

Ministry of Higher Education and Scientific Research
Northern Technical University
Technical Engineering College of Mosul
Cybersecurity and Cloud Computing Techniques Engineering

Database Systems

Submitted by:

مصطفى عبد العزيز حمزة

Introduction

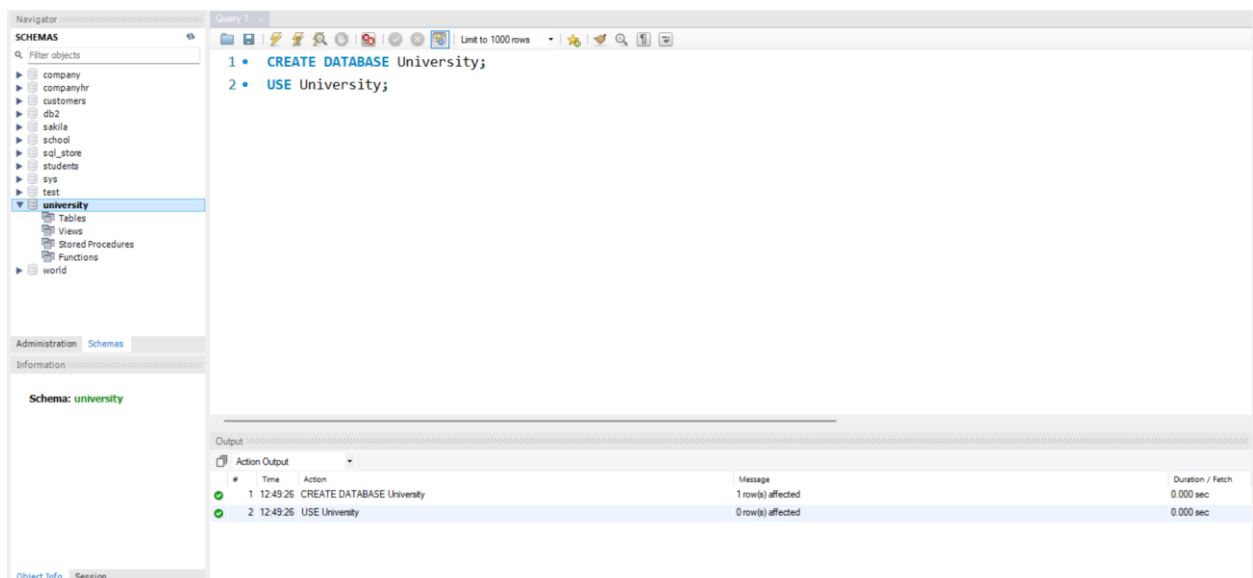
This report describes the creation and use of a database called University. The database contains three tables: Students, Teachers, and Degrees. The Students table stores information about students, such as their name, gender, age, email, and phone number. The Teachers table stores information about teachers, such as their name, gender, age, salary, email, and phone number. The Degrees table stores information about degrees, such as the student ID, teacher ID, and grades in five courses: Database Systems, Data Structures, Math, Cybersecurity, and Network Infrastructure.

The report also includes instructions on how to use SQL to query and update the database. For example, the report shows how to use the SELECT statement to get data from the tables, the UPDATE statement to update the data in the tables, and the DESCRIBE statement to get information about the structure of the tables.

The report concludes with a link to a GitHub repository containing the code, database dumps, and the report in PDF form.

Procedure

1- To create a database called University

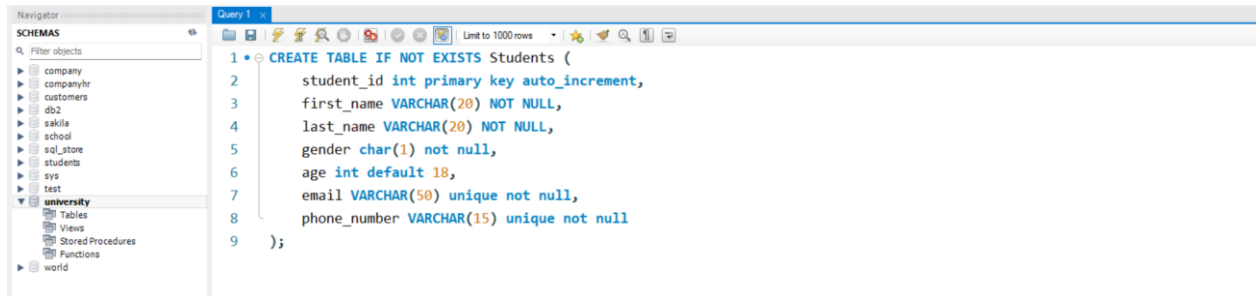


2- To create a table inside the database `University` called students, which includes student IDs, student names, gender, age, email, and phone number.

-AUTO_INCREMENT: Specifies that the ID will be automatically increased by one.

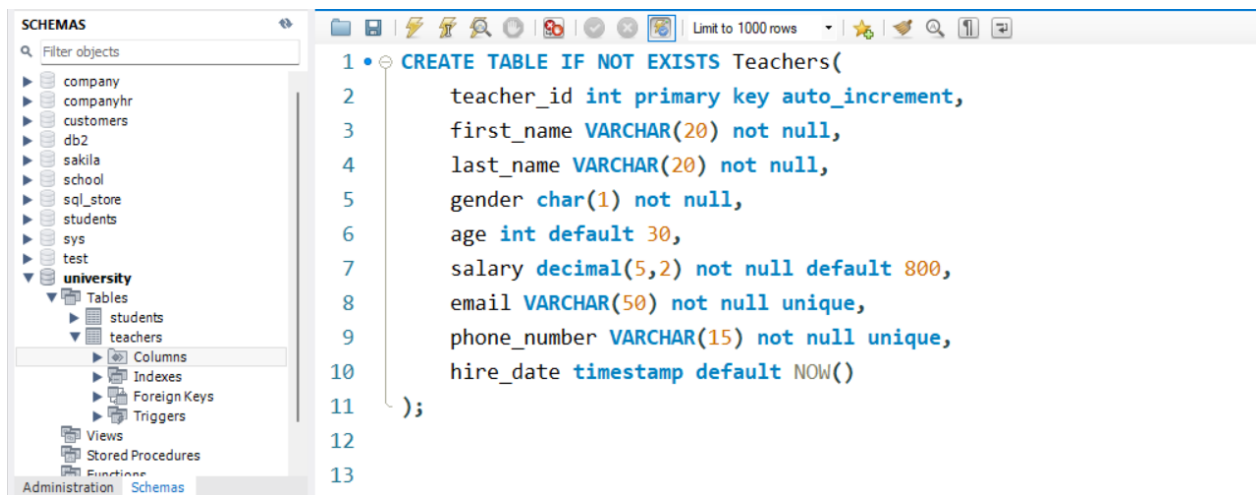
-NOT NULL specifies that all of the value can't be NULL, and email and phone number are unique values can't be repeated inside the table.

-UNIQUE specifies that the value is unique.



- 3- To create other table called teachers, which includes teacher ID, teacher name, gender, age, salary, email, phone number, and start date.

DEFAULT: Specifies a default for the column.

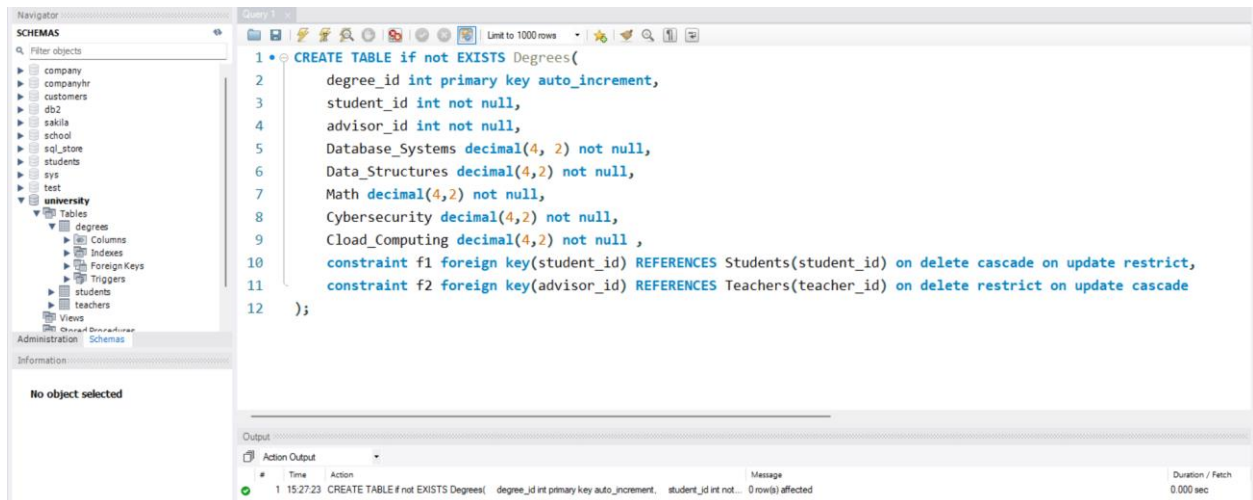


- 4- To create a table for degrees, which includes degree ID, student ID, teacher ID, and the degrees, with foreign key constraints to Students and Teachers tables.

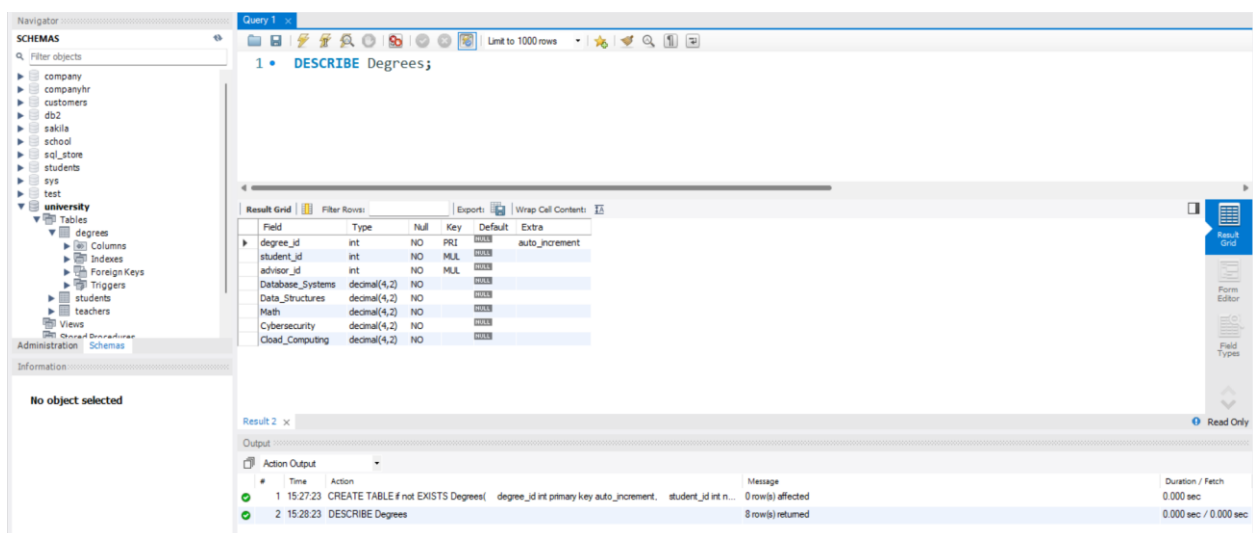
The constraint keyword is used to define the foreign key constraint. The foreign key keyword specifies that the `student_id` and `advisor_id` columns in the Degrees table are foreign keys that reference the `student_id` and `teacher_id` columns in the Students and Teachers tables, respectively.

ON DELETE and ON UPDATE specify what should happen if a row in the referenced table is deleted or updated.

- a. CASCADE: If the value of a column in the referenced table is changed, the corresponding values in the referencing table are also changed.
- b. RESTRICT: If the value of a column in the referenced table is changed, and there are any rows in the referencing table that reference that value, the change is prevented.



5- The DESCRIBE statement in SQL is used to describe the structure of a table. It provides information about the columns in the table, including the column name, data type, and constraints. The DESCRIBE statement can also be used to describe views and other database objects.



- 6- To alter the Degrees table to carry out the following alterations:
 - a. Drop the Cloud_Computing column.
 - b. Add the Network_Infrastructure column, which is a decimal data type with a precision of 4 and a scale of 2, and is not null.
 - c. Add a unique constraint called unique_IDs, which ensures that the combination of student_id and advisor_id is unique for each row in the table.

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane shows the 'university' schema with the 'degrees' table. The 'Table: degrees' information pane shows columns: degree_id (int, AI PK), student_id (int), advisor_id (int), Data_base (decimal(4,2)), Date_Structure (decimal(4,2)), Mathematics (decimal(4,2)), Cybersecurity (decimal(4,2)), and Network_Infrastructure (decimal(4,2)).

The main query window shows the following SQL statement:

```

1 ALTER TABLE Degrees
2 drop column Cloud_Computing,
3 add column Network_Infrastructure decimal(4,2) not null,
4 add constraint unique_IDs unique(student_id, advisor_id);

```

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	14:45:03	DESCRIBE Degrees	8 row(s) returned	0.000 sec / 0.000 sec
2	14:52:42	ALTER TABLE Degrees drop column: Cloud_Computing, add column Network_Infrastructure decimal(4,2) not null, add constraint unique_IDs unique(student_id, advisor_id);	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

- 7- We use the statement `INSERT INTO` to insert data into tables, as the following

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane shows the 'university' schema with the 'Students' table. The 'Table: Students' information pane shows columns: first_name, last_name, gender, age, email, phone_number.

The main query window shows the following SQL statement:

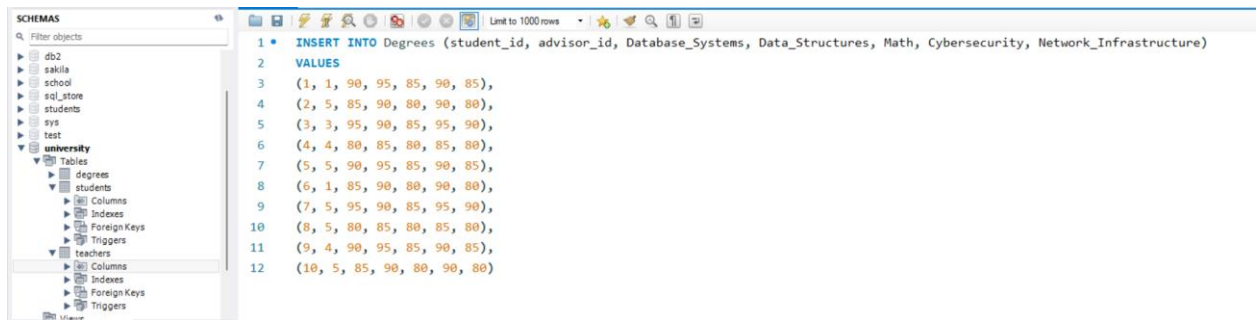
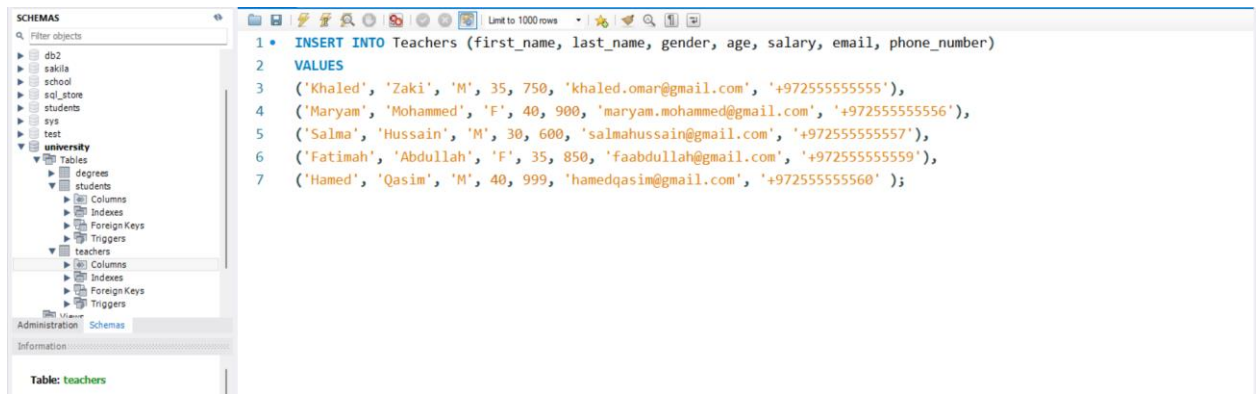
```

1 INSERT INTO Students (first_name, last_name, gender, age, email, phone_number)
2 VALUES
3 ('Ali', 'Ahmed', 'M', 18, 'ali.ahmed@gmail.com', '+97255555555'),
4 ('Omar', 'Mohammed', 'M', 19, 'omar.mohammed@gmail.com', '+97255555556'),
5 ('Fatima', 'Abdullah', 'F', 20, 'fatima.abdullah@gmail.com', '+97255555557'),
6 ('Khaled', 'Omar', 'M', 21, 'khaled.omar@gmail.com', '+97255555558'),
7 ('Sarah', 'Ali', 'F', 22, 'sara.ali@gmail.com', '+97255555559'),
8 ('Ahmed', 'Khaled', 'M', 23, 'ahmed.khaled@gmail.com', '+97255555560'),
9 ('Maryam', 'Omar', 'F', 24, 'maryam.omar@gmail.com', '+97255555561'),
10 ('Fatima', 'Salman', 'M', 25, 'fatimasalman@gmail.com', '+97255555562'),
11 ('Noura', 'Abdullah', 'F', 26, 'nora.abdullah@gmail.com', '+97255555563'),
12 ('Khaled', 'Mohammed', 'M', 27, 'khaled.mohammed@gmail.com', '+97255555564'),
13 ('Sarah', 'Salman', 'F', 28, 'sara.salman@gmail.com', '+97255555565');

```

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	15:27:23	CREATE TABLE IF NOT EXISTS Degrees(degree_id int primary key auto_increment, student_id int n...	0 row(s) affected	0.000 sec
2	15:28:23	DESCRIBE Degrees	8 row(s) returned	0.000 sec / 0.000 sec
3	15:31:36	INSERT INTO Students (first_name, last_name, gender, age, email, phone_number) VALUES ('Ali', 'Ahmed', 'M', 18, 'ali.ahmed@gmail.com', '+97255555555'), ('Omar', 'Mohammed', 'M', 19, 'omar.mohammed@gmail.com', '+97255555556'), ('Fatima', 'Abdullah', 'F', 20, 'fatima.abdullah@gmail.com', '+97255555557'), ('Khaled', 'Omar', 'M', 21, 'khaled.omar@gmail.com', '+97255555558'), ('Sarah', 'Ali', 'F', 22, 'sara.ali@gmail.com', '+97255555559'), ('Ahmed', 'Khaled', 'M', 23, 'ahmed.khaled@gmail.com', '+97255555560'), ('Maryam', 'Omar', 'F', 24, 'maryam.omar@gmail.com', '+97255555561'), ('Fatima', 'Salman', 'M', 25, 'fatimasalman@gmail.com', '+97255555562'), ('Noura', 'Abdullah', 'F', 26, 'nora.abdullah@gmail.com', '+97255555563'), ('Khaled', 'Mohammed', 'M', 27, 'khaled.mohammed@gmail.com', '+97255555564'), ('Sarah', 'Salman', 'F', 28, 'sara.salman@gmail.com', '+97255555565');	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.015 sec

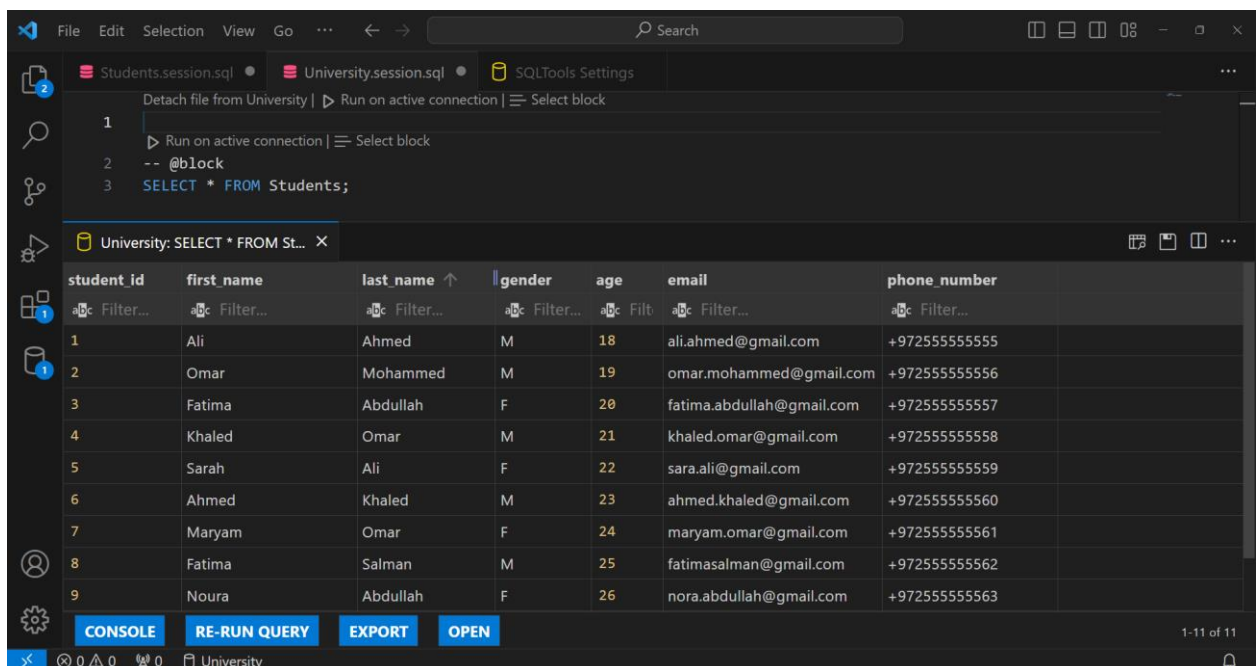


Note: All of the inserted names and data are AI generated.

8- We can use the `SELECT` statement to get data from tables, for examples:

a. To get all data from `Students` table:

SELECT * FROM Students;



- b. To get the data of Female students only:
SELECT * FROM students where gender = 'F';

The screenshot shows the SQLTools interface with a query editor and a results table. The query editor contains the following SQL code:

```
1 Run on active connection | Select block
2 -- @block
3 SELECT * from students where gender = 'F';
```

The results table, titled "University: SELECT * from st...", displays the following data:

student_id	first_name	last_name	gender	age	email	phone_number
3	Fatima	Abdullah	F	20	fatima.abdullah@gmail.com	+97255555557
5	Sarah	Ali	F	22	sara.ali@gmail.com	+97255555559
7	Maryam	Omar	F	24	maryam.omar@gmail.com	+97255555561
9	Noura	Abdullah	F	26	nora.abdullah@gmail.com	+97255555563
11	Sarah	Salman	F	28	sara.salman@gmail.com	+97255555565

- c. To get all rows from the `Teachers` table, and order them in ascending order by the salary column.
SELECT * from teachers ORDER BY salary ASC;

The screenshot shows the SQLTools interface with a query editor and a results table. The query editor contains the following SQL code:

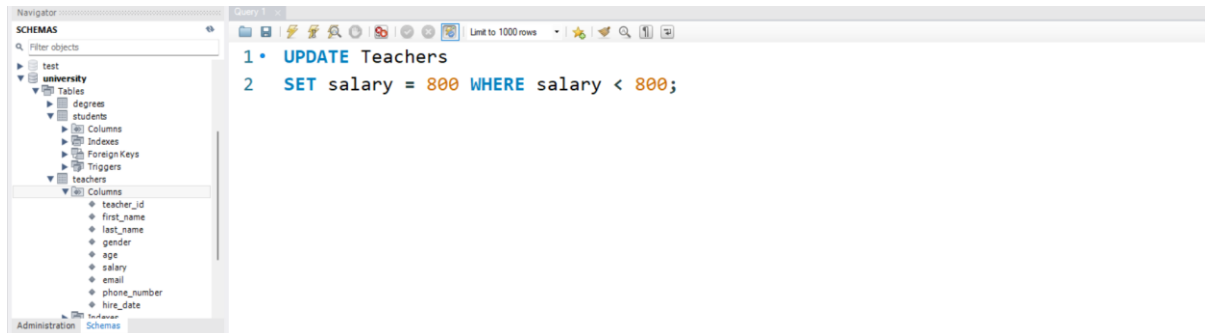
```
1 Run on active connection | Select block
2 -- @block
3 SELECT * from teachers ORDER BY salary ASC;
```

The results table, titled "University: SELECT * from te...", displays the following data:

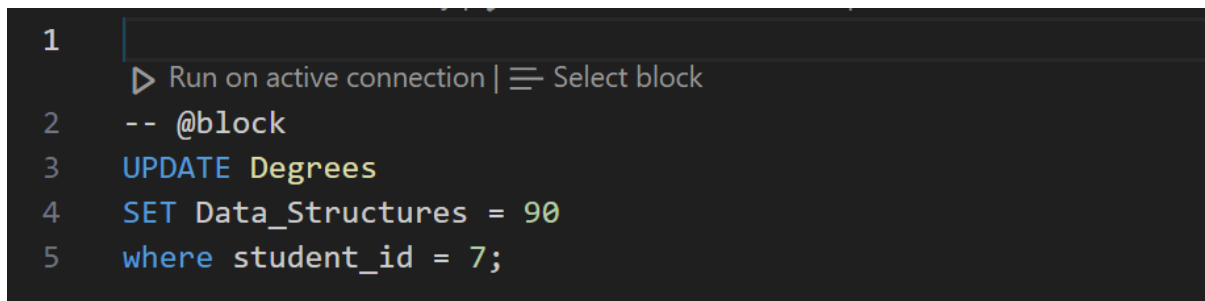
teacher_id	first_name	last_name	gender	age	salary	email	phone_number
3	Salma	Hussain	M	30	600.00	salmahussain@gmail.com	+97255555557
1	Khaled	Zaki	M	35	750.00	khaled.omar@gmail.com	+97255555555
4	Fatimah	Abdullah	F	35	850.00	faabdullah@gmail.com	+97255555559
2	Maryam	Mohammed	F	40	900.00	maryam.mohammed@gmail.com	+97255555556
5	Hamed	Qasim	M	40	999.00	hamedqasim@gmail.com	+97255555560

9- We use the `UPDATE` statement to update the content of a table.

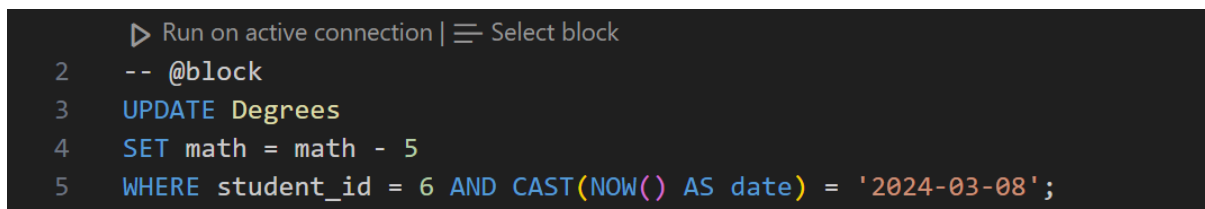
a. To update the salary to 800 if it is less than 800:



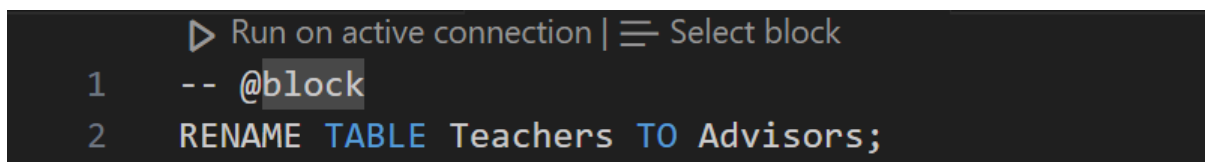
b. To set the degree of Maryam Omar in Data Structures to 95



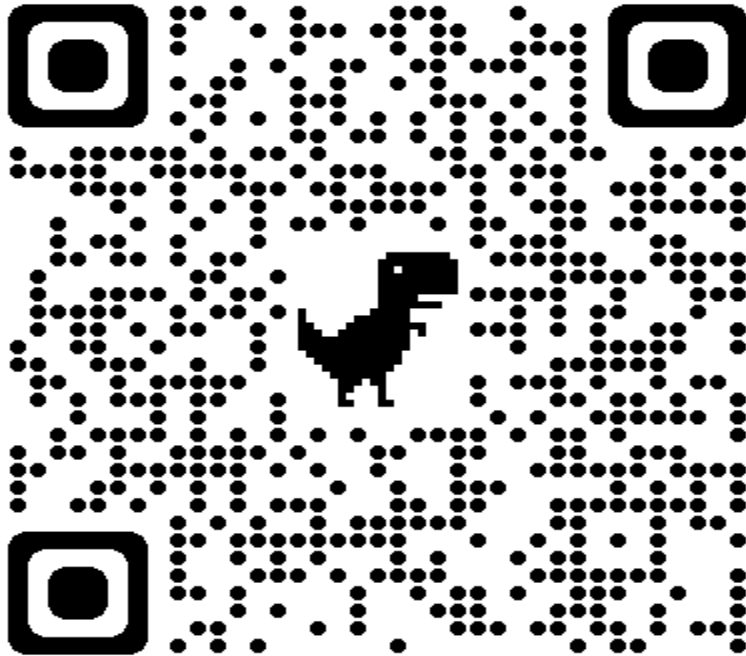
c. To minus Ahmed Khaled's degree by 5 in Math because it is March 8th!



10- To rename `Teachers` table:



Note: You can access the Github repository of the report, which includes the code, database dumps, and the report in PDF form from the QR code or the link below:



Link: <https://github.com/MustafaAbdulazizHamza/Report-01--Database-Systems>