CLOUD COMPUTING TECHNOLOGY PRACTICUM REPORT

CLOUD SQL AND CLOUD STORAGE
CLOUD COMPUTING TECHNOLOGY PRACTICUM
PLUG - H

By:

Nama	**: Sabila Khairina Saktiwati**
NIM	**: 123220085**

INFORMATICS STUDY PROGRAM
DEPARTMENT OF INFORMATICS
FACULTY OF INDUSTRIAL ENGINEERING
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2025

APPROVAL PAGE

PRACTICUM REPORT

CLOUD SQL AND CLOUD STORAGE
CLOUD COMPUTING TECHNOLOGY PRACTICUM
PLUG - H

By:

Sabila Khairina Saktiwati 123220085

Checked and approved by the Practicum Assistant of Cloud Computing Technology.
At the date of: …………………

.

Approved by.

Practicum Assistant	**Practicum Assistant**
<u>Muhammad Rafli</u>	**<u>Sayang Sani</u>**
NIM 123210078	**NIM 123210044**

PREFACE

Praise and gratitude are expressed to the presence of God Almighty for His blessings and grace, which have enabled the completion of this practicum report. This report is prepared as part of fulfilling academic requirements and as a form of accountability for the practicum activities that have been carried out.

The author would like to express sincere appreciation to Muhammad Rafli, and Sayang Sani for their guidance and assistance throughout the practicum.

The author is fully aware that this report is still far from perfect. Therefore, constructive criticism and suggestions are highly expected for future improvements. Hopefully, this report can provide benefits to all relevant parties.

Yogyakarta, 7 Maret 2025

Author

TABLE OF CONTENTS

LIST OF FIGURE

CHAPTER I

INTRODUCTION

1.1 Background

In modern web application development, data management plays a vital role in ensuring efficiency, accessibility, and security. Traditionally, many applications rely on locally hosted databases for storing and retrieving data. While this approach is effective for small-scale applications, it presents several challenges as the application grows in terms of users and complexity. Issues such as limited scalability, restricted remote access, and the need for manual maintenance can hinder the performance and long-term sustainability of an application (Egger, 2009).

With the increasing demand for cloud-based solutions, businesses and developers are shifting towards cloud-managed databases that offer better performance, flexibility, and ease of management. One such solution is Google Cloud SQL, a fully managed relational database service that supports MySQL, PostgreSQL, and SQL Server. Google Cloud SQL provides automated backups, high availability, and seamless integration with web applications (Buga, 2023). By migrating from a locally hosted SQL database to Google Cloud SQL, applications can take advantage of enhanced scalability, security, and reliability, ensuring better data accessibility and management.

NotesApp, a web-based note-taking application, currently utilizes a locally hosted SQL database to store and manage user-generated notes. While the current implementation functions adequately, it faces limitations in handling increased traffic, remote accessibility, and database maintenance. Migrating to Google Cloud SQL is expected to resolve these issues and provide a more scalable and efficient infrastructure for NotesApp. This migration will enable the application to maintain optimal performance, improve security measures, and reduce maintenance overhead.

By adopting a cloud-based database solution, NotesApp can align with modern best practices in web application development, ensuring that it remains competitive and meets the demands of its growing user base.

1.2 Problem Formulation

1. How can a local SQL database be successfully connected to Google Cloud SQL?
2. What are the steps required to set up a Cloud SQL instance and configure it for use with NotesApp?

1.3 Objectives

1. To establish a secure and efficient connection between the local SQL database and Google Cloud SQL, ensuring smooth data migration and minimal downtime.
2. To set up and configure a Cloud SQL instance for NotesApp, including instance creation, authentication setup, and connection management.

1.4 Benefits

1. Users can efficiently connect their local SQL database to Google Cloud SQL, ensuring uninterrupted access to their data.
2. Users can follow a structured process to set up and configure a Cloud SQL instance, making it easier to manage and deploy databases.
3. Users can benefit from Google Cloud SQL’s built-in encryption, authentication mechanisms, and automated backups, ensuring data protection.
4. Users can experience enhanced database performance through Cloud SQL’s automated indexing, query optimization, and resource scaling.
5. Users can rely on Google Cloud SQL’s automated updates, backups, and performance monitoring, reducing manual administrative tasks.
6. Users can scale their database resources as needed, allowing NotesApp to support increasing data storage and user demands efficiently.

CHAPTER II

LITERATURE REVIEW

2.1 SQL in Cloud Computing

Cloud computing has revolutionized database management by offering scalable, secure, and high-performance solutions that overcome the limitations of traditional on-premise databases. Egger (2009) explores SQL in the Cloud, discussing the advantages and challenges of migrating structured databases to cloud environments. One of the primary benefits of cloud-based SQL is its ability to dynamically allocate resources based on demand, improving database efficiency and cost management. Additionally, cloud-based SQL services provide built-in security features, automated backups, and failover mechanisms, ensuring data integrity and availability.

Compared to on-premise databases, SQL in the cloud offers greater scalability—allowing applications to handle increased traffic without significant performance degradation. The cloud also introduces managed services that eliminate the need for manual database administration, thus reducing operational overhead. However, Egger (2009) highlights latency and data transfer costs as potential challenges when migrating large databases to the cloud. These aspects must be carefully considered when designing cloud-based database solutions.

2.2 Google Cloud SQL

Google Cloud SQL is a fully managed relational database service that provides high availability, security, and scalability for SQL-based applications (Buga, 2023). It supports MySQL, PostgreSQL, and SQL Server, making it a versatile choice for businesses looking to migrate from traditional databases.

Buga (2023) describes key advantages of Google Cloud SQL, including:

1. Automated Backups & Maintenance – Cloud SQL provides automatic daily backups, system patches, and version updates, reducing the burden on database administrators.

2. High Availability & Failover Support – The platform ensures zero-downtime maintenance and supports replication for improved reliability.
3. Scalability & Performance Optimization – Google Cloud SQL automatically scales based on usage, ensuring optimized query execution and efficient resource utilization.
4. Security & Compliance – With end-to-end encryption, Identity and Access Management (IAM) controls, and firewall rules, Cloud SQL enhances security against unauthorized access.

Cloud SQL also integrates seamlessly with Google Kubernetes Engine (GKE), App Engine, and Compute Engine, allowing for efficient database connectivity in cloud-based applications.

Despite these advantages, Buga (2023) notes potential drawbacks, such as higher operational costs compared to self-managed databases and network latency issues when dealing with large-scale data migrations. These challenges can be mitigated by optimizing queries and implementing caching mechanisms.

2.3 SQL Fundamentals

Understanding SQL is crucial for database management, whether using traditional on-premise systems or cloud solutions. Beaulieu (2009) provides an in-depth exploration of SQL fundamentals, including:

1. Data Definition Language (DDL): Commands like CREATE, ALTER, and DROP to define and modify database structures.
2. Data Manipulation Language (DML): Commands like INSERT, UPDATE, and DELETE to manage data records.
3. Query Optimization: Techniques such as indexing, partitioning, and using efficient join operations to improve query execution speed.
4. Transactions & ACID Compliance: Ensuring Atomicity, Consistency, Isolation, and Durability in database operations.

Beaulieu (2009) emphasizes the importance of query performance tuning, particularly in cloud environments where inefficient queries can lead to higher costs due to excessive compute resource consumption. Additionally, SQL fundamentals provide the foundation for implementing data migration, cloud integration, and real-time analytics.

CHAPTER III

METHODOLOGY

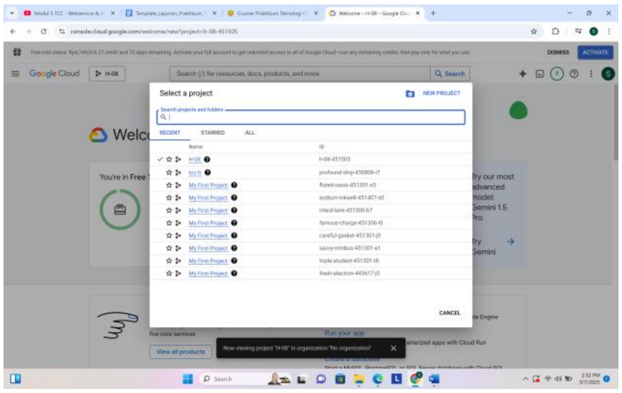
3.1 Problem Analysis

The task involves migrating a locally hosted SQL database to Google Cloud SQL to enhance scalability, security, and performance. The primary objectives are to establish a secure and efficient connection between the local database and Google Cloud SQL while ensuring minimal downtime and configuring the cloud instance to integrate seamlessly with NotesApp. Key challenges include setting up a reliable database connection, configuring the Cloud SQL instance, and managing authentication to prevent unauthorized access. Additionally, the data migration process must ensure the smooth transfer of database schema and records while maintaining data integrity. Furthermore, cost management and maintenance must be considered, leveraging Cloud SQL's automated backups, updates, and performance monitoring to reduce manual administrative tasks. Successfully implementing this migration will provide NotesApp with improved reliability, security, and scalability, ensuring a more efficient and robust database management system.

3.2 Solution Design

3.2.1 Create Instance

1. Sign in to the Google Cloud Console.
2. Choose a Project



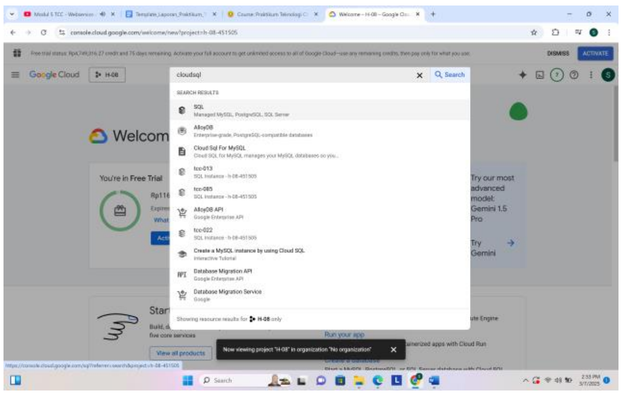
Select a project

Name	ID
H-08	h-08-451505
tcc-h	profound-ship-450806-i7
My First Project	fluted-oasis-451301-a5
My First Project	sodium-inkwell-451407-s0
My First Project	inlaid-lane-451306-b7
My First Project	famous-charge-451306-i0
My First Project	careful-gasket-451301-j0
My First Project	savvy-nimbus-451301-e1
My First Project	triple-student-451301-t8
My First Project	fresh-electron-443617-j5

Now viewing project "H-08" in organization "No organization"

Figure 1. Step 3.2.1.2

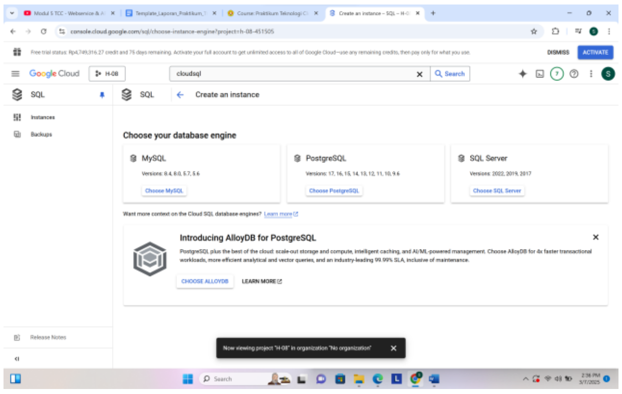
3. Navigate to the Cloud SQL section from the console.


console.cloud.google.com/welcome/new?project=h-08-451505
Google Cloud
H-08
cloudsql
Search
SEARCH RESULTS
SQL
Managed MySQL, PostgreSQL, SQL Server
AlloyDB
Enterprise-grade, PostgreSQL-compatible databases
Cloud Sql For MySQL
tcc-013
tcc-085
AlloyDB API
tcc-022
Create a MySQL instance by using Cloud SQL
Database Migration API
Database Migration Service
DISMISS
ACTIVATE
View all products

Now viewing project "H-08" in organization "No organization"

Figure 2. Step 3.2.1.3

4. Click Create Instance and select MySQl



Choose your database engine

MySQL — Versions: 8.4, 8.0, 5.7, 5.6

PostgreSQL — Versions: 17, 16, 15, 14, 13, 12, 11, 10, 9.6

SQL Server — Versions: 2022, 2019, 2017

Want more context on the Cloud SQL database engines? Learn more

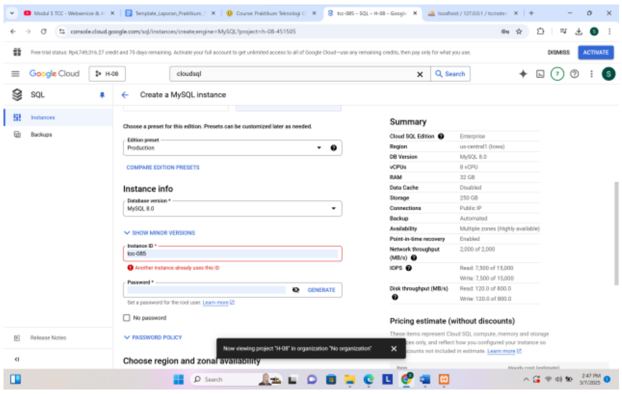
Introducing AlloyDB for PostgreSQL

PostgreSQL plus the best of the cloud: scale-out storage and compute, intelligent caching, and AI/ML-powered management. Choose AlloyDB for 4x faster transactional workloads, more efficient analytical and vector queries, and an industry-leading 99.99% SLA, inclusive of maintenance.

Now viewing project "H-08" in organization "No organization"

Figure 3. Step 3.2.1.4

5. Configure the instance settings:
   - Choose an instance ID (unique name for the database).
   - Select the region and zone for deployment.
   - Set the database version according to the compatibility of your application.
   - Configure machine type, storage, and memory allocation.


Create a MySQL instance

Choose a preset for this edition. Presets can be customized later as needed.

Edition preset: Production

COMPARE EDITION PRESETS

Instance info

Database version *: MySQL 8.0

SHOW MINOR VERSIONS

Instance ID *: tcc-085

Another instance already uses this ID

Password *

Set a password for the root user. Learn more

No password

PASSWORD POLICY

Choose region and zonal availability

Summary

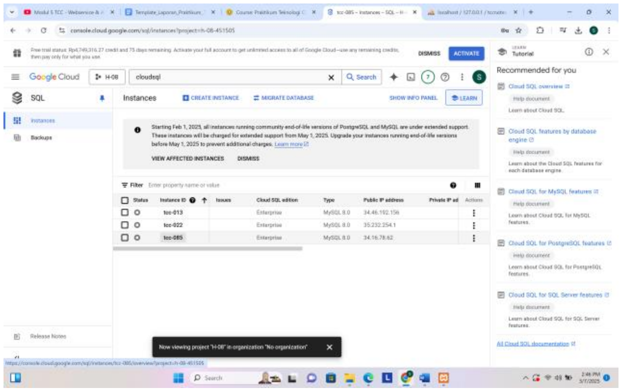
Cloud SQL Edition	Enterprise
Region	us-central1 (Iowa)
DB Version	MySQL 8.0
vCPUs	8 vCPU
RAM	32 GB
Data Cache	Disabled
Storage	250 GB
Connections	Public IP
Backup	Automated
Availability	Multiple zones (Highly available)
Point-in-time recovery	Enabled
Network throughput (MB/s)	2,000 of 2,000
IOPS	Read: 7,500 of 15,000 Write: 7,500 of 15,000
Disk throughput (MB/s)	Read: 120.0 of 800.0 Write: 120.0 of 800.0

Pricing estimate (without discounts)

Now viewing project "H-08" in organization "No organization"

Figure 4. Step 3.2.1.5

6. Click Create and wait for the instance to be provisioned.


Instances

CREATE INSTANCE MIGRATE DATABASE

Starting Feb 1, 2025, all instances running community end-of-life versions of PostgreSQL and MySQL are under extended support. These instances will be charged for extended support from May 1, 2025. Upgrade your instances running end-of-life versions before May 1, 2025 to prevent additional charges. Learn more

Status	Instance ID	Issues	Cloud SQL edition	Type	Public IP address	Private IP ad	Actions
	tcc-013		Enterprise	MySQL 8.0	34.46.192.156		
	tcc-022		Enterprise	MySQL 8.0	35.232.254.1		
	tcc-085		Enterprise	MySQL 8.0	34.16.78.62		

Now viewing project "H-08" in organization "No organization"

Figure 5. Step 3.2.1.6

3.2.2 Create Bucket

1. Go to Google Cloud Console and navigate to the Cloud Storage section.

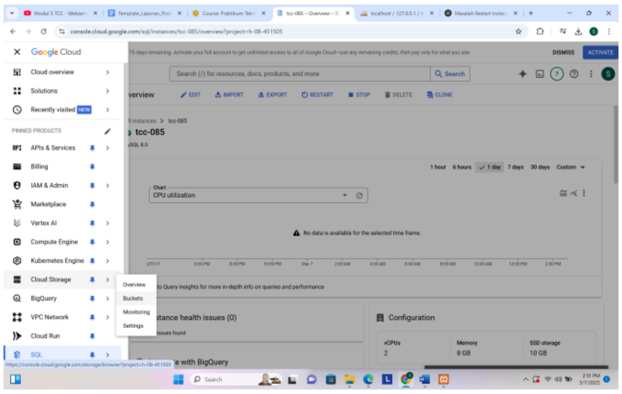
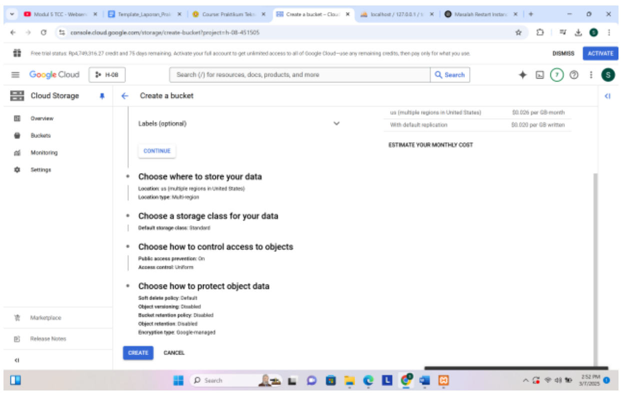


Figure 6. Step 3.2.2.1

2. Click Create Bucket and provide a unique name.
3. Choose a storage class based on data access needs.
4. Configure all the settings.


Create a bucket

Labels (optional)

CONTINUE

us (multiple regions in United States)	$0.026 per GB-month
With default replication	$0.020 per GB written

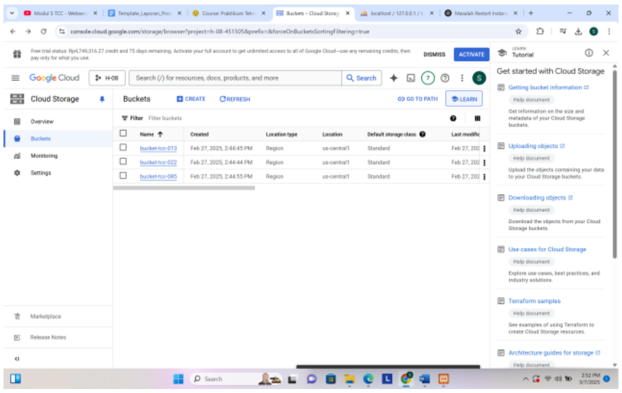
ESTIMATE YOUR MONTHLY COST

- Choose where to store your data
  Location: us (multiple regions in United States)
  Location type: Multi-region
- Choose a storage class for your data
  Default storage class: Standard
- Choose how to control access to objects
  Public access prevention: On
  Access control: Uniform
- Choose how to protect object data
  Soft delete policy: Default
  Object versioning: Disabled
  Bucket retention policy: Disabled
  Object retention: Disabled
  Encryption type: Google-managed

CREATE CANCEL

Figure 7. Step 3.2.2.4

5. Click Create to finalize the bucket setup.

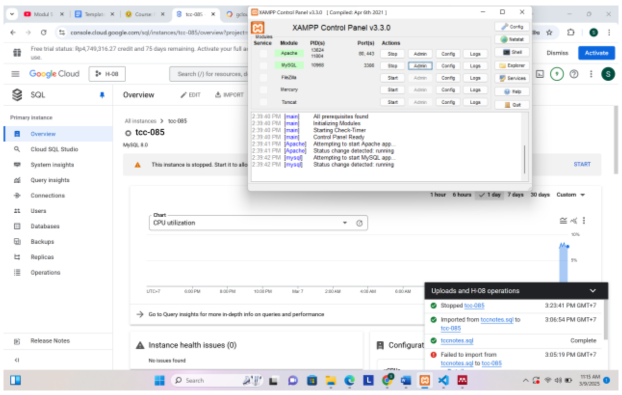

Buckets

Name	Created	Location type	Location	Default storage class	Last modific
bucket-tcc-013	Feb 27, 2025, 2:44:45 PM	Region	us-central1	Standard	Feb 27, 202
bucket-tcc-022	Feb 27, 2025, 2:44:44 PM	Region	us-central1	Standard	Feb 27, 202
bucket-tcc-085	Feb 27, 2025, 2:44:55 PM	Region	us-central1	Standard	Feb 27, 202

Figure 8. Step 3.2.2.5

3.2.3 Export Database SQL

1. Open XAMPP
2. Start Apache and MySQL

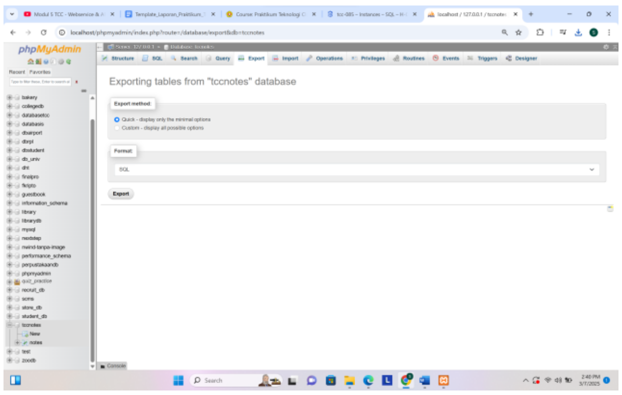

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	13624 11804	80, 443	Stop	Admin	Config	Logs
	MySQL	10968	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
2:39:40 PM  [main]    All prerequisites found
2:39:40 PM  [main]    Initializing Modules
2:39:40 PM  [main]    Starting Check-Timer
2:39:40 PM  [main]    Control Panel Ready
2:39:41 PM  [Apache]  Attempting to start Apache app...
2:39:41 PM  [Apache]  Status change detected: running
2:39:42 PM  [mysql]   Attempting to start MySQL app...
2:39:42 PM  [mysql]   Status change detected: running
```

Figure 9. Step 3.2.3.2

3. Go to Localhost Admin
4. Choose Database and Click Export



Exporting tables from "tccnotes" database
Export method:
Quick - display only the minimal options
Custom - display all possible options
Format:
SQL
Export

Figure 10. Step 3.2.3.4

3.2.4 Upload Database in Google Cloud

1. Go to Bucket Menu and Select a Bucket

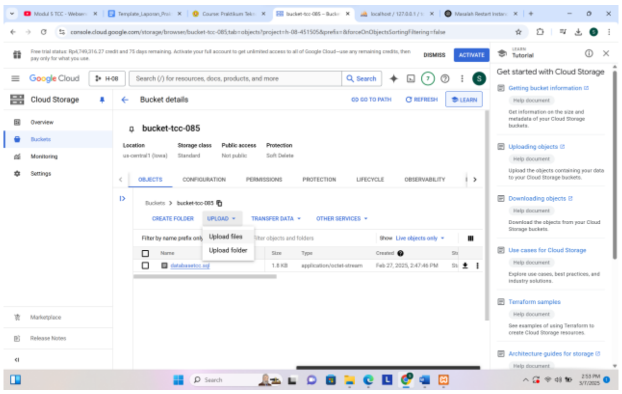


Figure 21. Step 3.2.4.1

2. Choose Upload files and Select the file

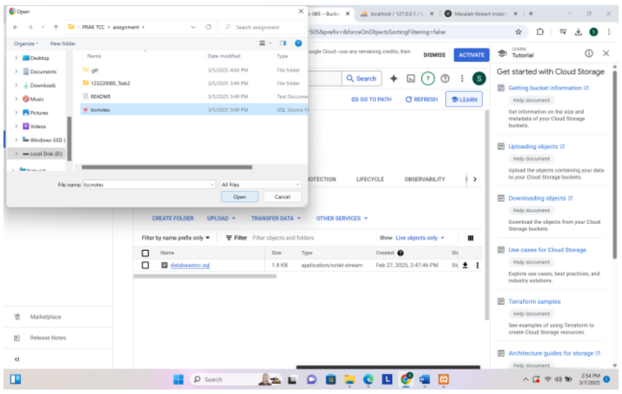
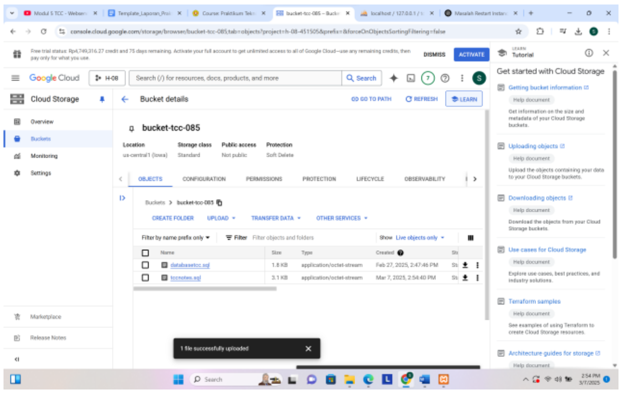


Figure 32. Step 3.2.4.2

3. Wait until the file uploaded



console.cloud.google.com/storage/browser/bucket-tcc-085;tab=objects?project=h-08-451505&prefix=&forceOnObjectsSortingFiltering=false

Free trial status: Rp4,749,316.27 credit and 75 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

Bucket details

bucket-tcc-085

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	Soft Delete

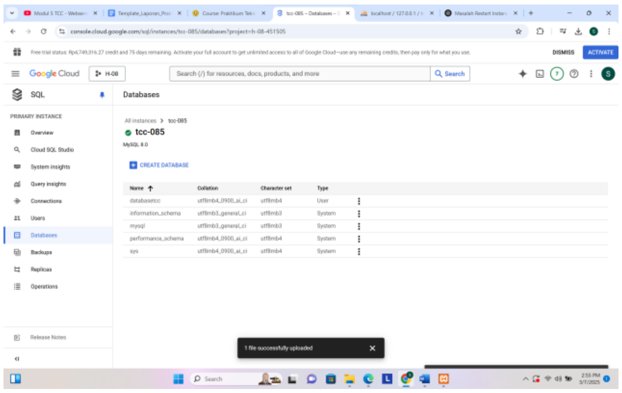
Buckets > bucket-tcc-085

Name	Size	Type	Created
databasetcc.sql	1.8 KB	application/octet-stream	Feb 27, 2025, 2:47:46 PM
tccnotes.sql	3.1 KB	application/octet-stream	Mar 7, 2025, 2:54:40 PM

1 file successfully uploaded

Figure 43. Step 3.2.4.3

4. Go to Instances Menu and Choose SQL Databases



console.cloud.google.com/sql/instances/tcc-085/databases?project=h-08-451505

Databases

All instances > tcc-085

tcc-085

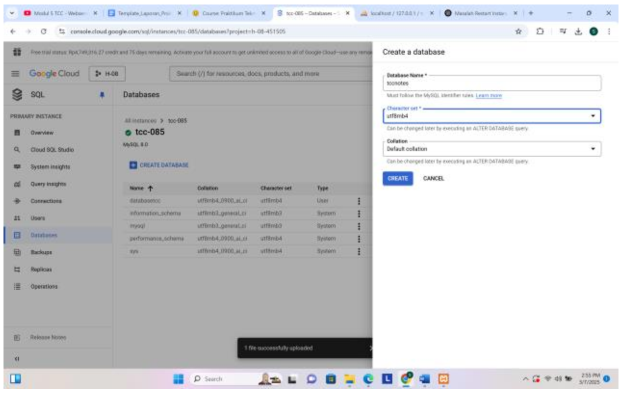
MySQL 8.0

Name	Collation	Character set	Type
databasetcc	utf8mb4_0900_ai_ci	utf8mb4	User
information_schema	utf8mb3_general_ci	utf8mb3	System
mysql	utf8mb3_general_ci	utf8mb3	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

1 file successfully uploaded

Figure 54. Step 3.2.4.4

5. Create a Database



Create a database

Database Name *: tccnotes

Must follow the MySQL identifier rules. Learn more

Character set *: utf8mb4

Can be changed later by executing an ALTER DATABASE query

Collation: Default collation

Can be changed later by executing an ALTER DATABASE query

Figure 65. Step 3.2.4.5

6. Go to Instances and Choose Import

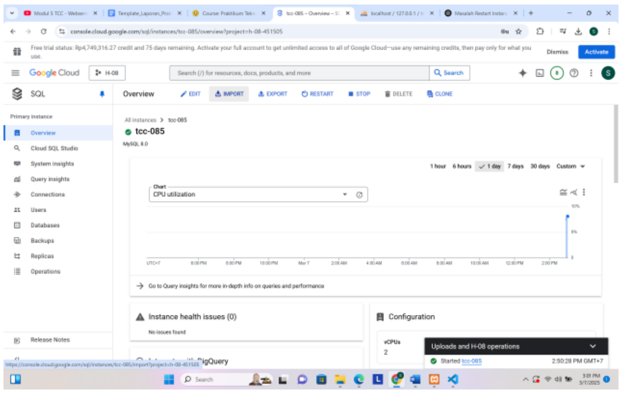


Figure 76. Step 3.2.4.6

7. Select the file that have uploaded before in Bucket Menu

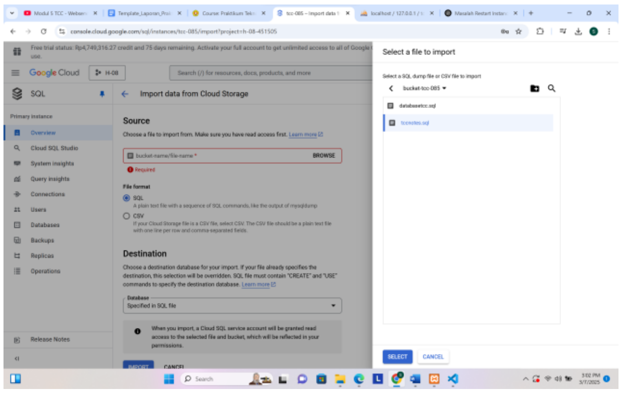


Figure 87. Step 3.2.4.7

8. Configure all the settings. For the destination choose database that have made before

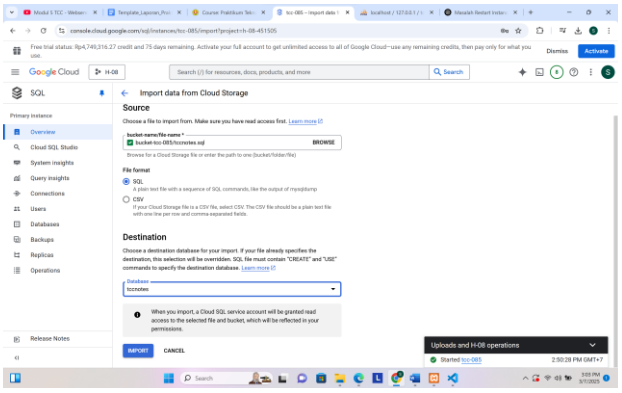
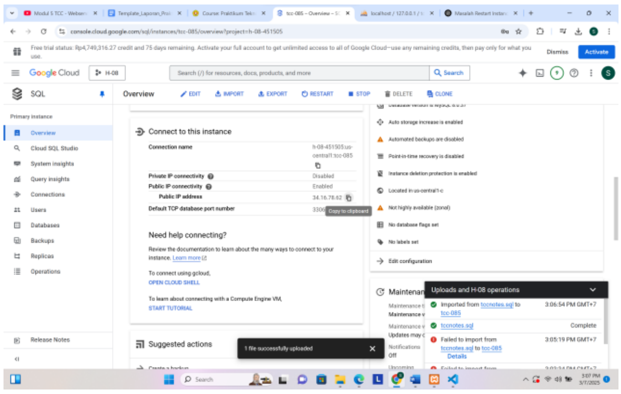


Figure 18. Step 3.2.4.8

9. Wait until import file successful

3.2.5 Connection Configuration from SQL to Google Cloud

1. Go to Instance Menu and Copy the Public IP Address



Connect to this instance

Connection name	h-08-451505:us-central1:tcc-085
Private IP connectivity	Disabled
Public IP connectivity	Enabled
Public IP address	34.16.78.62
Default TCP database port number	3306

Need help connecting?

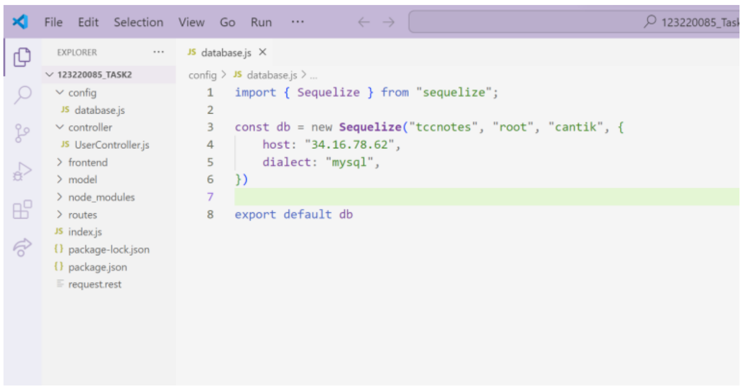
Review the documentation to learn about the many ways to connect to your instance. Learn more

To connect using gcloud, OPEN CLOUD SHELL

To learn about connecting with a Compute Engine VM, START TUTORIAL

Figure 99. Step 3.2.5.1

2. Paste IP to Source Code and Enter the instance password set during creation



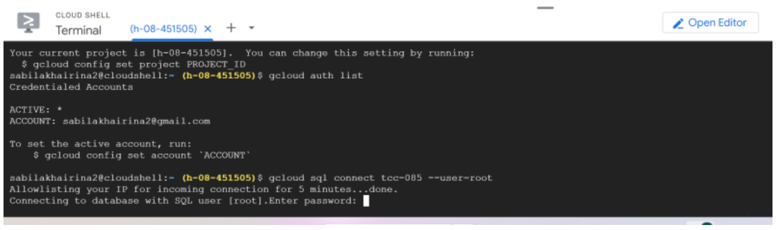
```
import { Sequelize } from "sequelize";

const db = new Sequelize("tccnotes", "root", "cantik", {
    host: "34.16.78.62",
    dialect: "mysql",
})

export default db
```

Figure 20. Step 3.2.5.2

3. In Google Cloud Shell, connect the database and enter the password



```
Your current project is [h-08-451505].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
sabilakhairina2@cloudshell:~ (h-08-451505)$ gcloud auth list
Credentialed Accounts

ACTIVE: *
ACCOUNT: sabilakhairina2@gmail.com

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

sabilakhairina2@cloudshell:~ (h-08-451505)$ gcloud sql connect tcc-085 --user=root
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
```

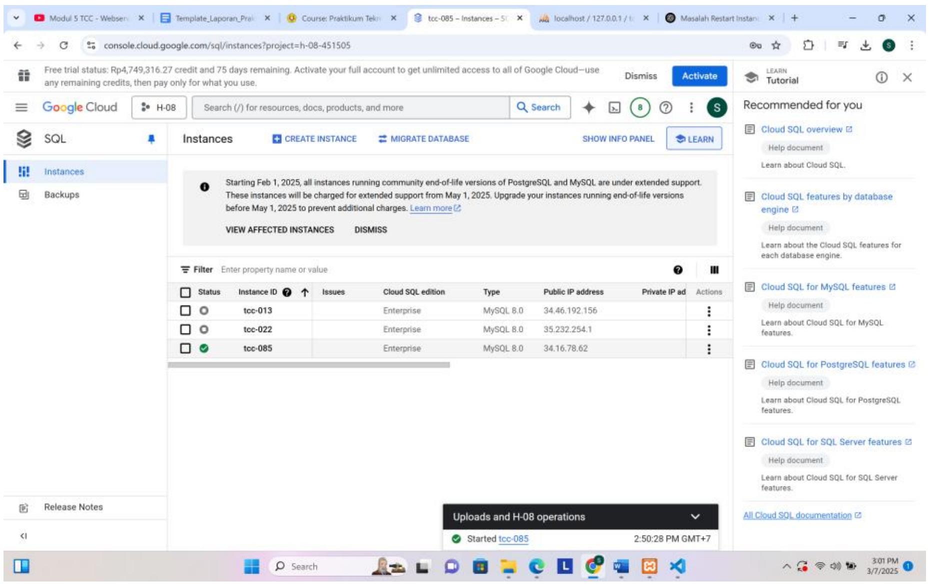
Figure 21. Step 3.2.5.3

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Results

4.1.1 Create Instance


console.cloud.google.com/sql/instances?project=h-08-451505

Free trial status: Rp4,749,316.27 credit and 75 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

Starting Feb 1, 2025, all instances running community end-of-life versions of PostgreSQL and MySQL are under extended support. These instances will be charged for extended support from May 1, 2025. Upgrade your instances running end-of-life versions before May 1, 2025 to prevent additional charges. Learn more

Status	Instance ID	Issues	Cloud SQL edition	Type	Public IP address	Private IP ad	Actions
	tcc-013		Enterprise	MySQL 8.0	34.46.192.156		
	tcc-022		Enterprise	MySQL 8.0	35.232.254.1		
✓	tcc-085		Enterprise	MySQL 8.0	34.16.78.62		

Uploads and H-08 operations

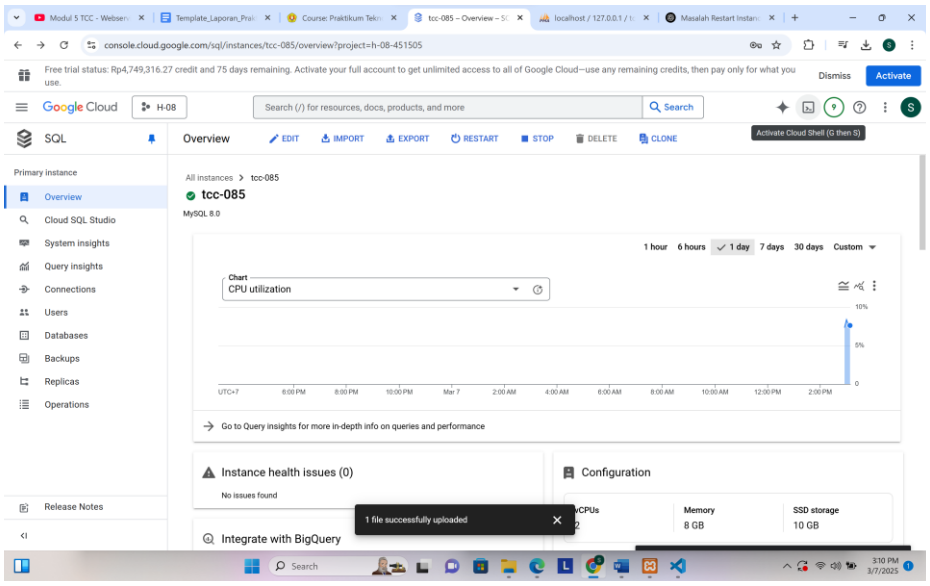
Started tcc-085 2:50:28 PM GMT+7

Instances
CREATE INSTANCE
MIGRATE DATABASE
SHOW INFO PANEL
LEARN
Backups
Release Notes
Recommended for you
Cloud SQL overview
Cloud SQL features by database engine
Cloud SQL for MySQL features
Cloud SQL for PostgreSQL features
Cloud SQL for SQL Server features
All Cloud SQL documentation

Figure 22. Create Instance

Google Cloud SQL Instances page displaying the created instances with public and private IP addresses. The instance will be used for hosting the NotesApp database.

4.1.2 Create Bucket


console.cloud.google.com/sql/instances/tcc-085/overview?project=h-08-451505

Free trial status: Rp4,749,316.27 credit and 75 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

All instances > tcc-085

tcc-085

MySQL 8.0

Go to Query insights for more in-depth info on queries and performance

Instance health issues (0)

No issues found

Configuration

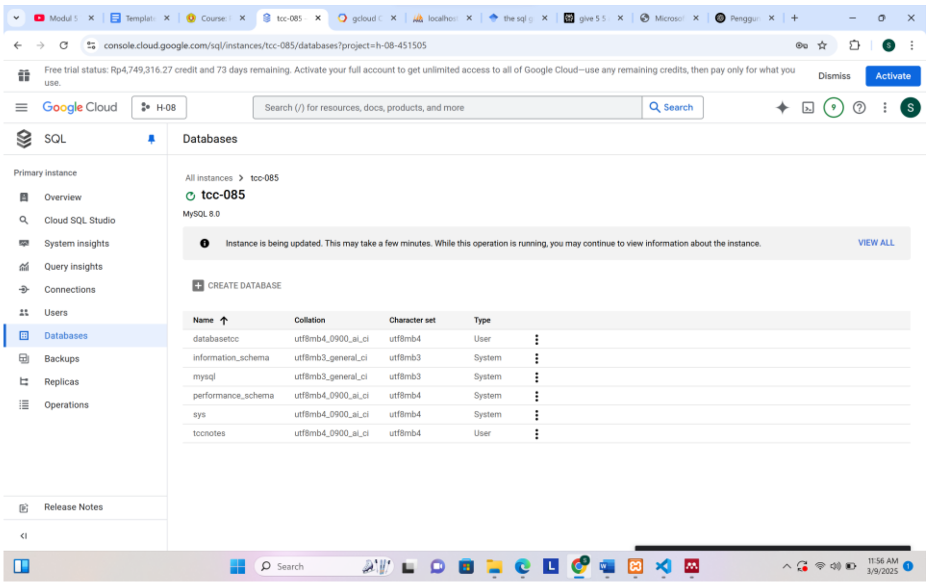
vCPUs	Memory	SSD storage
2	8 GB	10 GB

1 file successfully uploaded

Figure 23. Create Bucket

Google Cloud SQL Bucket page displaying the chosen bucket. The bucket will be used for saving the NotesApp database.

4.1.3 Upload Database in Google Cloud


Free trial status: Rp4,749,316.27 credit and 73 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.

All instances > tcc-085

tcc-085

MySQL 8.0

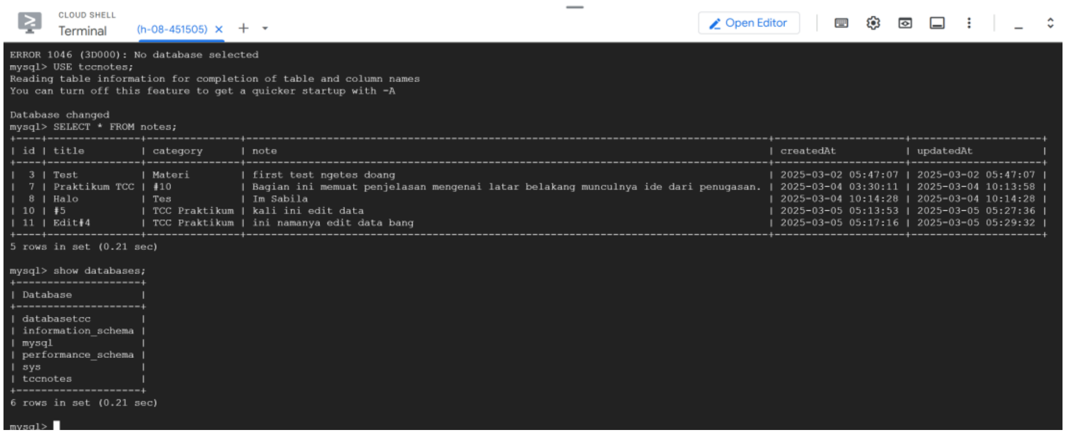
Instance is being updated. This may take a few minutes. While this operation is running, you may continue to view information about the instance.

Name	Collation	Character set	Type
databasetcc	utf8mb4_0900_ai_ci	utf8mb4	User
information_schema	utf8mb3_general_ci	utf8mb3	System
mysql	utf8mb3_general_ci	utf8mb3	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System
tccnotes	utf8mb4_0900_ai_ci	utf8mb4	User

Figure 24. Upload Database in Google Cloud

Database uploaded. Google Cloud SQL Databases page showing the created database within the selected instance. This database will store the NotesApp data.

4.1.4 Connection Configuration SQL to Google Cloud



```
ERROR 1046 (3D000): No database selected
mysql> USE tccnotes;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM notes;
+----+---------------+---------------+------------------------------------------------------------------------------------+---------------------+---------------------+
| id | title         | category      | note                                                                               | createdAt           | updatedAt           |
+----+---------------+---------------+------------------------------------------------------------------------------------+---------------------+---------------------+
|  3 | Test          | Materi        | first test ngetes doang                                                            | 2025-03-02 05:47:07 | 2025-03-02 05:47:07 |
|  7 | Praktikum TCC | #10           | Bagian ini memuat penjelasan mengenai latar belakang munculnya ide dari penugasan. | 2025-03-04 03:30:11 | 2025-03-04 10:13:58 |
|  8 | Halo          | Tes           | Im Sabila                                                                          | 2025-03-04 10:14:28 | 2025-03-04 10:14:28 |
| 10 | #5            | TCC Praktikum | kali ini edit data                                                                 | 2025-03-05 05:13:53 | 2025-03-05 05:27:36 |
| 11 | Edit#4        | TCC Praktikum | ini namanya edit data bang                                                         | 2025-03-05 05:17:16 | 2025-03-05 05:29:32 |
+----+---------------+---------------+------------------------------------------------------------------------------------+---------------------+---------------------+
5 rows in set (0.21 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| databasetcc        |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| tccnotes           |
+--------------------+
6 rows in set (0.21 sec)

mysql> 
```

Figure 25. Connection Successfu; to Google Cloud

Cloud Shell terminal displaying a successful connection to the Google Cloud SQL instance. The database 'tcconotes' is accessed, and SQL queries are executed to retrieve and display stored notes."

4.2 Discussion

The migration of the NotesApp database from a local SQL server to Google Cloud SQL was successfully implemented by following structured steps. The process included creating a Cloud SQL instance, setting up a Google Cloud Storage bucket, exporting and uploading the local database, and importing it into Google Cloud SQL. Once the migration was completed, verification was conducted using Cloud Shell, where SQL queries confirmed successful data transfer.

Several key aspects were observed during the migration:

1. The connection between NotesApp and the newly migrated Cloud SQL database was established using the public IP address of the instance. This ensured uninterrupted CRUD (Create, Read, Update, Delete) operations for managing user notes.
2. Google Cloud SQL provided automated indexing and query optimization, enhancing the responsiveness of NotesApp when fetching, updating, and deleting notes.
3. By migrating to Cloud SQL, NotesApp users can now access their notes from any device with an internet connection, improving availability and reliability. The cloud infrastructure also allows for automatic scaling, ensuring that increased user activity does not affect database performance.
4. Cloud SQL introduced automated backups, access control, and security updates, reducing the risks of data loss and unauthorized access. This eliminates the need for manual maintenance, making NotesApp more secure and efficient.

Despite these benefits, some challenges were encountered, such as configuring proper authentication methods and managing firewall settings to allow secure external access. However, these were resolved through Google Cloud's documentation and testing different connection methods.

CHAPTER V

CLOSING

5.1 Conclusion

The migration of the local SQL database to Google Cloud SQL for NotesApp was successfully completed, ensuring improved scalability, security, and performance. The process involved creating a Cloud SQL instance, exporting the local database, uploading it to Cloud Storage, and importing it into Cloud SQL, followed by configuring the connection for seamless integration with NotesApp. As a result, NotesApp now benefits from automated backups, enhanced security, better accessibility, and improved query performance, making it more reliable and efficient for users. This migration successfully ensures that NotesApp can handle growing data demands while reducing manual database management efforts.

5.2 Suggestions

For the future improvements, the recommendations are to explore further optimization of Cloud SQL performance, such as implementing read replicas, caching mechanisms, and query optimization techniques to enhance efficiency. Additionally, investigating the integration of serverless architectures like Firebase or Cloud Spanner could provide more scalability for applications like NotesApp. Security enhancements, such as advanced IAM roles, VPC configurations, and encryption strategies, should also be considered to strengthen data protection. Finally, conducting a comparative analysis between Cloud SQL and other cloud database solutions could offer valuable insights into cost-effectiveness, performance, and ease of management for large-scale applications. The following recommendations should be considered:

1. To enhance security, use private IP connections instead of public IP addresses to access Cloud SQL from NotesApp’s backend.
2. Monitor resource usage and adjust instance size based on NotesApp's database demands to balance performance and cost efficiency.

3. Maintain scheduled backups and set up replication to prevent data loss and ensure business continuity.

BIBLIOGRAPHY

[1] Egger, D. (2009). *SQL in the Cloud* (Master's thesis, ETH, Swiss Federal Institute of Technology, Department of Computer Science, Systems Group).

[2] Buga, S. (2023). Google Cloud SQL. In *Conferinţa tehnico-ştiinţifică a studenţilor, masteranzilor şi doctoranzilor* (Vol. 1, pp. 458-460).

[3] Beaulieu, A. (2009). *Learning SQL: master SQL fundamentals*. O'Reilly Media.

[4] Migrate a database to Cloud SQL for MySQL by using Database Migration Service. (2025). Retrieved March 8, 2025, from Google Cloud website: https://cloud.google.com/database-migration/docs/mysql/quickstart

[5] Gennick, J. (2010). *SQL Pocket Guide: A Guide to SQL Usage*. " O'Reilly Media, Inc.".

APPENDICES

1. https://github.com/stroberinanas/123220085-TCCPract.git