Functional Dependencies and Normalization

Jose M. Peña
jospe@ida.liu.se

* slides kindly provided by Vaida Jakonienė

Good Design

- Can we be sure that a translation from EER-diagram to relational tables results in good database design?
- Confronted with a deployed database, how can we be sure that it is well-designed?
- What is good database design?
  - Four informal measures
  - Formal measure: normalization

Informal design guideline

- Easy to explain semantics of the relation schema
- Reducing redundant information in tuples

Redundancy causes update anomalies:
- Insertion anomalies
- Deletion anomalies
- Modification anomalies

Informal design guideline

- Sometimes, it may be desirable to have redundancy to gain in runtime, i.e. trade space for time.
- In that case and to avoid update anomalies
  - either, use triggers or stored procedures to update the base tables
  - or, keep the base tables free of redundancy and use views (assuming that the views are materialized).

Informal design guideline

- Reducing NULL values in tuples
  - Why
    - Efficient use of space
    - Avoid costly outer joins
    - Ambiguous interpretation (unknown vs. doesn’t apply).
  - Disallow the possibility of generating spurious tuples
    - Figures 10.5 and 10.6: cartesian product results in incorrect tuples
    - Only join on foreign key/primary key-attributes
    - Lossless join property: guarantees that the spurious tuple generation problem does not occur
Functional dependencies (FD)

- Let R be a relational schema with the attributes $A_1, \ldots, A_n$.
- Let $r(R)$ denote a relation in relational schema R.

We say that $X$ functionally determines $Y$, $X \rightarrow Y$, if for each pair of tuples $t_1, t_2 \in r(R)$ and for all relations in $r(R)$:

\[ t_1[X] = t_2[X] \text{ then we must also have } t_1[Y] = t_2[Y] \]

Despite the mathematical definition an FD cannot be determined automatically. It is a property of the semantics of attributes.

Inference rules

1. If $X \supseteq Y$ then $X \rightarrow Y$, or $X \rightarrow X$ (reflexive rule)
2. $X \rightarrow Y \models XZ \rightarrow YZ$ (augmentation rule)
3. $X \rightarrow Y$, $Y \rightarrow Z \models X \rightarrow Z$ (transitive rule)
4. $X \rightarrow YZ \models X \rightarrow Y$ (decomposition rule)
5. $X \rightarrow Y$, $X \rightarrow Z \models X \rightarrow YZ$ (union or additive rule)
6. $X \rightarrow Y$, $WY \rightarrow Z \models WX \rightarrow Z$ (pseudotransitive rule)

Definitions

- **Superkey**: a set of attributes uniquely (but not minimally!) identifying a tuple of a relation.
- **Key**: A set of attributes that uniquely and minimally identifies a tuple of a relation.
- **Candidate key**: If there is more than one key in a relation, the keys are called candidate keys.
- **Primary key**: One candidate key is chosen to be the primary key.
- **Prime attribute**: An attribute $A$ that is part of a candidate key $X$ (vs. nonprime attribute)

Normal Forms

- **1NF, 2NF, 3NF, BCNF (4NF, 5NF)**
- **Minimize redundancy**
- **Minimize update anomalies**
- Normal form $\uparrow$ = redundancy and update anomalies $\downarrow$ and relations become smaller.
- Join operation to recover original relations.

1NF

- **1NF**: The relation should have no non-atomic values.
2NF

- 2NF: no nonprime attribute should be functionally dependent on a part of a candidate key (= partial dependency).

<table>
<thead>
<tr>
<th>EmpID</th>
<th>Dept</th>
<th>Work%</th>
<th>EmpName</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Dev</td>
<td>50</td>
<td>Baker</td>
</tr>
<tr>
<td>100</td>
<td>Support</td>
<td>50</td>
<td>Baker</td>
</tr>
<tr>
<td>200</td>
<td>Dev</td>
<td>80</td>
<td>Miller</td>
</tr>
</tbody>
</table>

Normalization

2NF:

- No 2NF: A part of a candidate key can have repeated values in the relation and, thus, so can have the nonprime attribute, i.e. redundancy + insertion and modification anomalies.

- An FD X \(\rightarrow\) Y is a full functional dependency (FFD) if removal of any attribute A, from X means that the dependency does not hold any more.

- 2NF: Every nonprime attribute is fully functionally dependent on every candidate key.

3NF

- 3NF: 2NF + no nonprime attribute should be functionally dependent on a set of nonprime attributes

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Zip</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Andersson</td>
<td>58214</td>
<td>Linköping</td>
</tr>
<tr>
<td>101</td>
<td>Björk</td>
<td>10223</td>
<td>Stockholm</td>
</tr>
<tr>
<td>102</td>
<td>Carlsson</td>
<td>58214</td>
<td></td>
</tr>
</tbody>
</table>

Normalization

3NF:

- No 3NF (but 2NF): A set of nonprime attributes can have repeated values in the relation and, thus, so can have the nonprime attribute, i.e. redundancy + insertion and modification anomalies.

- An FD X \(\rightarrow\) Y is a transitive dependency if there is a set of nonprime attributes Z such that both X \(\rightarrow\) Z and Z \(\rightarrow\) Y hold.

- 3NF: 2NF + no nonprime attribute is transitively dependent on any candidate key.

Little summary

- X \(\rightarrow\) A
- 2NF and 3NF do nothing if A is prime.
- Assume A is nonprime.
- 2NF = decompose if X is part of a candidate key.
- 3NF = decompose if X is part of a candidate key or X is nonprime, i.e. if X \(\rightarrow\) A is partial or transitive.
- 3NF = X is a superkey or A is prime.
- Should A be discriminated for being prime?

Boyce-Codd Normal Form

- BCNF: Every determinant is a superkey (in practice: every determinant is a candidate key)

- BCNF = decompose if X \(\rightarrow\) A is such that X is not a superkey and A is a prime attribute.

Example: Given R(A,B,C,D) and AB \(\rightarrow\) CD, C \(\rightarrow\) B. Then R is in 3NF but not in BCNF. C is a determinant but not a superkey (tuples are not uniquely identified in R)
BCNF: Example

At a gym, an instructor is leading an activity in a certain room at a certain time.

<table>
<thead>
<tr>
<th>Time</th>
<th>Room</th>
<th>Instructor</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon 17.00</td>
<td>Gym</td>
<td>Tina</td>
<td>IronWoman</td>
</tr>
<tr>
<td>Mon 17.00</td>
<td>Mirrors</td>
<td>Anna</td>
<td>Aerobics</td>
</tr>
<tr>
<td>Tue 17.00</td>
<td>Gym</td>
<td>Tina</td>
<td>Intro</td>
</tr>
<tr>
<td>Tue 17.00</td>
<td>Mirrors</td>
<td>Anna</td>
<td>Aerobics</td>
</tr>
<tr>
<td>Wed 18.00</td>
<td>Gym</td>
<td>Anna</td>
<td>IronWoman</td>
</tr>
</tbody>
</table>

Properties of decomposition

- Keep all attributes from the universal relation R.
- Preserve the identified functional dependencies.
- Lossless join
  - It must be possible to join the smaller tables to arrive at composite information without spurious tuples.

Normalization: Example

Given universal relation

\[ R(PID, \text{PersonNamn}, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{AntalBesökILandet}) \]

- Functional dependencies?
- Keys?

Normalization: Example

Is \( R(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{PersonNamn}, \text{AntalBesökILandet}) \) in 2NF?

No, PersonNamn depends on a part of the key (PID), then

\[ R_1(PID, \text{PersonNamn}) \]

\[ R_2(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{AntalBesökILandet}) \]

Is \( R_2 \) in 2NF?

No, Kontinent and KontinentYta depend on a part of the key (Land), then

\[ R_3(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{AntalBesökILandet}) \]

\[ R_4(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{PersonNamn}, \text{AntalBesökILandet}) \]

\[ R_1, R_2, R_3, R_4 \text{ are in 2NF} \]

2NF: no nonprime attribute should be functionally dependent on a part of a candidate key.

Is \( R_1(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{PersonNamn}, \text{AntalBesökILandet}) \) in 2NF?

No, PersonNamn depends on a part of the key (PID), then

\[ R_1(PID, \text{PersonNamn}) \]

\[ R_2(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{AntalBesökILandet}) \]

Is \( R_2 \) in 2NF?

No, Kontinent and KontinentYta depend on a part of the key (Land), then

\[ R_3(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{PersonNamn}, \text{AntalBesökILandet}) \]

\[ R_4(PID, \text{Land}, \text{Kontinent}, \text{KontinentYta}, \text{PersonNamn}, \text{AntalBesökILandet}) \]

\[ R_1, R_2, R_3, R_4 \text{ are in 2NF} \]

2NF: no nonprime attribute should be functionally dependent on a part of a candidate key.
Are R1, R21, R22 in 3NF?

R22(PID, Land, AntalBesökILandet),
R1(PID, PersonNamn):
Yes, a single nonprime attribute, no transitive
dependencies.

R21(Land, Kontinent, KontinentYta):
No, Kontinent defines KontinentYta, then
R211(Land, Kontinent)
R212(Kontinent, KontinentYta)

→ R1, R22, R211, R212 are in 3NF

Are R1, R22, R211, R212 in BCNF?

R22(PID, Land, AntalBesökILandet),
R1(PID, PersonNamn):
R211(Land, Kontinent)
R212(Kontinent, KontinentYta)

→ Yes (don’t be confused by candidate keys!)

Can the universal relation R be reproduced from R1, R22, R211 and R212 without spurious tuples?

Summary and open issues

- Good design: informal and formal properties of relations
- Functional dependencies, and thus normal
  forms, are about attribute semantics (= real-
  world knowledge), normalization can only be
  automated if FDs are given.
- Are high normal forms good design when it
  comes to performance?
  - No, denormalization may be required.