

**ESKİŞEHİR TECHNICAL UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING**

**2023-2024 SPRING SEMESTER**

**BIM312**

**DATABASE MANAGEMENT SYSTEMS**

**TERM PROJECT**

PHASE-2

**G63-PNo3: Online Examination System**

**Poyraz ÖZBEĞ poyrazozbeg@ogr.eskisehir.edu.tr**

**CHANGINGS:**

**+HAS RELATION BETWEEN LECTURES AND EXAMINATIONS WAS 1:1, NOW IT BECAME 1:N DUE TO 2 EXAM TYPE (FINAL AND MIDTERM)**

**+ER DIAGRAM CHANGED AND HAD SOME NEW ENTITIES DUE TO NEEDING**

**FEEDBACK NOTES:**

**+I ADDED ‘QUESTIONS’ TABLE TO HAVE QUESTIONS TO EXAMINATION**

**+DUE TO ‘ADMINS’ AND ‘STUDENTS’ WILL HAVE 2 DIFFERENT LOGIN PANEL I DID NOT MAKE ANY COMMON** AUTHORIZATION **TABLE**

**+IF THERE IS ANY PROBLEM PLEASE MAKE FEEDBACK FOR PHASE 3. THANK YOU FOR READING, PLEASE READ THE BRIDGE BELOW TO SEE PHASE 2 DEVELOPS.**

**page no.1**

**Phase 1-Step 1:Requirement Analysis(UPDATED) 3**

**Requirements and Functionalities 3**

**Some of the Possible Queries: 5**

**Phase 1 – Step 2: Entity-Relationship (E/R) Diagram(UPDATED) 5**

**Entity Sets: 5**

**Attributes & Primary Keys: 5**

**Relationships: 6**

**E/R Diagram 7**

**Phase 2 – Database Design(PHASE 2 STEPS) 8**

**The Relational Schema 8**

**Database Schema (SQL Codes) 9**

**SQL EER DIAGRAM 14**

**page no.2**

# Phase 1-Step 1:Requirement Analysis

Online Examination System is in a very important position, and it has become even more important, especially as education increasingly progresses online. Since this system has a lot of data for students, courses and grades, a database system is very important for this system.

I think that developing this database system will be a very efficient and permanent way of learning. I also believe it will be a prototype for future projects.

While I was planning how to create information and requirements for my database, after thinking about the functions of this database and investigeting some university examination systems and seeing these universities as customers, I came up with the following requirements and functions.

## Requirements and Functionalities

This online examination system database stores and keeps track of the students, lectures(courses), examinations, admins and main system that are organized by the admins each semester. The system will be used by several different users (with different authorities/roles) like a student who can see the list of exams and lectures or examination. Or a teacher (as one of the admin roles) for changing grades, determining date of exams. The system also be used by system admin (as another one of the admin roles) managing system, creating new student logins and determining their password. Additionally, there are many functions that will be included in the system in future stages.

**page no.3**

1. ‌ The system stores each student's and admin's unique ID, name, e-mail, and system password. For admins there is also roles that system stores in database.
2. Admins manage the system. There are two different role for administration. First and most privileged one is the system admin, system admins updates system on every semester, they create or delete students login as it needs, determine or change passwords, organize the lectures, exam dates and every users information.
3. Second role of administration is the teacher, they have only 3 functions on the system, they can see and change grades of all students on their lectures and organize exam date if its necessary.
4. The students can login system web page with email and password that administration gave before. They can see date of coming exams and grades on the lecture part of website.
5. ‌Every lecture has its exam and grade information on the system. Class that lesson is taught and credits of the course also in the giving information.
6. ‌ Examinations system stores exam names, exam dates and an unique ID for each exam.
7. System web page also be stored with its information such as url and a unique id. Because system may can used for different schools on same database. I am not sure but this might be more efficient for economical scale.

**page no.4**

## Some of the Possible Queries:

1. What is the average of exam grade of lecture?
2. What is the current number of students of the lecture?
3. Which student has the maximum gpa?
4. Which admin has the role of changing grades?
5. Which exam the hardest one based on compare average of exam grades?
6. Which student's grade in the lecture (course) is below or above average?
7. Provide a ranking of students based on grades.

# Phase 1 – Step 2: Entity-Relationship (E/R) Diagram

## Entity Sets:(UPDATED)

* ADMIN
* SYSTEMWEBPAGE
* LECTURES
* STUDENTS
* EXAMINATIONS
* QUESTIONS

## Attributes & Primary Keys:(UPDATED)

* ADMIN (Admin\_ID, Admin\_Name, Admin\_Email, Admin\_Password, Role)
* SYSTEMWEBPAGE (URL, System\_ID)
* LECTURES (Lecture\_ID, Lecture\_Name, Class, Credits)
* STUDENTS (Student\_ID, Student\_Name, Student\_Email, Student\_Password)
* EXAMINATIONS (Exam\_ID, Exam\_Name)
* QUESTIONS(Question\_ID, Question\_Text, Question\_Point, {Answer\_Options}, Correct\_Answer)

**page no.5**

## Relationships(UPDATED):

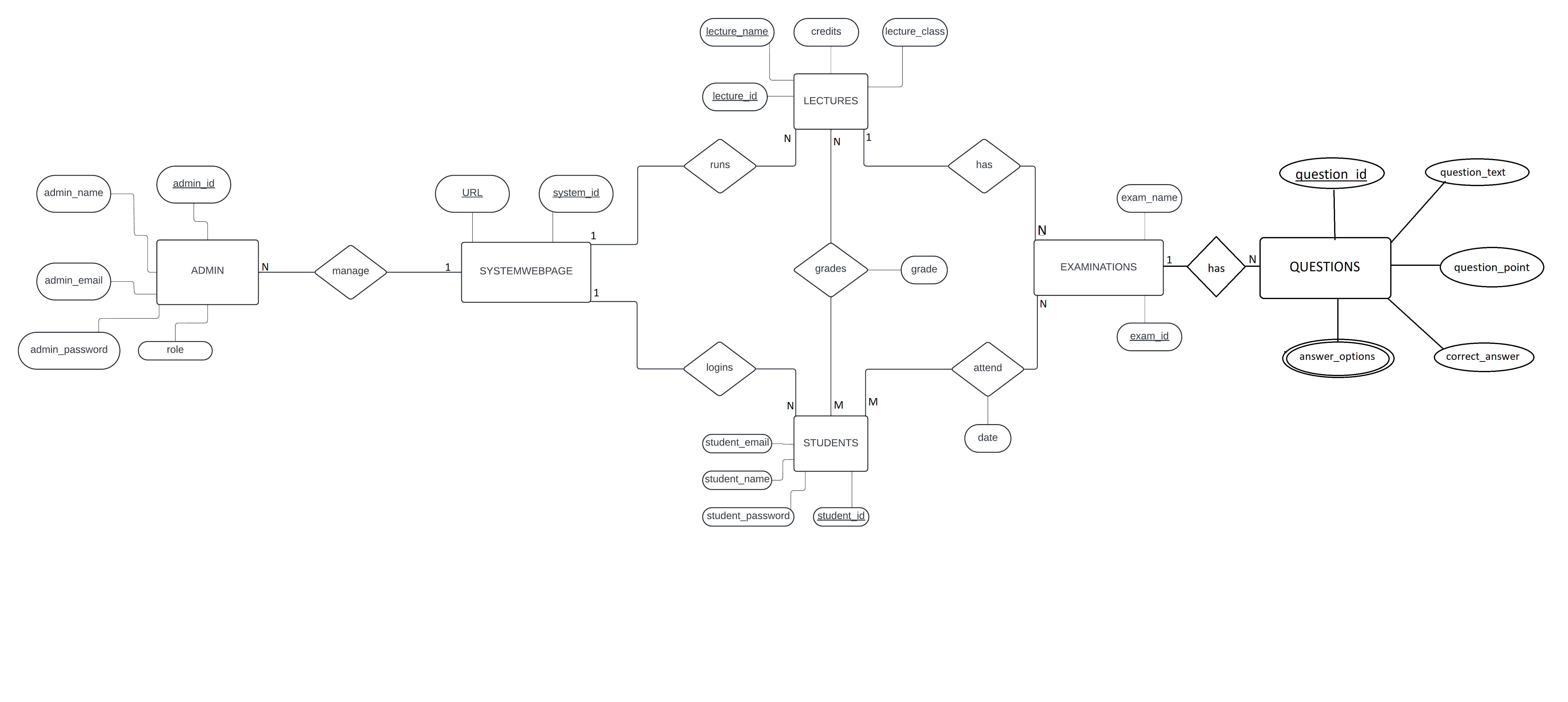
* **HAS,** a 1:N relationship type between LECTURES and EXAMINATIONS. LECTURES and EXAMINATIONS participations are both partial; one lecture can have one exam.
* **RUNS**, a 1:N relationship type between SYSTEMWEBPAGE and LECTURES. Both participation is partial. SYSTEMWEBPAGE runs LECTURES functions on application.
* **LOGINS**, a 1:N relationship type between STUDENTS and SYSTEMWEBPAGE. STUDENTS and STUDENTS participations are partial; this indicates that several students can login in one system.
* **ATTENDS**, a M:N relationship type between STUDENTS and EXAMINATIONS. STUDENTS and EXAMINATIONS participations are both partial; several students can attend several exams. Relation also have an attribute that points date of exam. It will be a table that has two FK and an attribute.
* **GRADES**, a M:N relationship type between STUDENTS and LECTURES. STUDENTS and LECTURES participations are both partial; several students can have grades from several lectures. Relation also have an attribute that points grade of exam for each student for their lectures. It will be a table that has two FK and an attribute.
* **MANAGES**, a 1:N relationship type between ADMIN and SYSTEMWEBPAGE, ADMIN and SYSTEMWEBPAGE participations are both partial; several admins can manage one system.

**page no.6**

## [**E/R Diagram**](https://github.com/RadwanH/University-Club-Membership/blob/main/Diagrams/ER-Diagram.drawio.png?raw=true) (UPDATED)

**CLICK TO SEE HD IMAGE:**

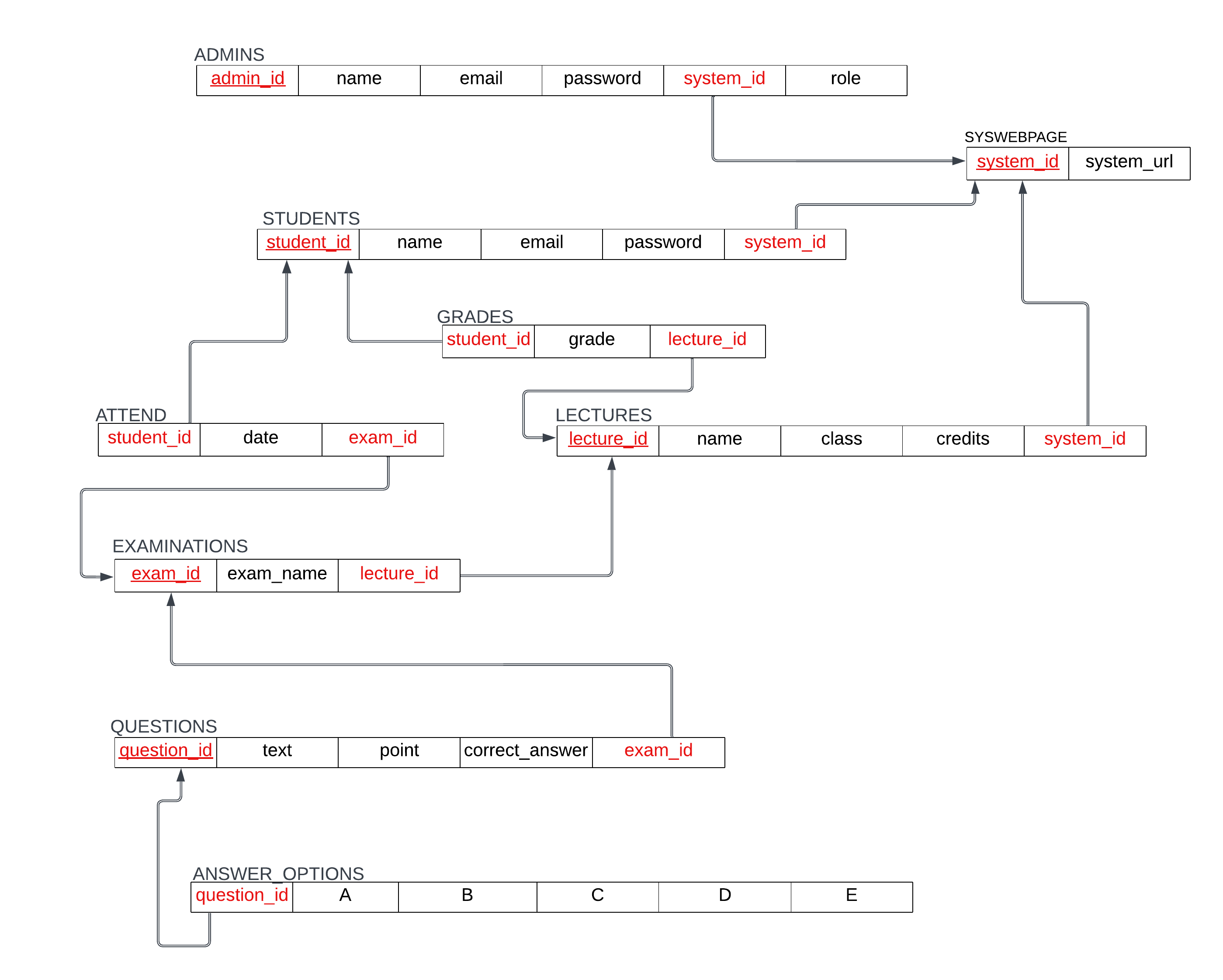
**https://github.com/poyrazozbeg1/OnlineExaminationSystemDatabase/blob/main/diagrams**

****

**page no.7**

# **Phase 2: Database Design**

# **The Relational Schema:**



**page no.8**

**Database Schema(SQL Codes):**

**NOTE: I USED ORACLE SQL TO CODING THIS PHASE**

**CLICK TO SEE SQL FILES:**

**https://github.com/poyrazozbeg1/OnlineExaminationSystemDatabase/tree/main/SQL**

**---------------------------------------------**

**--PHASE 1: CREATING TABLES**

**CREATE TABLE ADMINS (**

**ADMIN\_ID NUMBER(5) NOT NULL, --PK**

**ADMIN\_NAME VARCHAR2(40) NOT NULL,**

**ADMIN\_EMAIL VARCHAR2(40) NOT NULL,**

**ADMIN\_PASSWORD VARCHAR2(40) NOT NULL,**

**ADMIN\_ROLE VARCHAR2(20) NOT NULL CHECK (ADMIN\_ROLE IN ('SYSTEM\_ADMIN', 'TEACHER')),**

**SYSTEM\_ID NUMBER(5)--FK**

**);**

**CREATE TABLE SYSWEBPAGE (**

**SYSTEM\_ID NUMBER(5), --PK**

**SYSTEM\_URL VARCHAR2(40) NOT NULL**

**);**

**CREATE TABLE LECTURES (**

**LECTURE\_ID NUMBER(10) NOT NULL, --PK,**

**LECTURE\_NAME VARCHAR2(40) NOT NULL,**

**LECTURE\_CLASS VARCHAR2(10) NOT NULL,**

**CREDITS NUMBER NOT NULL,**

**SYSTEM\_ID NUMBER(5) --FK**

**);**

**CREATE TABLE EXAMINATIONS(**

**EXAM\_ID NUMBER(10) NOT NULL, --PK**

**EXAM\_NAME VARCHAR2(40) NOT NULL,**

**LECTURE\_ID NUMBER(10) NOT NULL**

**);**

**CREATE TABLE STUDENTS(**

**STUDENT\_ID NUMBER(20) NOT NULL, --PK**

**STUDENT\_NAME VARCHAR2(40) NOT NULL,**

**STUDENT\_EMAIL VARCHAR2(40) NOT NULL,**

**STUDENT\_PASSWORD VARCHAR2(40) NOT NULL,**

**SYSTEM\_ID NUMBER(5) --FK**

**);**

**CREATE TABLE GRADES(**

**GRADE NUMBER(3),**

**STUDENT\_ID NUMBER(20) NOT NULL,--FK**

**LECTURE\_ID NUMBER(10) NOT NULL**

**);**

**CREATE TABLE ATTEND(**

**EXAM\_ATTENDING\_DATE DATE NOT NULL,**

**STUDENT\_ID NUMBER(20) NOT NULL, --FK**

**EXAM\_ID NUMBER(10) NOT NULL --FK**

**);**

**CREATE TABLE QUESTIONS(**

**QUESTION\_ID NUMBER(20) NOT NULL,**

**QUESTION\_TEXT VARCHAR2(999),**

**QUESTION\_POINT NUMBER(2),**

**CORRECT\_ANSWER VARCHAR2(50),**

**EXAM\_ID NUMBER(10) NOT NULL --FK**

**);**

**CREATE TABLE ANSWER\_OPTIONS(**

**A\_ANSWER VARCHAR2(50),**

**B\_ANSWER VARCHAR2(50),**

**C\_ANSWER VARCHAR2(50),**

**D\_ANSWER VARCHAR2(50),**

**E\_ANSWER VARCHAR2(50),**

**QUESTION\_ID NUMBER(20) NOT NULL**

**);**

**--PHASE 2: ADDING PRIMARY KEYS**

**ALTER TABLE ADMINS ADD(**

**CONSTRAINT ADMIN\_PK PRIMARY KEY(ADMIN\_ID)**

**);**

**ALTER TABLE SYSWEBPAGE ADD(**

**CONSTRAINT SYSTEM\_PK PRIMARY KEY(SYSTEM\_ID)**

**);**

**ALTER TABLE LECTURES ADD(**

**CONSTRAINT LECTURE\_PK PRIMARY KEY(LECTURE\_ID)**

**);**

**ALTER TABLE EXAMINATIONS ADD(**

**CONSTRAINT EXAM\_PK PRIMARY KEY(EXAM\_ID)**

**);**

**ALTER TABLE STUDENTS ADD(**

**CONSTRAINT STUDENTS\_PK PRIMARY KEY(STUDENT\_ID)**

**);**

**ALTER TABLE QUESTIONS ADD(**

**CONSTRAINT QUESTION\_PK PRIMARY KEY(QUESTION\_ID)**

**);**

**--PHASE 2: ADDING FOREIGN KEYS**

**ALTER TABLE ADMINS ADD(**

**CONSTRAINT ADMIN\_SYSTEM\_FK FOREIGN KEY(SYSTEM\_ID) REFERENCES SYSWEBPAGE(SYSTEM\_ID)**

**);**

**ALTER TABLE LECTURES ADD(**

**CONSTRAINT LECTURES\_SYSTEM\_FK FOREIGN KEY(SYSTEM\_ID) REFERENCES SYSWEBPAGE(SYSTEM\_ID)**

**);**

**ALTER TABLE STUDENTS ADD(**

**CONSTRAINT STUDENTS\_SYSTEM\_FK FOREIGN KEY(SYSTEM\_ID) REFERENCES SYSWEBPAGE(SYSTEM\_ID)**

**);**

**ALTER TABLE GRADES ADD(**

**CONSTRAINT STUDENT\_ID\_FK FOREIGN KEY(STUDENT\_ID) REFERENCES STUDENTS(STUDENT\_ID),**

**CONSTRAINT LECTURE\_ID\_FK FOREIGN KEY(LECTURE\_ID) REFERENCES LECTURES(LECTURE\_ID)**

**);**

**ALTER TABLE ATTEND ADD(**

**CONSTRAINT STUDENT\_ATTENDING\_FK FOREIGN KEY(STUDENT\_ID) REFERENCES STUDENTS(STUDENT\_ID),**

**CONSTRAINT EXAM\_ATTENDING\_FK FOREIGN KEY(EXAM\_ID) REFERENCES EXAMINATIONS(EXAM\_ID)**

**);**

**ALTER TABLE EXAMINATIONS ADD(**

**CONSTRAINT LECTURE\_EXAM\_FK FOREIGN KEY(LECTURE\_ID) REFERENCES LECTURES(LECTURE\_ID)**

**);**

**ALTER TABLE QUESTIONS ADD(**

**CONSTRAINT EXAM\_QUESTION\_FK FOREIGN KEY(EXAM\_ID) REFERENCES EXAMINATIONS(EXAM\_ID)**

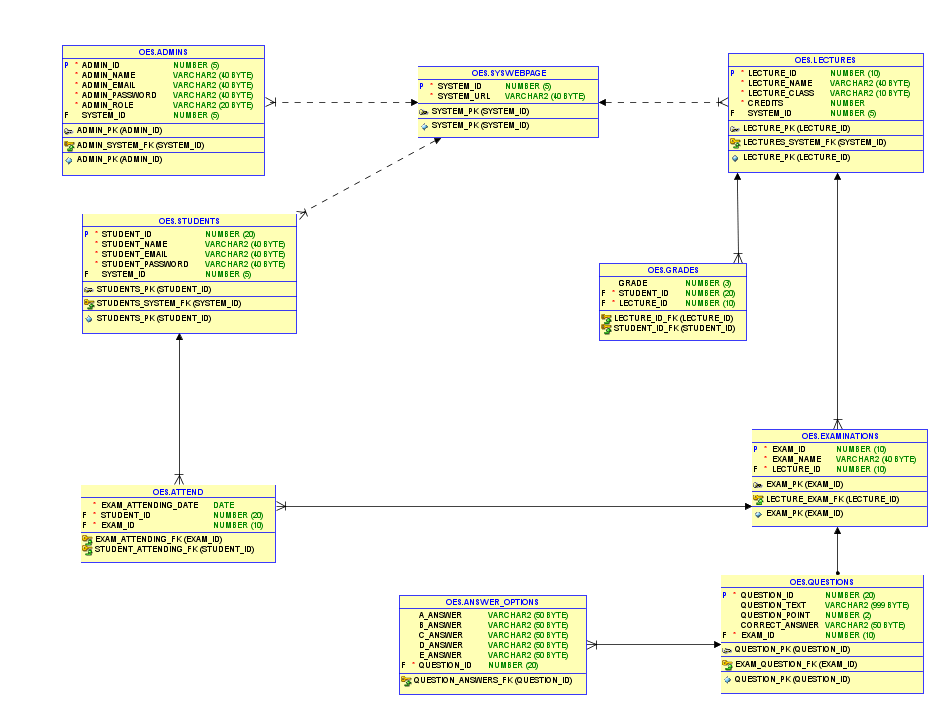
**);**

**ALTER TABLE ANSWER\_OPTIONS ADD(**

**CONSTRAINT QUESTION\_ANSWERS\_FK FOREIGN KEY(QUESTION\_ID) REFERENCES QUESTIONS(QUESTION\_ID)**

**);**

**SQL EER DIAGRAM:**

****

**page no.14**