IS5102
Database Management Systems
Lecture 13: Relational Database Design

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

ID	Name	Department	Salary	Building	Head
10210	Einstein	Physics	56000	Allen	Curie
45893	Franklin	Biology	48250	Purdie	Watson
31822	Curie	Physics	67000	Allen	Curie
33821	Brandt	Computer Science	81000	Cole	Knuth
45910	Tao	Psychology	51900	Bute	Piaget
28675	Turing	Computer Science	49750	Cole	Knuth
67450	Crick	Biology	64500	Purdie	Watson
57264	Porter	Management	92000	Gateway	Porter
10101	Wu	Finance	78500	Gateway	Black
15675	Pavlov	Psychology	62800	Bute	Piaget

Data Redundancy and Update Anomalies

- Schema on previous slide has **redundant** data; the details of a department are repeated for every instructor.
- Relations that contain redundant information may potentially suffer from update anomalies.
- Types of update anomalies include
  - Insertion
  - Deletion
  - Modification

Normalisation

Normalisation is a technique for producing a set of suitable relations that support the data requirements in a way that minimises redundancy and improves integrity of a database.

Functional dependency describes relationship between attributes.

Functional Dependency

If A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.

$A \rightarrow B$ is read as "B functionally dependent on A" or "A functionally determines B"

Functional Dependencies

- Property of the meaning or semantics of the attributes in a relation.
- Can be represented diagrammatically.
- The **determinant** of a functional dependency refers to the attribute or group of attributes on the left-hand side of the arrow.

Example: Functional Dependency

ID	Name	Department	Salary	Building	Head
10210	Einstein	Physics	56000	Allen	Curie
45893	Franklin	Biology	48250	Purdie	Watson
31822	Curie	Physics	67000	Allen	Curie
33821	Brandt	Computer Science	81000	Cole	Knuth
45910	Tao	Psychology	51900	Bute	Piaget
28675	Turing	Computer Science	49750	Cole	Knuth
67450	Crick	Biology	64500	Purdie	Watson
57264	Porter	Management	92000	Gateway	Porter
10101	Wu	Finance	78500	Gateway	Black
15675	Pavlov	Psychology	62800	Bute	Piaget

ID functionally determines Name $\quad$ ID $\rightarrow$ Name

ID functionally determines Department $\quad$ ID $\rightarrow$ Department

Name functionally determines Department $\quad$ Name $\rightarrow$ Department

But: Department does not functionally determine Name $\quad$ Department $\not\rightarrow$ Name

Example: Functional Dependency that holds for all time

Consider the values shown in ID and name attributes of the schema on the previous slide.

Based on sample data, the following functional dependencies appear to hold.

$ID \rightarrow name$
$name \rightarrow ID$

However, the only functional dependency that remains true for all possible values for the ID and name attributes of the instructor relation is:

$ID \rightarrow name$

Determinants should have the minimal number of attributes necessary to maintain the functional dependency with the attribute(s) on the right hand-side.

This requirement is called **full functional dependency**.

Full Functional Dependency

Let A and B are attributes of a relation. B is **fully functionally dependent** on A, if B is functionally dependent on A, but not on any proper subset of A.

Example: Partial Functional Dependency

ID, name $\rightarrow$ dept_name

This is indeed a functional dependency:
each value of (ID, name) is associated with a single value of dept_name.

However, dept_name is also functionally dependent on a subset of (ID, name), namely ID.

Example above is a **partial dependency**.

Important to recognise a **transitive dependency** because its existence in a relation can potentially cause update anomalies.

Transitive Dependency

Let A, B, and C are attributes of a relation. If $A \rightarrow B$ and $B \rightarrow C$ then C is said to be **transitively dependent** on A via B.

Example Transitive Dependency

Consider functional dependencies in the instructor_and_department relation

`instructor_and_department(name, salary, department, building, head)`

ID $\rightarrow$ name, salary, department, building, head

department $\rightarrow$ building, head

Transitive dependencies (both via department):

ID $\rightarrow$ building

ID $\rightarrow$ head

Identifying Functional Dependencies

- Identifying all functional dependencies between a set of attributes is relatively simple if the meaning of each attribute and the relationships between the attributes are well understood.
- This information should be obtained via discussions with users and/or documentation such as the user requirements specification.
- However, if the users are unavailable for consultation and/or the documentation is incomplete then depending on the database application it may be necessary for the database designer to use their common sense and/or experience to provide the missing information.
- Otherwise look at sample. Important to establish that sample data values shown in relation are representative of all possible values that can be held by attributes.

Identifying the Primary Key for a Relation using Functional Dependencies

- Main purpose of identifying a set of functional dependencies for a relation is to specify the set of integrity constraints that must hold on a relation.
- An important integrity constraint to consider first is the identification of candidate keys, one of which is selected to be the primary key for the relation.

Primary Key Identification Example

ID $\rightarrow$ name, salary, department, building, head

department $\rightarrow$ building, head

- The **determinants** are ID, department
- Task: Identify all candidate key(s), identify the attribute (or group of attributes) that uniquely identifies each tuple in this relation.

ID $\rightarrow$ name, salary, department, building, head

department $\rightarrow$ building, head

- The **determinants** are ID, department
- Task: Identify all candidate key(s), identify the attribute (or group of attributes) that uniquely identifies each tuple in this relation.

- **All** attributes that are not part of a candidate key should be functionally dependent on the key.
- The only candidate key and therefore primary key for instructor_and_department relation, is ID as all other attributes of the relation are functionally dependent on ID.

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.

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