IS5102
Database Management Systems
Lecture 14: Normalisation

Alexander Konovalov
alexander.konovalov@st-andrews.ac.uk

(with thanks to Susmit Sarkar)

2021

This week

- Features of Good Relational Design
- Update Anomalies
- Functional Dependencies
- Normalisation
- Problems of Normalisation

Process of Normalisation

- Formal technique for **analysing** a relation based on its primary key and the functional dependencies between the attributes of that relation.
- Often executed as a series of steps. Each step corresponds to a specific normal form, which has known properties.
- As normalisation proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.

Unnormalised form (UNF)

A table that may contain one or more repeating groups.

- For example, created when someone transformed the data from an information source (e.g. from a form used to collect the data) into table format with columns and rows
- Starting point for analysis

UNF data

Proj Code	Proj Title	Proj Manager	Proj Budget	Emp No	Emp Name	Dept No	Dept Name	Hours
1	Starship	Johnston	120000	12	Spratt	2	IT	15
				15	Smith	3	HR	10
				22	Brown	2	IT	44
2	Enterprise	Williamson	250000	23	Black	3	HR	10
				12	Spratt	2	IT	48
				22	Brown	2	IT	16

Useful reading:

Karl W. Broman & Kara H. Woo (2018), **Data Organization in Spreadsheets**, The American Statistician, 72:1, 2-10, https://doi.org/10.1080/00031305.2017.1375989

Data Organization in Spreadsheets for Ecologists. Data Carpentry lesson. https://datacarpentry.org/spreadsheet-ecology-lesson/

First Normal Form (1NF)

A relation in which the intersection of each row and column contains one and only one value.

- Nominate an attribute or group of attributes to act as the key for the unnormalised table.
- Identify the repeating group(s) in the unnormalised table which repeats for the key attribute(s).
- Remove the repeating group by
  - Entering appropriate data into the empty columns of rows containing the repeating data ('flattening' the table).

    Or by
  - Placing the repeating data along with a copy of the original key attribute(s) into a separate relation.

- Based on the concept of full functional dependency.
- Recall: Full functional dependency indicates that if A and B are attributes of a relation, B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.

Second normal form (2NF)

A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key.

1NF to 2NF

- Identify the primary key for the 1NF relation.
- Identify the functional dependencies in the relation.
- If partial dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their determinant.

- Based on the concept of transitive dependency.
- Recall: Transitive Dependency is a condition where A, B and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A through B.

Third Normal Form (3NF)

A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.

- Identify the primary key in the 2NF relation.
- Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their determinant.

Second normal form (2NF)

A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on *any candidate key*.

Third normal form (3NF)

A relation that is in first and second normal form and in which no non-primary-key attribute is transitively dependent on *any candidate key*.

MatricNo	Name	ModID	Module	EnrolDate	Credits	Grade
1200001	Brown	IS5102	DBMS	02/09/20	15	13
1200001	Brown	CS4160	Operating Systems	02/09/20	15	15
1200002	Sharma	IS5102	DBMS	02/09/20	15	13
1200002	Sharma	CS4160	Operating Systems	02/09/20	15	15
1200003	Liu	CS5111	System Analysis	05/09/20	15	12
1200002	Sharma	CS5200	Project	05/09/20	60	14
1200004	Lopez	CS4160	Operating Systems	05/09/20	15	15

The table is subject to update anomalies.

For example, suppose we change the name of student with matriculation number 1200001 from Brown to Black. We have to change it identically in two rows, otherwise there is an update anomaly.

If we delete the row of CS5200, we lose all information about that module in a delete anomaly.

1. Identify minimal set of functional dependencies:
MatricNo $\rightarrow$ Name
ModID $\rightarrow$ Module, Credits
MatricNo, ModID $\rightarrow$ EnrolDate, Grade
2. Identify primary key: (MatricNo, ModID)
3. The table is already in 1NF.
4. Remove partial functional dependencies of name, module and credits on the primary key (MatricNo, ModID) by splitting into tables (MatricNo, Name), (ModID, Module, Credits), and (MatricNo, ModID, EnrolDate, Grade). This makes the three tables in 2NF.
5. These three tables are already in 3NF, so we stop here.
Note that it does not satisfy the general definition of 3NF (no transitive dependencies on any candidate key) for the (ModID, Module, Credits) relation as Module is a candidate key as well. So we can split that relation into (ModID, Module) and (ModID, Credits) to get 3NF in the general sense.

Normalisation: Not a Cure-All

Normalisation is **definitely** not a cure-all solution

Bad database design can occur nevertheless

Normalisation: Inefficiencies?

- Normalisation typically leads to **more** (smaller) tables
- To retrieve information needs **joins**
- Joins: expensive operations

Denormalisation

- Very common queries might suggest denormalised tables
- Leads to faster lookups
- Leads to slower updates
- Leads to extra storage space
- Leads to possibility of programmer error
- Denormalise **after** normalising (and careful consideration)

Some Examples:

- Materialised views (handled by database)
- Precalculated summaries (rollup, cube) (Data Analytics)
- Star schemas (Data Warehouses)

Higher Normal Forms

- Normalisation in general: use constraints
- Create relational structure that will not violate constraints on update
- Several kinds of constraints not considered in this module
- Leads to higher normal forms: 4NF, 5NF, ...

Data Gathering and Normalisation

- In legacy (non DBMS) systems, common to have heirarchical data
- Often easiest to translate directly to UNF
- Can apply normalisation to create nice normalised relations
- Can go back and validate if relations make semantic sense

Chapters 14, 15 – Database Systems, Connolly and Begg

Chapter 8 – Database System Concepts, 6th Ed. Silberschatz, Korth and Sudarshan

Chapter 11 – Database Design, Watt and Eng