

Database Part 1

James McSweeney

Department(DepartmentName, chairName, NumFaculty)

Canidate Keys

(DepartmentName)

Assumption: DepartmentName is a unique ID for each department.

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)

Canidate Keys

(StudentID)

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, MajorCode)

Canidate Keys

(Major)

Assumption: Major names are unique

Primary Key: Major

Event(EventID, EventName, StartDate, EndDate)

Canidate Keys

(EventID)

(EventName, StartDate)

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 different names or make both lists of strings even though major will always be 1

Assumption: EventID will be a unique value associated with each event. there can't be two events with the same name starting at the same time

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.

Entity1	relationship	entity2	participation	cardinality	multiplicity	Type of Rel
department	Hosts	Events	0	*	0..*	Many to
Event	Funded	Departments	1	*	1..*	many
department	create	Major	1	*	1..*	One to
Major	references	departments	1	1	1...1	many
Student	IsEnrolledIn	Major	1	*	1..*	Many to
major	has	students	0	*	0..*	many
Student	GoTo	Events	1	*	1..*	Many to
Event	gathers	students	1	*	1...*	many

Constraints:

DepartmentName constraint like 'Department%'

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

James McSweeney

Conceptual Model Database Project 1



