NAME		INDEX NUMBER
NUERTEY OBED	-	10908266
AFIA DABANKA AKYAA	-	10908786
QUARSHIE FRANCIS	-	10909989
QUARTSON JERRY JUSTICE	-	10885857

CPEN 207 INTRODUCTION TO SOFTWARE ENGINEERING

LECTURER'S NAME: MR. JOHN ASSIAMAH

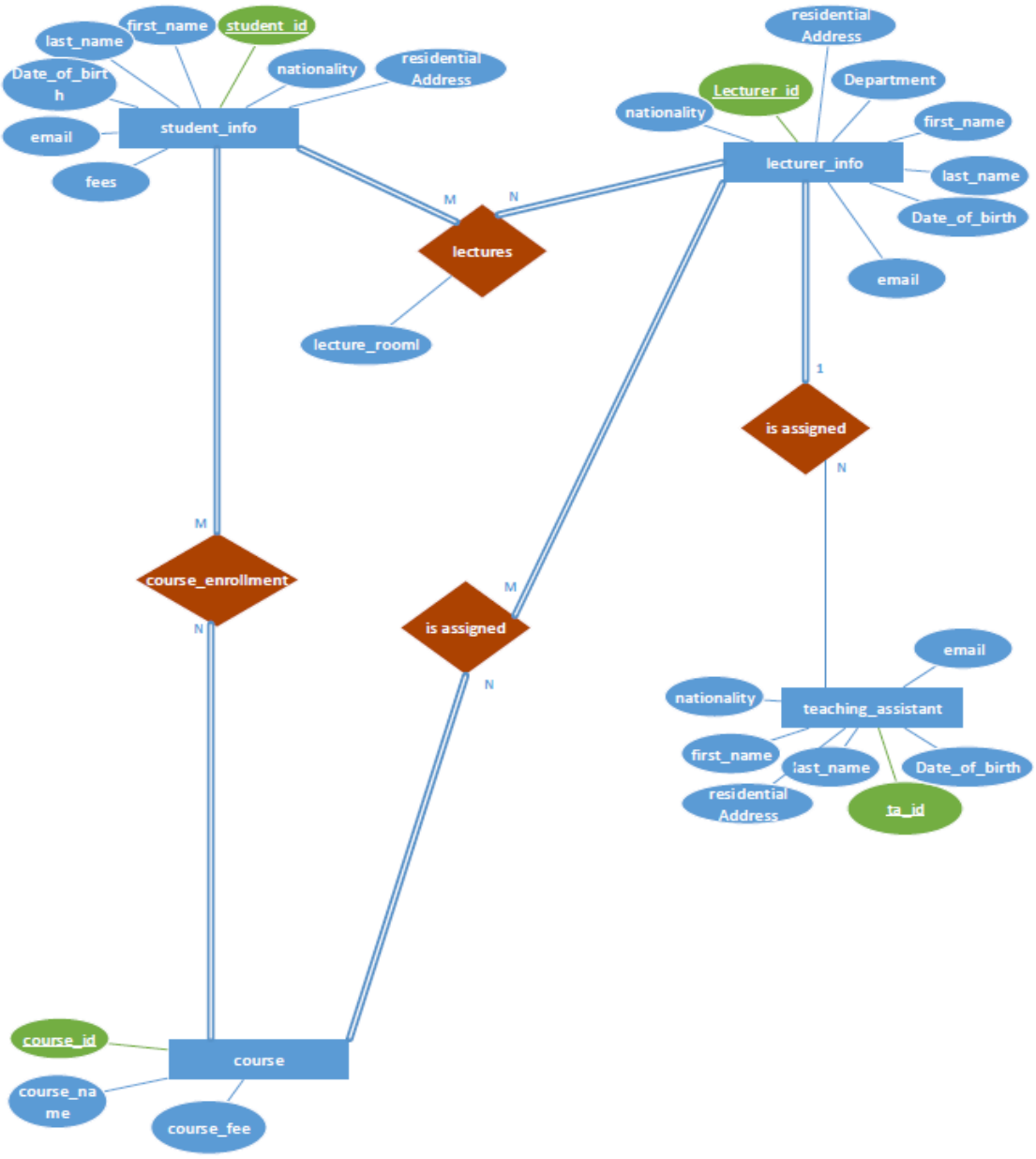
DATE: 5th MAY, 2022

TITLE: PROJECT 2

TABLE OF CONTENT

POSTGRES DATABASE FOR COMPUTER ENGINEERING DEPARTMENT

EER SCHEMA:



RELATIONAL SCHEMA;

SQL CODE FOR CREATING TABLES;

ONE-TO-MANY IMPLEMENTATION BETWEEN “lecturer_info” AND “teaching_assistant_info”

```
create table teaching_assistant_info (
    ta_id BIGSERIAL NOT NULL PRIMARY KEY,
    lecturer_info_id INT NOT NULL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    date_of_birth DATE NOT NULL,
    residential_Address VARCHAR(50) NOT NULL,
    nationality VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE,
    FOREIGN KEY(lecturer_info_id) REFERENCES lecturer_info(lecturer_id) ON DELETE CASCADE
);
```

```
create table student_info (
    student_id VARCHAR(50) NOT NULL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    date_of_birth DATE NOT NULL,
    nationality VARCHAR(50) NOT NULL,
    residential_Address VARCHAR(200),
    email VARCHAR(50) UNIQUE,
    fees float(2)
);
```

MANY-TO-MANY IMPLEMENTATION OF LECTURES BETWEEN “student_info” AND “lecturer_info”

```
create table lectures (
    student_info_id VARCHAR(50) NOT NULL,
    lecturer_info_id INT NOT NULL,
    lecture_room VARCHAR(100) NOT NULL,
    PRIMARY KEY(student_info_id, lecturer_info_id),
    FOREIGN KEY(student_info_id) REFERENCES student_info(student_id) ON DELETE CASCADE,
    FOREIGN KEY(lecturer_info_id) REFERENCES lecturer_info(lecturer_id) ON DELETE CASCADE

);
```

```sql
create table course (
    course_id BIGSERIAL NOT NULL PRIMARY KEY,
    course_name VARCHAR(50) NOT NULL,
    course_fee float(2) NOT NULL
);
```

```sql
create table lecturer_info (
    lecturer_id BIGSERIAL NOT NULL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    date_of_birth DATE NOT NULL,
    nationality VARCHAR(50) NOT NULL,
    residential_Address VARCHAR(50),
    email VARCHAR(50) UNIQUE
);
```

MANY-TO-MANY RELATIONSHIP IMPLEMENTATION BETWEEN “lecturer_info” AND “course_info”

```sql
CREATE TABLE lecturerToCourse_assignment(
    lecturer_info_id INT NOT NULL,
    course_info_id INT NOT NULL,
    PRIMARY KEY(lecturer_info_id, course_info_id),
    FOREIGN KEY(lecturer_info_id) REFERENCES lecturer_info(lecturer_id) ON UPDATE CASCADE,
    FOREIGN KEY(course_info_id) REFERENCES course(course_id) ON UPDATE CASCADE
);
```

MANY-TO-MANY RELATIONSHIP IMPLEMENTATION BETWEEN “course_info” AND “student_info”

```sql
CREATE TABLE course_enroll(
    course_info_id INT NOT NULL,
    student_info_id VARCHAR(50) NOT NULL,
    PRIMARY KEY(course_info_id, student_info_id),
    FOREIGN KEY(course_info_id) REFERENCES course(course_id) ON DELETE CASCADE,
    FOREIGN KEY(student_info_id) REFERENCES student_info(student_id) ON DELETE CASCADE
);
```

SQL CODE FOR CREATING FUNCTION FOR OUTSTANDING FEES;

```
CREATE OR REPLACE FUNCTION outstanding_fees()
returns JSON
AS
$$
declare
    calcFees JSON;

begin
    SELECT array_to_json(array_agg(row_to_json(student_outstanding_fees)))
    FROM (SELECT student_info.student_id AS INDEXNUMBER, student_info.first_name, student_info.last_name, (student_info.
    fees - SUM(course.course_fee)) AS OUTSTANDING_FEES into calcFees FROM course, course_enroll, student_info
        WHERE student_info.student_id = course_enroll.student_info_id AND course_enroll.course_info_id = course.course_id
        GROUP BY student_info.student_id, student_info.first_name, student_info.last_name, student_info.fees
        ) student_outstanding_fees;
    return calcFees;

end;
$$ language plpgsql;

--TO CALL FUNCTION WE USE "  SELECT outstanding_fees() " returns JSON ARRAY
```

NOTE:

All source code are included in the directory ".\group2_project2\ codeUsedToCreate_The_database"

Backup of the database ".\group2_project2\database_backup\softwareProject2_database.sql"

FLUTTER APPLICATION

INTRODUCTION

Flutter is an open-source User Interface SDK that is Software Development Kit. Flutter is an open-source project, and it is maintained by Google. For this project, the login ,sign up page and dashboard was built using flutter. The dashboard contains a to do list. The to do list is also implemented by the rest api.

GOAL

Its tasks could range from registration of new users to authenticating registered users to access the app and everything else needed to keep the app running.

METHODOLOGY

This project uses Dart programming language for creating all the pages. Also, Firebase was used for the authenticating system.

After the application was built, it was given to some selected users to test its functionalities.

Source code: https://github.com/obedNuertey1/group2_project2