Database Management Systems (DBMS)

Lec 14: Relational database design (Contd.)

Ramesh K. Jallu

IIIT Raichur

Date: 19/03/21

Recap

- Mapping between ER and Relational schema
- A quick recap of keys and constraints in relational schema
- Introduction to normalization

Today's plan

- Normal forms based on primary key and corresponding normalization
  - 1st normal form
  - 2nd normal form
  - 3rd normal form

Motivation

- So far, we presented various aspects of the relational model and the languages associated with it
- Each relation schema consists of a number of attributes, and the relational database schema consists of a number of relation schemas
- Attributes are grouped to form a relation schema
  - either by the intution of database designer
  - or by mapping a database schema design from an ER or EER data model

Motivation (Contd.)

- These models allow the designer identify entity types and relationship types and their respective attributes
  - which leads to a grouping of the attributes into relations using conversion procedures

- We should have some formal way of analyzing why one grouping of attributes into a relation schema may be better than another

- We discuss some of the theory that has been developed with the goal of evaluating relational schemas for design quality

Normalization (recap)

- Normalization of data we mean analyzing the given relation schemas based on their FDs and primary keys
- The objective of normalization are *information preservation* and *minimum redundancy*
- Minimizing redundancy we mean
  - minimizing redundant storage of the same information
  - reducing the need for multiple updates to maintain consistency across multiple copies of the same information

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	291Berry, Bellaire, TX	4
Narayan, Ramesh 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	Dmgr_ssn
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

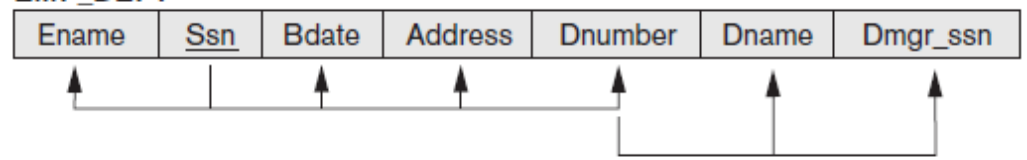
Ssn	Pnumber	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	Null

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

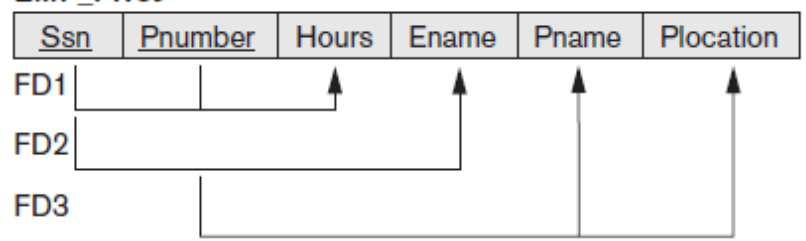
(a)

EMP_DEPT



(b)

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

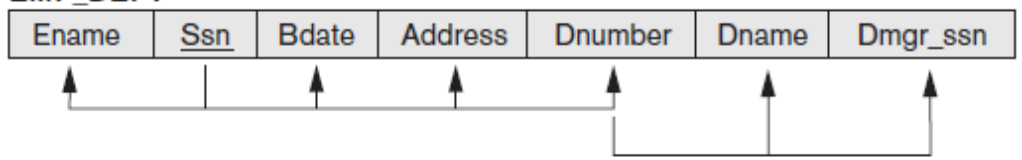
Redundancy Redundancy

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

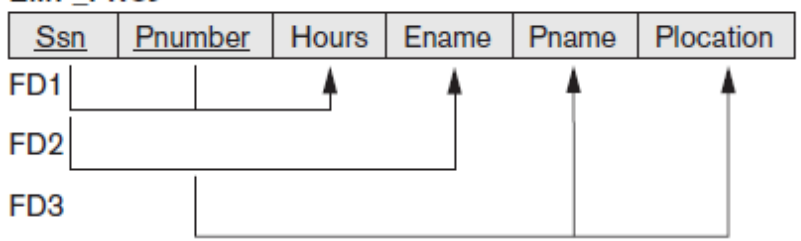
(a)

EMP_DEPT



(b)

EMP_PROJ



The first normal form (1NF)

- The first normal form states that, in a relation ***R***
  i. the domain of an attribute in ***R*** must include only ***atomic*** (simple, indivisible) values
  ii. the value of any attribute in a tuple must be a ***single*** value from the domain of that attribute

Illustration with an example

DEPARTMENT

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

1. The domain of Dlocations contains atomic values, but some tuples can have a set of these values
   - In this case, Dnumber → Dlocations is not satisfied

2. The domain of Dlocations contains sets of values and hence is nonatomic
   - In this case, Dnumber → Dlocations satisfied

Ways to achieve 1NF in previous example: The 1st way

- Remove Dlocations that violates 1NF and place it in a separate relation DEPT_LOCATIONS along with the primary key Dnumber of DEPARTMENT
- The primary key of this newly formed relation is the combination { Dnumber , Dlocation }
- A distinct tuple in DEPT_LOCATIONS exists for each location of a department.
- This decomposes the non-1NF relation into two 1NF relations.

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Ways to achieve 1NF in previous example: The 2nd way

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

- Expand the key so that there will be a separate tuple in the original DEPARTMENT relation for each location of a DEPARTMENT
- In this case, the primary key becomes the combination {Dnumber, Dlocation}
- This solution has the disadvantage of introducing *redundancy* in the relation and hence is rarely adopted

Ways to achieve 1NF in previous example: The 3rd way

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation 1	Dlocaton 2	Dlocation 3
Research	5	333445555	Bellaire	Sugarland	Houston
Administration	4	987654321	Stafford	**NULL**	**NULL**
Headquarters	1	888665555	Houston	**NULL**	**NULL**

- If a maximum number of values is known, for example, say 3, replace the Dlocations attribute by three atomic attributes: Dlocation1, Dlocation2, and Dlocation3

- This solution has the disadvantages: (i) introduces **NULL** values if most departments have fewer than three locations, and (ii) spurious semantics about the ordering that is not originally intended

The first normal form (Contd.)

- First normal form also disallows multivalued attributes that are themselves composite
  - These are called ***nested relations*** because each tuple can have a relation within it

  - EMP_PROJ(Ssn, Ename, {PROJS(Pnumber, Hours)})

(a)

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

(b)

EMP_PROJ

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

- Ssn is a primary key for EMP_PROJ
- Pnumber is a partial key of the nested relation; that is, within each tuple, the nested relation must have unique values of Pnumber

How to normalize a nested relation?

- To normalize this into 1NF, we remove the nested relation attributes into a new relation and propagate the primary key into it
  - the primary key of the new relation will combine the partial key with the primary key of the original relation

- Decomposition and primary key propagation yield the schemas EMP_PROJ1 and EMP_PROJ2

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours

Other type of nested relations

1. CANDIDATE (Ssn, Name, {JOB_HIST (Company, Highest_position, {SAL_HIST (Year, Max_sal)})})

   CANDIDATE_1 (Ssn, Name)
   CANDIDATE_JOB_HIST (Ssn, Company, Highest_position)
   CANDIDATE_SAL_HIST (Ssn, Company, Year, Max-sal)

2. PERSON (Ssn, {Car_lic}, {Phone_no})
   i. PERSON_IN_1NF (Ssn, Car_lic, Phone_no)
   ii. P1(Ssn, Car_lic) and P2(Ssn, Phone_no)

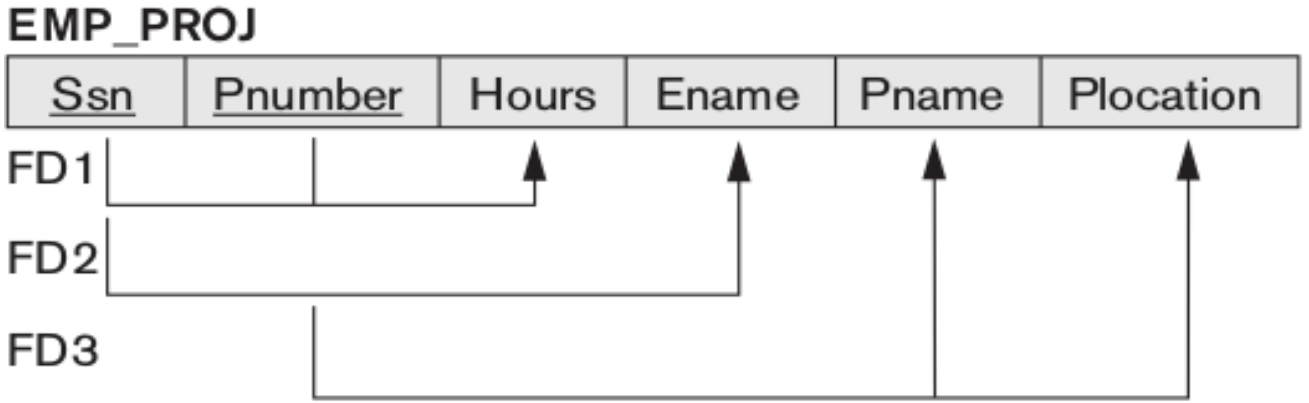
Full functional dependency

- A functional dependency is a constraint between two sets of attributes from the database
  - In the FD $X \rightarrow Y$, the values of the X component of a tuple uniquely determine the values of the Y component

- A functional dependency $X \rightarrow Y$ is a ***full functional dependency*** if removal of any attribute A from X means that the dependency does not hold anymore
  - For any $A \in X$, $(X - A) \rightarrow Y$ is not satisfied

Partial functional dependency

- A functional dependency $X \rightarrow Y$ is a ***partial functional dependency*** if some attribute $A \in X$ can be removed from X and the dependency still holds
  - For some $A \in X$, $(X - A) \rightarrow Y$ is satisfied



- {Ssn , Pnumber} → Hours is a full dependency
- {Ssn, Pnumber} →Ename is a partial dependency

Prime and nonprime attributes

- An attribute of relation schema ***R*** is called a ***prime*** attribute of ***R*** if it is a member of some candidate key of ***R***
- An attribute is called ***nonprime*** if it is not a prime attribute, that is if it is not a member of any candidate key

WORKS_ON

Essn	Pno	Hours

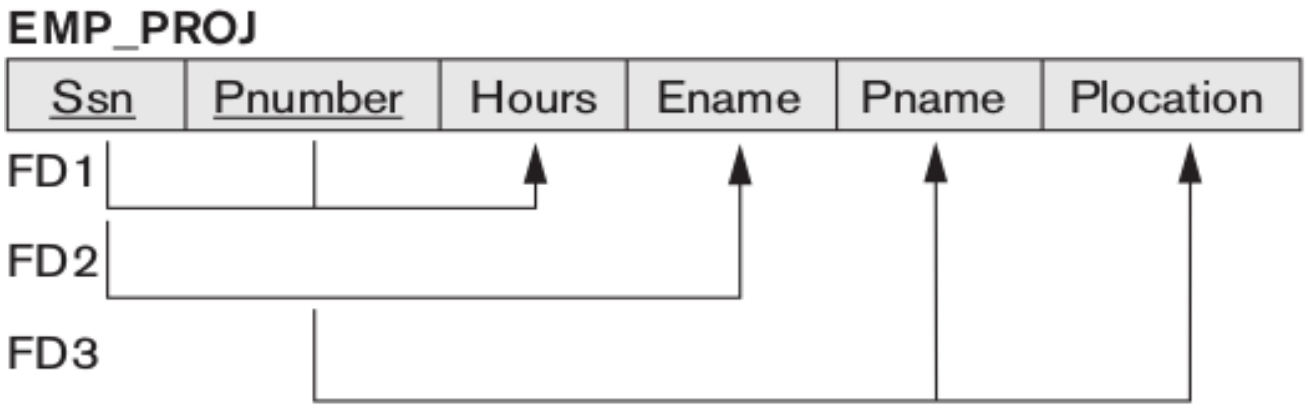
DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

The second normal form (2NF)

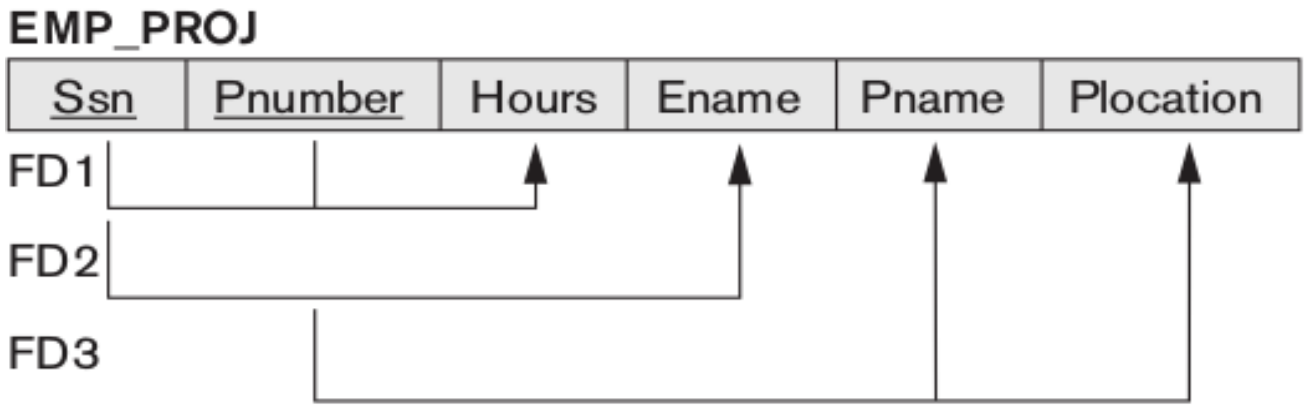
- A relation schema ***R*** is in 2NF if every nonprime attribute A in R is *fully functionally dependent* on the primary key of R
- The test for 2NF involves testing for functional dependencies whose left-hand side attributes are part of the primary key
- If the primary key contains a single attribute, the test need not be applied at all

Example



- The above relation is 1 NF but not in 2NF

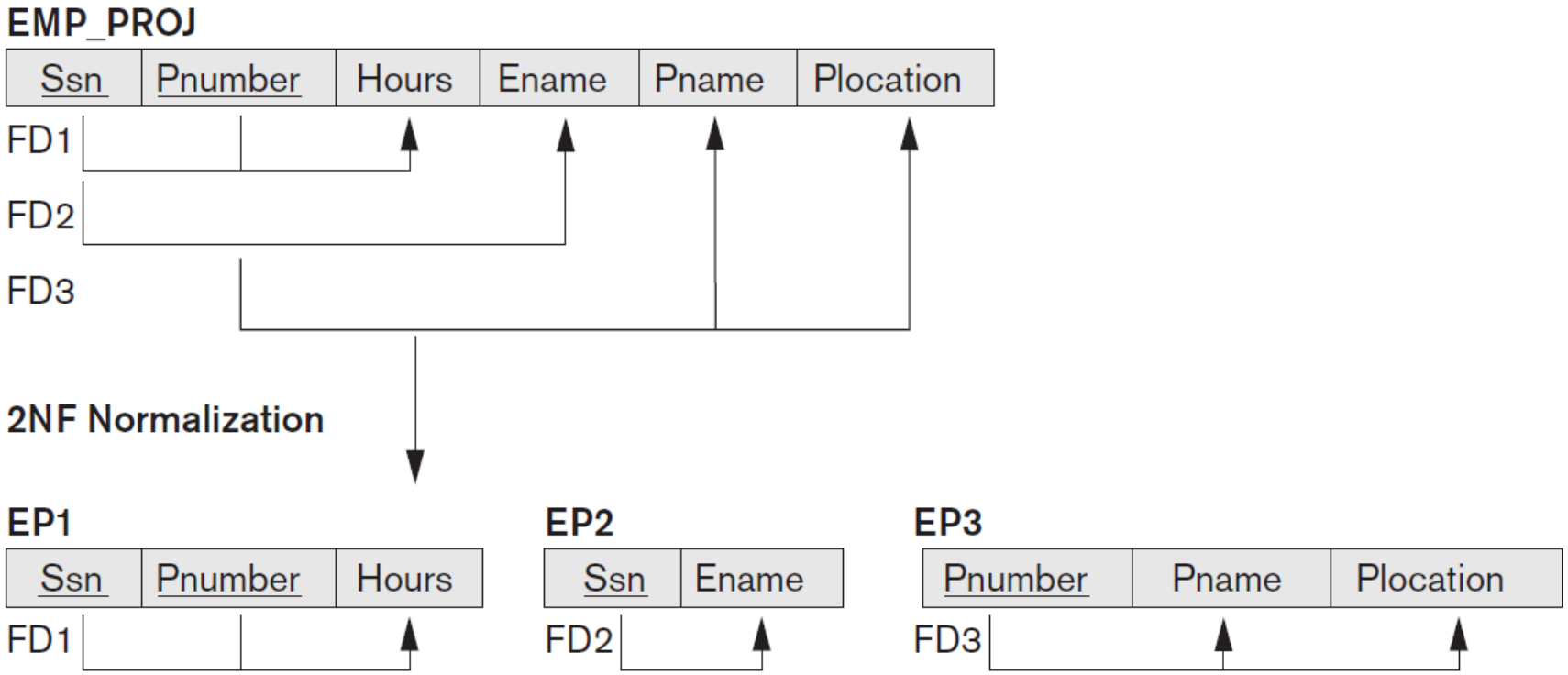
Example



- The above relation is 1 NF but not in 2NF
  - {Ssn, Pnumber} → Ename is not fully functionally dependent
  - FD3 violates 2NF aswell

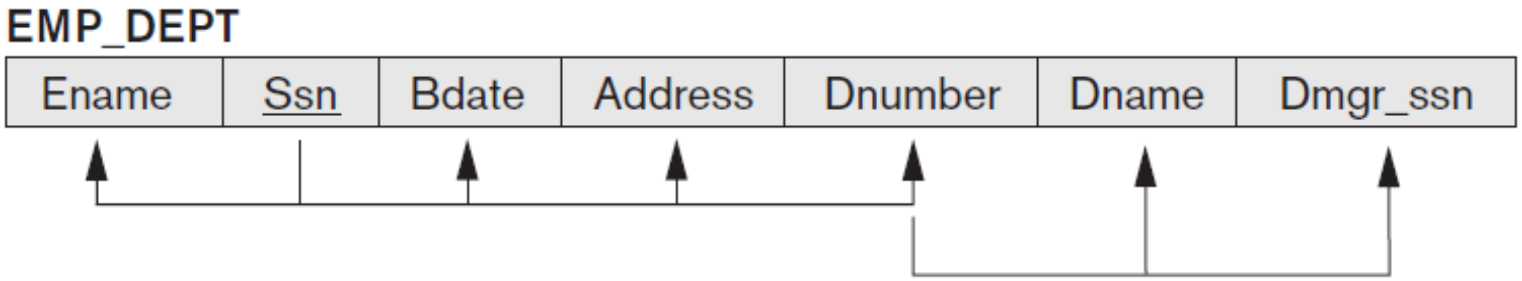
Normalizing FDs

- If a relation is not in 2NF, it can be decomposed into a number of 2NF relations in which nonprime attributes are associated only with the part of the primary key on which they are fully functionally dependent



Transitive dependency

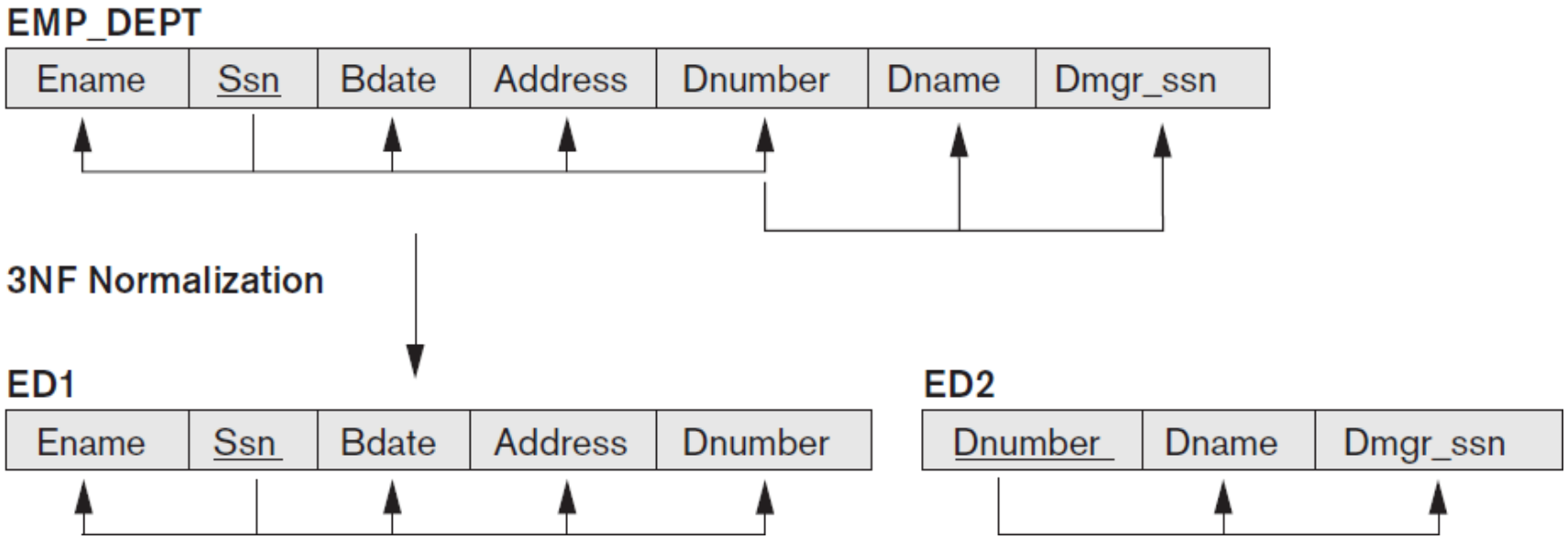
- A functional dependency X → Y in a relation ***R*** is a ***transitive dependency*** if there exists a set of attributes Z in ***R*** that is *neither a candidate key nor a subset of any key* of ***R***, and both X → Z and Z → Y hold



- The dependency Ssn → Dmgr_ssn is transitive through Dnumber in EMP_DEPT because both the dependencies Ssn → Dnumber and Dnumber → Dmgr_ssn hold and Dnumber is neither a key itself nor a subset of the key of EMP_DEPT

The third normal form (3NF)

- A relation schema ***R*** is in 3NF if
  - it satisfies 2NF, and
  - no nonprime attribute of ***R*** is transitively dependent on the primary key



Summary

Normal Form	Test	Remedy (Normalization)
First (1NF)	Relation should have no multivalued attributes or nested relations.	Form new relations for each multivalued attribute or nested relation.
Second (2NF)	For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
Third (3NF)	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes). That is, there should be no transitive dependency of a nonkey attribute on the primary key.	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).

Summary (Contd.)

- After normalization, any functional dependency in which the left-hand side is part (a proper subset) of the primary key
- Any functional dependency in which the left-hand side is a nonkey attribute, is a problematic FD
- Normalization remove these problem FDs by decomposing the original relation into new relations

Thank you!