CIS560

Design Patterns & Practices

Part 1

General Recommendations

- Table names should be nouns
  - Relationships in a conceptual model are often verbs, but consider a descriptive noun
  - Example: Rather than “Produced”, consider “ProducedAlbum”

General Recommendations

- Column names should be nouns
- Exception is the bit data type
  - Consider questions with a yes/no answer
  - Examples

    IsActive, IsRemoved
  - Avoid negatives

    IsInactive, IsNotRemoved

General Recommendations

- Column names should not be redundant
  - Don’t repeat the name of the table
  - Car.Price rather than Car.CarPrice
  - Exceptions are keys
    - They are used in other tables
    - Could use different names for references
    - However, matching names improves readability in queries
- Be CONSISTENT: PersonId vs. PersonID

General Recommendations

- Think twice before allowing DELETE
  - You never regret keeping the data
  - Use a column indicating deletion
- Use surrogate keys
  - Updating natural keys can be cumbersome
  - Updating natural keys has performance implications

Review: Relationship Types

- One-to-Many
  - Zero or more → One and only one
  - One or more → One and only one (logical only)
  - Zero or more → Zero or one
- One-to-One
  - Zero or one → One and only one
- Many-to-Many

  Implemented with a “linking” or “bridge” table

Variations

- Multi-way relationships
  - Multiple entities are referenced
- Multiple roles
  - Multiple foreign keys to same table
- Self-referencing entities
  - Example: Employee/Organization Chart

Subclasses – Three Approaches

- Object-Oriented Approach

  A table for each type, and possibly a general type

- Nullable columns

  A single table with nullable columns

- E/R Style

  Use a supertype or "base class"

Subclasses – OO Approach

- Common attributes in all types
- No foreign keys
- Tuples inserted in applicable type
  - No base type
  - May need a general type

Subclasses – Nullable Columns

- A single table
- Non-nullable columns for shared attributes
- Nullable columns for attributes of all other types

Subclasses – E/R Style

- A single supertype (base class)
  - Contains the key
  - Contains the common attributes
  - Contains all tuples
- Each subtype
  - Contains the key
  - Contains only specific attributes
  - Contains only tuples of that that subtype