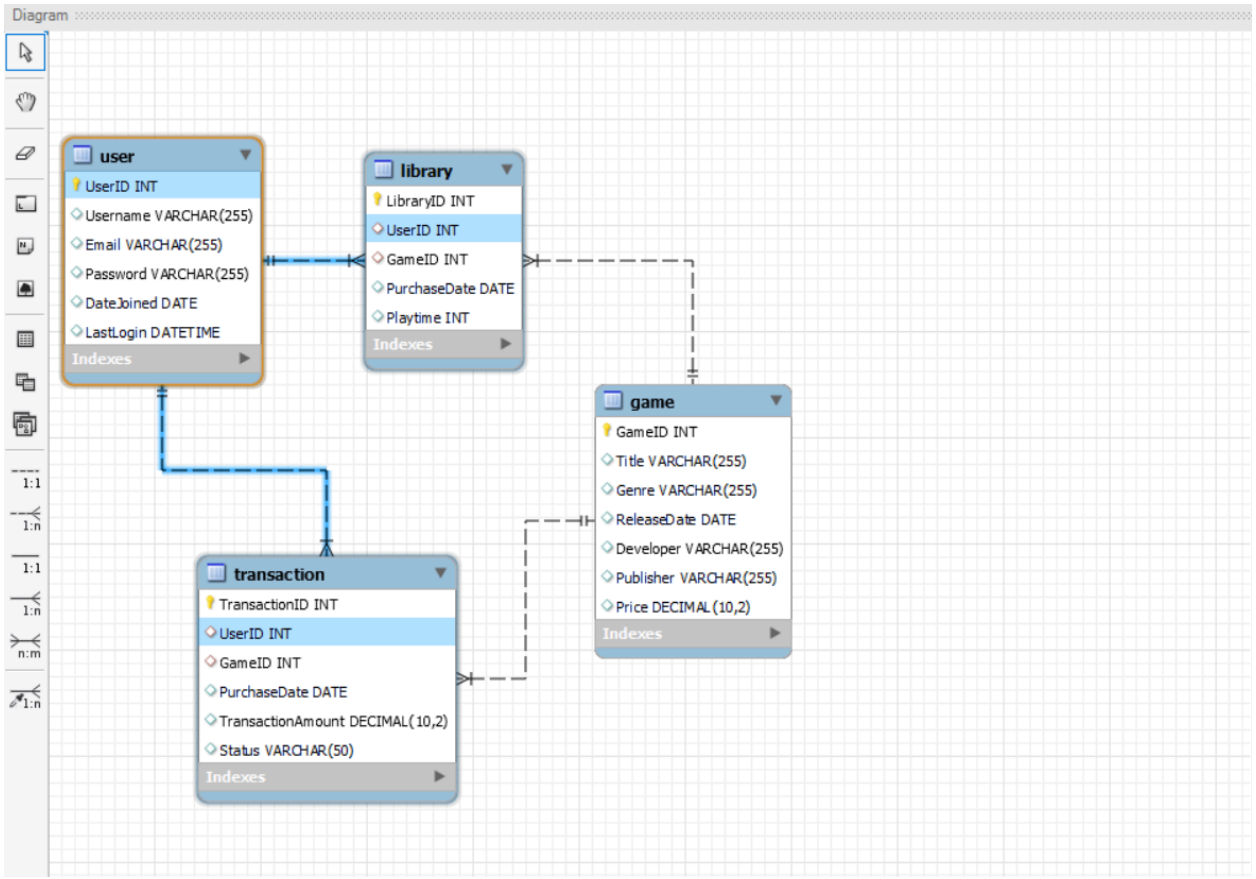
```
CREATE DATABASE steam_datab;

USE steam_datab;

CREATE TABLE User (
    UserID INT PRIMARY KEY,
    Username VARCHAR(255),
    Email VARCHAR(255),
    Password VARCHAR(255),
    DateJoined DATE,
    LastLogin DATETIME
);

CREATE TABLE Game (
    GameID INT PRIMARY KEY,
    Title VARCHAR(255),
    Genre VARCHAR(255),
    ReleaseDate DATE,
    Developer VARCHAR(255),
    Publisher VARCHAR(255),
    Price DECIMAL(10, 2)
);

CREATE TABLE Transaction (
    TransactionID INT PRIMARY KEY,
    UserID INT,
    GameID INT,
    PurchaseDate DATE,
    TransactionAmount DECIMAL(10, 2),
    Status VARCHAR(50),
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (GameID) REFERENCES Game(GameID)
);

CREATE TABLE Library (
    LibraryID INT PRIMARY KEY,
    UserID INT,
    GameID INT,
    PurchaseDate DATE,
    Playtime INT,
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (GameID) REFERENCES Game(GameID)
);
```

Output

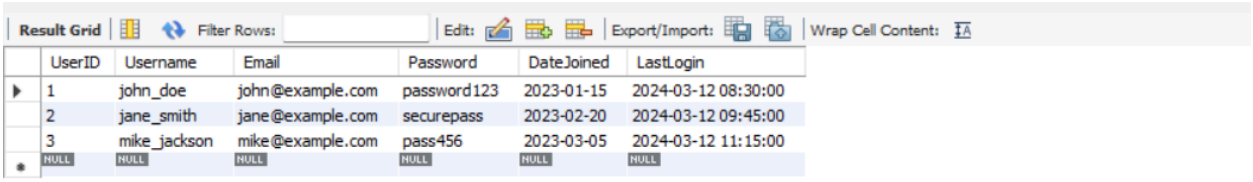
Action Output

#	Time	Action	Message	Duration / Fetch
1	16:44:47	CREATE DATABASE steam_datab	1 row(s) affected	0.047 sec
2	16:44:47	USE steam_datab	0 row(s) affected	0.000 sec
3	16:44:47	CREATE TABLE User (UserID INT PRIMARY KEY, Username VARCHAR(255), Email VARCHAR(255), ...	0 row(s) affected	0.031 sec
4	16:44:47	CREATE TABLE Game (GameID INT PRIMARY KEY, Title VARCHAR(255), Genre VARCHAR(255), R...	0 row(s) affected	0.031 sec
5	16:44:48	CREATE TABLE Transaction (TransactionID INT PRIMARY KEY, UserID INT, GameID INT, Purchase...	0 row(s) affected	0.156 sec
6	16:44:48	CREATE TABLE Library (LibraryID INT PRIMARY KEY, UserID INT, GameID INT, PurchaseDate DAT...	0 row(s) affected	0.094 sec

Schema Diagram

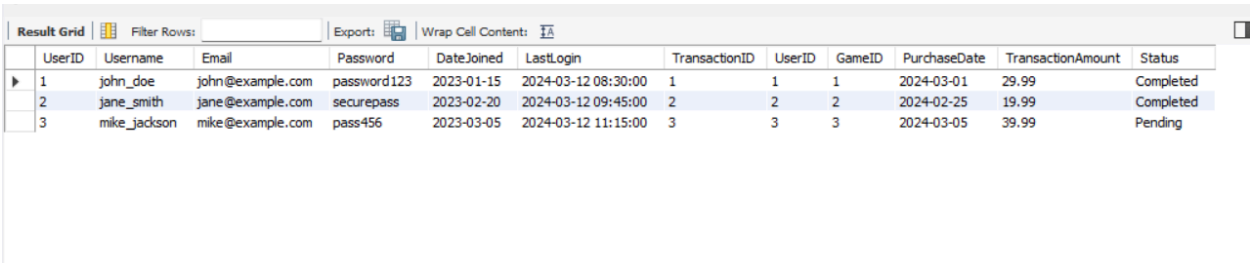


```
INSERT INTO User (UserID, Username, Email, Password, DateJoined, LastLogin)
VALUES
    (1, 'john_doe', 'john@example.com', 'password123', '2023-01-15', '2024-03-12 08:30:00'),
    (2, 'jane_smith', 'jane@example.com', 'securepass', '2023-02-20', '2024-03-12 09:45:00'),
    (3, 'mike_jackson', 'mike@example.com', 'pass456', '2023-03-05', '2024-03-12 11:15:00');

INSERT INTO Game (GameID, Title, Genre, ReleaseDate, Developer, Publisher, Price)
VALUES
    (1, 'The Witcher 3', 'RPG', '2015-05-19', 'CD Projekt', 'CD Projekt', 29.99),
    (2, 'Rocket League', 'Sports', '2015-07-07', 'Psyonix', 'Psyonix', 19.99),
    (3, 'Civilization VI', 'Strategy', '2016-10-21', 'Firaxis Games', '2K', 39.99);

INSERT INTO Transaction (TransactionID, UserID, GameID, PurchaseDate,
TransactionAmount, Status)
VALUES
    (1, 1, 1, '2024-03-01', 29.99, 'Completed'),
    (2, 2, 2, '2024-02-25', 19.99, 'Completed'),
    (3, 3, 3, '2024-03-05', 39.99, 'Pending');

INSERT INTO Library (LibraryID, UserID, GameID, PurchaseDate, Playtime)
VALUES
    (1, 1, 1, '2024-03-01', 50),
    (2, 2, 2, '2024-02-25', 30),
    (3, 3, 3, '2024-03-05', 10);

SELECT * FROM User;

SELECT * FROM User;
```



UserID	Username	Email	Password	DateJoined	LastLogin
1	john_doe	john@example.com	password123	2023-01-15	2024-03-12 08:30:00
2	jane_smith	jane@example.com	securepass	2023-02-20	2024-03-12 09:45:00
3	mike_jackson	mike@example.com	pass456	2023-03-05	2024-03-12 11:15:00
NULL	NULL	NULL	NULL	NULL	NULL

```
-- I. Using WHERE
SELECT * FROM User
WHERE DateJoined < '2023-03-01';

-- II. Using more than one table in FROM
SELECT User.*, Transaction.*
FROM User
JOIN Transaction ON User.UserID = Transaction.UserID;

-- III. Using SET operation
SELECT UserID, GameID FROM Library WHERE UserID = 1
UNION
SELECT UserID, GameID FROM Library WHERE UserID = 2;

-- IV. Using aggregate function and/or GROUP BY
SELECT UserID, SUM(Playtime) AS TotalPlaytime
FROM Library
GROUP BY UserID;

-- V. Using SUBQUERY
SELECT * FROM User
WHERE UserID IN (SELECT UserID FROM Transaction WHERE GameID = 1);

-- VI. Using EXISTS or UNIQUE
SELECT * FROM User
WHERE EXISTS (SELECT 1 FROM Transaction WHERE User.UserID = Transaction.UserID);

-- VII. Using WITH
WITH RecentTransactionCTE AS (
    SELECT UserID, MAX(PurchaseDate) AS LatestPurchaseDate
    FROM Transaction
    GROUP BY UserID
)

SELECT User.*, Transaction.*
FROM User
JOIN Transaction ON User.UserID = Transaction.UserID
JOIN RecentTransactionCTE ON User.UserID = RecentTransactionCTE.UserID AND
Transaction.PurchaseDate = RecentTransactionCTE.LatestPurchaseDate;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

UserID	Username	Email	Password	DateJoined	LastLogin	TransactionID	UserID	GameID	PurchaseDate	TransactionAmount	Status
1	john_doe	john@example.com	password123	2023-01-15	2024-03-12 08:30:00	1	1	1	2024-03-01	29.99	Completed
2	jane_smith	jane@example.com	securepass	2023-02-20	2024-03-12 09:45:00	2	2	2	2024-02-25	19.99	Completed
3	mike_jackson	mike@example.com	pass456	2023-03-05	2024-03-12 11:15:00	3	3	3	2024-03-05	39.99	Pending

```
UPDATE User
SET LastLogin = CURRENT_TIMESTAMP()
WHERE UserID = 1;

UPDATE Transaction
SET Status = 'Completed'
WHERE Status = 'Pending' AND PurchaseDate < '2024-03-05';
```