Relational Model Explained

Table: Unity Users

Functional dependencies:

- unityID fname
- unityID lname
- unityID password
- unityID email
- unityID gender
- unityID dateOfBirth
- unityID address
- unityID phone
- unityID userType

Table: Students

Constraints:

-Uniqueness constraints: The studentid should be unique for each entry. - Foreign key constraints: The unityid in table UnityUsers , the deptID in the Departments table and creditReqID in the CreditRequirements are the foreign keys. - Data entry constraints: The residency level for all students must be among 1,2,3 where 1=in state, 2= out-of-state and 3=international and the participation level among 1, 2 where 1=undergraduate and 2=graduate.

Functional dependencies:

- unityID deptID
- unityID overallGPA
- unityID residencyLevel
- unityID participationLevel
- unityID yearEnrolled

Table: Employee

Constraints:

- Uniqueness constraints: The ssn for all employees should be unique.
- Foreign key constraints: The unityID from the UnityUsers table is the foreign key.

Functional dependencies:

- unityID SSN

Table: Faculty

Constraints:

- Uniqueness constraints: The facultyID for each faculty should be unique.
- Foreign key constraints: The unityid from the Employee table and deptID from the Department table are the foreign keys.

Functional dependencies:

- unityID deptID
- unityID Field
- SSN deptID
- SSN Field

Table: Admin

Constraints:

- Uniqueness constraints: The adminID for all the entries should be unique.
- Foreign key constraints: The unityID from the Employee table is the foreign key.

Table: Department

- deptID deptName

Table: Course

Constraints:

- Foreign key contraints: deptID from the Department table is the foreign key.
- Data entry constraints: the minimum credits for all courses should be atleast 1 and the maximum credits for all the courses should be atleast equal to the respective minimum credits for the courses.

Functional Dependencies:

- courseID sectionID
- courseID title
- courseID maxCreditts
- courseID minCredits
- courseID participationLevel

Table: CourseOffering

Constraints:

- Uniqueness constraints: The courseofferingID for all the courses should be unique which is a combination of courseID and sectionID.
- Foreign key constraints: The combination of (courseID, deptID) from the Course table and the semesterID from the Semester are the foreign keys.

- Data entry Constraints: The classDays must be between 1 and 9. Where 1 ->Monday, 2->Tuesday, 3->Wednesday, 4->Thursday, 5->Friday, 6->Mon,wed, 7->Tue,Thur , 8->Mon, Wed, Fri, 9->any other combination. Also, The seats available at any point for a course offering should be at most the total number of seats offered for that course offering.

Functional dependencies:

- courseID, sectionID, semesterID classDays
- courseID, sectionID, semesterID startTime
- courseID, sectionID, semesterID endTime
- courseID, sectionID, semesterID location
- courseID, sectionID, semesterID totalSeatsOffered
- courseID, sectionID, semesterID seatsAvailable
- courseID, sectionID, semesterID waitlistLimit

Table: Semester

Constraints:

- Uniqueness Constraint: The semesterID should be unique for all the semesters.
- Data entry constraints: The drop date for all courses should be on or after the add date for the course and the semester end date should be on or after the semester start date.

Functional Dependencies:

- semesterID season
- semesterID year
- semesterID addDate
- semesterID dropDate
- semesterID semStartDate
- semesterID semEndDate

Table: CreditRequirements

Constraints:

- Uniqueness constraint: The creditReqID for all the entries should be unique.
- Data Entry Constraints: The minimum credit required should be equal to or more than 0 and the maximum credits required should be atleast equal to the minimum credits required. The participation level must be between 1 and 2 where 1=undergraduate and 2=graduate. The residency level must be between 1, 2 and 3 where 1=in state, 2=out of state and 3=international.

Functional dependencies:

We have decided to use participationLevel and residencyLevel as composite primary key as against using the creditReqID

- participationLevel, residencyLevel maxCreditsReq

- participationLevel, residencyLevel □ minCreditsReq

Table: Billing

Constraints:

- Uniqueness constraints: The unityID from the UnityUsers table is the foreign key.

Functional Dependencies:

- unityID, billID □ billAmount
- unityID, billID □ dueDate
- unityID, billID □ outstandingAmount

Table: BillingRate

Constraints:

- Foreign key constraints: the semesterID from Semester table and the creditReqID from CreditRequirements table are the foreign keys.

Functional dependencies:

- participationLevel, residencyLevel, semesterID □ $/CreditHour

Table: PermissionMapping

Constraints:

- Foreign key constraints: The specialPermissionID from SpecialPermissions table and courseOfferingID from CourseOffering table are the foreign keys.

Table: SpecialPermissions

- specialPermissionID □ specialPermissionCode

Table: SpecialPermissionRequests

Constraints:

- Foreign key constraints: The combination of (specialPermissionID, courseOfferingID) from PermissionMapping table , unityID from Students table and unityID from the Admin table are the foreign keys.

Table: Prerequisites

Constraints:

- Foreign key constraints: The courseOfferingID from CourseOffering table and the combination of (courseID, deptID) from the Course table are the foreign keys.
- Data entry constraints: The GPA for all courses for all students must be between 0.00 and 4.334.

Functional dependencies:

- currentCourseID corresponds to courseID in CourseOffering table.
- prereqCourseID corresponds to courseID in Course table.
- currentCourseID, sectionID, semesterID, prereqCourseID grade

Table: Transcript

Constraints:

- Foreign key constraints: The unityID from Students table and courseOfferingID from courseOffering table are the foreign keys.
- Data entry constraints: The course grades should be between A+, A, A-, B+, B. B-, C+, C, C-, F, S, U, AU, INC, PROG where F=Fail, S=Satisfactory, U=Unsatisfactory, AU=Audit, INC=Incomplete, PROG=in Progress. The credit hours taken for all students for all courses should be atleast 1.

Functional Dependencies:

- unityID, courseID, sectionID, semesterID grade
- unityID, courseID, sectionID, semesterID numberOfCredits

Table: Facilitates

Constraints:

- Foreign key constraints: The courseOfferingID from courseOffering table and unityID from the Faculty table are the foreign keys.

Table: Waitlist

Constraints:

- Foreign key constraints: The unityID from Students table and courseOfferingID from CourseOffering table are the foreign keys.

Functional dependencies:

- unityID, courseID, sectionID, semesterID waitlistPosition

Normal Form discussion:

- The application is in the 2 NF normal form. 2NF is easy to implement and less complex as compared to 3NF.
- 3NF increases the number of tables and thus the complexity.
- Initial design was devoid of transitive dependencies. But later there were many modifications as we built the database and there might have been transitive dependencies introduced. Hence it is not completely in 3NF.