Eitan Raitses
CSC 423
Professor Aguiar
Project Part 2

Assumptions (copied from Part 1)
A major does not need to be declared by a student to be offered by a department
Every department has a chair
Departments must have one or more faculty
Events must have at least one hosting department
Events hosted by multiple departments share the same eventID, with different deptIDs

New Assumptions for Part 2
A student can only attend events hosted by the department(s) that offers their major(s)
A student can only declare one major per department
Events can share the same name, yet have different eventIDs

A. Derived Relations

Department (deptID, deptName, chairName, numFaculty, **majorID**)
Major (majorID, majorName, **deptID**)
Student (sID, sName, sInit, **majorID**, **eventID**)
Event (eventID, **deptID,** eventName, sDate, fDate, **sID**)
(**sID, majorID**)
StudentEvent (**sID, eventID**)

B. Normalization to 3NF

UNF → 1NF (remove repeating groups, flatten table)
Repeating groups:

1. In Department:
    a. There can be multiple majors per department, so majorID is removed
2. In Student:
    a. Students can declare more than one major and attend more than one event, so majorID and eventID are removed
3. In Event:
    a. Each event will be attended by one or more students, so sID is removed

1NF → 2NF (remove partial dependencies)
Partial dependencies:

1. In event:
    a. eventID → eventName, sDate, fDate, so create new table to store event info related to specific eventID

2NF → 3NF (remove transitive dependencies)
Transitive dependencies:

1. In Student:
    a. sName → sInit, but do not remove sInit due to the complexity of potential joins of an additional table which stores the initials

3NF corrected

Department (deptID, deptName, chairName, numFaculty)
Major (majorID, majorName, **deptID**)
Student (sID, sName, sInit)
Event (eventID, eventName, sDate, fDate)
DeptEvent (**deptID, eventID**)
DeptMajor (**deptID, majorID**)
StudentEvent (**sID, eventID**)

C. Validate relations against user transactions
    a. List the details of all events hosted by the biology department
        i. Select deptID, deptName, eventID, eventName, sDate, fDate from the Department table, the Event table, and the EventInfo table where the eventID in Event and EventInfo match, the deptID in Event and Department match, and the the deptName in department is equal to BIO.
    b. List all students with majors in Computer Science and Finance
        i. Select sID from Major, Student, and StudentMajor where the sID in StudentMajor and Student match, the majorID in StudentMajor and Major match, and the majorName in Major contains both Finance and Computer Science.
    c. List details of all departments
        i. Select * from Department
    d. List names of all students enrolled in majors offered by departments with over 50 faculty members
        i. Select sName from Department, Major, Student, StudentMajor where sID in Student and StudentMajor match, majorID in Major and StudentMajor match, deptID in Major and Department match, and numFaculty in Department is > 50.
    e. List eventName and start dates for events attended by students with Marketing majors
        i. Select eventName, sDate from Major, StudentMajor, StudentEvent, EventInfo, where majorID in Major and StudentMajor match, sID in StudentMajor and StudentEvent match, eventID in studentEvent and EventInfo match, and majorName in Major equals Marketing.

D. Integrity Constraints

Department (deptID, deptName, chairName, numFaculty)
Primary Key deptID

Major (majorID, majorName, deptID)
Primary Key majorID
Foreign Key deptID **references** Department(deptID) ON UPDATE CASCADE ON DELETE CASCADE

Student (sID, sName, sInit)
Primary Key sID

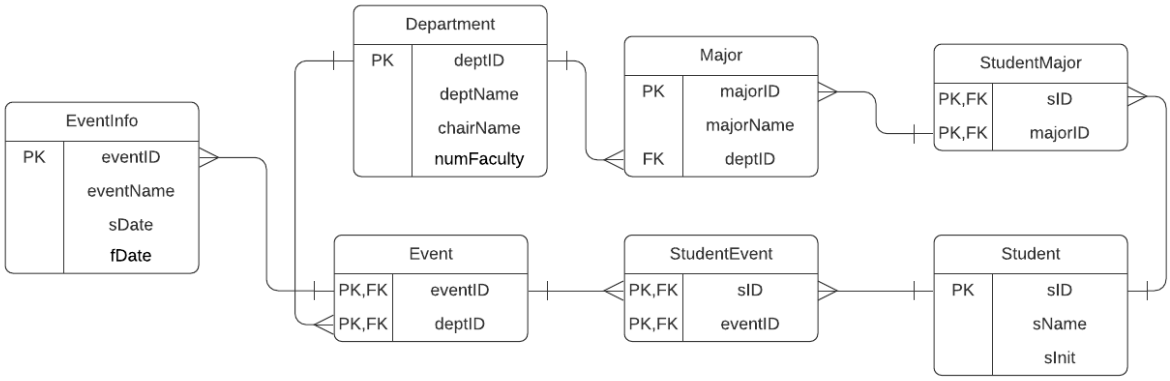
Event (eventID, deptID)
Primary Key eventID, deptID
Foreign Key eventID **references** EventInfo(eventID) ON UPDATE CASCADE ON DELETE CASCADE
Foreign Key deptID **references** Department(deptID) ON UPDATE CASCADE ON DELETE NO ACTION

EventInfo (eventID, eventName, sDate, fDate)
Primary Key eventID
Constraint fDate > sDate
Constraint ON INSERTION sDate > currentDate

StudentMajor (sID, majorID)
Primary Key sID, majorID
Foreign Key sID **references** Student(sID) ON UPDATE CASCADE ON DELETE CASCADE
Foreign Key majorID **references** Major(majorID) ON UPDATE CASCADE ON DELETE CASCADE

StudentEvent (sID, eventID)
Primary Key sID, eventID
Foreign Key sID **references** Student(sID) ON UPDATE CASCADE ON DELETE CASCADE
Foreign Key eventID **references** Event(eventID) ON UPDATE CASCADE ON DELETE CASCADE

E. ER Diagram


Department
PK
deptID
deptName
chairName
numFaculty
Major
PK
majorID
majorName
FK
deptID
StudentMajor
PK,FK
sID
PK,FK
majorID
EventInfo
PK
eventID
eventName
sDate
fDate
Event
PK,FK
eventID
PK,FK
deptID
StudentEvent
PK,FK
sID
PK,FK
eventID
Student
PK
sID
sName
sInit