VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DATABASE SYSTEMS (CO2013)
ASSIGNMENT REPORT

QUIZ MANAGEMENT SYSTEM

Instructor: Đỗ Thanh Thái, Ph.D.

Students: Nguyễn Thanh Bình - 2252082
Trần Quốc Bảo Long - 2252453
Đặng Duy Tiến - 2252808

Ho Chi Minh City, October 2024

CONTENTS

MEMBER LIST & WORKLOAD

No.	Full name	Student ID	Task	Percentage of work
1	Nguyễn Thanh Bình	2252082	Sketching and mapping diagrams	100%
2	Trần Quốc Bảo Long	2252453	Setting up and writing SQL commands Preparing the report and presentation	100%
3	Đặng Duy Tiến	2252808	Listing the entities and attributes	100%

I. INTRODUCTION

A university wants to develop a system to manage and hold exams with multiple-choice questions and answers. The Quiz Management System (QMS) aims to streamline the process of creating, administering, and evaluating quizzes and exams. The system will cater to the needs of students, instructors, and administrators, ensuring a seamless and efficient examination process. By leveraging modern technology, the QMS will enhance the overall educational experience, providing a robust platform for academic assessments.

The database system for all lecturers in a university includes these information: Citizen ID, name, date of birth, gender, address and phone number. Lecturers assign quizzes and exams to students. *The value for Citizen ID is a string of 12 numbers as per national law, and the phone number is a string of 10 numbers.*

The database also holds students' information such as Student ID, Full name (surname and first name), Date of Birth, Gender, Address, Phone Number and Scores. A student can see and edit their personal information, as well as seeing their scores for past exams.

All lecturers and students may access the system using account information including username and password. All accounts are authenticated by HCMUT-SSO. There are also administrators who manage the system, each of them also have an account to login and access the system.

Each quiz goes with only one name, start date and time, and time limit. Quizzes contain multiple-choice questions with up to five possible answers, in which only one of them is correct.

II. COLLECTING AND ANALYZING REQUESTS

1. Requirement(s)

- Build a system for lecturers to create and assign multiple-choice exams and assign them to students.

2. Database specifications

- The QMS' database manages information such as: Account Information, Lecturers, Students, Quizzes, Scores.

- **Account Information** includes username and password. There are three types of accounts for Administrators, Lecturers and Students.

- **Lecturers** include: Citizen ID, Full name (surname and first name), Date of Birth, Gender, Address, Phone Number. Each lecturer has a unique account information and Citizen ID and can create and manage quizzes.

Requirements: Citizen ID is a string of 12 numbers per national law, and Phone Number is a string of 10 numbers.

- **Students** include: Student ID, Full name (surname and first name), Date of Birth, Gender, Address, Phone Number, Scores. Each student has a unique account information and Student ID. They can enter and submit quizzes. Scores are evaluated and are saved in a student's database.

Requirements: Student ID is a string of 7 numbers. The score is a decimal number and falls between 0 and 10.

- **Quizzes** include Question, up to five answers A, B, C, D, E, and Correct Answer. Lecturers may set start date and time as well as time limit for a quiz.

III. DATABASE DESIGN

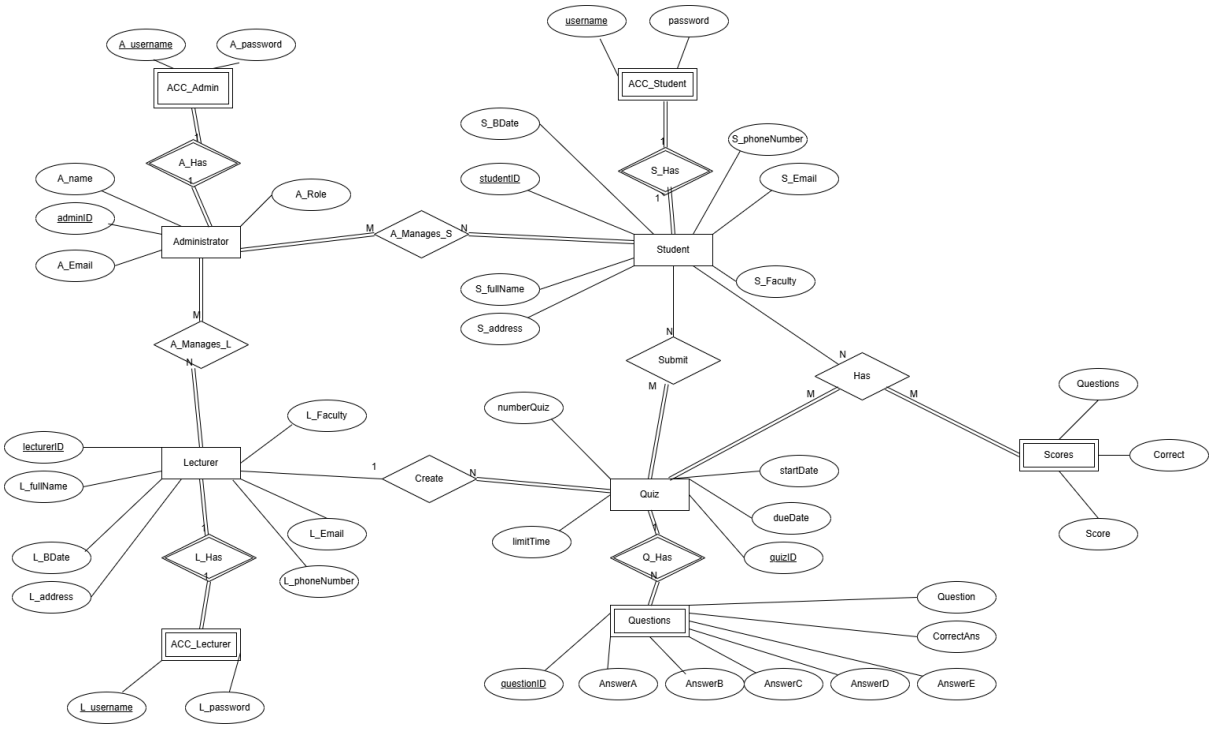
- Strong entities include:

+ **Student**{studentID, fullName, Bdate, gender, address, phoneNumber, email, Scores, faculty}
+ **Lecturer**{citizenID, name, Bdate, gender, address, phoneNumber, email, faculty}
+ **Administrator**{adminID, name, email, role}
+ **Quiz**{quizID, Questions, limitTime, startDate, endDate}

- Weak entities include:

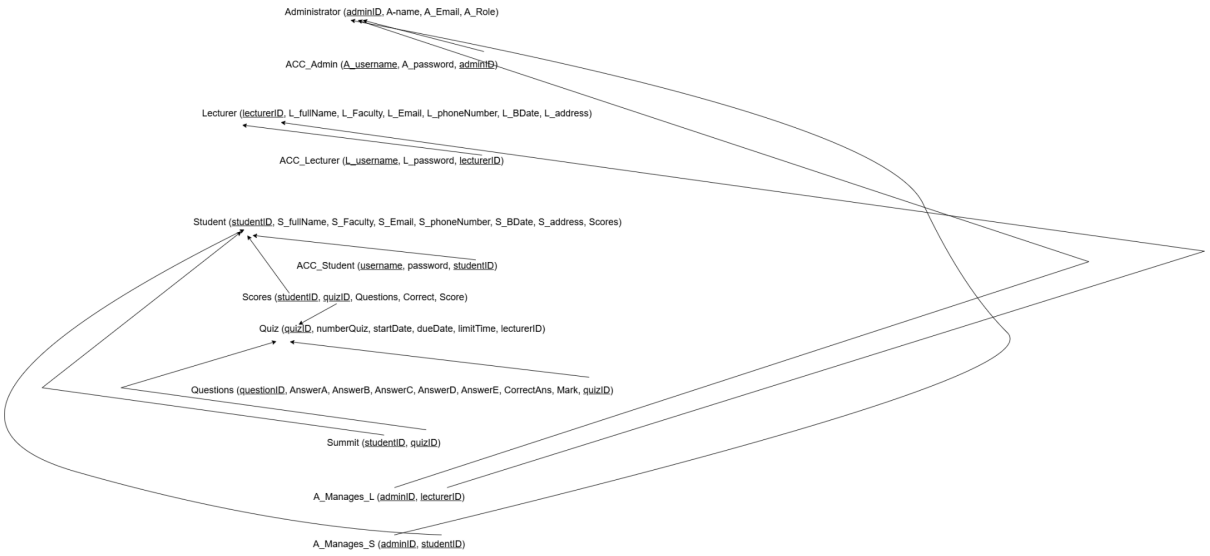
+ **Question** {quizID(FK), questionID, Question, AnswerA, AnswerB, AnswerC, AnswerD, AnswerE, CorrectAnswer}
  (A Question is belonged to the Quiz entity, a Question can not exist without being linked to a specific Quiz)
+ **Scores** {studentID, quizID, Questions, Correct, Score}
  (A score is belonged to the Student and Quiz entity, it cannot exist without being linked to a student and a particular Quiz)
+ **ACC_Lecturer**{userName, password, lecturerID}
+ **ACC_Student**{userName, password, studentID}
+ **ACC_Admin**{userName, password, adminID}
  (A Lecturer/Student/Admin has his/her own account to access the system, an account can be exist without being related to a lecture, student or admin)

1. Draw an Entity-Relationship Diagram (ERD)



Click here for the detailed image.

2. Mapping ERD Diagram into Relational Schema



Click here for the detailed image.

IV. DATABASE IMPLEMENTATION

1. Creating database tables

The following tables shall be created:

- Student
- Lecturer
- Administrator
- Quizzes
- Questions
- Scores
- Accounts
- ACC_Admin
- ACC_Lecturer
- ACC_Student

Creating the **Student** table:

```
CREATE TABLE Student (
    STUDENTID     VARCHAR(7) PRIMARY KEY,
    FULL_NAME     VARCHAR(30) NOT NULL,
    BDATE         DATE,
    GENDER        CHAR(1),
    ADDRESS       VARCHAR(60) NOT NULL,
    PHONE         VARCHAR(10),
    EMAIL         VARCHAR(30) NOT NULL,
    FACULTY       VARCHAR(10)
);
```

Creating the **Lecturer** table:

```
CREATE TABLE Lecturer (
    L_CITIZENID     VARCHAR(12) PRIMARY KEY,
    L_FULL_NAME     VARCHAR(30) NOT NULL,
    L_BDATE         DATE,
    L_GENDER        CHAR(1),
    L_ADDRESS       VARCHAR(60) NOT NULL,
    L_PHONE         VARCHAR(10),
    L_EMAIL         VARCHAR(30) NOT NULL,
    L_FACULTY       VARCHAR(10)
);
```

Creating the **Administrator** table:

```
CREATE TABLE Administrator (
    AD_ID       VARCHAR(10) PRIMARY KEY,
    AD_NAME     VARCHAR(30) NOT NULL,
    AD_EMAIL    VARCHAR(30) NOT NULL,
    AD_ROLE     VARCHAR(10)
);
```

Creating the **Quizzes** table:

```
CREATE TABLE Quizzes (
    QUIZ_ID       VARCHAR(6)  PRIMARY KEY,
    START_TIME    DATETIME,
    END_TIME      DATETIME,
    TIME_LIMIT    NUMBER(3)
    QUESTIONS     NUMBER(3)
);
```

Creating the **Questions** table:

```
CREATE TABLE Questions (
    QUIZ_ID       VARCHAR(6),
    QUESTION_ID   VARCHAR(6) PRIMARY KEY,
    QUESTION      VARCHAR(1000),
    ANSWER_A      VARCHAR(1000),
    ANSWER_B      VARCHAR(1000),
    ANSWER_C      VARCHAR(1000),
    ANSWER_D      VARCHAR(1000),
    ANSWER_E      VARCHAR(1000),
    CORRECT       CHAR(1),

    CONSTRAINT fk_QuizID FOREIGN KEY (QUIZ_ID)
       REFERENCES Quizzes(QUIZ_ID)
       ON DELETE CASCADE
);
```

Creating the **Scores** table:

```
CREATE TABLE Scores (
    STUDENT_ID    VARCHAR(7),
    QUIZ_ID       VARCHAR(6),
    QUESTIONS     INT(3),
    CORRECT       INT(3),
    SCORE         FLOAT,

    CONSTRAINT fk_StuID_Scores FOREIGN KEY (STUDENT_ID)
       REFERENCES Students(STUDENTID)
       ON DELETE CASCADE,
    CONSTRAINT fk_QuizID_Scores FOREIGN KEY (QUIZ_ID)
       REFERENCES Quizzes(QUIZ_ID)
       ON DELETE CASCADE,
```

```
);
```

Creating the **Accounts** table:

```
CREATE TABLE Accounts (
    USERNAME    VARCHAR(20) PRIMARY KEY,
    PASSWORD    VARCHAR(20)
);
```

Creating the **ACC_Student** table:

```
CREATE TABLE ACC_Student (
    USERNAME    VARCHAR(20) PRIMARY KEY,
    STUDENTID   VARCHAR( 7),

    CONSTRAINT fk_AccSTU FOREIGN KEY (USERNAME)
       REFERENCES Accounts(USERNAME)
       ON DELETE CASCADE
);
```

Creating the **ACC_Lecturer** table:

```
CREATE TABLE ACC_Lecturer (
    USERNAME    VARCHAR(20) PRIMARY KEY,
    CITIZENID   VARCHAR( 12),

    CONSTRAINT fk_AccLEC FOREIGN KEY (USERNAME)
       REFERENCES Accounts(USERNAME)
       ON DELETE CASCADE
);
```

Creating the **ACC_Admin** table:

```
CREATE TABLE ACC_Student (
    USERNAME    VARCHAR(20) PRIMARY KEY,

    CONSTRAINT fk_AccADMIN FOREIGN KEY (USERNAME)
       REFERENCES Accounts(USERNAME)
       ON DELETE CASCADE
);
```

After creating the tables, the script output displays:

```
Table STUDENT created.
Table LECTURER created.
Table ADMINISTRATOR created.
Table QUIZZES created.
Table QUESTIONS created.
Table SCORES created.
Table ACCOUNTS created.
```

```
Table ACC_STUDENT created.
Table ACC_LECTURER created.
Table ACC_ADMIN created.
```

2. Adding demo values into the tables

```
--------------------STUDENT
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
INSERT INTO STUDENT VALUES('2252001','NGUYEN VAN A','01-01-2004','M','HO CHI
MINH CITY','0123456789','A.NGUYEN@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252002','TRAN THI B','14-03-2004','F','HO CHI
MINH CITY','0123456790','B.TRAN@HCMUT.EDU.VN','SIM');
INSERT INTO STUDENT VALUES('2252003','DAO THANH C','11-01-2004','M','HO CHI
MINH CITY','0123456791','C.DAO@HCMUT.EDU.VN','GEOPET');
INSERT INTO STUDENT VALUES('2252004','LE THANH D','05-06-2004','M','HO CHI
MINH CITY','0123456792','D.LE@HCMUT.EDU.VN','DEE');
INSERT INTO STUDENT VALUES('2252005','NGUYEN VAN A','31-12-2004','M','HO CHI
MINH CITY','0123456793','A.NGUYEN@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252006','NGUYEN VAN E','15-08-2004','M','HO CHI
MINH CITY','0123456794','E.NGUYEN@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252007','NGUYEN VAN Y','09-01-2004','M','HO CHI
MINH CITY','0123456795','Y.NGUYEN@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252082','NGUYEN THANH BINH','13-03-2004','M','HO
CHI MINH CITY','0123456796','BINH.NGUYENTHANH@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252453','TRAN QUOC BAO
LONG','02-11-2004','M','HO CHI MINH
CITY','0904051758','LONG.TRAN041102@HCMUT.EDU.VN','CSE');
INSERT INTO STUDENT VALUES('2252808','DANG DUY TIEN','26-10-2004','M','HO CHI
MINH CITY','0123456798','TIEN.DANGDUYTIEN@HCMUT.EDU.VN','CSE');

--------------------LECTURER
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
INSERT INTO LECTURER VALUES('075080000001','DO THANH
THAI','06-06-1980','M','DONG NAI','0901234567','THAI@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075081000002','NGUYEN DUC
DUNG','12-03-1981','M','DONG NAI','0901234568','DUNGNN@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075082000003','LE THANH
VAN','09-10-1982','F','DONG NAI','0901234569','LTV@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075083000004','LE THANH
SACH','04-04-1983','M','DONG NAI','0901234570','SACHLE@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075084000005','NGUYEN VAN
BINH','01-09-1984','M','DONG
NAI','0901234571','BINHNGUYEN@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075085000006','TRAN THANH
BINH','13-12-1985','M','DONG NAI','0901234572','TTBINH@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075086000007','BUI DUC
BAO','08-09-1986','M','DONG NAI','0901234573','BDBAO@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075087000008','TRAN THI
HA','25-11-1987','F','DONG NAI','0901234574','HATRAN@HCMUT.EDU.VN','CSE');
INSERT INTO LECTURER VALUES('075088000009','CHAU VAN
LIEM','31-01-1988','M','DONG NAI','0901234575','CVL@HCMUT.EDU.VN','CSE');
```

```
INSERT INTO LECTURER VALUES('075089000010','PHAN TRONG
NHAN','05-12-1989','M','DONG
NAI','0901234576','PHANNHAN@HCMUT.EDU.VN','CSE');

--------------------ADMINISTRATOR
INSERT INTO ADMINISTRATOR
VALUES('0001','ADMIN_A','ABC@HCMUT.EDU.VN','ADMIN');
INSERT INTO ADMINISTRATOR VALUES('0101','ADMIN_B','ABB@HCMUT.EDU.VN','MOD');
INSERT INTO ADMINISTRATOR VALUES('0102','ADMIN_C','ABE@HCMUT.EDU.VN','MOD');
INSERT INTO ADMINISTRATOR VALUES('0103','ADMIN_D','ABF@HCMUT.EDU.VN','MOD');

----------------------QUIZZES
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
INSERT INTO QUIZZES VALUES('024001','01-11-2024','02-11-2024',30,10);
INSERT INTO QUIZZES VALUES('024002','03-11-2024','04-11-2024',30,10);
INSERT INTO QUIZZES VALUES('024003','05-11-2024','06-11-2024',30,10);

----------------------QUESTIONS
INSERT INTO QUESTIONS
VALUES('024001','000001','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000002','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000003','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000004','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000005','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000006','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000007','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000008','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000009','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024001','000010','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000011','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
```

```
INSERT INTO QUESTIONS
VALUES('024002','000012','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000013','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000014','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000015','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000016','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000017','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000018','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000019','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024002','000020','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000021','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000022','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000023','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000024','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000025','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000026','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000027','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000028','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
```

```
INSERT INTO QUESTIONS
VALUES('024003','000029','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');
INSERT INTO QUESTIONS
VALUES('024003','000030','QUESTION','ANSWER_A','ANSWER_B','ANSWER_C','ANSWER_
D','ANSWER_E','A');

----------------------SCORES
INSERT INTO SCORES VALUES('2252001','024001',10,8,8.0);
INSERT INTO SCORES VALUES('2252001','024002',10,9,9.0);
INSERT INTO SCORES VALUES('2252001','024003',10,8,8.0);
INSERT INTO SCORES VALUES('2252453','024001',10,9,9.0);
INSERT INTO SCORES VALUES('2252453','024002',10,10,10.0);
INSERT INTO SCORES VALUES('2252453','024003',10,8,8.0);
INSERT INTO SCORES VALUES('2252003','024001',10,7,7.0);
INSERT INTO SCORES VALUES('2252004','024001',10,10,10.0);

----------------------ACCOUNTS
INSERT INTO ACCOUNTS VALUES('STUDENT','ABCD1234');
INSERT INTO ACCOUNTS VALUES('LECTURER','ABCD1234');
INSERT INTO ACCOUNTS VALUES('ADMIN','ABCD1234');
```

3. Demonstrate desired data using SELECT

List all students:

```
--List all students
SELECT * FROM STUDENT;
```

	STUDENTID	FULL_NAME	BDATE	GENDER	ADDRESS	PHONE	EMAIL	FACULTY
1	2252001	NGUYEN VAN A	01-01-2004	M	HO CHI MINH CITY	0123456789	A.NGUYEN@HCMUT.EDU.VN	CSE
2	2252002	TRAN THI B	14-03-2004	F	HO CHI MINH CITY	0123456790	B.TRAN@HCMUT.EDU.VN	SIM
3	2252003	DAO THANH C	11-01-2004	M	HO CHI MINH CITY	0123456791	C.DAO@HCMUT.EDU.VN	GEOPET
4	2252004	LE THANH D	05-06-2004	M	HO CHI MINH CITY	0123456792	D.LE@HCMUT.EDU.VN	DEE
5	2252005	NGUYEN VAN A	31-12-2004	M	HO CHI MINH CITY	0123456793	A.NGUYEN@HCMUT.EDU.VN	CSE
6	2252006	NGUYEN VAN E	15-08-2004	M	HO CHI MINH CITY	0123456794	E.NGUYEN@HCMUT.EDU.VN	CSE
7	2252007	NGUYEN VAN Y	09-01-2004	M	HO CHI MINH CITY	0123456795	Y.NGUYEN@HCMUT.EDU.VN	CSE
8	2252082	NGUYEN THANH BINH	13-03-2004	M	HO CHI MINH CITY	0123456796	BINH.NGUYENTHANH@HCMUT.EDU.VN	CSE
9	2252453	TRAN QUOC BAO LONG	02-11-2004	M	HO CHI MINH CITY	0904051758	LONG.TRAN041102@HCMUT.EDU.VN	CSE
10	2252808	DANG DUY TIEN	26-10-2004	M	HO CHI MINH CITY	0123456798	TIEN.DANGDUYTIEN@HCMUT.EDU.VN	CSE

List all lecturers in a university:

```
--List all lecturers
SELECT * FROM LECTURER;
```

	L_CITIZENID	L_FULL_NAME	L_BDATE	L_GENDER	L_ADDRESS	L_PHONE	L_EMAIL	L_FACULTY
1	075080000001	DO THANH THAI	06-06-1980	M	DONG NAI	0901234567	THAI@HCMUT.EDU.VN	CSE
2	075081000002	NGUYEN DUC DUNG	12-03-1981	M	DONG NAI	0901234568	DUNGNN@HCMUT.EDU.VN	CSE
3	075082000003	LE THANH VAN	09-10-1982	F	DONG NAI	0901234569	LTV@HCMUT.EDU.VN	CSE
4	075083000004	LE THANH SACH	04-04-1983	M	DONG NAI	0901234570	SACHLE@HCMUT.EDU.VN	CSE
5	075084000005	NGUYEN VAN BINH	01-09-1984	M	DONG NAI	0901234571	BINHNGUYEN@HCMUT.EDU.VN	CSE
6	075085000006	TRAN THANH BINH	13-12-1985	M	DONG NAI	0901234572	TTBINH@HCMUT.EDU.VN	CSE
7	075086000007	BUI DUC BAO	08-09-1986	M	DONG NAI	0901234573	BDBAO@HCMUT.EDU.VN	CSE
8	075087000008	TRAN THI HA	25-11-1987	F	DONG NAI	0901234574	HATRAN@HCMUT.EDU.VN	CSE
9	075088000009	CHAU VAN LIEM	31-01-1988	M	DONG NAI	0901234575	CVL@HCMUT.EDU.VN	CSE
10	075089000010	PHAN TRONG NHAN	05-12-1989	M	DONG NAI	0901234576	PHANNHAN@HCMUT.EDU.VN	CSE

List all questions from a quiz:

```
--List all questions from Quiz ID 024001
SELECT * FROM QUESTIONS
WHERE QUIZ_ID = '024001';
```

	QUIZ_ID	QUESTION_ID	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	CORRECT
1	024001	000001	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
2	024001	000002	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
3	024001	000003	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
4	024001	000004	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
5	024001	000005	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
6	024001	000006	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
7	024001	000007	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
8	024001	000008	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
9	024001	000009	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A
10	024001	000010	QUESTION	ANSWER_A	ANSWER_B	ANSWER_C	ANSWER_D	ANSWER_E	A

List scores of students:

```
--List all scores from students
SELECT * FROM SCORES
```

	STUDENT...	QUIZ_ID	QUESTIONS	CORRECT	SCORE
1	2252001	024001	10	8	8
2	2252001	024002	10	9	9
3	2252001	024003	10	8	8
4	2252453	024001	10	9	9
5	2252453	024002	10	10	10
6	2252453	024003	10	8	8
7	2252003	024001	10	7	7
8	2252004	024001	10	10	10

List all students within a faculty, for example, all students in Faculty of Computer Science and Engineering (CSE):

```
--List all CSE students
SELECT STUDENTID, FULL_NAME, EMAIL, FACULTY FROM STUDENT
WHERE FACULTY = 'CSE';
```

STUDENT	FULL_NAME	EMAIL	FACULTY
2252001	NGUYEN VAN A	A.NGUYEN@HCMUT.EDU.VN	CSE
2252005	NGUYEN VAN A	A.NGUYEN@HCMUT.EDU.VN	CSE
2252006	NGUYEN VAN E	E.NGUYEN@HCMUT.EDU.VN	CSE
2252007	NGUYEN VAN Y	Y.NGUYEN@HCMUT.EDU.VN	CSE
2252082	NGUYEN THANH BINH	BINH.NGUYENTHANH@HCMUT.EDU.VN	CSE
2252453	TRAN QUOC BAO LONG	LONG.TRAN041102@HCMUT.EDU.VN	CSE
2252808	DANG DUY TIEN	TIEN.DANGDUYTIEN@HCMUT.EDU.VN	CSE