



Data Types and Precision

Relational Database Design

Session Outline

- Learn how to use data types and precision to make our database better designed

Data Types

- Each field needs to have a data type
- Data types such as text, number, date, yes/no
- This should be the same no matter what database management system (DBMS) you're using
- Fields need to also have a database-specific data type
- One that works for MySQL or Oracle or MS SQL or whatever DBMS you're using

Data Types

- Many data types also have additional information
- Size or length (numeric digits, or characters in a text field)
- Precision (decimal places)

Considerations

- Choose data types wisely
- It's hard to change them later
- They need to be efficient
- They should be long enough to handle the largest scenario possible, but not too large

Our Example

- For example, name fields
 - First and last names could be 100 characters long
 - Subject names could be 200 characters long
- Text limits may be 1000 characters
- No need to make all text fields 1000 characters long if it's not needed
- It wastes space, and wastes time in the database that implements it

Number Fields

- Numbers should be the appropriate length to handle all possible combinations and restrict bad data
- For example, customer numbers or other primary keys may be 10, or 12, or 15 digits long
- Dollar values may be decimal numbers:
 - 10 digits to the left
 - 2 to the right
 - Could store values such as 1,234,567,890.12
- Long enough to store all possible values, but not excessive

Consistency

- Keep data types and lengths consistent for similar fields
- For example, all first names should be the same length and type
- Same for last names
- This makes it easier to use and implement

MySQL Workbench

- Let's check our data types and consistency in MySQL Workbench

Summary

- Fields need to have data types assigned to them
- Data types should be long enough to handle large scenarios, but not too long that they are excessive
- Data types should be consistent between tables

Action

1. Check each of the columns in your tables to ensure that the data types are set appropriately
2. Check the size or length of each column to ensure that they are appropriate
3. Ensure that all columns that contain similar data are the same data type and length (e.g. all first names in the database)

What's Next?

- Integrity constraints and why they can be used