University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 21/07/2024

Group Number: 38

Name	Student Number	Alias (Userid)	Preferred E-mail Address
Ali Hasan	34975292	ahasan02	ali.hasan9712@gmail.com
Sharjeel Shahid	30717987	sshahi03	sshahi28@uwo.ca
Muhammad Zaid Tahir	26857201	mtahir03	zaidt221325@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Question 2

Our database intends to model a player-centric client for an online, match-based, multiplayer video game. The entirety of player persistent information, centered around individual player profiles and match history, will be modeled in the database. The database will also model a centralized game shop supporting multiple currencies with which a player can transact to populate their inventories.

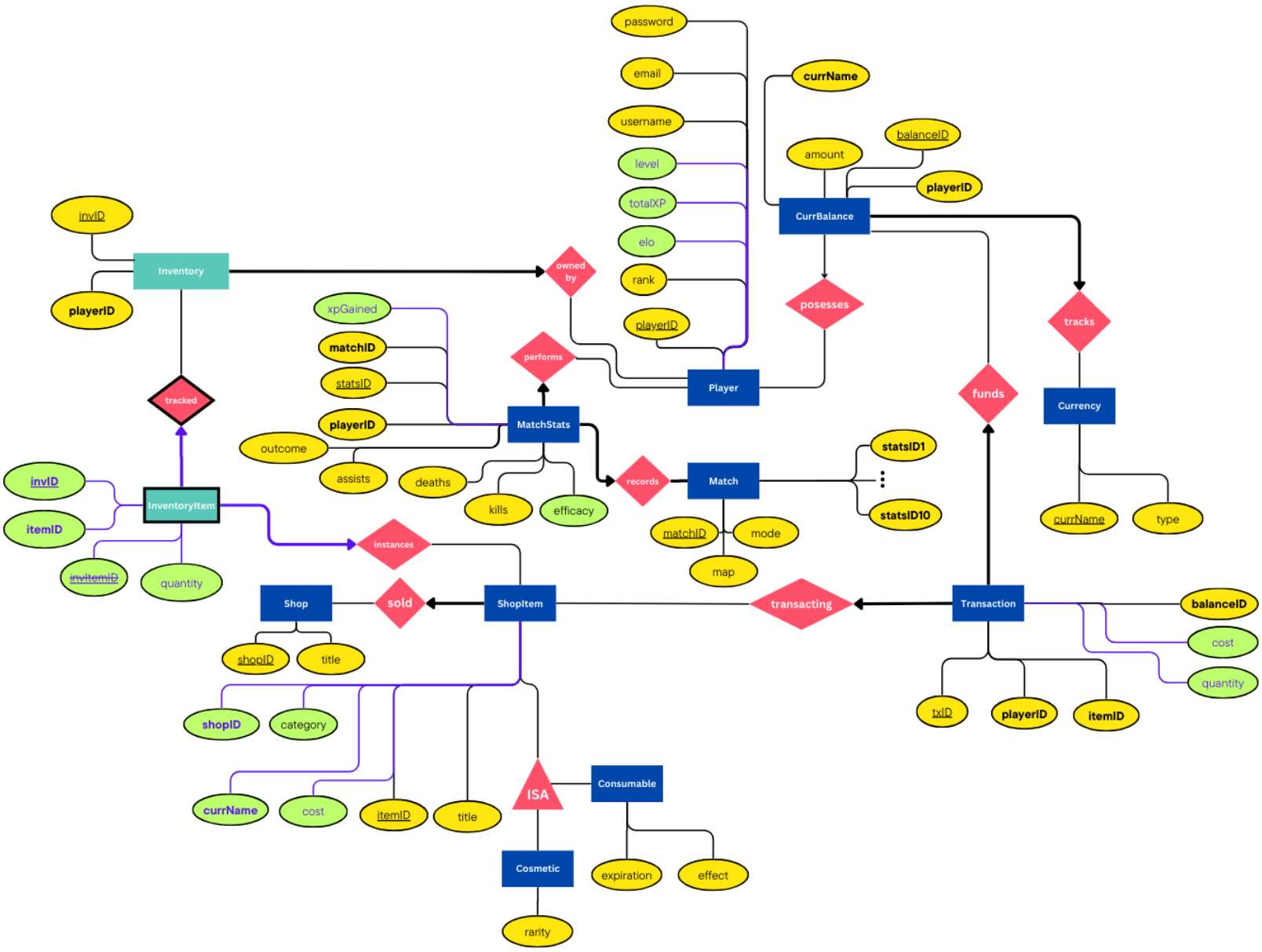
Question 3

The following changes were made to our ER diagram:

1. Cardinality arrows now point to relations rather than entities
2. The “InventoryItem” weak entity, previously connected to both ShopItem and Player has now been refactored into a separate “Inventory” (strong) entity which represents the many inventories a player can have. The InventoryItem is now a weak entity owned by Inventory, representing an instance of a ShopItem (linked by a foreign key). These changes are highlighted with cyan entities and purple connecting lines/arrows.
3. Partial keys are now represented by a strikethrough and underline, as opposed to just a strikethrough
4. Various attributes were added to existing entities to fulfill Question 5, as otherwise all of our functional dependencies already existed in BCNF (with the exception of MatchStats, which was in 3NF but could have been decomposed into BCNF). These attributes are colored in light green with purple text.
5. The “Currency Finances Transaction” relation was removed as it fulfills no purpose given the already existing “CurrBalance Funds Transaction” relation. Similarly, Player Plays Match was removed due to the redundancy from Player Performs MatchStats

These changes can be viewed in the next page

University of British Columbia, Vancouver
Department of Computer Science



Revised ER Diagram of the proposed application

University of British Columbia, Vancouver
Department of Computer Science

Question 4

Based strictly on our ER diagram, we have devised the following schemas:

Player

Formal Schema: Player(playerID: integer, username: char[30], email: char[50], password: char[24], elo: integer, rank: char[20], level: integer, totalXP: integer)
Primary Key: playerID
Candidate Key(s): username; email;
Foreign Key(s): N/A
Constraint: username NOT NULL UNIQUE, email NOT NULL UNIQUE, password NOT NULL

Match

Formal Schema: Match(matchID: integer, mode: char[50], map: char[50], **statsID1**: integer, **statsID2**: integer, **statsID3**: integer, **statsID4**: integer, **statsID5**: integer, **statsID6**: integer, **statsID7**: integer, **statsID8**: integer, **statsID9**: integer, **statsID10**: integer)
Primary Key: matchID
Candidate Key(s): statsIDx
Foreign Key(s): statsIDx (references MatchStats)
Constraint: statsIDx NOT NULL UNIQUE, mode NOT NULL, map NOT NULL

MatchStats (Records + Performance)

Formal Schema: MatchStats(statsID: integer, **matchID**: integer, **playerID**: integer, statsID: integer, kills: integer, deaths: integer, assists: integer, outcome: char[8], xpGained: integer, efficacy: float)
Primary Key: statsID
Candidate Key(s): matchID, playerID
Foreign Key(s): matchID (references Match), playerID (references Player)
Constraint: matchID NOT NULL, playerID NOT NULL, kills NOT NULL, deaths NOT NULL, assists NOT NULL, outcome NOT NULL, xpGained NOT NULL, efficacy NOT NULL

InventoryOwnedBy

Formal Schema: Inventory(invID: integer, **playerID**: integer)
Primary Key: invID
Candidate Key(s): N/A
Foreign Key(s): playerID (referencing Player)
Constraint: playerID NOT NULL

University of British Columbia, Vancouver
Department of Computer Science

InventoryItem (Tracked + Instances)

Formal Schema: InventoryItem(invItemID: integer, **invID**: integer, **itemID**: integer, quantity: integer)
Primary Key: (invID, invItemID)
Candidate Key(s): N/A
Foreign Key(s): invID (referencing Inventory), itemID (referencing ShopItem)
Constraint: itemID NOT NULL, invID NOT NULL

Shop

Formal Schema: Shop(shopID: integer, title: char[50])
Primary Key: shopID
Candidate Key(s): title
Foreign Key(s): N/A
Constraint: title NOT NULL UNIQUE

ShopItemSold

Formal Schema: ShopItem(itemID: integer, **shopID**: integer, title: char[100], category: char[100], cost: integer, currName: char[50])
Primary Key: itemID
Candidate Key(s): N/A
Foreign Key(s): shopID (referencing Shop), currName (referencing Currency)
Constraint: title NOT NULL, cost NOT NULL, currName NOT NULL, shopID NOT NULL

Consumable

Formal Schema: Consumable(**itemID**: integer, expiration: integer, effect: char[255])
Primary Key: itemID (ISA ShopItem)
Candidate Key(s): N/A
Foreign Key(s): itemID
Constraint: effect NOT NULL

Cosmetic

Formal Schema: Cosmetic(**itemID**: integer, rarity: char[24])
Primary Key: itemID (ISA ShopItem)
Candidate Key(s): N/A
Foreign Key(s): itemID
Constraint: rarity NOT NULL

Transaction (Funds + Transacting)

Formal Schema: Transaction(txID: integer, **playerID**: integer, **itemID**: integer, **balanceID**: integer, cost: integer, quantity: integer)
Primary Key: txID
Candidate Key(s): N/A
Foreign Key(s): playerID (references Player), balanceID (references CurrBalance), itemID (references ShopItem)
Constraint: playerID NOT NULL, balanceID NOT NULL, itemID NOT NULL, cost NOT NULL, quantity NOT NULL

Currency

Formal Schema: Currency(currName: char[50], type: boolean)
Primary Key: currName
Candidate Key(s): N/A
Foreign Key(s): itemID
Constraint: type NOT NULL, itemID NOT NULL

CurrBalance (Tracks + Possesses)

Formal Schema: CurrBalance(balanceID: integer, **playerID**: integer, **currName**: char[50], amount: integer)
Primary Key: balanceID
Candidate Key(s): N/A
Foreign Key(s): playerID (references Player), currName (references Currency)
Constraint:playerID NOT NULL, currName NOT NULL

Question 5

Based on our entity and relation design, we have identified the following functional dependencies:

Player

1) playerID → username, email, password, rank, level, totalXP, elo
2) username → playerID, email, password, rank, level, totalXP, elo
3) email → playerID, username, password, rank, level, totalXP, elo
4) totalXP → level
5) elo → rank

University of British Columbia, Vancouver
Department of Computer Science

InventoryItem (Tracked + Instances)

1) invID, invItemID → itemID, playerID, quantity

Shop

1) shopID → title
2) title → shopID

ShopItemSold

1) itemID → shopID, title, cost, currName, category
2) title → category

Consumable

1) itemID → expiration, effect

Cosmetic

1) itemID → rarity

Transaction (Funds + Transacting)

1) txID → playerID, itemID, balanceID, cost, quantity
2) itemID, cost → quantity
3) itemID, quantity → cost

Currency

1) currName → type

CurrBalance (Tracks + Possesses)

1) balanceID → playerID, currName, amount

Match

1) matchID → mode, map, statsID1..statsID10
2) statsID1 → matchID, map, statsID2..10
3) statsID2 → matchID, map, statsID1, statsID3..10
4) …
11) statsID10 → matchID, map, statsID1..9

MatchStats (Records + Performance)

1) statsID → matchID, playerID, kills, deaths, assists, efficacy, outcome, xpGained
2) efficacy, outcome → xpGained
3) kills, deaths, assists → efficacy

University of British Columbia, Vancouver

Department of Computer Science

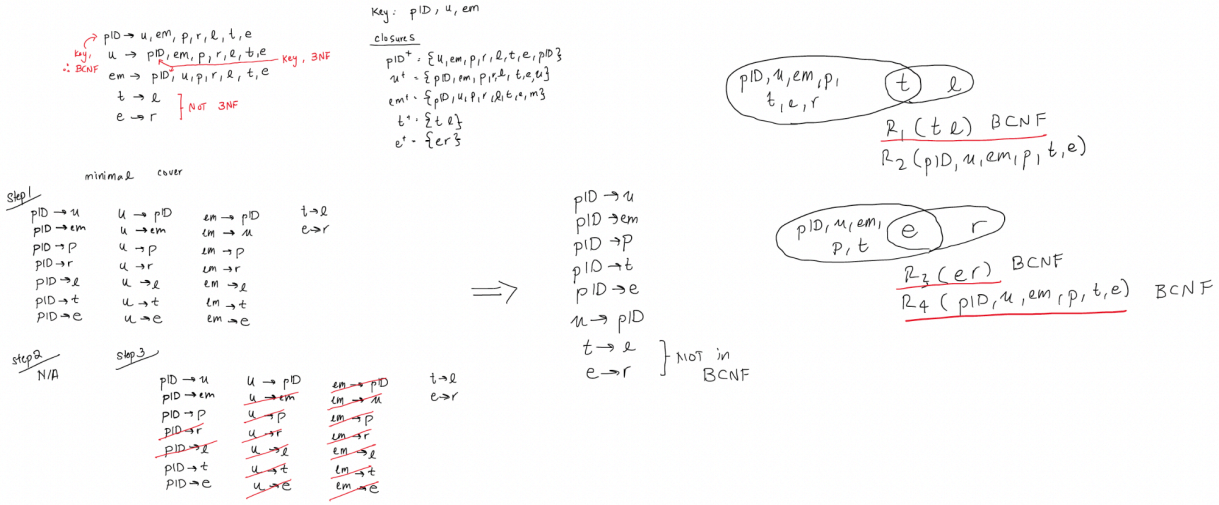
Question 6

Based on our initial functional dependencies, we have identified the following:

Match is in 3NF
CurrBalance, Currency, Cosmetic, Consumable, Shop, Inventory, and **InventoryItem** are all in BCNF
Player, MatchStats, Transaction, and **ShopItem** violate 3NF and thus violate BCNF

Player

Violates BCNF/3NF due to t -> l and e -> r. Let (pID, u, em, p, r, e, l, t) map to {playerID, username, email, password, rank, elo, level, totalXP}, respectively.



Once we reach R4, the remaining dependencies are in BCNF. Accordingly, our Player schema:

Player

Formal Schema: Player(playerID: integer, username: char[30], email: char[50], password: char[24], elo: integer, totalXP: integer)
Primary Key: playerID
Candidate Key(s): username; email;
Foreign Key(s): N/A
Constraint: username NOT NULL UNIQUE, email NOT NULL UNIQUE, password NOT NULL

PlayerRank

Formal Schema: PlayerRank(elo: integer, rank: char[20])
Primary Key: elo
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: rank NOT NULL

University of British Columbia, Vancouver

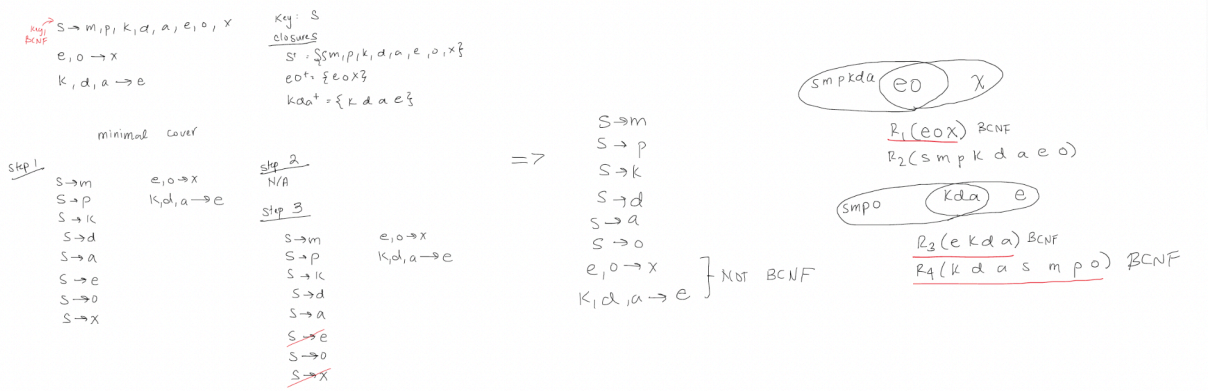
Department of Computer Science

PlayerLevel

Formal Schema: PlayerLevel(xp: integer, level: integer)
Primary Key: xp
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: level NOT NULL

MatchStats (Records + Performance)

Violates BCNF/3NF due to e,o -> x and k,d,a -> e.
Let (s, m, p, k, d, a, e, o, x) map to (statsID, matchID, playerID, kills, deaths, assists, efficacy, outcome, xpGained



At R4 the remaining dependencies are in BCNF, accordingly, the MatchStats schema:

MatchStats (Records + Performance)

Formal Schema: Match(statsID: integer, matchID: integer, playerID: integer, statsID: integer, kills: integer, deaths: integer, assists: integer, outcome: char[8])
Primary Key: statsID
Candidate Key(s): N/A
Foreign Key(s): matchID (references Match), playerID (references Player)
Constraint: matchID NOT NULL, playerID NOT NULL, kills NOT NULL, deaths NOT NULL, assists NOT NULL, outcome NOT NULL

MatchReward

Formal Schema: MatchReward(efficacy: float, outcome: char[8], xpGained: integer)
Primary Key: (efficacy, outcome)
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: xpGained NOT NULL

University of British Columbia, Vancouver

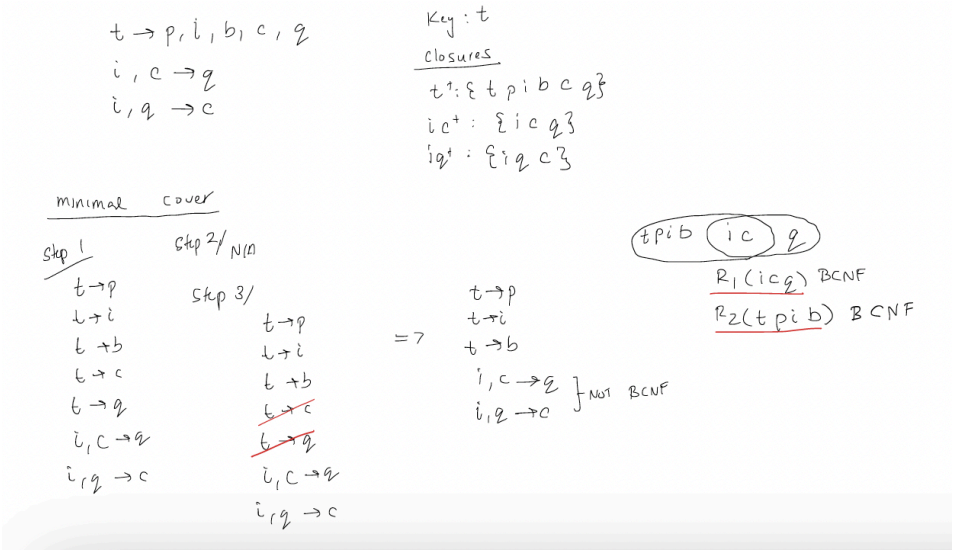
Department of Computer Science

Performance

Formal Schema: Match(kills: integer, deaths: integer, assists: integer, efficacy: float)
Primary Key: (kills, deaths, assists)
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: efficacy NOT NULL

Transaction (Funds + Transacting)

Violates BCNF/3NF due to i,c -> q and i,q -> c. Let (t, p, i, b, c, q) map to (txID, playerID, itemID, balanceID, cost, quantity), respectively.



At R2, the remaining dependencies are in BCNF and we decompose Transaction into:

Transaction (Funds + Transacting)

Formal Schema: Transaction(txID: integer, playerID: integer, itemID: integer, balanceID: integer, cost: integer)
Primary Key: txID
Candidate Key(s): N/A
Foreign Key(s): playerID (references Player), balanceID (references CurrBalance), itemID (references ShopItem)
Constraint: playerID NOT NULL, balanceID NOT NULL, itemID NOT NULL, cost NOT NULL

University of British Columbia, Vancouver

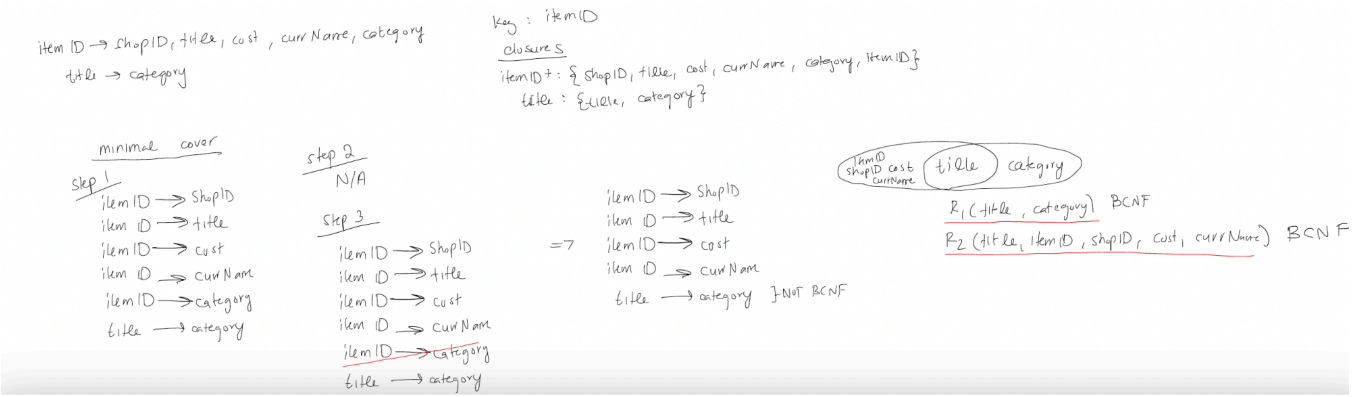
Department of Computer Science

SaleAmount

Formal Schema: SaleAmount(itemID: integer, cost: integer, quantity: integer)
Primary Key: (itemID, cost)
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: quantity NOT NULL

ShopItemSold

Violates BCNF/3NF due to title -> category.



At R2, the remaining dependencies are in BCNF, we decompose ShopItem into:

ShopItemSold

Formal Schema: ShopItem(itemID: integer, shopID: integer, title: char[100], cost: integer, currName: char[50])
Primary Key: itemID
Candidate Key(s): N/A
Foreign Key(s): shopID (referencing Shop), currName (referencing Currency)
Constraint: title NOT NULL, cost NOT NULL, currName NOT NULL

ItemCategory

Formal Schema: ItemCategory(title: char[100], category: char[100])
Primary Key: title
Candidate Key(s): N/A
Foreign Key(s): N/A
Constraint: N/A

Question 7

```
CREATE TABLE Match (
   matchID INTEGER PRIMARY KEY,
   mode VARCHAR(50) NOT NULL,
   map VARCHAR(50) NOT NULL,
   statsID1 INTEGER NOT NULL UNIQUE,
   statsID2 INTEGER NOT NULL UNIQUE,
   statsID3 INTEGER NOT NULL UNIQUE,
   statsID4 INTEGER NOT NULL UNIQUE,
   statsID5 INTEGER NOT NULL UNIQUE,
   statsID6 INTEGER NOT NULL UNIQUE,
   statsID7 INTEGER NOT NULL UNIQUE,
   statsID8 INTEGER NOT NULL UNIQUE,
   statsID9 INTEGER NOT NULL UNIQUE,
   statsID10 INTEGER NOT NULL UNIQUE,
   FOREIGN KEY (statsID1) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID2) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID3) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID4) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID5) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID6) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID7) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID8) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID9) REFERENCES MatchStats(statsID) ON DELETE CASCADE,
   FOREIGN KEY (statsID10) REFERENCES MatchStats(statsID) ON DELETE CASCADE
);


CREATE TABLE CurrBalance (
   balanceID INTEGER PRIMARY KEY,
   playerID INTEGER NOT NULL,
   currName VARCHAR(50) NOT NULL,
   amount INTEGER,
   FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCADE,
   FOREIGN KEY (currName) REFERENCES Currency(currName) ON DELETE CASCADE
);
```

```
CREATE TABLE Currency (
   currName VARCHAR(50),
   type BOOLEAN NOT NULL,
   PRIMARY KEY (currName, itemID)
);

CREATE TABLE ShopItemSold (
   itemID INTEGER PRIMARY KEY,
   shopID INTEGER NOT NULL,
   title VARCHAR(100) NOT NULL,
   cost INTEGER NOT NULL,
   currName VARCHAR(50) NOT NULL,
   FOREIGN KEY (shopID) REFERENCES Shop(shopID) ON DELETE CASCADE,
   FOREIGN KEY (currName) REFERENCES Currency(currName) ON DELETE CASCADE
);

CREATE TABLE Cosmetic (
   itemID INTEGER PRIMARY KEY,
   rarity VARCHAR(24) NOT NULL,
   FOREIGN KEY (itemID) REFERENCES ShopItem(itemID) ON DELETE CASCADE
);

CREATE TABLE Consumable (
   itemID INTEGER PRIMARY KEY,
   expiration INTEGER,
   effect VARCHAR(255) NOT NULL,
   FOREIGN KEY (itemID) REFERENCES ShopItem(itemID) ON DELETE CASCADE
);

CREATE TABLE Shop (
   shopID INTEGER PRIMARY KEY,
   title VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE Inventory (
   invID INTEGER PRIMARY KEY,
   playerID INTEGER NOT NULL,
   FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCADE
);
```

```
CREATE TABLE InventoryItem (
   invItemID INTEGER,
   invID INTEGER NOT NULL,
   itemID INTEGER NOT NULL,
   quantity INTEGER,
   PRIMARY KEY (invID, invItemID),
   FOREIGN KEY (invID) REFERENCES Inventory(invID) ON DELETE CASCADE,
   FOREIGN KEY (itemID) REFERENCES ShopItem(itemID) ON DELETE CASCADE
);

CREATE TABLE Player (
   playerID INTEGER PRIMARY KEY,
   username VARCHAR(30) NOT NULL UNIQUE,
   email VARCHAR(50) NOT NULL UNIQUE,
   password VARCHAR(24) NOT NULL,
   elo INTEGER,
   totalXP INTEGER
);

CREATE TABLE PlayerRank (
   elo INTEGER PRIMARY KEY,
   rank VARCHAR(20) NOT NULL
);

CREATE TABLE PlayerLevel (
   xp INTEGER PRIMARY KEY,
   level INTEGER NOT NULL
);

CREATE TABLE MatchStats (
   statsID INTEGER PRIMARY KEY,
   matchID INTEGER NOT NULL,
   playerID INTEGER NOT NULL,
   kills INTEGER NOT NULL,
   deaths INTEGER NOT NULL,
   assists INTEGER NOT NULL,
   outcome VARCHAR(8) NOT NULL,
   FOREIGN KEY (matchID) REFERENCES Match(matchID) ON DELETE CASCADE,
   FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCADE
);
```

```
CREATE TABLE MatchReward (
   efficacy FLOAT,
   outcome VARCHAR(8),
   xpGained INTEGER NOT NULL,
   PRIMARY KEY (efficacy, outcome)
);

CREATE TABLE Performance (
   kills INTEGER,
   deaths INTEGER,
   assists INTEGER,
   efficacy FLOAT NOT NULL,
   PRIMARY KEY (kills, deaths, assists)
);

CREATE TABLE Transaction (
   txID INTEGER PRIMARY KEY,
   playerID INTEGER NOT NULL,
   itemID INTEGER NOT NULL,
   balanceID INTEGER NOT NULL,
   cost INTEGER NOT NULL,
   FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCADE,
   FOREIGN KEY (balanceID) REFERENCES CurrBalance(balanceID) ON DELETE CASCADE,
   FOREIGN KEY (itemID) REFERENCES ShopItem(itemID) ON DELETE CASCADE
);

CREATE TABLE SaleAmount (
   itemID INTEGER,
   cost INTEGER,
   quantity INTEGER NOT NULL,
   PRIMARY KEY (itemID, cost)
);

CREATE TABLE ItemCategory (
   title VARCHAR(100) PRIMARY KEY,
   category VARCHAR(100)
);
```

QUESTION 8

INSERT INTO Player (playerID, username, email, password, elo, totalXP) VALUES
(1, 'Ali', 'ali@foo.com', 'password1234567890', 1000, 0),
(2, 'Sharjeel', 'sharjeel@foo.com', 'password1234567890', null, 0),
(3, 'Zaid', 'zaid@foo.com', ‘qwerty’, 700, 7000),
(4, 'Steve', 'steve@foo.com', ‘askldfjdsl’, null, null),
(5, 'Wozniak', 'wozniak@foo.com', 'password1234567890', 1550, 5500);

INSERT INTO PlayerRank (elo, rank) VALUES
(700, ‘Wood’),
(900, ‘Bronze’),
(1000, 'Silver'),
(1500, 'Gold'),
(1550, 'Gold'),
(1600, 'Platinum'),
(1700, 'Diamond'),
(2000, 'Elite'),

INSERT INTO PlayerLevel (xp, level) VALUES
(2000, 2),
(3000, 3),
(3200, 3),
(5000, 5),
(6000, 6),
(7000, 7),
(4000, 4),
(5500, 5);

INSERT INTO Match (matchID, mode, map, statsID1, statsID2, statsID3, statsID4, statsID5, statsID6, statsID7, statsID8, statsID9, statsID10) VALUES
(1, ‘5v5’', 'Map1', 101, 102, 103, 104, 105, 106, 107, 108, 109, 110),
(2, '2v2', 'Map2', 201, 202, 203, 204, 205, 206, 207, 208, 209, 210),
(3, 'PvE', 'Map3', 301, 302, 303, 304, 305, 306, 307, 308, 309, 310),
(4, '1v1', 'Map4', 401, 402, 403, 404, 405, 406, 407, 408, 409, 410),
(5, 'Training', 'Map5', 501, 502, 503, 504, 505, 506, 507, 508, 509, 510),
(6, ‘2v2’', 'Map1', 601, 602, 603, 604, 605, 606, 607, 608, 609, 610),
(7, ‘5v5’', 'Map3', 701, 702, 703, 704, 705, 706, 707, 708, 709, 710);

```
INSERT INTO MatchStats (statsID, matchID, playerID, kills, deaths, assists, outcome) VALUES
(101, 1, 1, 10, 2, 5, 'Win'),
(102, 1, 2, 8, 3, 4, 'Loss'),
(103, 1, 3, 6, 4, 3, 'Win'),
(104, 2, 4, 12, 1, 7, 'Win'),
(105, 2, 5, 9, 5, 6, 'Loss');

INSERT INTO MatchReward (efficacy, outcome, xpGained) VALUES
(0.8, 'Win', 500),
(0.7, 'Loss', 300),
(0.9, 'Win', 600),
(0.6, 'Loss', 200),
(0.85, 'Win', 550);

INSERT INTO Performance (kills, deaths, assists, efficacy) VALUES
(10, 2, 5, 0.8),
(8, 3, 4, 0.7),
(6, 4, 3, 0.65),
(12, 1, 7, 0.9),
(9, 5, 6, 0.75);

INSERT INTO Transaction (txID, playerID, itemID, balanceID, cost) VALUES
(1, 1, 1001, 201, 50),
(2, 2, 1002, 202, 75),
(3, 3, 1003, 203, 100),
(4, 4, 1004, 204, 150),
(5, 5, 1005, 205, 200);

INSERT INTO SaleAmount (itemID, cost, quantity) VALUES
(1001, 50, 10),
(1002, 75, 20),
(1003, 100, 15),
(1004, 150, 8),
(1005, 200, 5);
```

INSERT INTO ShopItemSold (itemID, shopID, title, cost, currName) VALUES
(1001, 1, 'Cape', 50, 'Gold'),
(1002, 1, 'Shield', 75, 'Silver'),
(1003, 2, 'Potion of Experience Gain', 100, 'Gold'),
(1004, 1, ‘Arrow’, 150, 'Silver'),
(1005, 1, 'Bow', 200, 'Essence');
(1006, 3, 'Festive Dress', 100, 'Essence'),
(1007, 2, 'Potion of Strength', 20, 'Gold'),
(1008, 3, 'Potion of Strength', 200, 'Essence'),
(1009, 3, 'Bow tie', 200, 'Essence'),
(1010, 3, 'Hat', 200, 'Essence'),

INSERT INTO Cosmetic (itemID, rarity) VALUES
(1001, 'Rare'),
(1002, 'Common'),
(1006, 'Common'),
(1009, 'Mythic'),
(1010, 'Legendary'),

INSERT INTO Consumable (itemID, expiration, effect) VALUES
(1003, 30, 'Gain 100 xp'),
(1004, 15, 'Does 20 damage'),
(1005, 20, 'Fire arrow, 20 uses'),
(1007, 20, 'Increases damage'),
(1008, 20, 'Increases damage'),

INSERT INTO ItemCategory (title, category) VALUES
('Cape', 'Cosmetic’),
('Shield', 'Cosmetic’),
('Potion of Experience Gain', 'Consumable'),
('Potion of Strength', 'Consumable'),
('Arrow', 'Consumable'),
('Bow', 'Consumable');

INSERT INTO CurrBalance (balanceID, playerID, currName, amount) VALUES
(201, 1, 'Gold', 1000),
(202, 2, 'Silver', 500),
(203, 1, 'Essence', 2000),
(204, 4, 'Gold', 1500),
(205, 5, 'Platinum', 3000);
(206, 2, 'Gold', 2000),

```
INSERT INTO Currency (currName, type) VALUES
('Gold', TRUE),
('Silver', FALSE),
('Platinum', TRUE);
(‘Essence’, FALSE)
(‘Artifacts’, FALSE)
(‘Relics’, TRUE)

INSERT INTO Shop (shopID, title) VALUES
(1, 'Weapon Shop'),
(2, 'Consumables Shop'),
(3, ‘Christmas Event Shop'),
(4, 'Magic Items Shop'),
(5, 'Rare Items Shop');

INSERT INTO Inventory (invID, playerID) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

INSERT INTO InventoryItem (invItemID, invID, itemID, quantity) VALUES
(1, 1, 1001, 2),
(2, 2, 1002, 3),
(3, 3, 1003, 5),
(4, 4, 1004, 1),
(5, 5, 1005, 4);
```