CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5th, 2024

Group Number: 5

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Bryan Jhutti	60442258	bjhutti	bryanjhutti@hotmail.com
Ronald Liu	10051118	rliu4936	ronaldliu2022@gmail.com
Victor Yao	26291401	vyao02	victoryao1039@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

2.

The script is in the GitHub repository under deliverables: https://github.students.cs.ubc.ca/CPSC304-2023W-
T2/project m5k1h s0b9v x7j4j/blob/e39118c82ed2eef936627c652ae48ddd15121f9
6/deliverables/startup.sql

3.

- a. The project is an e-sports tournament organization system, where organizers can track and manage participants, sponsors, venues, etc.
- b. The schema did not change from before.

C.

i. Broadcast(tid: int, host: string, viewership: int)

HostPlatform(host, platform)

ESportsOrg(id: int, name: string, region: string, contact: string)

Game(id: int, name: string, genre: string, company: string,

yearPublished: int)

IndividualGame(id: int)

TeamGame(id: int, teamSize: int, numTeams: int)

Organizer(id: int, name: string, email: string)

Participant(<u>id</u>: int, displayName: string)

Player(id: int, firstName: string, lastName: string, age: int)

Team(id: int, coach: string)

Sponsor(id: int, name: string)

Tournament(<u>id</u>: int, name: string, startDate: date, endDate: date, **streetAddress**: string, **city**: string, **country**: string, **organizerid**: int,

gameid: int)

StartDate(startDate: date, year: int)

Venue(streetAddress: string, postalCode: string)

PostalCode(postalCode: string, city: string, country: string)

Contract(**pid**: int, **orgid**: string, startDate: date, endDate: date, amount:

int)

Joins(pid: int, tid: int)

Member(playerid: int, teamid: int)

TournamentFunding(tid: int, sid, amount: int)

TeamPlays(teamid: int, gameid: int)

PlayerPlays(playerid: int, gameid: int)

d. All queries can be found in server/dbConnector.js.

INSERT: can be found in executeInsert() starting on line 131

INSERT ALL
INTO table (columns)

VALUES (valuesArr)
SELECT * FROM dual

DELETE: can be found in executeDelete() starting on line 197

DELETE FROM table WHERE where Clause

UPDATE: can be found in executeUpdate() starting on line 165

UPDATE table SET setList WHERE whereClause

Selection: can be found the in executeQuery() starting on line 87

SELECT selectList FROM fromList

WHERE whereClause GROUP BY groupList HAVING havingClause ORDER BY orderList Projection: can be found in executeQuery() starting on line 87

SELECT selectList FROM fromList WHERE whereClause GROUP BY groupList HAVING havingClause ORDER BY orderList

Join: can be found in getTeamPlayers() starting on line 225

SELECT *
FROM Member, Player
WHERE teamid = <teamId> AND
playerid = id

Aggregation with GROUP BY: can be found in getNumTournParticipants() starting on line 252

SELECT tid, COUNT(*) as PARTICIPANT_COUNT FROM Joins
GROUP BY tid

Aggregation with HAVING: can be found in getPopularGames() starting on line 277 or 286

SELECT id, name, genre, company, yearPublished FROM Game WHERE id IN (SELECT gameid FROM TeamPlays GROUP BY gameid HAVING COUNT(*) >= 3)

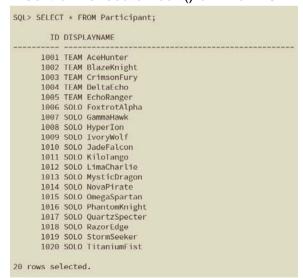
SELECT id, name, genre, company, yearPublished FROM Game WHERE id IN (SELECT gameid FROM PlayerPlays GROUP BY gameid HAVING COUNT(*) >= 5) **Nested Aggregation with GROUP BY:** can be found in getHighestAvgViewershipPlatform() starting on line 316

SELECT platform, AVG(viewership) as AVG_VIEWERSHIP FROM Broadcast NATURAL JOIN HostPlatform GROUP BY platform HAVING AVG(viewership) >= ALL (SELECT AVG(viewership) FROM Broadcast NATURAL JOIN HostPlatform GROUP BY platform)

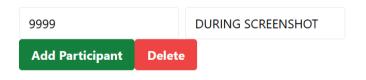
Division: can be found in getMVPs() on line 344

```
SELECT *
FROM Player p
WHERE NOT EXISTS (SELECT t.id
FROM Tournament t
WHERE t.gameid IN (SELECT g.id
FROM IndividualGame g)
MINUS
SELECT j.tid
FROM Joins j
WHERE j.pid = p.id)
```

e. **Insertion:** executeInsert() on line 119



Participants



```
SQL> SELECT * FROM Participant;
        ID DISPLAYNAME
      1001 TEAM AceHunter
      1002 TEAM BlazeKnight
      1003 TEAM CrimsonFury
      1004 TEAM DeltaEcho
      1005 TEAM EchoRanger
      1006 SOLO FoxtrotAlpha
      1007 SOLO GammaHawk
      1008 SOLO HyperIon
      1009 SOLO IvoryWolf
      1010 SOLO JadeFalcon
      1011 SOLO KiloTango
      1012 SOLO LimaCharlie
      1013 SOLO MysticDragon
      1014 SOLO NovaPirate
      1015 SOLO OmegaSpartan
      1016 SOLO PhantomKnight
      1017 SOLO QuartzSpecter
      1018 SOLO RazorEdge
      1019 SOLO StormSeeker
      1020 SOLO TitaniumFist
      9999 DURING SCREENSHOT
21 rows selected.
```

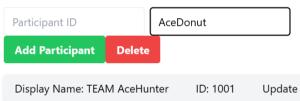


```
1001 TEAM AceHunter
      1002 TEAM BlazeKnight
      1003 TEAM CrimsonFury
      1004 TEAM DeltaEcho
      1005 TEAM EchoRanger
      1006 SOLO FoxtrotAlpha
      1007 SOLO GammaHawk
      1008 SOLO HyperIon
      1009 SOLO IvoryWolf
      1010 SOLO JadeFalcon
      1011 SOLO KiloTango
      1012 SOLO LimaCharlie
      1013 SOLO MysticDragon
      1014 SOLO NovaPirate
      1015 SOLO OmegaSpartan
      1016 SOLO PhantomKnight
      1017 SOLO QuartzSpecter
      1018 SOLO RazorEdge
      1019 SOLO StormSeeker
      1020 SOLO TitaniumFist
20 rows selected.
```

Update: executeUpdate() on line 153

```
SQL> SELECT * FROM Participant;
        ID DISPLAYNAME
      1001 TEAM AceHunter
      1002 TEAM BlazeKnight
      1003 TEAM CrimsonFury
      1004 TEAM DeltaEcho
      1005 TEAM EchoRanger
      1006 SOLO FoxtrotAlpha
      1007 SOLO GammaHawk
      1008 SOLO HyperIon
      1009 SOLO IvorvWolf
      1010 SOLO JadeFalcon
      1011 SOLO KiloTango
      1012 SOLO LimaCharlie
      1013 SOLO MysticDragon
      1014 SOLO NovaPirate
      1015 SOLO OmegaSpartan
      1016 SOLO PhantomKnight
      1017 SOLO QuartzSpecter
      1018 SOLO RazorEdge
      1019 SOLO StormSeeker
      1020 SOLO TitaniumFist
20 rows selected.
```

Participants



```
SQL> SELECT * FROM Participant;
       ID DISPLAYNAME
     1001 AceDonut
     1002 TEAM BlazeKnight
     1003 TEAM CrimsonFury
     1004 TEAM DeltaEcho
     1005 TEAM EchoRanger
     1006 SOLO FoxtrotAlpha
     1007 SOLO GammaHawk
     1008 SOLO HyperIon
     1009 SOLO IvoryWolf
     1010 SOLO JadeFalcon
     1011 SOLO KiloTango
     1012 SOLO LimaCharlie
     1013 SOLO MysticDragon
     1014 SOLO NovaPirate
     1015 SOLO OmegaSpartan
     1016 SOLO PhantomKnight
     1017 SOLO QuartzSpecter
     1018 SOLO RazorEdge
     1019 SOLO StormSeeker
     1020 SOLO TitaniumFist
     9999 DURING SCREENSHOT
21 rows selected.
```

Selection: executeQuery() on line 217

Participants id=1001 AND displayNam Participant ID Display Name Add Participant Delete Display Name: AceDonut ID: 1001 Update Show MVP

Join: getTeamPlayers() on line 224

Team Details

ID: 1001

Coach: Danny Sorensen

Players

- 1001 Max Carter, Age: 18
- 1002 Ruby Walsh, Age: 24
- 1003 Leo Fletcher, Age: 16
- 1004 Zoe Patel, Age: 21
- 1005 Oscar Hughes, Age: 19

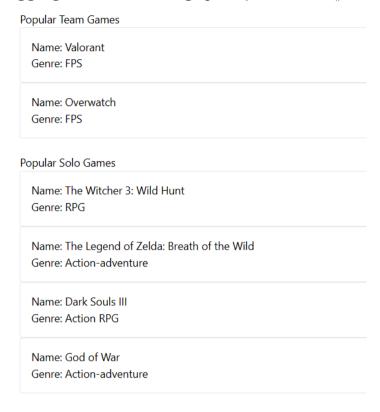
Aggregation with Group By: getNumTournParticipants() on line 252

Solo Cup

ID: 1

Participant count: 2

Aggregation with Having: getPopularGames() on line 269



Nested Aggregation with Group By: getHighestAvgViewershipPlatform() on line 304

Streams Page

Highest Viewership

Streams

SteamTV with 2000 viewers

Division: getMVPs() on line 332

Show MVP
ID: 1001 Name: Max Carter