IN 1400 - Fundamentals of Databases and Database Design

DATABASE ANALYSIS AND DESIGN 7

Week 9

Functional Dependencies and Normalization for Relational Databases

Outline

- Informal Design Guidelines for Relational Databases

- 3 Normal Forms Based on Primary Keys

Informal Design Guidelines for Relational Databases

- What is relational database design?
  - The grouping of attributes to form "good" relation schemas

- Two levels of relation schemas
  - The logical (or conceptual) level
  - The storage (or implementation) level – only applies to base relations

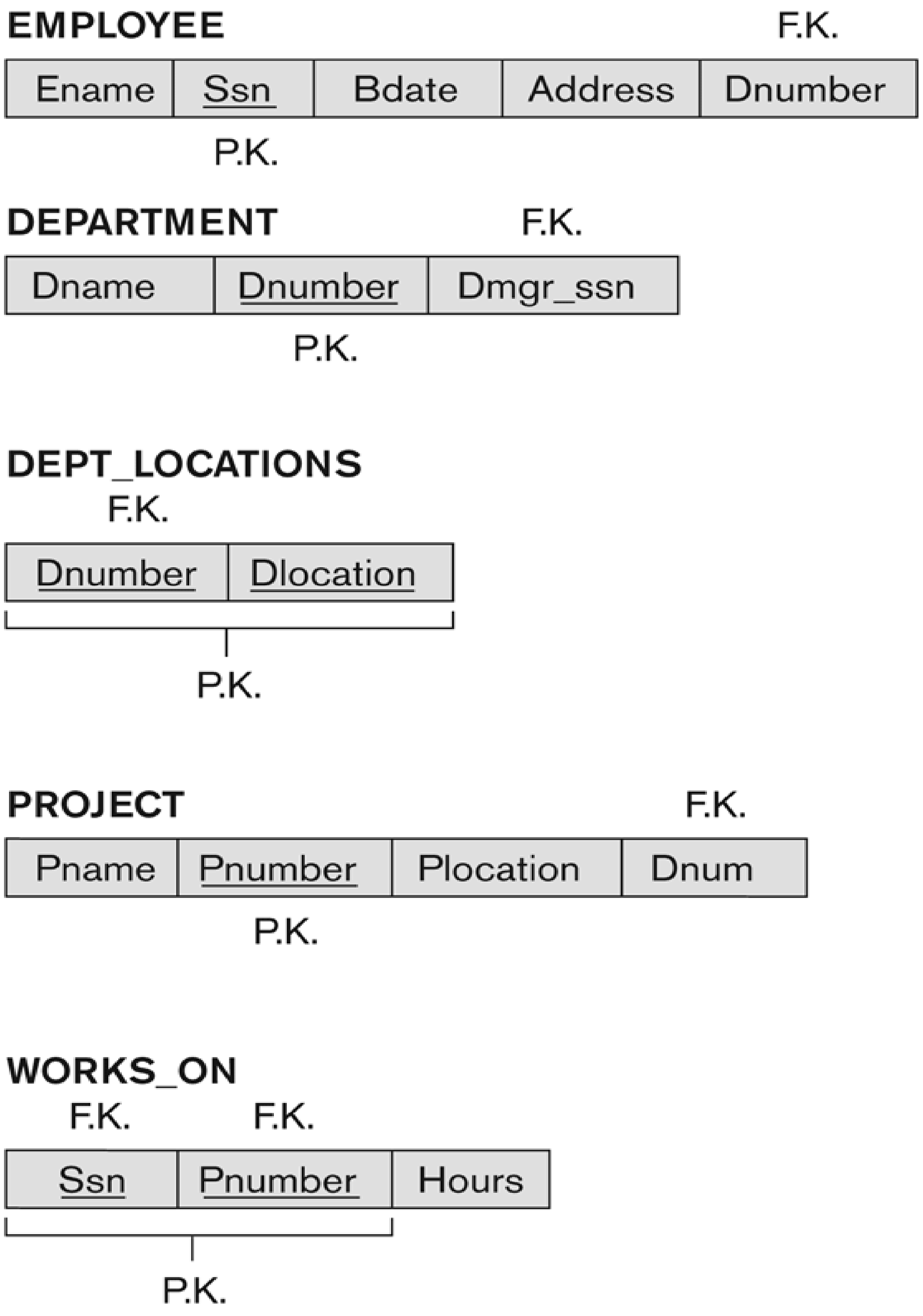
- What are the criteria for "good" base relations?

Semantics of the Relation Attributes

GUIDELINE 1:

- Informally, each tuple in a relation should represent one entity or relationship instance (Applies to individual relations and their attributes).
  - Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation.
  - Only foreign keys should be used to refer to other entities.
  - Entity and relationship attributes should be kept apart as much as possible.

A simplified COMPANY relational database schema



EMPLOYEE

				F.K.
Ename	Ssn	Bdate	Address	Dnumber
	P.K.			

DEPARTMENT

		F.K.
Dname	Dnumber	Dmgr_ssn
	P.K.	

DEPT_LOCATIONS

F.K.	
Dnumber	Dlocation

P.K.

PROJECT

			F.K.
Pname	Pnumber	Plocation	Dnum
	P.K.		

WORKS_ON

F.K.	F.K.	
Ssn	Pnumber	Hours

P.K.

Redundant Information in Tuples and Update Anomalies

If Information is stored redundantly

- Wastes storage
- Causes problems with update anomalies
  - Insertion anomalies
  - Deletion anomalies
  - Modification anomalies

- **Goal of schema design** -> Minimize storage space used by base relations

Example of an Modification Anomaly

- Consider the relation:
  - EMP_PROJ(EmpNo, ProjNo, Ename, Pname, No_hours)

- Modification 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.

Example of an Insert Anomaly

- Consider the relation:
  - EMP_PROJ(EmpNo, ProjNo, Ename, Pname, No_hours)

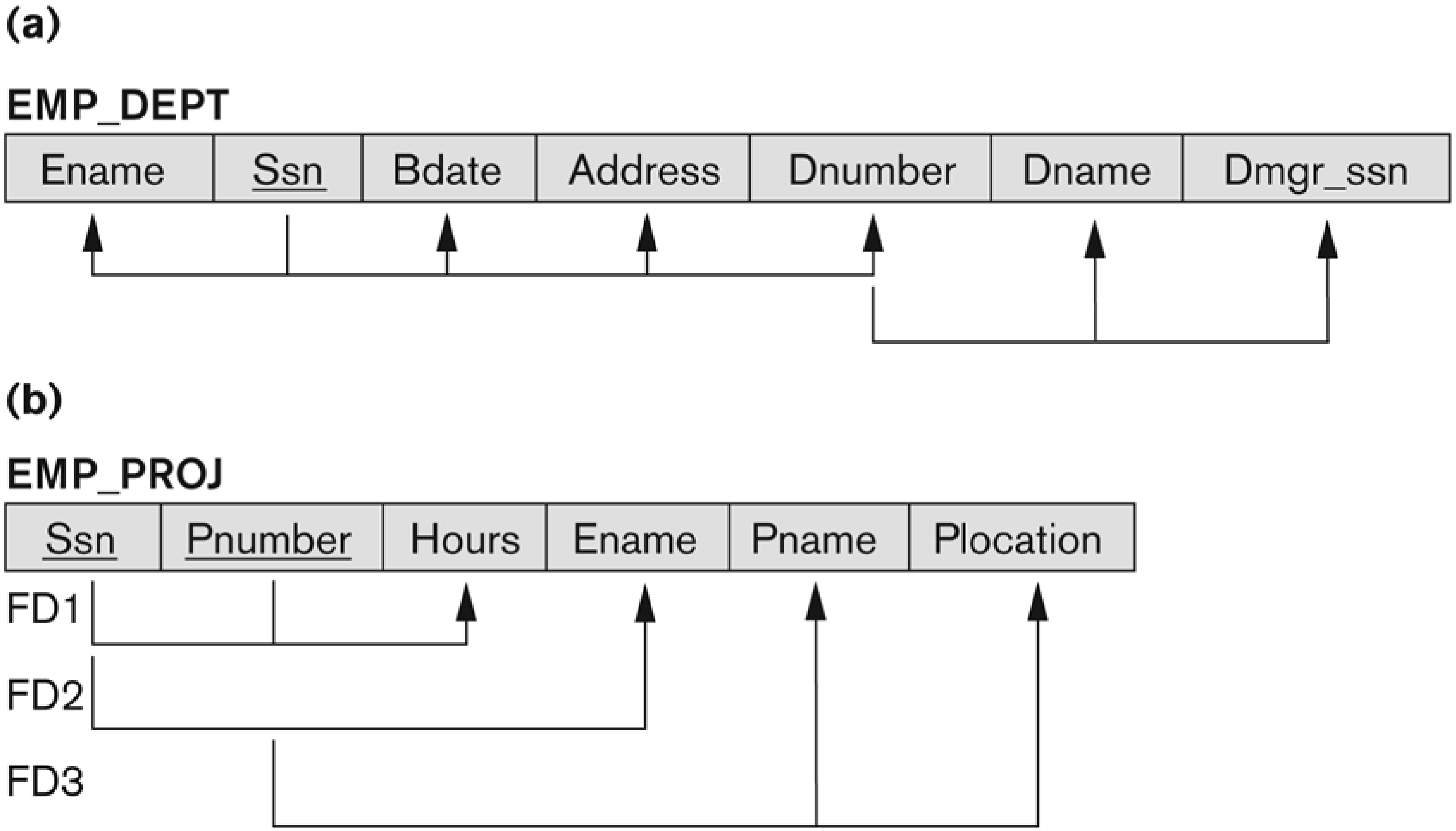
- Insert Anomaly:
  - Cannot insert a project unless an employee is assigned to it.
- Conversely
  - Cannot insert an employee unless an he/she is assigned to a project.

Example of a Delete Anomaly

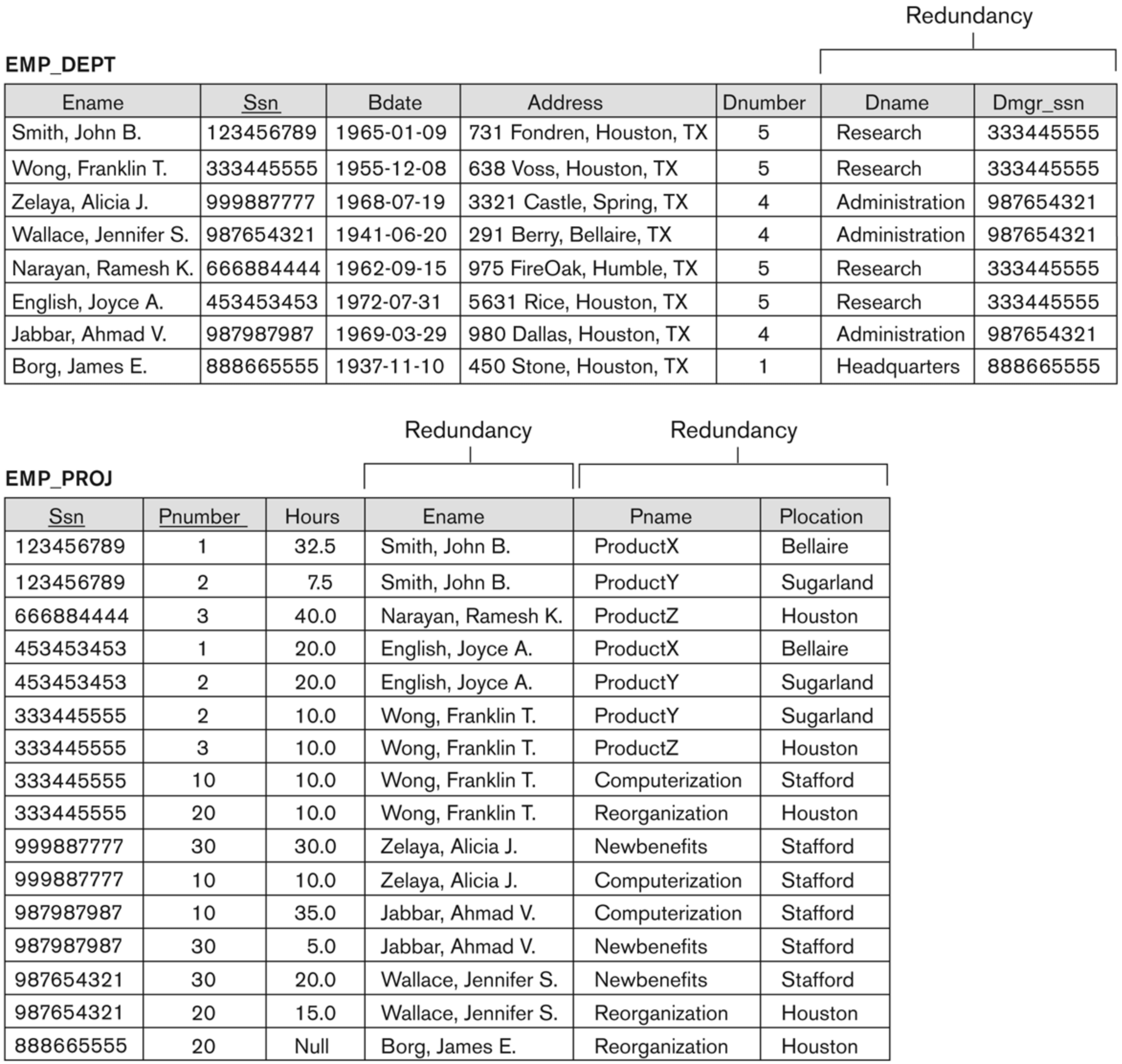
- Consider the relation:
  - EMP_PROJ(EmpNo, ProjNo, Ename, Pname, No_hours)

- 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



Example States for EMP_DEPT and EMP_PROJ



Redundancy

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

Redundancy

Redundancy

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston

Guideline to Redundant Information in Tuples and Update Anomalies

GUIDELINE 2:

- Design a schema that does not suffer from the insertion, deletion and modification (update) anomalies.
- If there are any anomalies present, then note them so that applications can be made to take them into account.

Null Values in Tuples

GUIDELINE 3:

- Relations should be designed such that their tuples will have as few NULL values as possible

- Attributes that are NULL frequently could be placed in separate relations (with the primary key)

Normalization of Relations

- **Normalization:**
  - The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

- **Normal form:**
  - Condition using keys and functional dependencies of a relation to certify whether a relation schema is in a particular normal form

- 3 Normal Forms based on Primary Key
  - First Normal Form – 1NF
  - Second Normal Form – 2NF
  - Third Normal Form - 3NF

- In addition
  - BCNF
  - Fourth Normal Form

Practical Use of Normal Forms

- **Normalization** is carried out in practice so that the resulting designs are of high quality and meet the desirable properties

- The practical utility of these normal forms becomes questionable when the constraints on which they are based are *hard to understand* or to *detect*

- The database designers *need not* normalize to the highest possible normal form
  - Usually up to 3NF, BCNF or 4NF

- **Denormalization**:
  - The process of storing the join of higher normal form relations as a base relation—which is in a lower normal form

First Normal Form

- This rule defines that all the attributes in a relation must have atomic domains (values from indivisible units).
- It is defined in the definition of relations (tables) itself.
- It **disallows**
  - Composite attributes
  - Multivalued attributes
  - **Nested relations**; attributes whose values for an *individual tuple* are non-atomic

Each attribute must contain only a single value

Student Relation

Student	Age	Subject
Peter	15	Physics, Maths
Alex	14	Maths
James	16	Maths

Any row must not have a column with more than one value saved.
We must separate such data into multiple rows.

1 NF

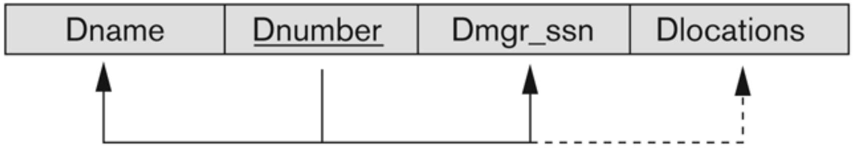
Student	Age	Subject
Peter	15	Physics
Peter	15	Maths
Alex	14	Maths
James	16	Maths

Data redundancy increases, as there will be many columns with same data in multiple rows.
But each row as a whole will be unique.

Normalization into 1NF

(a)

DEPARTMENT



A relation schema not in 1NF

(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

Example state of Relation DEPARTMENT

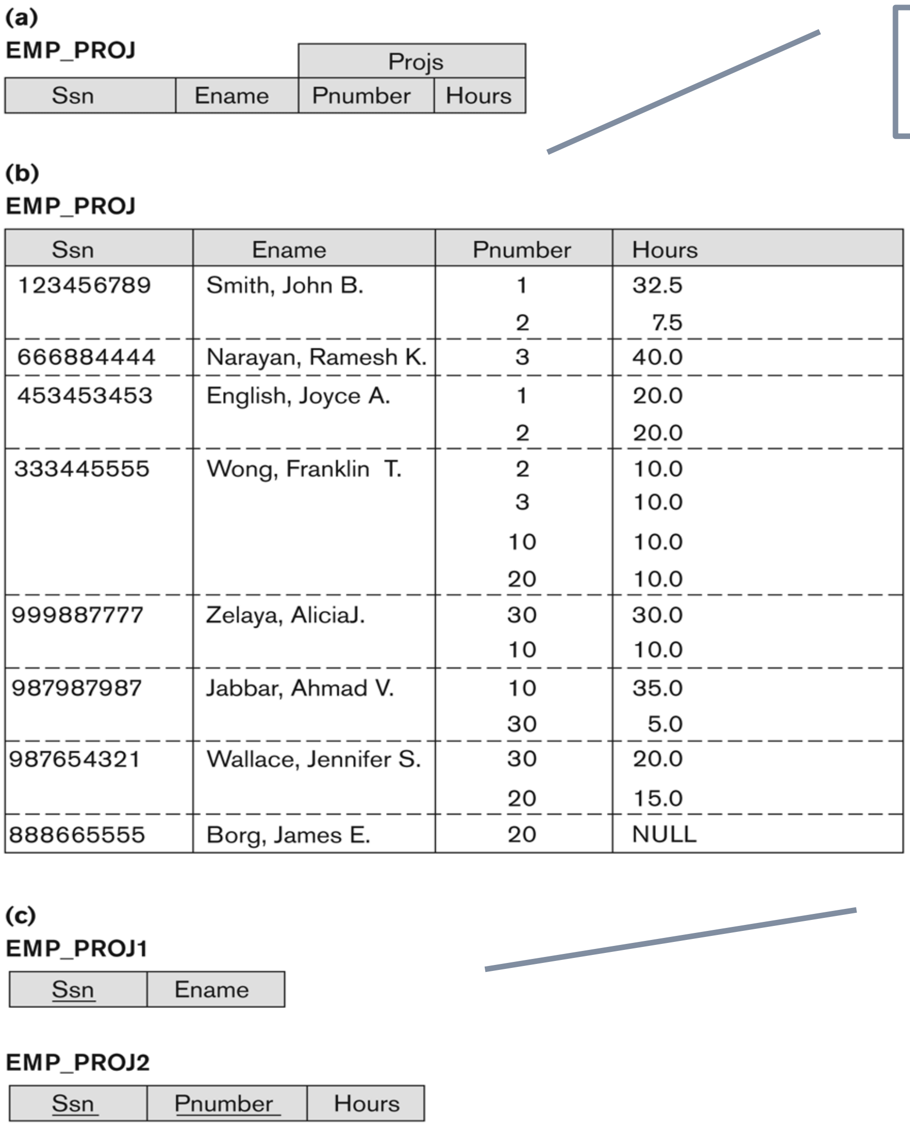
(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

1NF version of the same relation with redundancy

Normalization nested relations into 1NF



(a)

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

(b)

EMP_PROJ

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1 2	32.5 7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1 2	20.0 20.0
333445555	Wong, Franklin T.	2 3 10 20	10.0 10.0 10.0 10.0
999887777	Zelaya, AliciaJ.	30 10	30.0 10.0
987987987	Jabbar, Ahmad V.	10 30	35.0 5.0
987654321	Wallace, Jennifer S.	30 20	20.0 15.0
888665555	Borg, James E.	20	NULL

(c)

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours

Schema of the EMP_PROJ relation with a nested relation attribute Projs

Example extension of the EMP_PROJ relation showing nested relations within each tuple

Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key

Functional dependency (FD)

- FD is a set of constraints between two attributes in a relation.
- It says if two tuples have same values for attributes X_1, X_2,..., X_n, then those two tuples must have to have same values for attributes Y_1, Y_2, ..., Y_n.
  - For any two tuples t_1 and t_2 in any relation instance r(R): If $t_1[X]=t_2[X]$, *then* $t_1[Y]=t_2[Y]$
- Represented by an arrow sign (→)
- X→Y, where X functionally determines Y.
  - The left-hand side attributes determine the values of attributes on the right-hand side.

Functional Dependency (FD)

- 2NF and 3NF are defined in terms of functional dependencies
- FDs only exist when there are unique identifiers
- If K is a key of relation R, then K functionally determines all attributes in R

Examples

- Social security number determines employee name
  - `SSN -> ENAME`
- Project number determines project name and location
  - `PNUMBER -> {PNAME, PLOCATION}`
- Employee ssn and project number determines the hours per week that the employee works on the project
  - `{SSN, PNUMBER} -> HOURS`

Second Normal Form

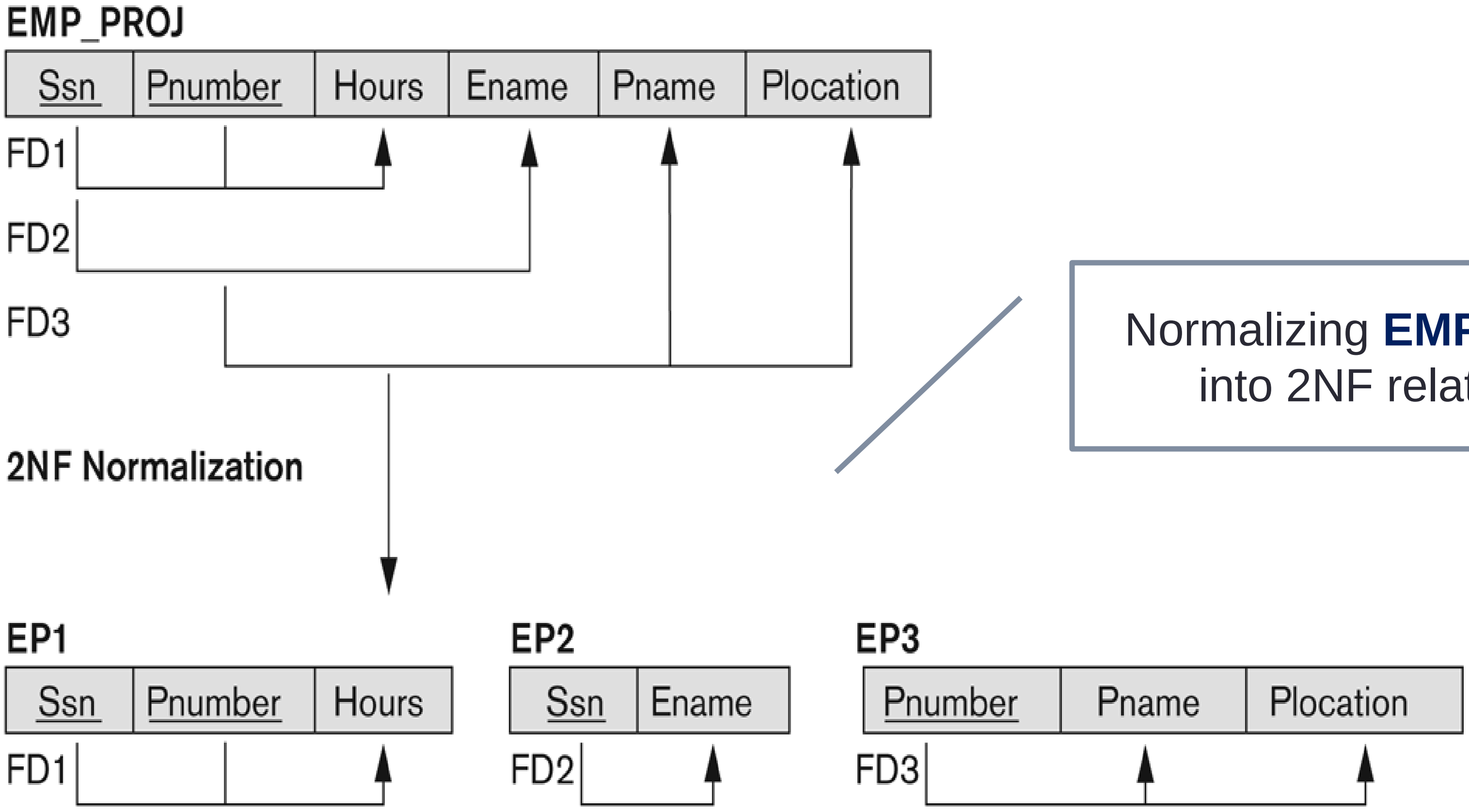
- Definitions
  - **Prime attribute:** An attribute that is member of the primary key K
  - **Full functional dependency:** a FD Y -> Z where removal of any attribute from Y means the FD does not hold any more

- Every non-prime attribute A in R is fully functionally dependent on the primary key
  - There must not be any partial dependency of any attribute on primary key.
  - For a relation that has composite (primary) key, each non-prime attribute must depend upon the entire composite key for its existence.

- Examples:
  - `{SSN, PNUMBER} -> HOURS` is a full FD since neither `SSN -> HOURS` nor `PNUMBER -> HOURS` hold

  - `{SSN, PNUMBER} -> ENAME` is not a full FD (it is called a partial dependency) since `SSN -> ENAME` also holds

Normalizing into 2NF



Normalizing **EMP_PROJ** into 2NF relations

Student_subject relation

SId	Student	Age	SubCode	Subject	SubCredit	Grade
A002	Peter	15	S01	Physics	3	A
A002	Peter	15	S02	Maths	2.5	A
A010	Alex	14	S02	Maths	2.5	B
A021	James	16	S02	Maths	2.5	C

2 NF

SId	Student	Age
A002	Peter	15
A010	Alex	14
A021	James	16

SId	SubCode	Grade
A002	S01	A
A002	S02	A
A010	S02	B
A21	S02	C

SubCode	Subject	SubCredit
S01	Physics	3
S02	Maths	2.5

Third Normal Form

Definition:

- **Transitive functional dependency:** a functional dependency (FD) X -> Z that can be derived from two FDs X -> Y and Y -> Z
- **Trivial functional dependency:** If a functional dependency (FD) X → Y holds, where Y is a subset of X, then it is called a trivial FD. Trivial FDs always hold.

- Examples:
  - `ENO-> DMGRENO` is a **transitive** FD
    - Since `ENO -> DNUMBER` and `DNUMBER -> DMGRENO` hold
  - `ENO-> ENAME` is **non-transitive**
    - Since there is no set of attributes `X` where `ENO-> X` and `X -> ENAME`

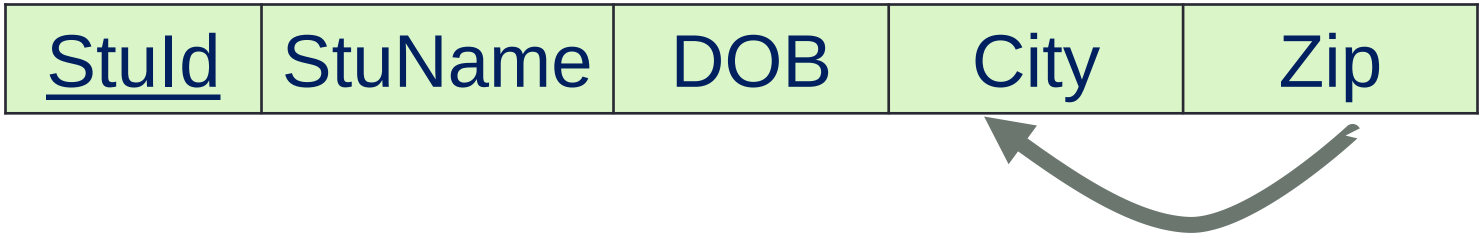
Third Normal Form

- A relation schema R is in **third normal form (3NF)** if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key

- 3NF Satisfies
  - No non-prime attribute is transitively dependent on prime key attribute.
  - For any non-trivial functional dependency, X → A, then either −
    - X is a superkey or,
    - A is prime attribute.

- NOTE:
  - In X -> Y and Y -> Z, with X as the primary key, we consider this a problem only if Y is not a candidate key.
  - When Y is a candidate key, there is no problem with the transitive dependency.
  - E.g., Consider `EMP (NIC, EmpNo, Salary)`.
    - Here, `NIC-> EmpNo -> Salary` and `EmpNo` is a candidate key.

Student_detail relation



StuId	StuName	DOB	City	Zip

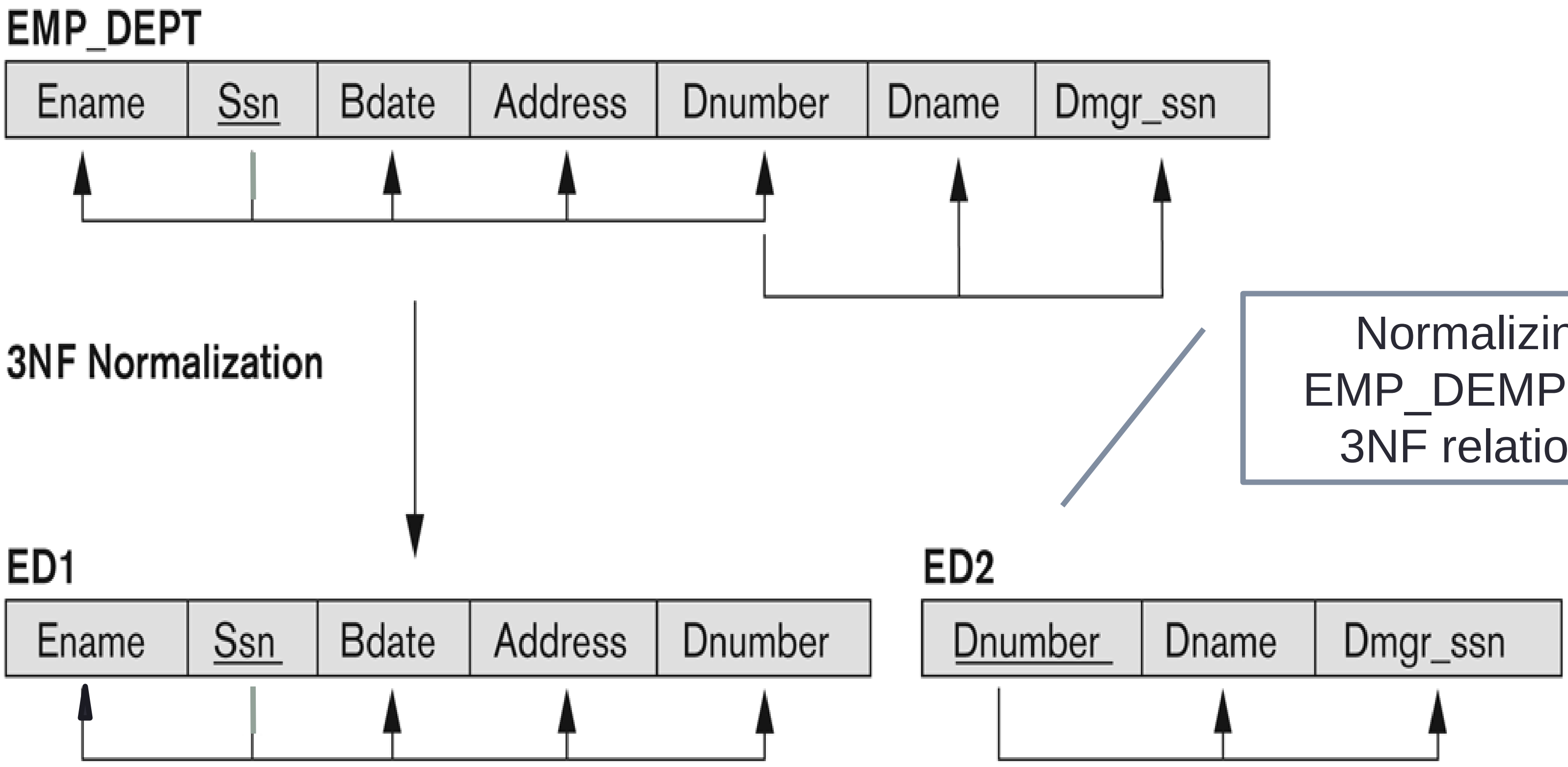
- StuId is the key and only prime key attribute.
- City can be identified by StuId as well as by Zip itself.
- Neither Zip is a superkey nor is City a prime attribute.
- Additionally, StuId → Zip → City, so there exists **transitive dependency**.

3NF

StuId	StuName	DOB	Zip

Zip	City

Normalizing into 3NF



EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn

3NF Normalization

ED1

Ename	Ssn	Bdate	Address	Dnumber

ED2

Dnumber	Dname	Dmgr_ssn

Normalizing EMP_DEMP into 3NF relations

Normal Forms Defined Informally…

- 1st normal form
    - All attributes depend on **the key**

- 2nd normal form
    - All attributes depend on **the whole key**

- 3rd normal form
    - All attributes depend on **nothing but the key**

Normalize to 3NF

Task

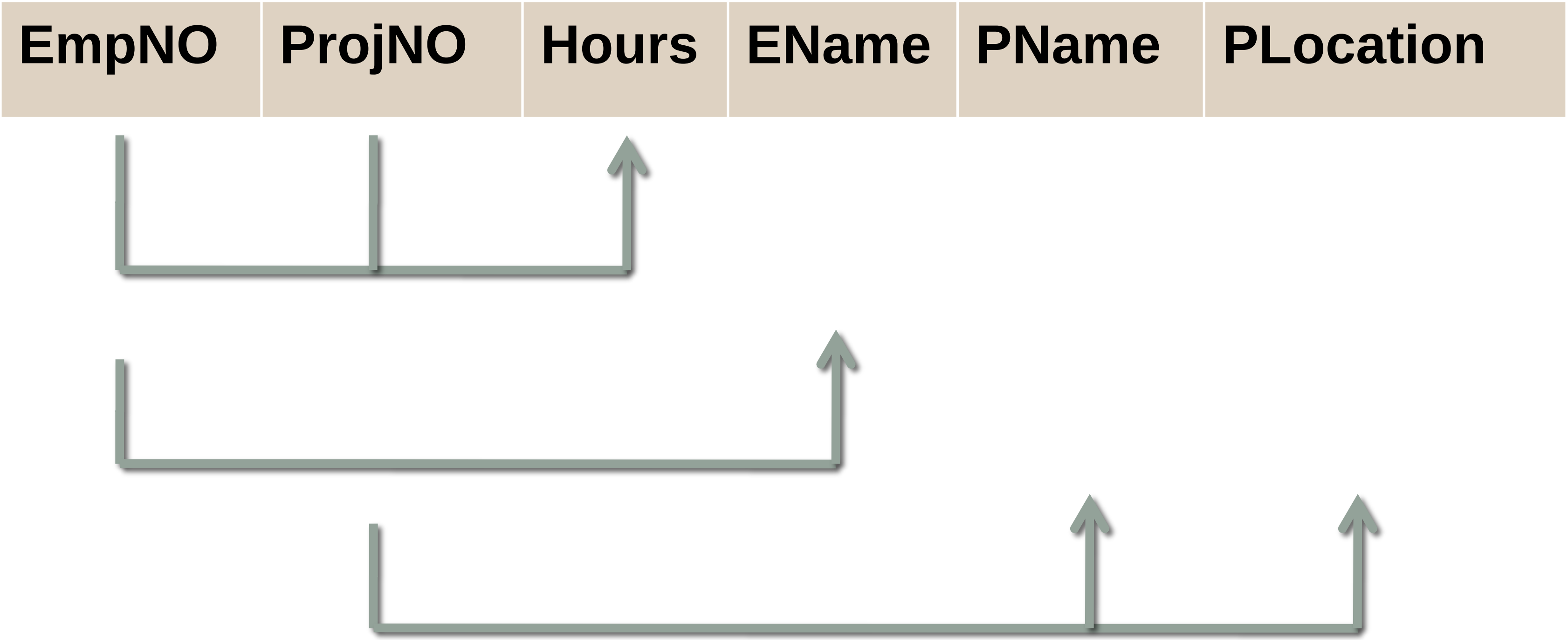
Employee_Project

EmpNO	ProjNO	Hours	EName	PName	PLocation

Student_Faculty

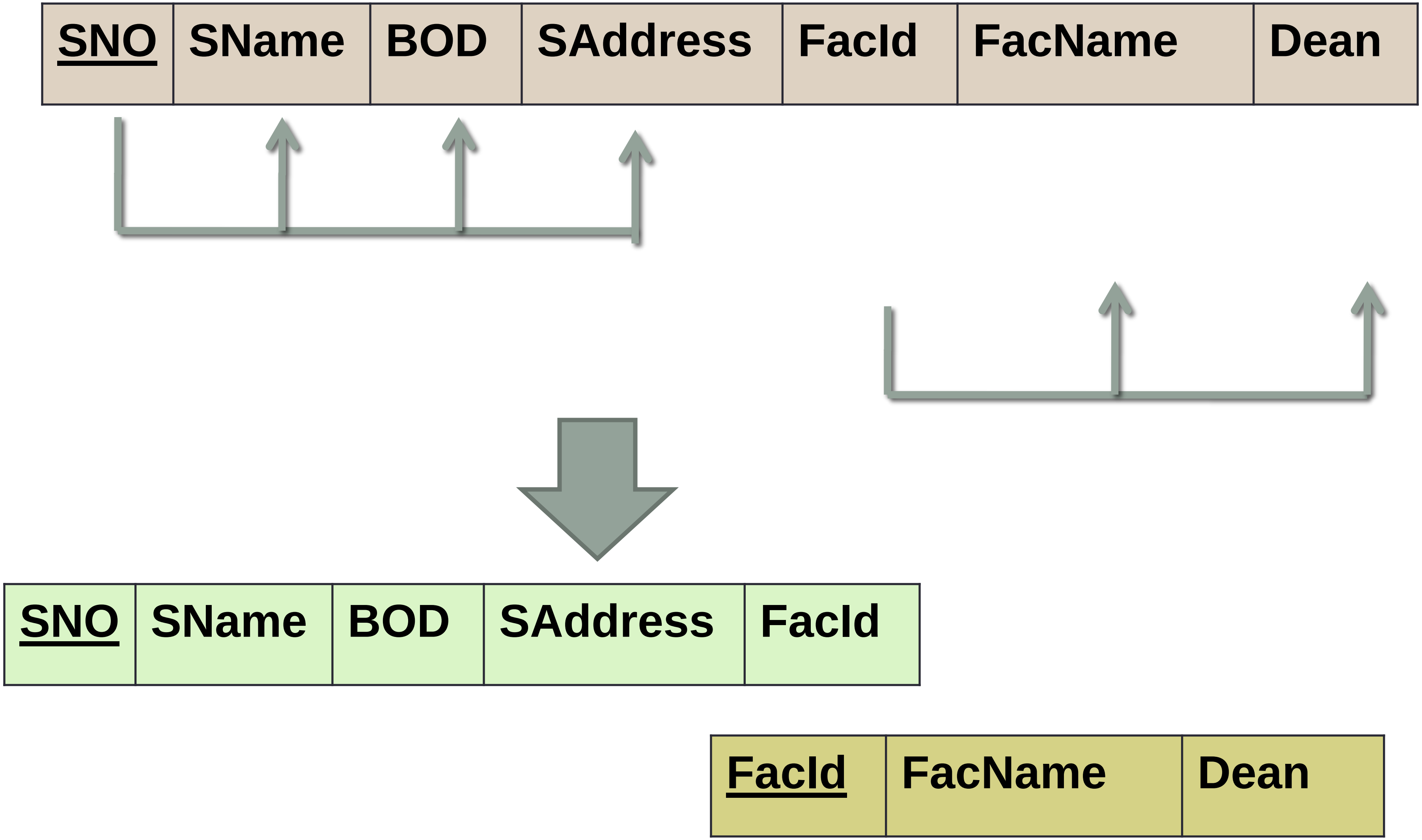
SNO	SName	BOD	SAddress	FacId	FacName	Dean

Employee_Project



EmpNO	ProjNO	Hours	EName	PName	PLocation

Student_Faculty



SNO	SName	BOD	SAddress	FacId	FacName	Dean

SNO	SName	BOD	SAddress	FacId

FacId	FacName	Dean