AIU Coursera Management System

The AIU Coursera Management System is a platform with a database management system, created to make it easier to include Coursera courses into the AIU (Alamein International University) curriculum.

Alamein International University
Computer Science Engineering
CSE221 Database

To Facilitate The Work Of The Project, We Divided The Project Into Phases:

Phase 1: Project Description
Business Rules
Potential Queries

Phase 2: Entity Relationship Diagram (ERD),(EERD)

Phase 3: Normalized Relational Schema

Phase 4: DDL Schema and DML Population

Phase 5: Query answers
Different views
Procedure

Alamein International University
Computer Science Engineering
CSE221 Database

Phase 1

- **Project Description:**

The application can allow AIU users (students, staff, administrators, visitors, tutors) to sign into several courses by many popular universities around the world and explore additional courses in many different fields.

Throughout every semester (Fall/Spring/Summer +Year) we need to keep track of AIU courses, their field (CSE, Pharmacy, Dentistry, Business, Public Health, Art and Design), their course code, their semester in the study plan, their main instructor this semester and his chosen Coursera course(s) for the course and the number of expected students in his AIU course (so also in Coursera) this semester. Each instructor can choose more than one Coursera course for each of his AIU courses.

For each Coursera course (also each guided project), we need to store its name, its instructor name, its link (the last part of the link is unique, it is called the course slug), the organization or university which offers this course, its number of weeks, number of hours, and, if it is part of a specialization (only for courses, not for guided projects), the link of the specialization. We need to know if a certain AIU course used Coursera before in one of the previous semesters, and if yes, which course(s) were used (with all their info).

Every AIU field has a program on Coursera where the students can be invited according to their field (CSE, Pharmacy, ...all AIU Fields) and each of these programs has an admin link for the instructors(who are invited as admins) and a learner link for students (who are invited as learners). We collect a list of all instructors from each field every semester to add them as admins on their programs and also as admins and learners on the Instructor's program (another Coursera program created for instructors).

- **Business Rules:**

1. The instructor can choose certain modules from Coursera and map them to certain weeks of his course for their students.

2. Weekly updates on student enrollment, completion, progress, and feedback are mandatory.

3. Every AIU field has a program on Coursera where the students can be invited according to their field (CSE, Pharmacy, ...all AIU Fields) and each of these programs has an admin link for the instructors (who are invited as admins) and a learner link for students (who are invited as learners).

4. A Students can be enrolled in one or more AIU and Coursera courses, with Coursera courses accommodating multiple enrolled students.

5. Each instructors can teach many AIU courses and many AIU courses can be taught by many instructors .

6. Only one administrator can invite instructors to instructor Coursera Programs and invite students to student Coursera Programs .

7. Each coursera course can have one or more module.

8. Instructors have the flexibility to select multiple Coursera courses for their AIU courses.

9. The Total Coursera licenses are tracked and managed, with limits enforced on simultaneous student usage.

10. Only AIU students can access the application system and Students from different faculties cannot access other field's courses for free.

11. The system can be used by the main Coursera coordinator (can see everything), the coordinators from the different fields (can see their fields only) and the instructors (can see their courses only and Coursera courses which were used before in their courses).

- **Potential Queries:**

1. Who is The most student who got courses certificates?

2. Students who haven’t finished their coursera courses.

3. Students who registered for a specific Aiu course in a specific semester.

4. Can you Retrieve a list of instructors and their corresponding AIU courses along with the Coursera courses?

5. Retrieve students who enrolled in specific coursera course.

6. Students that need to be added and removed from coursera program.

7. Display the enrollment status of students in a particular Coursera Courses, including their email, name and ID?

8. Mapping Aiu Courses in a spacific semester and spacific field.

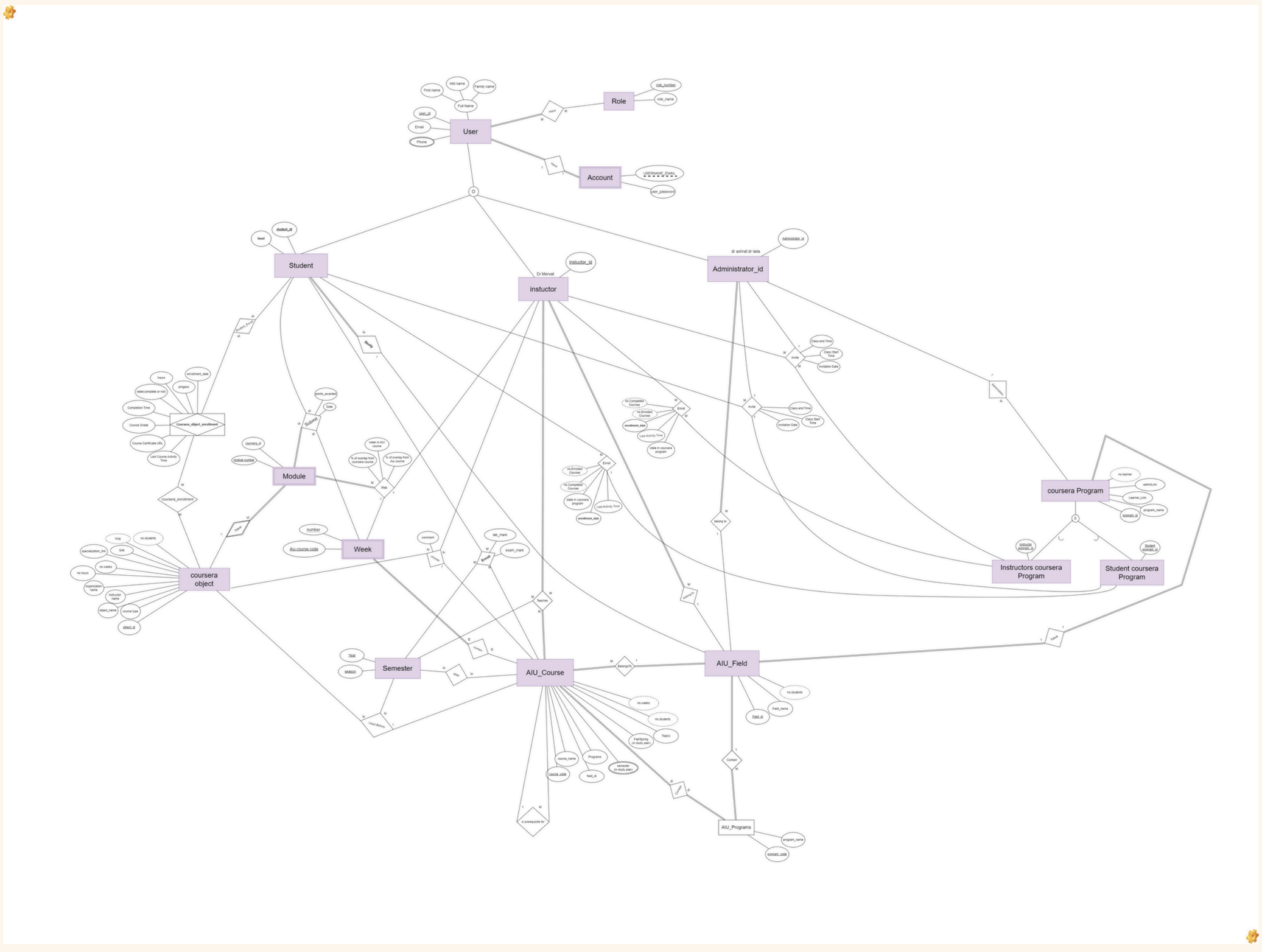
9. Mapping into our catalog and also calculate the overlap percentage from both sides: how much is covered from the AIU course and how much from the Coursera course.

10. Collect a list of all instructors from each field every semester to add them as admins on their programs and also as admins and learners on the Instructors program .

Alamein International University
Computer Science Engineering
CSE221 Database

Phase 2

- **Enhanced Entity Relationship Diagram (EERD)**



Alamein International University
Computer Science Engineering
CSE221 Database

Phase 3

- **Normalized Relational Schema:**

User (user_id [char(8)] , user_first_name[varchar], user_mid_name [varchar],user_family_name [varchar], email [varchar]);
  - user_id → user_first_name, user_mid_name, user_family_name, email

User_Contact (user_id [char(8)] , phone [int])
• user_id fk reference User (user_id)

User_Accounts (user_id [char(8)] , user-name-email [varchar] , user-password [varchar])
• user_id fk reference User (user_id)
• user-name-email fk reference User (email)
♦ user_id, user-name-email → user-password

Roles (role_number [int], role_name [varchar])
♦ role_number → role_name

User_Roles (user_id [char(8)] , role_number [varchar])
• user_id fk reference User (user_id)

Administrators (Administrator_id [char(8)])
• Administrator_id fk reference User (user_id)

Instractor(instractor_id[int],fild_id)
instractor_id fk reference User (user_id)
Fild_id fk reference User fild

Enroll(instractor_id, instractor_program_id,enrollment [date] [date,lastactivetime[date],state in courseraprogram[varchar],no_of_compeleted courses[int],no_of_enrolledcourses[int])
instractor_id fk reference instractor instractor_program_id fk reference instractor_coursera_program

Alamein International University
Computer Science Engineering
CSE221 Database

User (user_id [char(8)] , user_first_name[varchar], user_mid_name [varchar],user_family_name [varchar], email [varchar]);

- user_id → user_first_name, user_mid_name, user_family_name, email

Teaches(instractor_id,course_code)
instractor_id fk reference instractor Course_code fk reference aiu_coursera

Student (student_id [char (8)], field_id [int], level [int])
• student_id fk reference User(user_id)
• field_id fk reference Aiu_field(field_id)
♦ student_id → field_id, level

Student_program_Invitation (student_id [char (8)], admin_id [char (8)], programe_id [int], invitation_Date[Date], Class_Start_Time[date], Class_end_Time[date])
• student_id fk reference Student(student_id)
• admin_id fk reference Main_coordinator_in_field (coordinator_id)
• programe_id fk reference Student_coursera_programe(Student_program_id)
♦ student_id, admin_id, programe_id → invitation_Date, Class_Start_Time, Class_end_Time

Student_Program_enrollment (student_id [char (8)], program_id [int],enrollment_date [Date], Last Activity Time [Date], state (deleted or no) [varchar])
• student_id fk reference Student(student_id)
• program_id fk reference Student_Coursera_Program(Student_program_id)
♦ (student_id, program_id → enrollment_date, Last Activity Time, state

Coursera_Program (program_id [int], program_name [varchar], adminLink [varchar], learner_link [varchar], admin_id [char (8)], Access_by_id [char (8)],administrator_id)

• admin_id fk reference Main_coordinator_in_field(coordinator_id)
• Access_by_id fk reference Administrators(adminisstrators_id)
administrator_id fk reference administrator
♦ program_id → program_name, adminLink, learner_link, admin_id, Access_by_id

instractor_Coursera_Program (instractor_program_id [int], administrator_id)
• istractor_program_id fk reference Coursera_program (programe_id)
administrator_id fk reference administrator

Student_Coursera_Program (Student_program_id [int])
• Student_program_id fk reference Coursera_program(programe_id)

Coursera_object (object_id [int] , object_name [varchar], course_Type [varchar], instructor_name [varchar], offered_by [varchar], no.Of_hours[int], no.Of_weeks [int], object_link [varchar], specialization_link [varchar]);
♦ object_id → object_name, course_Type, instructor_name, offered_by, no.Of_hours, no.Of_weeks, object_link, specialization_link
✓ Coursera_enrollment (student_id [char(8)], coursera_objectID [int], enrollment_date [date], progress [double], hours [double], State (finish or no) [varchar], Completion_Time[date], Course_Grade [double], Coursera_certificate_URL[varchar], Last_active_time[date])
• student_id fk reference Student(Student_id)
• coursera_objectID fk reference Coursera_object(object_id)
♦ student_id, coursera_objectID → enrollment_date, progress, hours, State, Completion_Time, Course_Grade, Coursera_certificate_URL, Last_active_time

Coursera_Used_Before (coursera_object_ID [int], Aiu_cousre_code [varchar]); • coursera_object_ID fk reference Coursera_object(object_id) • Aiu_cousre_code fk reference Aiu_course(aiu_course_code)

Aiu_course (aiu_course_code [varchar], course_name [varchar], credit_hour[int], Field_id [int], Programs [varchar], semester (in study plan) [int], Fall /Spring (in study plan) [varchar], topics [varchar],preq_aiu_course_code [varchar])

• Field_id fk reference Aiu_Field (Field_id)
• Instructor_id reference Academic (Academic_id)
• preq_aiu_course_code reference Aiu_course (aiu_course_code)
♦ aiu_course_code → course_name, credit_hour, Programs, semester, Fall /Spring, topics, preq_aiu_course_code

Teaches (Academic_id [char(8)], Aiu_course_code[varchar], semester[int])
• Academic_id reference Academic (Academic_id)
• Aiu_course_code reference Aiu_course(aiu_course_code)
♦ Academic_id, Aiu_course_code → semester

Aiu_courses_enrollment (student_id [char(8)], aiu_course_code [varchar], lab_marks[int], exam_marks[int])
• student_id fk reference Student (Student_id)
• aiu_course_code fk reference Aiu_course(aiu_course_code)
♦ student_id → aiu_course_code, lab_marks, exam_marks

Alamein International University
Computer Science Engineering
CSE221 Database

Phase 4

- ## DDL Schema and DML Population:

Some Tables:

```
CREATE TABLE IF NOT EXISTS  `User` (
  User_ID CHAR(8) PRIMARY KEY
, User_FirstName VARCHAR(50) NOT NULL
, User_MidName VARCHAR(50)  NOT NULL
, User_FamilyName VARCHAR(50)
, User_Email VARCHAR(70) UNIQUE
, User_Field_ID INT
, FOREIGN KEY (User_Field_ID) REFERENCES Aiu_Field(Field_ID) ON UPDATE
CASCADE ON DELETE CASCADE
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

User_ID	User_FirstName	User_MidName	User_FamilyName	User_Email	User_Field_ID
21100167	Omyma	abdelgwad	hamed	omyma.abdelatef.2022@Aiu.edu.eg	8
21100168	Arwa	Mohamed	Fawzy	arwa.fawzy.2022@Aiu.edu.eg	8
21100169	Habiba	Hisham	Ahmed	habiba.ahmed.2022@aiu.edu.eg	8
21100170	Hussein	Tarek	Hussein	hussein.hussein.2022@aiu.edu.eg	8
21100171	Soaad	Tarek	Eltaib	soaad.mohamed.2022@aiu.edu.eg	8
21100172	Ali	Ayman	Ahmed	ali.ahmed.2022@Aiu.edu.eg	8
21100173	Fares	Elsayed	Ghoniem	fares.ghoniem.2022@Aiu.edu.eg	8
21100174	Mohamed	nsr	mohamed	mohamad.mohamad.2022@aiu.edu.eg	8
21100176	Nada	Tarek	Abdellah	nada.abdellah.2022@Aiu.edu.eg	8
21100177	Noha	Mohamed	Abd Elmeged	noha.abdelmeged.2022@aiu.edu.eg	8
21100178	Nouran	Muhammmed	Farid	nouran.farid.2022@Aiu.edu.eg	8
21100179	Hanaa	Taher	Abdelhameed	Hanaa.abdehameed.2022@Aiu.edu.eg	8
21100405	Ziyad	Mohamed	ali	ziyad.ali.2022@Aiu.edu.eg	8

```
CREATE TABLE IF NOT EXISTS  Student (
  Student_ID CHAR(8) PRIMARY KEY
, Field_ID INT NOT NULL
, Student_Level INT NOT NULL
, FOREIGN KEY (Student_ID) REFERENCES User(User_ID) ON UPDATE
CASCADE ON DELETE CASCADE
, FOREIGN KEY (Field_ID) REFERENCES Aiu_Field(Field_ID) ON UPDATE
CASCADE ON DELETE CASCADE
);
```

Result Grid | Filter Rows:

Student_ID	Field_ID	Student_Level
21100167	8	3
21100168	8	3
21100169	8	3
21100170	8	3
21100171	8	3
21100172	8	3
21100173	8	3
21100174	8	3
21100176	8	3
21100177	8	3
21100178	8	3
21100179	8	3

student 1

```
CREATE TABLE IF NOT EXISTS Coursera_Program (
  Program_ID INT PRIMARY KEY
, Program_Type VARCHAR(70)
, Program_Name VARCHAR(70)
, Admin_Link VARCHAR(255)
, Learner_Link VARCHAR(255)
, Field_ID INT NOT NULL
, FOREIGN KEY (Field_ID) REFERENCES Aiu_Field(Field_ID) ON UPDATE
CASCADE ON DELETE CASCADE
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Program_ID	Program_Type	Program_Name	Admin_Link	Learner_Link	Field_ID
1	Student Program	Computer Science & Engineering	https://www.coursera.org/course/3exbm64li2	https://www.coursera.org/course/abjynmngrh	1
2	Student Program	Pharmacy	https://www.coursera.org/course/ijrdbahbgu	https://www.coursera.org/course/1jru29pjhg	2
3	Student Program	Engineering	https://www.coursera.org/course/ejob2g53cg	https://www.coursera.org/course/2qffeemyw3	3
4	Student Program	Basic Sciences	https://www.coursera.org/course/evsizscyr9	https://www.coursera.org/course/t7l6xafby3	4
5	Student Program	Business	https://www.coursera.org/course/8dzz2df5yi	https://www.coursera.org/course/77fgk7oaxc	5
6	Student Program	Dentistry	https://www.coursera.org/course/mflzngrnfz	https://www.coursera.org/course/xggxctrcjo	6
7	Student Program	Art and Design	https://www.coursera.org/course/csowp855bs	https://www.coursera.org/course/8sxu38o0w7	7
8	Student Program	Public Health	https://www.coursera.org/course/1vymrtls7y	https://www.coursera.org/course/87k47ewu70	8
9	Student Program	International Legal Studies	https://www.coursera.org/course/ac54r21i2i	https://www.coursera.org/course/ad68f3ewb5	9
10	Instructor Program	Computer Science & Engineering for Instructors	https://www.coursera.org/course/26jr7p1kec	https://www.coursera.org/course/f02j2w0oaj	1
11	Instructor Program	Pharmacy for Instructors	https://www.coursera.org/course/hnx0wpqfz4	https://www.coursera.org/course/awgezx0btu	2

```
CREATE TABLE IF NOT EXISTS  Instructors_Choosen_Coursera_Courses (
S_Season VARCHAR(15)
, S_Year INT
, Aiu_Course_Code VARCHAR(10)
, Instructor_ID CHAR(8)
, Coursera_Course_ID INT
, Comments VARCHAR(300) DEFAULT NULL
, PRIMARY KEY(S_Season, S_Year, Aiu_Course_code, Instructor_ID,
Coursera_Course_ID )
, FOREIGN KEY (S_Season) REFERENCES Semester(S_Season) ON UPDATE
CASCADE
, FOREIGN KEY (S_Year) REFERENCES Semester(S_Year) ON UPDATE CASCADE
, FOREIGN KEY (Aiu_Course_Code) REFERENCES
Aiu_Course(Aiu_Course_Code) ON UPDATE CASCADE
, FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID) ON
UPDATE CASCADE
, FOREIGN KEY (Coursera_Course_ID) REFERENCES
Coursera_Courses(Course_ID) ON UPDATE CASCADE
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

S_Season	S_Year	Aiu_Course_Code	Instructor_ID	Coursera_Course_ID	Comments
Fall	2023	AIE111	INST1110	9	NULL
Fall	2023	AIE111	INST1110	55	NULL
Fall	2023	AIE231	INST1110	19	NULL
Fall	2023	CSE014	INST1121	4	NULL
Fall	2023	CSE111	INST1325	146	NULL
Fall	2023	CSE113	INST1118	8	NULL
Fall	2023	CSE221	INST1110	5	NULL
Fall	2023	CSE251	INST1122	184	NULL
Fall	2023	CSE261	INST1145	138	NULL
Fall	2023	CSE344	INST1188	179	NULL
Fall	2023	CSE352	INST1122	40	NULL

n_coursera_c...

```
CREATE TABLE IF NOT EXISTS  Aiu_Courses_Enrollment (
S_Season VARCHAR(15)
, S_Year INT
, Student_ID CHAR(8)
, Aiu_Cousre_Code VARCHAR(10)
, Grade VARCHAR(5)
, Marks INT
, PRIMARY KEY(S_Season, S_Year, Student_ID, Aiu_Cousre_Code)
, FOREIGN KEY (S_Season) REFERENCES Semester(S_Season) ON UPDATE
CASCADE ON DELETE CASCADE
, FOREIGN KEY (S_Year) REFERENCES Semester(S_Year) ON UPDATE CASCADE
ON DELETE CASCADE
, FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID)  ON UPDATE
CASCADE ON DELETE CASCADE
, FOREIGN KEY (Aiu_Cousre_Code) REFERENCES
Aiu_Course(Aiu_Course_Code) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Result Grid | Filter Rows: | Edit: | Export/I

S_Season	S_Year	Student_ID	Aiu_Cousre_Code	Grade	Marks
Fall	2022	22100198	AIE111	B-	77
Fall	2022	22100198	CSE111	B+	86
Fall	2022	22100198	CSE281	B+	88
Fall	2022	22100198	MAT212	B+	92
Fall	2022	22100274	AIE111	B+	88
Fall	2022	22100274	CSE014	B+	88
Fall	2022	22100274	CSE111	D+	65
Fall	2022	22100274	CSE281	D	61
Fall	2022	22100274	MAT212	C-	69
Fall	2023	22100105	CSE014	D+	65

s_enrollment 1 ×

```
CREATE TABLE IF NOT EXISTS  Instructor_Teaches (
 S_Season VARCHAR(15)
, S_Year INT
, Instructor_ID CHAR(8)
, Aiu_Course_Code VARCHAR(10)
, PRIMARY KEY (S_Season,S_Year,Instructor_ID,Aiu_Course_Code )
,FOREIGN KEY (S_Season) REFERENCES Semester(S_Season) ON UPDATE
CASCADE
,FOREIGN KEY (S_Year) REFERENCES Semester(S_Year) ON UPDATE CASCADE
,FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID) ON
UPDATE CASCADE ON DELETE CASCADE
,FOREIGN KEY (Aiu_Course_Code) REFERENCES Aiu_Course(Aiu_Course_Code)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

Result Grid | Filter Rows: | Edit:

S_Season	S_Year	Instructor_ID	Aiu_Course_Code
Fall	2023	INST1110	AIE111
Fall	2023	INST1110	AIE231
Fall	2023	INST1110	CSE221
Fall	2023	INST1118	CSE113
Fall	2023	INST1121	CSE014
Fall	2023	INST1122	CSE251
Fall	2023	INST1122	CSE352
Fall	2023	INST1122	CSE363
Fall	2023	INST1145	CSE261
Fall	2023	INST1188	CSE344

ctor_teaches 1 ×

```
CREATE TABLE IF NOT EXISTS  Aiu_Course (
  Aiu_Course_Code VARCHAR(30) PRIMARY KEY
, Course_Name VARCHAR(255)
, Field_ID INT
, Programs VARCHAR(255)
, Semester INT
, Fall_Spring VARCHAR(35)
, Topics VARCHAR(3000)
, Preq_Aiu_Course VARCHAR(30)
, Expected_Student_Number INT DEFAULT 0
, FOREIGN KEY (Field_ID) REFERENCES Aiu_Field(Field_ID) ON UPDATE
CASCADE ON DELETE CASCADE
, FOREIGN KEY (Preq_Aiu_Course) REFERENCES Aiu_Course(Aiu_Course_Code)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

Aiu_Course_Code	Course_Name	Field_ID	Programs	Semester	Fall_Spring	Topics	Preq_Aiu
ACC112	Principles of Managerial Accounting	5	All Programs	2	Spring	Management Accounting and the business orga…	NULL
ACC213	Intermediate Accounting 2	5	Accounting & Information Systems	4	Spring	Topics covered in this course include: Accountin…	NULL
ACC411	Advanced Financial Accounting	5	Accounting & Information Systems	7	Fall	The contents of this course include: Equity Met…	NULL
AIE111	Artificial Intelligence	1	Health promotion,CE,AIE,CS,AIS	4	Fall	Knowledge representation and organization, Se…	CSE015
AIE121	Machine Learning	1	CE,AIE,CS,AIS	4	Spring	Linear Regression and regularization. Instance-…	AIE111
AIE212	Knowledge-based Systems	1		6	Spring	Propositional and predicate logic, non-classical l…	NULL
AIE213	Optimization Techniques	1		6	Spring	Linear Algebra and Matrices. Probability Theory …	NULL
AIE231	Neural Networks	1	CE,AIE,CS,AIS	5	Fall	Simple perceptron for classification, BackProp a…	NULL
AIE241	Natural Language Processing	1		6	Spring	Overview of NLP. Statistical Machine Translatio…	NULL
AIE323	Data Mining	1	CE,AIE,CS,AIS	5	Fall	Knowledge discovery in databases. Data mining	NULL

```
CREATE TABLE IF NOT EXISTS Mapping_Modules (
 Coursera_Course_ID INT
, Module_Number INT
, Aiu_Course_Code VARCHAR(30)
, Choosen_Week INT
, Percentage_OfOverlap_from_Aiu_course CHAR(20)
, Percentage_OfOverlap_from_Coursera_course CHAR(20)
, Instructor_ID CHAR(8)
, PRIMARY KEY (Coursera_Course_ID, Module_Number, Aiu_Course_Code,
Choosen_Week, Instructor_ID)
, FOREIGN KEY (Coursera_Course_ID) REFERENCES Modules(Coursera_ID) ON
UPDATE CASCADE ON DELETE CASCADE
, FOREIGN KEY (Module_Number) REFERENCES Modules(Module_number) ON
UPDATE CASCADE ON DELETE CASCADE
, FOREIGN KEY (Aiu_Course_Code) REFERENCES Weeks(Aiu_Course_Code) ON
UPDATE CASCADE ON DELETE CASCADE
, FOREIGN KEY (Choosen_Week) REFERENCES Weeks(Week_number) ON
UPDATE CASCADE ON DELETE CASCADE
, FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID) ON
UPDATE CASCADE ON DELETE CASCADE
);
```

Coursera_Course_ID	Module_Number	Aiu_Course_Code	Choosen_Week	Percentage_OfOverlap_from_Aiu_course	Percentage_OfOverlap_from_Coursera_course	Instructor_ID
4	1	CSE014	2	9%	7%	INST1121
4	2	CSE014	4	20%	8%	INST1121
5	1	CSE221	2	18%	7%	INST1110
5	2	CSE221	3	7%	6%	INST1110
5	3	CSE221	4	16%	20%	INST1110
5	4	CSE221	5	6%	8%	INST1110
5	5	CSE221	6	12%	10%	INST1110
8	1	CSE113	2	15%	4%	INST1118
8	2	CSE113	3	10%	13%	INST1118
8	3	CSE113	4	18%	18%	INST1118

Alamein International University
Computer Science Engineering
CSE221 Database

Example of insertion to tables:

```sql
INSERT INTO `user` (User_ID, User_FirstName, User_MidName, User_FamilyName, User_Email)
VALUES ('20100221', 'Mohamed', 'Abdelmegeed', 'Hegazy', 'mohammed.hegazy@Aiu.edu.eg'),
('21200132', 'Ahmed ', 'Mamdouh Mohamed Abdelmoaty', 'Mohamed', 'ahmed.abdelmoaty.2022@Aiu.edu.eg'),
('21200133', 'Nadim ', 'Ali Mohamed Tawfik ', 'Shehata', 'nadim.shehata.2023@Aiu.edu.eg'),
('22100105', 'Mazen  ', 'Ashraf Rashed Mohamed  ', 'Abdelmalek', 'mazen.abdelmalek.2023@Aiu.edu.eg');


INSERT INTO `user_contact` (User_ID, Phone)
VALUES ('20100221', 0120345678),
 ('21200132', 0110357597),
 ('21200133', 01274537574),
 ('INST3122', 01035368463),
('INST3112',0126483567),
('INST3125',01267325446);

INSERT INTO `user_Account` (User_ID, User_Email, User_Password)
VALUES ('20100221', 'mohammed.hegazy@Aiu.edu.eg', 'password123'),
('21200132', 'ahmed.abdelmoaty.2022@Aiu.edu.eg', 'password123'),
('21200133', 'nadim.shehata.2023@Aiu.edu.eg', 'password123'),
('22100105', 'mazen.abdelmalek.2023@Aiu.edu.eg', 'password123'),
('22100133', 'mohamed.motawea.2023@Aiu.edu.eg', 'password123'),
('22100159', 'sara.saleh.2023@Aiu.edu.eg', 'password123');
```

```sql
INSERT INTO `Roles` (Role_Number, Role_Name)
VALUES (1, 'Student'),
(2, 'Academic'),
(3, 'Main coordinator of Field'),
(4, 'Administrator');

 INSERT INTO `User_Roles` ( User_ID, Role_Number )
 VALUES ('20100221', 1),
('21200132', 1),
('21200133', 1),
('22100105',1),
('22100133',1),
('22100159', 1);
```

```sql
INSERT INTO `Administrators` (Administrator_ID)
VALUES ('INST1111'),
('INST1121'),
('INST1122'),
('INST3122'),
('INST3112'),
('INST3125');
```

Alamein International University
Computer Science Engineering
CSE221 Database

Phase 5

- **Query answers:**

Q1. Who is The most student who got courses certificates?

Relational Algebra Q1:

π (Student_ID(σ Completed_Courses=max(Completed_Courses)(S))
S: Student_Programe_Enrollment

```
243     #1
244 •   SELECT Student_ID
245     FROM Student_Programe_Enrollment
246     WHERE Completed_Courses = (
247         SELECT MAX(Completed_Courses)
248         FROM Student_Programe_Enrollment
249     );
```

Result Grid | Filter Rows: | Edit:

Student_ID
22100578
22101352
22101470
NULL

Q2. Students who haven't finished their coursera courses.

Relational Algebra Q2:

πStudent_ID(σCompleted='No'(Coursera_Enrollment))

```
262     #2
263 •   SELECT Student_ID
264     FROM Coursera_Enrollment
265     WHERE Completed = 'No';
```

Result Grid | Filter Rows:

Student_ID
22100165
22100512
22100828
22100951
22101078
22101115
22101123

Coursera_Enrollment 9

Q3. Students who registered for a specific Aiu course in a specific semester.

Relational Algebra Q3:

π Student_ID(σ Aiu_Cousre_Code= 'specific_course_code '∧S_Season= 'specific_season '∧S_Year=specific_year(Aiu_Courses_Enrollment))

```
269     #3
270 •   SELECT Student_ID
271     FROM Aiu_Courses_Enrollment
272     WHERE Aiu_Cousre_Code = 'CSE014'
273     AND S_Season = 'Fall'
274     AND S_Year = 2023;
275
```

Result Grid | Filter Rows: | Ex

Student_ID
22100105
22100252
22100811

Q4. Can you Retrieve a list of instructors and their corresponding AIU courses along with the Coursera courses?

Relational Algebra Q4:

ρ(User_ID,Full_Name,Coursera_Course_ID,Course_Type,Course_Name(πUser_ID,user_firstname,user_midName,user_familyName,Coursera_Course_ID,Course_Type,Coure_Name(USER⋈ User_ID=Instructor_ID instructors_choosen_coursera_courses ⋈ Coursera_Course_ID=Course_ID(coursera_courses))

```sql
#4
SELECT User_ID,
    CONCAT(user.user_firstname, ' ', user.user_midName, ' ', user.user_familyName) AS "Full Name",
    Coursera_Course_ID,
    Course_Type ,
    Course_Name
FROM USER , instructors_choosen_coursera_courses , coursera_courses
WHERE User_ID = instructors_choosen_coursera_courses.Instructor_ID
AND Coursera_Course_ID = Course_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_ID	Full Name	Coursera_Course_ID	Course_Type	Course_Name
INST1110	Hesham Marei	9	Course	Practical Python for AI Coding 1
INST1110	Hesham Marei	55	Course	Introduction to AI
INST1110	Hesham Marei	19	Course	Neural Networks and Deep Learning
INST1121	Mohamed El Tonsy	4	Course	Java Programming: Solving Problems with Softw…
INST1325	Dalia Ossama	146	Course	Ordered Data Structures
INST1118	Amany Taha	8	Course	Linear Circuits 1: DC Analysis
INST1110	Hesham Marei	5	Course	Relational database systems
INST1122	Amr Rizk	184	Course	Introduction to Virtual Reality
INST1145	Amany Farouk	138	Course	Fundamentals of Network Communication
INST1188	Nermin Saad Fathalla	179	Course	Healthcare Data Quality and Governance
INST1122	Amr Rizk	49	Course	Software Engineering: Software Design and Pro…
INST1122	Amr Rizk	131	Course	introduction to cloud computing

Result 3

Q5. Retrieve students who enrolled in specific coursera course.

Relational Algebra Q5:

σCoursera_Course_ID=9(Coursera_Enrollment)

```sql
-- ================================
#5
SELECT Student_ID
FROM Coursera_Enrollment
WHERE Coursera_Course_ID = 9;
```

Result Grid | Filter Rows: | Export:

Student_ID
22100252
22100259
22100317
22100397
22100506
22100512

Coursera_Enrollment 25

Alamein International University
Computer Science Engineering
CSE221 Database

Q6. Students that need to be added and removed from coursera program.

Relational Algebra Q6:

π Student_ID,Member_State(σ Member_State="MEMBER" (Student_Programe_Enrollment))

```
306 •   SELECT Student_ID ,
307         Member_State
308         FROM student_programe_enrollment
309         WHERE Member_State = "MEMBER";
```

Result Grid | Filter Rows: | Edit:

Student_ID	Member_State
21200132	MEMBER
21200133	MEMBER
22100105	MEMBER
22100133	MEMBER
22100159	MEMBER
22100165	MEMBER
22100179	MEMBER
22100193	MEMBER
22100194	MEMBER
22100226	MEMBER

student_programe_enrollment 21

Q7. Display the enrollment status of students in a particular Coursera Courses, including their email, name and ID?

Relational Algebra Q7:

σCoursera_Course_ID=9(Coursera_Enrollment)

```
290     #5
291 •   SELECT * FROM coursera_enrollment
292     WHERE Coursera_Course_ID = 9;
293
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Student_ID	Coursera_Course_ID	Progress	Hours	Completed	Completion_Time	Course_Grade	Coursera_Certificate_URL	Enrollment_Date	Cl
22100252	9	100	2.07	Yes	2023-10-16 17:07:36	88	https://www.coursera.org/account/accomplish...	2023-10-08 16:06:36	20
22100259	9	100	1.68	Yes	2023-10-16 19:28:05	95	https://www.coursera.org/account/accomplish...	2023-10-08 00:59:16	20
22100317	9	100	0.21	Yes	2023-10-16 14:34:52	91	https://www.coursera.org/account/accomplish...	2023-10-16 13:56:01	20
22100397	9	100	1.93	Yes	2023-10-16 07:28:37	92	https://www.coursera.org/account/accomplish...	2023-10-07 17:08:35	20
22100506	9	89.58	1.62	Yes	2023-10-16 14:42:44	92	https://www.coursera.org/account/accomplish...	2023-10-05 05:54:08	20
22100512	9	100	1.49	Yes	2023-10-09 17:28:22	94	https://www.coursera.org/account/accomplish...	2023-10-07 16:36:06	20
22100540	9	100	0.35	Yes	2023-10-15 18:02:36	91	https://www.coursera.org/account/accomplish...	2023-10-07 16:08:11	20
22100578	9	100	6.49	Yes	2023-10-15 17:27:30	92	https://www.coursera.org/account/accomplish...	2023-10-07 16:11:25	20
22100591	9	97.92	0.44	Yes	2023-10-10 09:11:35	95	https://www.coursera.org/account/accomplish...	2023-10-10 07:53:00	20
22100638	9	97.92	1.79	Yes	2023-11-16 14:51:55	94	https://www.coursera.org/account/accomplish...	2023-10-14 21:05:40	20
22100786	9	100	0.3	Yes	2023-10-16 17:27:09	94	https://www.coursera.org/account/accomplish...	2023-10-16 07:31:44	20
22100828	9	0	0.08	No	NULL	0		2023-10-06 15:51:27	20
22100881	9	87.5	2.68	Yes	2023-10-15 03:42:53	91	https://www.coursera.org/account/accomplish...	2023-10-07 05:56:55	20
22100908	9	100	4.86	Yes	2023-10-04 12:51:57	91	https://www.coursera.org/account/accomplish...	2023-10-01 16:06:15	20

Alamein International University
Computer Science Engineering
CSE221 Database

Q8. Mapping Aiu Courses in a spacific semester and spacific field.

Relational Algebra Q8:

π aiu_course.aiu_course_code(σ S_Season="Fall"∧S_Year=2023∧aiu_course.Field_ID=1(map_aiu_courses_in_semesters⋈map_aiu_courses_in_semesters.aiu_course_code=aiu_course.aiu_course_code (aiu_course))

```
322    #8
323 •  SELECT aiu_course.aiu_course_code
324    FROM map_aiu_courses_in_semesters, aiu_course
325    WHERE map_aiu_courses_in_semesters.aiu_course_code = aiu_course.aiu_course_code
326    AND S_Season = "Fall" AND S_Year = 2023 AND aiu_course.Field_ID = 1;
327
328
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

aiu_course_code
AIE111
AIE231
CSE014
CSE111
CSE113
CSE221
CSE241
CSE261
CSE344
CSE363
MAT231

Q9. Show Mapping catalog about spacific AIU course and also calculate the overlap percentage from both sides: how much is covered from the AIU course and how much from the Coursera course.

Relational Algebra Q9:

σ Aiu_Course_Code="CSE221(coursera_project.mapping_modules)

```
329    #9
330 •  SELECT * FROM coursera_project.mapping_modules
331    WHERE Aiu_Course_Code = "CSE221";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Coursera_Course_ID	Module_Number	Aiu_Course_Code	Choosen_Week	Percentage_OfOverlap_from_Aiu_course	Percentage_OfOverlap_from_Coursera_course	Instructor_ID
5	1	CSE221	2	18%	7%	INST1110
5	2	CSE221	3	7%	6%	INST1110
5	3	CSE221	4	16%	20%	INST1110
5	4	CSE221	5	6%	8%	INST1110
5	5	CSE221	6	12%	10%	INST1110
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Q10. Collect a list of all instructors from each field every semester to add them as admins on their programs and also as admins and learners on the Instructors program .

Relational Algebra Q10:

$$\sigma_{S\_Season="Fall" \wedge S\_Year=2023}(instructor\_teaches)$$

```
333     -- ======================
334     #10
335 •   SELECT * FROM coursera_project.instructor_teaches
336     WHERE S_Season = "Fall" AND S_Year = 2023;
```

Result Grid | Filter Rows: | Edit: | Export/Im

S_Season	S_Year	Instructor_ID	Aiu_Course_Code
Fall	2023	INST1110	AIE111
Fall	2023	INST1110	AIE231
Fall	2023	INST1110	CSE221
Fall	2023	INST1118	CSE113
Fall	2023	INST1121	CSE014
Fall	2023	INST1122	CSE251
Fall	2023	INST1122	CSE352
Fall	2023	INST1122	CSE363
Fall	2023	INST1145	CSE261
Fall	2023	INST1188	CSE344
Fall	2023	INST1325	CSE111
Fall	2023	INST1325	MAT231
NULL	NULL	NULL	NULL

- **Different views:**

```
-- #1CourseraState
CREATE VIEW CourseraState AS
SELECT User_id , user_firstname, user_midname, user_familyname, user_email, completed, coursera_course_id, course_name
FROM user, coursera_enrollment ,coursera_courses
WHERE User_id = student_id AND coursera_course_id = course_id AND completed = "No";
SELECT * FROM CourseraState;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_id	user_firstname	user_midname	user_familyname	user_email	completed	coursera_course_id	course_name
22100165	Nour	Mohamed	Elbarawi	nour.elbarawi.2023@Aiu.edu.eg	No	8	Linear Circuits 1: DC Analysis
22100512	Youssef	Mohamed	Sabri	youssef.sabri.2023@Aiu.edu.eg	No	8	Linear Circuits 1: DC Analysis
22100828	Mahmoud	Hossam	Elmoaty	mahmoud.abdelsalam.2023@Aiu.edu.eg	No	9	Practical Python for AI Coding 1
22100951	Alhassan	Mohamed	Hassan	alhassan.hassan.2023@Aiu.edu.eg	No	4	Java Programming: Solving Problems with Softw
22101078	Omar	Essam	Alim	omar.okil.2023@Aiu.edu.eg	No	55	Introduction to AI
22101115	Mohamed	Ahmed	Ahmed	mohamed.ahmed.2023@Aiu.edu.eg	No	4	Java Programming: Solving Problems with Softw
22101123	Ahmed	Mohamed	Mohamed	ahmed.elmoslmany.2023@Aiu.edu.eg	No	9	Practical Python for AI Coding 1
22101140	Ahmed	Abdallah	Yosef	ahmed.yosef.2023@Aiu.edu.eg	No	55	Introduction to AI
22101153	Fares	Mohammed	Abdullatif	fares.abdullatif.2023@Aiu.edu.eg	No	9	Practical Python for AI Coding 1
22101153	Fares	Mohammed	Abdullatif	fares.abdullatif.2023@Aiu.edu.eg	No	147	Inferential Statistics
22101425	Abdelrhman	Mostafa	Abdelrady	abdelrhman.abdelrady.2023@Aiu.edu.eg	No	55	Introduction to AI

```
CREATE VIEW AiucCourseCode AS
SELECT ac.Aiu_course_code , course_name, s_season, s_year
from aiu_course ac , map_aiu_courses_in_semesters mac
where ac.Aiu_course_code = mac.Aiu_Course_Code AND S_Season = "Fall" AND s_year = 2022;
SELECT * FROM AiucCourseCode;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Aiu_course_code	course_name	s_season	s_year
AIE111	Artificial Intelligence	Fall	2022
AIE231	Neural Networks	Fall	2022
AIE323	Data Mining	Fall	2022
ARC222	Architectural Studio 2	Fall	2022
BIO131	Biology 1	Fall	2022
BIS351	Database Design & Management	Fall	2022
CHE142	Engineering Chemistry	Fall	2022
CSE014	Structured Programming	Fall	2022
CSE111	Data Structures	Fall	2022
CSE112	Design & Analysis of Algorithms	Fall	2022
CSE221	Database Systems	Fall	2022
CSE281	Image Processing	Fall	2022
ELE111	Electrical Circuits (30 Std.) 8%	Fall	2022
EVD211	Interior Design Studio I	Fall	2022
EVD213	Representation& Rendering Te...	Fall	2022

AiucCourseCode 35 ×

Alamein International University
Computer Science Engineering
CSE221 Database

```sql
CREATE VIEW GuidedProject AS
SELECT course_id, course_type, course_name
from coursera_courses
where course_type = "Guided Project";
SELECT * FROM GuidedProject;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conter

course_id	course_type	course_name
11	Guided Project	Get started with Photoshop
24	Guided Project	How to Design a space - saving Table using sket…
38	Guided Project	Sars cov 2 protein modeling and drug docking
39	Guided Project	Graphic Design: Make Interior's Project Mood Bo…
58	Guided Project	Sketchup : How to model a 3D mockup to show …
63	Guided Project	Creating Perspective Visuals Using Adobe illustr…
71	Guided Project	Tobacco Control: Agile Policy, Research and Pra…
148	Guided Project	How to Design a Space-Saving Table Using Sket…

GuidedProject 39

```sql
-- #1
CREATE VIEW GradeInfo AS
SELECT user_id, User_FirstName, User_FamilyName, User_Email, grade, marks
FROM user INNER JOIN aiu_courses_enrollment
ON user_id = Student_ID AND grade like "A%" AND Aiu_Cousre_Code = "CSE111"
ORDER BY marks DESC;
SELECT * FROM GradeInfo;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	User_FirstName	User_FamilyName	User_Email	grade	marks
22100252	Walid	Abdel Meguid	walid.abdelmeguid.2023@Aiu.edu.eg	A	93
22100811	Mohamed	Mheina	mohamed.mheina.2023@Aiu.edu.eg	A	93

```sql
CREATE VIEW userData AS
SELECT
    ur.role_number,
    r.Role_Name,
    u1.User_ID,
    CONCAT(u1.user_firstname, ' ', u1.user_midName, ' ', u1.user_familyName) AS "Full Name",
    u1.User_Field_ID,
    af.Field_Name,
    u1.User_Email,
    ua.user_password,
    ua.security_Question_id,
    sq.Question_Name,
    ua.answer
FROM
    user_roles ur
    JOIN user u1 ON ur.user_id = u1.user_id
    JOIN roles r ON ur.Role_Number = r.Role_Number
    JOIN user_account ua ON u1.user_id = ua.user_id
    JOIN security_question sq ON ua.security_Question_id = sq.Question_ID
    JOIN aiu_field af ON u1.User_Field_ID = af.Field_ID ;
SELECT * FROM userData ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

role_number	Role_Name	User_ID	Full Name	User_Field_ID	Field_Name	User_Email	user_password	security_Question_id	Question_Name
2	Instructor	INST9999	Ahmed Ramadan Mohamed	9	International Legal Studies	amostafa@aiu.edu.eg	lKgg1NczOk@aiu.eg	1	What is your nick na
3	Main Coordinator of Field	INST9999	Ahmed Ramadan Mohamed	9	International Legal Studies	amostafa@aiu.edu.eg	lKgg1NczOk@aiu.eg	1	What is your nick na
2	Instructor	INST8021	Aida Mohey Mohamed	8	Public Health	aramadan@aiu.edu.eg	eGVPU2ZuxY@aiu.eg	1	What is your nick na
2	Instructor	INST8080	Nesrine Nabil	8	Public Health	Nyassa@aiu.edu.eg	L4GIzsSlAe@aiu.eg	1	What is your nick na

- **Procedure:**

```sql
DELIMITER //
CREATE PROCEDURE GetCourseraState(IN state VARCHAR(5) )
BEGIN
    SELECT User_id, user_firstname, user_midname, user_familyname, user_email, completed, coursera_course_id, course_name
    FROM user, coursera_enrollment, coursera_courses
    WHERE User_id = student_id AND coursera_course_id = course_id AND completed = state;
END //
DELIMITER ;
CALL GetCourseraState("No");
-- ================================================
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_id	user_firstname	user_midname	user_familyname	user_email	completed	coursera_course_id	course_name
22100165	Nour	Mohamed	Elbarawi	nour.elbarawi.2023@Aiu.edu.eg	No	8	Linear Circuits 1: DC Analysis
22100512	Youssef	Mohamed	Sabri	youssef.sabri.2023@Aiu.edu.eg	No	8	Linear Circuits 1: DC Analysis
22100828	Mahmoud	Hossam	Elmoaty	mahmoud.abdelsalam.2023@Aiu.edu.eg	No	9	Practical Python for AI Coding 1
22100951	Alhassan	Mohamed	Hassan	alhassan.hassan.2023@Aiu.edu.eg	No	4	Java Programming: Solving Problems with So
22101078	Omar	Essam	Alim	omar.okil.2023@Aiu.edu.eg	No	55	Introduction to AI
22101115	Mohamed	Ahmed	Ahmed	mohamed.ahmed.2023@Aiu.edu.eg	No	4	Java Programming: Solving Problems with So
22101123	Ahmed	Mohamed	Mohamed	ahmed.elmoslmany.2023@Aiu.edu.eg	No	9	Practical Python for AI Coding 1

Result 41

```sql
DELIMITER //
CREATE PROCEDURE GetStudentProgrameState(IN state VARCHAR(50))
BEGIN
    SELECT user_id ,
    CONCAT(user_firstname, ' ', user_midName, ' ', user_familyName) AS "Full Name",
    User_Email,
    User_Field_ID,
    Member_State
    FROM user,student_programe_enrollment
    WHERE user.user_id = student_programe_enrollment.Student_ID
    AND Member_State = state;
END //
DELIMITER ;
CALL GetStudentProgrameState("DELETED_MEMBER");
CALL GetStudentProgrameState("MEMBER");
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	Full Name	User_Email	User_Field_ID	Member_State
21200132	Ahmed Mamdouh Abdelmoaty	ahmed.abdelmoaty.2022@Aiu.edu.eg	1	MEMBER
21200133	Nadim Ali Shehata	nadim.shehata.2023@Aiu.edu.eg	1	MEMBER
22100105	Mazen Ashraf Abdelmalek	mazen.abdelmalek.2023@Aiu.edu.eg	1	MEMBER
22100133	Mohamed Hatem Motawea	mohamed.motawea.2023@Aiu.edu.eg	1	MEMBER
22100159	Sara Saeed Saleh	sara.saleh.2023@Aiu.edu.eg	1	MEMBER
22100165	Nour Mohamed Elbarawi	nour.elbarawi.2023@Aiu.edu.eg	1	MEMBER

Result 44

Alamein International University
Computer Science Engineering
CSE221 Database

```sql
DELIMITER //
CREATE PROCEDURE GetAiucCourseCode(IN season VARCHAR(15), IN yaer INT, IN field INT )
BEGIN
    SELECT ac.Aiu_course_code, course_name, s_season, s_year
    FROM aiu_course ac, map_aiu_courses_in_semesters mac
    WHERE ac.Aiu_course_code = mac.Aiu_Course_Code AND S_Season = season AND s_year = yaer AND ac.Field_ID = field;
END //
DELIMITER ;
CALL GetAiucCourseCode("Fall", 2022, 1);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Aiu_course_code	course_name	s_season	s_year
AIE111	Artificial Intelligence	Fall	2022
AIE231	Neural Networks	Fall	2022
AIE323	Data Mining	Fall	2022
CSE014	Structured Programming	Fall	2022
CSE111	Data Structures	Fall	2022
CSE112	Design & Analysis of Algorithms	Fall	2022
CSE221	Database Systems	Fall	2022
CSE281	Image Processing	Fall	2022
MAT212	Linear Algebra	Fall	2022

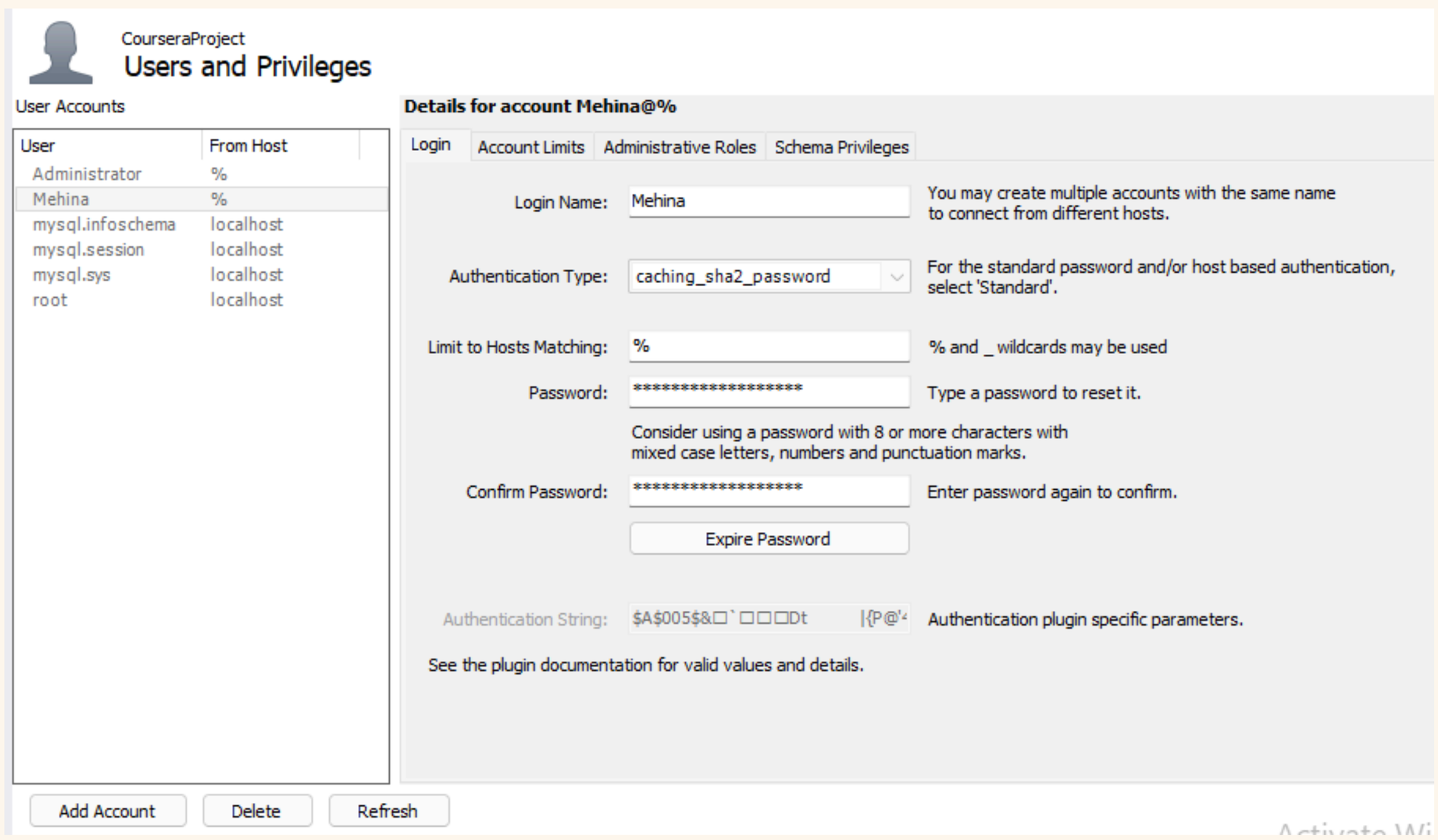
Result 42

```sql
DELIMITER $$
CREATE PROCEDURE GetGradeInfo(IN Grade varchar(5), IN course_code varchar(15) )
BEGIN
    SELECT user_id, User_FirstName, User_FamilyName, User_Email, grade, marks
    FROM user
    INNER JOIN aiu_courses_enrollment ON user_id = Student_ID
    WHERE grade LIKE Grade AND Aiu_Cousre_Code = course_code
    ORDER BY marks DESC;
END$$
DELIMITER ;
CALL GetGradeInfo("A" , "CSE111");
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	User_FirstName	User_FamilyName	User_Email	grade	marks
22100252	Walid	Abdel Meguid	walid.abdelmeguid.2023@Aiu.edu.eg	A	93
22100811	Mohamed	Mheina	mohamed.mheina.2023@Aiu.edu.eg	A	93
22100198	Ismail	Mousa	ismail.mousa.2023@Aiu.edu.eg	A	86
22100105	Mazen	Abdelmalek	mazen.abdelmalek.2023@Aiu.edu.eg	A	84
22100274	Hassan	Fayed	hassan.fayed.2023@Aiu.edu.eg	A	65

- **Users Access:**



Alamein International University
Computer Science Engineering
CSE221 Database

- **Team Members:**

1- Mohamed sobhy mehina	**ID: 22100811**
2- Ziad hossam Eldin mostafa	**ID: 22100793**
3- Mohamed ahmed fawzy	**ID: 22100881**
4- Mahmoud hossam abd elmoaty	**ID: 22100828**
5- Malak Yasser aly hassan eid	**ID: 22100400**

Thanks