Table 1: Bad Mines Course Table

Instructor	Course ID	Section	Title	Office	Email
Painter-Wakefield, Christopher	CSCI 403	A	DB MANAGEMENT	BB 280I	cpainter@mines.edu
Painter-Wakefield, Christopher	CSCI 262	A	Data Structures	BB 280I	cpainter@mines.edu
Painter-Wakefield, Christopher	CSCI 262	B	Data Structures	BB 280I	cpainter@mines.edu
Mehta, Dinesh	CSCI 406	A	Algorithms	BB 280J	dmehta@mines.edu
Mehta, Dinesh	CSCI 561	A	Theory of Computation	BB 280J	dmehta@mines.edu
Hellman, Kieth	CSCI 274	A	Intro to Linux OS	BB 310 F	khellman@mines.edu

1 Definitions

1.1 Superkey

A subset X of attributes of R: no two tuples of R will ever agree on X.
If a is a superkey $a \to \{a, b, \dots\}$

1.2 Functional Dependency (FD)

If it is always true that whenever two tuples agree on X, they also agree on Y, then $X \to Y$.

Example: {instructor} →{office, email}

Trivial FD: An FD is *trivial* if $Y \subseteq X$, then $X \to Y$
Non-trivial FD: And FD is *non-trivial* if $X \to Y$ but $Y \not\subseteq X$.

1.3 Boyce-Codd Normal Form (BCNF)

A relation R is a **Boyce-Codd Normal Form (BCNF)** if for every non-trivial functional dependency $X \to A$ on R, X is a superkey of R.

Example: Clearly, instructor is not a superkey of the relation. Therefore, we say that the FD {instructor} →{office} *violates* BCNF, and the relation schema is not BCNF.

2 Decomposition Algorithm

while some relation schema is not in BCNF:
    choose some relation schema R not in BCNF
    choose some FD $X \to Y$ on R that violates BCNF
    (optional) expand Y so that $Y = X^+$ (closure of X)
    let Z be all attributes of R not included in X or Y
    replace R with two new relations
        R_1, containing attributes X, Y
        R_2, containing attributes X, Z

Note, this algorithm is ***not*** deterministic - you can decompose differently if you choose differently

3 Decomposition Notes

- The final step above is accomplished simply by projection onto the attributes in R_1 and R_2. (Recall that this may result in fewer tuples.)
- After decomposing, you will need to establish which FDs now apply to R_1 and R_2 as well as determine their superkeys, in order to determine if they are now in BCFN.
- The optional step of expanding Y is recommended, as it tends to result in fewer larger relation schemas, and may reduce the problem faster - e.g. consider decomposing instructor →office.

4 Decomposition Example

Let's use the relation schema in Figure 1 as an example.

For this schema, we listed the following FDs:

- instructor →office (violates BCNF)
- instructor →email (violates BCNF)
- {course id, section} →instructor (does not violate BCNF)
- course id →title (violates BCNF)

What superkeys do we have?

Answer: any superset of our only key, which is {course id, section}.

- Let's pick our first violating FD to work with first: instructor →office
- Next, expand the RHS as much as possible (we want the closure of instructor)
  instructor →{instructor, office, email}
- Now we decompose into two new tables, shown in the next slide:
  - $R_1 = \pi_{instructor,\ office,\ email}(R)$
  - $R_2 = \pi_{instructor,\ course\ id,\ section,\ title}(R)$
- Now table R_1 is in BCNF (note that this is not guaranteed by the algorithm – we could have had other violating FDs)
- Table R_2 has a violating FD though: course id →title
- Decomposition of R_2 via course id →title:

$$\text{course\_id}^+ = \{\text{course\_id, title}\}$$

- Decompose into

$$R_3 = \pi_{\text{course\_id, title}}(R_2)$$

$$R_4 = \pi_{\text{instructor, course\_id, title}}(R_2)$$

- Done!
  - Three tables remain: R_1, R_3, R_4
  - All non-essential redundancy has been removed
  - Each table now represents a fundamental entity
    - R_1 = instructor info
    - R_3 = course info
    - R_4 = section info

5 Correctness of Decomposition

Two requirements for correct decomposition so that we can recover original relation from decomposition using natural joins

1. natural join of decomposition must contain all attributes of the original relation
2. lossless join property: natural join of decomposition relations results in exactly the same tuples we had before decomposition

6 Multivalued Dependencies (MVD)

Def: An MVD $X \twoheadrightarrow Y$ exists on a relation R if whenever there are two tuples t_1 and t_2 which agree on attribute X, then there also exists a tuple t_3 (which could be t_1 or t_2) such that the following are true:

1. $t_3[x] = t_1[x] = t_2[x]$
2. $t_3[y] = t_2[y]$
3. $t_3[z] = t_1[z]$ where z is everything in R *not* in X and Y.

By symmetry, there must also exist t_4 with $t_4[x] = t_1[x]$, $t_4[y] = t_4[y]$, $t_4[z] = t_2[z]$.