Group No.: Z **Project No.: Z**

Project Name: IIT Bhilai Course Management System

Phase-III

Logical Database Design

I. Changes Made:

1. In the requirement specification:
   None.

2. In the ER Diagram:
   a. A student cannot be enrolled in two different semesters at the same time, so converted **Sem** from a multi-valued attribute to a single valued attribute.
   b. Changed **CourseInfo** into a weak entity as it already contained the primary key of **Course**.
   c. Removed **Department_ID** attribute as our college does not do any such allocation.
   d. Made **Deparment_Name** as the primary key of **Department** entity.

Then what is "CSE", "EE", "ME", etc.?

II. Converting ER Diagram into Relationship Schema:

1. **Step-1:**
   This step involves creating a relation for all the entities and assigning a primary keyto it. They are as follows:

   Student
   RollNo, Fname, Lname, Type, EmailID, Qualification, DOB, City, State, Street, PinCode, MobNo

   Department
   DName, DHead, Approver

   Instructor
   InstId, Fname, Lname, EmailID, Qualification, Type

Course
CourseID, Course_Name, Term, StartingSeg, EndingSeg

2. **Step 2:**
   This step involves defining all binary relationships between two entities S and T and dividing them into 1:1, 1:n or m:n relationship.

   EnrolledIn (m:n relationship)
   S → Student, T → Course
   RollNo, CourseID, Grade, Sem, Year

   StudentOf (1:n relationship)
   S → Student, T → Department
   RollNo, Fname, Lname, Type, EmailID, Qualification, DOB, City, State, Street, PinCode, MobNo, DName

   FacultyIn (1:n relationship)
   S → Instructor, T → Department
   InstId, Fname, Lname, EmailID, Qualification, Type, DName

3. **Step 3:**
   In this step we create a relation R with all simple attributes of a weak entity W and the primary key of owner entity E.
   In our case we have the weak entity **CourseInfo** and it's owner entity is **Course**. So the new relation R is:

   CourseDetails (1:1 relationship)
   W → CourseInfo, E → Course
   CourseID, Prerequisite, Type, Duration, Credit, Content, Archives, Is-Active

4. **Step 4:**

In this step we create a new relation for each multi-valued attribute which contains the attribute and the primary key of the original relation it was a member of.

- Multi-valued attributes in **Student**: Type, MobNo

  Student_Type
  RollNo, Type

  Student_MobNo
  RollNo, MobNo

- Multi-valued attributes in **StudentOf**: Type, MobNo

  StudentOf_Type
  RollNo, Type

  StudentOf_MobNo
  RollNo, MobNo

- Multi-valued attributes in **Department**: Approver

  D_Approver
  DName, Approver

- Multi-valued attributes in **Instructor**: Type

  Inst_Type
  Inst_ID, Type

- Multi-valued attributes in **FacultyIn**: Type

FacultyIn_Type
Inst_ID, Type

- Multi-valued attributes in **CourseDetails**: Type, Prerequisite, Archives

  CourseDetails_Type
  CourseID, Type

  CourseDetails_Prerequisite
  CourseID, Prerequisite

  CourseDetails_Archives
  CourseID, Archives

5. **Step 5:**
   This step involves creating a new relation for n-ary relationship containing the primary keys of all the entities it connects as its attributes.

   OfferedBy
   DName, InstID, CourseID

III. Normalizing the Relationship Schema:

1. Converting to 1NF:
   In this step we create new relations for each non-atomic and nested attributes and remove these attributes from original relation.

   - Non-atomic attributes in **Student**: Type, MobNo
     Student_Type
     RollNo, Type

Student_MobNo
RollNo, MobNo

Student
RollNo, Fname, Lname, EmailID, Qualification, DOB, City, State, Street, PinCode

- Non-atomic attributes in **StudentOf**: Type, MobNo

StudentOf_Type
RollNo, Type

StudentOf_MobNo
RollNo, MobNo

StudentOf
RollNo, Fname, Lname, EmailID, Qualification, DOB, City, State, Street, PinCode, DName

- Non-atomic attributes in **Department**: Approver

D_Approver
DName, Approver

Department
DName, DHead

- Non-atomic attributes in **Instructor**: Type

Inst_Type
Inst_ID, Type

Instructor

InstID, Fname, Lname, EmailID, Qualification

- Non-atomic attributes in **FacultyIn**: Type

  FacultyIn_Type
  Inst_ID, Type

  FacultyIn
  InstID, Fname, Lname, EmailID, Qualification, DName

- Non-atomic attributes in **CourseDetails**: Type, Prerequisite, Archives

  CourseDetails_Type
  CourseID, Type

  CourseDetails_Prerequisite
  CourseID, Prerequisite

  CourseDetails_Archives
  CourseID, Archives

  CourseDetails
  CourseID, Duration, Credit, Content, IsActive

2. Converting to 2NF:
   In this step we decompose and set up a new relation of partial key with its dependent attributes keeping a relation with the original primary key and any attributes that are fully functionally dependent on it.

   - Attributes in **EnrolledIn** which are not fully functionally dependent on the superkey: Sem

CourseGradeDetails
RollNo, CourseID, Grade, Year

Semester
CourseID, Sem

Steps 3 and 4 are not needed because the schema already satisfies 3NF and BCNF.

IV. Final Relational Schema:

Student

RollNo	Fname	Lname	EmailID	Qualification	DOB	City	State	Street	PinCode

DepartmentID FK

Add to ER

Student_Type

RollNo	Type

FK

Student_MobNo

RollNo	MobNo

FK

Department

DName	DHead

DepartmentID

Removed it from primary key

D_Approver

DName	Approver

DepartmentID FK

Instructor

InstID	Fname	Lname	EmailID	Qualification

DepartmentID FK

Instructor_Type

InstID	Type

FK

Course

CourseID	Course_Name	Term	StartingSeg	EndingSeg

CourseDetails

CourseID	Duration	Credit	Content	IsActive

FK

CourseDetails_Type

CourseID	Type

FK

CourseDetails_Prerequisite

CourseID	Prerequisite

FK

CourseDetails_Archives

CourseID	Archives

FK

CourseGradeDetails

RollNo	CourseID	Grade	Year

Semester

CourseID	Sem

FK

StudentOf

Have to remove this b/c N:1 relation

RollNo	Fname	Lname	EmailID	Qualification	DOB	City	State	Street	PinCode

FacultyIn

Have to remove this b/c N:1 relation

InstID	Fname	Lname	EmailID	Qualification	DName

OfferedBy

DepartmentID

DName	InstID	CourseID

FK FK FK

Remark:

- Table of Student_Type and StudentOf_Type were the same so only one is taken.
- Table of Student_MobNo and StudentOf_MobNo were the same so only one is taken.
- Table of Instructor_Type and FacultyIn_Type were the same so only one is taken.

V. Challenges Faced:

1. Many of the tables ended up being nearly the same, for example StudentOf has all the attributes of Student with the addition of only one. So it made us to look into the books and verify our concepts, time and again, but in the end, this made our concepts clearer than ever.

2. One of the major challenges was to read and understand the ER diagram, as it was one the first time for us to do this.

3. Last but not the least, changing the format and writing into the one as asked was in itself a challenge.