addUser()

INSERT INTO Users(displayName, email,password) VALUES(X,Y,Z);

- a unique UID should be automatically generated by the db

modifyUser( )

UPDATE Users

SET displayName = X,

email=Y

WHERE

usid = currentUser;

- X and Y should default to their original values if not given on form

removeUser( )

DELETE

FROM

Users

WHERE

usid = currentUser;

- this should be a cascade, not sure if it does that by default. Possible that we will have to modify the db with ON DELETE CASCADE constraints for foreign keys.

SearchGame( )

SELECT \*

FROM Games

WHERE userInput IN (title, genre, developer, price);

--works on exact matches only

SELECT title, date, price, genre, developer

FROM Games

WHERE title LIKE %userInput%' or genre LIKE '%userInput%' or developer LIKE '%userInput%'

--should work for substrings and less precise searches

buyGame( )

// if payment accepted

INSERT INTO Owns VALUES (currentUser, gid);

earnCard( )

giftGame( )

INSERT INTO Owns Values

(

(

SELECT usid

FROM Users

WHERE email = recipientEmail

)

, gid);

earnCard( )

INSERT INTO Collected VALUES (usid,

(SELECT tid

FROM TradingCard

WHERE gid = game

ORDER BY random() LIMIT 1

),game);

- select statement returns a random trading cards for a game, not sure how to make it insert into collected for each row

Displaying a users library of games

SELECT title, developer, genre

FROM Owns NATURAL JOIN Games

WHERE usid = currentUser

Display earned Achievements for a game

select title,name,ptValue

from Earn natural join Achievements natural join (select gid,title from Games)

where usid = currentUser

Calculating totalPoints for a User

SELECT SUM(ptValue)

FROM Earn NATURAL JOIN Achievements

Where usid = currentUser

Calculating number of trading cards in circulation

WITH SomebodyOwns AS (SELECT gid, tid, title, usid FROM Collected NATURAL JOIN TradingCard)

SELECT game,title,abundance

FROM

(

SELECT \*, 0 abundance

FROM

(

SELECT gid,tid,title

FROM TradingCard

EXCEPT

SELECT gid,tid,title

FROM SomebodyOwns

)

UNION

SELECT gid,tid,title,count(\*) abundance

FROM SomebodyOwns

GROUP BY title

)NATURAL JOIN (select gid,title game from Games);

Calculate how many copies a game has sold

WITH SomebodyOwns AS (SELECT title FROM Games NATURAL JOIN Owns)

SELECT \*

FROM(

SELECT title, 0 abundance

FROM(

SELECT title from Games

EXCEPT

SELECT title from SomebodyOwns)

UNION

SELECT title, COUNT(\*) abundance

FROM Games NATURAL JOIN Owns GROUP BY title);