Normalization & Functional Dependencies

Some questions during last 3 weeks

- Does a primary key have to the unique?
- Is a primary key only numeric?
- Can not key attributes repeat in a relation?
- Functional Dependencies?
- Anomalies during insertion, update & deletion?
- Why breaking **salutation** apart?

You should know the answer, as it is Databases I material!!!

A simplified COMPANY Relational Database Schema

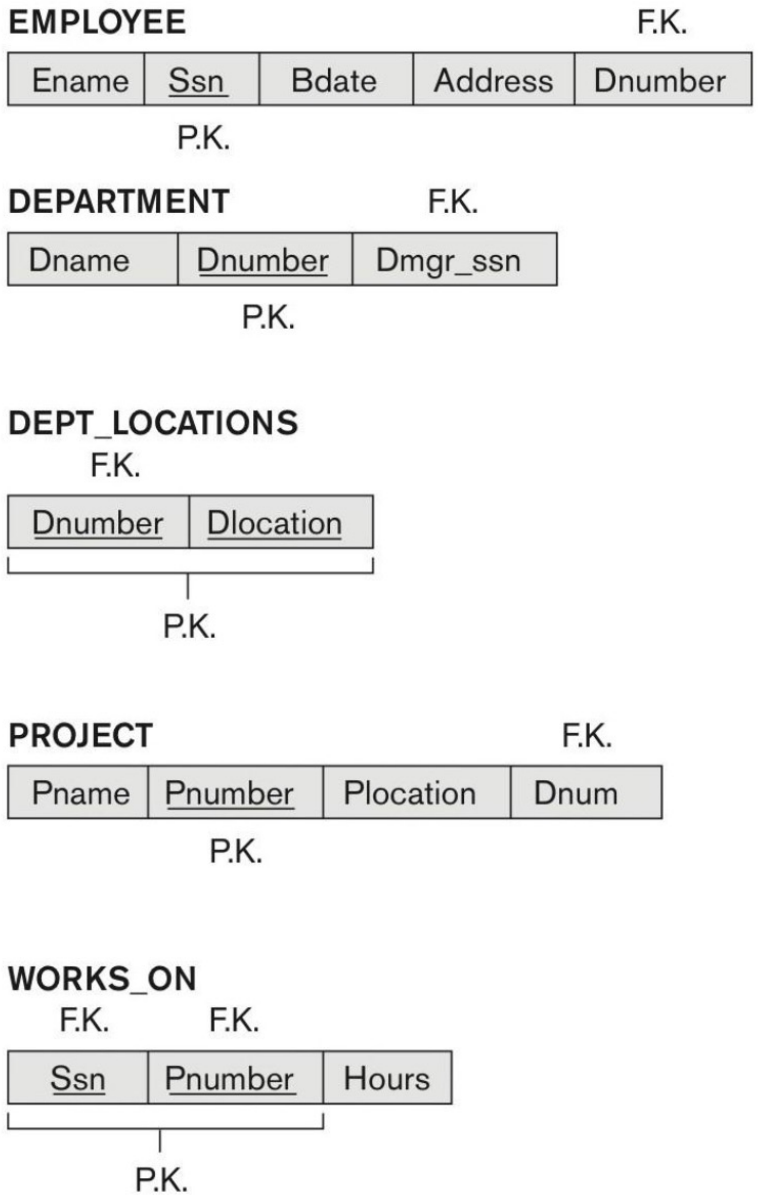


Figure 14.1 A simplified COMPANY relational database schema.

When information is stored redundantly

- Information is stored redundantly
  - Wastes storage
  - Causes problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

Update Anomaly

- Consider the relation:
  - » EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Update Anomaly:
  - » Changing the name of project number P1 from “Billing” to “Customer-Accounting” may cause this update to be made for all 100 employees working on project P1.

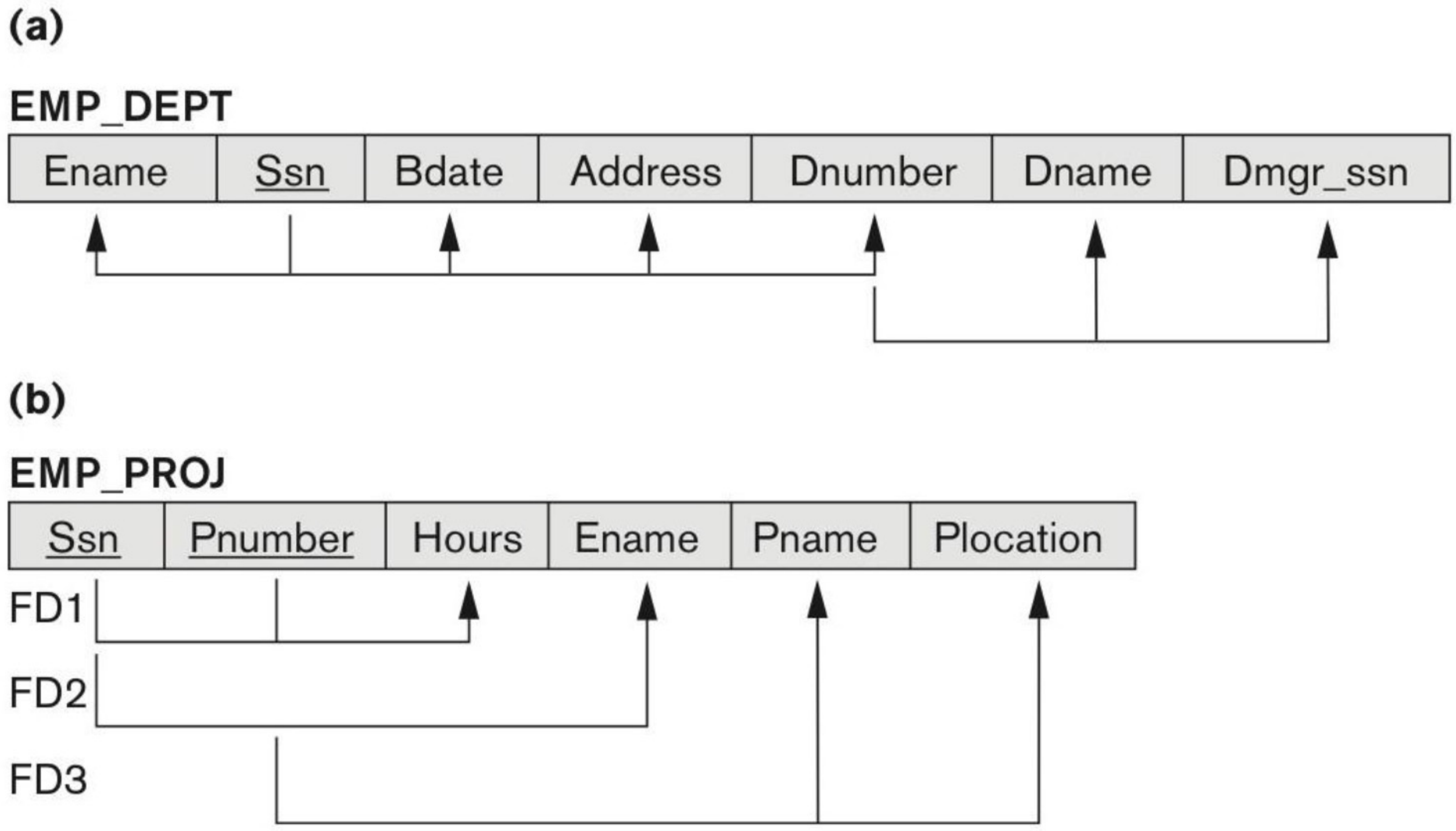
Insert Anomaly

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Insert Anomaly:
  - Cannot insert a project unless an employee is assigned to it.
- Conversely
  - Cannot insert an employee unless an he/she is assigned to a project.

Delete Anomaly

- Consider the relation:
  - EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)
- Delete Anomaly:
  - When a project is deleted, it will result in deleting all the employees who work on that project.
  - Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

Two relation schemas suffering from update anomalies



DB of Customer Orders

EmployeeID	EmployeeName	ManagerID	DepartmentID
1	John Smith	3	1
2	Jane Doe	3	1
3	Bob Johnson	4	2
4	Mary Williams	NULL	2

Primary Key: EmployeeID
Foreign Keys: ManagerID, DepartmentID

- ManagerID is dependent on the EmployeeID, because it determines the employee's manager.
- DepartmentID is dependent on the ManagerID, because it determines the department the employee works in.
- This table is in 2NF: it is in 1NF and does not have any partial dependencies, but it is **not** in 3NF

DepartmentID is dependent on ManagerID, which is **not** the Primary Key.

DB of Customer Orders

- To eliminate this **transitive dependency**, we can create a *separate table* for departments and store the DepartmentID and DepartmentName in that table.
- We can then update the employees table to store the DepartmentID as a *foreign key*.

EmployeeID	EmployeeName	ManagerID	DepartmentID
1	John Smith	3	1
2	Jane Doe	3	1
3	Bob Johnson	4	2
4	Mary Williams	NULL	2

DepartmentID	DepartmentName
1	Sales
2	Marketing

Functional Dependencies

- A set of attributes X functionally determines a set of attributes Y, if the value of X determines a unique value for Y.
- Written as $X \rightarrow Y$

  - $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they *must have* the same value for Y
    - For any two tuples t1 and t2 in any relation instance r(R): If $t1[X]=t2[X]$, *then* $t1[Y]=t2[Y]$
  - $X \rightarrow Y$ in R specifies a *constraint* on all relation instances r(R)
  - FDs are derived from the real-world constraints on the attributes

Inference Rules for FDs (1/3)

- **Definition:** An FD $X \rightarrow Y$ is **inferred from** or **implied by** a set of dependencies F specified on R if $X \rightarrow Y$ holds in *every* legal relation state r of R; that is, whenever r satisfies all the dependencies in F, $X \rightarrow Y$ also holds in r.

- Given a set of FDs F, we can **infer** additional FDs that hold whenever the FDs in F hold

Inference Rules for FDs (2/3)

- Armstrong's inference rules:
  - IR1. (**Reflexive**) If Y *subset-of* X, then X → Y
  - IR2. (**Augmentation**) If X → Y, then XZ → YZ
    - (Notation: XZ stands for X U Z)
  - IR3. (**Transitive**) If X → Y and Y → Z, then X → Z

- IR1, IR2, IR3 form a **sound** and **complete** set of inference rules
  - These are rules hold and all other rules that hold can be deduced from these

Inference Rules for FDs (3/3)

- Some additional inference rules that are useful:
  - **Decomposition:** If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
  - **Union:** If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
  - **Psuedotransitivity:** If $X \rightarrow Y$ and $WY \rightarrow Z$, then $WX \rightarrow Z$

Completeness Property

- Last 3 inference rules and any other inference rules, can be deduced from IR1, IR2, and IR3

Closure

- **Closure** of a set F of FDs is the set F^+ of all FDs that can be inferred from F

- **Closure** of a set of attributes X with respect to F is the set X^+ of all attributes that are functionally determined by X

- X^+ can be calculated by repeatedly applying IR1, IR2, IR3 using the FDs in F