Journal of Physics: Conference Series

Data Structure Comparison Between MySql Relational Database and Firebase Database NoSql on Mobile Based Tourist Tracking Application

Data Structure Comparison Between MySql Relational Database and Firebase Database NoSql on Mobile Based Tourist Tracking Application

I K G Sudiartha[1], I N E Indrayana[2], I W Suasnawa[3], S A Asri[4], Putu Wijaya Sunu[5]

[1234]Electrical Engineering Department, Politeknik Negeri Bali, Bali, Indonesia
[5]Mechanical Engineering Department, Politeknik Negeri Bali, Bali, Indonesia
[1]email : itutde@pnb.ac.id

Abstract. Tracking application is widely used to help outdoor activities. One of them is used to help tourists who do tourist destinations in group activities. Tourist tracking application requires data on the position of tourists on earth. This coordinate data can be obtained using latitude and longitude data from GPS (Global Positioning System). Coordinate latitude and longitude position of tourists from time to time are stored in a database. Database Selection for media storage will determines the level of speed and ease in accessing data. There are some differences in designing the structure of the MySql and Firebase databases. Relational Database begins with identification of entities involved in the system. One entity with another entity is related and has a cardinality level of the relationship. Relational database implementation in the form of tables in normal forms that are interconnected. Data will be stored in columns and rows. Normalization of tables causes high query computation when calling data that involves many tables. Firebase, which is one of the NoSql databases, in building data structures is stored in JSON objects. JSON tree structures usually consider the display of information that will be presented to users. So that the generated JSON tree has high efficiency and speed in processing data. In this study a comparison between the design of the relational database structure, namely MySql and NoSql Firebase database structure was developed. Firebase is more power full for the development of applications which update data in realtime and and handle a lot of coordinate data such as tourist tracking applications. Applications that do not have complex data relations such as tourist tracking applications are more suitable to use firebase than using MySQL.

1. Introduction

Today mobile-based applications are very mostly used. One of the uses of mobile applications in the tourism sector is tracking tourists visiting groups. This group of tourists is usually guided by a tour guide. To build a tourist tracking application, coordinate data is needed from the position of the traveler on earth. To store data from the position of tourists requires storage media, namely the database. Databases are of various kinds. In this study we will discuss creating a database data structure for 2 types of databases, namely relational databases and firebase databases which are NoSql Databases. Performance of each database needs to be considered considering that the tourist tracking application is a cloud application that requires high accessibility.

Currently developing Firebase database which is one of the NoSql databases. Firebase was developed by Google. Firebase is one of the real time databases that is suitable for applications that require fast data updates. Google developed a firebase database to facilitate programs in developing applications.

Programmers in coding do not need to think about server-side coding because it has been handled by the firebase itself. Programmers in developing client-server applications seem to be coding like a stand alone program (client side) because the server side has been handled by its own firebase.
NoSQL database is the best solution for application using cloud databases. Because of the characteristics that determine the NoSQL database are very desirable for cloud databases. NoSQL databases support increased availability, scalability, performance, and flexibility, also related to unstructured, semi-structured, or structured data [1][2]. Data types from relational databases are determined at the beginning and very strong table structure while NoSql databases have the ability to easily and quickly change data structures [3].
In this research, several step were carried out, for the first step, conducting all of a literature study. Literature study that contains a relational database and NoSQL Database theory. The second step, analyzing the data requirements. The third step, conceptual design. At the conceptual design step, the ERD design is used for relational databases. The approach to related objects is carried out on the NoSql Database. The fourth step is mapping from conceptual design to physical design the database. The physical scheme describes the implementation of data structures in MySql Relational Database and Firebase Database NoSQL. The fifth step is comparison the data structure between MySql and Firebase. Comparisons are made to the physical structure of each database by taking into account the relations produced, and related to the results of the literature studies that have been carried out in the previous step. Finally, make conclusions on research that has been done.

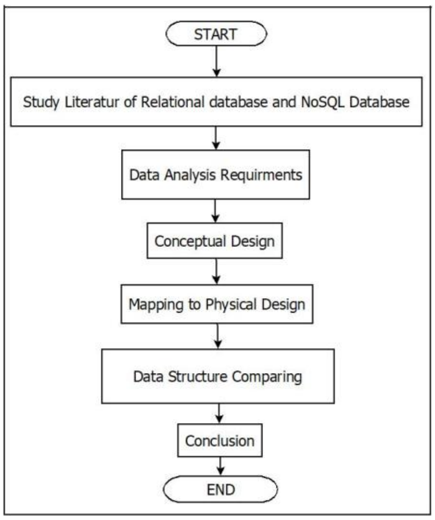


Figure 1. Research Methodology

1.1. MySql Relational Database
Relational Database Structure such as MySql, can be formed by determining the entities involved in the system and relations that connect between one entity and another entity. Entity is a matter that can be clearly identified [15]. So an entity can be a person, place, object, event or concept that we want to store data. Entities are usually nouns [16]. Each entity has attributes that explain the entity. Can be numbered attributes, a character string, a date, an image, a sound, and even more [16]. Then after we can stop the entities involved, we proceed to build the Entity Relationship Diagram (ERD). In ERD there are 3 basic relations between entities. These relationships are one to one, one to many and many to many relationships. Transformation is needed to form a table that will be implemented in the database. One to one relationship, resulting in a key from one entity attracted to another entity which then becomes a foiregn key. Because of its one to one nature, the key can be drawn to one of the entities. As a consideration, drag the key to the entity that we predict has fewer row records. One to many relations, resulting in keys being drawn from entities that have lower cardinality to entities that have higher cardinality. For many to many relations, resulting in the formation of a new table from the existing relationship. The table will draw both keys from related entities. The end goal is a normal

table. Tables that have not been normal can be done further table normalization process [17]. Data in Relational Databases are stored in tables as lines and columns. Columns represent fields and rows of data referred to as records. Data types from fields must be defined definitively at the beginning [18]. Is typenya varchar, int, date, logint, etc. Techniques for manipulating data in tables using SQL queries [18].

1.2. NoSql Firebase Database

The Firebase Database is NoSql cloud-based database that syncronous data across all clients in realtime[19] .Data from the Firebase realtime database is stored as JSON and synchronized in realtime to each connected client. This data is stored cloud hosted. When developers create cross-platform applications with Android, iOS, and JavaScript SDK [20], all clients will share a Realtime Database instance and receive the latest data updates automatically. Fire base is not like an SQL database, firebase databases do not have tables or records. When you add data to a JSON tree, the data will become a node in the existing JSON structure with the associated key. You can enter your own key, such as a user ID or semantic name. The key can also be provided for you to use push (). The Firebase Realtime Database uses data synchronization every time the data changes, all connected devices will receive updates without the need to think about network codes. When offline users can still save data into the application, and will update data after the network is connected to the server. The server updates the data to each device that is connected automatically. So it's responsive even when it's offline [21][22]. The Firebase Realtime Database can be accessed directly from a mobile device or web browser, so it doesn't require an application server. In certain packages that are promising, Google as the owner of Firebase provides flexibility for developers to share storage into several databases according to their needs

The data structure of Firebase must pay attention to the data display on the client side. Designing an efficient structure will speed up the data access process. Firebase allows developers to make data structures reach 32 levels [23]. When an application retrieves data at a location in the database, firebase will retrieve all derivative nodes. In addition, when we provide read and write access to someone in a data node, we will provide data access to all data. Examples of firebase data structures for chat applications can be seen in figure 1.

```
{
  // Chats contains only meta info about each conversation
  // stored under the chats's unique ID
  "chats": {
    "one": {
      "title": "Historical Tech Pioneers",
      "lastMessage": "ghopper: Relay malfunction found. Cause: moth.",
      "timestamp": 1459361875666
    },
    "two": { ... },
    "three": { ... }
  },
```

Figure 2. The JSON tree structure of the chat application has been flattened[24]

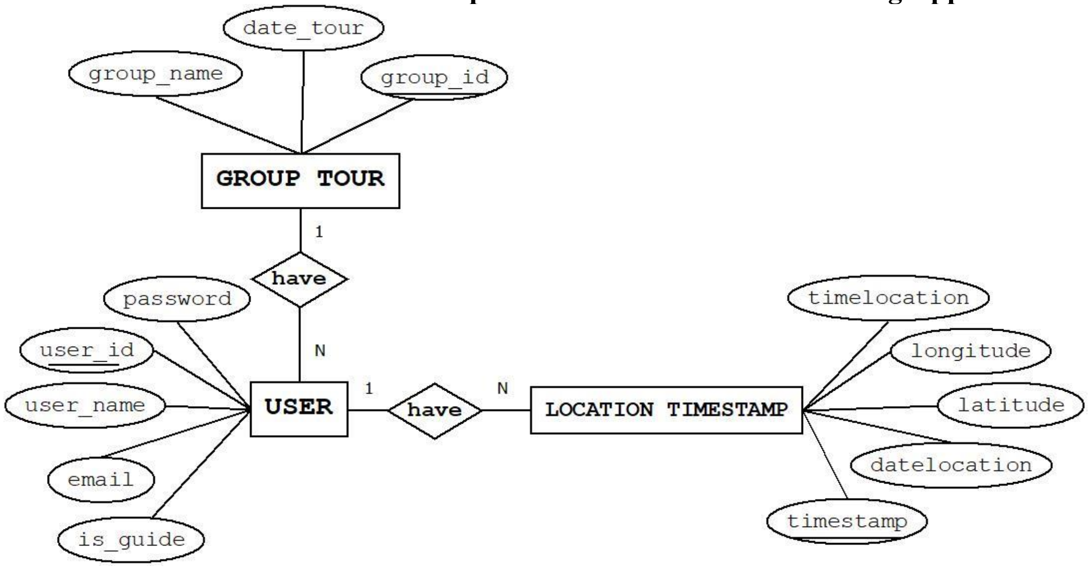
2. Data and Information Application Requirement

There are several entities involved in tourist tracking applications, namely User, Group Tour and TimeStamp Location. The User Entity can be a tourist as a Guide. Group Tour has group id and group name attributes. The User Entity has a user ID, user name, password, e-mail and the is guide attribute. If the user is a tourist, the is guide attribute is filled with false, whereas if the user is a guide, the guide is filled with true. The Location Entity records all latitude and longitude data from the route carried out by the user. Location Entity has location id attributes, latitude, longitude, date and time.

In this tourist tracking application there are 2 relations. The first relation is the relation between the User and the Location Timestamp. One User can occupy several Location TimeStamp positions at different times, while Location TimeStamp also has only one User. The second relation is the relationship between the User entity and the Group Tour entity. One Group Tour can have several member users. While one User may only be a member of the Group Tour at a time.

This tourist tracking application is expected to display information as follows:
1. Display the Group Tour Name list (point 1).
2. Displays a list of User names for one Group Tour (point 2).
3. Showing Position of one User for one Group Tour with visualization on Google Map (point 3).
4. Displays the route of travel from one User to one Group Tour on a certain date with visualization on the Google Map (point 4).
5. Displays all the last positions of Users in a Group Tour with visualization on the Google Map (point 5).

3. ERD Database and Relational Table Implementation for Tourist Tracking Application



Figur 3. Entitas Relationship Diagram Of Tourist Tracking Application

Transformation for relations between User and Location TimeStamp (aggregation) has one to many relationships. The implementation of this relation is a key from an entity that has a lower cardinality drawn to an entity that has a higher cardinality as a foreign key. Transforming Group Tour relations with Users is a one to many relationship so that the key from the Group Tour will be drawn to the User as a foreign key. So the table is formed like this:

Table 1. Table GroupTour

No	Field Name	Field Type	length	Description
1	group_id	int	10	Primary key (auto increment)
2	group_name	varchar	100	
3	date_tour	varchar	50	

Table 2. Table User

No	Field Name	Field Type	length	Description
1	user_id	varchar	20	Primary key
2	user_name	varchar	100	
3	Password	varchar	50	
4	Email	varchar	50	
5	is_guide	boolean		True for user as guide, false for user as tourist
6	group_id	int		Foreign key

Journal of Physics: Conference Series **1569** (2020) 032092 doi:10.1088/1742-6596/1569/3/032092

Table 3. Table LocationTimeStamp

No	Field Name	Field Type	length	Description
1	timestamp	long		Primary key
2	latitude	double		
3	longitude	double		
4	datelocation	date		
5	timelocation	time		
6	user_id	varchar	20	Foreign key

4. Firebase Database Structure for Tourist Tracking Application

Tourist Tracking Application with Firebase begins with building JSON data structures. We also consider the information presented to the users. Because firebase does not use SQL query in the process of reading data, then when entering new data into JSON, we can use the key of that data as a JSON node tree.

Figure 4 is the initial design of the firebase JSON structure from the tourist tracking application. Here we need to consider the JSON structure that has been generated, is it optimal and can already handle the information needed by the user. If our firebase JSON structure is like Figure 4, then to display the Group Tour name list it will call all derivative nodes in it, that is until the "locationTimeStamp" data will be loaded. If not, do the flatten process of leveling the data structure. If the firebase structure more flatten, then the data will more optimal for loading . after we done the flatten process, the structure in figure 4 becomes figure 5. Firebase Data Structure in figure 5 can display information for points 1-3 (see the Data and Information Application Requirement) . To be able to display information on points 4 and 5, we need to change the Firebase structure again to figure 6. In figure 6, "datelocation" becomes a node to be able to display point 4 information, and adds a "lastlocation" node to display point 5 information.


```
{touristtracking:{
    ”group01”:{
        ”datetour”:”12-05-2019”,
        ”groupname”:”Bedugul May 12 2019”,
        ”user”:{
            ”cris01”:{
                ”email”:”cris010190@gmail.com”,
                ”is_guide”:”true”,
                ”locationTimeStamp”: {
                    ”1560047336”: {
                        ”datelocation”: ”12-05-2019”,
                        ”latitude”: -8.6916919,
                        ”longitude”: 115.2355783,
                        ”timelocation”: ” 10:28:56”
                        },
```


Figure 4. First Firebase Data Struktur for Tourist Tracking Application


```
{”touristtracking”:{
    ”group01”:{
        ”datetour”:”12-05-2019”,
        ”groupname”:”Bedugul May 12 2019”,
        }
    }

 ”user”:{
    ”group01”:{
        ”cris01”:{
            ”email”:”cris010190@gmail.com”,
            ”is_guide”:”true”,
            ”password”:”crispass01”,
            ”user_name”:”Cristopher Brain”
            },
        ”danish01”:{
            ”email”:”danish9999@gmail.com”,
            ”is_guide”:”false”,
            ”password”:”danish999”,
            ”user_name”:”Danish Cooper”
            }
        }
    }
```


Figure 5. Firebase Data Structure for Tourist Tracking Application After Flatting Proses

```
{"touristtracking":{
    "group01":{
        "datetour":"12-05-2019",
        "groupname":"Bedugul May 12 2019",
        }
    }

"user":{
    "group01":{
        "cris01":{
            "email":"cris010190@gmail.com",
            "is_guide":"true",
            "password":"crispass01",
            "user_name":"Cristopher Brain"
            },
        "danish01":{
            "email":"danish9999@gmail.com",
            "is_guide":"false",
            "password":"danish999",
            "user_name":"Danish Cooper"
            }
        }
    }
```

Figure 6. Final Structure Firebase Data for Tourist Tracking Application After Flatting Proses

5. Conclusion

From the explanation above, it can be concluded that relational databases such as MySql and Firebase database (NoSql) can be used to build the tourist tracking system application. However, there are a number of conditions that Firebase is very suitable for large amounts of cloud data applications and does not have complex relationships. The tourist tracking application in this paper saves a lot of GPS location data from the user's position from time to time. And requires a stable network connection in order to send data continously. Firebase provides facilities for storing data while offline, which then updates data automatically after a network connection is connected. So that programmers don't need to make special code to handle conditions if the network is lost. Tranformation from ERD to be a table structure in a relational database is easier by following table formation theory. In the beginning Firebase requires analysis and more complicated to build JSON tree structures . JSON tree structure usually consider the display of information that will be presented to the user. So that the JSON tree produced has high efficiency and speed in accessing data.

6. References

[1] Mohamed M A, Altrafi O G, Ismail M O 2014, " Relational vs. NoSQL Databases: A Survey" *"International Journal of Computer and Information Technology "*.(ISSN: 2279 – 0764) Volume 03 – Issue 03

[2] Behjat U Nisa 2018, "A Comparison between Relational Databases and NoSQL Databases" *"International Journal of Trend in Scientific Research and Development (IJTSRD)"*. ISSN: 2456-6470 , Volume :2 Issue : 3

[3] Mitko Radoev 2017 "A Comparison between Characteristics of NoSQL Databases and Traditional Databases" *"Computer Science and Information Technology 5"*. DOI: 10.13189/csit.2017.050501, page 149-153

[4] Sawsan Alshattnawi 2013, "Building Mobile Tourist Guide Applications using Different" *International Journal of Advanced Science and Technology* vol. 54.

[5] Panahi M S, Woods P, Thwaites H 2013, " Designing and Developing a Location-Based Mobile Tourism Application by Using Cloud-Based Platform" *International Conference on Technology, Informatics, Management, Engineering and Environment* Publisher: IEEE DOI: 10.1109/TIME-E.2013.6611982.

[6] Shashikant K S, Baban L N , Bajrang G A, Suresh J M , Bhosale D V 2015, "Android Based Mobile Smart Tracking System" *International Journal of Latest Trends in Engineering and Technology* ISSN: 2278-621X, Vol. 5 Issue 1.

[7] Surender R, Hymavathi M, Shilpa E 2016, " Real Time Vehicle Monitoring and Tracking System based on Embedded Linux Board" *International Journal of Research in Computer Engineering and Electronics* Vol. 4, Issue 10.

[8] Uddin P, Nadim, Zahidul I, Afjal M I 2013, " GPS-based Location Tracking System via Android Device" *International Journal of Research in Computer Engineering and Electronics*. ISSN 2319-376X, Vol 2 Issue: 5.

[9] Godavarthi B, Nalajala P, Ganapuram V 2017, " Design and Implementation of Vehicle Navigation System in Urban Environments using Internet of Things (Iot) " *IOP Conf. Series:*

Materials Science and Engineering 225 (2017) 012262 , IOP Publishing. doi:10.1088/1757-899X/225/1/012262
[10] Dhumal A, Naikoji A, Patwa Y, Shilimkar M, Nighot M K 2015, " Vehicle Tracking System using GPS and Android OS " *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET).* Volume 4 Issue 4
[11] Jumb V, Gupta G, Aloysius E, Kamat S, Wakankar N 2019, " Smart City Traveller" *"International Journal Of Information And Computing Science".*Volume 6, Issue 4
[12] Sudiartha I K G , Indrayana I N E , Suasnawa I W , Ciptayani P I 2018, " Design And Implementation of Group Tourist Monitoring Application With Realtime Database Firebase" *"International Conference on Science and Technology (ICST 2018) ,Atlantis Highlights in Engineering".*ISBN : 978-94-6252-650-1, Vol 1
[13] Phiri H , Kunda D 2017 " A Comparative Study of NoSQL and Relational Database" *"Zambia Information Communication Technology (ICT) Journal".*Volume 1 (Issue 1) Pages 1-4
[14] Almootassem O, Husain S H, Parthipan D, Mahmoud Q H 2017 "A Cloud-based Service for Real-Time Performance Evaluation of NoSQL Databases" *" https: //arxiv.org/abs/ 1705.08317*
[15] Sikha Bagui , Richard Earp 2003 "Database Design Using Entity-Relationship Diagrams" *"Auerbach Publications* ISBN:0849315484
[16] Jan Speelpenning, Patrice Daux, Jeff Gallus, 2001 "Data Modeling and Relational Database Desig Edition 1.2" *Oracle Corporation*
[17] Hugh Darwen 2012 " An Introduction to Relational Database Theory 3rd Edition" *Ventus Publishing ApS* ISBN:978-87-403-0202-8
[18] Khawas C, Shah P 2018 "Application of Firebase in Android App Development-A Study" *International Journal of Computer Applications (0975 – 8887)* Volume 179 – No.46
[19] Laurence Moroney 2017 "The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform 1st ed. Edition" *Apress; 1st ed. Edition* ISBN-13: 978-1484229422
[20] Khedkar S, Thube S 2017 "Real Time Databases for Applications" *International Research Journal of Engineering and Technology (IRJET)* Volume: 04 Issue: 06
[21] Neil Smyth 2017 "Firebase Essentials Android Edition-First Edition" *Payload Media, Inc* ISBN 9781546660330
[22] Ashok Kumar S 2018"Mastering Firebase for Android Development" *Packt Publishing Ltd* ISBN:978-1-78862-471-8
[23] Google Developers 2018 Structure Your Database https://firebase.google.com/docs/database/web/structure-data download : May 17 2018

Acknowledgments
This research was support by the Directorate of Research and Community Service, Director General of Development and Research Enhancement, Ministry of Research, Technology and Higher Education, In accordance with the research contract: Nomor:133/SP2H/LT/DRPM/2019.
SP DIPA-042.06.1.401516/2019 December 5, 2018