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 A₁,...,Aₙ and let X and Y be subsets of {A₁,...,Aₙ}.
- 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 \( t₁, t₂ \in r(R) \) and for all relations in \( r(R) \):

- If \( t₁[X] = t₂[X] \), then we must also have \( t₁[Y] = t₂[Y] \).

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. \( X \rightarrow Y \mid \rightarrow XZ \rightarrow YZ \) (augmentation rule)
3. \( X \rightarrow Y, Y \rightarrow Z \mid \rightarrow X \rightarrow Z \) (transitive rule)
4. \( X \rightarrow YZ \rightarrow X \rightarrow Y \) (decomposition rule)
5. \( X \rightarrow Y, X \rightarrow Z \mid \rightarrow X \rightarrow YZ \) (union or additive rule)
6. \( X \rightarrow Y, WY \rightarrow Z \mid \rightarrow WX \rightarrow Z \) (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.

\[ R_{\text{non1NF}} \]

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Livesin</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Pettersson</td>
<td>Stockholm, Linköping</td>
</tr>
<tr>
<td>101</td>
<td>Andersson</td>
<td>Linköping</td>
</tr>
<tr>
<td>102</td>
<td>Svensson</td>
<td>Ystad, Hjo, Berlin</td>
</tr>
</tbody>
</table>

\[ R_{\text{1NF}} \]

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Pettersson</td>
</tr>
<tr>
<td>101</td>
<td>Andersson</td>
</tr>
<tr>
<td>102</td>
<td>Svensson</td>
</tr>
</tbody>
</table>

What about multi-valued composite attributes?
2NF
- 2NF: no nonprime attribute should be functionally dependent on a part of a candidate key (= partial dependency).

<table>
<thead>
<tr>
<th>R1_{non2NF}</th>
<th>R2_{non2NF}</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmpID</td>
<td>Dept</td>
</tr>
<tr>
<td>100</td>
<td>Dev</td>
</tr>
<tr>
<td>200</td>
<td>Support</td>
</tr>
<tr>
<td>300</td>
<td>Dev</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→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>R1_{non3NF}</th>
<th>R2_{non3NF}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Name</td>
</tr>
<tr>
<td>102</td>
<td>Andersson</td>
</tr>
<tr>
<td>101</td>
<td>Björk</td>
</tr>
<tr>
<td>100</td>
<td>Carlsson</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→Y is a transitive dependency if there is a set of nonprime attributes Z such that both X→Z and Z→Y hold.

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

Little summary
- X → 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 → 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 → A is such that X is not a superkey and A is a prime attribute.

Example: Given R(A,B,C,D) and AB→CD, C→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

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

- Functional dependencies?
- Keys?

Normalization: Example

PID \rightarrow PersonNamn

PID, Land \rightarrow AntalBesökILandet

Land \rightarrow Kontinent

Kontinent \rightarrow KontinentYta

- Based on FDs, what are keys for R?
- Use inference rules

Normalization: Example

\[ Land \rightarrow Kontinent, Kontinent \rightarrow KontinentYta, \]
\[ \text{then} \]
\[ Land \rightarrow Kontinent, KontinentYta \]

PID, Land \rightarrow Kontinent, KontinentYta (augmentation rule),

PID, Land \rightarrow PersonNamn (augmentation rule),

PID, Land \rightarrow AntalBesökILandet (transitive rule),

\[ \text{then} \]

PID, Land \rightarrow Kontinent, KontinentYta, PersonNamn, AntalBesökILandet (additive rule)

Person, Land is the key for R.

Normalization: Example

Is \[ R(PID, Land, Kontinent, KontinentYta, PersonNamn, AntalBesökILandet) \] in 2NF?

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

R1(PID, PersonNamn)

R2(PID, Land, Kontinent, KontinentYta, AntalBesökILandet)

Is R2 in 2NF?

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

R1(PID, PersonNamn)

R21(Land, Kontinent, KontinentYta)

R22(PID, Land, AntalBesökILandet) \rightarrow R1, R21, R22 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

Summary and open issues

- Good design: The canonical properties of relations
- Functional dependencies:
- Normal forms, are real-world knowledge, not only be automated.
- Are high normal form when it comes to performance.

1. Which normal form?

- The database contains data about cars, their owners and when the car was registered for that owner.

<table>
<thead>
<tr>
<th>PersonID</th>
<th>FirstName</th>
<th>LastName</th>
<th>LicensePlate</th>
<th>RegistrationDate</th>
<th>Birthdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Ann</td>
<td>Anderson</td>
<td>ABC123</td>
<td>2004-10-12</td>
<td>1981-04-04</td>
</tr>
<tr>
<td>1010</td>
<td>Ben</td>
<td>Benson</td>
<td>DEF234</td>
<td>2003-02-12</td>
<td>1945-12-12</td>
</tr>
<tr>
<td>1005</td>
<td>Ann</td>
<td>Anderson</td>
<td>ABC123</td>
<td>2001-04-23</td>
<td>1981-04-04</td>
</tr>
</tbody>
</table>

2. Which normal form?

- A database contains data about registered cars and their make (type).

<table>
<thead>
<tr>
<th>LicensePlate</th>
<th>Type</th>
<th>Maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC123</td>
<td>C70</td>
<td>Volvo</td>
</tr>
<tr>
<td>DEF234</td>
<td>S40</td>
<td>Volvo</td>
</tr>
<tr>
<td>FGH345</td>
<td>Corolla</td>
<td>Toyota</td>
</tr>
</tbody>
</table>

3. Which normal form?

- The database contains data about flights, aircrafts and their pilots. Flights use different aircrafts depending on the number of booked passengers.

<table>
<thead>
<tr>
<th>Date</th>
<th>Flight</th>
<th>Aircraft</th>
<th>Pilot</th>
</tr>
</thead>
<tbody>
<tr>
<td>13-Jan-2005</td>
<td>TGU7</td>
<td>Airbus 300</td>
<td>John</td>
</tr>
<tr>
<td>14-Jan-2005</td>
<td>TGU7</td>
<td>Boeing 747</td>
<td>Daniel</td>
</tr>
<tr>
<td>12-Jan-2005</td>
<td>SKX6</td>
<td>Airbus 300</td>
<td>John</td>
</tr>
<tr>
<td>13-Jan-2005</td>
<td>SKX6</td>
<td>Boeing 747</td>
<td>Ann</td>
</tr>
<tr>
<td>14-Jan-2005</td>
<td>SKX6</td>
<td>Fokker 50</td>
<td>Mary</td>
</tr>
</tbody>
</table>