James McSweeney

Logical Database Project

2A)

DepartmentName-> chairName, NumFacualty

StudentID-> Major, Name, Initials AttendedEvent

Major-> MajorCode

EventID-> EventName, StartDate,EndDate

2B)

All of my entities(Department, Students, Major, Event) are in third normal form. Since there are no repeating groups, transitive dependencies, or partial dependencies.

I did find that Student had a transitive dependency with name and initial, but adding that entity would create more complexity then it would reduce complexity. Therefor I choose not to include it in the project.

2c) User transactions:

1)The user wants to know details about the events the CS department is having:

The details for departments are store in Department() and the details for events are stored in Event(). So by using the relationship departments hosts event.

2)If a user wanted to know what department the Economics major is in.

The details for majors are store in Major() and the details for Departments are stored in Department(). So by using the relationship departments creates major.

3) If a student wants to know what their major code is.

The student would find their name and studentID in Student() and use the relationship Student isEnrolledIn major. To access the majorCode from the Major entity.

4) How many Majors are there?

Preform a count operation on the Major attribute in entity major.

5) What events are starting on thanksgiving?

Return the names of the events, with a WHERE statement to filter the StartDate attribute for thanksgiving which will show the names of events with that starting date.

2D)

Primary Key Constraints

DepartmentName has to start with “Department” (Department%)

Major is not NULL

StudentID is unique

EventID is unique

Other Constraints

Initials must be more than one character long.

Major codes must be three characters

As it is logical, an event cannot end before the start date

event cannot be a past date or the current date

James McSweeney Logical Model Database Project 1