Introduction to Data Management
CSE 414

Unit 6: Conceptual Design
E/R Diagrams
Integrity Constraints
BCNF

(3 lectures)
Introduction to Data Management
CSE 414

E/R Diagrams
Class Overview

• Unit 1: Intro
• Unit 2: Relational Data Models and Query Languages
• Unit 3: Non-relational data
• Unit 4: RDMBS internals and query optimization
• Unit 5: Parallel query processing
• Unit 6: DBMS usability, conceptual design
  – E/R diagrams
  – Schema normalization
• Unit 7: Transactions
• Unit 8: Advanced topics (time permitting)
Database Design

What it is:

• Starting from scratch, design the database schema: relation, attributes, keys, foreign keys, constraints etc

Why it’s hard

• The database will be in operation for a very long time (years). Updating the schema while in production is very expensive (why?)
Database Design

• Consider issues such as:
  – What entities to model
  – How entities are related
  – What constraints exist in the domain

• Several formalisms exists
  – We discuss E/R diagrams
  – UML, model-driven architecture

• Reading: Sec. 4.1-4.6
Database Design Process

Conceptual Model:

Relational Model:
Tables + constraints
And also functional dep.

Normalization:
Eliminates anomalies

Conceptual Schema

Physical storage details
Physical Schema
Entity / Relationship Diagrams

- Entity set = a class
  - An entity = an object
- Attribute
- Relationship

CSE 414 – Autumn 2018
Keys in E/R Diagrams

• Every entity set must have a key

![Diagram showing a Product entity with attributes name and price]
What is a Relation?

• A mathematical definition:
  – if A, B are sets, then a relation R is a subset of $A \times B$

• $A=\{1,2,3\}$, $B=\{a,b,c,d\}$,
  $A \times B = \{(1,a),(1,b), \ldots, (3,d)\}$
  $R = \{(1,a), (1,c), (3,b)\}$

• makes is a subset of $\text{Product} \times \text{Company}$:
Multiplicity of E/R Relations

• one-one:
Multiplicity of E/R Relations

- one-one:

- many-one
Multiplicity of E/R Relations

- one-one:

- many-one

- many-many
What does this say?
Attributes on Relationships

Person

name

date

address

Buys

What does this say?

Product

name

price
Multi-way Relationships

How do we model a purchase relationship between buyers, products and stores?

Can still model as a mathematical set (How?)

As a set of triples \( \subseteq \text{Person} \times \text{Product} \times \text{Store} \)
Q: What does the arrow mean?

A: Any person buys a given product from at most one store

[Fine print: Arrow pointing to E means that if we select one entity from each of the other entity sets in the relationship, those entities are related to at most one entity in E]
Arrows in Multiway Relationships

Q: What does the arrow mean?

A: Any person buys a given product from at most one store AND every store sells to every person at most one product.
Converting Multi-way Relationships to Binary

Arrows go in which direction?
Converting Multi-way Relationships to Binary

- Purchase
  - ProductOf
    - Product
  - StoreOf
    - Store
  - BuyerOf
    - Person

Make sure you understand why!
What’s wrong?

Moral: Be faithful to the specifications of the application!
Design Principles: What’s Wrong?

Moral: pick the right kind of entities.
Design Principles: What’s Wrong?

Moral: don’t complicate life more than it already is.
From E/R Diagrams to Relational Schema

- Entity set $\rightarrow$ relation
- Relationship $\rightarrow$ relation
Entity Set to Relation

**Product**($prod$-ID, category, price)

<table>
<thead>
<tr>
<th>prod-ID</th>
<th>category</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo55</td>
<td>Camera</td>
<td>99.99</td>
</tr>
<tr>
<td>Pokemn19</td>
<td>Toy</td>
<td>29.99</td>
</tr>
</tbody>
</table>
N-N Relationships to Relations

Represent this in relations
N-N Relationships to Relations

Orders\((\text{prod-ID}, \text{cust-ID}, \text{date})\)

Shipment\((\text{prod-ID}, \text{cust-ID}, \text{name}, \text{date})\)

Shipping-Co\((\text{name}, \text{address})\)

<table>
<thead>
<tr>
<th>prod-ID</th>
<th>cust-ID</th>
<th>name</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo55</td>
<td>Joe12</td>
<td>UPS</td>
<td>4/10/2011</td>
</tr>
<tr>
<td>Gizmo55</td>
<td>Joe12</td>
<td>FEDEX</td>
<td>4/9/2011</td>
</tr>
</tbody>
</table>
N-1 Relationships to Relations

Represent this in relations
Orders\((\text{prod-ID}, \text{cust-ID}, \text{date1}, \text{name}, \text{date2})\)
Shipping-Co\((\text{name}, \text{address})\)

Remember: no separate relations for many-one relationship
Modeling Subclasses

Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
<th>Platforms</th>
<th>Age-group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
<td>unix</td>
<td>NULL</td>
</tr>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
<td>NULL</td>
<td>infant</td>
</tr>
</tbody>
</table>
## Modeling Subclasses

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
<th>Platforms</th>
<th>Age-group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
<td>unix</td>
<td>NULL</td>
</tr>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
<td>NULL</td>
<td>infant</td>
</tr>
</tbody>
</table>

- **Products**
  - **Software products**
  - **Educational products**
### Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
</tr>
</tbody>
</table>

### Software Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
<td>unix</td>
</tr>
</tbody>
</table>

### Educational Product

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
<th>Age-group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
<td>infant</td>
</tr>
</tbody>
</table>
Modeling Subclasses

Some objects in a class may be special
  • define a new class
  • better: define a subclass

Products

  Software products

  Educational products

So --- we define subclasses in E/R
Subclasses to Relations

Product
- name
- category
- price

Software Product
- isa
- platforms

Educational Product
- isa
- Age Group

Sw.Product
- isa

Ed.Product
- Name
- Age Group

Other ways to convert are possible

<table>
<thead>
<tr>
<th>Name</th>
<th>Price</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>99</td>
<td>gadget</td>
</tr>
<tr>
<td>Camera</td>
<td>49</td>
<td>photo</td>
</tr>
<tr>
<td>Toy</td>
<td>39</td>
<td>gadget</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>unix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Age Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>toddler</td>
</tr>
<tr>
<td>Toy</td>
<td>retired</td>
</tr>
</tbody>
</table>
Modeling Union Types with Subclasses

Say: each piece of furniture is owned either by a person or by a company
Modeling Union Types with Subclasses

Say: each piece of furniture is owned either by a person or by a company

Solution 1. Acceptable but imperfect (What’s wrong?)
Modeling Union Types with Subclasses

Solution 2: better, more laborious

- Person isa FurniturePiece
- Company isa FurniturePiece
- Owner ownedBy FurniturePiece

CSE 414 – Autumn 2018
Weak Entity Sets

Entity sets are weak when their key comes from other classes to which they are related.

Team(sport, number, universityName)
University(name)
Introduction to Data Management
CSE 344

Integrity Constraints
Integrity Constraints Motivation

An integrity constraint is a condition specified on a database schema that restricts the data that can be stored in an instance of the database.

- ICs help prevent entry of incorrect information
- How? DBMS enforces integrity constraints
  - Allows only legal database instances (i.e., those that satisfy all constraints) to exist
  - Ensures that all necessary checks are always performed and avoids duplicating the verification logic in each application
Constraints in E/R Diagrams

Finding constraints is part of the modeling process. Commonly used constraints:

**Keys:** social security number uniquely identifies a person.

**Single-value constraints:** a person can have only one father.

**Referential integrity constraints:** if you work for a company, it must exist in the database.

**Other constraints:** peoples’ ages are between 0 and 150.
Keys in E/R Diagrams

Underline:

No formal way to specify multiple keys in E/R diagrams
Single Value Constraints

makes

vs.

makes
Referential Integrity Constraints

Each product made by at most one company. Some products made by no company.

Each product made by exactly one company.