CSCI-UA:0060-02
Database Design &
Web Implementation
Professor Evan Sandhaus
sandhaus@cs.nyu.edu
evan@nytimes.com
Lecture #5: Data Modeling
Administrivia

- **Readings:** NEW SCHEDULE
  - Today: Churcher 1-2
  - Thursday: Churcher 3-4
  - Next Tuesday: Churcher: 7-9

- **Homework**
  - Due by the start of class
  - Keep the questions coming
On Today’s Menu

- Meet the relational database
  - Relational Tables
  - Column Types
  - Keys
- Example 1: Facebook Simplified
- More relational items
  - Foreign Key
  - Composite Key
  - Id Column
- Example 2: Simple CMS
- Example 3: Normalizing Spotify
Relational databases consist of Tables which are collections where the data records are stored as rows and the values consist of fields that are defined by columns.

<table>
<thead>
<tr>
<th>name</th>
<th>latitude</th>
<th>longitude</th>
<th>population</th>
<th>country_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>35</td>
<td>105</td>
<td>1330044000</td>
<td>CN</td>
</tr>
<tr>
<td>India</td>
<td>20</td>
<td>77</td>
<td>1147995000</td>
<td>IN</td>
</tr>
<tr>
<td>USA</td>
<td>39.76</td>
<td>-98.5</td>
<td>303824000</td>
<td>US</td>
</tr>
<tr>
<td>Indonesia</td>
<td>-5</td>
<td>120</td>
<td>237512000</td>
<td>ID</td>
</tr>
<tr>
<td>Brazil</td>
<td>-10</td>
<td>-55</td>
<td>191908000</td>
<td>BR</td>
</tr>
<tr>
<td>Pakistan</td>
<td>30</td>
<td>70</td>
<td>167762000</td>
<td>PK</td>
</tr>
<tr>
<td>Bangladesh</td>
<td>24</td>
<td>90</td>
<td>153546000</td>
<td>BD</td>
</tr>
<tr>
<td>Russia</td>
<td>60</td>
<td>100</td>
<td>140702000</td>
<td>RU</td>
</tr>
<tr>
<td>Nigeria</td>
<td>10</td>
<td>8</td>
<td>138283000</td>
<td>NG</td>
</tr>
</tbody>
</table>
Relational databases consist of Tables which are collections where the **data records are stored as rows** and the values consist of fields that are defined by columns.

<table>
<thead>
<tr>
<th>name</th>
<th>latitude</th>
<th>longitude</th>
<th>population</th>
<th>country_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>35</td>
<td>105</td>
<td>1330044000</td>
<td>CN</td>
</tr>
<tr>
<td>India</td>
<td>20</td>
<td>77</td>
<td><strong>1147995000</strong></td>
<td>IN</td>
</tr>
<tr>
<td>USA</td>
<td>39.76</td>
<td>-98.5</td>
<td>303824000</td>
<td>US</td>
</tr>
<tr>
<td>Indonesia</td>
<td>-5</td>
<td>120</td>
<td>237512000</td>
<td>ID</td>
</tr>
<tr>
<td>Brazil</td>
<td>-10</td>
<td>-55</td>
<td>191908000</td>
<td>BR</td>
</tr>
<tr>
<td>Pakistan</td>
<td>30</td>
<td>70</td>
<td>167762000</td>
<td>PK</td>
</tr>
<tr>
<td>Bangladesh</td>
<td>24</td>
<td>90</td>
<td>153546000</td>
<td>BD</td>
</tr>
<tr>
<td>Russia</td>
<td>60</td>
<td>100</td>
<td>140702000</td>
<td>RU</td>
</tr>
<tr>
<td>Nigeria</td>
<td>10</td>
<td>8</td>
<td>138283000</td>
<td>NG</td>
</tr>
</tbody>
</table>
Tables

Relational databases consist of Tables which are collections where the data records are stored as rows and the values consist of **fields that are defined by columns**.

<table>
<thead>
<tr>
<th>name</th>
<th>latitude</th>
<th>longitude</th>
<th>population</th>
<th>country_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>35</td>
<td>105</td>
<td>1330044000</td>
<td>CN</td>
</tr>
<tr>
<td>India</td>
<td>20</td>
<td>77</td>
<td>1147995000</td>
<td>IN</td>
</tr>
<tr>
<td>USA</td>
<td>39.76</td>
<td>-98.5</td>
<td>303824000</td>
<td>US</td>
</tr>
<tr>
<td>Indonesia</td>
<td>-5</td>
<td>120</td>
<td>237512000</td>
<td>ID</td>
</tr>
<tr>
<td>Brazil</td>
<td>-10</td>
<td>-55</td>
<td>191908000</td>
<td>BR</td>
</tr>
<tr>
<td>Pakistan</td>
<td>30</td>
<td>70</td>
<td>167762000</td>
<td>PK</td>
</tr>
<tr>
<td>Bangladesh</td>
<td>24</td>
<td>90</td>
<td>153546000</td>
<td>BD</td>
</tr>
<tr>
<td>Russia</td>
<td>60</td>
<td>100</td>
<td>140702000</td>
<td>RU</td>
</tr>
<tr>
<td>Nigeria</td>
<td>10</td>
<td>8</td>
<td>138283000</td>
<td>NG</td>
</tr>
</tbody>
</table>
In a relational database columns may store many different kinds of data

- Integer
- Floating Point
- Boolean
- Character
- String
- Binary
- Date
- Time
- Datetime
Unique Keys
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.
- Real world
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.

- Real world
  - Zip Codes
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.

- Real world
  - Zip Codes
  - Area Codes
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.

- Real world
  - Zip Codes
  - Area Codes
  - Social Security Numbers
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.

- Real world
  - Zip Codes
  - Area Codes
  - Social Security Numbers
  - UPC
Unique Keys

- Every table in a relational database should contain a field or combination of fields that uniquely identify each record in that database.

- Real world
  - Zip Codes
  - Area Codes
  - Social Security Numbers
  - UPC
  - ISBN
Candidate Keys

- For a given table, all possible unique keys constitute the set of candidate keys.

<table>
<thead>
<tr>
<th>Row</th>
<th>Abbreviation</th>
<th>Name</th>
<th>Fips</th>
<th>Country</th>
<th>Timezone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AL</td>
<td>Alabama</td>
<td>1779775</td>
<td>USA</td>
<td>UTC-6</td>
</tr>
<tr>
<td>2</td>
<td>AK</td>
<td>Alaska</td>
<td>1785533</td>
<td>USA</td>
<td>UTC-9</td>
</tr>
<tr>
<td>3</td>
<td>AZ</td>
<td>Arizona</td>
<td>1779777</td>
<td>USA</td>
<td>UTC-7</td>
</tr>
<tr>
<td>4</td>
<td>AR</td>
<td>Arkansas</td>
<td>68085</td>
<td>USA</td>
<td>UTC-6</td>
</tr>
<tr>
<td>5</td>
<td>CA</td>
<td>California</td>
<td>1779778</td>
<td>USA</td>
<td>UTC-8</td>
</tr>
<tr>
<td>6</td>
<td>CO</td>
<td>Colorado</td>
<td>1779779</td>
<td>USA</td>
<td>UTC-8</td>
</tr>
<tr>
<td>7</td>
<td>CT</td>
<td>Connecticut</td>
<td>1779780</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
<tr>
<td>8</td>
<td>DE</td>
<td>Delaware</td>
<td>1779781</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
<tr>
<td>9</td>
<td>FL</td>
<td>Florida</td>
<td>294478</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
</tbody>
</table>
Primary Key - THIS IS IMPORTANT

For every table we select one Primary Key from the set of candidate keys to relate this table to other tables in the database.

<table>
<thead>
<tr>
<th>Row</th>
<th>Abbreviation</th>
<th>Name</th>
<th>Fips</th>
<th>Country</th>
<th>Timezone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AL</td>
<td>Alabama</td>
<td>1779775</td>
<td>USA</td>
<td>UTC-6</td>
</tr>
<tr>
<td>2</td>
<td>AK</td>
<td>Alaska</td>
<td>1785533</td>
<td>USA</td>
<td>UTC-9</td>
</tr>
<tr>
<td>3</td>
<td>AZ</td>
<td>Arizona</td>
<td>1779777</td>
<td>USA</td>
<td>UTC-7</td>
</tr>
<tr>
<td>4</td>
<td>AR</td>
<td>Arkansas</td>
<td>68085</td>
<td>USA</td>
<td>UTC-6</td>
</tr>
<tr>
<td>5</td>
<td>CA</td>
<td>California</td>
<td>1779778</td>
<td>USA</td>
<td>UTC-8</td>
</tr>
<tr>
<td>6</td>
<td>CO</td>
<td>Colorado</td>
<td>1779779</td>
<td>USA</td>
<td>UTC-8</td>
</tr>
<tr>
<td>7</td>
<td>CT</td>
<td>Connecticut</td>
<td>1779780</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
<tr>
<td>8</td>
<td>DE</td>
<td>Delaware</td>
<td>1779781</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
<tr>
<td>9</td>
<td>FL</td>
<td>Florida</td>
<td>294478</td>
<td>USA</td>
<td>UTC-5</td>
</tr>
</tbody>
</table>
Simple Example

Like
Is It Possible to Defend Comic Sans?

It's nearly impossible to use Comic Sans on the Internet and not get tarred and feathered. It's an Internet sin of the highest level. A crime against human decency and people's eyeballs. A parody of a joke of a fool. Universally hated. So... is it possible to defend the font? Is Comic Sans wrongfully reviled? Maybe!
Is It Possible to Defend Comic Sans?

It's nearly impossible to use Comic Sans on the Internet and not get tarred and feathered. It's an Internet sin of the highest level. A crime against human decency and people's eyeballs. A parody of a joke of a fool. Universally hated. So... is it possible to defend the font? Is Comic Sans wrongfully reviled? Maybe!
Is It Possible to Defend Comic Sans?

It's nearly impossible to use Comic Sans on the Internet and not get tarred and feathered. It's an Internet sin of the highest level. A crim...See More

Evan likes Is It Possible to Defend... on Gizmodo.

Like · Comment · 21 seconds ago 📣
Use Cases
Use Cases

- Use cases are simple statements that express the functionality we expect the database to implement.
Use Cases

- Use cases are simple statements that express the functionality we expect the database to implement.
- We want to build a database that:
  1) Captures all of the unique likes for Facebook users.
  2) Captures the home State for Facebook users.
user

name

home_state

page

url

title

tumbnail_uri

states

name
**Foreign Key**

- A foreign key is a column in one table that may only contain values that correspond to a candidate key of another table.

![Database diagram showing a relationship between states and user tables with a foreign key](image)
Cardinality
Cardinality

- Relations between tables come in a variety of forms
Cardinality

- Relations between tables come in a variety of forms
- 1-1:
  - Users having a primary profile picture.
Cardinality

- Relations between tables come in a variety of forms
  - 1-1:
    - Users having a primary profile picture.
  - 1-n
    - Users having multiple profile pictures.
    - States being the home state of multiple users.
Cardinality

- Relations between tables come in a variety of forms
- 1-1:
  - Users having a primary profile picture.
- 1-n
  - Users having multiple profile pictures.
  - States being the home state of multiple users.
- m-n
  - Users liking pages
Optionality
Optionality

- We can also need to concern ourselves with optionality, that is whether or not a relationship is required.
Optionality

- We can also need to concern ourselves with optionality, that is whether or not a relationship is required.

- A user **must have** one and only one a primary email address
  - 1 - 1...1
Optionality

- We can also need to concern ourselves with optionality, that is whether or not a relationship is required.

- A user **must have** one and only one a primary email address
  - 1 - 1...1

- A user **may have** one and only one primary profile picture
  - 1 - 0...1
Optionality

- We can also need to concern ourselves with optionality, that is weather or not a relationship is required.
- A user **must have** one and only one a primary email address
  - \( 1 - 1 \ldots 1 \)
- A user **may have** one and only one primary profile picture
  - \( 1 - 0 \ldots 1 \)
- A user may have children
  - \( 1 - 0 \ldots n \)
Optionality

- We can also need to concern ourselves with optionality, that is whether or not a relationship is required.

- A user **must have** one and only one a primary email address
  - 1 - 1...1

- A user **may have** one and only one primary profile picture
  - 1 - 0...1

- A user may have children
  - 1 - 0...n

- Users may have friends
Cardinality and Optionality

```
  states
     state_name
```

```
  user
     name
     state_name
```

Relationship: 1..1 from states to user
Cardinality and Optionality

states
  state_name

user
  name
  state_name

page
  url
  title
  thumbnail_uri

From 0..n

0..n

likes
Reference Tables

- **Reference Tables** store a vocabulary that is used to constrain the column values of another table or tables.
- Vocabulary is called a **Controlled Vocabulary**
Composite Keys

- Some tables have no single column that makes a suitable key and require that we craft a key from multiple columns.

<table>
<thead>
<tr>
<th>AccountNumber</th>
<th>AccountType</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>123-45-6789</td>
<td>Personal</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>123-45-6790</td>
<td>Business</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>123-45-6791</td>
<td>Personal</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>123-45-6792</td>
<td>Business</td>
<td>$1,234.56</td>
</tr>
<tr>
<td>123-45-6790</td>
<td>Personal</td>
<td>$1,234.56</td>
</tr>
</tbody>
</table>
**ID Column**

- Oftentimes we introduce an ‘id’ column when a data table has no ‘natural’ primary key (e.g. person names).

<table>
<thead>
<tr>
<th>id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Clark Kent</td>
</tr>
<tr>
<td>2</td>
<td>Bruce Wayne</td>
</tr>
<tr>
<td>3</td>
<td>John Smith</td>
</tr>
<tr>
<td>4</td>
<td>Peter Parker</td>
</tr>
<tr>
<td>5</td>
<td>Bruce Banner</td>
</tr>
<tr>
<td>6</td>
<td>John Smith</td>
</tr>
</tbody>
</table>
Use Cases

- Display an article including headline, byline, text, section, and comments, images
- Aggregate articles by author
- Aggregate articles by section
- Show all comments on an article
- Show all comments for a user