CPSC 304 Project Cover Page

Milestone #: 4

Date: April 2, 2025

Group Number: 114

Name	Student Number	CS Alias (userid)	Preferred Email Address
Alan Zhou	70409610	l5p5q	alanzhou318@gmail.com
Kevin Lei	60813573	l9b0l	kevinleimc@gmail.com
Wendi Liu	65854515	m5j4e	liu.wendi@yahoo.ca

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

A short description of the final project, and what it accomplished.

Our final project is focused on modelling data from a generic RPG video game using Oracle SQL. This database was designed to efficiently store and manage game-related data, such as a player's details, item inventory, quests, as well as missions. Our final project was optimized using BCNF in our database design, which led to more efficient data storage and retrievals. Constraints were also added to ensure data integrity and used to ensure that the necessary fields were never null. This final project is an interactive website that demonstrates the 10 SQL queries, and allows the user to see how some of the tables are structured.

A description of how your final schema differed from the schema you turned in. If the final schema differed, explain why.

We found that there were some other attributes that should be NOT NULL and added the constraints accordingly.
For the create table statements, we also added a few ON DELETE CASCADE statements to some of our tables as we realized they were missing.

Constraints: if assertions or triggers are required, you may simply say what they would need to do, but not implement them.

We would need an assertion that each entry in Guild is in the relation PlayerJoins due to its total participation in Joins. It also makes sense to have a check for Guild's memberCount to be > 0. For the guild member count update, whenever a new player joins a guild (or is deleted/updated), we want the Guild.memberCount to update automatically. We implemented this in JavaScript, with custom logic instead of using TRIGGER in SQL on the PlayerJoins table.
For Consumable and Collectible, we would need to assert the total constraint in the ISA relation, so that all entries in Item have a corresponding entry in either Consumable or Collectible.

A list of all SQL queries used to satisfy the rubric items and where each query can be found in the code (file name and line number(s)).

The SQL for the following queries can be found in *AppService.js*:

1. Query #1 - *insertPlayertable() - Line 112*

   INSERT INTO PlayerJoins VALUES (player_seq.NEXTVAL, :username, :email, SYSDATE, NULL, NULL)

2. Query #2 - *updateUserGuild() - Line 135*

   UPDATE PlayerJoins SET guildName = :guildName, role = :guildRole WHERE username = :username

   UPDATE Guild SET memberCount = (SELECT COUNT(*) FROM PlayerJoins P WHERE p.guildName = :guildName)

3. Query #3 - *deletePlayer() - Line 161*

   DELETE FROM PlayerJoins WHERE username = :username

4. Query #4 - *selectPlayerTuples() - Line 177*

   SELECT * FROM PlayerJoins WHERE ${query}

   The query variable is a Boolean expression consisting of:
   - Attributes (column names)
   - Operators (=, <, >)
   - Literals (user-input values wrapped in single quotes)
   - Logical operators (AND, OR)

5. Query #5 - *selectArmourTuples() - Line 191*

   SELECT ${query} FROM ArmourName

6. Query #6 - getUserArmour*() - Line 87*

   SELECT armourID, boostType, name
   FROM PlayerJoins PJ
   INNER JOIN CraftsArmour CA
   ON PJ.accountID = CA.accountID
   WHERE PJ.username = :username

7. Query #7 - *getMostPopularItems() - line 203*

   SELECT name AS itemName, COUNT(*) AS ownershipCount
        FROM Owns
        GROUP BY name
        ORDER BY ownershipCount DESC

   Aggregation GROUP BY query - this query gets the most popular items from the Owns table. Uses count and groups by the item name.

8. Query #8 - *getGuildsWithMoreThanTwoMembers() - line 218*

    SELECT guildName, COUNT(*) AS memberCount
        FROM PlayerJoins
        WHERE guildName IS NOT NULL
        GROUP BY guildName
        HAVING COUNT(*) > 1
        ORDER BY memberCount DESC

    Aggregation with HAVING query - this query retrieves all guids from the PlayerJoins table that have more than one member, counting the number of members in each guild. The results are grouped by guildName, filtered using the HAVING clause, and ordered by the member count in descending order.

9. Query #9 - *getGuildsWithAboveAverageFriendship() - line 235*

    WITH OverallAverage AS (
        SELECT AVG(B2.friendshipLevel) AS overallAvgFriendship
            FROM Befriends B2
)
    SELECT P.guildName,
            AVG(B.friendshipLevel) AS avgFriendshipLevel,
            AVG(B.friendshipLevel) - (SELECT overallAvgFriendship FROM
    OverallAverage) AS differenceFromAverage
    FROM PlayerJoins P
    JOIN Befriends B ON P.accountID = B.account1ID
    GROUP BY P.guildName
    HAVING AVG(B.friendshipLevel) > (SELECT overallAvgFriendship FROM
    OverallAverage)

    Nested Aggregation with GROUP BY query - this query calculates the average friendship level for each guild and compares it to the overall average friendship level across all friendships. It returns guilds where the average friendship level is higher than the overall average along with the difference from the overall average.

10. Query #10 - *getPlayersMissionsCompletedDivision() - line 274*

    CREATE VIEW GivenMissions(missionID) AS
        SELECT missionID
        FROM Mission
        WHERE (Note: the checkmarked missions in the frontend are given here in the
                format missionID = <id> OR missionID = <id> OR …>)

    SELECT accountID, username

```
FROM PlayerJoins PJ
WHERE NOT EXISTS (
        (SELECT missionID
         FROM GivenMissions)
         MINUS
        (SELECT DISTINCT missionID
        FROM Completes C
        WHERE PJ.accountID = C.accountID)
)
```

Division query - this query performs division on the players' accountIDs using the chosen missionIDs that are checkmarked by the user. It returns the accountIDs and usernames of the players who have completed all of the given missions.

Schema

- PlayerJoins(<u>accountID</u>: integer (PK), username: varchar(16) unique not null (CK), email: varchar(255) not null, createDate: date, role: varchar, **guildName**: varchar(16) (FK referencing Guild(name)))
- Guild(<u>name</u>: varchar(16), memberCount: integer not null, description: varchar(255))
- Befriends(**<u>accountID1</u>**: integer (PK, FK referencing PlayerJoins(accountID)), **<u>account2ID</u>**: integer (PK, FK referencing PlayerJoin(accountID)), friendshipLevel: integer not null)
- Mission(<u>missionID</u>: integer (PK), requirement: varchar(255), reward: varchar(255))
- Completes(**<u>accountID</u>**: integer (PK, FK referencing PlayerJoins), **<u>missionID</u>**: integer (PK, FK referencing Mission), completionTimestamp: date)
- Item(<u>name</u>: varchar(16) (PK), description: varchar(64))
- ConsumableStat(<u>rarity</u>: varchar(16) (PK), stat: integer not null, <u>boostType</u>: varchar(8) (PK))
- Consumable(**<u>name</u>**: varchar (PK, FK referencing Item), count: integer not null, rarity: varchar(16) not null, boostType: varchar(8) not null)
- Collectible(**<u>name</u>**: varchar (PK, FK referencing Item), obtainedFrom: varchar(255))
- Owns(**<u>accountID</u>**: integer (PK, FK referencing PlayerJoins), **<u>itemName</u>**: varchar(16) (PK, FK referencing Item))
- CharacterStats(<u>level</u>: integer (PK), attack: integer not null, defence: integer not null, <u>class</u>: varchar(8) (PK))
- CreatesCharacter(<u>name</u>: varchar(16) (PK), level: integer not null, class: varchar(8) not null, **playerID**: integer unique not null (FK referencing PlayerJoins(accountID)))
- ArmourStat(<u>rarity</u>: varchar(16) (PK), stat: integer, <u>boostType</u>: varchar(8) (PK), <u>equipType</u>: varchar(8) (PK))
- ArmourName(<u>name</u>: varchar(16) (PK), rarity: varchar(16) not null, equipType: varchar(8) not null, boostType: varchar(8) not null)
- CraftsArmour(<u>armourID</u>: integer (PK), name: varchar(16) not null, boostType: varchar(8) not null, **accountID**: integer not null (FK referencing PlayerJoins))
- Equips(**<u>armourID</u>**: integer (PK, FK referencing ArmourCrafts), **characterName**: varchar(16) (FK referencing Character(name)))
- LearnsSkill(<u>name</u>: varchar (PK), effect: varchar(64), **characterName**: varchar(16) (FK referencing Character(name)))
- QuestLocation(location: varchar, <u>bossCharacter</u>: varchar(255) (PK))
- QuestInfo(<u>questID</u>: integer (PK), bossCharacter: varchar(255))
- HaveStage(<u>stageID</u>: integer (PK), objective: varchar, **<u>questID</u>**: integer (PK, FK referencing Quest))
- Challenges(**<u>characterName</u>**: varchar(16) (PK, FK referencing Character(name)), **<u>questID</u>**: integer (PK, FK referencing Quest), attemptTimestamp: smalldatetime)

ArmourName

name varchar	rarity varchar	equiptype varchar	boosttype varchar
Breastplate	rare	upper	attack
Cool Shoes	epic	boots	attack
Fleet footwear	rare	boots	attack
Light headgear	common	helmet	defence
Nice helmet	rare	helmet	defence

ArmourStat

rarity varchar	boosttype varchar	equiptype varchar	stat int4
rare	attack	boots	3
common	attack	helmet	1
common	defence	helmet	2
rare	defence	helmet	4
rare	attack	upper	5
rare	defence	upper	6

Befriends

Account1ID	Account2ID	FriendshipLevel
1	2	10
2	3	23
2	5	5
3	5	1
7	8	5

Challenges

charactername varchar	q... i..	attempttimestamp timestamp
Bob	1	2025-01-01 00:00:00
Joe	1	2025-01-05 00:00:00
Bob	2	2025-01-01 00:00:00
Joe	2	2025-01-05 00:00:00
Joe	3	2025-01-06 00:00:00

CharacterStats

class varchar	characterlevel int4	attack int4	defence int4
fighter	1	5	3
mage	1	7	1
tank	1	3	5
fighter	2	6	4
mage	2	8	2
tank	2	4	6
fighter	3	8	4
mage	3	9	3
tank	3	4	8

Completes

AccountID	MissionID	CompletionTimestamp
1	101	2025-01-01T07:00:00.000Z
1	102	2025-01-01T07:00:00.000Z
1	103	2025-01-05T07:00:00.000Z
1	104	2025-01-05T07:00:00.000Z
1	105	2025-01-05T07:00:00.000Z
2	101	2025-01-01T07:00:00.000Z
2	103	2025-01-05T07:00:00.000Z
2	104	2025-01-05T07:00:00.000Z
3	101	2025-01-01T07:00:00.000Z
3	103	2025-02-05T07:00:00.000Z
5	101	2025-02-22T07:00:00.000Z
5	103	2025-02-22T07:00:00.000Z
7	103	2025-02-25T07:00:00.000Z
8	103	2025-02-25T07:00:00.000Z

Consumable

Name	Count	Rarity	BoostType
Sandwich	5	common	attack
Strength potion	2	rare	attack
Healing Herb	3	rare	health
Energy Drink	4	common	stamina
Elixir of Wisdom	1	legendary	mana

ConsumableStat

Rarity	Stat	BoostType
common	2	attack
common	4	defence
common	5	stamina
common	5	health
rare	5	attack
rare	8	defence
rare	10	stamina
rare	10	health
rare	10	mana
legendary	10	attack
legendary	16	defence
legendary	20	mana

Collectible

Name	ObtainedFrom
Bracelet	Mission reward
Crown	Mission reward
Ancient Coin	Hidden treasure
Golden Feather	Rare bird drop
Dragon Scale	Defeated a dragon

CraftsArmour

armourid int4	name varchar	boosttype varchar	accountid int4	
1	Light headgear	defence	1	→
2	Nice helmet	defence	1	→
3	Breastplate	attack	1	→
4	Breastplate	defence	2	→
5	Fleet footwear	attack	5	→

CreatesCharacter

name varchar	characterlevel int4	class varchar	playerid int4
Alice	2	mage	3
Bob	3	fighter	1
Cody	1	tank	9
Eve	2	fighter	4
Joe	3	tank	2
Kai	3	mage	7
Luna	2	tank	6
Max	1	mage	5
Zane	1	fighter	8

Equips

armourid int4	charactername varchar
1 →	Joe
2 →	Bob
3 →	Bob
4 →	Joe
5 →	Joe

Guild

Name	MemberCount	Description
Goats Guild	3	A guild for goats all around the world
Best Players	2	Only the best players are in this guild
Guild 1	1	guild 1 description
Guild 2	1	guild 2 description
Guild 3	1	guild 3 description

HaveStage

stageid int4	questid int4	objective varchar
1	1	defeat 3 enemies
2	1	find treasure chest
3	1	defeat all enemies
1	2	find hidden path
2	2	defeat all enemies
1	3	find hidden path
2	3	defeat all enemies
1	4	collect 5 herbs
1	5	reach location without being seen
2	5	defeat all enemies

Item

Name	Description
Bracelet	A token of your friendship
Crown	Reward for defeating your first boss
Ancient Coin	A rare coin from ancient times
Golden Feather	A shimmering feather from a rare bird
Dragon Scale	A tough scale from a defeated dragon
Sandwich	A tasty sandwich that boosts attack
Strength potion	A potion that increases your strength temporarily
Healing Herb	A magical herb that restores health
Energy Drink	A beverage that restores stamina
Elixir of Wisdom	A rare potion that increases mana

LearnsSkill

name varchar	effect varchar	charactername varchar
Alice Cast	Grants a 1.5x buff for magical attacks	Alice
Alice Dodge	50% chance of taking no damage	Alice
Bob Block	Decreases incoming damage by defence :	Bob
Bob Punch	Deals damage to enemy based on attack	Bob
Joe Protect	Decreases incoming damage by 2x defen	Joe

Mission

MissionID	Requirement	Reward
101	kill 3 monsters	10 coins
102	defeat a boss character	crown
103	add a friend	friendship bracelet
104	finish first quest	10 coins
105	finish all quests	100000 coins

Owns

accountid int4	name varchar
1	Bracelet
1	Crown
2	Bracelet
3	Bracelet
4	Ancient Coin
5	Bracelet
6	Dragon Scale
7	Bracelet
8	Bracelet

PlayerJoins

AccountID	Username	Email	CreateDate	Role	GuildName
1	playerone	iplaygames@email.com	2025-01-01T07:00:00.000Z	member	Best Players
2	thegoat	goatedgamer@email.com	2025-01-05T07:00:00.000Z	leader	Goats Guild
3	legoat	goatedgamer@email.com	2025-02-05T07:00:00.000Z	deputy	Goats Guild
4	dabest	bestest@email.com	2025-02-20T07:00:00.000Z	leader	Best Players
5	mountaingoat	mountaingoat@email.com	2025-02-21T07:00:00.000Z	member	Goats Guild
6	solo	soloplayer@email.com	2025-02-22T07:00:00.000Z		
7	lesunshine	a@gmail.com	2025-02-22T07:00:00.000Z	leader	Guild 1
8	lebron	b@gmail.com	2025-02-22T07:00:00.000Z	leader	Guild 2
9	leGM	c@gmail.com	2025-02-22T07:00:00.000Z	leader	Guild 3

QuestInfo

questid int4	bosscharacter varchar
1	Skeleton King
2	Ferocious Bear
3	Prickly Cactus
4	Aquatic Monstrosity
5	Arrogant Tyrant

QuestLocation

bosscharacter varchar	location varchar
Aquatic Monstrosity	Enchanted Lake
Arrogant Tyrant	Castle
Ferocious Bear	Forest
Prickly Cactus	Desert
Skeleton King	Dungeon