Overview

- BCNF exercise example
- Decomposition to BCNF
  - algorithm for lossless decomposition
- 4NF
- Normalisation example
- Start of indexing?
Last time

- To reason about BCNF or 3NF we need to know all logically implied FDs, not just the given set.
- One way to enumerate all FDs is to compute all attribute closures.
- Attribute closure:
  - Let \( \alpha \) be set of attributes
  - Attribute closure \( \alpha^+ \) is also a set of attributes
  - \( X \in \alpha^+ \) if \( \alpha \rightarrow X \)

Example

- Consider schema \( R(A,B), S(B,C,D,E) \) with dependencies 
  \[
  B \rightarrow DE \\
  B \rightarrow A \\
  A \rightarrow CD \\
  CE \rightarrow B 
  \]
- Compute all candidate keys
- Is \( R \) in BCNF?
- Is it in 3NF?
- (this is a typical exam exercise)
Decomposition into BCNF

- Not a BCNF

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>salary</th>
<th>dept_name</th>
<th>building</th>
<th>budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>22222</td>
<td>Einstein</td>
<td>95000</td>
<td>Physics</td>
<td>Watson</td>
<td>70000</td>
</tr>
<tr>
<td>12121</td>
<td>Wu</td>
<td>90000</td>
<td>Finance</td>
<td>Painter</td>
<td>120000</td>
</tr>
<tr>
<td>32343</td>
<td>El Said</td>
<td>60000</td>
<td>History</td>
<td>Painter</td>
<td>50000</td>
</tr>
<tr>
<td>45565</td>
<td>Katz</td>
<td>75000</td>
<td>Comp. Sci.</td>
<td>Taylor</td>
<td>100000</td>
</tr>
<tr>
<td>98345</td>
<td>Kim</td>
<td>80000</td>
<td>Elec. Eng.</td>
<td>Taylor</td>
<td>85000</td>
</tr>
<tr>
<td>76766</td>
<td>Crick</td>
<td>72000</td>
<td>Biology</td>
<td>Watson</td>
<td>90000</td>
</tr>
<tr>
<td>10101</td>
<td>Srinivasan</td>
<td>65000</td>
<td>Comp. Sci.</td>
<td>Taylor</td>
<td>100000</td>
</tr>
<tr>
<td>58883</td>
<td>Califieri</td>
<td>62000</td>
<td>History</td>
<td>Painter</td>
<td>50000</td>
</tr>
<tr>
<td>83823</td>
<td>Brandt</td>
<td>92000</td>
<td>Comp. Sci.</td>
<td>Taylor</td>
<td>100000</td>
</tr>
<tr>
<td>15151</td>
<td>Mozart</td>
<td>40000</td>
<td>Music</td>
<td>Packard</td>
<td>80000</td>
</tr>
<tr>
<td>33456</td>
<td>Gold</td>
<td>87000</td>
<td>Physics</td>
<td>Watson</td>
<td>70000</td>
</tr>
<tr>
<td>76843</td>
<td>Singh</td>
<td>80000</td>
<td>Finance</td>
<td>Painter</td>
<td>120000</td>
</tr>
</tbody>
</table>

- Decompose to BCNF:
  - instructor(ID, name, salary, dept_name)
  - department(dept_name, building, budget)

BCNF decomposition

- Compute F^+

- Repeat the following while the schema is not BCNF
  - Find a BCNF violation A_1 A_2 ... A_n \rightarrow B_1 B_2 ... B_m in schema R(\alpha)
  - Decompose R into ((\alpha-B_1 B_2 ... B_m) \cup A_1 A_2 ... A_n) and (A_1 A_2 ... A_n B_1 B_2 ... B_m)
Example

- Decompose the relation

\[ cd\_shop(cd\_id, artist, title, order\_id, order\_date, quantity, customer\_id, name, address) \]

- With the functional dependencies

\[ cd\_id \rightarrow artist, title \]
\[ customer\_id \rightarrow name, address \]
\[ order\_id \rightarrow order\_date, customer\_id \]
\[ order\_id, cd\_id \rightarrow quantity \]

Non determinancy

- Much depends on the choice of BCNF violation
- Try e.g. decomposing first using

\[ order\_id \rightarrow order\_date, customer\_id \]
- There is no guarantee that decomposition is dependency preserving
- (even if there is a dependency preserving decomposition)
- One heuristic is to maximise right hand sides of BCNF violations
Correctness

• Correctness:
  - Tables become smaller for every decomposition
  - Every 2-attribute table is BCNF
  - So in the end, the schema must be BCNF
• Every decomposition is lossless
• In fact if $\alpha \rightarrow \beta$ then decomposition of $R(\alpha\beta\gamma)$ into $(\alpha\beta)$ and $(\alpha\gamma)$ is always lossless (book page 346)

Discussion

• BCNF algorithm suggests a new strategy to DB design:
  - Put everything in one table
  - Write up all FDs
  - Apply BCNF decomposition
• This is not a good strategy
• This can lead to terrible designs
• Without ER diagram, it is also harder to extend an existing DB design
4NF

Example

• Consider a database storing information about movie stars

<table>
<thead>
<tr>
<th>name</th>
<th>address</th>
<th>movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samuel L Jackson</td>
<td>Sunshine Blvd 1</td>
<td>Pulp Fiction</td>
</tr>
<tr>
<td>Samuel L Jackson</td>
<td>Rainy Street 134</td>
<td>Pulp Fiction</td>
</tr>
<tr>
<td>Samuel L Jackson</td>
<td>Sunshine Blvd 1</td>
<td>Snakes on a Plane</td>
</tr>
<tr>
<td>Samuel L Jackson</td>
<td>Rainy Street 134</td>
<td>Snakes on a Plane</td>
</tr>
</tbody>
</table>

• Clearly lots of redundancy here
• But no non-trivial functional dependencies!
• So BCNF
Problem in a nutshell

- Attributes *address* and *movie* are independent and not determined by other attributes
- For every pair of tuples

<table>
<thead>
<tr>
<th>name</th>
<th>address</th>
<th>movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor name</td>
<td>Address 1</td>
<td>Movie 1</td>
</tr>
<tr>
<td>Actor name</td>
<td>Address 2</td>
<td>Movie 2</td>
</tr>
</tbody>
</table>

- There are also tuples

<table>
<thead>
<tr>
<th>name</th>
<th>address</th>
<th>movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor name</td>
<td>Address 1</td>
<td>Movie 2</td>
</tr>
<tr>
<td>Actor name</td>
<td>Address 2</td>
<td>Movie 1</td>
</tr>
</tbody>
</table>

- This is called a **multivalued dependency**

Multivalued dependencies

- Consider a table $R(\alpha\beta\gamma)$
- **Definition.** There is a multivalued dependency $\alpha \rightarrow \beta$ if for all tuples $t,u$ in all legal instances
  - if $t[\alpha] = u[\alpha]$
  - then there exists tuple $s$ such that
    
    $s[\alpha] = t[\alpha]$
    $s[\beta] = t[\beta]$
    $s[\gamma] = u[\gamma]$

- In example *name* $\rightarrow$ *address*
Rules for multivalued dependencies

- In $R(\alpha\beta\gamma)$ if $\alpha \rightarrow \beta$ then also $\alpha \rightarrow \gamma$
- If $\alpha \rightarrow \beta$ then also $\alpha \rightarrow \beta$ (can take $s = u$)
- Consequences
  - if $\beta \subseteq \alpha$ then $\alpha \rightarrow \beta$
  - if $\alpha$ is a superkey then $\alpha \rightarrow \beta$
- It is **not** the case that if $\alpha \rightarrow \beta\gamma$ then $\alpha \rightarrow \beta$

4NF

- **Definition.** A table $r(R)$ is in **4NF** if for all multivalued dependencies $\alpha \rightarrow \beta$ either
  - $\beta \subseteq \alpha$ ($\alpha \rightarrow \beta$ is trivial)
  - or $\alpha$ is a superkey
- **Definition.** A schema is in **4NF** if all tables are in 4NF
- **Theorem.** A schema in 4NF is also BCNF
4NF decomposition

- There is a lossless decomposition algorithm for 4NF
- It is the same as the one for BCNF but uses multivalued dependencies

Normal forms

- A tower of normal forms
  - 4NF
  - BCNF
  - 3NF
  - 2NF
  - 1NF
- Any schema satisfying a normal form also satisfies the ones below
- (there do exist even higher normal forms)
create table Cargo
    (ID integer PRIMARY KEY,
     RD varchar (3), -- Region Departure
     RA varchar (3), -- Region Arrival
     CD varchar (3), -- Country Departure
     CA varchar (3), -- Country Arrival
     AL varchar (3), -- Airline
     FNR varchar (5), -- Flight Number
     SNR integer, -- Header Segment Number
     DEP varchar (3), -- City Departure
     ARR varchar (3), -- City Arrival
     STD varchar (4), -- Schedule Time of Departure
     DDC integer, -- Date Variation of Departure
     STA varchar (4), -- Schedule Time of Arrival
     ADC integer, -- Date Variation of Arrival
     MO integer (1), -- Monday
     TU integer (1), -- Tuesday
     WE integer (1), -- Wednesday
     TH integer (1), -- Thursday
     FR integer (1), -- Friday
     SA integer (1), -- Saturday
     SU integer (1), -- Sunday
     ACTYPE varchar(3), -- Aircraft Type Code
     ACTYPEFULLNAME varchar(30), -- Aircraft Name
     AG varchar (1), -- Aircraft Group Code
     AGFULLNAME varchar(30), -- Aircraft Group Name
     START_OP_1 varchar(8), -- Start Date of Flight Operations
     END_OP_1 varchar(8) ); -- End Date of Flight Operations
We are told that the following functional dependencies hold:

\[
\begin{align*}
ACTYPE & \rightarrow ACTYPEFULLNAME \\
AG & \rightarrow AGFULLNAME \\
ACTYPE & \rightarrow AG
\end{align*}
\]

Task 1: Check that the cargo database respects these three functional dependencies:

```sql
mysql> select actype from cargo group by actype having count(distinct actypefullname)>1;
Empty set (0.03 sec)
```

```sql
mysql> select ag from cargo group by ag having count(distinct agfullname)>1;
Empty set (0.03 sec)
```

```sql
mysql> select actype from cargo group by actype having count(distinct ag)>1;
Empty set (0.03 sec)
```

Task 2: Check if ACTYPE and AG are candidate keys:

Inspect the data, or look at the number of occurrences of each actype value:

```sql
mysql> select actype,count(*) from cargo group by actype having count(*)>1;
```

```
+--------+----------+
<table>
<thead>
<tr>
<th>actype</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>781</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>SHP</td>
<td>11</td>
</tr>
</tbody>
</table>
+--------+----------+
```

44 rows in set (0.03 sec)

```sql
mysql> select ag,count(*) from cargo group by ag having count(*)>1;
```

```
+------+----------+
<table>
<thead>
<tr>
<th>ag</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1572</td>
</tr>
<tr>
<td>O</td>
<td>14889</td>
</tr>
<tr>
<td>T</td>
<td>2409</td>
</tr>
<tr>
<td>W</td>
<td>6472</td>
</tr>
</tbody>
</table>
+------+----------+
```

4 rows in set (0.03 sec)

Conclusion: None of them are keys, should split into separate tables to achieve BCNF.

Task 3. Compute tables according to the FDs

\[
\begin{align*}
ACTYPE & \rightarrow ACTYPEFULLNAME, AG \\
AG & \rightarrow AGFULLNAME
\end{align*}
\]

```sql
mysql> create table aircraft as (SELECT distinct actype,actypefullname,ag from cargo);
```

Query OK, 46 rows affected (0.15 sec)

```sql
create table acgroup as (select distinct ag, agfullname from cargo);
```

```sql
mysql> 
```
Summary

- Algorithm for lossless decomposition into BCNF
- Even BCNF schemes may have redundancy
- 4NF normalisation gets rid of even more redundancy

Intended learning outcomes:
- Derive FDs from real world description
- Judge if schema is BCNF or 3NF
- Normalise to BCNF