

# **CPSC 304 Project Cover Page**

Milestone #: 4

Date: 27/11/23

Group Number: 135

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Yahya Abouelmagd	54403621	x0p5w	yaya.almajd@gmail.com
Andrew Piemonte	90501727	a0z2r	andrewpiemonte@gmail.com
Jana Sheirah	75867630	j4c7k	jana.sheirah@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

## Section 1 - Project Description

This section highlights a short description of final project and what it accomplished

This app is all about entertainment, focusing on video games and tournaments. It aims to include various gaming genres and tournaments within the industry. The main idea is to connect these elements through the players.

The database provides a wealth of gaming-related features for users. It allows you to easily access detailed information about a particular game, including its edition, reviews, compatible platforms, and genres. You can also discover the company behind the game and learn about both casual and professional players associated with it. Furthermore, the database offers comprehensive tournament details, such as the number of participants, the roster of players who competed, and the ultimate victor.

## Accomplished

Users can effortlessly insert, update, and delete tuples within any table in the database through an intuitive and user-friendly interface. Additionally, users have the flexibility to project any desired combination of columns from tables in the database. When utilizing the company table, users are empowered to perform searches for tuples using any number of AND/OR clauses, enhancing the precision of their queries.

For more advanced queries, our system provides the capability to:

- Retrieve all players in a given tournament.
- Determine the number of games on each platform.
- Calculate the average rating of games surpassing a specified threshold.
- **Identify all players who are yet to win a tournament. (NEEDS USER INPUT)**
- Find the average number of participants in games produced by a specific company.

## How final schema differed from the schema was turned in

The identical schema as the previous milestone was employed; there is no distinction in the schema.

## Schema and screenshots that show what data is present

Player(Player ID, Name)  
Professional Player(Rank, Benefits)  
ProfessionalPlayer (PlayerID,Rank,)  
Platform (Type,Company)  
CompanyAddressInfo (PostalCode,City,Province)  
Company (PostalCode,OfficeNum,Street,Name,CEO)  
Genres (Type)  
GameSeriesPlayers (PlayersNum,Popularity)  
GameSeriesWebsite (GameWebsite,GameName)  
GameSeriesMadeBy (GameID,OfficeNum,Street,PostalCode,PlayersNum,GameWebsite)  
TournamentWasAbout(TournamentName,TournamentDate,GameID,Winner,Prize,ParticipantsN  
um)  
PlayedOn (TournamentName,TournamentDate,PlayerID)  
Classification (GenreType,GameID)  
PlaysOn (PlatformType,GameID)  
Play (PlayerID,GameID)  
GameEditionHave (GameID,Edition,ReleaseDate)  
ReviewEvaluatesRating (RatingScore,RatingCategory)  
ReviewEvaluates (ReviewID,PlayerID,GameID,RatingScore)

Note that the rubric states: “This rubric item can be accomplished by mocking up the database instance in a spreadsheet program like Excel and exporting it as a PDF.”

Refer to **instance.pdf** for this requirement under milestone 4 directory.

## Section 1 - Queries

A list of all SQL queries used and where it can be found in the code (file name and line numbers)

Screenshots demonstrating the functionality of each query using the GUI

- Before you run the query (data in table) /during (how the query is triggered) /after progression of events (what data is in your table after)
- Label each screenshot with name of query it is meant to address from rubric

## Queries: INSERT

## Code at: projection.php: 145

Query:

INSERT INTO ProfessionalPlayer values (\$placeholdersString);

Ex: INSERT INTO ProfessionalPlayer values (playerId, Rank);

Before:

Retrieved data from table PROFESSIONALPLAYER:	
PLAYERID	RANK
1	Bronze
2	Silver
3	Gold
4	Platinum
5	Diamond

**Database Tables Projection Interface**

[Back](#)

Select a table ▾

During inserting:

## Database Tables Projection Interface

Back

PROFESSIONALPLAYER ↴

Insert    Update    Delete    Project

### Insert into PROFESSIONALPLAYER

PLAYERID:

RANK:

Insert

After:

Retrieved data from table PROFESSIONALPLAYER:

PLAYERID	RANK
6	Bronze
1	Bronze
2	Silver
3	Gold
4	Platinum
5	Diamond

## Database Tables Projection Interface

Back

Select a table ↴

## Queries: DELETE

Code at: projection.php: 262

Query:

DELETE FROM ProfessionalPlayerRank Where \$conditionString  
Ex: DELETE FROM ProfessionalPlayerRank WHERE playerid = \$playerid;

Screenshots

Before:

Retrieved data from table PROFESSIONALPLAYERRANK:	
RANK	BENEFITS
Bronze	3
Silver	3
Gold	2
Platinum	2
Diamond	1

**Database Tables Projection Interface**

Back

Select a table ▾

During:

**Database Tables Projection Interface**

Back

PROFESSIONALPLAYEI ▾

Insert   Update   Delete   Project

**Delete from PROFESSIONALPLAYERRANK**

RANK:  BENEFITS:

Delete

After:

Retrieved data from table PROFESSIONALPLAYERRANK:

RANK	BENEFITS
Silver	3
Gold	2
Platinum	2
Diamond	1

**Database Tables Projection Interface**

[Back](#)

Select a table ▾

## Queries: UPDATE

### Code at: projection.php: 198

Query:

UPDATE TournamentWasAbout Set Winner = \$winner WHERE Winner = \$oldWinner

UPDATE \$tableName SET \$updateString WHERE \$conditionString

Screenshots

Before:

Retrieved data from table TOURNAMENTWASABOUT:

TOURNAMENTNAME	TOURNAMENTDATE	GAMEID	WINNER	PRIZE	PARTICIPANTSNUM
Tournament A	20-OCT-23	12118088	Andrew	\$10,000	100
Tournament B	21-OCT-23	12446708	Andrew	\$9,000	75
Tournament C	22-OCT-23	1255546	Jessica	\$7,500	50
Tournament D	23-OCT-23	26419909	Jessica	\$2,500	40
Tournament E	24-OCT-23	7887676	Jana	\$0	30

**Database Tables Projection Interface**

[Back](#)

Select a table ▾

During:

## Database Tables Projection Interface

[Back](#)

TOURNAMENTWASABOUT

[Insert](#) [Update](#) [Delete](#) [Project](#)

### Update TOURNAMENTWASABOUT

Conditions (Specify which record to update):

TOURNAMENTNAME (condition): Tournament C  
 TOURNAMENTDATE (condition): 22-OCT-23  
 GAMEID (condition): 1255546  
 WINNER (condition):  
 PRIZE (condition):  
 PARTICIPANTSNUM (condition):

New Values:

TOURNAMENTNAME (new value):  
 TOURNAMENTDATE (new value):  
 GAMEID (new value):  
 WINNER (new value): Updated Winner  
 PRIZE (new value):  
 PARTICIPANTSNUM (new value):

[Update](#)

After:

Retrieved data from table TOURNAMENTWASABOUT:

TOURNAMENTNAME	TOURNAMENTDATE	GAMEID	WINNER	PRIZE	PARTICIPANTSNUM
Tournament A	20-OCT-23	12118088	Andrew	\$10,000	100
Tournament B	21-OCT-23	12446708	Andrew	\$9,000	75
Tournament C	22-OCT-23	1255546	Updated Winner	\$7,500	50
Tournament D	23-OCT-23	26419909	Jessica	\$2,500	40
Tournament E	24-OCT-23	7887676	Jana	\$0	30

## Database Tables Projection Interface

[Back](#)

Select a table

## Queries: Selection

### Code at: company.php:304

Query:

```
SELECT * FROM Company WHERE $inputString
```

```
SELECT * FROM Company WHERE PostalCode = $postalCode AND Street = $street
```

Screenshots:

All tuples:

PostalCode:

AND	OfficeNum:	
AND	Street:	
Back	AND	Name:
		Result:
AND	CEO:	

**displayTuples**

PostalCode ID	OfficeNum	Street	Name	CEO
K1A0A1	111	Main Street	ABC Inc	John Doe
M5V2H1	234	Ross Street	Six Guys LLC	Ahmed Bin Sulaiman
H3B4G7	244	Jordan Street	Jordan Inc	Jordan Jordan
V6B4M9	455	Wall Street	New York	Jordan Belfort
R3C0K6	678	Main Street	DEF Inc	Dohn Joe

Showing the query that we will run:

PostalCode:

K1A0A1	OR	OfficeNum:
234		
AND	Street:	
Back	AND	Name:
		Result:
AND	CEO:	

**displayTuples**

PostalCode ID	OfficeNum	Street	Name	CEO
K1A0A1	111	Main Street	ABC Inc	John Doe
M5V2H1	234	Ross Street	Six Guys LLC	Ahmed Bin Sulaiman
H3B4G7	244	Jordan Street	Jordan Inc	Jordan Jordan
V6B4M9	455	Wall Street	New York	Jordan Belfort
R3C0K6	678	Main Street	DEF Inc	Dohn Joe

Result of the above query:

The screenshot shows a user interface for querying a database. On the left, there is a search form with the following fields and logic:

- PostalCode:
- AND  OfficeNum:
- AND  Street:
- Back
- AND  Name:
- AND  CEO:

On the right, there is a "displayTuples" button and a results table:

PostalCode ID	OfficeNum	Street	Name	CEO
K1A0A1	111	Main Street	ABC Inc	John Doe
M5V2H1	234	Ross Street	Six Guys LLC	Ahmed Bin Sulaiman

Showing the query that we will run:

The screenshot shows a user interface for querying a database. On the left, there is a search form with the following fields and logic:

- PostalCode:
- AND  OfficeNum:
- AND  Street:   
Main Street
- Back
- OR  Name:   
New York
- OR  CEO:   
Ahmed Bin Sulaiman

On the right, there is a "displayTuples" button and a results table:

PostalCode ID	OfficeNum	Street	Name	CEO
K1A0A1	111	Main Street	ABC Inc	John Doe
M5V2H1	234	Ross Street	Six Guys LLC	Ahmed Bin Sulaiman
H3B4G7	244	Jordan Street	Jordan Inc	Jordan Jordan
V6B4M9	455	Wall Street	New York	Jordan Belfort
R3C0K6	678	Main Street	DEF Inc	Dohn Joe

Result of the above query:

PostalCode:

AND	OfficeNum:	
AND	Street:	
Back	AND	Name:
AND	CEO:	
<b>displayTuples</b>		

Result:

PostalCode ID	OfficeNum	Street	Name	CEO
K1A0A1	111	Main Street	ABC Inc	John Doe
M5V2H1	234	Ross Street	Six Guys LLC	Ahmed Bin Sulaiman
V6B4M9	455	Wall Street	New York	Jordan Belfort
R3C0K6	678	Main Street	DEF Inc	Dohn Joe

## Queries: Projection

Code at: **projection.php: 114**

Query:

SELECT \$columnString FROM \$tableName

Ex: SELECT PostalCode FROM Company

Search input:

## Database Tables Projection Interface

Back

COMPANY ▾

Insert Update Delete Project

### Project Columns from COMPANY

POSTALCODE  
 OFFICENUM  
 STREET  
 NAME  
 CEO

Project Columns

Result of the search:

POSTALCODE	
H3B4G7	
K1A0A1	
M5V2H1	
R3C0K6	
V6B4M9	

## Database Tables Projection Interface

Back

Select a table ▾

Search input:

## Database Tables Projection Interface

Back

COMPANY

Insert

Update

Delete

Project

### Project Columns from COMPANY

- POSTALCODE
- OFFICENUM
- STREET
- NAME
- CEO

Project Columns

Result of the search:

Retrieved data from table COMPANY:

POSTALCODE	STREET
H3B4G7	Jordan Street
K1A0A1	Main Street
M5V2H1	Ross Street
R3C0K6	Main Street
V6B4M9	Wall Street

## Database Tables Projection Interface

Back

Select a table

Search input:

**Database Tables Projection Interface**

Back

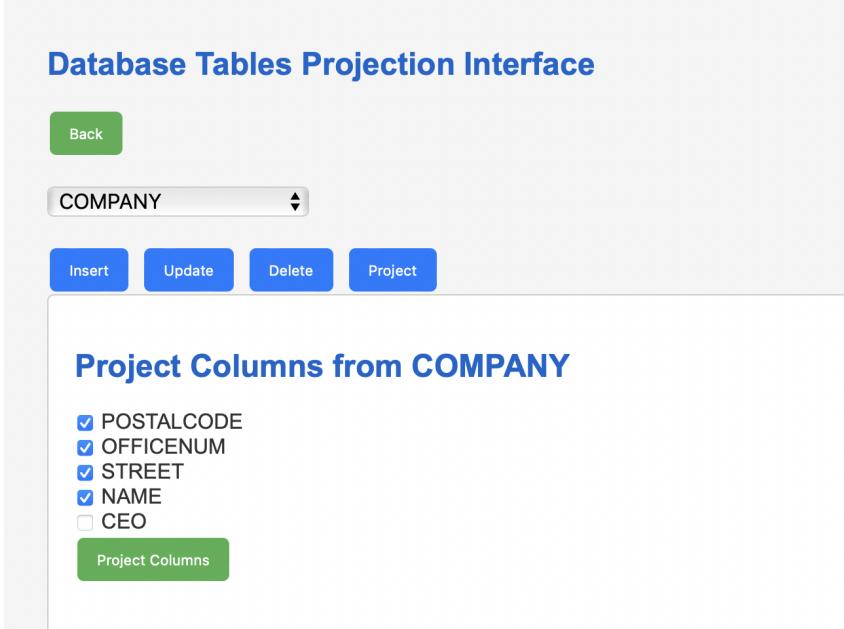
COMPANY

Insert    Update    Delete    Project

**Project Columns from COMPANY**

POSTALCODE  
 OFFICENUM  
 STREET  
 NAME  
 CEO

Project Columns



Result of the search:

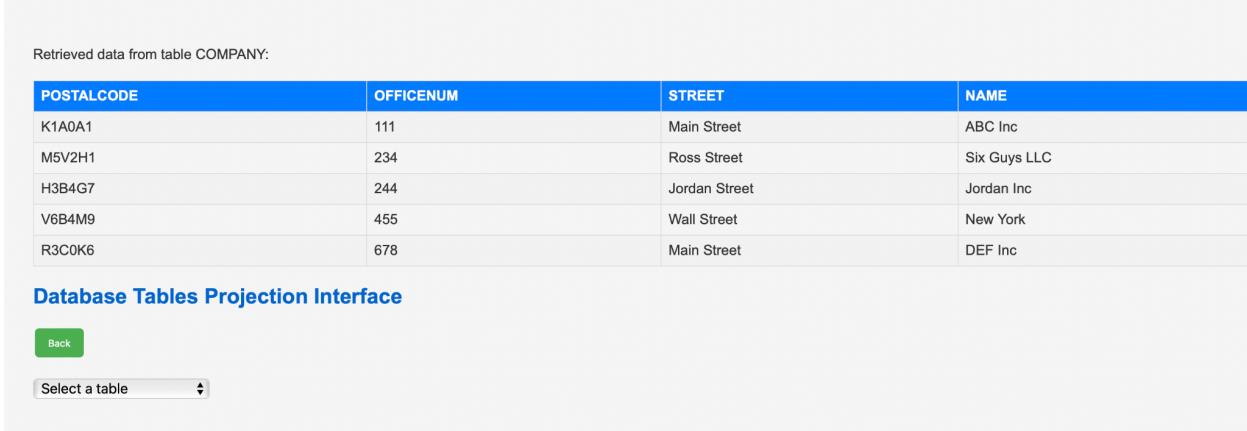
Retrieved data from table COMPANY:

POSTALCODE	OFFICENUM	STREET	NAME
K1A0A1	111	Main Street	ABC Inc
M5V2H1	234	Ross Street	Six Guys LLC
H3B4G7	244	Jordan Street	Jordan Inc
V6B4M9	455	Wall Street	New York
R3C0K6	678	Main Street	DEF Inc

**Database Tables Projection Interface**

Back

Select a table



## Queries: Join

Code at: advancedQueries.php at line 352

Query:

```
SELECT DISTINCT P.name
FROM Player P, PlayedOn PO
```

```
WHERE PO.TournamentName = $tournamentName AND PO.TournamentDate =  
TO_DATE($tournamentDate, 'YYYY-MM-DD') AND PO.PlayerId = P.PlayerId);
```

## Screenshots

Input:

**Advanced Search Page**

[Back](#)

**Finds all players in a given tournament**

Data format should be: YYYY-MM-DD

TournamentName(Key):  
Tournament A

TournamentDate(Key):  
2023-10-20

2023-10-20

Result:

Retrieved data:	
Name	
Sam	
Jana	
Andrew	
Yahya	
Jessica	

## Queries: Aggregation with Group By

Code at: advancedQueries.php at line 383

Query:

### All platforms:

```
SELECT Count(*), PlatformType  
FROM PlaysOn  
GROUP BY PlatformType);
```

### Specific platforms:

```
SELECT Count(*), PlatformType  
FROM PlaysOn  
WHERE PlatformType = $platformType GROUP BY PlatformType
```

## Screenshots

Count for all platforms:

Retrieved data:	
Count	Platform Type
1	G16 Gaming Laptop
1	M18 Gaming Laptop
1	Playstation 4
1	Victus Gaming Laptop
1	XBOX

Input a specific platform:

#### Finds the number of games on each platform

PlatformType:

To insert new values into the platforms table: PlatformType(Key):

Output:

Retrieved data:

Count	Platform Type
1	XBOX

## Queries: Aggregation with Having

Code at: advancedQueries.php at line 412

Query:

```
SELECT g.gameId, AVG(r.ratingScore) AS AverageRating
FROM GameSeriesMadeBy g, ReviewEvaluates r
WHERE g.gameId = r.gameId
GROUP BY g.gameId
HAVING AVG(r.ratingScore) > $ratingScore
```

Screenshots

Input:

#### Find the average rating of games above a threshold

Input a rating score to set the threshold RatingScore:

Note: combinations of keys MUST be unique

Insert new values for rating table: ReviewID(Key):

Output:

Retrieved data:	
Game ID	Average Rating Score
1255546	4
12446708	2.5
26419909	5

## Queries: Nested Aggregation with Group By

Code at: advancedQueries.php at line 442

Query:

```
SELECT T.GameId, AVG(T.ParticipantsNum) AS AvgParticipantsPerGame
FROM TournamentWasAbout T
WHERE T.GameId IN (SELECT GSM.GameId
                    FROM GameSeriesMadeBy GSM
                    WHERE GSM.PostalCode IN (SELECT C.PostalCode
                                              FROM Company C
                                              WHERE C.Name = $companyName))
GROUP BY T.GameID);
```

Screenshots:

Query result:

**Find Average Number of Participants for Games Made by a Specific Company**

Company Name:

Output:

Retrieved data:	
Game ID	Average Participants Per Game
12118088	100

## Queries: Division

Code at: advancedQueries.php at line 497 and line 469

Query:

```
SELECT P.PlayerID, P.Name
FROM Player P
```

MINUS

```
SELECT PL.PlayerID, PL.Name  
FROM Player PL, TournamentWasAbout T  
WHERE PL.Name = T.Winner AND T.TournamentDate >=  
TO_DATE("$.startdate.",'YYYY-MM-DD') AND T.TournamentDate <=  
TO_DATE("$.enddate.",'YYYY-MM-DD')
```

**OR To find all players who were in all tournaments:**

```
SELECT PlayerID, Name  
FROM Player P  
WHERE NOT EXISTS (SELECT 1  
                  FROM TournamentWasAbout T  
                  WHERE NOT EXISTS (SELECT 1  
                                    FROM PlayedOn PO  
                                    WHERE PO.PlayerID = P.PlayerID AND  
                                          PO.TournamentName = T.TournamentName AND  
                                          PO.TournamentDate = T.TournamentDate));
```

Screenshots

Input:

**Find All Players Who are Yet to Win a Tournament given a time frame**

If no time frame is given, it will show all players who have never won a tournament

DateStart:  
2023-10-20

DateEnd:  
2023-10-24

Result: Note in this period the other players won tournaments, therefore they don't show up

Retrieved data:  
>

Player ID	Player Name
1	Yahya
5	Sam

Input:

**Find All Players Who are Yet to Win a Tournament given a time frame**

If no time frame is given, it will show all players who have never won a tournament

DateStart:

DateEnd:

Result: In this period none of the player won tournaments, so they all show up

Retrieved data:  
>

Player ID	Player Name
1	Yahya
2	Jana
3	Andrew
4	Jessica
5	Sam
6	John

OR Finding players who were in all tournaments:

**Find Players who were in all tournaments**

Retrieved data:  
>

Player ID	Player Name
1	Yahya