CIT 381

Modeling Basics

-relationship types
-foreign keys
-ER Studio
Entities

• A concrete item we wish to represent in our database.
• Examples: books, cars, employees, wombats, accounts, offices.
• Completely described by its attributes.

important!!
Relationship

• An association between two or more entities.
• Orders is a relationship between entities Customers and Books.
• Assign is a relationship between Cars and Employees.
First Step: Design Entities

• What are the important objects that we want to keep information for?

  • Example: cars, employees, customers,…

• For each of these objects, write down all of the attributes we will need to keep.

  • Example: cars - licenseNum, vin, make, model, year, odometerRead
The primary key is the chosen identifier. At this stage, all other attributes belong solely to that entity. (Foreign keys come later.)
Next Step: Relationships

- Cars are **assigned** to employees (or vice versa)
- **assign** is a *relationship* between cars and employees
Relationship Cardinality

• Relationships can be
  – one-to-many (1:N)
  – many-to-one (N:1)
  – many-to-many (N:M)
  – one-to-one (1:1)

• Important to know the cardinality of each relationship as it affects the table design.
One-to-many

- Suppose each car can be assigned to more than one employee, but each employee has exactly one car.
One-to-many (optional)

- Same as previous, but an employee does not have to have a car – each employee has zero or one cars.

The open diamond on the left indicates an optional relationship: an employee does not have to have a car. This means that the `car_license_num` field in `employees` can be NULL (that is, blank).
Many-to-one

• All cars are assigned to one employee.
• An employee can have more than one car.
Many-to-Many

- A car can be assigned to many employees.
- An employee can be assigned to many cars.

ER Studio calls this a non-specific mandatory relationship.
Resolving many-to-many

- A M:N relationship needs to be represented by its own table, sometimes called a bridging table.

Notice the solid lines and curved corners. It is a weak entity.
Weak entities

• A weak entity is one that does not have enough information in its attributes to form a key (unique identifier).

• Consider a bank check: check number 3512 does not identify it (unless we know the account number).

• Note that account number is an attribute of account, not of check.
Weak entity example (1)

custID identifies customer, accountNum identifies account, checkNum only partially identifies check
Weak entity example (2)

We added a mandatory one-to-many relationship, as before.
Weak entity example (3)

We insert an identifying relationship:

- solid line
- makes foreign key (accountNum) part of the primary key of check
- curved corners on check indicate it is a weak entity
Another weak entity

From the company database, handed out in class.

Oops, this is IDEF1X notation – how do we convert it to Crowsfoot?