CIT 381

introduction to relations
### Tables

<table>
<thead>
<tr>
<th>movieId</th>
<th>title</th>
<th>genre</th>
<th>length</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>The Thirty-Nine Steps</td>
<td>mystery</td>
<td>101</td>
<td>R</td>
</tr>
<tr>
<td>123</td>
<td>Annie Hall</td>
<td>romantic comedy</td>
<td>110</td>
<td>R</td>
</tr>
<tr>
<td>145</td>
<td>Lady and the Tramp</td>
<td>animated comedy</td>
<td>93</td>
<td>PG</td>
</tr>
<tr>
<td>189</td>
<td>Animal House</td>
<td>comedy</td>
<td>87</td>
<td>PG-13</td>
</tr>
<tr>
<td>450</td>
<td>Elizabeth</td>
<td>costume drama</td>
<td>123</td>
<td>PG-13</td>
</tr>
<tr>
<td>553</td>
<td>Stagecoach</td>
<td>western</td>
<td>130</td>
<td>R</td>
</tr>
<tr>
<td>987</td>
<td>Duck Soup</td>
<td>comedy</td>
<td>99</td>
<td>PG-13</td>
</tr>
</tbody>
</table>

This is the movie table (or relation). Its columns are attributes (or fields).
Key basics

• The primary key is a unique identifier of each entry in the table. In the movie table, movieId is clearly the key.

• A table can also have what is called a foreign key, which is the primary key of another table.
Foreign key example

<table>
<thead>
<tr>
<th>accountId</th>
<th>videoId</th>
<th>dateRented</th>
<th>dateDue</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>101</td>
<td>2002-01-03 00:00:00</td>
<td>2002-01-04 00:00:00</td>
<td>1.59</td>
</tr>
<tr>
<td>101</td>
<td>111</td>
<td>2002-04-24 00:00:00</td>
<td>2002-05-02 00:00:00</td>
<td>3.99</td>
</tr>
<tr>
<td>101</td>
<td>112</td>
<td>2002-04-24 00:00:00</td>
<td>2002-04-30 00:00:00</td>
<td>1.99</td>
</tr>
<tr>
<td>101</td>
<td>113</td>
<td>2002-02-22 00:00:00</td>
<td>2002-02-25 00:00:00</td>
<td>3</td>
</tr>
</tbody>
</table>

This is part of the rental table from bighitvideo.

The videoId is the primary key - these are current rentals, and each video can be rented by one customer at a time.

The attribute accountId is a foreign key, and it refers to an entry in the customer table.
Schema

The schema, also known as a data model, is a description of the tables.

movie
  - movieId: int PRIMARY KEY,
  - title: varchar,
  - genre: varchar,
  - length: int,
  - rating: varchar

rental
  - accountId: int,
  - videoId: int PRIMARY KEY,
  - dateRented: datetime,
  - dateDue: datetime,
  - cost: decimal(5,2),
  - FOREIGN KEY accountId REFERENCES customer,
  - FOREIGN KEY videoId REFERENCES video

No data is involved.
Database design

- ER Design
- Beyond ER Design

1. Requirements analysis
2. Conceptual database design
3. Logical database design
4. Schema refinement
5. Physical database design
6. Application and security design.
Requirements analysis

• Meet with intended database users
• Decide what data is to be stored
• Decide what applications are needed
• Determine which operations are the most frequent.
• This is an ongoing process, and may need to be repeated several times.
  – Users always change their minds.
Conceptual database design

• Develop high-level description of the data

• Typically uses ER diagrams
  (ER = “entity-relationship”)

• buzzword: semantic data model
Example ER design

Built using ER Studio, but could even be done by hand.
Logical database design

- After choosing our DBMS: MySQL, Oracle, SQLServer, DB2, PostgreSQL, Informix, etc...
- convert the ER model into tables supported by the DBMS.
- The schema - table descriptions - is the logical design.

- Note: DBMS = “database management software”
Schema refinement

• Apply normalization and ...
• common sense to table design

• The conversion from ER diagrams
  – can be inelegant
  – might not capture all business rules.
  – At this stage we eyeball the design and clean it up.
Physical database design

• Consider workload demands
• Refine design for performance
  (possibly denormalize)

• Add indexes, cluster tables.
Application and security design

- Design and construct applications to be used by the outside world. Take CIT 382.

- Worry about the security of the connection. Take CIT 383.

- Well, also worry about database security
  - Set table permissions, create views and roles.
  - SQL Injection
Entities

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