Basics of Functional Dependencies and Normalization

Part 2

Functional Dependencies

Definition of Functional Dependency

- **Definition.** A functional dependency, denoted by $X \rightarrow Y$, between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation state r of R. The constraint is that, for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, they must also have $t_1[Y] = t_2[Y]$.

- This means that the values of the Y component of a tuple in r depend on, or are *determined by*, the values of the X component; alternatively, the values of the X component of a tuple uniquely (or **functionally**) *determine* the values of the Y component.
- We also say that there is a functional dependency from X to Y, or that Y is **functionally dependent** on X.
- The abbreviation for functional dependency is **FD** or **f.d.**
- The set of attributes X is called the **left-hand side** of the FD, and Y is called the **right-hand side**.

- Thus, X functionally determines Y in a relation schema R if, and only if, whenever two tuples of $r(R)$ agree on their X-value, they must necessarily agree on their Y-value.
- Note the following:
  - If a constraint on R states that there cannot be more than one tuple with a given X-value in any relation instance $r(R)$—that is, X is a **candidate key** of R—this implies that $X \rightarrow Y$ for any subset of attributes Y of R (because the key constraint implies that no two tuples in any legal state $r(R)$ will have the same value of X). If X is a candidate key of R, then $X \rightarrow R$.
  - If $X \rightarrow Y$ in R, this does not say whether or not $Y \rightarrow X$ in R.

- A functional dependency is a property of the **semantics** or **meaning of the attributes**.
- The database designers will use their understanding of the semantics of the attributes of *R*—that is, how they relate to one another—to specify the functional dependencies that should hold on *all* relation states (extensions) *r* of *R*.
- Whenever the semantics of two sets of attributes in *R* indicate that a functional dependency should hold, we specify the dependency as a constraint.
- Relation extensions *r*(*R*) that satisfy the functional dependency constraints are called **legal relation states**(or **legal extensions**) of *R*.
- Hence, the main use of functional dependencies is to describe further a relation schema *R* by specifying constraints on its attributes that must hold *at all times*.

- It is possible that certain functional dependencies may cease to exist in the real world if the relationship changes.

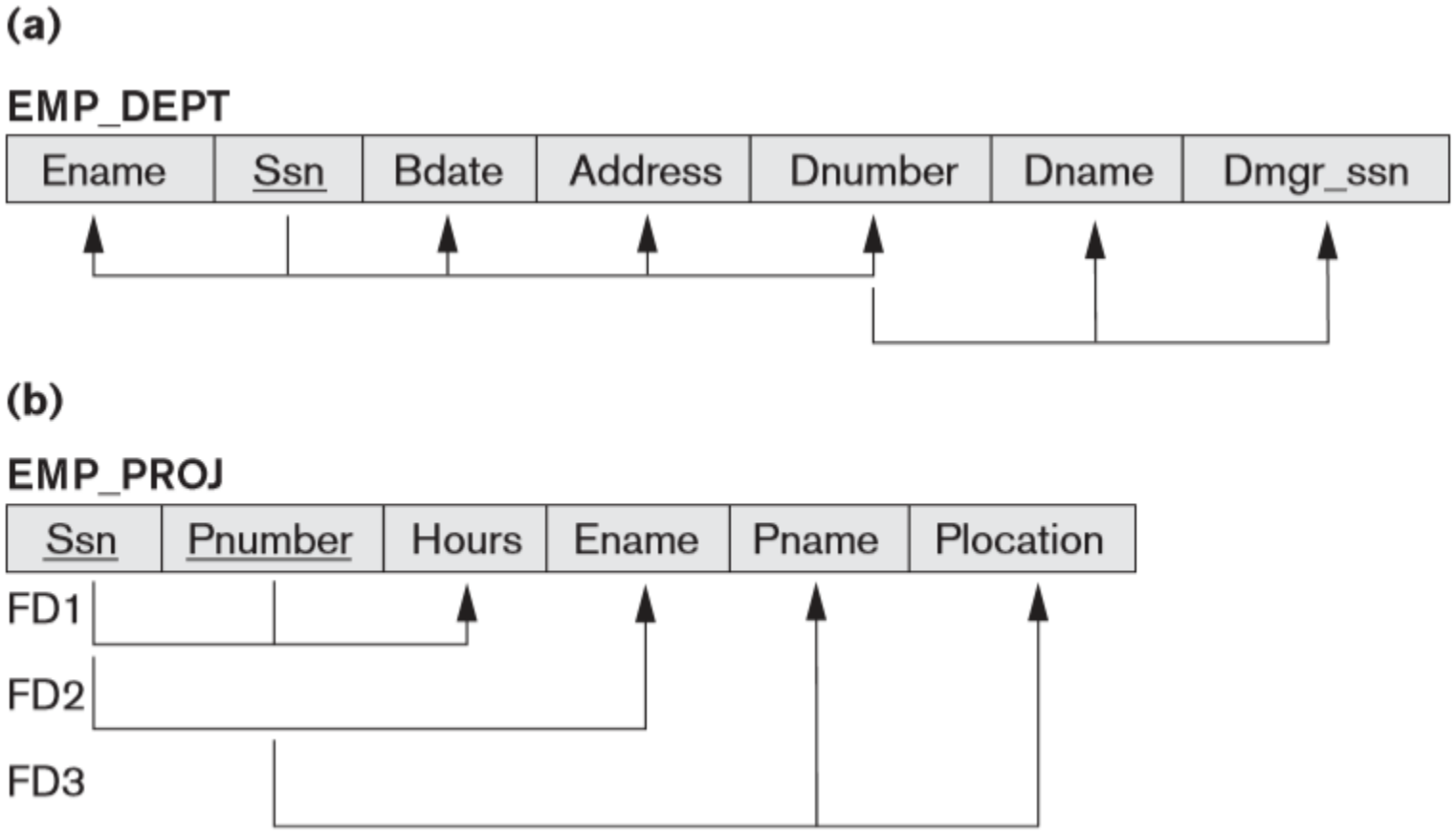
- A functional dependency is a *property of the relation schema* R, not of a particular legal relation state r of R.
- Therefore, an FD *cannot* be inferred automatically from a given relation extension r but must be defined explicitly by someone who knows the semantics of the attributes of R.

- TEXT → COURSE ?
- TEACHER → COURSE ?

TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

- **Diagrammatic notation** for displaying FDs



- We denote by F the set of functional dependencies that are specified on relation schema R.
- Typically, the schema designer specifies the functional dependencies that are *semantically obvious*; usually, however, numerous other functional dependencies hold in *all* legal relation instances among sets of attributes that can be derived from and satisfy the dependencies in F.
- Those other dependencies can be *inferred* or *deduced* from the FDs in F.