COS221
L20 - Relational Model - Functional Dependencies
(Chapter 15 in Edition 6 and Chapter 14 in Edition 7)

Linda Marshall

21 April 2022

Database design process

Bottom-up vs Top-down

- A **bottom-up approach** uses relationships between attributes as a starting point for creating relational schemas. This approach requires collecting a large number of binary relationships between attributes and is therefore not very popular.
- The **top-down approach**, begins by using the natural grouping of attributes into relations. These relations are analysed and decomposed until all requirements are met.

Functional dependencies and normalisation apply to both approaches. However, it is more applicable to top-down.

Overall goal

The overall goals of relational database design is:

- *information preservation* - preserve all attribute types, entity types, relationship types described in a model such as an EER model.
  The relational design must preserve all concepts captured in the conceptual design after the conceptual design to logical design mapping.
- *minimum redundancy* - means minimising the redundant storage and reduce the need for multiple updates to maintain consistency of the data.

Informal design guidelines for relational schemas

Measures to ensure quality of relation schema design:

- clear semantics of attributes in relations
- reduce redundant information in tuples
- reduce NULL values in tuples
- do not allow the generation of spurious tuples

Informal design guidelines for relational schemas - Clear semantics of attributes in relations

The meaning resulting from the interpretation of the attribute values of the tuple.

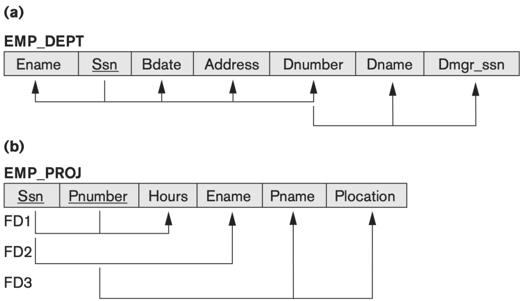
By careful design of the conceptual model and a systematic following of the mapping procedure, the resulting relational schema should have clear meaning

Informal design guidelines for relational schemas - Clear semantics of attributes in relations

Guideline 1 - The relational schema should be easily explained.

- Do not combine the attributes of multiple entity types and relationship types into a single relation.

Figure 14.3
Two relation schemas suffering from update anomalies.
(a) EMP_DEPT and
(b) EMP_PROJ.



Informal design guidelines for relational schemas - Reduce redundant information in tuples

- Reduce the storage space used by the base relation.
- How attributes are grouped into a schema has an effect on storage space.
- Storing natural joins of base relations leads to update anomalies. Remember, these are classified as insertion, deletion and modification anomalies.

Informal design guidelines for relational schemas - Reduce redundant information in tuples

- **Insertion anomalies** - 2 types:
  - insertion into a relation where one of the participating enities does not have values. For example, adding a department that has not got employees as yet.
  - insertion into a relation where the primary key is given a value of NULL.
- **Deletion anomalies** - deletion results in information going missing. For example, removal of the last employee in a department will result in the department also disappearing.
- **Modification anomalies** - If one value changes and this effects multiple tuples, the other tuples need to change as well. For example, the manager of a department changes.

Informal design guidelines for relational schemas - Reduce redundant information in tuples

Redundancy

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Redundancy Redundancy

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston

Figure 14.4
Sample states for EMP_DEPT and EMP_PROJ resulting from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

Informal design guidelines for relational schemas - Reduce redundant information in tuples

Guideline 2 - The base relation should be designed to mitigate update anomalies.

- If update anomalies cannot be mitigated, they must be clearly documented and all programs using the database must make the necessary adjustments.

Informal design guidelines for relational schemas - Reduce NULL values in tuples

- If many attributes do not apply to a tuple, NULL values will occur. This wastes storage space and may also lead to a misunderstanding of the meaning of the attributes.
- When aggregates are applied, how should NULL values be handled?
- How should joins be managed?
- Interpretations of NULL values:
  - attribute does not apply to this tuple
  - attribute value is unknown
  - attribute value is known, but absent

Informal design guidelines for relational schemas - Reduce NULL values in tuples

Guideline 3 - Avoid placing attributes in a base relation if its value may frequently be NULL.

- Only use NULL in exceptional cases.

Informal design guidelines for relational schemas - Do not allow the generation of spurious tuples

- Spurious tuples are additional tuples that represent invalid information.
- Invalid information is usually as a result of a join.

Informal design guidelines for relational schemas - Do not allow the generation of spurious tuples

Guideline 4 - Design relational schemas so that they can be joined on appropriate attributes (preferably primary key, foreign key pairs) without resulting in additional tuples of inappropriate data.

Formal theory for relational schema design

Formal introduction of tools that can be used to detect and describe the problems addressed by the guidelines in precise terms.

Functional dependency is the single most important concept in relational schema design theory.

- It deals with the property of semantics of attributes.
- Relation states that satisfy the functional dependency are legal relation states.
- It is a property of a relation R, not its states r.

Formal theory for relational schema design

Functional dependency is a constraint between two sets of attributes from the database.

Suppose a relational database schema has n attributes: $A_1, A_2, ..., A_n$. The whole database can be described by: $R(A_1, A_2, ..., A_n)$, a universal relational schema

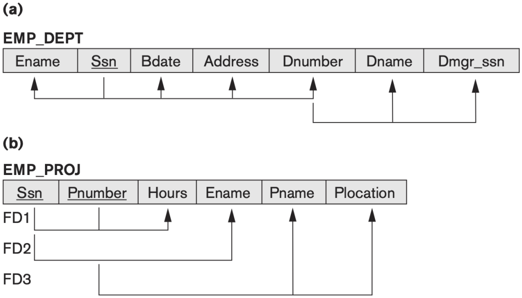
- A functional dependency is then defined as a constraint that can form a relation state r of R between two sets of attributes X and Y, denoted by $X \to Y$. That is for two tuples t_1 and t_2, that have $t_1[X] = t_2[X]$, must have $t_1[Y] = t_2[Y]$.
- FD (or f.d.) can be read as follows:
  - the values of the Y components of a tuple in r are determined by the values of the X component; or
  - the values of the X component functionally determine the values of the Y component.
- The attributes X are called the left-hand side of the FD and the attributes Y the right-hand side.
- If X is a **candidate key** of R, then $X \to R$.

Formal theory for relational schema design

Examples of FDs for Fig 14.3(b):

- Ssn $\rightarrow$ Ename, that is *Ssn uniquely determines the employee name*
- Pnumber $\rightarrow$ {Pname, Plocation}, that is the *project number uniquely determines the project name and location*
- {Ssn, Pnumber} $\rightarrow$ Hours, that is *a combination of Ssn and Pnumber uniquely determines the hours the employee worked on the project*

Figure 14.3
Two relation schemas suffering from update anomalies.
(a) EMP_DEPT and
(b) EMP_PROJ.



Normalisation forms based on Primary Keys

- Assume each relation has a *set of functional dependencies* and a *primary key*. Along with *tests (conditions) for normal forms*, the **normalisation process** is driven.
  - The normalisation process was first proposed by Codd. This process takes a relation schema through tests to determine whether it complies with certain normal forms.
  - Codd proposed 3 normal forms, 1NF, 2NF and 3NF
  - Later Boyce and Codd proposed a stronger 3NF, called BCNF
  - 1NF, 2NF, 3NF and BCNF are based on FDs of attributes of a relation.
  - Later still, based on multivalued dependencies and join dependencies, 4NF and 5NF were proposed.
- Approach to relation design:
  - Perform the conceptual design and create a model (ER or EER) and map this model to the relations.
  - Design relations based on knowledge derived from the use of the data and existing implementations
- After relations have been determined, evaluate them for *goodness*. Decompose them further as needed.

Normalisation of data

- Normalisation of data is the process of analysing a relational schema based on FDs and primary keys to achieve:
  - minimum redundancy
  - minimum insertion, deletion and update anomalies
- By following a process of normalisation, the following are achieved:
  - nonadditive (or lossless) join - spurious tuples are not generated
  - dependency preservation - each FD is represented in an individual relation schema
- By applying normal form tests, relations are decomposed into smaller relation schemas in order to meet the tests.
- Most databases in industry are in 3NF or BCNF, and at most 4NF. Lower forms may be kept for performance reasons.

Normalisation of data - Definitions

- **Normal Form**: The normal form of a relation refers to the highest normal form condition that it meets, indicating the degree to which it has been normalised.
- **Superkey**: For $R(A_1, ..., A_n)$, the set of attributes $S \subseteq R$ is the superkey when for t_1 and t_2 from a legal relation state r of R, $t_1[S] = t_2[S]$.
- **Key**: A key K is a superkey with the property that the removal of an attribute from K, will no longer render it a superkey.
  - A key is therefore minimal.
  - A candidate key is one of the attributes forming the key for relations with more than one key. One of the candidate keys is designated as the **primary key**. If a relation has only one candidate key, it is the primary key.
- **Prime attribute**: An attribute in R is a prime attribute of R if it is an attribute of a candidate key.
- **Nonprime attribute**: An attribute that is not prime is nonprime.

What is coming in lectures 22 and 23

The normalisation process:

- Normal forms 1 to 3
- BCNF
- Normal forms 4 and 5