

# CPSC 304 Project Cover Page

Milestone #: 4

Date: 2024.11.29

Group Number: 20

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Akira Tong	46190278	d2g9x	at200611@gmail.com
Simon Fang	59536557	o7m7e	simonfang2002@gmail.com
Lucia Song	68849942	o3c4r	luciasong0415@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

All milestones and files can be found in: [https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project\\_d2g9x\\_o3c4r\\_o7m7e](https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_d2g9x_o3c4r_o7m7e)  
SQL script containing all DDL/insert/drop table statements can be found in the above repository under the name [all.sql](#)

## Project Description

This project develops a management system for **Identity V**, an asymmetrical survival horror game, featuring a relational database and responsive web interface. It tracks **player** profiles, **servers**, **characters** and their ownership through **character cards** (*survivors* and *hunters*), **guilds**, **teams**, and **talents**.

The backend supports dynamic data operations, including insertions, updates, cascading deletions, and advanced queries to retrieve insights like top-performing guilds and teams, or server rankings. The frontend offers intuitive forms and quick actions to interact with the database, enabling users to explore character stats, player-group relationships, and progression data.

## Differences Between Final Schema and Initial Schema

### Merged Table

- Initial Schema: Separate CardPrice and CardOwnership tables.
- Final Schema: Merged them into a single CharacterCard table, including price, cName, and pid to reduce redundancy and simplify queries

## SQL Queries for 2.1.1-2.1.10

### INSERT

(appService.js-row606)

```
INSERT INTO CharacterCard (cid, price, cName, pid) VALUES (:cid,  
:price, :cName, :pid)
```

### UPDATE

(appService.js-row659)

```
UPDATE Talent  
SET totalPoints = :tp, pid = :pid  
WHERE taid = :taid
```

(appService.js-row667)

```
UPDATE HunterTalent  
SET dread = :dr, vigilance = :v, deceit = :de, strength = :s  
WHERE taid = :taid
```

(appService.js-row730) note: this is for satisfying the requirement of displaying available tuples

```
SELECT *  
FROM HunterTalent  
NATURAL JOIN Talent
```

### DELETE

(appService.js-row686)

```
DELETE FROM Player WHERE pid = :pid
```

### SELECTION

(appService.js-row702)

```
SELECT pid, pName, hunterRank, survRank  
FROM Player  
WHERE pid = :pid
```

### PROJECTION

(appService.js-row763)

```
SELECT cName, ${stat} FROM Character
```

### JOIN

(appService.js-row781)

```
SELECT p.pid, p.pName, t.teid, t.tName, t.teamRank, t.tMemberNum  
FROM Player p
```

```
LEFT JOIN Team t ON p.teid = t.teid
WHERE :teid = t.teid
```

(appService.js-row797)

```
SELECT player.pid, player.pName, guild.gid, guild.gName, guild.gLevel,
guild.gMemberNum, guild.points
FROM player
LEFT JOIN guild ON player.gid = guild.gid
WHERE guild.gid = :gid
```

(appService.js-row813)

```
SELECT player.pid, player.pName, server.sid, server.country
FROM player
JOIN server ON player.sid = server.sid
WHERE :sid = server.sid
```

(appService.js-row829)

```
SELECT cName, codeName, speed, agility, ability, damage
FROM Hunter
NATURAL JOIN Character
NATURAL JOIN HunterStats
```

(appService.js-row844)

```
SELECT cName, employment, speed, agility, skill, decode
FROM Survivor
NATURAL JOIN Character
NATURAL JOIN SurvivorStats
```

(appService.js-row864)

```
SELECT MAX(${stat}) AS max_stat
FROM Survivor
NATURAL JOIN Character
NATURAL JOIN SurvivorStats
```

(appService.js-row886)

```
SELECT MAX(${stat}) AS max_stat
FROM Hunter
NATURAL JOIN Character
NATURAL JOIN HunterStats
```

## Aggregation with GROUP BY

(appService.js-row904)

```
SELECT sid, MAX(hunterRank)
FROM Player
```

```
NATURAL JOIN Server
GROUP BY sid
```

This query groups the players by server and retrieves the highest hunter rank for each server.

### Aggregation with HAVING

(appService.js-row934)

```
SELECT sid, AVG(survRank) AS avg_surv_rank
FROM Player
NATURAL JOIN Server
GROUP BY sid
HAVING sid = :sid
```

This query retrieves the average survivor rank for the server with the given server ID.

(appService.js-row952)

```
SELECT sid, AVG(hunterRank) AS avg_hunter_rank
FROM Player
NATURAL JOIN Server
GROUP BY sid
HAVING sid = :sid
```

This query retrieves the average hunter rank for the server with the given server ID.

### Nested aggregation with GROUP BY

(appService.js-row970)

```
SELECT gName
FROM Guild
NATURAL JOIN Player
GROUP BY gName, gid
HAVING AVG(hunterRank) >= ALL (SELECT AVG(hunterRank)
                                FROM Guild
                                NATURAL JOIN Player
                                GROUP BY gid)
```

This query finds the guild with maximum average hunter rank.

(appService.js-row989)

```
SELECT gName
FROM Guild
NATURAL JOIN Player
GROUP BY gName, gid
HAVING AVG(survRank) >= ALL (SELECT AVG(survRank)
                                FROM Guild)
```

```
NATURAL JOIN Player
GROUP BY gid)
```

This query finds the guild with maximum average survivor rank.

## DIVISION

(appService.js-row1008)

```
SELECT P.pName, P.pid
FROM Player P
WHERE NOT EXISTS (SELECT C.cName
                  FROM Character C
                  WHERE NOT EXISTS (SELECT CC.pid
                                   FROM CharacterCard CC
                                   WHERE C.cname = CC.cname
                                   AND CC.pid = P.pid))
```

This query finds players who own at least one character card for all characters.

A Table for quick reference :D

	Button/Section Name	Position in Code
INSERT	Insert Character Card	appService.js row 606
UPDATE	Update Hunter Talent	appService.js row 659, 667
	Show existing hunter talents	appService.js row 730
DELETE	Delete Player	appService.js row 686
Selection	Select Player	appService.js row 702
Projection	Show Character Speed	appService.js row 763
	Show Character Agility	
Join	Show Team Player Match	appService.js row 781
	Show Guild Player Match	appService.js row 797
	Show Server Player Match	appService.js row 813
	Show All Hunters	appService.js row 829

	Show All Survivors	appService.js row 844
	Max Survivor Decode	appService.js row 864
	Max Survivor Speed	
	Max Survivor Agility	
	Max Hunter Damage	appService.js row 886
	Max Hunter Speed	
	Max Hunter Agility	
Aggregation with GROUP BY	Max Hunter Rank for Each Server	appService.js row 904
Aggregation with HAVING	Average Survivor Rank in Server	appService.js row 934
	Average Hunter Rank in Server	appService.js row 952
Nested Aggregation with GROUP BY	Guild with Max Avg Hunter Rank	appService.js row 970
	Guild with Max Avg Survivor Rank	appService.js row 989
Division	Return Player With All Characters	appService.js row 1008