

|  |
| --- |
|  |
| DATABASE FOR FACEBOOK  2018 |
| BUS243 – DATABASE MANAGEMENT  Submitted by:  Megha R Rao (SJSU ID: 013709488)  Rajasree Rajendran (SJSU ID: 013774358)  Sai Chaitanya Tolem (SJSU ID: 013008788) |



# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc532226079)

[Description of the Data Model 4](#_Toc532226080)

[Query description & Analysis 7](#_Toc532226081)

[CONCLUSION 20](#_Toc532226082)

[REFERENCES 23](#_Toc532226083)

Overview

|  |
| --- |
| Facebook Inc. is an American social media and social networking company which was established in the year of 2004 by Mark Zuckerberg, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz and Chris Hughes. It was a school-based social network in Harvard University until 2006. In 2006, Facebook opened its doors to anyone older 13 years or older in the world. Facebook has a very user-friendly interface, and anyone with basic computer knowledge can use Facebook. The primary purpose of Facebook was to find friends who have not been in touch and to help them re-connect. Among the many other social networking sites, Facebook emerged to be the most successful one due to its unique features such as the ‘Like’ option, News Feed, Games and Business-friendly approach. The usage of Facebook has grown over time, the number of users crossing 2.27 billion monthly active users, as of September 2018, according to statistics. In the last decade, the use of all social networking sites has grown exponentially, with Facebook leading the list. This exponential growth means there is a huge amount of data available from all these users. Facebook was built around Big Data from its beginning, data was the driving force that kept it alive. But recently, Facebook has run into a lot of trouble for its usage of user data. Some attackers accessed personal data of at least 50 million Facebook users by exploiting a vulnerability in the system. This led to a huge uproar about the data usage of social networks, and increased privacy concerns among users, which led Facebook to shut down almost all of its open source data. In this project, we have attempted to create a database that is similar to that of Facebook, write queries to see how the database works and find some specific details using SQL queries such the most expensive game, a specific name search, etc. Description of the Data Model In order to proceed with this project, it is essential to understand how Facebook database works. Information on Facebook is mostly represented in the form of a social graph. The content is usually highly customizable based on the user’s privacy settings. Hence, the data has to be stored in its original form and then filtered when needed. Facebook uses a combination of MySQL and Memcache for its database. Every user has his/her own dedicated database. Facebook uses MySQL because of its speed and reliability. Facebook stores friend relationships in a system called ‘Tao’ which uses MySQL. All tables have a hashed name and they are spread over a number of servers, similar to graph databases. Tao only stores the relations between entities. According to Facebook Inc., they collect information based on how a user uses their products. Information is collected from and about all computers and other devices the user uses to access Facebook, and this information is combined by them. The collected information is used to personalize features and content and to make suggestions for the user. The collected data is used to help advertisers to measure the effectiveness of their ads and services and to understand how the users interact with their services. They store the data until it is no longer necessary to provide the services or until the user deletes the account, whichever is first. While trying to understand the database structure of Facebook, we went through a number of resources and stumbled upon a resource where the class diagram was created by reverse engineering various Facebook business entities. Since only a very small portion of database details of Facebook is available as open source, we decided to utilize the available reverse-engineered diagram in order to create our database, by reverse-engineering the ER diagram and populating our own data. |
| *“Without a systematic way to start and keep data clean, bad data will happen”- Donato Diorio* |
| *ER Diagram for Facebook Database*  Creating the Entity Relationship (ER) diagram was the most fun part while doing this project. We initially thought this would be an easier job, but the unique issue about dealing with Facebook data was that, it was all over the internet, but nowhere specific for open source use. After referring to many resources, we realized that News Feed does not have a relation to the other entities. The above ER diagram represents the Facebook profile database of a single user as an entity.  The ER diagram has the following entities with their own attributes:   * *User\_basic*: Has various basic attributes of the user namely *User\_id* (Primary Key), *First\_name,* *Last\_name, Password, Email, Mobile number, Birth\_date* and *Gender.* * *User\_about:* with attributes *Occupation, Education, Lives\_in, Address\_id* and *Date\_of\_joining.* * *Address:* Attributes are *Address\_id, Street, City, State, Country, Zip code* and *Region*. * *Pages:* Attributes are *Page\_id, Category, User\_id, No\_of\_followers.* * *MarketPlace:* Attributes are *item\_id, category, quantity, location, price, seller\_id* and *item\_description*. * *Events:* Attributes are *event\_id, interested\_userid, Going\_userid, Date\_time, Venue.* * *Photos:* Attributes are *photo\_id, user\_id, caption, tags, location, upload\_date\_time*. * *Activity\_log:* with attributes *User\_id, friend\_id, action\_performed, activity\_date*. * *Games:* attributes are *game\_id, user\_id, category, captions, score, paid.* * *Posts:* with attributes *post\_id, user\_id, action\_performed, privacy\_level.* * *Friends:* with attributes *friend\_id, user\_id, friends\_user\_id, category, start\_date, since\_when.* * *Advertisements: ad\_id, page\_id, no\_of\_clicks, traffic, location, audience, budget, schedule.* * *DA Toolkit: tool\_id, tool\_name, trial\_weeks, cost, description.*  Query description & Analysis Based on the above data, we decided to find open source Facebook user data in order to do the queries in mySQL. Unfortunately, it was very hard to find open datasets for Facebook, due to all the recent data breach incidents Facebook Inc. has gone through. After a great amount of research, we decided to create a Facebook database on our own with some fictitious data. Thus, data preparation was done. Since the fictitious data was in Google Sheets format, we converted it into csv format and then later, converted the data to sql format and uploaded to mySQL workbench. The following are the queries we performed based on our prepared data:   * Our first goal was to create tables and entities to accommodate the data. After obtaining the sql data, we used CREATE function to create tables. One such was to create a table called ‘Sales\_table’ which shows all the items ordered by price and quantity from the ‘Marketplace’ table. The following query was used to create the table:   CREATE TABLE Sales\_Table AS (SELECT Item\_id, Category, Price, Quantity FROM marketplace);    This statement creates a table called ‘Sales\_Table’ with entities from ‘Marketplace’.   * After creating the Sales Table, we went ahead and calculated which items were being sold the most, with the following statement:   SELECT \* FROM Sales\_Table  WHERE Quantity IN (SELECT MAX(Quantity) FROM Sales\_Table)  ORDER BY Price DESC;  Output:     * We also calculated the total number of items currently in marketplace along with the numbers in each category.   SELECT category, quantity, count(category) FROM marketplace GROUP BY category WITH rollup;  Output:     * Since we wanted to know the total cost distributed for each category, we did the following query and sorted the results in 5 quantile ranges of price.   SELECT item\_id, category, quantity\*price AS cost, NTILE (5) OVER (ORDER BY quantity\*price) AS quantile FROM marketplace GROUP BY category;  Output:     * As a part of performing routine activities, we calculated the percentage of female users and male users in our dataset.   SELECT ROUND(((SELECT COUNT(\*) FROM user\_basic WHERE gender = 'Male') /  (SELECT COUNT(\*) FROM user\_basic))\*100,2) AS "Percentage of Male users",ROUND(((SELECT COUNT(\*) FROM user\_basic WHERE gender = 'Female')  /(SELECT COUNT(\*) FROM user\_basic))\*100,2) AS "Percentage of Female users" FROM DUAL;  Output:     * Next was the calculation of Age of users as a derived attribute.   SELECT \*, YEAR(CURDATE()) - YEAR(birth\_date) AS Age, NTILE(4) OVER (ORDER BY 'Age' ASC) AS "Quantile (Age groups - 1 to 4)" FROM user\_basic;  Output:     * In order to execute a realistic scenario, we decided to find the users who are from California, and then found their education details.   SELECT concat (First\_name," ", Last\_name) as Name, education\_level, lives\_in, occupation FROM user\_basic JOIN user\_about ON user\_basic.user\_id = user\_about.user\_id WHERE lives\_in = 'California';  Output:     * As a part of the routine activities, we decided to find out a specific user’s friends. We chose the user “Harry Potter” and tried to find out the friends of Mr. Potter. Even though we were expecting Ron Weasley and Hermione Granger to be in the list, we were quite surprised by the results.   SELECT user\_basic.user\_id, friends.start\_date, CONCAT (user\_basic.first\_name," ",user\_basic.last\_name) AS Friend, (CURDATE() - friends.start\_date) AS FriendshipinDays FROM friends JOIN user\_basic ON user\_basic.user\_id = friends.friends\_user\_id WHERE friends.user\_id = (SELECT user\_id FROM user\_basic WHERE user\_basic.first\_name = 'Harry');  Output:     * In order to find the details of an event, we created a query with respect to event entity. The following query finds the count of all events listed: * SELECT Venue AS "Event Venue", Date\_Time, COUNT(Invitees\_Userid) AS "No. of Invites" FROM Events GROUP BY Venue, Date\_Time;   Output:     * We found the names of the invitees to the events:   SELECT CONCAT(first\_name," ",last\_name) AS Name, Venue, Date\_Time FROM user\_basic JOIN events ON events.invitees\_userid = user\_basic.user\_id ORDER BY Venue, Date\_Time;  Output:     * Among its millions of users, there are a lot of inactive users who could be considered as shadow profiles/ inactive users. We executed a query to find out who were the users who had not logged in the past six months, thus classifying them as inactive users. We deleted such users without compromising the referential integrity.   SET SQL\_SAFE\_UPDATES = 0;  DELETE FROM user\_basic WHERE user\_id IN  (SELECT u.user\_id FROM User\_about u JOIN activity\_log a ON a.User\_id = u.User\_id  WHERE (a.Activity\_date < (NOW() - INTERVAL 6 MONTH)) GROUP BY u.user\_id);  SET SQL\_SAFE\_UPDATES = 1;  DELETE FROM user\_basic WHERE user\_id = 100528;  Output:           * We imagined a scenario where the FBI asks Facebook team to help them out by finding the users whose names end with “er”, which could help them in an ongoing investigation. We decided to help them out by doing this query:   SELECT \* FROM user\_basic WHERE Last\_name LIKE '%er';  Output:     * The FBI seemed to have obtained an anonymous tip on the person, the tip was that the name ends with “otter”, which luckily narrows down the suspects.   SELECT \*from user\_basic WHERE Last\_name LIKE '%otter%';  Output:     * We decided to do a descriptive analysis of budget allocation of Advertisements by finding total, minimum, maximum and average of budgets.   SELECT COUNT(budget) AS 'Total No.’, SUM(budget) AS 'Total Budget’, AVG(budget) AS 'Average Budget’, MIN(budget) AS 'Minimum’, MAX(budget) AS 'Maximum Bdget’ FROM advertisements;  Output:     * After finding out the details about the budget, we explored various mathematical functins such as midrange, standard deviation, harmonic average, arithmetic mean and geometric mean.   SELECT ((MAX(budget) + MIN(budget))/2) AS 'Midrange’, STDDEV(budget) AS 'Standard Deviation', VARIANCE(budget) AS 'Variance’, SUM(budget) / COUNT(budget) AS 'Arithmetic Mean’, COUNT(budget) / SUM(1/budget) AS 'Harmonic Average’, EXP(SUM(LOG(budget))) AS 'Geometric Mean’ FROM advertisements;  Output:     * Next, as a part of our objective, we did recommendations of least expensive games from the Games data.   SELECT name AS ' Are you ready? - Exciting games!', category 'Category', paid as 'Do I pay?', price AS 'Recommended - Low to High Price!' FROM games ORDER BY price ASC;  Output:     * As per our objectives of the project, we planned to create views, and we created views for Ad plans.   CREATE VIEW Ads\_Premium ASSELECT Tool\_name AS "Ad Tools you get!", Description AS "Details", Cost AS "Price - only from",Trial\_weeks AS "Trial weeks" FROM ad\_metrics ORDER BY Cost;  CREATE VIEW Ads\_Booster ASSELECT Tool\_name AS "Ad Tools you get!", Description AS "Details", Cost AS "Price - only from",Trial\_weeks AS "Trial weeks" FROM ad\_metrics WHERE Ad\_tool\_group = 1 OR Ad\_tool\_group = 2 ORDER BY Cost;  CREATE VIEW Ads\_Basic ASSELECT Tool\_name AS "Ad Tools you get!", Description AS "Details", Cost AS "Price - only from",Trial\_weeks AS "Trial weeks" FROM ad\_metrics WHERE Ad\_tool\_group = 1 ORDER BY Cost;  Output:     * One of our goals was to predict user interests. For this we used the following query by joining three tables, thus doing the prediction of user interests.   SELECT a.user\_id AS "User", p.post\_id AS "Recommended post",  a.friend\_id AS "Related to friend", a.action\_performed AS "Friend's action"  FROM activity\_log a INNER JOIN friends f ON a.user\_id = f.friends\_user\_id  INNER JOIN posts p ON f.friends\_user\_id = p.user\_id  WHERE p.privacy\_level <> 'Only Me';  Output:     * After creating the views, we could view the Views with Data Analysis toolkits for each Ad plan.   SELECT \* FROM Ads\_Basic;  SELECT \* FROM Ads\_Booster;  SELECT \* FROM Ads\_Premium;  Output:         CONCLUSION As a part of the project, we were able to manually create a database, run queries on the database and find results using various queries we learnt in class. We were able to fulfill most of our project objectives and learned a lot in the process. The objectives were to create tables and entities, and perform routine Facebook activities. The database we built satisfies all the required entities for performing a variety of queries based on user preference. In an organization like Facebook, where database and extraction of data from database plays a major role, it is imperative to work with tools like MySQL and help in carrying out required operations. Using the queries like the ones used in this project, it helps to narrow down data from 2.2 billion monthly users. Queries which use functions such as CREATE, JOIN, SELECT, UPDATE, DELETE are all most common queries used in a scenario like that of Facebook’s. During this project, we faced a number of challenges : one of them being the unavailability of open datasets for Facebook users. We rectified the same by recreating databases on our own. The next challenge we faced was converting the csv files to sql format, many online tools were tried in vain. Thanks to Professor Shirani for guiding us during the challenging times, as per Professor Shirani’s instructions, we were able to successfully convert the csv files to sql files using SQLite Studio software. We believe we were able to fulfill most of our objectives and we did learn a lot during the process.     |  | | --- | | **TABLEAU DASHBOARD :** |   MongoDB Analysis:  Of the games which are free under the simulation category - Most people prefer to play 8 Ball Pool (75%) – using the Analyze Schemas feature in MongoDB.   REFERENCES  * <http://web.archive.org/web/20121031052327/http://blogs.x2line.com/al/archive/2007/06/02/3124.aspx> * <https://www.facebook.com/full_data_use_policy> * <https://www.usenix.org/conference/atc13/technical-sessions/presentation/bronson> * <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/> * <https://www.facebook.com/notes/facebook-engineering/tao-the-power-of-the-graph/10151525983993920/> * <https://www.makeuseof.com/tag/facebook-work-nuts-bolts-technology-explained/> |