Database Part 1

James McSweeney

Department(**DepartmentName**, chairName, NumFacualty)

Canidate Keys

(DepartmentName)

(Chairname)

Assumption: DepartmentName is a unique ID for each department. Chair names are also unique.

Primary key: DepartmentName

I choose DepartmentName as the primary key since it is unique and less likely to be updated then the chairname.

Students**(StudentID,** Major, Name, initials, AttendedEvent)

Commented [MJ1]: The Major in students will be a list of strings, while the major in major will just be a string. should I give these differnet names or make both lists of strings even though major will always be 1

Canidate Keys

(StudentID)

(Major,name)

Assumption: StudentID is a unique value associated for each student.

Primary Key StudentID

I choose StudentID as the primary key because it is one attribute and unique for each student.

Major(**Major,** DepartmentCode)

Canidate Keys

(Major)

Assumption: Major names are unique

Primary Key: Major

Event(**EventID**, EventName, StartDate,EndDate)

Canidate Keys

(EventID)

(EventName,StartDate)

Assumption: EventID will be a unique value associated with each event.

Primary Key: EventID

ASSUMPTION: the university allows majors to be available without a student in them. Events can have the same name. Departments do not have the same name.

Enity1	relationship	enity2	partipation	cardinatlity	mulitplicty	Type of Rel
department Event	**Hosts** Funded	Events Departments	1 1	* *	1..* 1..*	Many to many
department Major	**create** references	Major departments	1 1	* 1	1..* 1...1	One to many
Student major	**IsEnrolledIn** has	Major students	1 0	* *	1..* 0..*	Many to many
Student Event	**GoTo** gathers	Events students	1 1	* *	1..* 1...*	Many to many

Constraints:

DepartmentName constraint like ‘Department%’

Major constaint is not NULL (can be a vector of strings for multiple majors)

James McSweeney

Conceptual Model Database Project 1