**Database Management**

**CSVI 1550:60**

**May 10, 2017**

Shilo Yahya

Jon Ferneau

Blaine Jacques

**CSVI 1550:60 Database Management May 10th, 2017**

Shilo Yahya

Jon Ferneau

Blaine Jacques

**Title:**

Our Company for Couplesis focused on matching married couples with similar interests to build great friendships.

**Description:**

* It's easy, exciting, and fun to connect with other couples.
* You'll improve and strengthen your own marriage.
* You'll build friendships with people that can relate to your marital experiences.
* Mentor younger couples through some of the hurdles of Marriage.
* Be Mentored by more mature marriages to better weather the seasons of Marriage.

Company for Couples was created by our desire for married couples to find other, compatible, married couples to go on double dates! We are committed to helping you find other couples that can help grow and encourage your own relationship through fellowship. Company for Couples matches based on compatibility of interests, and what type of couple you are looking for. You and are wife are best friends right? Why

**Information Needs:**  We need to have users be protected with a user-name and password. To simplify the registration we need two couples under one registration. They then put in some demographics about themselves. We then need to have them put in the interests they have and what they are looking for. Do they want a mentor couple, new friends in the same age group, are they new to the area? We then need to match the couples with each other depending on what they are interested in getting and the area they live in.

**Phase 1 - Model**

* Shiloh, Jon, and Blaine all worked on the design of the business and ER Diagram, including putting it into a diagram (It was a Google Drive based program that allowed multiple users to edit it at the same time).
* Blaine worked on documentation/business plan

**Phase 2 - Implementation**

* Shiloh supplied information to be input into tables.
* Jon built the tables and went through the coding with Blaine and Shiloh.
* Blaine worked on documentation.
* We all worked on troubleshooting the code.
* We had saw that it would be easier to implement our tables by breaking all of the interests into separate tables. This will make it easier to implement how we are going to match users together.
* Had some trouble understanding the Oracle Application Express, but by teamwork and help from the professor, we were able to get it.
* Foresee a hard time getting the matching criteria together and thinking about trying to implement the frontside to work with our database, but think that might be a bit too ambitious.
* Having a bit of trouble with the table creation and the inserting values into the table.

**Phase 3 - Final Reports**

* Shiloh worked on the documentation, business questions and Final thoughts
* Jon worked on the business questions and built sql statements.
* Blaine filled holes in the project, cleaned up code, ER Diagram, and up final paper.
* We all worked on troubleshooting the code.

**Business Questions**

Question 1. Which users like American football?

Question 2. Which users like baseball?

Question 3. Which users like soccer?

Question 4. Grab users from activity table whose USERID is greater than 4 and group them by the exercise they do?

Question 5.Get every user and group by exercise?

Question 6. What users like asian food?

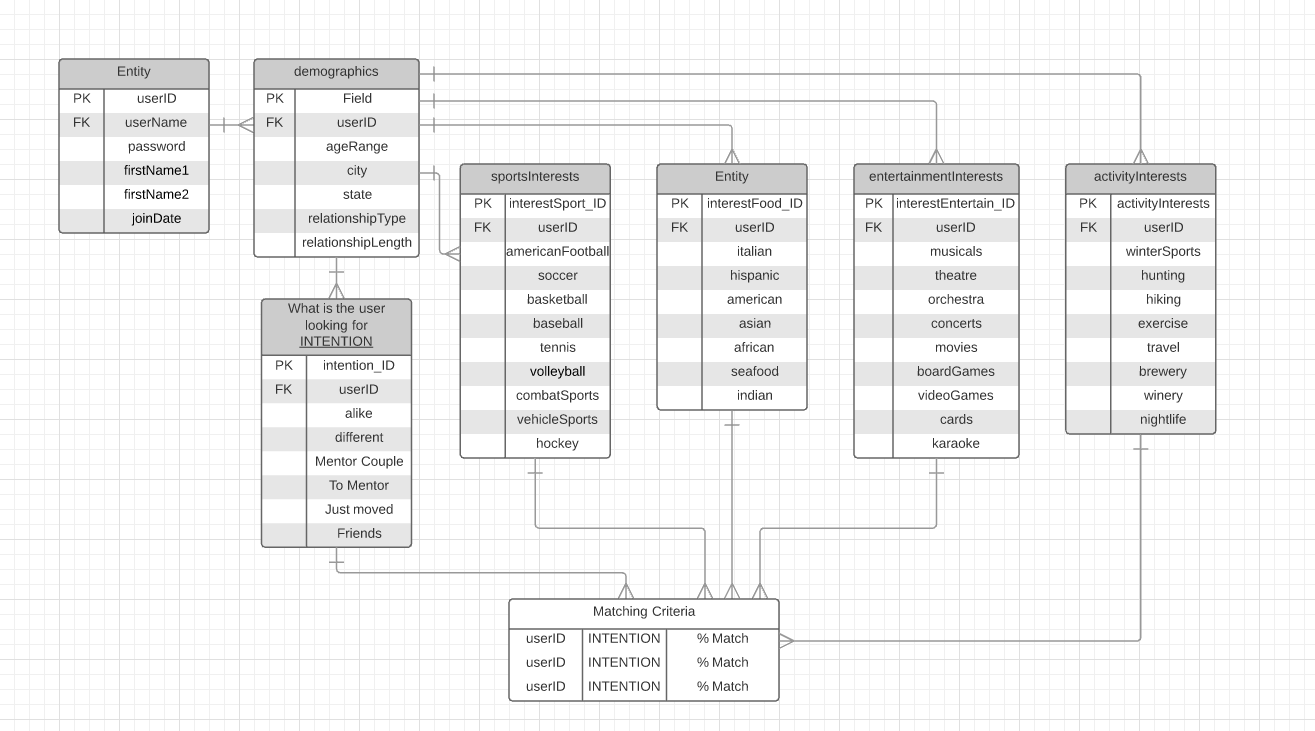
I ended up adding more questions, trying a lot of different ways to get responses out of sequel (‘the especially fun part’)

**Final Thoughts**

We learned about a complexed of data for the last 16 weeks, and the database needs a great effort to understand the database concepts entirely. The easiest part our project was a logical diagram, conceptual diagram, and physical diagram because of the data model for given the relational schema. After we had finished our logical and physical design, we all agreed to focus on the SQL by using DDL and DML.

We defined the data manipulation languages, and syntax correlation consumed us a significant time because SQL needs to pay particular attention when we manipulated the DDL and DML coding languages. We went to step by step when we wrote a SQL because wanted to understand the SQL concept and manage codes correctly. The takeaway motive of metadata is to provide context for data. The SQL coding languages are unique, and it needs to define the data structure because our business requires SQL statement implementation and cares as well. When we wrote the queries, we revised some tables because we need to define data with an important data type. Thus, to write the SQL statement for creating the tables, it requires a specify data types and field lengths. The SQL needs to find primary keys and foreign keys to implement any other constraints that we might have identified.

The challenges parts for our project that we experienced is SQL coding because it requires a lot effort and carefully detail to determine a right table. If we had to do it all over again our project, we would start the SQL statement because it needs a great effort to establish the SQL statement. Conclusion, we realized that we need to practice SQL coding because SQL statements need a lot of practices and practice makes perfect.



**--Code to build and fill necessary tables from the ER Diagram while giving necessary Primary keys and Foreign Keys to each table.**

**--ONLY NEED THE DROP TABLE STATEMENTS IN THE TEST PHASE**

DROP TABLE intention;

DROP TABLE activityInterests;

DROP TABLE entertainInterests;

DROP TABLE foodInterests;

DROP TABLE sportsInterests;

DROP TABLE demographic;

DROP TABLE my\_users;

CREATE TABLE my\_users

(

userID NUMBER NOT NULL CONSTRAINT my\_users\_pk PRIMARY KEY,

userName VARCHAR2(25) NOT NULL ,

password VARCHAR2(25) NOT NULL,

firstName1 VARCHAR2(25) NOT NULL,

firstName2 VARCHAR2(25) NOT NULL

);

INSERT INTO my\_users (userID, userName, password, firstName1, firstName2)

VALUES ('1', 'BrynJen', 'CoolCars4', 'Bryan', 'Jennifer');

INSERT INTO my\_users (userID, userName, password, firstName1, firstName2)

VALUES ('2', 'BritRick', 'DogLover2', 'Brittney', 'Richard');

INSERT INTO my\_users (userID, userName, password, firstName1, firstName2)

VALUES ('3', 'Mikeytana', 'LoveydoveY', 'Mike', 'Tana');

INSERT INTO my\_users (userID, userName, password, firstName1, firstName2)

VALUES ('4', 'JamieJames', 'JSquared45', 'Jamie', 'James');

INSERT INTO my\_users (userID, userName, password, firstName1, firstName2)

VALUES ('5', 'LizChad', 'craZyLove333', 'Liz', 'Chad');

CREATE TABLE demographic (

demographicID NUMBER NOT NULL,

ageRange INTEGER,

city VARCHAR2(26),

USstate CHAR(2),

relationshipType CHAR(1),

relationshipLength VARCHAR2(10),

userID NUMBER NOT NULL,

CONSTRAINT demographic\_pk PRIMARY KEY (demographicID),

CONSTRAINT demographic\_users\_fk FOREIGN KEY(userID) REFERENCES my\_users(userID)

);

INSERT INTO demographic (demographicID, ageRange, city, USState, relationshipType, relationshipLength, userID)

VALUES (1, 27, 'Saint Paul', 'MN', 'D', '2 years', 1);

INSERT INTO demographic (demographicID, ageRange, city, USState, relationshipType, relationshipLength, userID)

VALUES (2, 28, 'Woodbury', 'MN', 'D', '4 years', 2);

INSERT INTO demographic (demographicID, ageRange, city, USState, relationshipType, relationshipLength, userID)

VALUES (3, 29, 'Eagan', 'MN', 'D', '8 months', 3);

INSERT INTO demographic (demographicID, ageRange, city, USState, relationshipType, relationshipLength, userID)

VALUES (4, 33, 'Cottage Grove', 'MN', 'M', '3 years', 4);

INSERT INTO demographic (demographicID, ageRange, city, USState, relationshipType, relationshipLength, userID)

VALUES (5, 33, 'Minneapolis', 'MN', 'M', '6 years', 5);

CREATE TABLE sportsInterests

(

interestSport\_ID NUMBER NOT NULL,

americanFootball INTEGER,

soccer INTEGER,

basketball INTEGER,

baseball INTEGER,

tennis INTEGER,

volleyball INTEGER,

combatSports INTEGER,

vehicleSports INTEGER,

hockey INTEGER,

userID NUMBER NOT NULL,

CONSTRAINT sportsInterests\_pk PRIMARY KEY (interestSport\_ID),

CONSTRAINT sportsInterests\_my\_users\_fk FOREIGN KEY (userID) REFERENCES my\_users (userID)

);

INSERT INTO sportsInterests

VALUES (1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1);

INSERT INTO sportsInterests

VALUES (2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 2);

INSERT INTO sportsInterests

VALUES (3, 1, 0, 1, 0, 1, 1, 1, 1, 0, 3);

INSERT INTO sportsInterests

VALUES (4, 1, 0, 1, 0, 0, 1, 0, 1, 0, 4);

INSERT INTO sportsInterests

VALUES (5, 1, 0, 1, 1, 1, 1, 0, 1, 1, 5);

CREATE TABLE foodInterests

(

interestFood\_ID NUMBER NOT NULL,

italian INTEGER,

hispanic INTEGER,

american INTEGER,

asian INTEGER,

african INTEGER,

seafood INTEGER,

indian INTEGER,

userID NUMBER NOT NULL,

CONSTRAINT foodInterests\_pk PRIMARY KEY (interestFood\_ID),

CONSTRAINT foodInterests\_my\_users\_fk FOREIGN KEY (userID) REFERENCES my\_users (userID)

);

INSERT INTO foodInterests

VALUES (1, 1, 0, 1, 0, 0, 1, 0, 1);

INSERT INTO foodInterests

VALUES (2, 1, 0, 1, 0, 1, 1, 1, 2);

INSERT INTO foodInterests

VALUES (3, 1, 0, 1, 0, 0, 1, 1, 3);

INSERT INTO foodInterests

VALUES (4, 1, 0, 1, 1, 0, 0, 1, 4);

INSERT INTO foodInterests

VALUES (5, 1, 0, 0, 1, 0, 1, 1, 5);

CREATE TABLE entertainInterests

(

interestEntertain\_ID NUMBER NOT NULL ,

musicals INTEGER,

theatre INTEGER,

orchestra INTEGER,

concerts INTEGER,

movies INTEGER,

boardGames INTEGER,

videoGames INTEGER,

cards INTEGER,

karaoke INTEGER,

userID NUMBER NOT NULL,

CONSTRAINT entertainInterests\_pk PRIMARY KEY (interestEntertain\_ID),

CONSTRAINT entertainInterests\_my\_users\_fk FOREIGN KEY (userID) REFERENCES my\_users (userID)

);

INSERT INTO entertainInterests

VALUES (1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1);

INSERT INTO entertainInterests

VALUES (2, 1, 0, 1, 0, 0, 1, 0, 0, 0, 2);

INSERT INTO entertainInterests

VALUES (3, 1, 0, 1, 0, 0, 1, 0, 0, 0, 3);

INSERT INTO entertainInterests

VALUES (4, 1, 0, 1, 0, 0, 1, 0, 0, 0, 4);

INSERT INTO entertainInterests

VALUES (5, 1, 0, 1, 0, 0, 1, 0, 0, 0, 5);

CREATE TABLE activityInterests

(

interestActivity\_ID NUMBER NOT NULL ,

winterSports INTEGER,

hunting INTEGER,

hiking INTEGER,

exercise INTEGER,

travel INTEGER,

brewery INTEGER,

winery INTEGER,

nightlife INTEGER,

userID NUMBER NOT NULL ,

CONSTRAINT activityInterests\_pk PRIMARY KEY (interestActivity\_ID),

CONSTRAINT activityInterests\_my\_users\_fk FOREIGN KEY (userID) REFERENCES my\_users (userID)

);

INSERT INTO activityInterests

VALUES (1, 0, 0, 1, 0, 1, 1, 0, 1, 1);

INSERT INTO activityInterests

VALUES (2, 0, 1, 1, 0, 1, 1, 0, 1, 2);

INSERT INTO activityInterests

VALUES (3, 0, 1, 1, 1, 1, 0, 1, 0, 3);

INSERT INTO activityInterests

VALUES (4, 0, 1, 1, 0, 1, 0, 1, 0, 4);

INSERT INTO activityInterests

VALUES (5, 1, 0, 1, 0, 1, 0, 0, 0, 5);

CREATE TABLE intention

(

intention\_ID NUMBER NOT NULL ,

alike INTEGER,

different INTEGER,

mentor INTEGER,

beMentored INTEGER,

justMoved INTEGER,

friends INTEGER,

userID NUMBER NOT NULL ,

CONSTRAINT intention\_pk PRIMARY KEY (intention\_ID),

CONSTRAINT intention\_my\_users\_fk FOREIGN KEY (userID) REFERENCES my\_users (userID)

);

INSERT INTO intention

VALUES (1, 1, 0, 1, 0, 1, 0, 1);

INSERT INTO intention

VALUES (2, 1, 0, 1, 0, 1, 0, 2);

INSERT INTO intention

VALUES (3, 1, 0, 1, 0, 1, 0, 3);

INSERT INTO intention

VALUES (4, 1, 0, 1, 0, 1, 0, 4);

INSERT INTO intention

VALUES (5, 1, 0, 1, 0, 1, 0, 5);

**--Business Questions and their associated queries to answer.**

**Question 1. Which users liked Tennis, and in descending order did like soccer.**

SELECT

my\_users.userid,

sportsinterests.tennis,

my\_users.username,

sportsinterests.soccer

FROM

my\_users,

sportsinterests

WHERE

my\_users.userid = sportsinterests.userid

AND

sportsinterests.tennis = 1

ORDER BY sportsinterests.tennis DESC

**Question 3. Which users like soccer**

SELECT

my\_users.username,

my\_users.userid,

sportsinterests.soccer

FROM

sportsinterests

INNER JOIN my\_users ON my\_users.userid = sportsinterests.userid

WHERE

sportsinterests.soccer = 1

**Question 5.What user’s like breweries and American football?**

SELECT

my\_users.username,

my\_users.userid,

activityinterests.hiking,

activityinterests.brewery,

sportsinterests.americanfootball,

sportsinterests.soccer

FROM

sportsinterests

INNER JOIN my\_users ON my\_users.userid = sportsinterests.userid

INNER JOIN activityinterests ON my\_users.userid = activityinterests.userid

WHERE

activityinterests.brewery = 1

AND

sportsinterests.americanfootball = 1

**Question 6. What users like asian food, Sorting the user’s that like it before the one’s that don’t.**

SELECT

foodinterests.asian,

my\_users.userid AS userid1,

my\_users.username,

my\_users.firstname1,

my\_users.firstname2

FROM

foodinterests

INNER JOIN my\_users ON my\_users.userid = foodinterests.userid

ORDER BY foodinterests.asian DESC

**Question 8. Query to inner join to view user’s American Football responses (0 = don’t like, 1= like) using Username and userID as an identifier.**

SELECT my\_users.USERID, my\_users.USERNAME, interestSport\_ID, sportsInterests.AMERICANFOOTBALL FROM

sportsInterests INNER JOIN my\_users

ON (my\_users.userID = sportsInterests.userID);

**Question 9. Query to inner join to view user’s Baseball responses (0 = don’t like, 1= like) using Username and userID as an identifier.**

SELECT my\_users.USERID, my\_users.USERNAME, sportsInterests.BASEBALL FROM

sportsInterests INNER JOIN my\_users

ON (my\_users.userID = sportsInterests.userID);

**Question 10. What sports interests does UserID(sportsInterestID) 1 like?**

SELECT \*

FROM SPORTSINTERESTS

WHERE INTERESTSPORT\_ID = 1;

**Question 11. What users are the travelers, Travelers at the top of the list**

SELECT

my\_users.username,

activityinterests.travel,

my\_users.userid AS userid1,

activityinterests.interestactivity\_id,

activityinterests.userid

FROM

my\_users

INNER JOIN activityinterests ON my\_users.userid = activityinterests.userid

ORDER BY activityinterests.travel DESC

**Question 12. What’s user one name and did they like Hockey?**

SELECT my\_users.userid,

sportsinterests.hockey

FROM my\_users,

sportsinterests

WHERE my\_users.userid = sportsinterests.userid

AND my\_users.userid = 1

**Question 13. What are the usernames for the activity users?**

SELECT

activityinterests.interestactivity\_id,

my\_users.username,

activityinterests.wintersports,

activityinterests.hunting,

activityinterests.hiking,

activityinterests.travel,

activityinterests.exercise,

activityinterests.brewery,

activityinterests.winery,

activityinterests.nightlife,

activityinterests.userid AS userid1

FROM

my\_users

INNER JOIN activityinterests ON my\_users.userid = activityinterests.userid

**Question 14. What is the average of the Sports activities for User1?**

SELECT

my\_users.userid

,my\_users.username

,sportsinterests.interestsport\_id

,((sportsinterests.americanfootball +

sportsinterests.soccer +

sportsinterests.basketball +

sportsinterests.baseball +

sportsinterests.tennis +

sportsinterests.volleyball +

sportsinterests.combatsports +

sportsinterests.vehiclesports +

sportsinterests.hockey)/9) AS SPORT\_AVG

FROM

my\_users,

sportsinterests

WHERE

((my\_users.userid = 1) AND (sportsinterests.userid = 1))

**Result**

**1 BrynJen 1 0.3333333333333333333333333333333333333333**

**Question 15. What is the average of the Sports activities for User1?**

**(trying to build as a reusable function(NOT QUITE WORKING))**

CREATE VIEW USERSPORT AS

(

activityinterests.wintersports AS S1,

activityinterests.hunting AS S2,

activityinterests.hiking AS S3,

activityinterests.travel AS S4,

activityinterests.brewery AS S5,

activityinterests.exercise AS S6,

activityinterests.winery AS S7,

activityinterests.nightlife AS S8,

activityinterests.userid AS userid1

FROM Dual

);

SELECT \* FROM USERSPORT

CREATE VIEW USER\_AVG\_SPORT AS

SELECT (S1+S2+S3+S4+S5+S6+S7+S8)/8 AS AVG\_SPORT, userid1 AS SPORT\_USERS

FROM USERSPORT

SELECT \*

FROM USER\_AVG\_SPORT

WHERE SPORT\_USERS = 'userid1'

**Question 15. This is broken partially, an outer join that picks people that don’t have responses to indian food and vehiclesports and an ID less than 4.**

SELECT

\*

FROM

interestFood\_ID AS foID

LEFT OUTER JOIN interestsport\_ID as spID

ON foID.indian = spID.vehiclesports

WHERE

my\_users.userid = <4