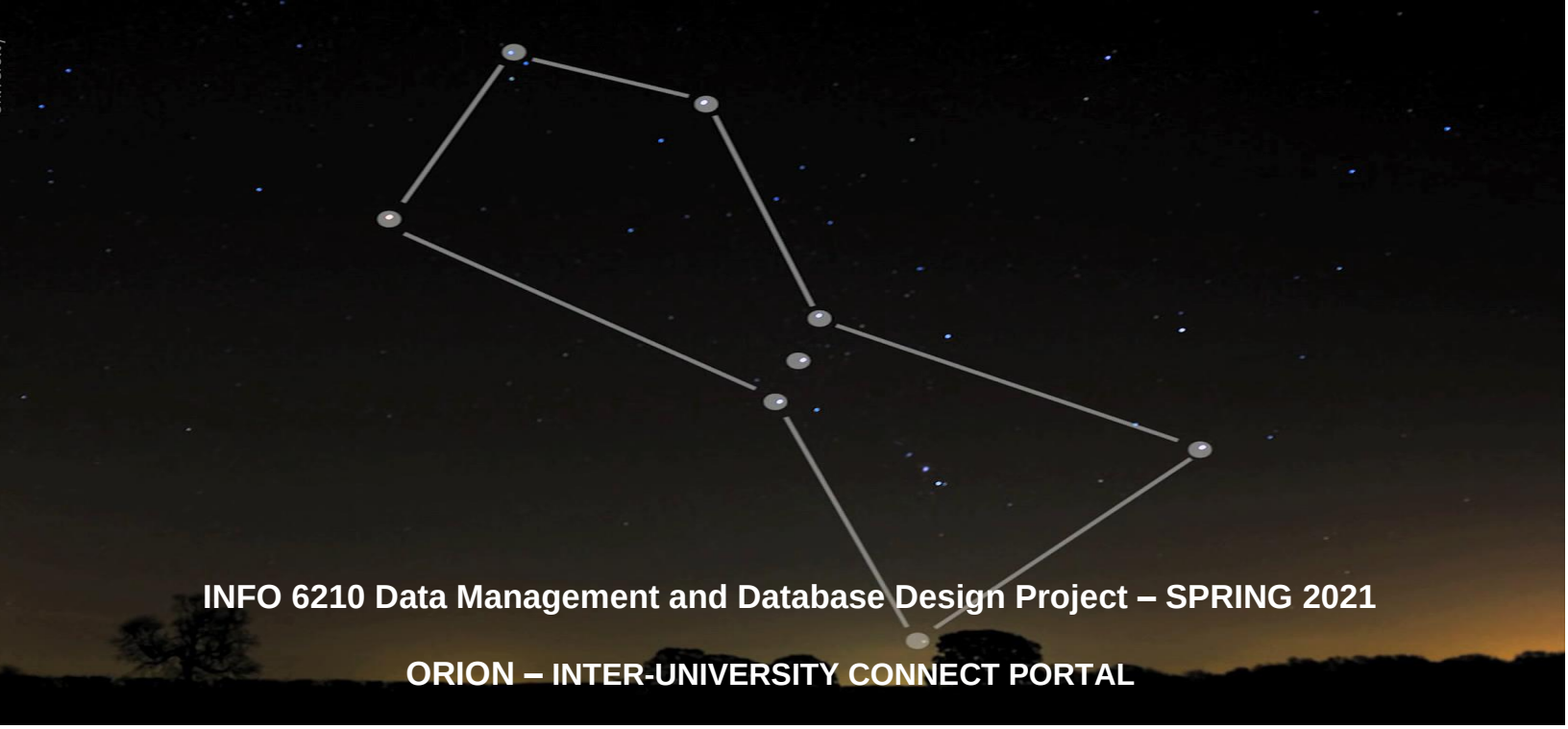


INFO 6210 Data Management and Database Design Project – SPRING 2021

ORION – INTER-UNIVERSITY CONNECT PORTAL

TEAM DETAILS:

Team Name: **ORION**

Team members:

- Priyanka Girish Chaudhari (NUID: 001007299)
- Ankita Senapati (NUID: 001003695)
- Aaradhy Sharma (NUID: 001040799)

SUMMARY:

Database system to connect students, professors, and alumni to expand their network, connect with people with similar interests, to share knowledge, thoughts, opportunities, and information. Users can have access to all the information on one single platform. Students will have major benefit as they can connect with their as well as other Universities' professors, alumni, and fellow peers to discuss their areas of interest and opportunities.

OBJECTIVES:

To design a database for an Inter-University Connect Portal which will provide the following features to its members –

1. Users including students, alumni, and professors will be able to create and manage their profiles.

2. Portal members will be able to create and host various types of events.

3. Portal members can register and get event details.

4. Portal members can connect with other members as well based on their interests.

5. Maintain details of Open-Source Projects done by various Universities.

6. A student from one University can connect with students, professors, alumni of any other registered University.

PROBLEM STATEMENT:

1. It is difficult to keep track of various social media platforms for student and alumni connect from other Universities.

2. It is uneconomical to follow various universities individual websites for various events hosted by them.

3. Students are always keen to read about various projects done by Universities and therefore follow them on blogs and other platforms.

4. International students, especially, face these challenges connecting with domestic or other international students as well as professors.

PROPOSED SOLUTION:

To overcome the above problems and challenges faced by students, a database model can be used –

1. Using the database model, a portal can be designed which enables maintain profiles for students, alumni, and professors.

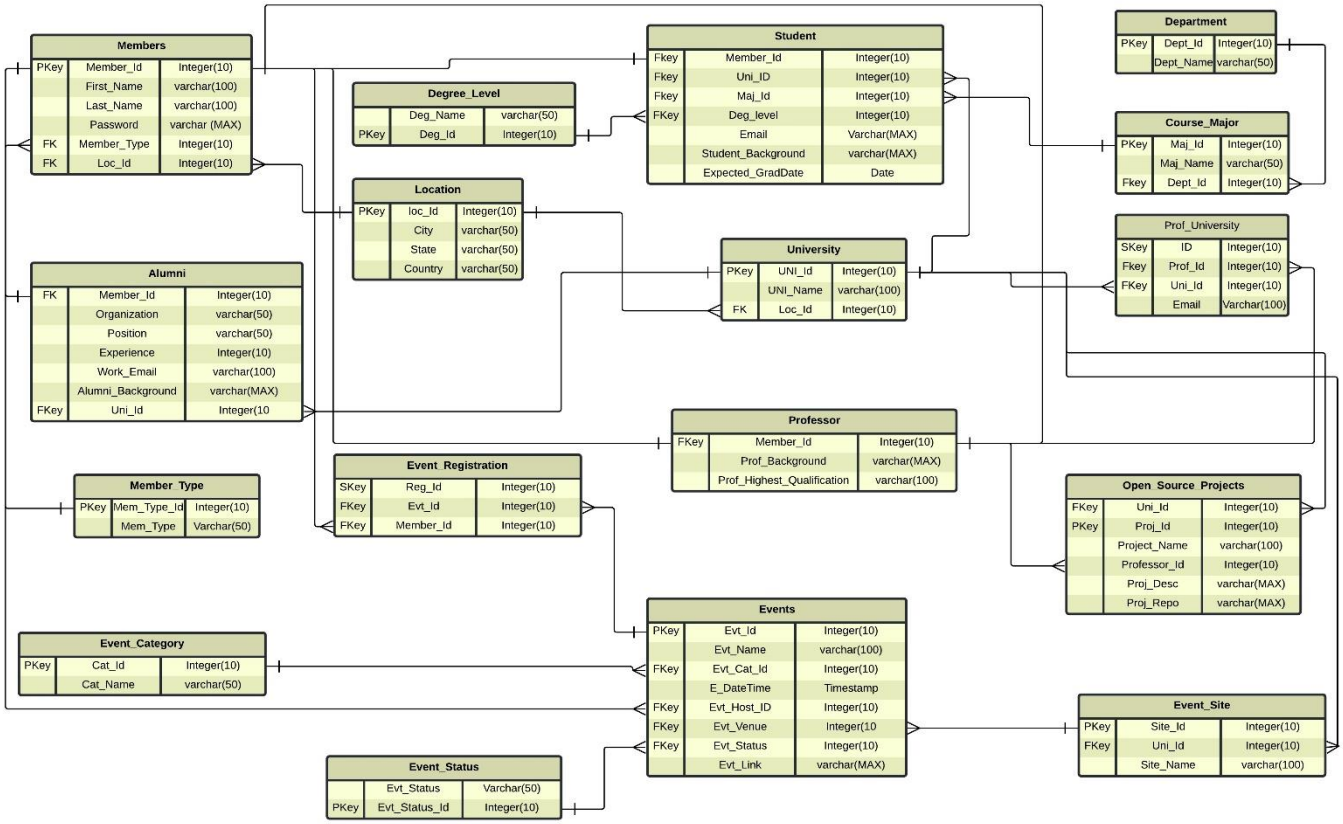
2. The Event table will have details about different types of events like Career Fairs, Speaker Sessions, Cultural or technical connections.

3. The Student table will have details like AreaOfInterest, Course Major information, which will help students to connect with other students having similar interests.

4. The Professor table will have information about Professors from various Universities, therefore giving students a vast opportunity to associate with them.

5. Open Projects table will have all the data related to the open-source projects led by a University or Professor. Interested students will also have the opportunity to reach out to them if it interests them.

DATA MODEL ENTITY-RELATIONSHIP DIAGRAM:



ENTITY DETAILS:

<UNIVERSITY>

Attribute	Datatype and width	Comments
UNI_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
UNI_NAME	VARCHAR (100)	NOT NULL
LOC_ID	INTEGER (10)	A Foreign Key which REFERENCES LOC_ID(PK) from LOCATION table

<MEMBERS>

Attribute	Datatype and width	Comments
MEMBER_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
FIRST_NAME	VARCHAR (100)	NOT NULL
LAST_NAME	VARCHAR (100)	NOT NULL
PASSWORD	VARCHAR (max)	NOT NULL
MEMBER_TYPE	INTEGER (10)	A Foreign Key which REFERENCES MEM_TYPE (PK) from MEMBER_TYPE table
LOC_ID	INTEGER (10)	A Foreign Key which REFERENCES LOC_ID (PK) from LOCATION table

<MEMBER_TYPE>

Attribute	Datatype and width	Comments
MEM_TYPE_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
MEM_TYPE	VARCHAR (50)	NOT NULL

<LOCATION>

Attribute	Datatype and width	Comments
LOC_ID	INTEGER (10)	PRIMARY KEY
CITY	VARCHAR (50)	NOT NULL
STATE	VARCHAR (50)	NOT NULL
COUNTRY	VARCHAR (50)	NOT NULL

<DEPARTMENT>

Attribute	Datatype and width	Comments
DEPT_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
DEPT_NAME	VARCHAR (50)	NOT NULL

<COURSE_MAJOR>

Attribute	Datatype and width	Comments
MAJ_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
MAJ_NAME	VARCHAR (50)	NOT NULL
DEPT_ID	INTEGER (10)	A Foreign Key which REFERENCES DEPT_ID (PK) from DEPARTMENT table

<DEGREE_LEVEL>

Attribute	Datatype and width	Comments
DEG_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
DEG_NAME	VARCHAR (50)	NOT NULL

<STUDENT>

Attribute	Datatype and width	Comments
MEMBER_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table, UNIQUE KEY
UNI_ID	INTEGER (10)	A Foreign Key which REFERENCES UNI_ID (PK) from UNIVERSITY table
MAJ_ID	INTEGER (10)	A Foreign Key which REFERENCES MAJ_ID(PK) from COURSE_MAJOR table, NOT NULL
DEGREE_LEVEL	INTEGER (10)	A Foreign Key which REFERENCES DEG_ID(PK) from DEGREE_LEVEL, NOT NULL
EMAIL	VARCHAR(max)	CHECK Constraint – Contains '%.edu'
STUDENT_BACKGROUND	VARCHAR (50)	NOT NULL
EXPECTED_GRADDATE	DATE	NOT NULL

<ALUMNI>

Attribute	Datatype and width	Comments
MEMBER_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table
ORGANIZATION	VARCHAR (50)	DEFAULT VALUE- NOT SET
POSITION	VARCHAR (50)	DEFAULT VALUE- NOT SET
EXPERIENCE	INTEGER (10)	DEFAULT VALUE- NOT SET
WORK_EMAIL	VARCHAR (100)	DEFAULT VALUE- NOT SET, UNIQUE
ALUMNI_BACKGROUND	VARCHAR (max)	DEFAULT VALUE- NOT SET
UNI_ID	INTEGER (10)	A Foreign Key which REFERENCES UNI_ID (PK) from UNIVERSITY table

<PROFESSOR>

Attribute	Datatype and width	Comments
MEMBER_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table, UNIQUE KEY
PROF_BACKGROUND	VARCHAR (max)	NOT NULL
PROF_HIGHEST_QUAL	VARCHAR (100)	NOT NULL

<PROF_UNIVERSITY>

Attribute	Datatype and width	Comments
ID	INTEGER (10)	SURROGATE KEY
PROF_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table, UNIQUE KEY
UNI_ID	INTEGER (10)	A Foreign Key which REFERENCES UNI_ID (PK) from UNIVERSITY table
EMAIL	VARCHAR (100)	CHECK Constraint – Contains '%.edu'

<EVENT>

Attribute	Datatype and width	Comments
EVT_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
EVT_NAME	VARCHAR (100)	NOT NULL
EVT_CAT_ID	INTEGER (10)	A Foreign Key which REFERENCES EVT_CATEGORY_ID (PK) from EVENT_CATEGORY table
EVT_DATETIME	TIMESTAMP	NOT NULL
EVT_HOST_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table

EVT_VENUE	INTEGER (10)	A Foreign Key which REFERENCES SITE_ID (PK) from EVENT_SITE table.
EVT_STATUS	INTEGER (10)	A Foreign Key which REFERENCES EVT_STATUS (PK) from EVENT_STATUS table
EVT_LINK	VARCHAR(MAX)	NOT NULL

<EVENT_CATEGORY>

Attribute	Datatype and width	Comments
CAT_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
CAT_NAME	VARCHAR (50)	NOT NULL

<EVENT_STATUS>

Attribute	Datatype and width	Comments
EVT_STATUS_ID	INTEGER (10)	PRIMARY KEY, AUTOGENERATED
EVT_STATUS	VARCHAR (50)	Not Null

<EVENT_REGISTRATION>

Attribute	Datatype and width	Comments
REG_ID	INTEGER (10)	SURROGATE KEY, AUTO GENERATE. Since there are TWO COMPOSITE KEYS (MEMBER_ID, EVT_ID)
EVT_ID	INTEGER (10)	A Foreign Key which REFERENCES EVT_ID (PK) from EVENT table
MEMBER_ID	INTEGER (10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table

<EVENT_SITE>

Attribute	Datatype and width	Comments
SITE_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
UNI_ID	INTEGER (10)	A Foreign Key which REFERENCES UNI_ID (PK) from UNIVERSITY table
SITE_NAME	VARCHAR (100)	NOT NULL

<OPEN_SOURCE_PROJECTS>

Attribute	Datatype and width	Comments
PROJ_ID	INTEGER (10)	PRIMARY KEY, AUTO GENERATE
PROJ_NAME	VARCHAR (100)	NOT NULL
PROJ_DESC	VARCHAR (max)	NOT NULL
PROJ_REPO	VARCHAR (max)	DEFAULT VALUE- NOT SET
PROF_ID	INTEGER(10)	A Foreign Key which REFERENCES MEMBER_ID (PK) from MEMBER table,

Business Rules

1. MEMBERs can create one or more EVENTs
2. An EVENT can be created by only one MEMBER
3. Every member will have a MEMBER_ID which will be auto-generated and cannot be NULL
4. A MEMBER can edit events only created by that member.
5. Only the EVENT HOST can view the registrations for that event.

6. The EVENT_SITE entity contains various sites in the University where events can be hosted. A University can have multiple sites and their respective capacity.

7. Evt_Venue in the Events entity will be NULL for online events and will have a value in Evt_Link.

8. A MEMBER can exist only if the UNIVERSITY exists.

9. One professor can be a part of multiple universities and thus can have more than one email address.

10. All the email addresses in Student and Prod_University entities will be restricted to the .edu domain.

11. STATUS of the MEMBER can only be Professor, Alumni, or Student

12. DEG_LEVEL can be undergraduate, graduate, or PhD

13. EXPECTED_GRADDATE should not be null (can be editable). It would execute a trigger once the date is reached and the student will be transferred to the ALUMNI table with some values set as null.

14. Evt_Status in Events entity can have only three values- open for registration, closed for registration, and archived.

15. When the CAPACITY of the event is reached, a trigger will be executed to change the status to “closed for registration” until then it will be shown as “open for registration”

16. Once the event date is passed, a trigger will be executed, and the status will change to “Archived”

17. LOCATION entity will have multiple countries. Each Country can have multiple States. Each State can have multiple Cities.

SECURITY CONSTRAINTS:
(User level Access/Permissions)

1. ADMIN
    - Has full access to all database entities

2. PROFESSOR
    - Has read/write/update access on Member, Professor, Prof_University, and Open_Source_Projects entity.

- Has read access on all other entities.

3. UNIVERSITY
    - Has read/write/update access on University, Event_Site entities
    - Has read access on all other entities.
4. USER
    - Has read access on all the entities
    - Has write/update access on Member, Student, Alumni entity.