

Logical Data Design

Using ER Methodology to Design
Relational Database Schemas

John Sieg, UMass Lowell

The Development Process, 1

- Decide what miniworld (part of the real world) the database and its applications will model. Collect requirements. Analyze the requirements.
- Conceptually design the data by coming up with an ER diagram.
- Logically design the data by mapping the ER diagram onto relation names, schemas, and keys.
- Physically design the data. Choose indices and other access structures.

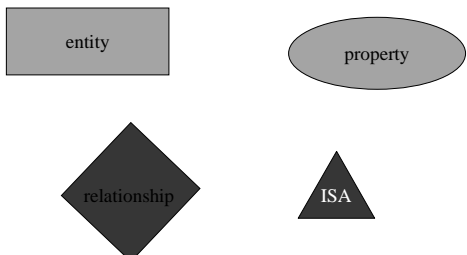
John Sieg, UMass Lowell

The Development Process, 2

- Create the database. Insert sample data.
- Design and implement the application.
- Populate the database with “real” data.
- Gain experience running the application.
- Evaluate the earlier decisions and iterate, if necessary.

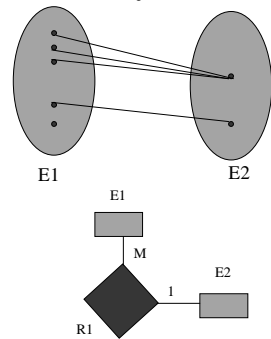
John Sieg, UMass Lowell

ER Diagrams: Primitives



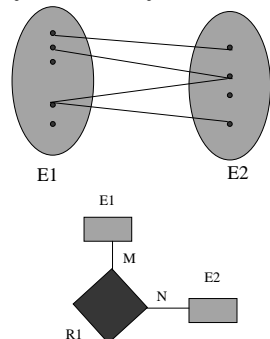
John Sieg, UMass Lowell

One-to-Many Relationships

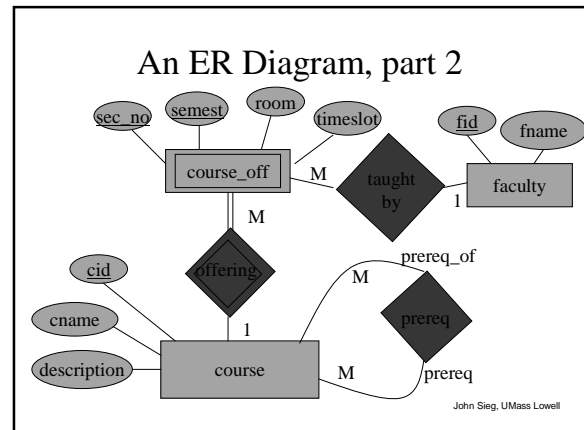
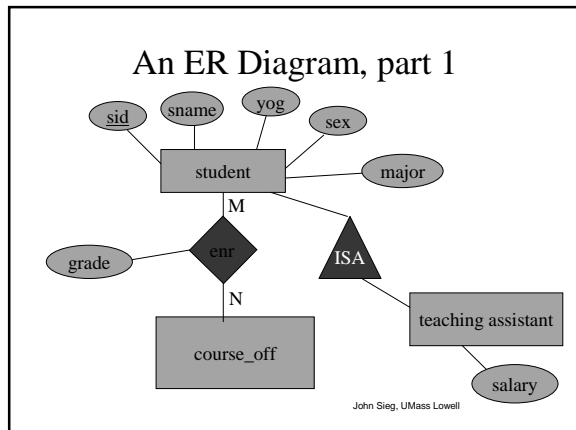


John Sieg, UMass Lowell

Many-to-Many Relationships



John Sieg, UMass Lowell



- ### Guidelines for ER Diagrams
- Be sure that entity classes correspond to collections.
 - There should be no isolated entity classes, i.e., entity classes that aren't related to any other entity classes.
 - Eliminate aggregate properties, set-valued properties, and foreign keys.
 - Don't forget to underline properties that are part of a key.
- John Sieg, UMass Lowell

- ### ER Diagrams → Relations: Rule 1
- A **strong entity class** becomes a relation.
 - The relation schema is identical to the entity class's set of properties.
 - The key of the entity class becomes the primary key of the relation schema.
- John Sieg, UMass Lowell

- ### ER Diagrams → Relations: Rule 2
- A **weak entity class** becomes a relation.
 - The relation schema contains the entity class's properties and the key of the entity class on which the weak entity class depends.
 - The **full** key of the entity class becomes the primary key of the schema.
 - The weak relationship “disappears.” If this relationship has any properties, they become attributes of the “weak” relation's schema.
- John Sieg, UMass Lowell

- ### ER Diagrams → Relations: Rule 3
- A **many-to-many relationship** becomes a relation.
 - The relation schema contains the keys of the entity classes related by the relationship, plus any properties of the relationship.
 - Each of the keys of the related entity classes is a foreign key of the new entity class.
 - The union of the keys of the related entities becomes the primary key of the relation schema.
- John Sieg, UMass Lowell

ER Diagrams → Relations: Rule 4

- A *many-to-one relationship* is “absorbed” by the relation corresponding to the entity class at the “many” side.
- This is done by adding to the schema of the relation corresponding to the “many” side the key of the relation corresponding to the “one” side.
- The added attributes are a foreign key.

John Sieg, UMass Lowell

ER Diagrams → Relations: Rule 4, cont.

- If the relationship has any properties, these are also added to the schema of the relation corresponding to the “many” side.
- If the many-to-one relationship connects a weak entity class with the entity class it depends on, this rule has no effect beyond the effect of Rule 2.

John Sieg, UMass Lowell

ER Diagrams → Relations: Rule 5

- An entity class E2 that is a *subclass* of an entity class E1 becomes a relation R2 whose schema includes E2's properties and E1's key.
- An instance of entity class E2 is represented by a tuple in R2 and a tuple in the relation (R1) corresponding to entity class E1.
- E1's key becomes R2's primary key.
- R2's primary key is also a foreign key referring to R1.

John Sieg, UMass Lowell

Relation Schemas for Sample ER Diagram

- `student(sid, sname, yog, gpa, sex, major)`
- `faculty(fid, fname)`
- `course(cid, name, description)`
- `course_off(cid, sec_no, semester, room, time_slot)`

John Sieg, UMass Lowell

Relation Schemas for Sample ER Diagram, cont'd

- `enr(sid, cid, sec_no, semester, grade)`
- `prereq(prereq_cid, prereq_of_cid)`
- **Revision:**
`course_off(cid, sec_no, semester, room, time_slot, fid)`
- `ta(sid, salary)`

John Sieg, UMass Lowell

Conclusions

- ER methodology often yields a good relational schema design, because “real” objects get mapped onto database objects.
- Transformation to relation schemas can be automated. But choosing ER diagrams requires good judgment.
- ER methodology isn't perfect. Other methodologies sometimes have to be applied.

John Sieg, UMass Lowell