Databases 1 – Organisation and Creation

Lecture 14 – COMPSCI111/111G S1 2019
Today’s lecture

• What is a database?
• Understanding how data is organised in a database
• Creating a database in Microsoft Access
What is a database?

• A (typically large) collection of data about a particular topic, organized systematically

• Examples:
  – Catalogue of library books
  – Patients’ files in a clinic
  – Entries in an address book
  – Students in a class

• Computers allow us to store and manage databases that contain very large amounts of information
Aspects of a database

• Before we can create our database, we need to decide how to:
  1. Organize data in our database
  2. Enter data in our database
  3. Retrieve data from our database
  4. Present the retrieved data to the user
Question?

• What websites have you visited that probably use a database?
1. Organising data - models

• A **model** defines how data is organized and structured within the database
  – We’re going to look at the **relational model** in this course

• When deciding what data to store in a database, we need to think about:
  – **Entities**: things about which we store information
    • Eg. students in uni, courses in uni
  – **Relationships**: specific connections among entities
    • Eg. *students* enrolled in *CompSci111/111G*
1. Organising data - tables

- The **relational model** was developed by Edgar Codd in 1970

- Data is stored and organized in tables
  - A table’s columns are called **fields**; an entity’s attributes
  - A table’s rows are called **records**; one instance of an entity

- A collection of tables form a **database**

<table>
<thead>
<tr>
<th>Field</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentId</td>
<td>C. Brown</td>
</tr>
<tr>
<td>12345</td>
<td>12 Apple St.</td>
</tr>
<tr>
<td>67890</td>
<td>34 Pear Ave.</td>
</tr>
<tr>
<td>22222</td>
<td>56 Grape Blvd.</td>
</tr>
<tr>
<td>Name</td>
<td>L. Van Pelt</td>
</tr>
<tr>
<td>555-1234</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>555-5678</td>
</tr>
<tr>
<td>Phone</td>
<td>555-9999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentId</td>
<td>P. Patty</td>
</tr>
<tr>
<td>12345</td>
<td>12 Apple St.</td>
</tr>
<tr>
<td>67890</td>
<td>34 Pear Ave.</td>
</tr>
<tr>
<td>22222</td>
<td>56 Grape Blvd.</td>
</tr>
<tr>
<td>Name</td>
<td>L. Van Pelt</td>
</tr>
<tr>
<td>555-1234</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>555-5678</td>
</tr>
<tr>
<td>Phone</td>
<td>555-9999</td>
</tr>
</tbody>
</table>
1. Organising data

- Tables are connected together using **relationships**, thereby creating connections between different entities.
1. Organising data

• There are two parts to a relationship; **primary key** and **foreign key**

• 1. Primary key:
  – Generally, all tables must have a primary key field
  – All records must have a value in the primary key field
  – The primary key’s value must be unique

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AutoNumber</td>
</tr>
<tr>
<td>First Name</td>
<td>Text</td>
</tr>
<tr>
<td>E-mail Address</td>
<td>Text</td>
</tr>
<tr>
<td>Level</td>
<td>Text</td>
</tr>
<tr>
<td>Room</td>
<td>Text</td>
</tr>
<tr>
<td>Date of Birth</td>
<td>Date/Time</td>
</tr>
</tbody>
</table>
Question?

• Which field makes a good primary key in a table?
1. Organising data

• 2. Foreign key
  – A field in one table that is related to a primary key field in another table
  – Creates a connection between the two fields
  – Can take blank values and/or repeated value depending on the relationship

<table>
<thead>
<tr>
<th>StudentID</th>
<th>Code</th>
<th>Date enrolled</th>
</tr>
</thead>
<tbody>
<tr>
<td>5468975</td>
<td>COMPSCI101</td>
<td>01/01/2016</td>
</tr>
<tr>
<td>1258956</td>
<td>COMPSCI101</td>
<td>15/12/2015</td>
</tr>
<tr>
<td>1258956</td>
<td>COMPSCI107</td>
<td>15/12/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Date of birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>5468975</td>
<td>Joe Cameron</td>
<td>19/08/1992</td>
</tr>
<tr>
<td>1258956</td>
<td>Steve Smith</td>
<td>17/05/1995</td>
</tr>
<tr>
<td>6697826</td>
<td>Tom Bloggs</td>
<td>30/06/1965</td>
</tr>
</tbody>
</table>
Referential integrity

• An important concept underlying relationships between tables
• Referential integrity requires all values of a foreign key field to be:
  – Present in the related primary key field, OR
  – Null (ie. blank)
Referential integrity

<table>
<thead>
<tr>
<th>Students</th>
<th>Enrolments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
<td><strong>StudentID</strong></td>
</tr>
<tr>
<td>5468975</td>
<td>5468975</td>
</tr>
<tr>
<td>1258956</td>
<td>1258956</td>
</tr>
<tr>
<td>6697826</td>
<td>1258956</td>
</tr>
</tbody>
</table>

Insert 9998881, COMPSCI111, 22/12/2015 into Enrolments ✗

Insert 6697826, COMPSCI105, 16/12/2015 into Enrolments ✓

Insert , COMPSCI101, 01/12/2015 into Enrolments ✓

Delete 5468975, from Students ✗

Delete 5468975, from Enrolments ✓
Types of relationships

• There are three kinds of relationship that can exist between tables

• **One to one**: one record in PK related to one record in FK
  – Eg. student can only have one transcript

• **One to many**: one record in PK related to multiple records in FK
  – Eg. student can have multiple emergency contacts

• **Many to many**: multiple records in PK related to multiple records in FK
  – Eg. many students can be enrolled in many papers
Many to Many

• The many-to-many relationships are usually implemented by a pair of one-to-many relationships using three tables.
Exercises

1. What is the primary key and the foreign key (if one exists) for the *Label* table?
2. What is the primary key and the foreign key (if one exists) for the *Artist* table?
3. What is the primary key and the foreign key (if one exists) of the *Albums* table?
Answers

- **Label**
  - PK: ID
  - FK: none

- **Artist**
  - PK: ID
  - FK: LabelID

- **Albums**
  - PK: ID
  - FK: ArtistID
Aspects of a database

• Before we can create our database, we need to decide how to:
  1. **Organize** data in our database
     • Models, tables, relationships
  2. **Enter** data in our database
  3. **Retrieve** data from our database
  4. **Present** the retrieved data to the user
Database Management System (DBMS)

- Application software that is used to manage databases.
- Four main functions:
  - Definition
  - Update
  - Querying
  - Administration
- Examples:
  - Microsoft Access
  - Microsoft SQL Server
Creating a database

Creating a new database
Creating a table
Creating a table

- **Design view:** create/view the fields in the table
- **Datasheet view:** create/view data in the table
Design view
Design view
Datasheet view

- Allows us to enter data into our table
- Need to ensure that we enter the correct type of data in each field (e.g. no text in a number field)
Creating relationships

• Relationships view allows us to create relationships between fields in different tables

• Database Tools tab → Relationships button
Creating relationships
Creating relationships
Creating relationships
Inserting data

- Can we insert this record in the Enrolments table?
Inserting data

- This won’t work; StudentID’s value (‘5’) doesn’t exist in the primary key ID
Summary

• A database is used to store information in a systematic and orderly manner
• The relational model uses tables to store information about entities and relationships to connect tables together
• Relationships require tables, primary keys, foreign keys. Referential integrity is an important concept
• Microsoft Access is a popular DBMS that we can use to insert and manage data in our database