3NF Synthesis: Why it works

- Preserves dependencies
  - Each FD from a minimal basis is contained in a relation (therefore preserved).
- Lossless Join:
  - We can use the chase to show that the **row for the relation containing the key** can be made of all-unsubscripted variables.
- Hard part:
  - Computing a minimal basis.

Multivalued dependency

- Intuition:
  - We may want to state that if certain tuples are in a relation instance...
  - ... then **other tuples must also be** in the relation instance.
- A **multivalued dependency** (MVD) on R:
  - $X \twoheadrightarrow Y$
  - If two tuples of R agree on all attributes of X...
  - ... then their components in Y **must be swapped between the tuples**...
  - ... and the result will be two tuples that are also in the relation.

Multivalued dependency

- Given an MVD of the form:

  $A_1A_2...A_n \twoheadrightarrow B_1B_2...B_m$

- We say this MVD holds if:
  - For each pair of tuples **t** and **u** of relation R that agree on all As, we can find in R some tuple **v** that agrees...
  - ... with both **t** and **u** on the As, ...
  - ... with **t** on the Bs, and ...
  - with **u** on all attributes of R that are not among the As or Bs

Example: multivalued dependency

- **Patrons(name, addr, phone, beersLiked)**
  - (This is somewhat modified from previous Patrons schema examples.)
- The phone numbers of a patron are independent of the beers they happen to like.
  - name $\twoheadrightarrow$ phone
  - name $\twoheadrightarrow$ beersLiked
- Thus:
  - Each patron's phone appears with each of the beers they like in all combinations
- Note: This repetition is not the same as a functional-dependency redundancy!

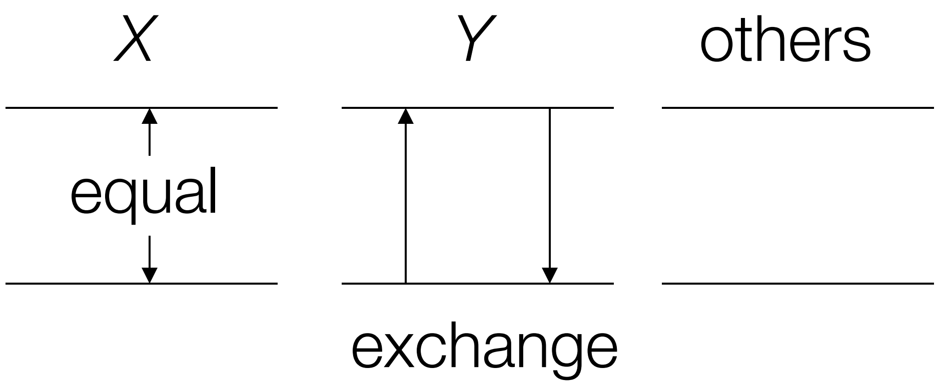
Example: MVD

If the top two tuples are in the relation...

name	addr	phone	beersLiked
Carla	Tyndall	250-893-1111	Stella Artois
Carla	Tyndall	250-213-9999	König Ludwig
Carla	Tyndall	***250-213-9999***	***Stella Artois***
Carla	Tyndall	***250-893-1111***	***König Ludwig***

... then given the MVDs, the bottom two tuples must also be in the relation.

Picture of MVD X $\twoheadrightarrow$ Y



MVD rules

- Every functional dependency is also a multi-valued dependency
  - If X \rightarrow Y then swapping Ys between two tuples that agree on X does not change the tuples.
  - Therefore the so-called new tuples are surely in the relation, and we we know $X \twoheadrightarrow Y$
  - (This rule is called **promotion**.)
- If $X \twoheadrightarrow Y$...
  - ... and Z makes up all other attributes in the schema...
  - ... then $X \twoheadrightarrow Z$
  - (This rule is called **complementation**.)

No splitting allowed...

- Like FDs we cannot generally split the left side of an MVD
- However, unlike FDs, we cannot split the right side either!
  - Sometimes we must leave several attributes on the right side.
- (Text also makes mention of trivial MVDs, but we'll leave this aside until we cover 4NF)

Example: RHS of multivalued dependencies

- **Patrons(name, areaCode, phoneNum, beersLiked, manf)**
- Observations:
  - A patron can have several phones; here the full phone number is divided between areaCode and phoneNum
  - A patron can like several beers; each beer has its own manufacturer.
- We expect the following to be independent for a patron
  - areaCode / phoneNum combination
  - beersLiked / manf combination
- Therefore we expect the following MVDs to hold
  - name $\twoheadrightarrow$ areaCode phoneNum
  - name $\twoheadrightarrow$ beersLiked manf

Example: RHS of multivalued dependencies

Example data for Patrons

name	areaCode	phoneNum	beersLiked	manf
Carla	250	893-1111	Stella Artois	Anheuser-Busch
Carla	250	893-1111	König Ludwig	Kaltenberg
Carla	250	213-9999	Stella Artois	Anheuser-Busch
Carla	250	213-9999	König Ludwig	Kaltenberg

But we cannot swap area codes or phone numbers by themselves. Neither of the following hold for this relation:

name $\twoheadrightarrow$ areaCode

name $\twoheadrightarrow$ phoneNum

Fourth Normal Form

- In our examples so far we have detected some redundancy
  - However, redundancy introduced by MVDs cannot be removed by normalizing to BCNF
- There is a stronger normal form
  - 4NF treats MVDs as FDs when it comes to decomposition...
  - .. but does not treat them as FDs when determining keys for a relation.

Fourth Normal Form

- A relation R is in fourth normal form:
  - If whenever X $\twoheadrightarrow$ Y is a non-trivial MVD...
  - ... then X is a superkey.
- Superkey:
  - This still depends on the FD definition only
- Non-trivial MVD X $\twoheadrightarrow$ Y
  1. Y is not a subset of X, **and**
  2. X and Y are not all of the attributes of the schema

BCNF vs. 4NF

- Recall:
  - Every FD $X \rightarrow Y$ is also an MVD $X \twoheadrightarrow Y$
- Thus if R is in 4NF, it is also in BCNF
  - Any BCNF violation is also a 4NF violation (i.e., after conversion to an MVD)
- Note the reverse is not necessarily true
  - R could be in BCNF and not 4NF.
  - MVDs are – in effect – invisible to the definition of BCNF

4NF decomposition

- If $X \twoheadrightarrow Y$ causes a 4NF violation in R:
  - We can decomposed R using the same technique as for BCNF
- $R_1 = XY$ becomes one of the decomposed relations
- $R_2 = R - (Y - X)$ becomes the other

Example 4NF decomposition

- **Patrons(name, addr, phone, beersLiked)**
- FDs:
  - name $\rightarrow$ addr
- MVDs
  - name $\twoheadrightarrow$ phone
  - name $\twoheadrightarrow$ beersLiked
- It appears that the key for this relation is:
  - {name, phone, beersLiked}
- Therefore all dependencies violate 4NF
- To begin decomposition, we must pick a violating FD or MVD...

Example (continued)

- We'll choose:
  - name $\rightarrow$ addr
  - Two new schemas result
- **Patrons$_1$(name, addr)**
  - Note that this is in 4NF.
  - Only dependency now for Patrons$_1$: name $\rightarrow$ addr
- **Patrons$_2$(name, phone, beersLiked)**
  - Not in 4NF. Why?
  - Valid: name $\twoheadrightarrow$ phone & name $\twoheadrightarrow$ beersLiked
  - However, no FDs apply
  - All three attributes form the key – so we must decompose further.

Example (continued)

- Decompose **$Patrons_2$(name, phone, beersLiked)**
  - MVDs: name $\twoheadrightarrow$ phone & name $\twoheadrightarrow$ beersLiked
  - Choose either (resulting decomposition is the same in this case)
  - $Patron_3$(name, phone)
  - $Patron_4$(name, beersLiked)

Reasoning about MVDs and FDs

- A problem:
  - Given a set of MVDs or FDs or both that hold for a relation R, **does an inferred FD or MVD** also hold in R?
- We saw earlier how to do this for FDs (i.e., involve computing closures of attribute sets)
  - However, it is not quite as straightforward when MVDs are involved
- Why do we care?
  1. We may also need to project FDs and MVDs during projection
  2. 4NF technically requires an MVD violation, so... **we may sometimes need to infer MVDs from given FDs and MVDs that are themselves not in violation**

Idea: Use the chase again

- It turns out our "closure" procedure is similar to the chase.
- To prove if an FD or MVD is valid:
  1. Build a tableau with knowns and unknowns properly noted (i.e., variables without subscripts and those with subscripts)
  2. Apply FDs from the set F as before in order to equate symbols
  3. Apply MVDs by generating one or both of the tuples we know must also be in the relation represented by the tableau.

Example: Try to prove A → C

Goal: Given A ↠ BC & D → C, Somehow prove that $c_1 = c_2$

A	B	C	D
a	b_1	$\cancel{c_1}$ c_2	d_1
a	b_2	c_2	d_2
a	b_2	c_2	d_1

Use A↠ BC (first row's D with second row's BC).

Use D→C (first and third row agree on *D*, therefore agree on *C*).

Example: Show transitivity works for MVDS

- Transitivity on MVDs:
  - If $A \twoheadrightarrow B$ and $B \twoheadrightarrow C$ then $A \twoheadrightarrow C$
- ??? Does it?
- Seems to be obvious from the **complementation rule** if schema is ABC
- But let us see if it holds no matter what the schema is
  - Let us assume ABCD

Example: Show A ↠ C

Goal: derive tuple (a, b_1, c_2, d_1).

A	B	C	D
a	b_1	c_1	d_1
a	b_2	c_2	d_2
a	b_1	c_2	d_2
a	b_1	c_2	d_1

Use A↠B to swap *B* from the first row into the second (hence producing third tuple)

Use B↠ C to swap *C* from the third row into the first.

Rules for inferring MVDs and FDs

- Start with a tableau of two rows
- These two rows agree on the attributes of the left-side from the dependency we want to prove as inferred
  - And the rows, of course, disagree on all other attributes
- We use:
  - unsubscripted variables where attributes agree
  - subscripted variables where to do not agree

Inference

- Applying an FD X $\rightarrow$ Y
  - Find rows that agree on all attributes X.
  - Force the rows to agree on all attributes of Y
  - Replace on variable by the other
  - If the replaced variable is part of the goal tuple, replace it there too.
- Applying a MVD X $\twoheadrightarrow$ Y by finding two rows that agree on X
  - Add to the tableau one or both rows that are formed by swapping Y components.
  - Remember that we are starting with the two rows and will be adding other rows.

Inference: Goals

- If we want to test of U $\rightarrow$V holds:
  - We succeed by inferring that the two variables in each column of V are actually the same.
- If we want to test if U $\twoheadrightarrow$ V holds:
  - We succeed if we infer in the tableau a row that results from the original two rows with components of V swapped.

Inference: Ending

- Apply all the given FDs and MVDs until we cannot change the tableau
- If we meet the goal: then the dependency is inferred
- If we do not meet the goal: then the tableau is a counterexample relation
  - That is, rows all satisfy given dependencies
  - However, the two original rows violate the target dependency.

Summary

- Functional Dependencies
- Closures
- Normalization: motivation
- BCNF
- Minimal basis
- 3NF
- Multivalue Dependencies
- 4NF

Colophon

- Some slide material is from Stanford CS145 (Jeffrey D. Ullman, Fall 2007)