Dependency-Preserving Decomposition

Designing Schemas

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

- Define dependency preserving, and explain why it is desired.
- Explain why BCNF may not be dependency preserving.

A Decomposition May Not Preserve Dependency

- A schema *S* is dependency preserving if, for every FD f of it:
  - f can be checked in a table *T* in *S*, or
  - f can be implied by those FDs that can be checked in single tables.

- Favorites(drinker, bar, beer)
- f_1: beer → bar, f_2: drinker, bar → beer
- Decomposition:
  $R_1 = (\text{beer, bar}), R_2 = (\text{beer, drinker})$
- Preserve dependencies?
  - f_1: beer → bar
  - f_2: drinker, bar → beer

drinker	bar	beer
Alex	John Bar	Sam Adams
Carissa	Green Bar	Bud Light
Alex	Purple Bar	Coors

beer	bar
Sam Adams	John Bar
Bud Light	Green Bar
Coors	Purple Bar

beer	drinker
Sam Adams	Alex
Bud Light	Carissa
Coors	Alex

Example decomposition

BCNF May **Not** Preserve Dependency

- Favorites(drinker, bar, beer)
- f_1: beer → bar, f_2: drinker, bar → beer
- Decomposition:

  $R_1 =$ (beer, bar), $R_2 =$ (beer, drinker)

- Yes, this is lossless.
- But it does not preserve dependency!

drinker	bar	beer
Alex	John Bar	Sam Adams
Carissa	Green Bar	Bud Light
Alex	Purple Bar	Coors

beer	bar
Sam Adams	John Bar
Bud Light	Green Bar
Coors	Purple Bar

beer	drinker
Sam Adams	Alex
Bud Light	Carissa
Coors	Alex

Example decomposition

Food for Thought

For a relation R and FDs F, a dependency-preserving BCNF may not exist.
Agree?

drinker	bar	beer
Alex	John Bar	Sam Adams
Carissa	Green Bar	Bud Light
Alex	Purple Bar	Coors

beer	bar
Sam Adams	John Bar
Bud Light	Green Bar
Coors	Purple Bar

beer	drinker
Sam Adams	Alex
Bud Light	Carissa
Coors	Alex

Example decomposition

Any BCNF that would preserve all dependencies?

f_1: beer → bar,
f_2: drinker, bar → beer