Prevent Inconsistent and Conflicting Data

Relational Database Design

Session Outline

- Why relational databases are used to prevent inconsistent and conflicting data

Data Types

- Fields in tables have data types to specify what kind of data is stored in them
- Text, number, date, yes/no
- All fields that store a similar value can be set to the same data type

Naming Consistency

- Similar fields in different tables can also be named in a similar way
  - Teachers and students both have birth days
  - Teacher: birth_date
  - Student: birth_date
  - Similar information for two different tables
  - Same name
- Names should be same format throughout the databases
- Easily done with relational databases
  - Separating of words: underscore or capital letter? (birth_date or birthDate)
  - Other considerations depending on what you're storing

Date Example

- Dates are often used in databases:
  - Birth date
  - Enrolment date
  - Updated date
  - Added date
- If all of these fields are set to the same type and format, it makes the database more efficient and easier to work with
- Store all of these as a “date” data type, with similar format
  - YYYY-MM-DD
  - MM-DD-YYYY
  - However you like – as long as it’s consistent

Calculated Fields

- A common problem with databases is storing calculated fields
- Fields can contain values which are used to perform calculations to determine other values
- For example, to find someone's age:
  - Store their birth date
  - Calculate the number of years between their birth date and today
- Should we store the "age" value in the database?

Calculated Fields

- No, we shouldn't store "age"
- It is a calculated field which can easily be recalculated by a system
- You can get data inconsistencies
- Many issues with storing calculated data
  - What happens next year?
  - Do we recalculate the "age" for each year?
  - What happens if there is a lot of records?
  - Do we run the calculation every day for all records and update the table?
  - What if there is a problem updating this value?
- It's better to get the source values and perform the calculation as needed

Format Validation

- Another advantage is using format validation
- You can specify what formats the data needs to be in to be stored
- One example is date fields
- Dates are treated differently around the world
- You can specify a format in your design, such as:
  - YYYY-MM-DD
  - MM-DD-YYYY
  - YYYY (year only)
- All applications that use the database must stick to this format
- Keeps data consistent and efficient

Summary

- Using consistent data types is an advantage of a relational database
- Calculated fields should not be stored, they should be recalculated as needed
- Formats of data can be restricted to ensure data integrity

What's Next?

- A further look at some of the advantages of relational databases