CS 4400: Intro to Database Systems

- Translating Your Enhanced Entity-Relationship (EER) Diagram to Relational Schema
- It's all about "the algorithm"...
- See Chapter 9 of the 7th Edition textbook

- These slides show the **Relational Schema generated** from the corresponding EER Diagram

Building = (BldgID, Motto

Start with the Strong Entities
Adding the Building Entity

Adding the Department Entity

Building = (BldgID, Motto

Department = (DeptID, Name

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits

fk1: DeptID -> Department.DeptID

Next, work on the Weak Entities
Adding the Course Entity

Next, work on the Superclass and Subclass
Entities “from the top down”
Adding the User Superclass

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

There are multiple ways to add the Instructor and Student Subclasses...

Version A

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear
fk3: LoginID -> User.LoginID

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

Version B

~~User = (LoginID, SSN, Email, FirstName, LastName, Title~~

Instructor = (LoginID, SSN, Email, FirstName, LastName, Title, Field

Student = (LoginID, SSN, Email, FirstName, LastName, Title, GradYear

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

Version C

User = (LoginID, SSN, Email, FirstName, LastName, Title, Role, Field, GradYear

~~Instructor = (LoginID [fk2], Field~~
~~fk2: LoginID -> User.LoginID~~

~~Student = (LoginID [fk3], GradYear~~
~~fk3: LoginID -> User.LoginID~~

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

Version D

User = (LoginID, SSN, Email, FirstName, LastName, Title, Is_Instructor, Is_Student, Field, GradYear

~~Instructor = (LoginID [fk2], Field~~
~~fk2: LoginID -> User.LoginID~~

~~Student = (LoginID [fk3], GradYear~~
~~fk3: LoginID -> User.LoginID~~

We are using Version A for demonstration purposes – any of the approved versions are OK, and all have strengths and weaknesses

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear
fk3: LoginID -> User.LoginID

Building = (BldgID, Motto

Next, work on the Multi-Valued Attributes
Adding the Awards multi-valued Attribute

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title, ~~Awards~~

one tuple, one attribute => one value!

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear
fk3: LoginID -> User.LoginID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Building = (BldgID, Motto

Next, work on the Relationships
Any reasonable order is possible – one to one, one to many or many to many
Adding the Mentor 1:N relationship

You could also create a new relation

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field, ~~Students Mentored~~
fk2: LoginID -> User.LoginID

remember – one tuple, one attribute => one value!

Student = (LoginID [fk3], GradYear, Mentor [fk5]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Adding the Live-At and Major-In 1:N relationships

Building = (BldgID, Motto

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Building = (BldgID, Motto, Advisor [fk8]
fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Adding the Advisor 1:1 relationship

You could also add a new relation, or even merge the two relations representing the entities

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits
fk1: DeptID -> Department.DeptID

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

avoid nulls when reasonably possible...

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Adding the Director 1:1 relationship

Building = (BldgID, Motto, Advisor [fk8]
fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits, Director [fk9]
fk1: DeptID -> Department.DeptID
fk9: Director -> Instructor.LoginID, Director is non-null

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Adding the Teach M:N relationship

Building = (BldgID, Motto, Advisor [fk8]
fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits, Director [fk9]
fk1: DeptID -> Department.DeptID
fk9: Director -> Instructor.LoginID, Director is non-null

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Teach = (LoginID [fk10], DeptID, Number [fk11]
fk10: LoginID -> Instructor.LoginID,
fk11: DeptID, Number -> Course.DeptID, Course.Number

Adding the Take M:N relationship with relationship-based attributes

Building = (BldgID, Motto, Advisor [fk8]

fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits, Director [fk9]

fk1: DeptID -> Department.DeptID

fk9: Director -> Instructor.LoginID, Director is non-null

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field

fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]

fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null

fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)

fk4: LoginID -> User.LoginID

Teach = (LoginID [fk10], DeptID, Number [fk11]

fk10: LoginID -> Instructor.LoginID,

fk11: DeptID, Number -> Course.DeptID, Course.Number

Take = (LoginID [fk12], DeptID, Number [fk13], Grade, Semester

fk12: LoginID -> Student.LoginID,

fk13: DeptID, Number -> Course.DeptID, Course.Number

Adding the recursive Prereq M:N relationship – modify the names for clarity

Building = (BldgID, Motto, Advisor [fk8]
fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Department = (DeptID, Name

Course = (DeptID [fk1], Number, Name, Credits, Director [fk9]
fk1: DeptID -> Department.DeptID
fk9: Director -> Instructor.LoginID, Director is non-null

User = (LoginID, SSN, Email, FirstName, LastName, Title

Instructor = (LoginID [fk2], Field
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7]
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Teach = (LoginID [fk10], DeptID, Number [fk11]
fk10: LoginID -> Instructor.LoginID,
fk11: DeptID, Number -> Course.DeptID, Course.Number

Take = (LoginID [fk12], DeptID, Number [fk13], Grade, Semester
fk12: LoginID -> Student.LoginID,
fk13: DeptID, Number -> Course.DeptID, Course.Number

Prereq = (Before-DeptID, Before-Number [fk14], After-DeptID, After-Number [fk15]
fk14: Before-DeptID, Before-Number -> Course.DeptID, Course.Number
fk15: After-DeptID, After-Number -> Course.DeptID, Course.Number

Final results: relational schema conversion

Building = (BldgID, Motto, Advisor [fk8])
fk8: Advisor -> Instructor.LoginID, Advisor is non-null

Department = (DeptID, Name)

Course = (DeptID [fk1], Number, Name, Credits, Director [fk9])
fk1: DeptID -> Department.DeptID
fk9: Director -> Instructor.LoginID, Director is non-null

User = (LoginID, SSN, Email, FirstName, LastName, Title)

Instructor = (LoginID [fk2], Field)
fk2: LoginID -> User.LoginID

Student = (LoginID [fk3], GradYear, Mentor [fk5], Live-At [fk6], Major-In [fk7])
fk3: LoginID -> User.LoginID, fk5: Mentor -> Instructor.LoginID, Mentor is non-null
fk6: Live-At -> Building.BldgID, Live-At is non-null, fk7: Major-In -> Department.DeptID

Awards = (LoginID [fk4], Award-Name)
fk4: LoginID -> User.LoginID

Teach = (LoginID [fk10], DeptID, Number [fk11])
fk10: LoginID -> Instructor.LoginID,
fk11: DeptID, Number -> Course.DeptID, Course.Number

Take = (LoginID [fk12], DeptID, Number [fk13], Grade, Semester)
fk12: LoginID -> Student.LoginID,
fk13: DeptID, Number -> Course.DeptID, Course.Number

Prereq = (Before-DeptID, Before-Number [fk14], After-DeptID, After-Number [fk15])
fk14: Before-DeptID, Before-Number -> Course.DeptID, Course.Number
fk15: After-DeptID, After-Number -> Course.DeptID, Course.Number