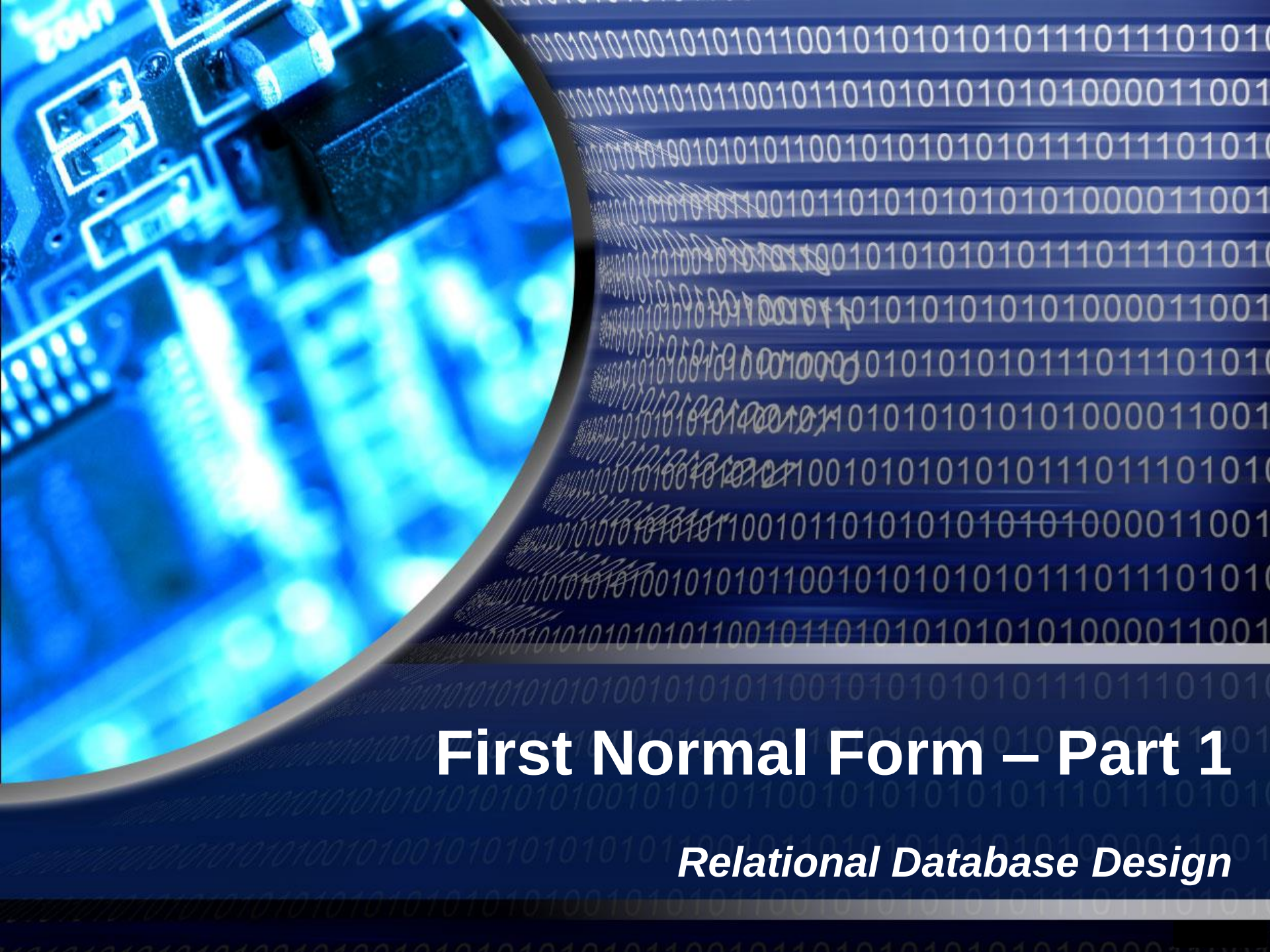

First Normal Form – Part 1

Relational Database Design

Session Outline

- Learn what first normal form is
- Learn about primary and foreign keys
- Apply first normal form to our database

First Normal Form

- What is “first normal form”?
- It’s the first stage of the normalisation process

Each set of columns must uniquely identify a row

- A combination of all of the columns must be unique and only define a single row
- If there is more than one row, it’s not unique

Primary Key

- What is a “primary key”?
- A primary key is a field (or column) in a table that contains a value that is unique to the table
- The value can only be stored once in that column
- Used to uniquely identify a record in a database
- This should be used to identify our entities
- Customer numbers and student IDs are common examples of primary keys
- Names are not, they are not always unique

Applying First Normal Form

- It's now time to apply "first normal form" to our database
- We can ask ourselves some questions of each table:

1. Does the combination of all columns make a unique row *every single time*?
2. What *field* can be used to uniquely identify the row?

Our Example

- Remember our example so far:
- Student
  - First name (text)
  - Last name (text)
  - Date of birth (date)
  - Address (text)
- Subject
  - Subject name (text)
  - Subject category (text)
  - Student name (text)
- Teacher
  - First name (text)
  - Last name (text)
  - Date of birth (date)
  - Address (text)
  - Subject taught (text)
- University
  - University name (text)
  - University address (text)

Student Table

- Student: First name, last name, date of birth, address
- Does this combination uniquely identify the row *every time*?
- No
  - There could be someone of the same name and date of birth at the same address
  - Chances are very rare!
  - But still possible

Student Table

- Student: First name, last name, date of birth, address
- What can be used to uniquely identify a row?
- None of these fields individually
- There could be multiple fields, but not in this case
- Not even the combination of all columns

Student Table

- Student: First name, last name, date of birth, address
- We need to create a new field for the primary key
- It can be called whatever you like
- It should be **obvious** and **consistent**
- Something like student number or student ID
- Let’s use ID in this case: **student ID**

Student Table

- Also, let’s divide up our address column
- An address contains a lot of other information, and is a name for a group of fields
- You might have seen the address split up on websites
- Let’s keep it simple for this example
- Address fields: unit number, street number, street name, suburb, city, state, code, country

Student Table Updated

- Our new Student table:
- Student: Student ID, first name, last name, date of birth, unit number, street number, street name, suburb, city, state, code, country
- Underlined the student ID
- This is common notation for a primary key
- Primary key usually goes first in the field list

MySQL Workbench

- Let’s update the **student** table in our MySQL Workbench file

What's Next?

- Continue with part 2 of First Normal Form