National Teachers College

629 J Nepomuceno, Quiapo, Manila, 1001 Metro Manila
Bachelor of Science in Information Technology

Database Management System for Sustainable Development Goal

Database Management System for Sustainable Development Goal

A Final Project

Presented to the Faculty of the
College of Information Technology

In partial fulfillment
of the Course Requirements for the degree
of Bachelor of Science in Information Technology

Submitted by:

Aguirre, Sean Xander C.

Iloseo, Allen Gene U.

Vera, Meljohn D.

BSIT 2.4

National Teachers College

Instructor:

Justin Louise R. Neypes

I. INTRODUCTION

I.I Project Overview & UN SDG Target

Our group put together a small database system. It deals with building safety inspections right there in Manila. The main aim was to sort out records for building owners. It covers their properties too. Plus all the inspections done on each one.

We tied this setup to the UN SDG 11 on Sustainable Cities and Communities. That means targets like 11.1. "Ensure access for all to adequate, safe and affordable housing and basic services". And also 11.3. "Enhance inclusive and sustainable urbanization". It pushes for cities that are safer. They end up more resilient. And better managed overall. The system keeps track of inspection outcomes. It spots buildings that fail. So it aids in stronger oversight. That way it boosts safer living areas.

This database management system for sustainable development makes it possible to catch unsafe buildings early. It does that with ongoing data gathering and review. Local officials and city planners can step in fast. They act before dangers turn into big problems. Such quick spotting cuts down on risks to lives and buildings. It makes sure fixes and upgrades follow real facts. Not just late checks that drag things out.

I.II Problem Statement

In today’s world, faulty buildings are often seen and fail to receive proper maintenance. This causes serious safety risks for occupants, increases the likelihood of structural failures, and contributes to unsafe and unsustainable urban environments. Without a reliable inspection and monitoring system, small issues such as cracks, water leaks, or electrical faults can escalate into major hazards. This problem directly affects the goal of creating inclusive, safe, resilient, and sustainable cities, as emphasized under UN SDG 11.

In real situations, building inspection data is often scattered, outdated, or not centralized. This causes delays in identifying unsafe buildings and makes it difficult for authorities to track which areas need immediate attention. This becomes a risk for residents and public safety. Our system solves this problem by placing all inspection information into one organized database. It allows users to track owners, buildings, and inspection results clearly, which helps monitor unsafe structures and supports SDG 11's goal of improving city safety.

II. REQUIREMENTS & ANALYSIS

II.I Functional Requirements and Non-Functional Requirements

The system is required to support several core functions to keep building safety information organized and reliable. It must store and maintain complete records for Owners, Buildings, and Inspections inside a relational database. The system also needs to show at least three DBMS concepts, including Views, a Self-Join View, and Stored Procedure, because these enhance the functionality and reporting features.

It must have the ability to record new inspections through a Stored Procedure that follows ACID properties to ensure safe and consistent data insertion. The system must also generate reports using Views. These include a list of buildings that failed inspection and a list of nearby buildings within the same location. These functions make it easier to analyze inspection results and understand potential safety concerns.

Functional Requirements

FR1. The system must store and maintain complete records for Owners, Buildings, and Inspections in the relational schema.

FR2. The system must demonstrate at least three DBMS concepts: two Views, one Self-Join View, one Stored Procedure.

FR3. The system must record new inspection entries using a Stored Procedure that follows ACID properties.

FR4. The system must generate reports using Views, including failed inspections and nearby buildings within the same district.

Non-Functional Requirements

NFR1 (Robustness): The database must reject invalid entries using constraints such as NOT NULL, CHECK, and Foreign Keys.

NFR2 (Maintainability): The schema and stored logic must be organized and easy to update. Views help simplify complex SELECT queries.

NFR3 (Performance): Queries should run efficiently and return results quickly, including the two Views.

II.II Data Requirements

The system manages three main entities, with more than 50 total records included in the initial dataset. Each table stores specific information and must follow the required data types: Owners Table, Buildings Table, and Inspection Table.

Owners Table - Be able to store basic information about each building owner.

- Attributes: owner_id (PK), owner_name, contact_number
- Data Types: Integer primary key, VARCHAR for name and contact
- Estimated Records: 10 sample owners

Buildings Table - Be able to hold information about buildings located across different districts in Manila.

- Attributes: building_id (PK), building_number, location, owner_id (FK)
- Data Types: VARCHAR for building number and location; integer foreign key
- Estimated Records: 20 buildings

Inspection Table - Be able to contain inspection details for each building.

- Attributes: inspection_id (PK), building_id (FK), inspection_date, status, notes
- Data Types: DATE, ENUM for status, TEXT for notes
- Estimated Records: 20 inspections

All together, the system is able to load more than 50 records during initialization, meeting the project requirement.

II.III Schema Normalization Analysis (3NF)

The schema follows Third Normal Form (3NF) because each table contains only attributes that depend on its primary key, and no unnecessary duplication exists between tables.

Selected Table for Analysis: Inspections - this is the most complex table and shows proper normalization. (Primary Key:Inspection_id)

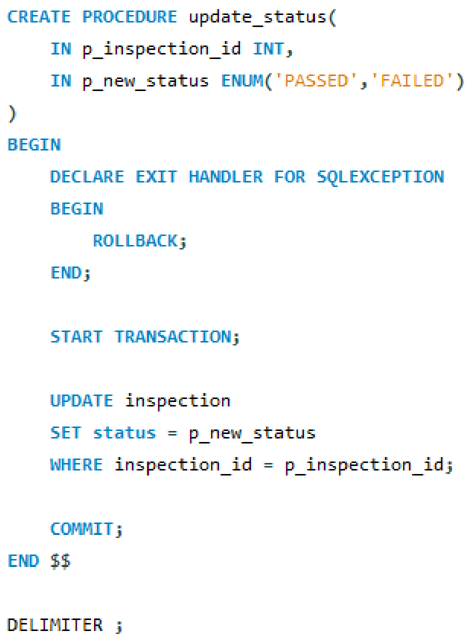
***Functional Dependencie**s*

1. inspection_id → building_id, inspection_date, status, notes (All attributes depend completely on the inspection_id.)
2. building_id → owner_id (exists in Buildings table, not inside Inspections)

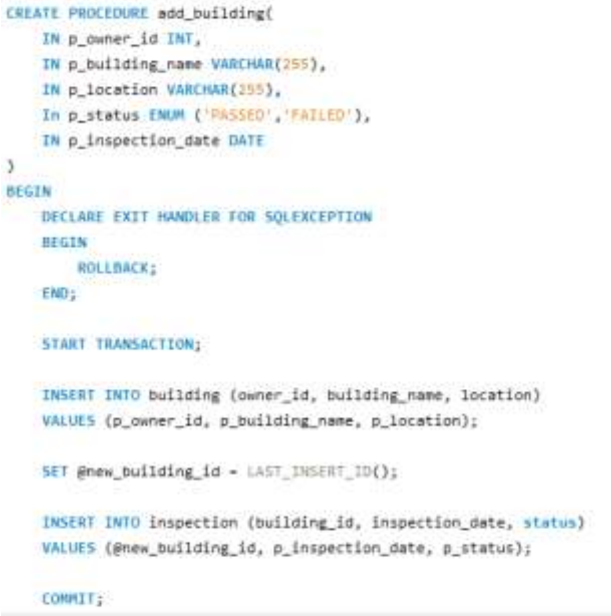
III. DESIGN SPECIFICATION

III.I Core DBMS Concepts Used

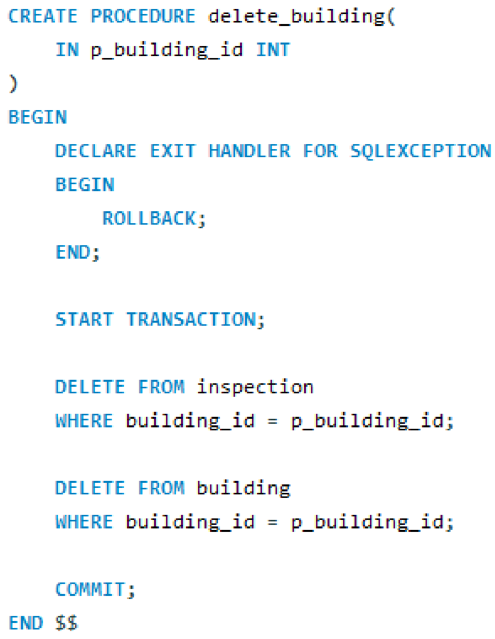
1. **Stored Procedures -** were used to handling inspection operations and adding/ deleting a building, because they allow safe updates and inserts from the CLI and also makes the system easy to navigate for the user.



```
CREATE PROCEDURE update_status(
    IN p_inspection_id INT,
    IN p_new_status ENUM('PASSED','FAILED')
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    UPDATE inspection
    SET status = p_new_status
    WHERE inspection_id = p_inspection_id;

    COMMIT;
END $$

DELIMITER ;
```



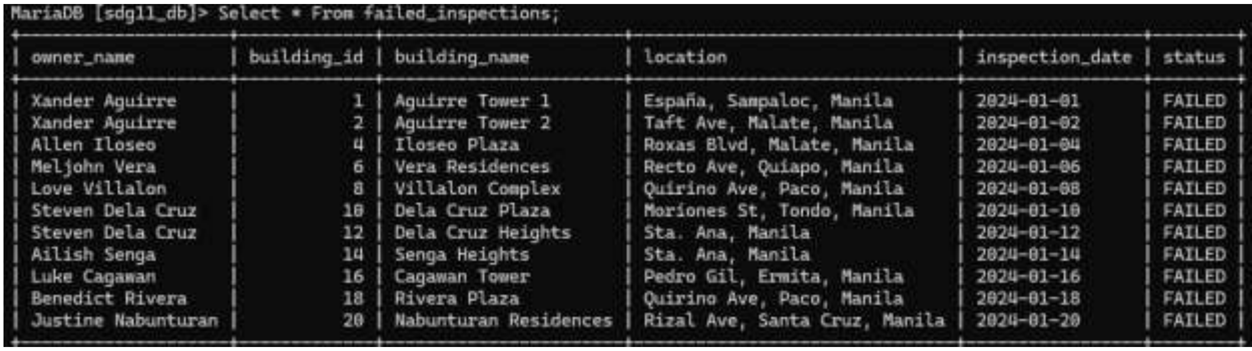
```
CREATE PROCEDURE add_building(
    IN p_owner_id INT,
    IN p_building_name VARCHAR(255),
    IN p_location VARCHAR(255),
    In p_status ENUM ('PASSED','FAILED'),
    IN p_inspection_date DATE
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    INSERT INTO building (owner_id, building_name, location)
    VALUES (p_owner_id, p_building_name, p_location);

    SET @new_building_id = LAST_INSERT_ID();

    INSERT INTO inspection (building_id, inspection_date, status)
    VALUES (@new_building_id, p_inspection_date, p_status);

    COMMIT;
```



```
CREATE PROCEDURE delete_building(
    IN p_building_id INT
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    DELETE FROM inspection
    WHERE building_id = p_building_id;

    DELETE FROM building
    WHERE building_id = p_building_id;

    COMMIT;
END $$
```

2. **Views** - Makes it easier for user to see inspection reports because it’s a ready made data sets (already uses combine multiple tables (Joins) **(Failed)**

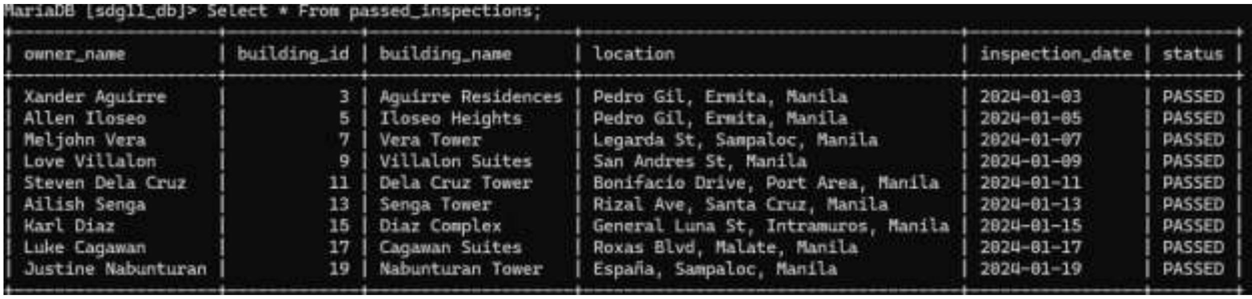
```
CREATE VIEW failed_inspections AS
SELECT
    o.owner_name,
    b.building_name,
    b.location,
    i.inspection_date,
    i.status
FROM inspection i
JOIN building b ON i.building_id = b.building_id
JOIN owner o ON b.owner_id = o.owner_id
WHERE i.status='FAILED';
```



```
MariaDB [sdg11_db]> Select * From failed_inspections;
+--------------------+-------------+-----------------------+-------------------------------+-----------------+--------+
| owner_name         | building_id | building_name         | location                      | inspection_date | status |
+--------------------+-------------+-----------------------+-------------------------------+-----------------+--------+
| Xander Aguirre     |           1 | Aguirre Tower 1       | España, Sampaloc, Manila      | 2024-01-01      | FAILED |
| Xander Aguirre     |           2 | Aguirre Tower 2       | Taft Ave, Malate, Manila      | 2024-01-02      | FAILED |
| Allen Iloseo       |           4 | Iloseo Plaza          | Roxas Blvd, Malate, Manila    | 2024-01-04      | FAILED |
| Meljohn Vera       |           6 | Vera Residences       | Recto Ave, Quiapo, Manila     | 2024-01-06      | FAILED |
| Love Villalon      |           8 | Villalon Complex      | Quirino Ave, Paco, Manila     | 2024-01-08      | FAILED |
| Steven Dela Cruz   |          10 | Dela Cruz Plaza       | Moriones St, Tondo, Manila    | 2024-01-10      | FAILED |
| Steven Dela Cruz   |          12 | Dela Cruz Heights     | Sta. Ana, Manila              | 2024-01-12      | FAILED |
| Ailish Senga       |          14 | Senga Heights         | Sta. Ana, Manila              | 2024-01-14      | FAILED |
| Luke Cagawan       |          16 | Cagawan Tower         | Pedro Gil, Ermita, Manila     | 2024-01-16      | FAILED |
| Benedict Rivera    |          18 | Rivera Plaza          | Quirino Ave, Paco, Manila     | 2024-01-18      | FAILED |
| Justine Nabunturan |          20 | Nabunturan Residences | Rizal Ave, Santa Cruz, Manila | 2024-01-20      | FAILED |
+--------------------+-------------+-----------------------+-------------------------------+-----------------+--------+
```

3. **(Passed)**

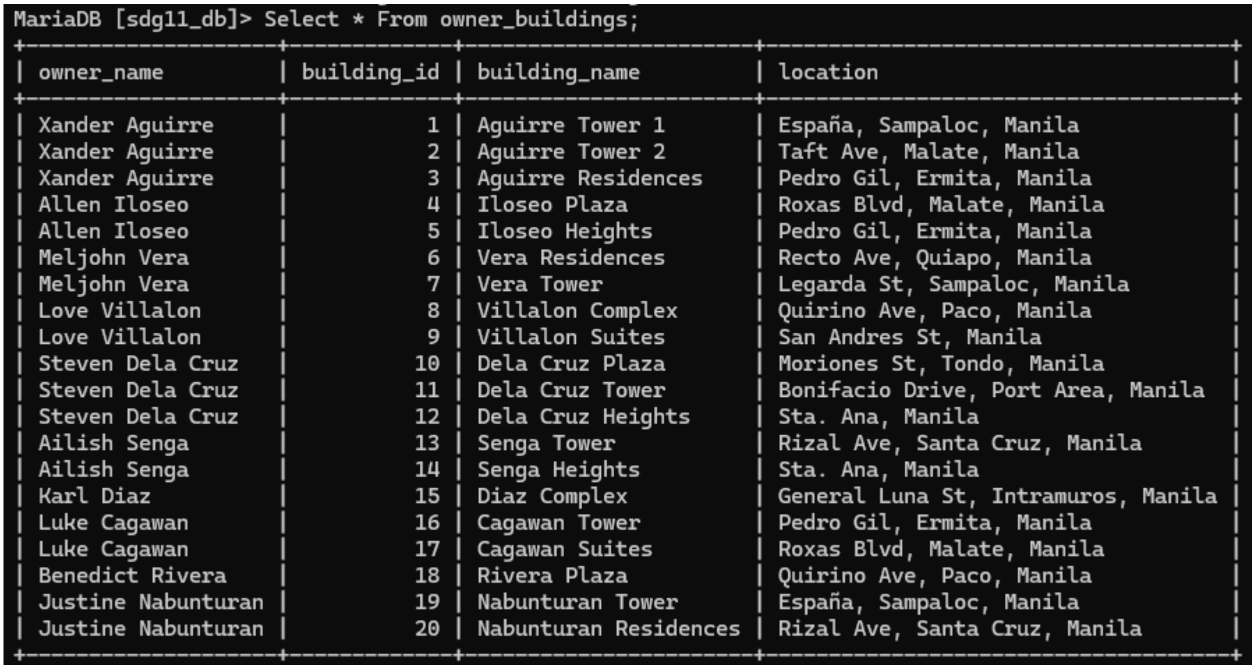
```
CREATE VIEW passed_inspections AS
SELECT
    o.owner_name,
    b.building_name,
    b.location,
    i.inspection_date,
    i.status
FROM inspection i
JOIN building b ON i.building_id = b.building_id
JOIN owner o ON b.owner_id = o.owner_id
WHERE i.status='PASSED';
```



```
MariaDB [sdg11_db]> Select * From passed_inspections;
+--------------------+-------------+--------------------+-------------------------------------+-----------------+--------+
| owner_name         | building_id | building_name      | location                            | inspection_date | status |
+--------------------+-------------+--------------------+-------------------------------------+-----------------+--------+
| Xander Aguirre     |           3 | Aguirre Residences | Pedro Gil, Ermita, Manila           | 2024-01-03      | PASSED |
| Allen Iloseo       |           5 | Iloseo Heights     | Pedro Gil, Ermita, Manila           | 2024-01-05      | PASSED |
| Meljohn Vera       |           7 | Vera Tower         | Legarda St, Sampaloc, Manila        | 2024-01-07      | PASSED |
| Love Villalon      |           9 | Villalon Suites    | San Andres St, Manila               | 2024-01-09      | PASSED |
| Steven Dela Cruz   |          11 | Dela Cruz Tower    | Bonifacio Drive, Port Area, Manila  | 2024-01-11      | PASSED |
| Ailish Senga       |          13 | Senga Tower        | Rizal Ave, Santa Cruz, Manila       | 2024-01-13      | PASSED |
| Karl Diaz          |          15 | Diaz Complex       | General Luna St, Intramuros, Manila | 2024-01-15      | PASSED |
| Luke Cagawan       |          17 | Cagawan Suites     | Roxas Blvd, Malate, Manila          | 2024-01-17      | PASSED |
| Justine Nabunturan |          19 | Nabunturan Tower   | España, Sampaloc, Manila            | 2024-01-19      | PASSED |
+--------------------+-------------+--------------------+-------------------------------------+-----------------+--------+
```

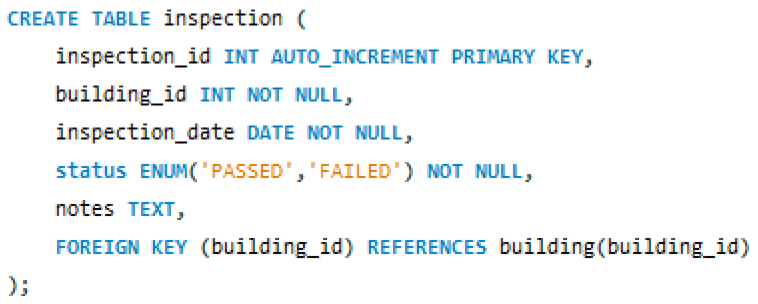
4. (Owner and Buildings)

```
CREATE VIEW owner_buildings AS
SELECT
    o.owner_name,
    b.building_id,
    b.building_name,
    b.location
FROM owner o
LEFT JOIN building b ON o.owner_id = b.owner_id;
```

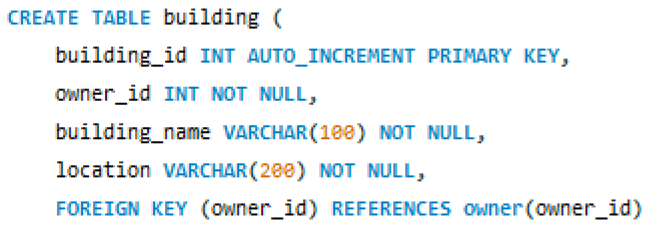


```
MariaDB [sdg11_db]> Select * From owner_buildings;
+--------------------+-------------+-----------------------+-------------------------------------+
| owner_name         | building_id | building_name         | location                            |
+--------------------+-------------+-----------------------+-------------------------------------+
| Xander Aguirre     |           1 | Aguirre Tower 1       | España, Sampaloc, Manila            |
| Xander Aguirre     |           2 | Aguirre Tower 2       | Taft Ave, Malate, Manila            |
| Xander Aguirre     |           3 | Aguirre Residences    | Pedro Gil, Ermita, Manila           |
| Allen Iloseo       |           4 | Iloseo Plaza          | Roxas Blvd, Malate, Manila          |
| Allen Iloseo       |           5 | Iloseo Heights        | Pedro Gil, Ermita, Manila           |
| Meljohn Vera       |           6 | Vera Residences       | Recto Ave, Quiapo, Manila           |
| Meljohn Vera       |           7 | Vera Tower            | Legarda St, Sampaloc, Manila        |
| Love Villalon      |           8 | Villalon Complex      | Quirino Ave, Paco, Manila           |
| Love Villalon      |           9 | Villalon Suites       | San Andres St, Manila               |
| Steven Dela Cruz   |          10 | Dela Cruz Plaza       | Moriones St, Tondo, Manila          |
| Steven Dela Cruz   |          11 | Dela Cruz Tower       | Bonifacio Drive, Port Area, Manila  |
| Steven Dela Cruz   |          12 | Dela Cruz Heights     | Sta. Ana, Manila                    |
| Ailish Senga       |          13 | Senga Tower           | Rizal Ave, Santa Cruz, Manila       |
| Ailish Senga       |          14 | Senga Heights         | Sta. Ana, Manila                    |
| Karl Diaz          |          15 | Diaz Complex          | General Luna St, Intramuros, Manila |
| Luke Cagawan       |          16 | Cagawan Tower         | Pedro Gil, Ermita, Manila           |
| Luke Cagawan       |          17 | Cagawan Suites        | Roxas Blvd, Malate, Manila          |
| Benedict Rivera    |          18 | Rivera Plaza          | Quirino Ave, Paco, Manila           |
| Justine Nabunturan |          19 | Nabunturan Tower      | España, Sampaloc, Manila            |
| Justine Nabunturan |          20 | Nabunturan Residences | Rizal Ave, Santa Cruz, Manila       |
+--------------------+-------------+-----------------------+-------------------------------------+
```

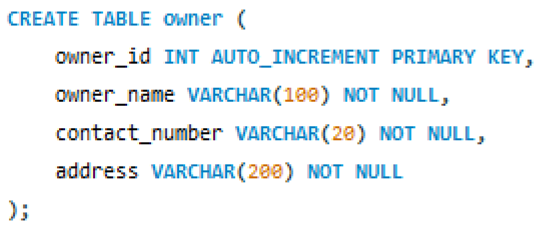
5. **Constraints** – this ensures that the data in the database are accurate, valid, and don’t have an repetition. (PK, FK, Check, Not Null)



```
CREATE TABLE inspection (
    inspection_id INT AUTO_INCREMENT PRIMARY KEY,
    building_id INT NOT NULL,
    inspection_date DATE NOT NULL,
    status ENUM('PASSED','FAILED') NOT NULL,
    notes TEXT,
    FOREIGN KEY (building_id) REFERENCES building(building_id)
);
```



```
CREATE TABLE building (
    building_id INT AUTO_INCREMENT PRIMARY KEY,
    owner_id INT NOT NULL,
    building_name VARCHAR(100) NOT NULL,
    location VARCHAR(200) NOT NULL,
    FOREIGN KEY (owner_id) REFERENCES owner(owner_id)
```



```
CREATE TABLE owner (
    owner_id INT AUTO_INCREMENT PRIMARY KEY,
    owner_name VARCHAR(100) NOT NULL,
    contact_number VARCHAR(20) NOT NULL,
    address VARCHAR(200) NOT NULL
);
```

6. **ERM Entity Relationship Model** – We used a one – many relationship model because an owner can own more than one building.
7. **SQL Joins** – used in stored procedure

III.II ER Diagram

Conceptual Diagram

OWNER

Attributes:

- Owner_ID (PK),
- Owner_Name,

- Contact_Number,

BUILDING

Attributes:

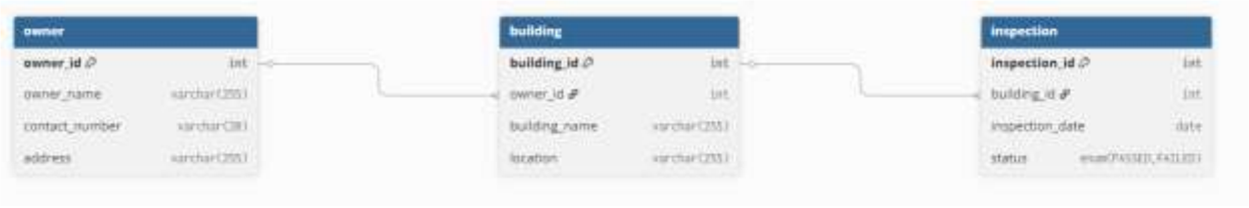
- Building ID (PK)
- Owner ID (FK)
- Building Name
- Location

INSPECTION

Attributes:

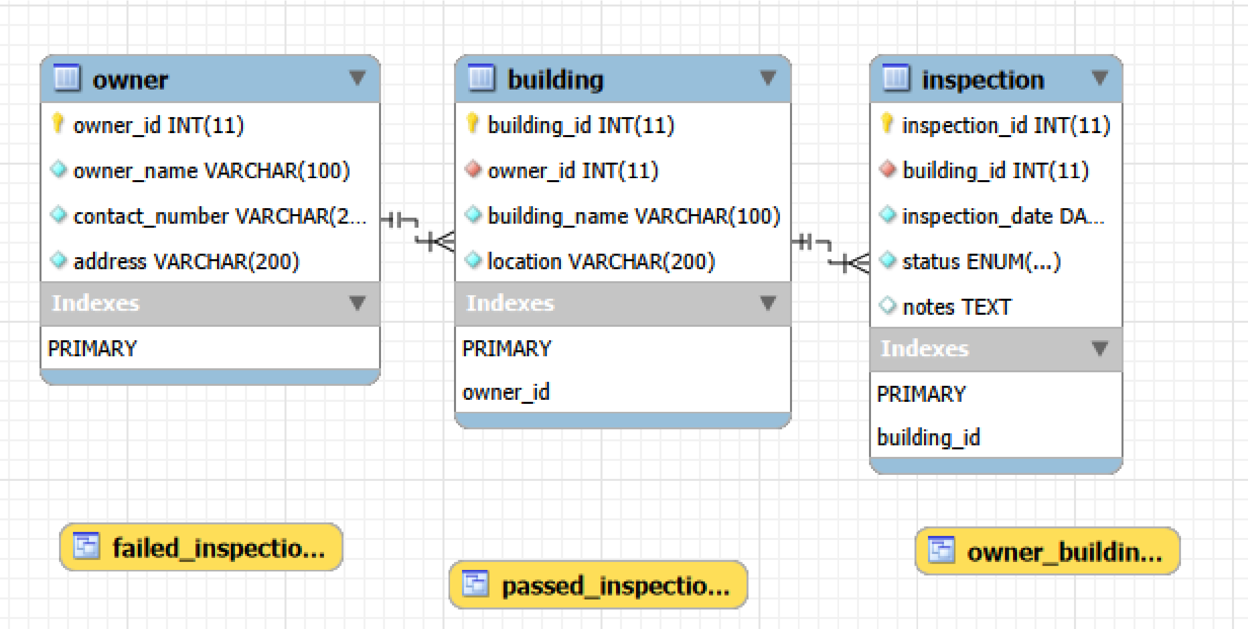
- Inspection ID (PK)
- Building ID (FK)
- Inspection Date
- Status

Logical Diagram



Shows the entities Owner, Building, and Inspection with their attributes and relationships, highlighting that one owner can have many buildings and each building has exactly one inspection.

Physical Diagram



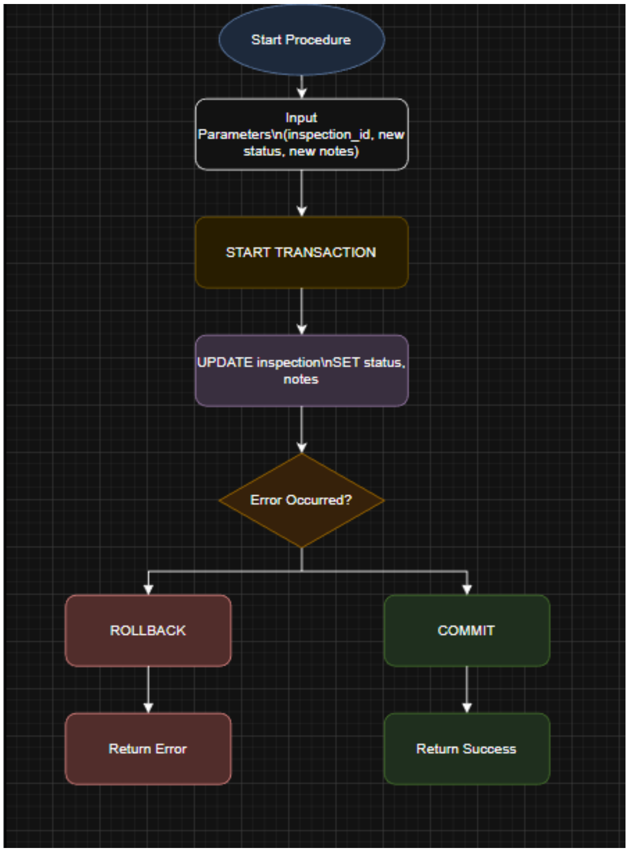
owner
owner_id INT(11)
owner_name VARCHAR(100)
contact_number VARCHAR(2...
address VARCHAR(200)
Indexes
PRIMARY

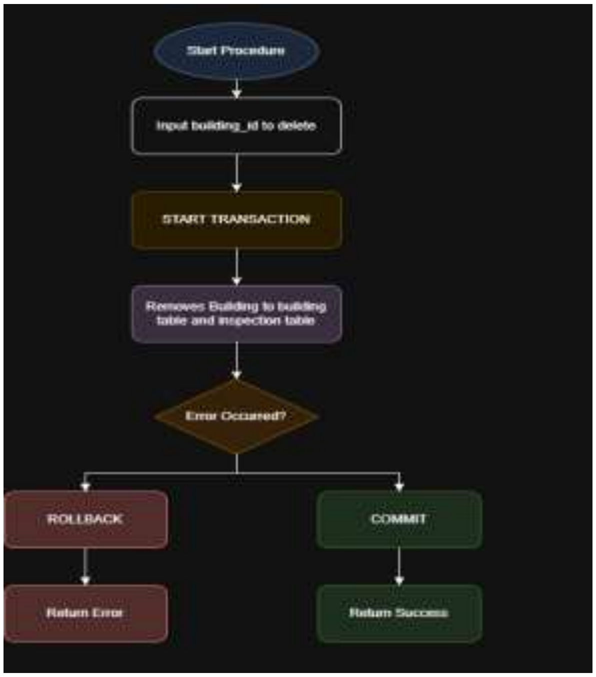
building
building_id INT(11)
owner_id INT(11)
building_name VARCHAR(100)
location VARCHAR(200)
Indexes
PRIMARY
owner_id

inspection
inspection_id INT(11)
building_id INT(11)
inspection_date DA...
status ENUM(...)
notes TEXT
Indexes
PRIMARY
building_id

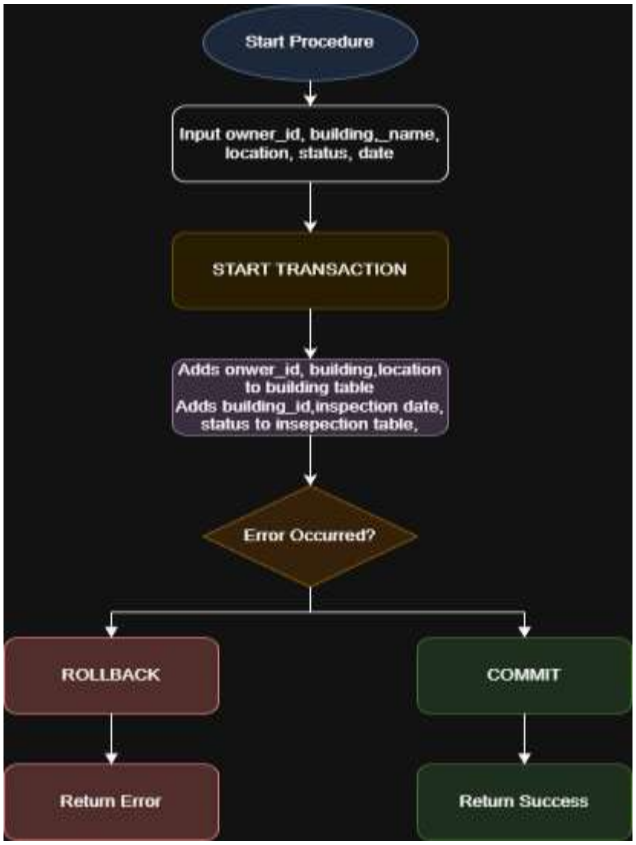
Illustrates the actual table implementation with primary keys, foreign keys, data types, and constraints, ensuring data integrity and supporting efficient queries for building inspections.

III.III Transaction Flowchart


Start Procedure
Input Parameters\n(inspection_id, new status, new notes)
START TRANSACTION
UPDATE inspection\nSET status, notes
Error Occurred?
ROLLBACK
COMMIT
Return Error
Return Success



This graphs shows the flowchart of each transaction inside the stored procedures


Start Procedure
Input owner_id, building,_name, location, status, date
START TRANSACTION
Adds onwer_id, building,location to building table
Adds building_id,inspection date, status to insepection table,
Error Occurred?
ROLLBACK
COMMIT
Return Error
Return Success

IV. Conclusion and Contributions

V.I Conclusions

The system supports SDG 11 by organizing building inspection data in a clear and reliable way. With normalization, proper constraints, Views, and a Stored Procedure, the system ensures data accuracy and helps produce useful reports. It demonstrates how database systems can help improve safety and decision-making in communities.

The project is able to apply the DBMS concept to be able to bridge the theoretical database concepts, demonstrating that a well-thought database management system serve not only as a temporary solution for the implementation of the buildings, but make it sustainable and safe for urban development. Through the implementation of building inspection in the projects, makes it much more accessible, reliable, and analyzable, as this system are able to contribute to the creation of safer urban environments.

V.II Individual Contributions

Aguirre – contributed by helping refine the SDG and problem statement, helped developing ERD Diagrams, led the process for stored procedures and views. , participated in ACID testing and issue resolution, and took the lead in consolidating final documentation, completing the transaction flowchart, coordinating the presentation, and assisting with final testing and debugging using the Command Line Interface (CLI).

Iloseo – assisted in finalizing the SDG and problem statement, contributed to the logical and physical ERD, and wrote and reviewed sections of the SDAD document. They took part in managing the GitHub workflow, implementing DDL structures and normalization, and generating mock data. Iloseo led the development of transactional stored procedures, reporting views, conducted ACID compliance tests, and contributed to final documentation, flowchart refinement, and presentation preparation, including system testing and debugging.

Vera – contributed to refining the SDG and problem statement, assisted in conceptual ERD development, and supported the writing and editing of the SDAD document. They helped set GitHub collaboration guidelines, participated in DDL implementation and normalization, and assisted in producing consistent mock data. Vera also supported the development of stored procedures, and reporting views, participated in ACID testing and issue resolution, and took the lead in consolidating final documentation, completing the transaction flowchart, coordinating the presentation, and assisting with final testing and debugging.

GITHUB LINK: https://github.com/424000756-design/20MinutesLate__SDG11_Project.git