CS 157A Final Project

Ajeet Kotturu

Jovindio Wieuse

Nate Agpaoa

CS 157A **(**Jahan Ghofraniha)

December 12th, 2022

Table of Contents

I. EXECUTIVE SUMMARY

There are many schools across the globe in which they are still heavily reliant on paper to keep track of their important data. This is not ideal as they degrade, and the risk of losing them in an incident such as natural disaster is very likely with the chance of recoverability being near zero. Migrating to a digital-based database will solve the problem entirely provided it is properly implemented. The solution that we are pushing will be useful since it is open source (it costs $0 for them to use). The system we offer provides core functionality that is likely to apply to all schools, and each school needs to tweak it to adopt a database system each one of them really wants.

II. BACKGROUND

Many primary education institutes around the world, especially developing countries, still use paper to keep important data. This itself is a problem because physical matter degrades over time. Schools around the world try to organize their data in a neat manner but due to a lack of resources schools have around the world they do not have the capability to organize the data. By managing data from students for each single year, the schools must create new data for every student that joins and students who leave the school the data needs to be destroyed or transferred. This results in data being misplaced or mixed up which becomes very problematic when transferring the records to other schools.

In addition, natural disasters can damage paper data. This makes backing up data very hard because of the ridiculous amount of work trying to regather the lost data. Even if the paper data were to be recovered, the damage caused either directly or indirectly from natural disasters can make the written data hard to transcribe or even impossible.

III. PROBLEM STATEMENT

Natural disasters occur everywhere, almost every minute. This puts schools in the vicinity of the danger at great risk of potentially losing all their important data permanently. Not only this, attacks from burglary may also pose a threat to students' private information. While it is something that people must face with, technology has been evolving for many decades now that it is long overdue for those schools to be modernized for both ease of access and safety perspectives.

IV. PURPOSE AND MOTIVATION

One of the main reasons for using a digital database is that it solves the issue of storing paper data. You won't need to worry about digital database degrading because it is stored in the computer. That data can be backed up in a cloud server to be on the safer side. In addition, institutions will not need to pay for using digital databases and storing data inside the database because there are programs and services that are free and open source.

Another main reason for using a digital database is so that the data can easily be transferred to other schools or institutions without worrying about the data being disorganized. With multiple primary education institutes around the world using the same kind of database schools do not need to spend extra time converting the data and can avoid having anomalies with the student data such as having record for two students with the same name.

V. DESIGN PROCESS

5.1. REQUIREMENTS GATHERING

The primary goal of this database is to store data regarding students and staff members. When gathering this information, we focus on information that the students will need when attending a school along with crucial data that is needed for the courses the students take.

We've gathered the type of information that a school should have. Obviously, we need students and staff for a school. So, we must create tables for students and staff and the information about them.

Attendance is required for primary schools, so we designed a table that stores a student's attendance, whether they were absent or present.

Now we need to determine the type of absence because there are multiple reasons why a student may be absent for a class. With the students and staff information table created, at least the conceptual design, we also need courses that these students take and who teaches these courses. That’s where we create the course catalog. We also realize that these courses must have a prerequisite for students to take these courses. As a result, we must create a table for the course prerequisites for the course offerings.

A course must have a location. So that’s where we must create a table for locations and their maximum capacity. However, there would be a limit on how many people can take a course

Now that we gathered the types of data to be stored in our school database, below is the initial table data:

STUDENT_INFORMATION = STUDENT_ID, LAST_NAME, FIRST_NAME, DATE_OF_BIRTH, ADMITTED_DATE, STUDENT_ADDRESS, SEX

STUDENT_GRADE_YEAR = STUDENT_ID, GRADE_YEAR

STUDENT_COURSE = STUDENT_ID, COURSE_ID, TERM, FINAL_GRADE

STUDENT_ATTENDANCE = STUDENT_ID, DATE, ABSENCE_ID, REASON

ABSENCE_TYPES = ABSENCE_ID, ABSENCE_TITLE, DESCRIPTION

STAFF_INFORMATION = INSTRUCTOR_ID, ROLE_ID, LAST_NAME, FIRST_NAME, SSN, SALARY, PHONE_NUMBER, DATE_OF_BIRTH, ADMITTED_DATE, STAFF_ADDRESS

STAFF_ROLE = ROLE_ID, ROLE_TITLE, ROLE_DESCRIPTION

COURSE_CATALOG = COURSE_ID, ASSIGNED_INSTRUCTOR_ID, REQUIREMENT_ID, LOCATION, TITLE, DESCRIPTION, DEPARTMENT_TITLE, DEPARTMENT_DESCRIPTION

COURSE_REQUIREMENTS = REQUIREMENT_ID, REQUIREMENT_TITLE, REQUIREMENT_DESCRIPTION

LOCATION = LOCATION_ID, LOCATION_NAME, MAX_CAPACITY

Highlighted items are some of the fields from the table that are not mandatory in order for the database to function.

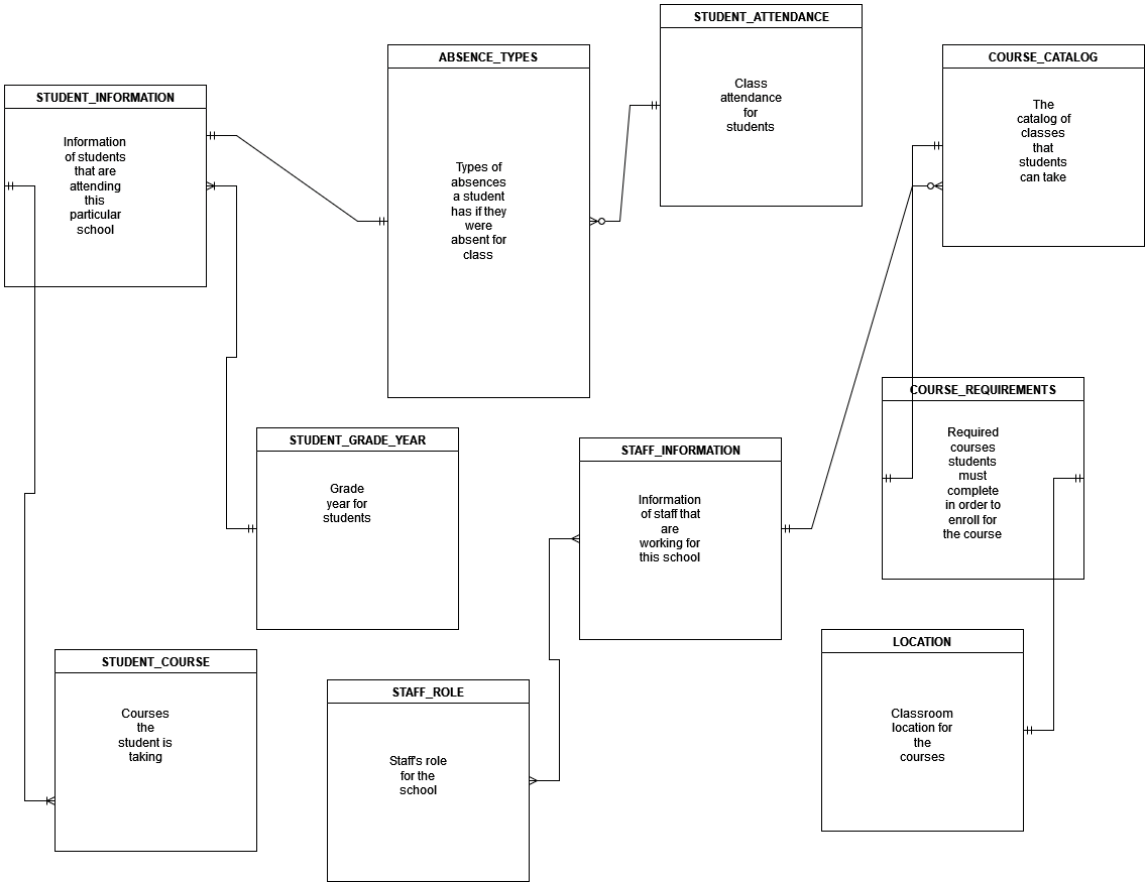
In addition to the table data, we created three views based on the individual's role in that school as follows:

Headmaster: read general information about students or staff and their respective information

Teacher: read and write access of student's information on specific class (needs filter by teacher's embedded class)

Student: read access of student's course history (needs filter by student's ID)

Initial Table Diagram:



5.2. CONCEPTUAL DESIGN

5.2.1. BUSINESS RULES

i. Titles and descriptions need to match (e.g., role titles and role description must match).

ii. Instructors must be experienced (must be born no later than today's date, 1997).

iii. The maximum number of students for a course is based on the room capacity.

iv. Locations are based off a single country for consistency per database, meaning students and staff must be from the same country.

v. The information that is visible is based on the views (e.g., students can only see their own data, staff can see their students' information and their own information as well, etc.).

5.2.2. DESIGN SCREEN

The design screens are the views that will be represented on the MySQL workbench (note that this is after we finished the normalization process of the tables and implement the physical design):

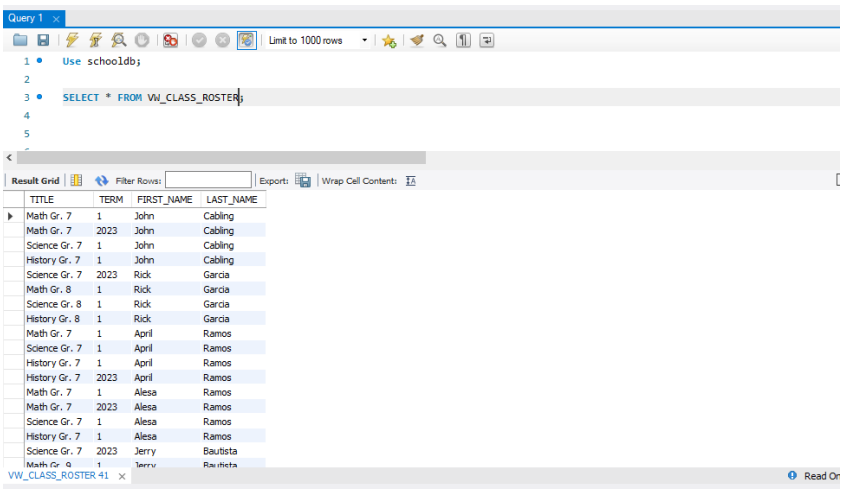


Figure 1: Class Roster View

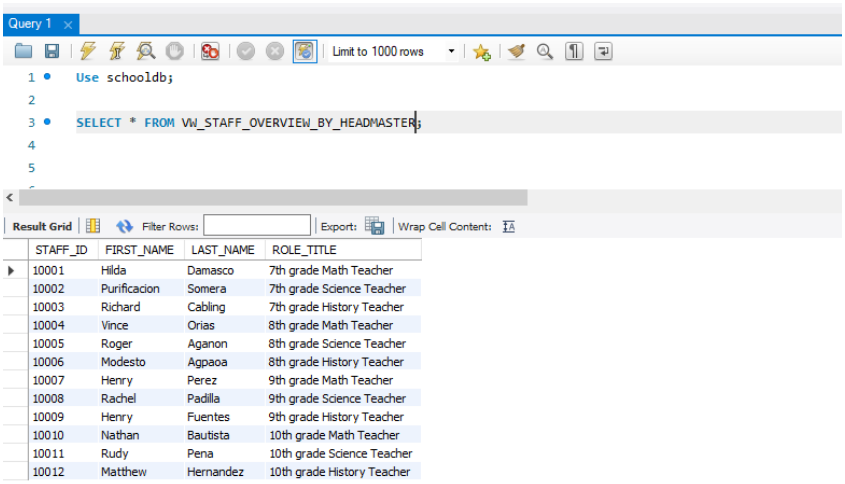


Figure 2: Staff Overview by Headmaster View

```
Query 1
Limit to 1000 rows
1  Use schooldb;
2
3  SELECT * FROM VW_STUDENT;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

STUDENT_ID	COURSE_ID
68391	1
68391	1
68391	2
68391	3
68392	2
68392	4
68392	5
68392	6
68393	1
68393	2
68393	3
68393	3
68394	1
68394	1
68394	2
68394	3
68395	2
68395	7

Figure 3: Student View

```
Query 1
Limit to 1000 rows
1  Use schooldb;
2
3  SELECT * FROM VW_STUDENT_OVERVIEW_BY_HEADMASTER;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

STUDENT_ID	FIRST_NAME	LAST_NAME	GRADE_YEAR
68391	John	Cabling	7
68392	Rick	Garcia	8
68393	April	Ramos	7
68394	Alesa	Ramos	6
68395	Jerry	Bautista	9
68396	Sylvia	Velasco	10
68397	Gus	Perez	9
68398	Felix	Mercado	8
68399	Cindy	Antonio	7
68400	Alesa	Angeles	8
68401	Sherry	Ortiz	7
68402	Billy	Bernardo	9
68403	Theo	Luna	8
68404	Rick	Mateo	7
68405	Shannon	Bartolome	7
68406	Alesa	Ramos	9
68407	Jerry	Carpio	9
68408	Mark	Cervantes	10

VW_STUDENT_OVERVIEW_BY_...

Figure 4: Student Overview by Headmaster View

5.3. LOGICAL DESIGN

We'll need to normalize the data from the conceptual design to get it to at least 3NF. To get to at least 3NF, we need to get it to 1NF, which is to remove any multivalued data. But first, we need to create a primary key for each entity. Any attribute that has the word ID should be the key for each entity. However, STAFF_INFORMATION has two attributes that have ID in it: INSTRUCTOR_ID and ROLE_ID. Since this table for staff, we decided to use INSTRUCTOR_ID for the primary key.

Now that we figured out the primary keys for each entity, we remove STUDENT_ADDRESS from STUDENT_INFORMATION table and create a new table called STUDENT_ADDRESS using STUDENT_ID as the primary key. Furthermore, we create new attributes NUMBER, STREET_ADDRESS, CITY, STATE_OR_PROVINCE, COUNTRY, and ZIP_CODE for STUDENT_ADDRESS. The same goes for STAFF_INFORMATION, where the entities are the same as STUDENT_ADDRESS for the STAFF_ADDRESS table.

Now that the table is in 1NF, we'll need to get it to 2NF. However, it's already in 2NF because all entities are fully dependent on the primary key. However, we still need to get it to 3NF because there are some transitive dependencies that are found in our design.

To achieve 3NF, we need to separate DEPARTMENT_TITLE and DEPARTMENT_DESCRIPTION from COURSE_CATALOG. While DEPARTMENT_TITLE is dependent on COURSE_ID, DEPARTMENT_DESCRIPTION is dependent on DEPARTMENT_TITLE even though DEPARTMENT_TITLE is not a key. We create a new table called

DEPARTMENT and give it a primary key called DEPARTMENT_ID. DEPARTMENT_ID also becomes a foreign key for the table COURSE_CATALOG.

Below are the normalized primary keys, foreign keys, and the ERD of the logical design:

STUDENT_INFORMATION = STUDENT_ID (PK), LAST_NAME, FIRST_NAME, DATE_OF_BIRTH, ADMITTED_DATE, SEX

STUDENT_ADDRESS = STUDENT_ID (PK), NUMBER (PK), STREET_ADDRESS (PK), CITY (PK), STATE_OR_PROVINCE (PK), COUNTRY (PK), ZIP_CODE (PK)

STUDENT_GRADE_YEAR = STUDENT_ID (PK), GRADE_YEAR (PK)

STUDENT_COURSE = STUDENT_ID (PK), COURSE_ID (PK), TERM (PK), FINAL_GRADE

STUDENT_ATTENDANCE = STUDENT_ID (PK), DATE (PK), ABSENCE_ID (FK)

ABSENCE_TYPES = ABSENCE_ID (PK), ABSENCE_TITLE, DESCRIPTION

STAFF_INFORMATION = INSTRUCTOR_ID (PK), ROLE_ID (FK), LAST_NAME, FIRST_NAME, SSN, SALARY, PHONE_NUMBER, DATE_OF_BIRTH, ADMITTED_DATE

STAFF_ADDRESS = STAFF_ID (PK), NUMBER (PK), STREET_ADDRESS (PK), CITY (PK), STATE_OR_PROVINCE (PK), COUNTRY (PK), ZIP_CODE (PK)

STAFF_ROLE = ROLE_ID (PK), ROLE_TITLE, ROLE_DESCRIPTION

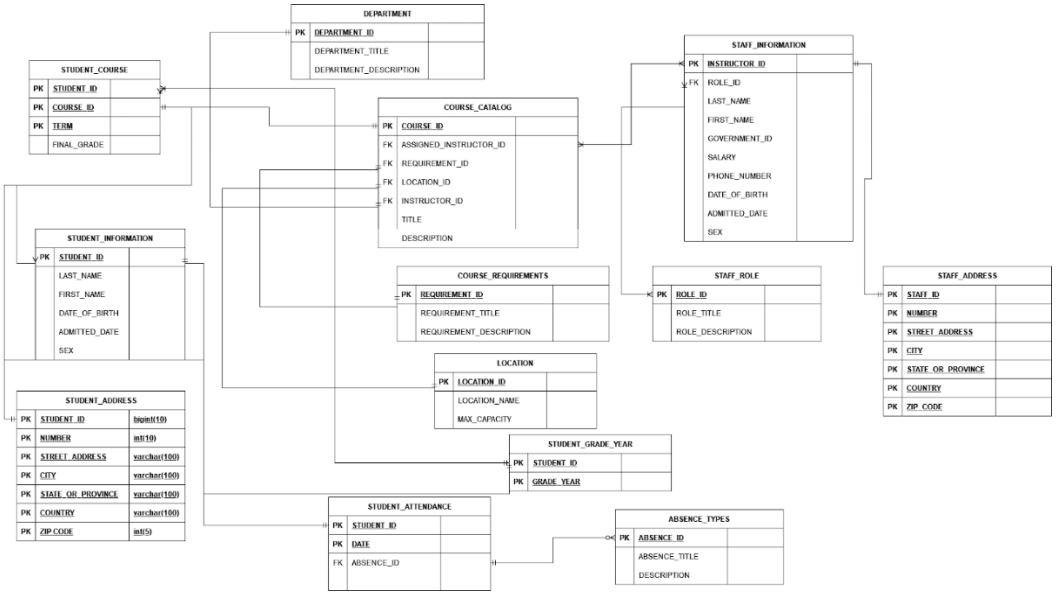
DEPARTMENT = DEPARTMENT_ID (PK), DEPARTMENT_TITLE, DEPARTMENT_DESCRIPTION

COURSE_CATALOG = COURSE_ID (PK), ASSIGNED_INSTRUCTOR_ID (FK), REQUIREMENT_ID (FK), LOCATION (FK), DEPARTMENT_ID (FK), TITLE, DESCRIPTION

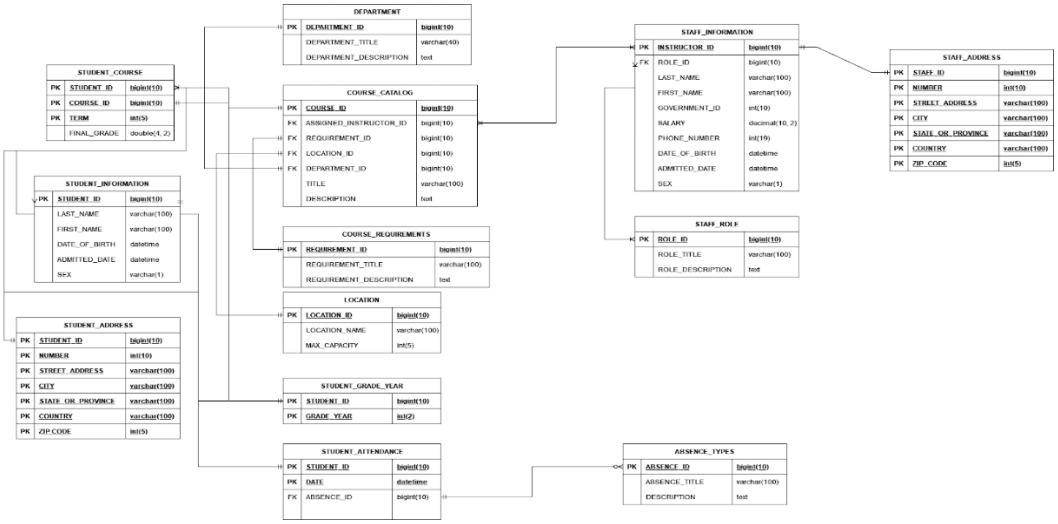
COURSE_REQUIREMENTS = REQUIREMENT_ID (PK), REQUIREMENT_TITLE, REQUIREMENT_DESCRIPTION

LOCATION = LOCATION_ID (PK), LOCATION_NAME, MAX_CAPACITY

Logical Design in 3NF:



ERD Diagram:



5.4. PHYSICAL DESIGN

The physical design's ERD is like the logical design, but we added the data types for all the entities. The primary key values are always an int, unique, and not null because they represent the relationship for the tables. Entities like DATE_OF_BIRTH and ADMITTED_DATE for STUDENT_INFORMATION

and STAFF_INFORMATION table are represented by DATETIME. The rest is self-explanatory.

We also added a check constraint to check whether SEX for STUDENT_INFORMATION and STAFF_INFORMATION is M or F.

As for the data to insert, the views below show the data that was inserted for each table. Each table has a minimum of six entries. All the commands to create the tables and insert the data are done with MySQL.

Below are some of the many tables we have populated.

Student Information Table:

STUDENT_ID	LAST_NAME	FIRST_NAME	DATE_OF_BIRTH	ADMITTED_DATE	SEX
68391	Cabling	John	2006-10-03 00:00:00	2020-08-19 00:00:00	M
68392	Garcia	Rick	2007-08-12 00:00:00	2021-08-21 00:00:00	M
68393	Ramos	April	2006-04-03 00:00:00	2019-08-18 00:00:00	F
68394	Ramos	Alesa	2008-02-25 00:00:00	2021-08-19 00:00:00	F
68395	Bautista	Jerry	2006-07-15 00:00:00	2019-08-19 00:00:00	M
68396	Velasco	Sylvia	2008-05-19 00:00:00	2022-02-19 00:00:00	F
68397	Perez	Gus	2006-11-30 00:00:00	2020-08-21 00:00:00	M
68398	Mercado	Felix	2007-08-26 00:00:00	2021-08-20 00:00:00	M
68399	Antonio	Cindy	2008-07-31 00:00:00	2022-08-18 00:00:00	F
68400	Angeles	Alesa	2007-09-07 00:00:00	2021-08-20 00:00:00	F
68401	Ortiz	Sherry	2008-06-03 00:00:00	2022-08-19 00:00:00	F
68402	Bernardo	Billy	2005-01-11 00:00:00	2021-02-19 00:00:00	M
68403	Luna	Theo	2007-10-25 00:00:00	2020-08-21 00:00:00	M
68404	Mateo	Rick	2007-04-20 00:00:00	2021-08-20 00:00:00	M
68405	Bartolome	Shannon	2008-03-23 00:00:00	2022-08-19 00:00:00	F
68406	Ramos	Alesa	2006-08-07 00:00:00	2021-08-19 00:00:00	F
68407	Carpio	Jerry	2007-12-04 00:00:00	2021-08-20 00:00:00	M
68408	Cervantes	Mark	2007-04-08 00:00:00	2022-02-19 00:00:00	M
68409	Malinao	Janice	2008-06-16 00:00:00	2022-08-19 00:00:00	F
68410	Benitez	Megan	2006-08-15 00:00:00	2020-08-21 00:00:00	F

Student Address Table:

STUDENT_ID	NUMBER	STREET_ADDRESS	CITY	STATE_OR_PROVINCE	COUNTRY	ZIP_CODE
68391	4504	17 N. Domingo Street	Quezon City	Metro Manila	Philippines	1100
68392	4505	21 Juan Luna Street	San Juan	Metro Manila	Philippines	1100
68393	4506	35 Teodora Alonzo Street	Manila	Metro Manila	Philippines	1100
68394	4507	51 Antonio Villegas Street	Manila	Metro Manila	Philippines	1100
68395	4508	29 N. Domingo Street	Quezon City	Metro Manila	Philippines	1100
68396	4509	47 Antonio Villegas Street	Manila	Metro Manila	Philippines	1100
68397	6379	229 Maharlika Hwy	Cabanatuan	Central Luzon	Philippines	3100
68398	6380	7866 Quezon Ave	Quezon City	Metro Manila	Philippines	1104
68399	6381	7186 Macabulos Dr	Tarlac City	Central Luzon	Philippines	2300
68400	6382	1389 Diosdado Macapagal Blvd	Paranaque	Metro Manila	Philippines	1702
68401	6383	7352 Gomez St	Tacloban	Eastern Visayas	Philippines	6500
68402	6384	59 Visayas Ave	Quezon City	Metro Manila	Philippines	1128
68403	6385	6433 Sabayle St	Tiligan	Northern Mindanao	Philippines	9200
68404	6386	3535 Topaz Rd	Pasig	Metro Manila	Philippines	1605
68405	6387	9643 Rizal St	Davao City	Davao Region	Philippines	8016
68406	6388	1298 C. Bautista St	Marikina	Metro Manila	Philippines	1807
68407	6389	504 Bustos St	Manila	Metro Manila	Philippines	1001
68408	6340	7495 Justiniano R. Borja St	Cagayan de...	Northern Mindanao	Philippines	9000
68409	6341	6794 Hyacinth St	Cebu City	Central Visayas	Philippines	6045

Course Catalog Table:

COURSE_ID	ASSIGNED_INSTRUCTOR_ID	REQUIREMENT_ID	LOCATION_ID	DEPARTMENT_ID	TITLE	DESCRIPTION
1	10001	54001	30627	30005	Math Gr. 7	Math for 7th grade.
2	10002	54001	30628	30006	Science Gr. 7	Science for 7th grade.
3	10003	54001	30629	30007	History Gr. 7	History for 7th grade.
4	10004	54002	30630	30005	Math Gr. 8	Math for 8th grade.
5	10005	54002	30631	30006	Science Gr. 8	Science for 8th grade.
6	10006	54002	30632	30007	History Gr. 8	History for 8th grade.
7	10007	54003	30627	30005	Math Gr. 9	Math for 9th grade.
8	10008	54003	30628	30006	Science Gr. 9	Science for 9th grade.
9	10009	54003	30629	30007	History Gr. 9	History for 9th grade.
10	10010	54004	30630	30005	Math Gr. 10	Math for 10th grade.
11	10011	54004	30631	30006	Science Gr. 10	Science for 10th grade.
12	10012	54004	30632	30007	History Gr. 10	History for 10th grade.

Department Table:

DEPARTMENT_ID	DEPARTMENT_TITLE	DEPARTMENT_DESCRIPTION
30005	Mathematics	Learning math fundamental and calculations
30006	Science	Learning biology, physcis, chemistry, etc.
30007	History	Learning history of the country and world
30008	English Literature	Reading books and analyzing them
30009	Art	Learning how to create art
30010	Music	Learning how to make music

Staff Role Table:

ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
83745	7th grade Math Teacher	Teacher for 7th grade math
83746	7th grade Science Teacher	Teacher for 7th grade math
83747	7th grade History Teacher	Teacher for 7th grade math
83748	8th grade Math Teacher	Teacher for 8th grade math
83749	8th grade Science Teacher	Teacher for 8th grade math
83750	8th grade History Teacher	Teacher for 8th grade math
83751	Headmaster	Headmaster
83752	Janitor	Daytime janitor
83753	Advisor	advisor for 8th grade students with last names ...
83754	9th grade Math Teacher	Teacher for 9th grade math
83755	9th grade Science Teacher	Teacher for 9th grade math
83756	9th grade History Teacher	Teacher for 9th grade math
83757	10th grade Math Teacher	Teacher for 10th grade math
83758	10th grade Science Teacher	Teacher for 10th grade math
83759	10th grade History Teacher	Teacher for 10th grade math
NULL	NULL	NULL

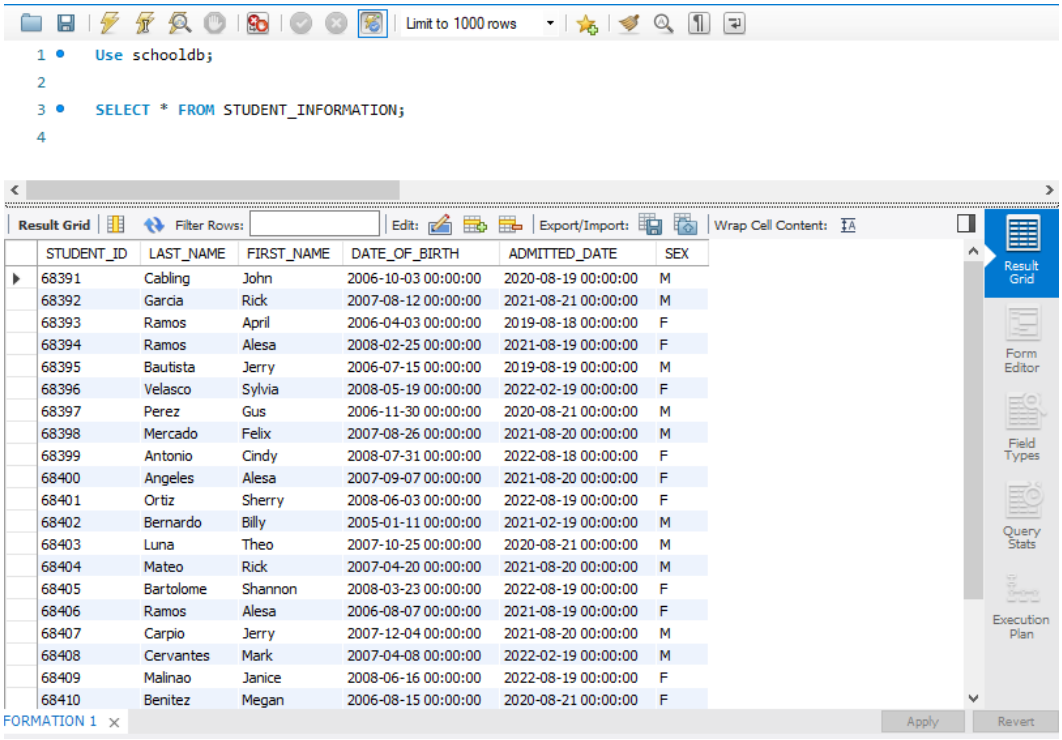
VI. IMPLEMENTATION AND TEST REPORT

6.1. TESTING METHOD

By populating the tables with more students, staff, and courses, we can move on to the testing process. The purpose of this is to see if all the queried data returns all inserted data as intended. By using the SQL command SELECT FROM (Table), we can return all the data from the table.

6.2. POPULATED TABLES

Student Information table:



This table shows the information of students for a school.

Student Grade Year table:

```
1  Use schooldb;
2
3  SELECT * FROM STUDENT_GRADE_YEAR;
4
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

STUDENT_ID	GRADE_YEAR
68391	7
68392	8
68393	7
68394	6
68395	9
68396	10
68397	9
68398	8
68399	7
68400	8
68401	7
68402	9
68403	8
68404	7
68405	7
68406	9
68407	9
68408	10
68409	8
68410	9

STUDENT_GRADE_YEAR 2 | Apply | Revert

This table shows the grade year of the students.

Student Address table:

```
1  Use schooldb;
2
3  SELECT * FROM STUDENT_ADDRESS;
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

STUDENT_ID	NUMBER	STREET_ADDRESS	CITY	STATE_OR_PROVINCE	COUNTRY	ZIP_CODE
68391	4504	17 N. Domingo Street	Quezon City	Metro Manila	Philippines	1100
68392	4505	21 Juan Luna Street	San Juan	Metro Manila	Philippines	1100
68393	4506	35 Teodora Alonzo Street	Manila	Metro Manila	Philippines	1100
68394	4507	51 Antonio Villegas Street	Manila	Metro Manila	Philippines	1100
68395	4508	29 N. Domingo Street	Quezon City	Metro Manila	Philippines	1100
68396	4509	47 Antonio Villegas Street	Manila	Metro Manila	Philippines	1100
68397	6379	229 Maharlika Hwy	Cabanatuan	Central Luzon	Philippines	3100
68398	6380	7866 Quezon Ave	Quezon City	Metro Manila	Philippines	1104
68399	6381	7186 Macabulos Dr	Tarlac City	Central Luzon	Philippines	2300
68400	6382	1389 Diosdado Macapagal Blvd	Paranaque	Metro Manila	Philippines	1702
68401	6383	7352 Gomez St	Tacloban	Eastern Visayas	Philippines	6500
68402	6384	59 Visayas Ave	Quezon City	Metro Manila	Philippines	1128
68403	6385	6433 Sabayle St	TIligan	Northern Mindanao	Philippines	9200
68404	6386	3535 Topaz Rd	Pasig	Metro Manila	Philippines	1605
68405	6387	9643 Rizal St	Davao City	Davao Region	Philippines	8016
68406	6388	1298 C. Bautista St	Marikina	Metro Manila	Philippines	1807
68407	6389	504 Bustos St	Manila	Metro Manila	Philippines	1001
68408	6340	7495 Justiniano R. Borja St	Cagayan de...	Northern Mindanao	Philippines	9000
68409	6341	6794 Hyacinth St	Cebu City	Central Visayas	Philippines	6045

STUDENT_ADDRESS 5 | Apply | Revert

This table shows the address of the students.

Location table:

```
1  Use schooldb;
2
3  SELECT * FROM LOCATION;
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

LOCATION_ID	LOCATION_NAME	MAX_CAPACITY
30627	Room D-01	35
30628	Room D-07	35
30629	Room C-03	35
30630	Room B-13	35
30631	Room B-12	35
30632	Room A-05	35
30640	Room D-03	35
30641	A-03	20
30642	Room E-03	35
30643	Room A-12	35
30644	Large Gym	100
30645	Room E-01	35
NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

LOCATION 6 | Apply | Revert

These are classroom locations showing the maximum number of students allowed.

Course Catalog table:

Query 1 | Limit to 1000 rows

```
1  Use schooldb;
2
3  SELECT * FROM COURSE_CATALOG;
4
5
6
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

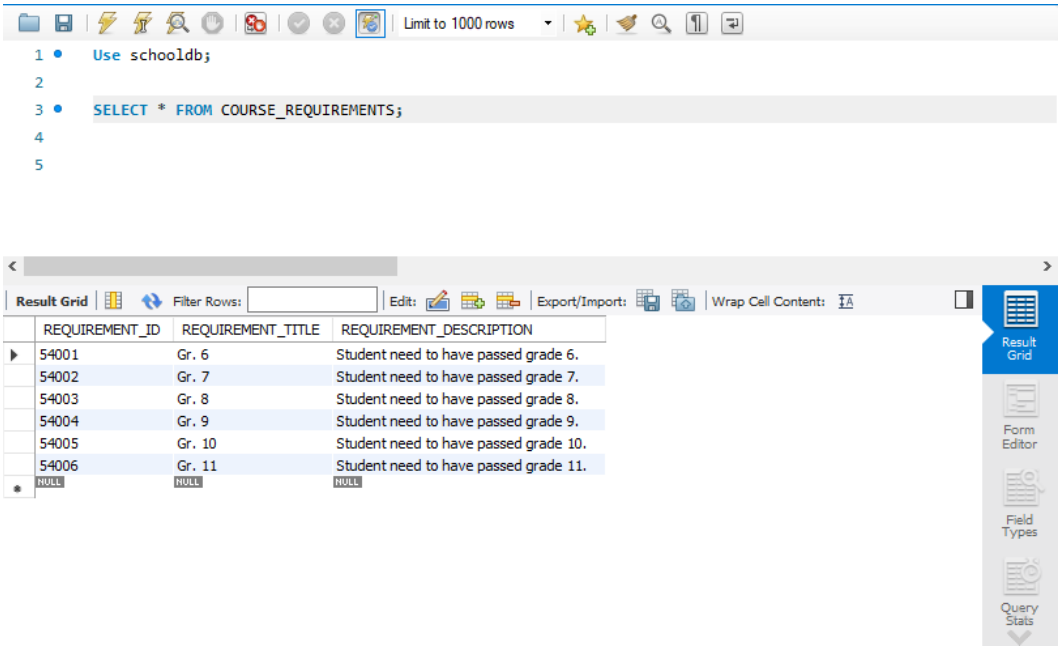
COURSE_ID	ASSIGNED_INSTRUCTOR_ID	REQUIREMENT_ID	LOCATION_ID	DEPARTMENT_ID	TITLE	DESCRIPTION
1	10001	54001	30627	30005	Math Gr. 7	Math for 7th grade.
2	10002	54001	30628	30006	Science Gr. 7	Science for 7th grade.
3	10003	54001	30629	30007	History Gr. 7	History for 7th grade.
4	10004	54002	30630	30005	Math Gr. 8	Math for 8th grade.
5	10005	54002	30631	30006	Science Gr. 8	Science for 8th grade.
6	10006	54002	30632	30007	History Gr. 8	History for 8th grade.
7	10007	54003	30627	30005	Math Gr. 9	Math for 9th grade.
8	10008	54003	30628	30006	Science Gr. 9	Science for 9th grade.
9	10009	54003	30629	30007	History Gr. 9	History for 9th grade.
10	10010	54004	30630	30005	Math Gr. 10	Math for 10th grade.
11	10011	54004	30631	30006	Science Gr. 10	Science for 10th grade.
12	10012	54004	30632	30007	History Gr. 10	History for 10th grade.
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats

COURSE_CATALOG 33 | Apply | Revert

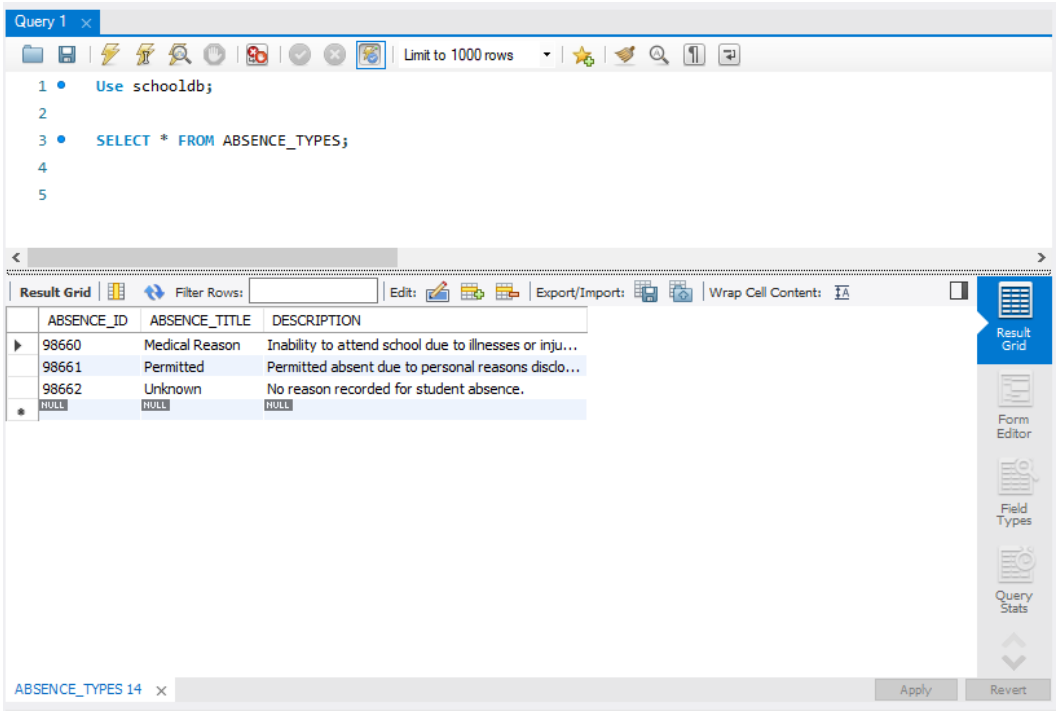
The course catalog shows the details of the course.

Course Requirements table:



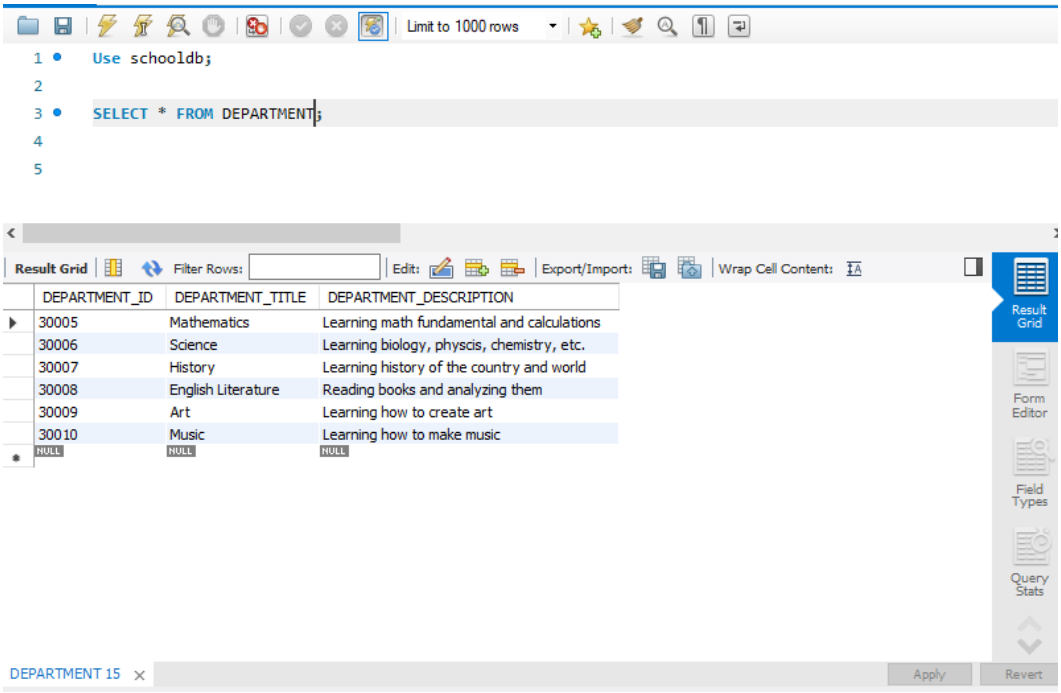
The course requirement table shows the requirements needed to take a course.

Absence Types table:



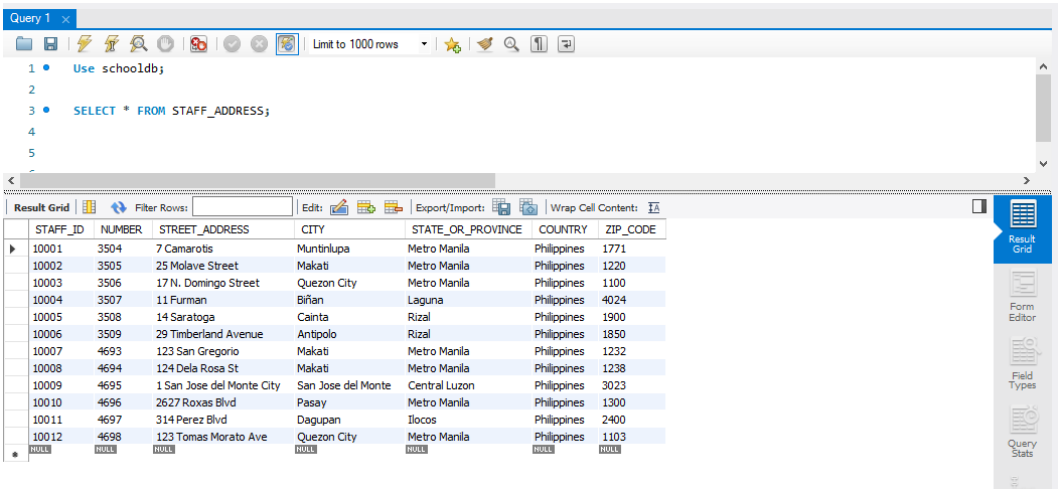
The shows the type of absence and the description of the absence.

Department table:



The department table shows their title and description.

Staff Address table:



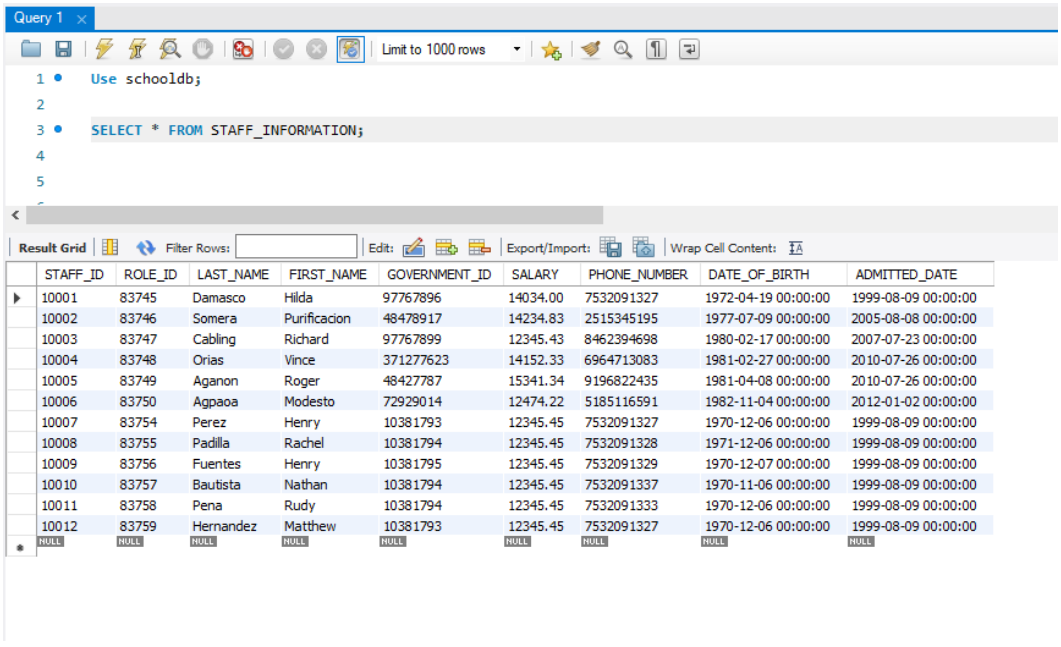
```
Use schooldb;

SELECT * FROM STAFF_ADDRESS;
```

STAFF_ID	NUMBER	STREET_ADDRESS	CITY	STATE_OR_PROVINCE	COUNTRY	ZIP_CODE
10001	3504	7 Camarotis	Muntinlupa	Metro Manila	Philippines	1771
10002	3505	25 Molave Street	Makati	Metro Manila	Philippines	1220
10003	3506	17 N. Domingo Street	Quezon City	Metro Manila	Philippines	1100
10004	3507	11 Furman	Biñan	Laguna	Philippines	4024
10005	3508	14 Saratoga	Cainta	Rizal	Philippines	1900
10006	3509	29 Timberland Avenue	Antipolo	Rizal	Philippines	1850
10007	4693	123 San Gregorio	Makati	Metro Manila	Philippines	1232
10008	4694	124 Dela Rosa St	Makati	Metro Manila	Philippines	1238
10009	4695	1 San Jose del Monte City	San Jose del Monte	Central Luzon	Philippines	3023
10010	4696	2627 Roxas Blvd	Pasay	Metro Manila	Philippines	1300
10011	4697	314 Perez Blvd	Dagupan	Ilocos	Philippines	2400
10012	4698	123 Tomas Morato Ave	Quezon City	Metro Manila	Philippines	1103

The staff address table shows the street address, city, state or province if they're outside the United States, country, and ZIP code. Depending on the country, the ZIP code may be different and not represent a 5-digit number.

Staff Information table:



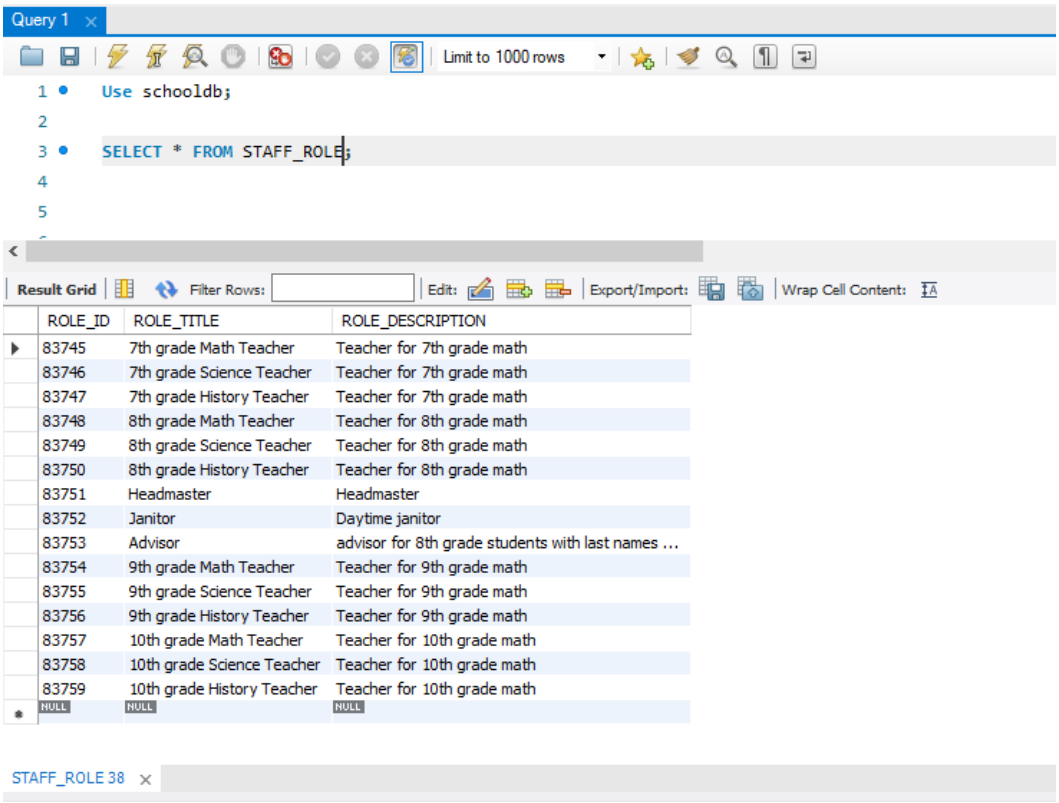
```
Use schooldb;

SELECT * FROM STAFF_INFORMATION;
```

STAFF_ID	ROLE_ID	LAST_NAME	FIRST_NAME	GOVERNMENT_ID	SALARY	PHONE_NUMBER	DATE_OF_BIRTH	ADMITTED_DATE
10001	83745	Damasco	Hilda	97767896	14034.00	7532091327	1972-04-19 00:00:00	1999-08-09 00:00:00
10002	83746	Somera	Purificacion	48478917	14234.83	2515345195	1977-07-09 00:00:00	2005-08-08 00:00:00
10003	83747	Cabling	Richard	97767899	12345.43	8462394698	1980-02-17 00:00:00	2007-07-23 00:00:00
10004	83748	Orias	Vince	371277623	14152.33	6964713083	1981-02-27 00:00:00	2010-07-26 00:00:00
10005	83749	Aganon	Roger	48427787	15341.34	9196822435	1981-04-08 00:00:00	2010-07-26 00:00:00
10006	83750	Agpaoa	Modesto	72929014	12474.22	5185116591	1982-11-04 00:00:00	2012-01-02 00:00:00
10007	83754	Perez	Henry	10381793	12345.45	7532091327	1970-12-06 00:00:00	1999-08-09 00:00:00
10008	83755	Padilla	Rachel	10381794	12345.45	7532091328	1971-12-06 00:00:00	1999-08-09 00:00:00
10009	83756	Fuentes	Henry	10381795	12345.45	7532091329	1970-12-07 00:00:00	1999-08-09 00:00:00
10010	83757	Bautista	Nathan	10381794	12345.45	7532091337	1970-11-06 00:00:00	1999-08-09 00:00:00
10011	83758	Pena	Rudy	10381794	12345.45	7532091333	1970-12-06 00:00:00	1999-08-09 00:00:00
10012	83759	Hernandez	Matthew	10381793	12345.45	7532091327	1970-12-06 00:00:00	1999-08-09 00:00:00

This shows the information of the staff. GOVERNMENT_ID is similar to SSN, but since this school is based around a developing country, we use GOVERNMENT_ID to represent residents of that country.

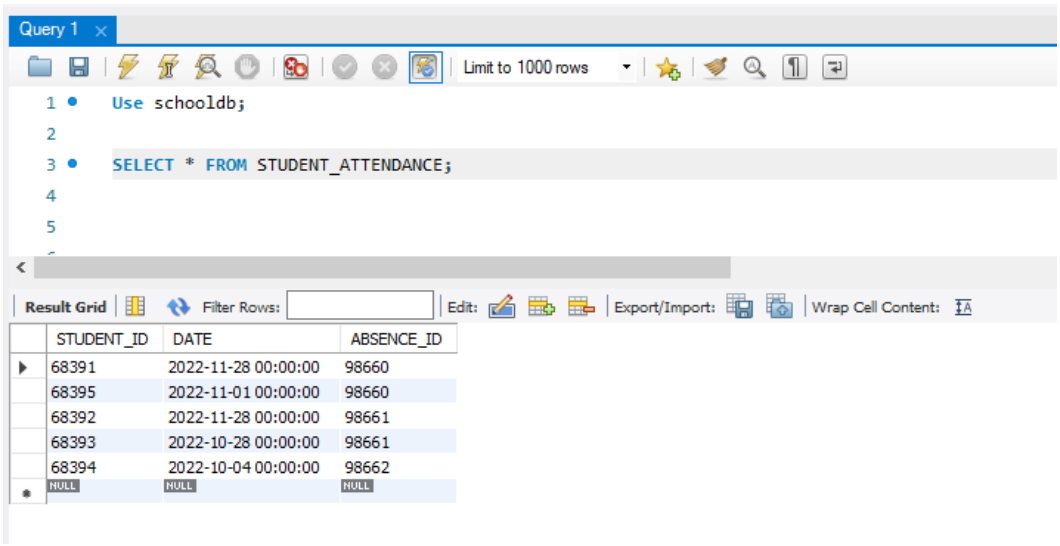
Staff Role table:



ROLE_ID	ROLE_TITLE	ROLE_DESCRIPTION
83745	7th grade Math Teacher	Teacher for 7th grade math
83746	7th grade Science Teacher	Teacher for 7th grade math
83747	7th grade History Teacher	Teacher for 7th grade math
83748	8th grade Math Teacher	Teacher for 8th grade math
83749	8th grade Science Teacher	Teacher for 8th grade math
83750	8th grade History Teacher	Teacher for 8th grade math
83751	Headmaster	Headmaster
83752	Janitor	Daytime janitor
83753	Advisor	advisor for 8th grade students with last names ...
83754	9th grade Math Teacher	Teacher for 9th grade math
83755	9th grade Science Teacher	Teacher for 9th grade math
83756	9th grade History Teacher	Teacher for 9th grade math
83757	10th grade Math Teacher	Teacher for 10th grade math
83758	10th grade Science Teacher	Teacher for 10th grade math
83759	10th grade History Teacher	Teacher for 10th grade math
NULL	NULL	NULL

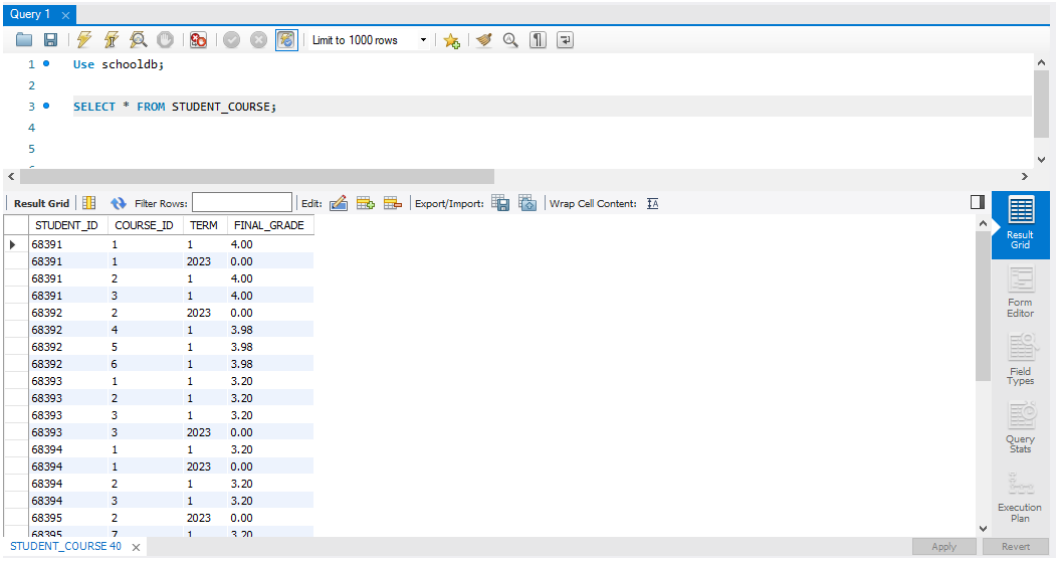
This shows the role of the staff and the description.

Student Attendance table:



This shows the attendance of students on the date given.

Student Course table:



This shows the courses the students took, the term taken and the final grade that they received.

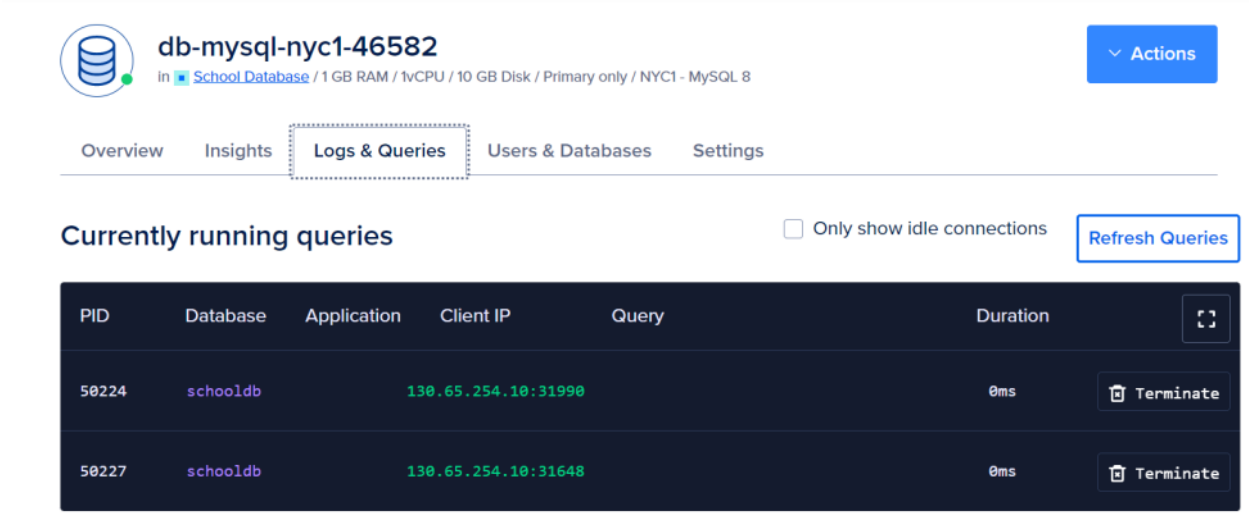
VII. DEPLOYMENT

Our database is deployed online on DigitalOcean. The following steps show how the school database is created:

1. Create Digital Ocean account.
2. Select “New Project”.
3. Name the new project and provide a brief description.
4. Select “Create Managed Database”.
5. Select the data center that will host the database live and the engine (MySQL).
6. Select the plan appropriate for the scale of the project.
7. Choose a unique database cluster name if desired.
8. Select “Create Database Cluster”.
9. Wait several minutes for the server to initialize and set everything up.
10. Collect the connection link and password that will be used to access the database from RDBMS.
11. Open client-based RDBMS (e.g., MySQL Workbench) and fill in the connection credentials (username, password, host link, port, and SSL mode).
12. Connect to the database and start creating tables.

To connect to an existing online database:

1. Open client-based RDBMS (e.g., MySQL Workbench) and fill in the connection credentials (username, password, host link, port, and SSL mode).
2. Connect to the database and start creating extra tables as necessary.
3. To entry data, use standard CLI on the RDBMS.



Connection details

Public network ✓ | VPC network | Connection parameters

```
username = doadmin
password = AVNS_f6M_o9Hb7o-_MbwAw2Z hide
host = db-mysql-nyc1-46582-do-user-12984535-
0.b.db.ondigitalocean.com
port = 25060
database = schooldb
sslmode = REQUIRED
```

User: doadmin | Database: schooldb

Copy | Download CA certificate

VIII. CONCLUSION

Our school database has been executed on DigitalOcean. We've explained the steps on how to implement the database through DigitalOcean in the previous step. The overall development from the conceptual design to the deployment went within our favor. If we have the credentials for the database, we can access the database remotely. Due to time constraints, we weren't able to create a GUI for the database.

Going through the design process, this relational database is a more effective way for institutes in developing countries to store data regarding their students, staff, courses, and other important data. Not only that, but the deployment of this relational database through DigitalOcean also provides a cleaner way access and back up data in the events of any natural disaster to occur. The steps to create and deploy the database may be daunting, but this is a far better way to store data and we hope institutes in developing countries will follow along.

IX. Appendix

Github Repository: https://github.com/Triparadox/SchoolDB

Live DB Host on DigitalOcean: db-mysql-nyc1-46582-do-user-12984535-0.b.db.ondigitalocean.com