LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC4057NI Introduction to Information Systems

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2021 Spring

Student Name: Sarthak Bikram Rana

Group: N1

London Met ID:

College ID: NP01NT4S210129

Assignment Due Date: April 30,2021

Assignment Submission Date: April 29,2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Database

Database are the gathering of logically associated records in a scientific way that is stored in an prepared way to fulfill the necessities of various customers of an enterprise that may be without difficulty accessed, managed and updated. Data is an illustration of facts, standards or commands in a formalized manner, that is appropriate for communication, interpretation or processing with the aid of using human or digital machine. Data is represented with the assist of characters which include alphabets, digits or unique characters. Whereas, Information can be defined as when data is processed, organized, structured or presented in a given context so it is made useful.

1.2 Description of Organization

Talking about the business of the student enrollment system, it is the process to arrange all the students in the specific classes according to their chosen courses while taking admission in which every course contains different subjects and each subject has their own lecturer issued to. It is needed for every school, college and universities while enrolling students at the start of the semester. We can take example of Islington College which has developed its own enrollment system with its criteria. At the time of admission of any student the admission department stores various information such as their name, DOB, address, email, the course they choose, the subjects that the course contains and the lecturer name as well as their certain information which are stored in their database.

1.3 Description of Project

The goal of the project student enrollment system is to collect all of the required data for college or university admission process. This system is created to make the college administration operate in effective way while enrolling students and managing their courses. The overall objective of an enrollment system is to help admission teams ultimately enroll more students in one place, both online and offline.

2. Database Model

2.1 Business Rules

The business rule of this student enrollment system is, we consider that a course may have many subject but each subject is unique and each courses have different set of subject i.e. a subject is unique in each course. Each subject is taught by many lecturer but one single lecturer cannot teach more than one subject. A single course can have many enrollments but one student enrolment cannot have many courses and one student can only take one enrolment and access one course.

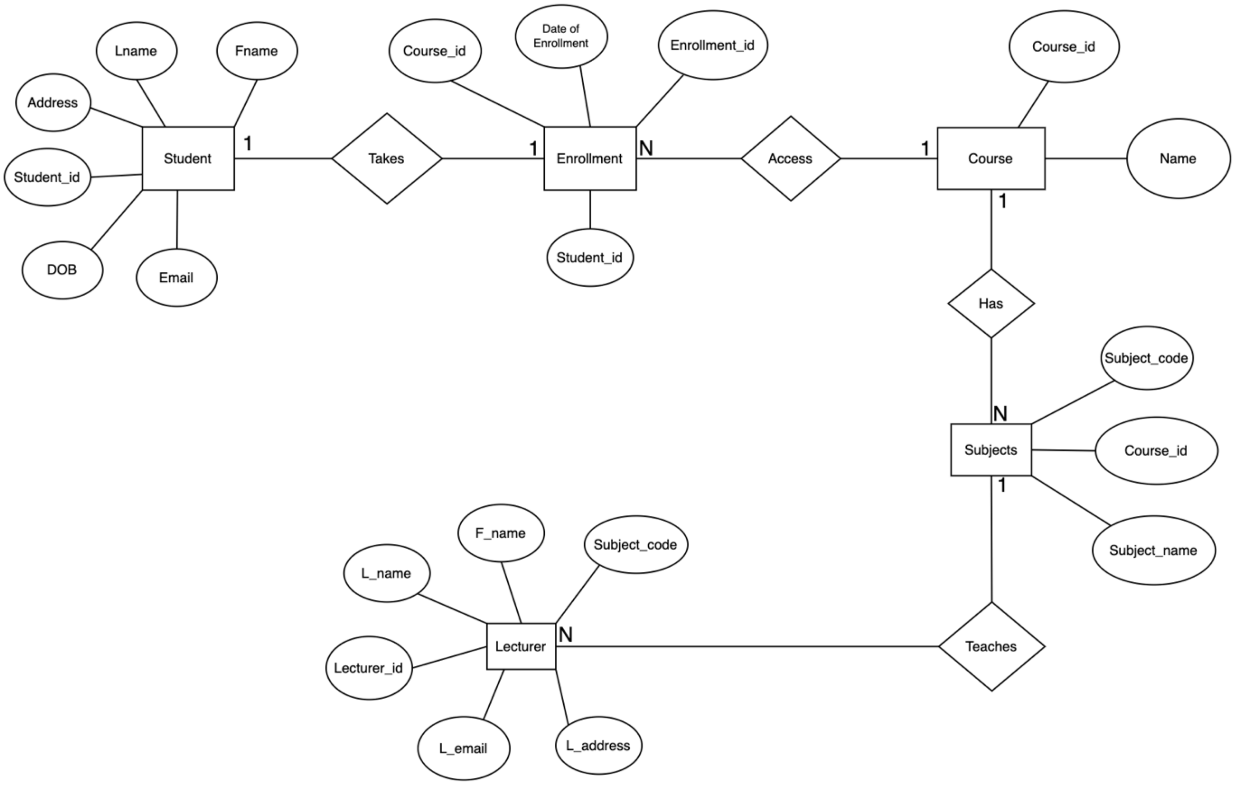


Figure 1: ER-Diagram

3. Relational Model

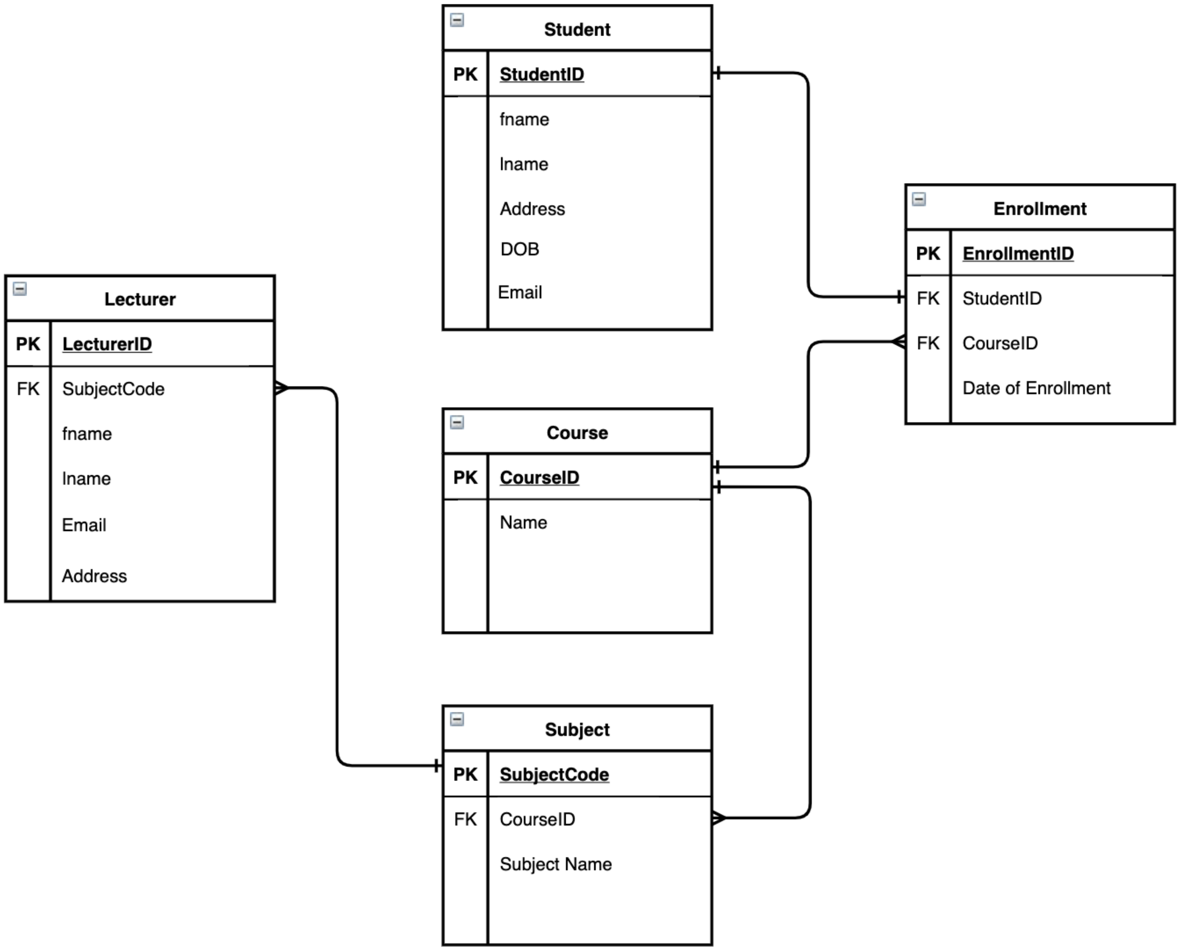
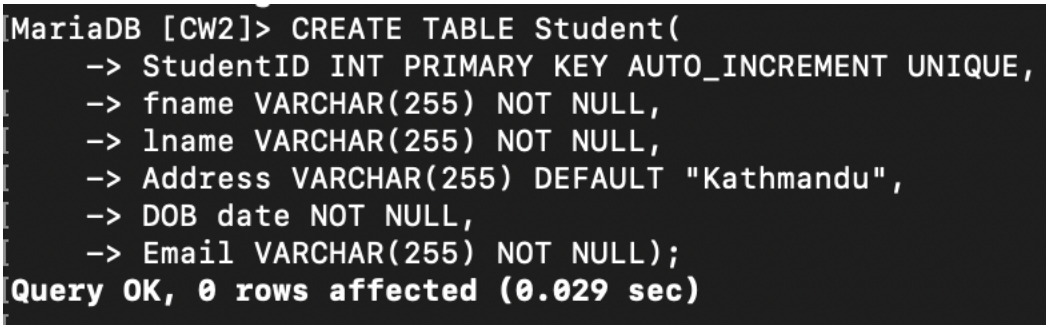


Figure 2: Relational Diagram

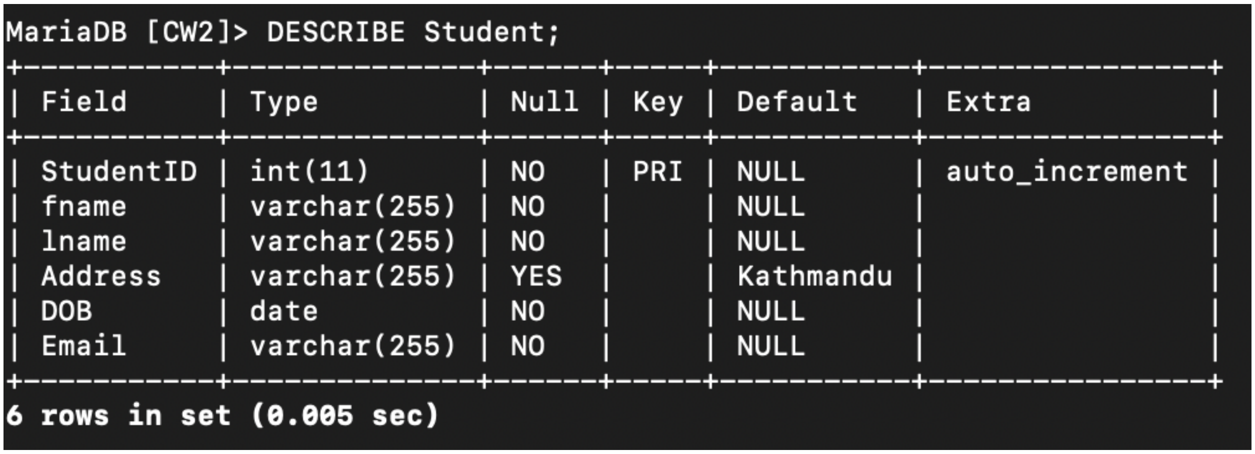
3.1 Student

A table student is formed to store records of the students when they are being enrolled by the admission department. In this table Student ID, Student’s fname, lname, address, date of birth and email is created to store its attributes. Student ID is the primary key of the table which uniquely identifies other attributes. Whereas, Student’s fname, lname, address, date of birth and email stores the name of the students, as well as their address, date of birth and their email respectively.



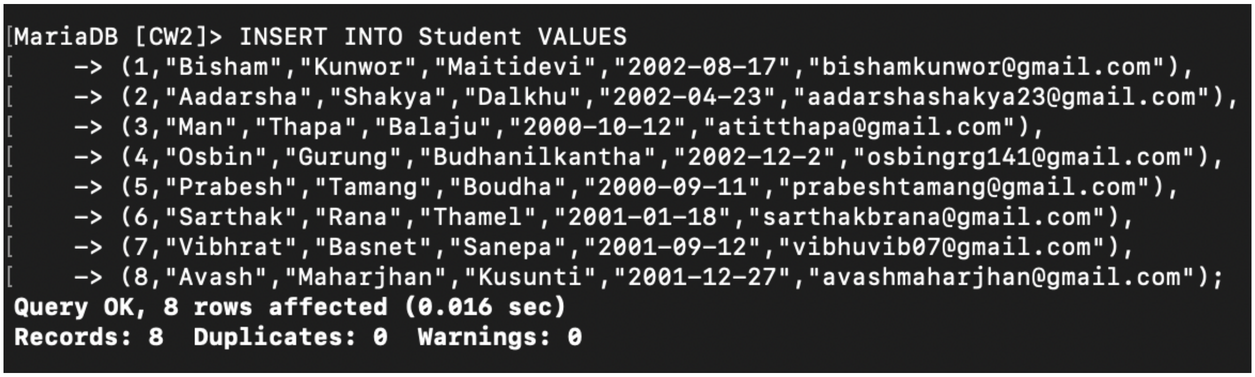
```
[MariaDB [CW2]> CREATE TABLE Student(
    -> StudentID INT PRIMARY KEY AUTO_INCREMENT UNIQUE,
[    -> fname VARCHAR(255) NOT NULL,
[    -> lname VARCHAR(255) NOT NULL,
[    -> Address VARCHAR(255) DEFAULT "Kathmandu",
[    -> DOB date NOT NULL,
[    -> Email VARCHAR(255) NOT NULL);
[Query OK, 0 rows affected (0.029 sec)
```

Figure 3: Creating Student Table



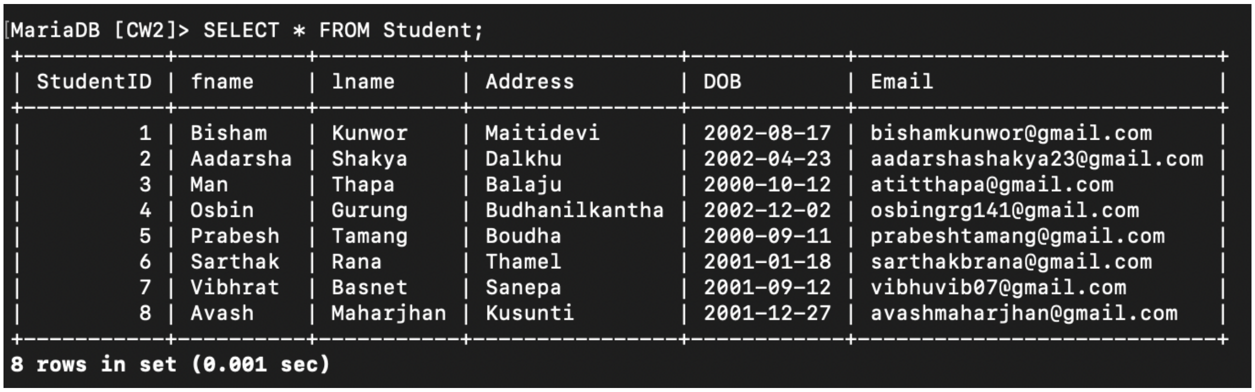
```
MariaDB [CW2]> DESCRIBE Student;
+-----------+--------------+------+-----+-----------+----------------+
| Field     | Type         | Null | Key | Default   | Extra          |
+-----------+--------------+------+-----+-----------+----------------+
| StudentID | int(11)      | NO   | PRI | NULL      | auto_increment |
| fname     | varchar(255) | NO   |     | NULL      |                |
| lname     | varchar(255) | NO   |     | NULL      |                |
| Address   | varchar(255) | YES  |     | Kathmandu |                |
| DOB       | date         | NO   |     | NULL      |                |
| Email     | varchar(255) | NO   |     | NULL      |                |
+-----------+--------------+------+-----+-----------+----------------+
6 rows in set (0.005 sec)
```

Figure 4: Describing Student Table



```
[MariaDB [CW2]> INSERT INTO Student VALUES
[    -> (1,"Bisham","Kunwor","Maitidevi","2002-08-17","bishamkunwor@gmail.com"),
[    -> (2,"Aadarsha","Shakya","Dalkhu","2002-04-23","aadarshashakya23@gmail.com"),
[    -> (3,"Man","Thapa","Balaju","2000-10-12","atitthapa@gmail.com"),
[    -> (4,"Osbin","Gurung","Budhanilkantha","2002-12-2","osbingrg141@gmail.com"),
[    -> (5,"Prabesh","Tamang","Boudha","2000-09-11","prabeshtamang@gmail.com"),
[    -> (6,"Sarthak","Rana","Thamel","2001-01-18","sarthakbrana@gmail.com"),
[    -> (7,"Vibhrat","Basnet","Sanepa","2001-09-12","vibhuvib07@gmail.com"),
[    -> (8,"Avash","Maharjhan","Kusunti","2001-12-27","avashmaharjhan@gmail.com");
Query OK, 8 rows affected (0.016 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

Figure 5: Inserting into Student Table

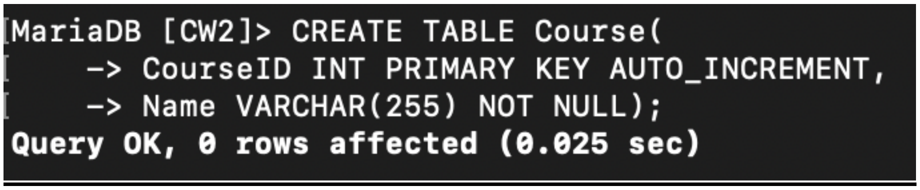


```
[MariaDB [CW2]> SELECT * FROM Student;
+-----------+----------+-----------+----------------+------------+----------------------------+
| StudentID | fname    | lname     | Address        | DOB        | Email                      |
+-----------+----------+-----------+----------------+------------+----------------------------+
|         1 | Bisham   | Kunwor    | Maitidevi      | 2002-08-17 | bishamkunwor@gmail.com     |
|         2 | Aadarsha | Shakya    | Dalkhu         | 2002-04-23 | aadarshashakya23@gmail.com |
|         3 | Man      | Thapa     | Balaju         | 2000-10-12 | atitthapa@gmail.com        |
|         4 | Osbin    | Gurung    | Budhanilkantha | 2002-12-02 | osbingrg141@gmail.com      |
|         5 | Prabesh  | Tamang    | Boudha         | 2000-09-11 | prabeshtamang@gmail.com    |
|         6 | Sarthak  | Rana      | Thamel         | 2001-01-18 | sarthakbrana@gmail.com     |
|         7 | Vibhrat  | Basnet    | Sanepa         | 2001-09-12 | vibhuvib07@gmail.com       |
|         8 | Avash    | Maharjhan | Kusunti        | 2001-12-27 | avashmaharjhan@gmail.com   |
+-----------+----------+-----------+----------------+------------+----------------------------+
8 rows in set (0.001 sec)
```

Figure 6: Selecting Student Table

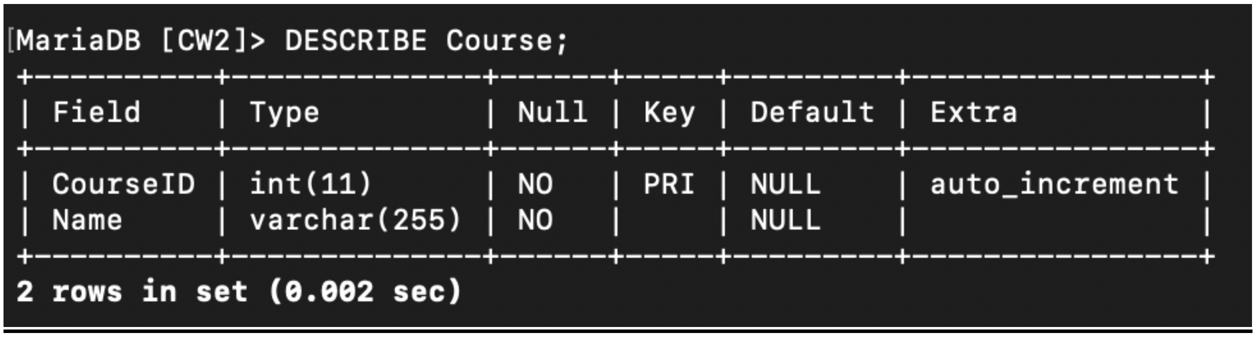
3.2 Course

A table course is formed to store records of the different courses that the college or university provides. In this table Course ID and name of the course is created to store its attributes. Course ID is the primary key of the table which uniquely identifies other attributes. Whereas, name stores the name of the course.



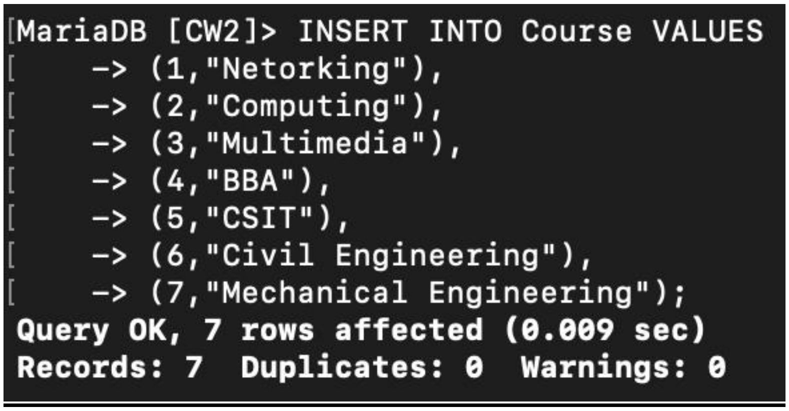
```
[MariaDB [CW2]> CREATE TABLE Course(
[    -> CourseID INT PRIMARY KEY AUTO_INCREMENT,
[    -> Name VARCHAR(255) NOT NULL);
Query OK, 0 rows affected (0.025 sec)
```

Figure 7: Creating Course Table



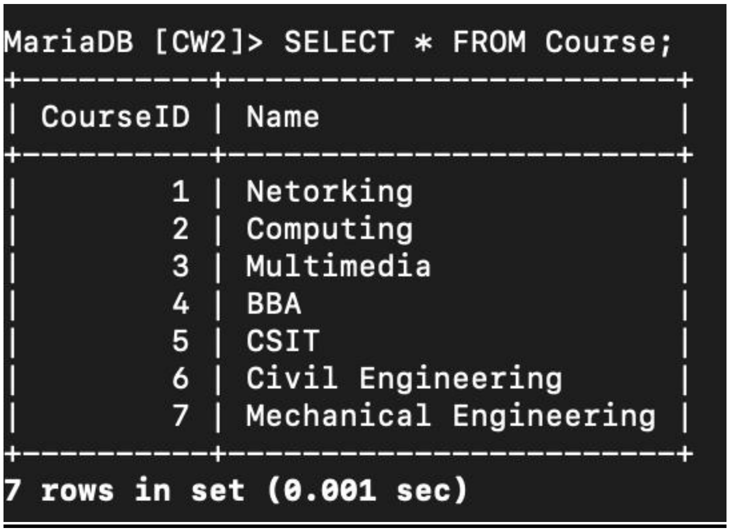
```
[MariaDB [CW2]> DESCRIBE Course;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| CourseID | int(11)      | NO   | PRI | NULL    | auto_increment |
| Name     | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
2 rows in set (0.002 sec)
```

Figure 8: Describing Course table



```
[MariaDB [CW2]> INSERT INTO Course VALUES
[    -> (1,"Netorking"),
[    -> (2,"Computing"),
[    -> (3,"Multimedia"),
[    -> (4,"BBA"),
[    -> (5,"CSIT"),
[    -> (6,"Civil Engineering"),
[    -> (7,"Mechanical Engineering");
Query OK, 7 rows affected (0.009 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

Figure 9: Inserting into Course Table

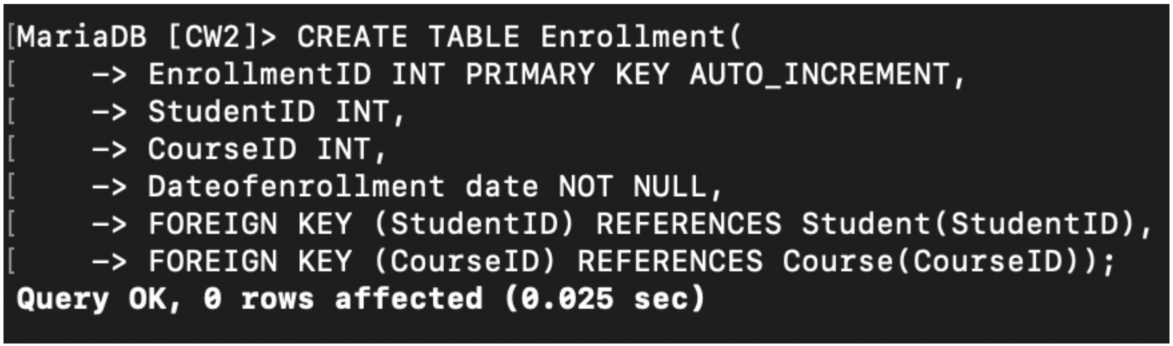


```
MariaDB [CW2]> SELECT * FROM Course;
+----------+------------------------+
| CourseID | Name                   |
+----------+------------------------+
|        1 | Netorking              |
|        2 | Computing              |
|        3 | Multimedia             |
|        4 | BBA                    |
|        5 | CSIT                   |
|        6 | Civil Engineering      |
|        7 | Mechanical Engineering |
+----------+------------------------+
7 rows in set (0.001 sec)
```

Figure 10: Selecting Course Table

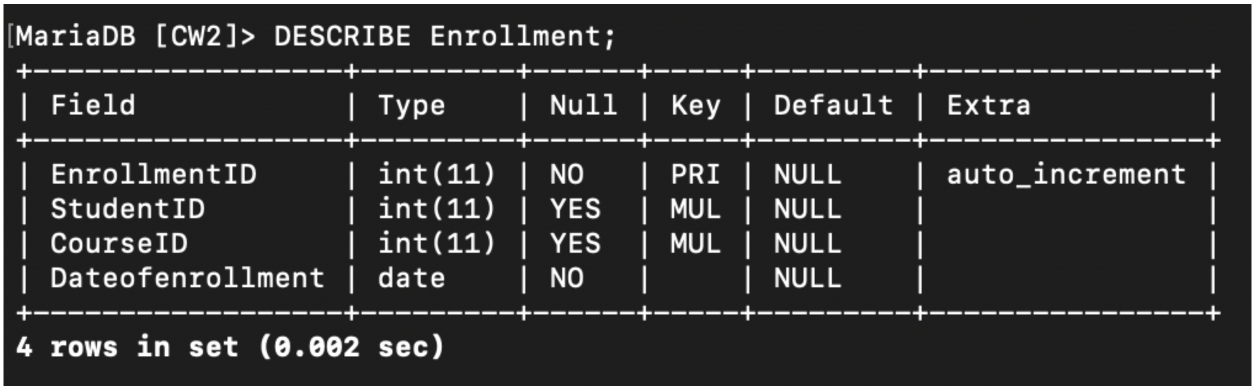
3.3 Enrollment

A table enrollment is formed to store records of the students, courses at the time of admission. In this table Enrollment ID, Student ID, Course ID and date of enrollment is created to store its attributes. Enrollment ID is the primary key of the table which uniquely identifies other attributes. Whereas, date of enrollment stores the date when student was enrolled to. Student ID and Course ID are the foreign keys with reference from Student and Course table.



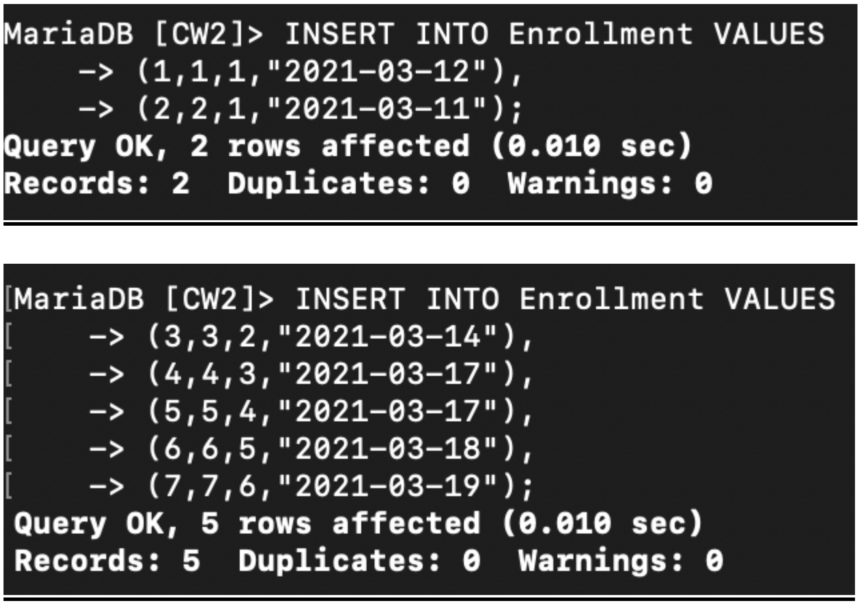
```
[MariaDB [CW2]> CREATE TABLE Enrollment(
[    -> EnrollmentID INT PRIMARY KEY AUTO_INCREMENT,
[    -> StudentID INT,
[    -> CourseID INT,
[    -> Dateofenrollment date NOT NULL,
[    -> FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
[    -> FOREIGN KEY (CourseID) REFERENCES Course(CourseID));
Query OK, 0 rows affected (0.025 sec)
```

Figure 11: Creating Enrollment Table



```
[MariaDB [CW2]> DESCRIBE Enrollment;
+------------------+---------+------+-----+---------+----------------+
| Field            | Type    | Null | Key | Default | Extra          |
+------------------+---------+------+-----+---------+----------------+
| EnrollmentID     | int(11) | NO   | PRI | NULL    | auto_increment |
| StudentID        | int(11) | YES  | MUL | NULL    |                |
| CourseID         | int(11) | YES  | MUL | NULL    |                |
| Dateofenrollment | date    | NO   |     | NULL    |                |
+------------------+---------+------+-----+---------+----------------+
4 rows in set (0.002 sec)
```

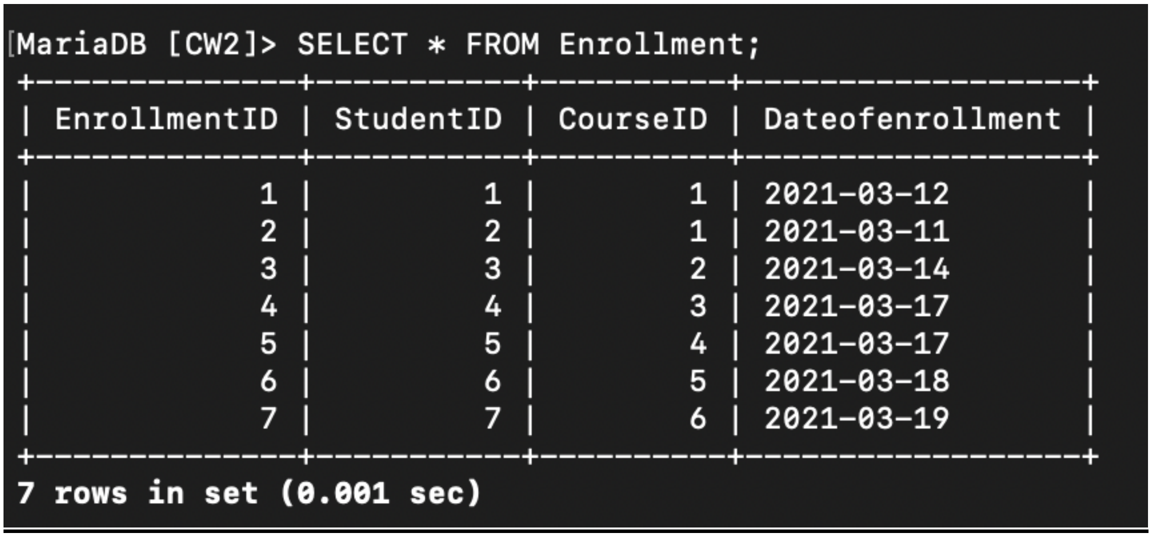
Figure 12: Describing Enrollment Table



```
MariaDB [CW2]> INSERT INTO Enrollment VALUES
    -> (1,1,1,"2021-03-12"),
    -> (2,2,1,"2021-03-11");
Query OK, 2 rows affected (0.010 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
[MariaDB [CW2]> INSERT INTO Enrollment VALUES
[    -> (3,3,2,"2021-03-14"),
[    -> (4,4,3,"2021-03-17"),
[    -> (5,5,4,"2021-03-17"),
[    -> (6,6,5,"2021-03-18"),
[    -> (7,7,6,"2021-03-19");
 Query OK, 5 rows affected (0.010 sec)
 Records: 5  Duplicates: 0  Warnings: 0
```

Figure 13: Inserting into Enrollment Table

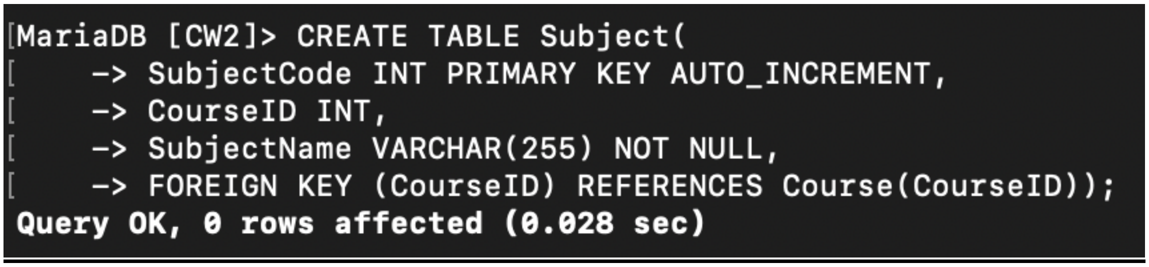


```
[MariaDB [CW2]> SELECT * FROM Enrollment;
 +--------------+-----------+----------+------------------+
 | EnrollmentID | StudentID | CourseID | Dateofenrollment |
 +--------------+-----------+----------+------------------+
 |            1 |         1 |        1 | 2021-03-12       |
 |            2 |         2 |        1 | 2021-03-11       |
 |            3 |         3 |        2 | 2021-03-14       |
 |            4 |         4 |        3 | 2021-03-17       |
 |            5 |         5 |        4 | 2021-03-17       |
 |            6 |         6 |        5 | 2021-03-18       |
 |            7 |         7 |        6 | 2021-03-19       |
 +--------------+-----------+----------+------------------+
 7 rows in set (0.001 sec)
```

Figure 14: Selecting Enrollment Table

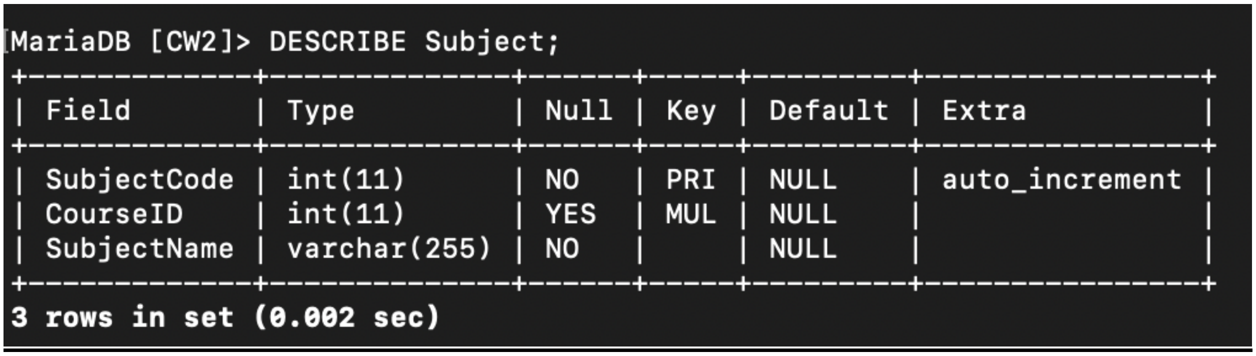
3.4 Subject

A table subject is formed to store records of the different subjects that each course contains. In this table Subject Code, Course ID and Subject Name is created to store its attributes. Subject Code is the primary key of the table which uniquely identifies other attributes. Whereas, subject name stores the name of the subject of different course. Course ID is the foreign key with reference from Course table.



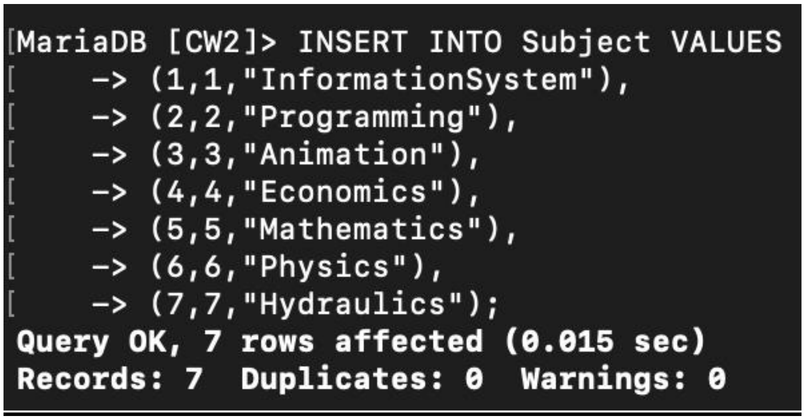
```
[MariaDB [CW2]> CREATE TABLE Subject(
[    -> SubjectCode INT PRIMARY KEY AUTO_INCREMENT,
[    -> CourseID INT,
[    -> SubjectName VARCHAR(255) NOT NULL,
[    -> FOREIGN KEY (CourseID) REFERENCES Course(CourseID));
 Query OK, 0 rows affected (0.028 sec)
```

Figure 15: Creating Subject Table



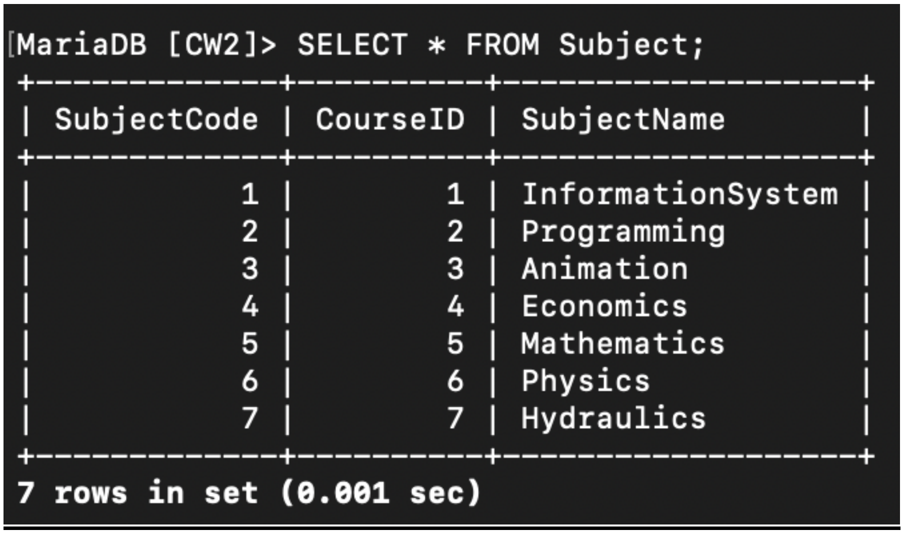
```
[MariaDB [CW2]> DESCRIBE Subject;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| SubjectCode | int(11)      | NO   | PRI | NULL    | auto_increment |
| CourseID    | int(11)      | YES  | MUL | NULL    |                |
| SubjectName | varchar(255) | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
3 rows in set (0.002 sec)
```

Figure 16: Describing Subject Table



```
[MariaDB [CW2]> INSERT INTO Subject VALUES
[    -> (1,1,"InformationSystem"),
[    -> (2,2,"Programming"),
[    -> (3,3,"Animation"),
[    -> (4,4,"Economics"),
[    -> (5,5,"Mathematics"),
[    -> (6,6,"Physics"),
[    -> (7,7,"Hydraulics");
Query OK, 7 rows affected (0.015 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

Figure 17: Inserting into Subject Table

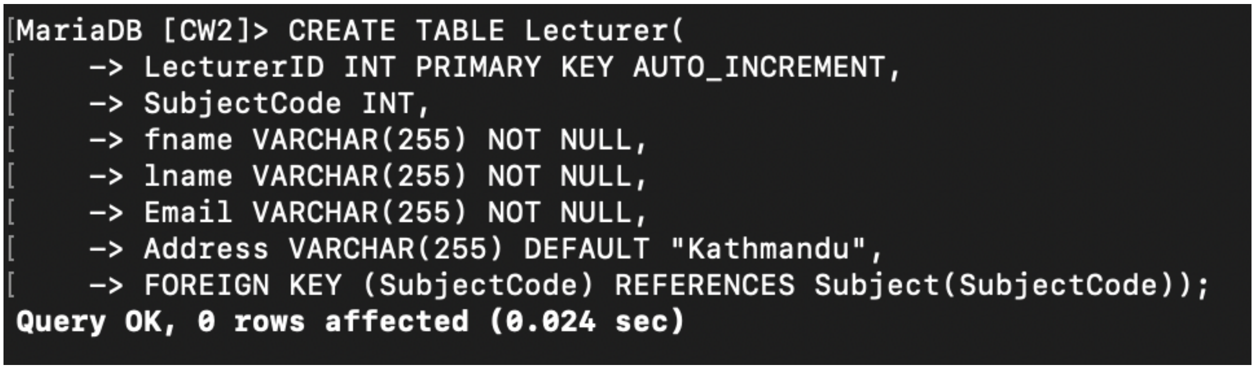


```
[MariaDB [CW2]> SELECT * FROM Subject;
+-------------+----------+-------------------+
| SubjectCode | CourseID | SubjectName       |
+-------------+----------+-------------------+
|           1 |        1 | InformationSystem |
|           2 |        2 | Programming       |
|           3 |        3 | Animation         |
|           4 |        4 | Economics         |
|           5 |        5 | Mathematics       |
|           6 |        6 | Physics           |
|           7 |        7 | Hydraulics        |
+-------------+----------+-------------------+
7 rows in set (0.001 sec)
```

Figure 18: Selecting Subject Table

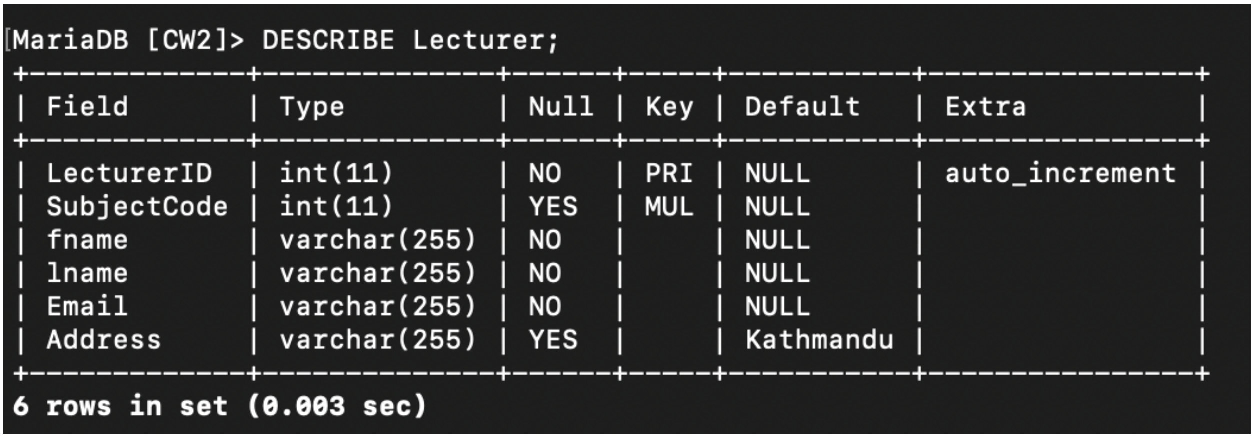
3.5 Lecturer

A table lecturer is formed to store records of the lecturers who teaches different subjects to the students by the admission department. In this table Lecturer ID, Subject Code, Lecturer’s fname, lname, address and email is created to store its attributes. Lecturer ID is the primary key of the table which uniquely identifies other attributes. Whereas, lecturer’s fname, lname, address and email stores the name of the lecturer’s, as well as their address and their email respectively. Subject Code is the foreign key with reference from Subject table.



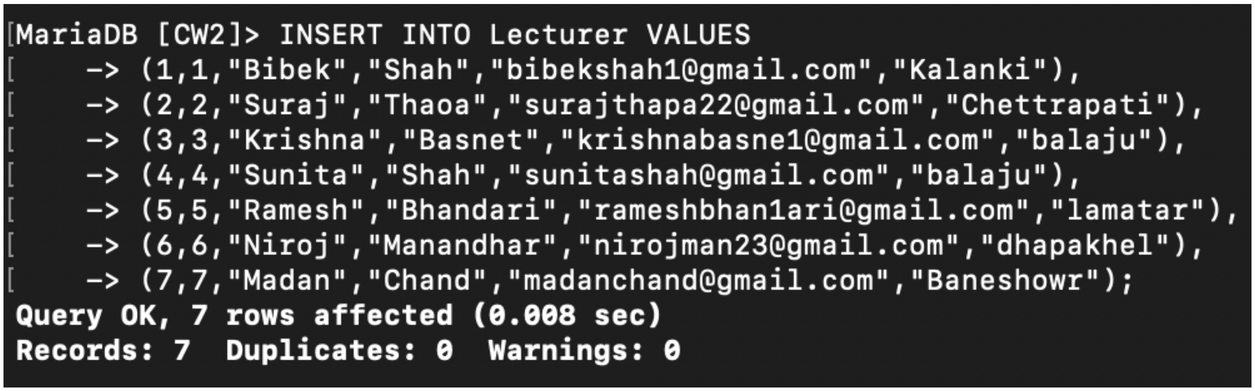
```
[MariaDB [CW2]> CREATE TABLE Lecturer(
[    -> LecturerID INT PRIMARY KEY AUTO_INCREMENT,
[    -> SubjectCode INT,
[    -> fname VARCHAR(255) NOT NULL,
[    -> lname VARCHAR(255) NOT NULL,
[    -> Email VARCHAR(255) NOT NULL,
[    -> Address VARCHAR(255) DEFAULT "Kathmandu",
[    -> FOREIGN KEY (SubjectCode) REFERENCES Subject(SubjectCode));
Query OK, 0 rows affected (0.024 sec)
```

Figure 19: Creating Lecturer Table



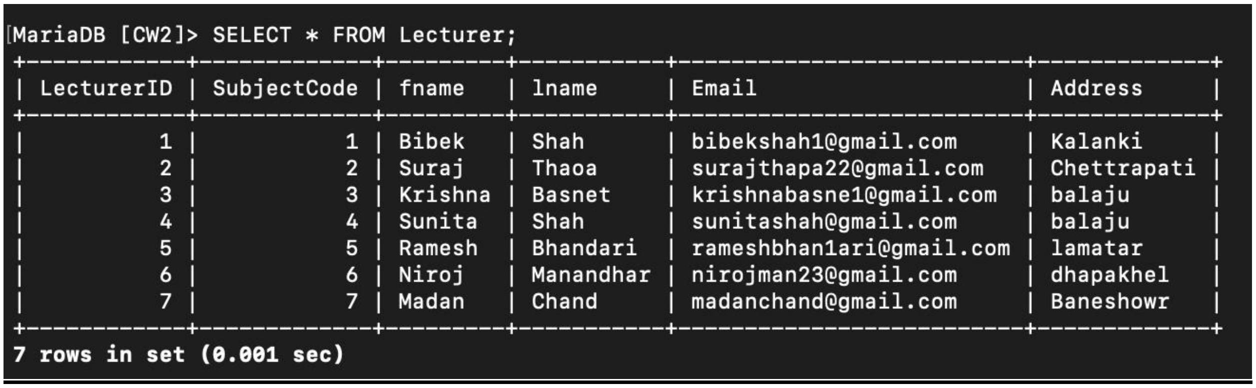
```
[MariaDB [CW2]> DESCRIBE Lecturer;
+-------------+--------------+------+-----+-----------+----------------+
| Field       | Type         | Null | Key | Default   | Extra          |
+-------------+--------------+------+-----+-----------+----------------+
| LecturerID  | int(11)      | NO   | PRI | NULL      | auto_increment |
| SubjectCode | int(11)      | YES  | MUL | NULL      |                |
| fname       | varchar(255) | NO   |     | NULL      |                |
| lname       | varchar(255) | NO   |     | NULL      |                |
| Email       | varchar(255) | NO   |     | NULL      |                |
| Address     | varchar(255) | YES  |     | Kathmandu |                |
+-------------+--------------+------+-----+-----------+----------------+
6 rows in set (0.003 sec)
```

Figure 20: Describing Lecturer Table



```
[MariaDB [CW2]> INSERT INTO Lecturer VALUES
[    -> (1,1,"Bibek","Shah","bibekshah1@gmail.com","Kalanki"),
[    -> (2,2,"Suraj","Thaoa","surajthapa22@gmail.com","Chettrapati"),
[    -> (3,3,"Krishna","Basnet","krishnabasne1@gmail.com","balaju"),
[    -> (4,4,"Sunita","Shah","sunitashah@gmail.com","balaju"),
[    -> (5,5,"Ramesh","Bhandari","rameshbhan1ari@gmail.com","lamatar"),
[    -> (6,6,"Niroj","Manandhar","nirojman23@gmail.com","dhapakhel"),
[    -> (7,7,"Madan","Chand","madanchand@gmail.com","Baneshowr");
Query OK, 7 rows affected (0.008 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

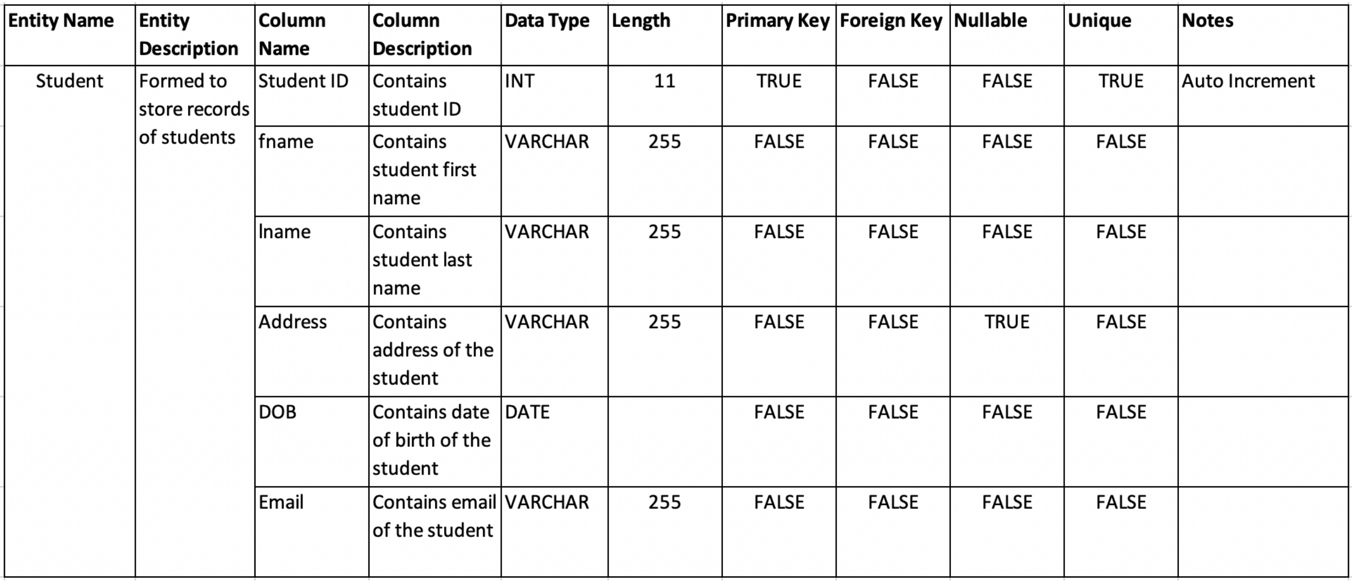
Figure 21: Inserting into Lecturer Table



```
[MariaDB [CW2]> SELECT * FROM Lecturer;
+------------+-------------+---------+-----------+--------------------------+-------------+
| LecturerID | SubjectCode | fname   | lname     | Email                    | Address     |
+------------+-------------+---------+-----------+--------------------------+-------------+
|          1 |           1 | Bibek   | Shah      | bibekshah1@gmail.com     | Kalanki     |
|          2 |           2 | Suraj   | Thaoa     | surajthapa22@gmail.com   | Chettrapati |
|          3 |           3 | Krishna | Basnet    | krishnabasne1@gmail.com  | balaju      |
|          4 |           4 | Sunita  | Shah      | sunitashah@gmail.com     | balaju      |
|          5 |           5 | Ramesh  | Bhandari  | rameshbhan1ari@gmail.com | lamatar     |
|          6 |           6 | Niroj   | Manandhar | nirojman23@gmail.com     | dhapakhel   |
|          7 |           7 | Madan   | Chand     | madanchand@gmail.com     | Baneshowr   |
+------------+-------------+---------+-----------+--------------------------+-------------+
7 rows in set (0.001 sec)
```

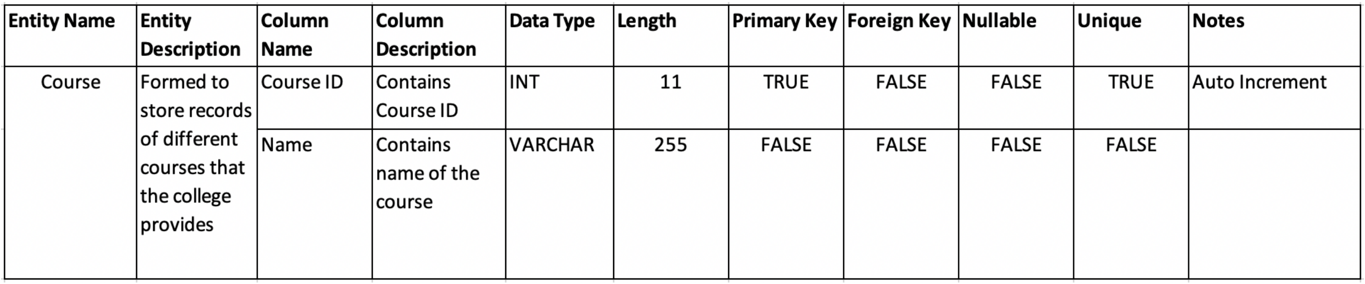
Figure 22: Selecting Lecturer Table

4. Data Dictionary



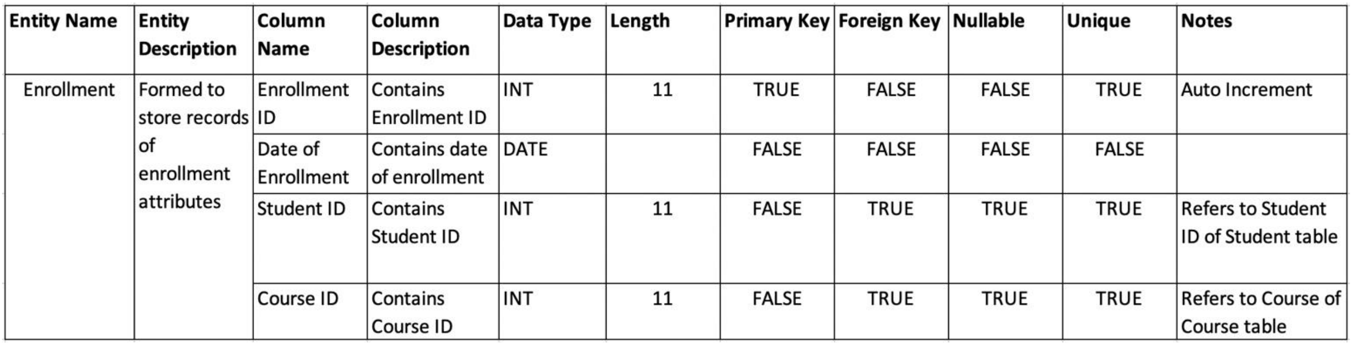
Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Foreign Key	Nullable	Unique	Notes
Student	Formed to store records of students	Student ID	Contains student ID	INT	11	TRUE	FALSE	FALSE	TRUE	Auto Increment
		fname	Contains student first name	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		lname	Contains student last name	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Address	Contains address of the student	VARCHAR	255	FALSE	FALSE	TRUE	FALSE	
		DOB	Contains date of birth of the student	DATE		FALSE	FALSE	FALSE	FALSE	
		Email	Contains email of the student	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	

Table 1: Student Entity Data Dictionary



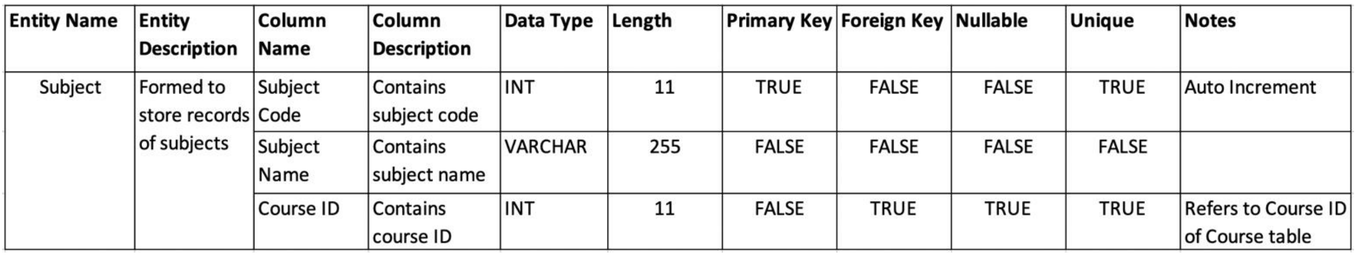
Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Foreign Key	Nullable	Unique	Notes
Course	Formed to store records of different courses that the college provides	Course ID	Contains Course ID	INT	11	TRUE	FALSE	FALSE	TRUE	Auto Increment
		Name	Contains name of the course	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	

Table 2: Course Entity Data Dictionary



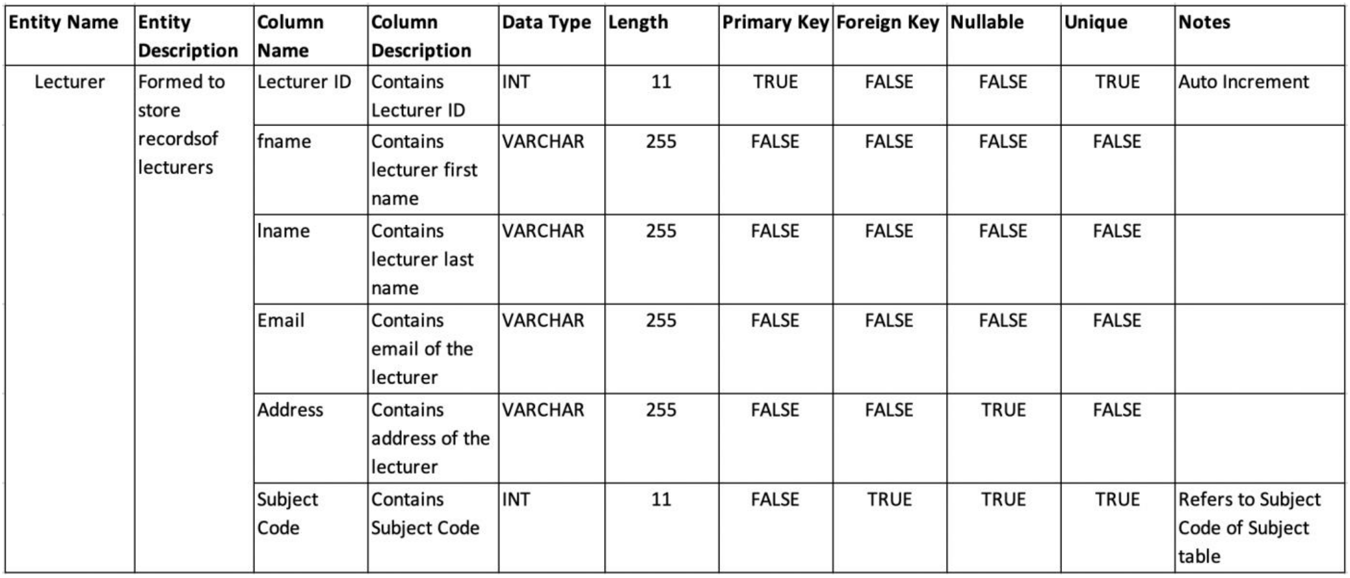
Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Foreign Key	Nullable	Unique	Notes
Enrollment	Formed to store records of enrollment attributes	Enrollment ID	Contains Enrollment ID	INT	11	TRUE	FALSE	FALSE	TRUE	Auto Increment
		Date of Enrollment	Contains date of enrollment	DATE		FALSE	FALSE	FALSE	FALSE	
		Student ID	Contains Student ID	INT	11	FALSE	TRUE	TRUE	TRUE	Refers to Student ID of Student table
		Course ID	Contains Course ID	INT	11	FALSE	TRUE	TRUE	TRUE	Refers to Course of Course table

Table 3: Enrollment Entity Data Dictionary



Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Foreign Key	Nullable	Unique	Notes
Subject	Formed to store records of subjects	Subject Code	Contains subject code	INT	11	TRUE	FALSE	FALSE	TRUE	Auto Increment
		Subject Name	Contains subject name	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Course ID	Contains course ID	INT	11	FALSE	TRUE	TRUE	TRUE	Refers to Course ID of Course table

Table 4: Subject Entity Data Dictionary



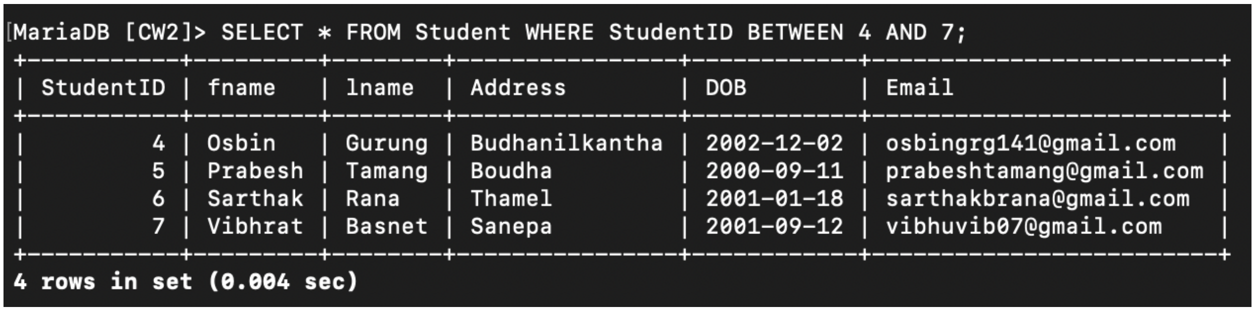
Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Foreign Key	Nullable	Unique	Notes
Lecturer	Formed to store recordsof lecturers	Lecturer ID	Contains Lecturer ID	INT	11	TRUE	FALSE	FALSE	TRUE	Auto Increment
		fname	Contains lecturer first name	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		lname	Contains lecturer last name	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Email	Contains email of the lecturer	VARCHAR	255	FALSE	FALSE	FALSE	FALSE	
		Address	Contains address of the lecturer	VARCHAR	255	FALSE	FALSE	TRUE	FALSE	
		Subject Code	Contains Subject Code	INT	11	FALSE	TRUE	TRUE	TRUE	Refers to Subject Code of Subject table

Table 5: Lecturer Entity Data Dictionary

5. Queries

Query Number	Query 1
Query	SELECT * FROM Student WHERE Student ID BETWEEN 4 AND 7;
Keywords Used	SELECT, FROM, WHERE, BETWEEN, AND
Purpose/Result	Selecting students whose Student ID is between 4 to 7

Table 6: Query 1

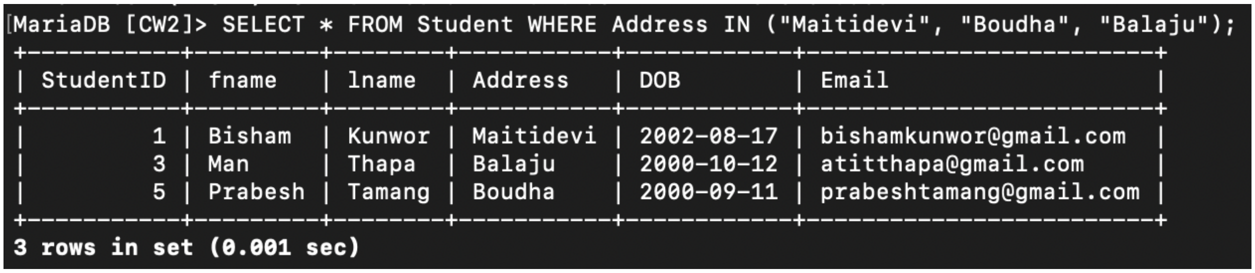


```
[MariaDB [CW2]> SELECT * FROM Student WHERE StudentID BETWEEN 4 AND 7;
+-----------+---------+--------+----------------+------------+-------------------------+
| StudentID | fname   | lname  | Address        | DOB        | Email                   |
+-----------+---------+--------+----------------+------------+-------------------------+
|         4 | Osbin   | Gurung | Budhanilkantha | 2002-12-02 | osbingrg141@gmail.com   |
|         5 | Prabesh | Tamang | Boudha         | 2000-09-11 | prabeshtamang@gmail.com |
|         6 | Sarthak | Rana   | Thamel         | 2001-01-18 | sarthakbrana@gmail.com  |
|         7 | Vibhrat | Basnet | Sanepa         | 2001-09-12 | vibhuvib07@gmail.com    |
+-----------+---------+--------+----------------+------------+-------------------------+
4 rows in set (0.004 sec)
```

Figure 23: Query 1

Query Number	Query 2
Query	SELECT * FROM Student WHERE Address IN (“Maitidevi”, “Boudha”, “Balaju”);
Keywords Used	SELECT, FROM, WHERE, IN
Purpose/Result	Selecting data that have Address Maitidevi, Boudha and Balaju

Table 7: Query 2

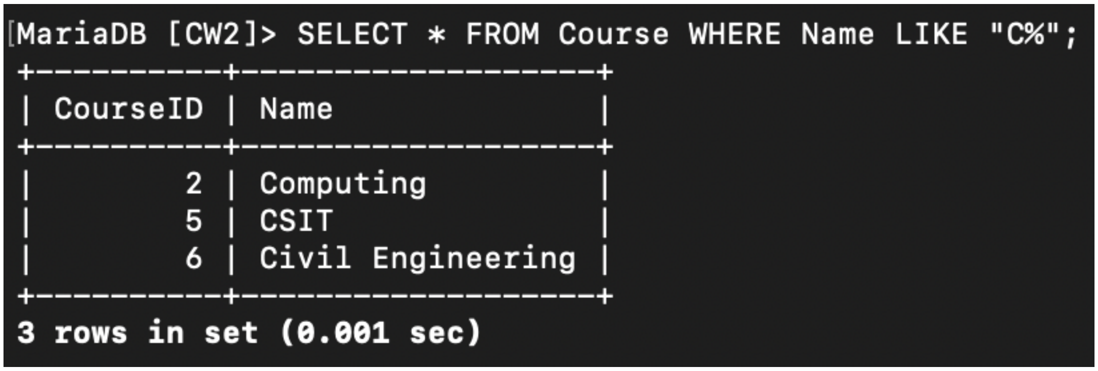


```
[MariaDB [CW2]> SELECT * FROM Student WHERE Address IN ("Maitidevi", "Boudha", "Balaju");
+-----------+---------+--------+-----------+------------+-------------------------+
| StudentID | fname   | lname  | Address   | DOB        | Email                   |
+-----------+---------+--------+-----------+------------+-------------------------+
|         1 | Bisham  | Kunwor | Maitidevi | 2002-08-17 | bishamkunwor@gmail.com  |
|         3 | Man     | Thapa  | Balaju    | 2000-10-12 | atitthapa@gmail.com     |
|         5 | Prabesh | Tamang | Boudha    | 2000-09-11 | prabeshtamang@gmail.com |
+-----------+---------+--------+-----------+------------+-------------------------+
3 rows in set (0.001 sec)
```

Figure 24: Query 2

Query Number	Query 3
Query	SELECT * FROM Course WHERE Name LIKE “C%”;
Keywords Used	SWLWCT, FROM, WHERE, LIKE
Purpose/Result	Selecting Course who’s first character starts from “C”

Table 8: Query 3

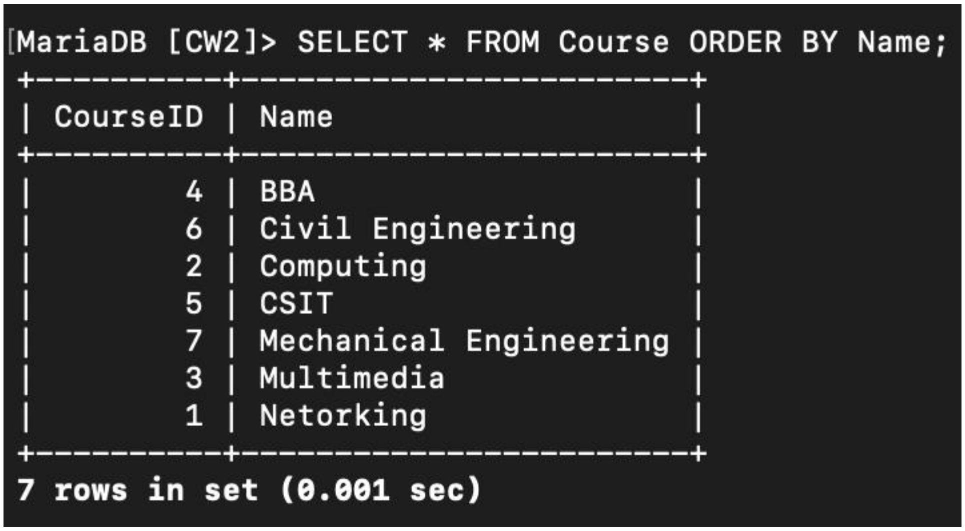


```
[MariaDB [CW2]> SELECT * FROM Course WHERE Name LIKE "C%";
+----------+-------------------+
| CourseID | Name              |
+----------+-------------------+
|        2 | Computing         |
|        5 | CSIT              |
|        6 | Civil Engineering |
+----------+-------------------+
3 rows in set (0.001 sec)
```

Figure 25: Query 3

Query Number	Query 4
Query	SELECT * FROM Course ORDER BY Name;
Keywords Used	SELECT, FROM ORDER BY
Purpose/Result	Selecting Courses name in order

Table 9: Query 4

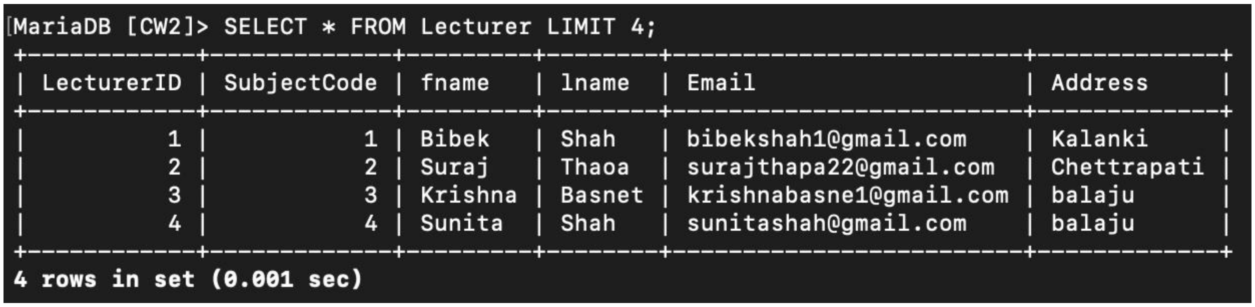


```
[MariaDB [CW2]> SELECT * FROM Course ORDER BY Name;
+----------+------------------------+
| CourseID | Name                   |
+----------+------------------------+
|        4 | BBA                    |
|        6 | Civil Engineering      |
|        2 | Computing              |
|        5 | CSIT                   |
|        7 | Mechanical Engineering |
|        3 | Multimedia             |
|        1 | Netorking              |
+----------+------------------------+
7 rows in set (0.001 sec)
```

Figure 26: Query 4

Query Number	Query 5
Query	SELECT * FROM Lecturer LIMIT 4;
Keywords Used	SELECT, FROM, LIMIT
Purpose/Result	Selecting top 4 data from Lecturer table

Table 10: Query 5

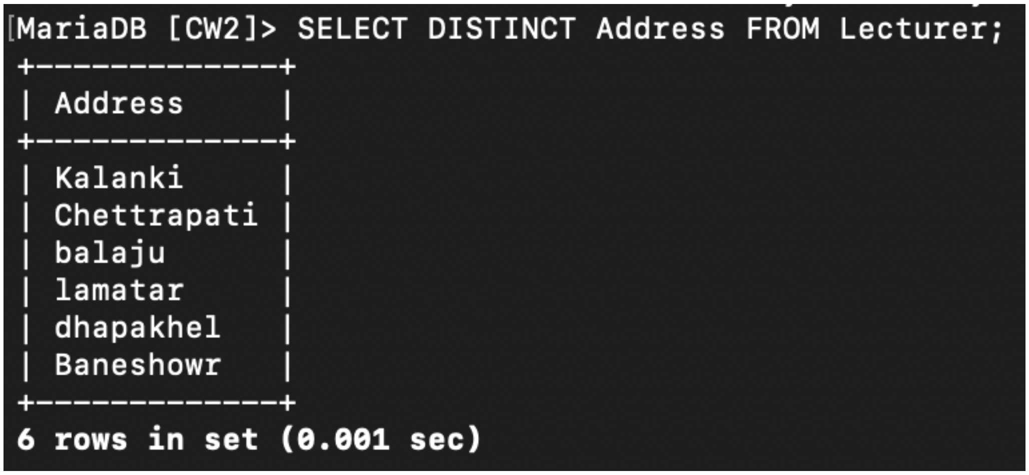


```
[MariaDB [CW2]> SELECT * FROM Lecturer LIMIT 4;
+------------+-------------+---------+--------+-------------------------+-------------+
| LecturerID | SubjectCode | fname   | lname  | Email                   | Address     |
+------------+-------------+---------+--------+-------------------------+-------------+
|          1 |           1 | Bibek   | Shah   | bibekshah1@gmail.com    | Kalanki     |
|          2 |           2 | Suraj   | Thaoa  | surajthapa22@gmail.com  | Chettrapati |
|          3 |           3 | Krishna | Basnet | krishnabasne1@gmail.com | balaju      |
|          4 |           4 | Sunita  | Shah   | sunitashah@gmail.com    | balaju      |
+------------+-------------+---------+--------+-------------------------+-------------+
4 rows in set (0.001 sec)
```

Figure 27: Query 5

Query Number	Query 6
Query	SELECT DISTINCT Address FROM Lecturer;
Keywords Used	SELECT, DISTINCT, FROM
Purpose/Result	Selecting unique address from Lecturer table

Table 11: Query 6

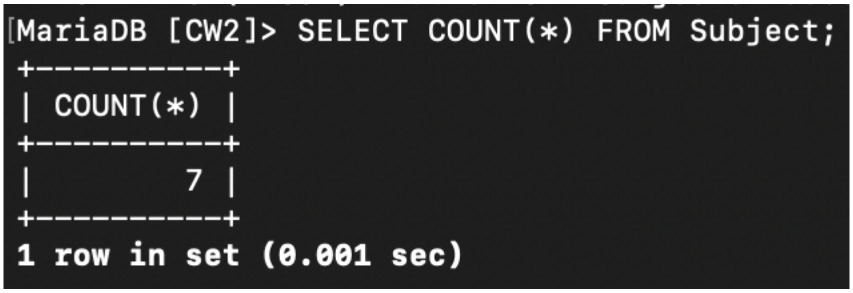


```
[MariaDB [CW2]> SELECT DISTINCT Address FROM Lecturer;
+-------------+
| Address     |
+-------------+
| Kalanki     |
| Chettrapati |
| balaju      |
| lamatar     |
| dhapakhel   |
| Baneshowr   |
+-------------+
6 rows in set (0.001 sec)
```

Figure 28: Query 6

Query Number	Query 7
Query	SEKECT COUNT(*) FROM Subject;
Keywords Used	SELECT, COUNT, FROM
Purpose/Result	Counting total number of Subjects from Subject table

Table 12: Query 7

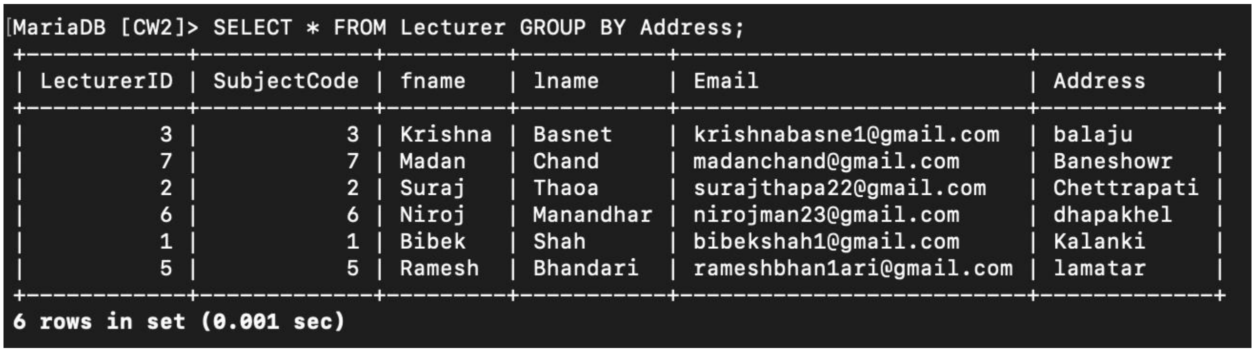


```
[MariaDB [CW2]> SELECT COUNT(*) FROM Subject;
+----------+
| COUNT(*) |
+----------+
|        7 |
+----------+
1 row in set (0.001 sec)
```

Figure 29: Query 7

Query Number	Query 8
Query	SELECT * FROM Lecturer GROUP BY Address;
Keywords Used	SELECT, FROM, GROUP BY
Purpose/Result	Selecting same address from address table

Table 13: Query 8

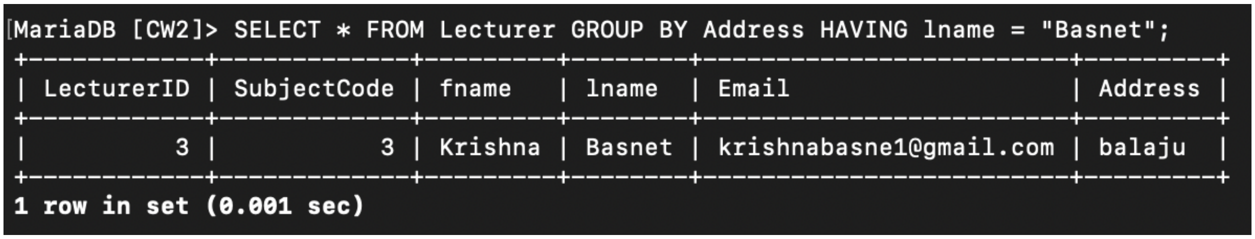


```
[MariaDB [CW2]> SELECT * FROM Lecturer GROUP BY Address;
+------------+-------------+---------+-----------+--------------------------+-------------+
| LecturerID | SubjectCode | fname   | lname     | Email                    | Address     |
+------------+-------------+---------+-----------+--------------------------+-------------+
|          3 |           3 | Krishna | Basnet    | krishnabasne1@gmail.com  | balaju      |
|          7 |           7 | Madan   | Chand     | madanchand@gmail.com     | Baneshowr   |
|          2 |           2 | Suraj   | Thaoa     | surajthapa22@gmail.com   | Chettrapati |
|          6 |           6 | Niroj   | Manandhar | nirojman23@gmail.com     | dhapakhel   |
|          1 |           1 | Bibek   | Shah      | bibekshah1@gmail.com     | Kalanki     |
|          5 |           5 | Ramesh  | Bhandari  | rameshbhan1ari@gmail.com | lamatar     |
+------------+-------------+---------+-----------+--------------------------+-------------+
6 rows in set (0.001 sec)
```

Figure 30: Query 8

Query Number	Query 9
Query	SELECT * FROM Lecturer GROUP BY Address HAVING lname = “Basnet”;
Keywords Used	SELECT, FROM, GROUP BY, HAVING
Purpose/Result	Selecting last name of the lecturer and also showing their address from lecturer table

Table 14: Query 9

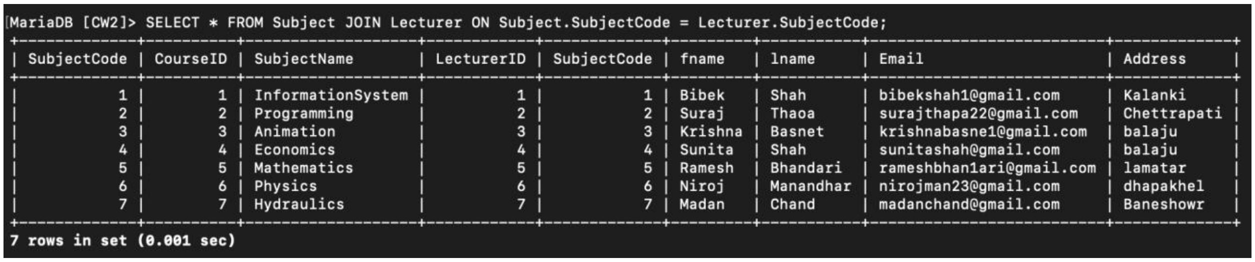


```
[MariaDB [CW2]> SELECT * FROM Lecturer GROUP BY Address HAVING lname = "Basnet";
+------------+-------------+---------+--------+-------------------------+---------+
| LecturerID | SubjectCode | fname   | lname  | Email                   | Address |
+------------+-------------+---------+--------+-------------------------+---------+
|          3 |           3 | Krishna | Basnet | krishnabasne1@gmail.com | balaju  |
+------------+-------------+---------+--------+-------------------------+---------+
1 row in set (0.001 sec)
```

Figure 31: Query 9

Query Number	Query 10
Query	SELECT * FROM Subject JOIN Lecturer ON Subject.SubjectCode = Lecturer.SubjectCode;
Keywords Used	SELECT, FROM, JOIN, ON
Purpose/Result	Joining two tables with similar SubjectCode

Table 15: Query 10



```
[MariaDB [CW2]> SELECT * FROM Subject JOIN Lecturer ON Subject.SubjectCode = Lecturer.SubjectCode;
+-------------+----------+-------------------+------------+-------------+---------+-----------+--------------------------+-------------+
| SubjectCode | CourseID | SubjectName       | LecturerID | SubjectCode | fname   | lname     | Email                    | Address     |
+-------------+----------+-------------------+------------+-------------+---------+-----------+--------------------------+-------------+
|           1 |        1 | InformationSystem |          1 |           1 | Bibek   | Shah      | bibekshah1@gmail.com     | Kalanki     |
|           2 |        2 | Programming       |          2 |           2 | Suraj   | Thaoa     | surajthapa22@gmail.com   | Chettrapati |
|           3 |        3 | Animation         |          3 |           3 | Krishna | Basnet    | krishnabasne1@gmail.com  | balaju      |
|           4 |        4 | Economics         |          4 |           4 | Sunita  | Shah      | sunitashah@gmail.com     | balaju      |
|           5 |        5 | Mathematics       |          5 |           5 | Ramesh  | Bhandari  | rameshbhan1ari@gmail.com | lamatar     |
|           6 |        6 | Physics           |          6 |           6 | Niroj   | Manandhar | nirojman23@gmail.com     | dhapakhel   |
|           7 |        7 | Hydraulics        |          7 |           7 | Madan   | Chand     | madanchand@gmail.com     | Baneshowr   |
+-------------+----------+-------------------+------------+-------------+---------+-----------+--------------------------+-------------+
7 rows in set (0.001 sec)
```

Figure 32: Query 10

6. Conclusion

In conclusion, databases are the gathering of logically associated records in a scientific way that is stored in an prepared way to fulfill the necessities of various customers of an enterprise that may be without difficulty accessed, managed and updated. Relational Diagram, Entity Relation Diagram and Data dictionary are formed for a clear representation before creating a database. Data dictionary contains constraints like primary key, foreign key, unique, not null and default which makes any user clear before creating a database.

Bibliography

R, V. M., 2020. *edureka.* [Online]
Available at: https://www.edureka.co/blog/what-is-a-database/#Database
[Accessed 21 04 2021].