Functional Dependencies and Normalization for Relational Databases

Informal Design Guidelines for Relational Databases

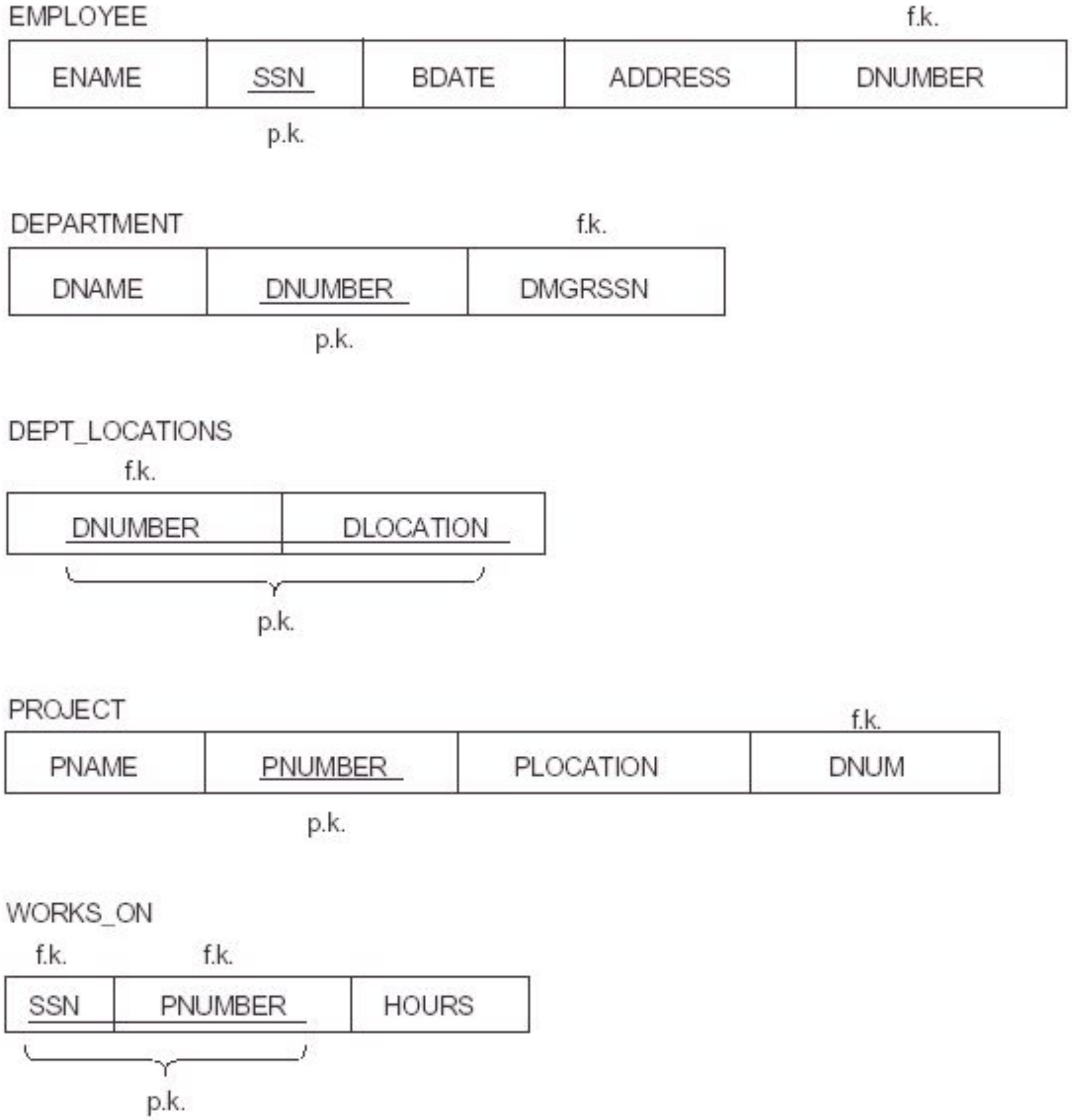
- Relational database design: The grouping of attributes to form "good" relation schemas
- Two levels of relation schemas:
  - The logical "user view" level
  - The storage "base relation" level
- Design is concerned mainly with base relations
- Criteria for "good" base relations:
  - Discuss informal guidelines for good relational design
  - Discuss formal concepts of functional dependencies and normal forms 1NF 2NF 3NF BCNF

Semantics of the Relation Attributes

- Each tuple in a relation should represent one entity or relationship instance
  - Only foreign keys should be used to refer to other entities
  - Entity and relationship attributes should be kept apart as much as possible
  - Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.

  - Tip: do not mix attributes from multiple entities

Figure 14.1 Simplified version of the COMPANY relational database schema.



EMPLOYEE

f.k.

ENAME	SSN	BDATE	ADDRESS	DNUMBER

p.k.

DEPARTMENT

f.k.

DNAME	DNUMBER	DMGRSSN

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.

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Remesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

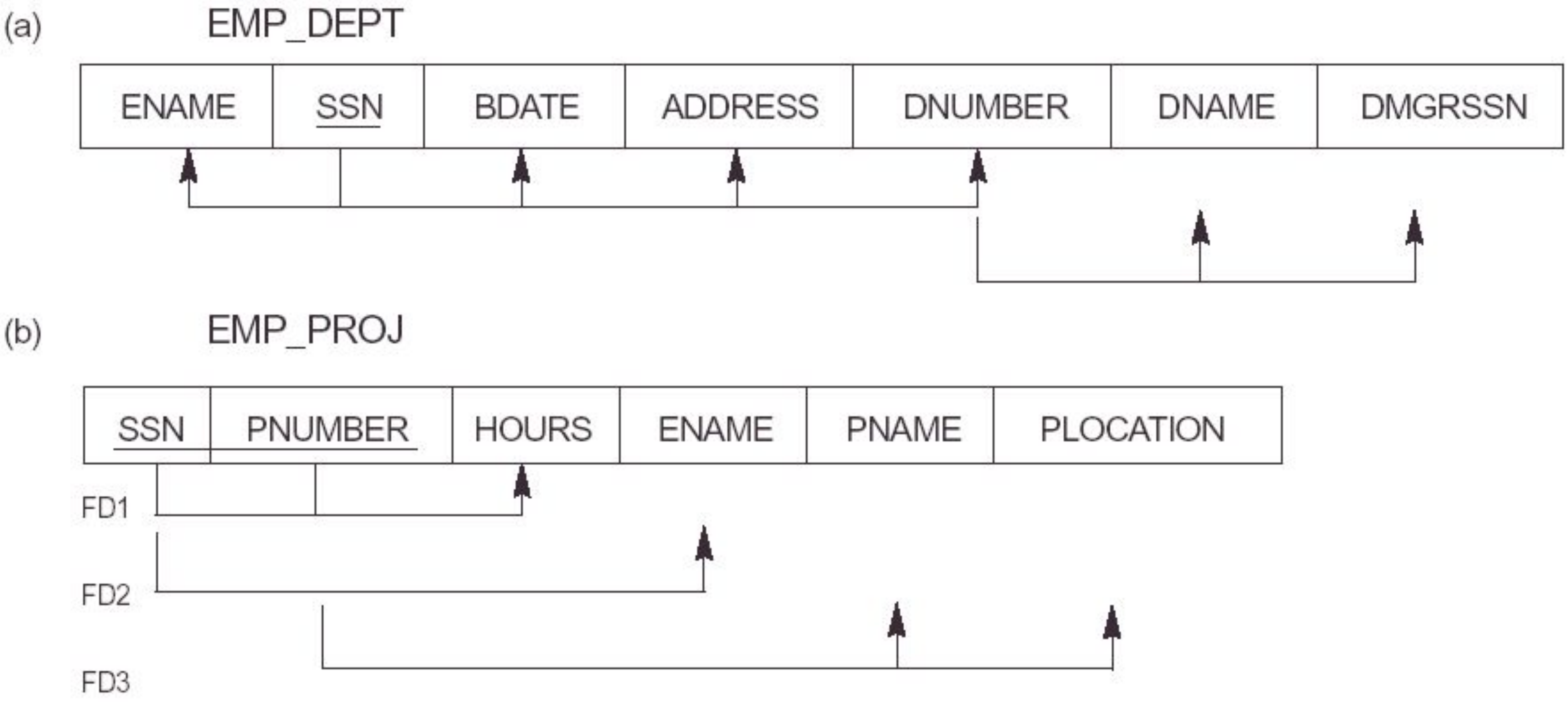
SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Redundant Information in Tuples and Update Anomalies

- Mixing attributes of multiple entities may cause problems
  - Information is stored redundantly wasting storage
  - Problems with update anomalies:
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies



(a) EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

(b) EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION

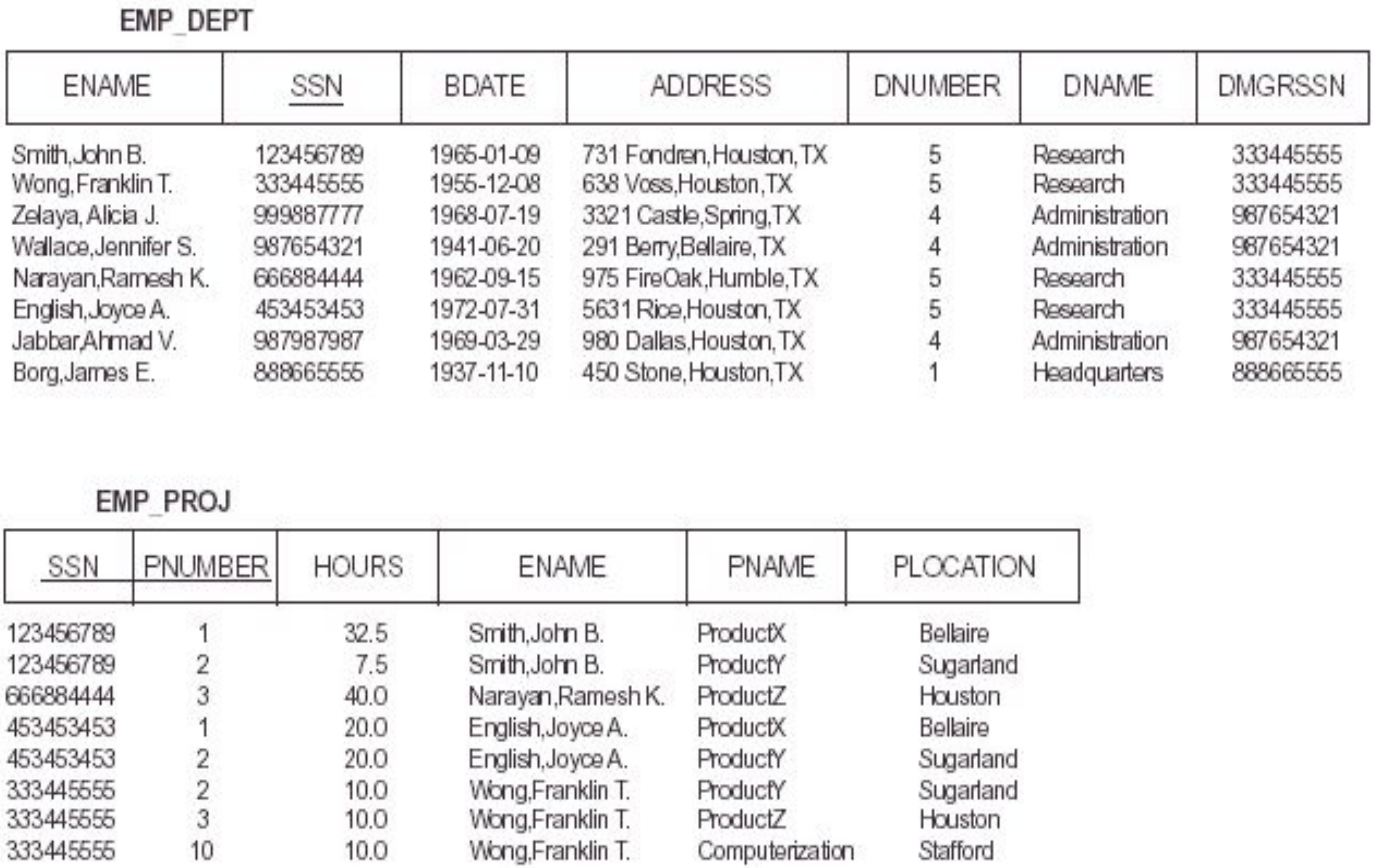
FD1

FD2

FD3

(a) Is combined details of employee and department entities

(b) Is combined details of employee and project entities



EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
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

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

EXAMPLE OF AN UPDATE ANOMALY

Consider the relation:

EMP_PROJ (Emp#, Proj#, Ename, Pname, No_hours)

- **Update Anomaly**
  - Changing the name of project number 1 from “ProductY” to “Customer-Accounting” may cause this update to be made for all 3 employees working on project 1
- **Insert Anomaly**
  - Cannot insert a project unless an employee is assigned to .
  - Inversely- Cannot insert an employee unless he/she is assigned to a project.

EXAMPLE OF AN UPDATE ANOMALY (2)

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

Null Values in Tuples

- 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)
  - Reasons for nulls:
  - a. attribute not applicable or invalid
  - b. attribute value unkown (may exist)
  - c. value known to exist, but unavailable

Spurious Tuples

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations
- The relations should be designed to satisfy the lossless join condition. No spurious tuples should be generated by doing a natural-join of any relations

Informal Guidelines

Guideline 1:

- Informally, each tuple in a relation should represent one entity or relationship instance. (Applies to individual relations and their attributes).

Guideline 2:

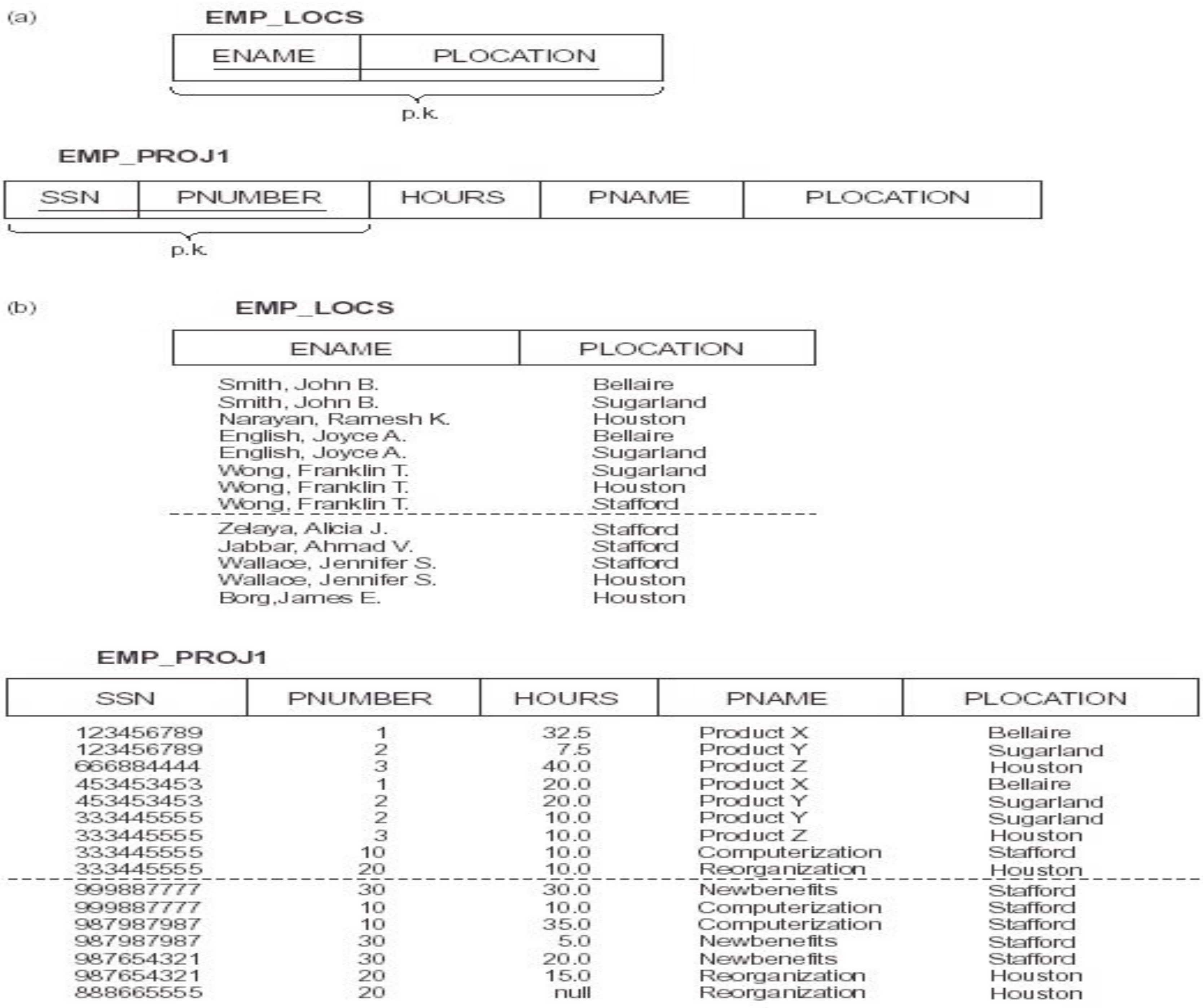
- Design a schema that does not suffer from the insertion, deletion and update anomalies.
- If there are any present, then note them so that applications can be made to take them into account

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)

Guideline 4:

- The relations should be designed to satisfy the lossless join condition.
- No spurious tuples should be generated by doing a natural-join of any relations



(a)

EMP_LOCS

ENAME	PLOCATION

p.k.

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION

p.k.

(b)

EMP_LOCS

ENAME	PLOCATION
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg,James E.	Houston

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION
123456789	1	32.5	Product X	Bellaire
123456789	2	7.5	Product Y	Sugarland
666884444	3	40.0	Product Z	Houston
453453453	1	20.0	Product X	Bellaire
453453453	2	20.0	Product Y	Sugarland
333445555	2	10.0	Product Y	Sugarland
333445555	3	10.0	Product Z	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	null	Reorganization	Houston

Functional Dependencies

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations
- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes

Functional Dependencies (2)

- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y
- X □Y holds if whenever two tuples have the same value for X, they *must have* the same value for Y

  If t1[X]=t2[X], *then* t1[Y]=t2[Y] in any relation instance r(R)
- X □ Y in R specifies a *constraint* on all relation instances r(R)
- FDs are derived from the real-world constraints on the attributes

Examples of FD constraints

- 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

In EMPLOYEE relation given in Table 1,

- FD **E-ID->E-NAME** holds because for each E-ID, there is a unique value of E-NAME.
- FD **E-ID->E-CITY** and **E-CITY->E-STATE** also holds.
- FD E-NAME->E-ID **does not hold** because E-NAME ‘John’ is not uniquely determining E-ID. There are 2 E-IDs corresponding to John (E001 and E003).

EMPLOYEE

E-ID	**E-NAME**	**E-CITY**	**E-STATE**
E001	John	Delhi	Delhi
E002	Mary	Delhi	Delhi
E003	John	Noida	U.P.

- The FD set for EMPLOYEE relation given in Table 1 are:

{E-ID->E-NAME, E-ID->E-CITY, E-ID->E-STATE, E-CITY->E-STATE

X->Y will always hold if X ⊇ Y it is trival dependency

Eg. E-ID, E-NAME->E-ID

Else it is nontrival FD

Functional Dependencies (3)

- An FD is a property of the attributes in the schema R
- The constraint must hold on *every relation instance* r(R)
- If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples with t1[K]=t2[K])

Inference Rules for FDs

- Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold
- Armstrong's inference rules

  A1. (Reflexive) If Y subset-of X, then X □ Y

  A2. (Augmentation) If X □ Y, then XZ □ YZ

  (Notation: XZ stands for X U Z)

  A3. (Transitive) If X □ Y and Y □ Z, then X □ Z
- A1, A2, A3 form a *sound* and *complete* set of inference rules

Additional Useful Inference Rules

- Decomposition
  - If X □ YZ, then X □ Y and X □ Z
- Union
  - If X □ Y and X □ Z, then X □ YZ
- Psuedotransitivity
  - If X □ Y and WY □ Z, then WX □ Z
- **Closure** of a set F of FDs is the set F+ of all FDs that can be inferred from F

Steps to Find the Closure of Attribute

Q. Given FD set of a Relation R, The attribute closure set S be the set of A

- Add A to S.
- Recursively add attributes which can be functionally determined from attributes of the set S until done.
- From Table 1, FDs are

Given R(E-ID, E-NAME, E-CITY, E-STATE)

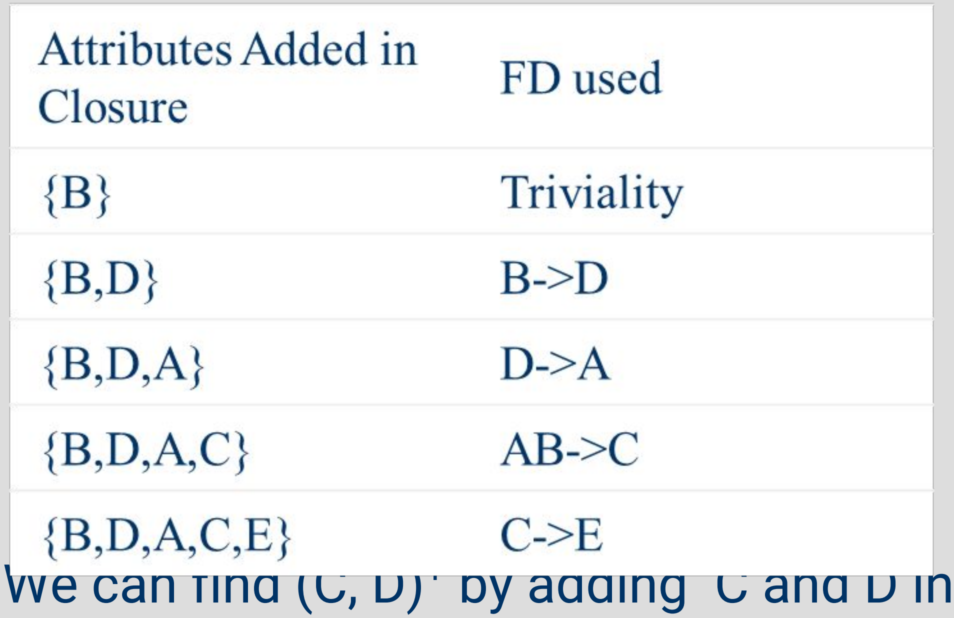
FDs = { E-ID->E-NAME, E-ID->E-CITY, E-ID->E-STATE, E-CITY->E-STATE }

The attribute closure of E-ID can be calculated as:

- Add E-ID to the set {E-ID}
- Add Attributes which can be derived from any attribute of set. In this case, E-NAME and E-CITY, E-STATE can be derived from E-ID. So these are also a part of closure.
- As there is one other attribute remaining in relation to be derived from E-ID. So result is:
- **(E-ID)$^+$ = {E-ID, E-NAME, E-CITY, E-STATE }**Similarly,
- **(E-NAME)$^+$ = {E-NAME} (E-CITY)$^+$ = {E-CITY, E_STATE}**

Find the attribute closures of given FDs
R(ABCDE) = {AB->C, B->D, C->E, D->A}

To find $(B)^+$,we will add attribute in set using various FD which has been shown in table below.



Attributes Added in Closure	FD used
{B}	Triviality
{B,D}	B->D
{B,D,A}	D->A
{B,D,A,C}	AB->C
{B,D,A,C,E}	C->E

- We can find $(C, D)^+$ by adding C and D into the set (triviality) and then E using(C->E) and then A using (D->A) and set becomes. **$(C,D)^+$ = {C,D,E,A}**
- Similarly we can find $(B,C)^+$ by adding B and C into the set (triviality) and then D using (B->D) and then E using (C->E) and then A using (D->A) and set becomes **$(B,C)^+$ ={B,C,D,E,A}**

The attribute(s) whose closure constitutes all the attributes in the given relation are candidate keys

- **R = (A, B, C, D, E, H) on which the following functional dependencies hold: {A–>B, BC–> D, E–>C, D–>A}. What are the candidate keys of R? [GATE 2005]**
  (a) AE, BE
  (b) AE, BE, DE
  (c) AEH, BEH, BCH
  (d) AEH, BEH, DEH
- **Answer:** (AE)+ = {ABECD} which is not set of all attributes. So AE is not a candidate key. Hence option A and B are wrong.
  (AEH)+ = {ABCDEH}
  (BEH)+ = {BEHCDA}
  (BCH)+ = {BCHDA} which is not set of all attributes. So BCH is not a candidate key. Hence option C is wrong.
  So correct answer is D.