Logical Database Modeling, Part 2: Normalization

A tool to validate and improve a logical design so that it satisfies certain constraints that avoid unnecessary duplication of data and resulting problems with data consistency.

The process of decomposing relations with anomalies to produce smaller, well-structured relations

1. **Motivation**: avoiding anomalies.

Consider the **StudentAdvisor** relation below:

Note the following rules:

- A student has a single advisor in a given semester.
- An advisor is assigned from a department that corresponds to the student’s major.
- An advisor can have more than one advisees per semester.

<table>
<thead>
<tr>
<th>St_id</th>
<th>St_fname</th>
<th>St_lname</th>
<th>St_major</th>
<th>Semester</th>
<th>Adv_fname</th>
<th>Adv_lname</th>
<th>Adv_email</th>
<th>Adv Dept</th>
<th>Adv_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Adam</td>
<td>Gopnik</td>
<td>Literature</td>
<td>F’02</td>
<td>William</td>
<td>Shakespeare</td>
<td><a href="mailto:ws@lu.ac.uk">ws@lu.ac.uk</a></td>
<td>Lit</td>
<td>352</td>
</tr>
<tr>
<td>035</td>
<td>Sandra</td>
<td>Smith</td>
<td>Literature</td>
<td>F’02</td>
<td>James</td>
<td>Joyce</td>
<td><a href="mailto:jj@dbl.edu">jj@dbl.edu</a></td>
<td>Lit</td>
<td>465</td>
</tr>
<tr>
<td>735</td>
<td>Barbara</td>
<td>Richards</td>
<td>Astronomy</td>
<td>S’03</td>
<td>Galileo</td>
<td>Galilei</td>
<td><a href="mailto:gg@pdv.it">gg@pdv.it</a></td>
<td>Phys</td>
<td>773</td>
</tr>
<tr>
<td>136</td>
<td>Preston</td>
<td>Jones</td>
<td>Astronomy</td>
<td>F’02</td>
<td>Galileo</td>
<td>Galilei</td>
<td><a href="mailto:gg@pdv.it">gg@pdv.it</a></td>
<td>Phys</td>
<td>773</td>
</tr>
<tr>
<td>035</td>
<td>Sandra</td>
<td>Smith</td>
<td>Literature</td>
<td>S’03</td>
<td>William</td>
<td>Shakespeare</td>
<td><a href="mailto:ws@lu.ac.uk">ws@lu.ac.uk</a></td>
<td>Lit</td>
<td>352</td>
</tr>
</tbody>
</table>

What are the candidate keys of this relation?

- **Insertion anomaly**: new data cannot be added without a violation of the constraint on a non-NULL values of all primary key attributes.
- **Deletion anomaly**: deleting rows may cause a loss of data that is needed.
- **Modification (update) anomaly**: changing data in a row forces changes to other rows because of duplication.

**General rule of thumb**: a table should not pertain to more than one entity type.

2. **Functional Dependency** - the value of one attribute (the determinant) determines the value of another attribute.

C → D means C determines D,
A,B → C means combination of A and B determines C.
In other words, for any two rows that have same values of the determinant (C), the dependent value (D) is guaranteed to be the same.

C is a determinant of D  
A and B together are determinants of C

**Practice problem** – identify and depict all functional dependencies in the StudentAdvisor table.

Normalization oath:

No Repeating,
All fields depend upon the key,
The entire key,
And nothing but the key,
So help me Codd.

3. **First normal form (1NF)** – no multivalued attributes.
   All relational tables are in the first normal form by definition.
   **Normalization**: 1NF is achieved by forming a separate relation for each multivalued attribute

4. **Second normal form (2NF)**

   Relation must be in 1NF plus every non-key attribute is FULLY functionally dependent on the ENTIRE primary key, i.e.
   - In other words, every non-key attribute must be defined by the entire key, not by only part of the key. There are no partial functional dependencies

   **Normalization**: To achieve 2NF - Decompose and set up a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.

   **Practice problem**: transform *StudentAdvisor* relation into a set of tables in the second normal form.
5. **Third normal form** -
   relation must be in 2NF *plus no transitive dependencies* (one attribute functionally determines a second, which functionally determines a third).

   **Normalization:** To achieve 3NF - decompose and set up a new relation that includes the nonkey attributes that are determinants for other nonkey attributes.

   **Practice problem:** Identify transitive dependencies in the following relation and normalize to the 3NF:

   $\text{StHousing}(\text{StudentID, Dorm, Fee})$, where each student is assigned to a single dorm, and the fee paid by the student depends on the dorm.

6. **Other normal forms.** Boyce-Codd NF, Fourth, Fifth, Domain/Key

   **Practice Problems:**
   1. Problem 6, page 204
   2. Identify functional dependencies, partial and transitive dependencies, and transform the following schema into a set of tables that is in the Third Normal Form:

   **Doctor** (doctor#, doctor_name, office phone, patient, appointment time, doctor’s specialties)

   **Patient** (patient-SSN, patient name, address, primary insurance, secondary insurance, appointment time, doctor)

   **Appointment** (patient name, doctor_name, appointment date, appointment time, exam room)

   **Billing** (patient-SSN, date of appointment, list of procedures, list of costs of those procedures, primary insurance company’s name, primary insurance address, % paid by insurance company)