Second Normal Form – Part 1

Relational Database Design

Session Outline

- What second normal form is
- Apply it to our sample database

Second Normal Form

- What is second normal form?
- Second stage of the normalisation process

Fulfil the requirements of first normal form

Each non-key attribute must be functionally dependent on the primary key

- What does this even mean?

Second Normal Form

- Each non-key attribute must be functionally dependent on the primary key
- A non-key attribute is one that is **not the primary key**
- Functionally dependent means that the attribute is determined by the primary key
- It’s *specific* to that record

Foreign Keys

- Let’s learn this before we move forward
- Foreign key is a field in a table that is a primary key in another table
- Used to link two tables together to link the unique record
- We’ll add them to our example here as part of second normal form

Our Example

- Student: Student ID, first name, last name, date of birth, unit number, street number, street name, suburb, city, state, code, country
- Subject: subject ID, subject name, subject category, student name
- Teacher: teacher ID, first name, last name, date of birth, subject taught, unit number, street number, street name, suburb, city, state, code, country
- University: university ID, university name, unit number, street number, street name, suburb, city, state, code, country

Student Table

- Student: Student ID, first name, last name, date of birth, unit number, street number, street name, suburb, city, state, code, country
- Looks OK
- All non-key attributes are specific to that student

Subject Table

- Subject: subject ID, subject name, subject category, student name
- Subject category is not dependent on the subject ID
- Category could be duplicated, and it could change
- This means there is an “update anomaly”
  - What happens if a category name changes?
  - It needs to be updated for all records
  - What if a record is missed?
  - Needs to be updated in more than one location

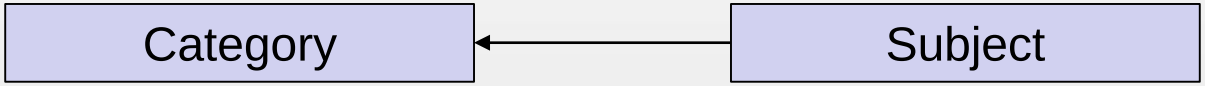
Subject and Category

- What is the solution?
- Move it into a new table, with its own ID field
- Subject: subject ID, subject name, student name
- Category: category ID, category name

- How do we link these together? How do we know what category applies to a subject?

Subject and Category

- We use a foreign key
- This is based on a primary key
- We add the primary key from one table to the other table, which is then the foreign key
- It’s used to identify the row in another table



Category

Subject

Adding a Foreign Key

- But how do we know what key to use, as there are two?
  - Student: student ID
  - Category: category ID
- Depends on the relationship between the two tables
- I ask myself a question to work this out
- “Does table1 have many table2s, or does table2 have many table1s?”
- Substitute table1 and table2 for your tables, of course

Adding a Foreign Key

- In this example, “**does a subject have many categories, or does a category have many subjects?**”
- Second statement is true – a category (e.g. science) has many subjects (e.g. intro to biology, advanced chemistry)
- This means we add the **primary key for the second table** into the **first table as a foreign key**

Subject and Category

- Subject: subject ID, *category ID,* subject name, student name
- Category: category ID, category name
- Italics represent the foreign key
- Added after the primary keys in the field order

MySQL Workbench

- Let’s see how this foreign key is shown in MySQL Workbench

What’s Next?

- Continue applying second normal form to the rest of our tables in Part 2