Chapter 8: Relational Database Design

Database System Concepts, 6th Ed.

Design of Relation Schemas

- A *database* can be modeled as:
  - a collection of entities,
  - Attributes are properties used to describe an entity.
- Relational database design?
  - The grouping of attributes to form "good" relation schemas.
  - To measure the quality of the design.
  - The goals of the design activity are information protection and minimum redundancy.

Informal Design Guidelines for Relation Schemas

1. Clear Semantics to Attributes in Relations
2. Redundant Information in Tuples and Anomalies
3. NULL Values in Tuples
4. Generation of Spurious Tuples

1. Clear Semantics to Attributes in Relations

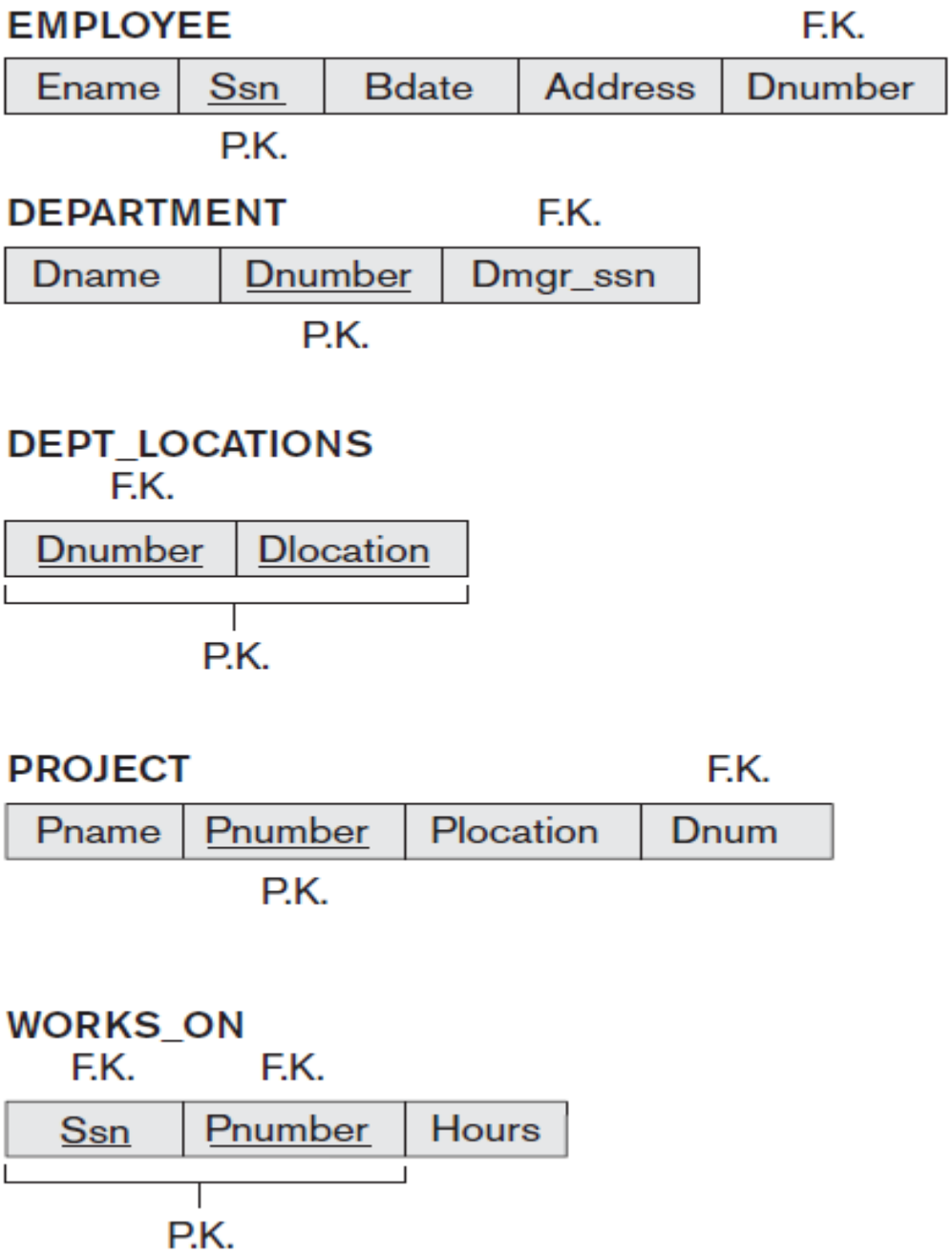
Guideline #01:

Informally, each tuple in a relation should represent one entity or relationship instance.

- 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

- Design a schema that can be explained easily relation by relation.

A simplified COMPANY relational database schema.



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

PROJECT

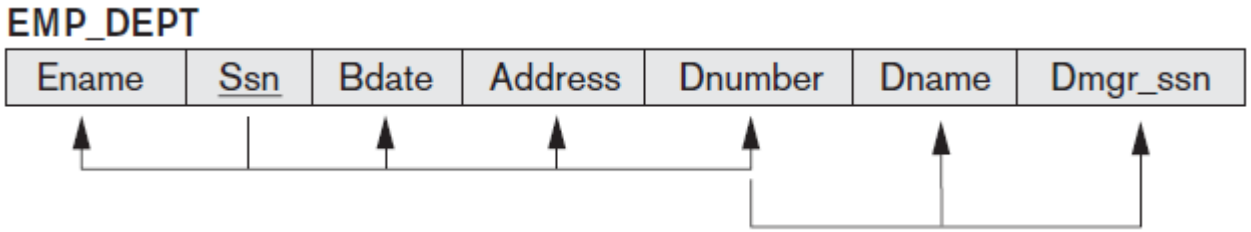
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Sample database state for some of the COMPANY relational database

2. Redundant Information in Tuples and Anomalies

- One goal of schema design is to minimize the storage space used by the base relations.
- Grouping attributes into relation schemas has a significant effect on storage space
- Information is stored redundantly wasting storage
- Example:
  - Two relations EMPLOYEE and DEPARTMENT.
  - Single EMP_DEPT relation.

Sample schema and states for EMP_DEPT



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

- In EMP_DEPT, the attribute values of a particular department (Dnumber, Dname, Dmgr_ssn) are repeated for every employee who works for that department. *In contrast*, each department's information appears only once in the DEPARTMENT relation.

2. Redundant Information in Tuples and Anomalies

Insertion Anomalies:

- To insert a new employee tuple into EMP_DEPT, we must include either the attribute values for the department that the employee works for, or NULLs (if the employee does not work for a department as yet).
- It is difficult to insert a new department that has no employees as yet in the EMP_DEPT relation.

2. Redundant Information in Tuples and Anomalies

Deletion Anomalies:

- If we delete from EMP_DEPT an employee tuple that happens to represent the last employee working for a particular department, the information concerning that department is lost from the database.

2. Redundant Information in Tuples and Anomalies

Modification Anomalies:

- In EMP_DEPT, if we change the value of one of the attributes of a particular department (say, the manager of department 5) we must update the tuples of all employees who work in that department; otherwise, the database will become inconsistent.

2. Redundant Information in Tuples and Anomalies

Guideline #02:

Design a schema that does not suffer from the insertion, deletion and update anomalies.

3. NULL Values in Tuples

- Many attributes may be grouped together into a “fat” relation.
- If many of the attributes do not apply to all tuples in the relation, we end up with many NULLs in those tuples. This can waste space at the storage level.
- Reasons for nulls:
  - attribute not applicable or invalid.
  - attribute value unknown (may exist).
  - value known to exist, but not yet available.

3. NULL Values in Tuples

Guideline #03:

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)

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.
- X -> Y *holds* if whenever two tuples have the same value for X, they must have the same value for Y.
- We also say that there is a functional dependency from X to Y, or that Y is functionally dependent on X.
  - A functional dependency is a generalization of the notion of a *key*

Functional Dependencies (Cont…)

- A functional dependency X -> Y is a *full dependency* if removal of any attribute A from X means that the dependency does not hold any more.
- A functional dependency X->Y is a *partial dependency* if some attribute A can be removed from X and the dependency still holds.

Normalization

- Normalization: The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations
  - Considered as a “filtering” or “purification” process to make the design have better quality, through a series of tests to certify whether a relation schema satisfies a certain *normal form.*
- Normalization is a process of analyzing relation schemas to achieve:
  - Minimizing redundancy.
  - Minimizing the insertion, deletion, and update anomalies

Key and Attributes

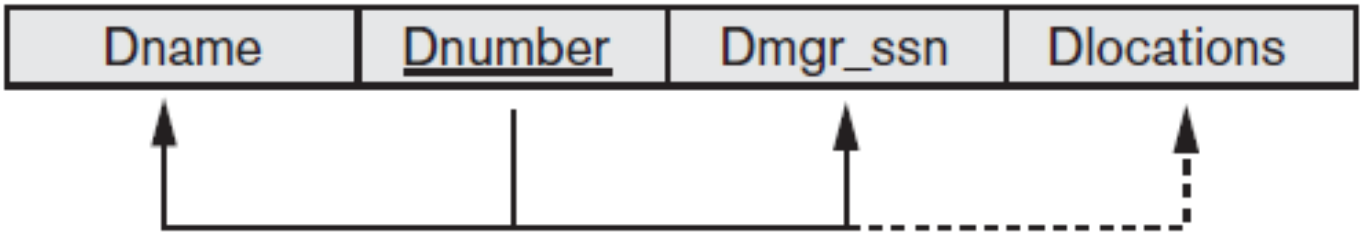
- If a relation has more than one key, each is called a *candidate key*.
- One of the *candidate keys* is designated to be the *primary key*, and the others are called secondary keys.
- A *Prime attribute* must be a member of some candidate key.
- A *Nonprime attribute* is not a prime attribute—that is, it is not a member of any candidate key.

First Normal Form (1NF)

- Disallows composite attributes and multivalued attributes; attributes whose values for an individual tuple are non-atomic.
- The value of any attribute in a tuple must be atomic (simple) value.

First Normal Form (1NF) Cont…

DEPARTMENT



(b)

DEPARTMENT

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

First Normal Form (1NF) Cont…

- DEPARTMENT relation is not in 1NF:
  - The domain of Dlocations contains sets of values and hence is nonatomic.
  - Dlocations is not functionally dependent on the primary key Dnumber.

First Normal Form (1NF) Cont…

- There are *three main techniques* to achieve first normal form for such a relation:
  1. Remove the attribute 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 relation is the combination{Dnumber, Dlocation}

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

First Normal Form (1NF) Cont…

2. Expand the key so, 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.

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

First Normal Form (1NF) Cont…

3. If a maximum number of values is known for the attribute (for example, if it is known that at most three locations can exist for a department) replace the Dlocations attribute by three atomic attributes: Dlocation1, Dlocation2, and Dlocation3.
   - This solution has the disadvantage of introducing NULL values if most departments have fewer than three locations.

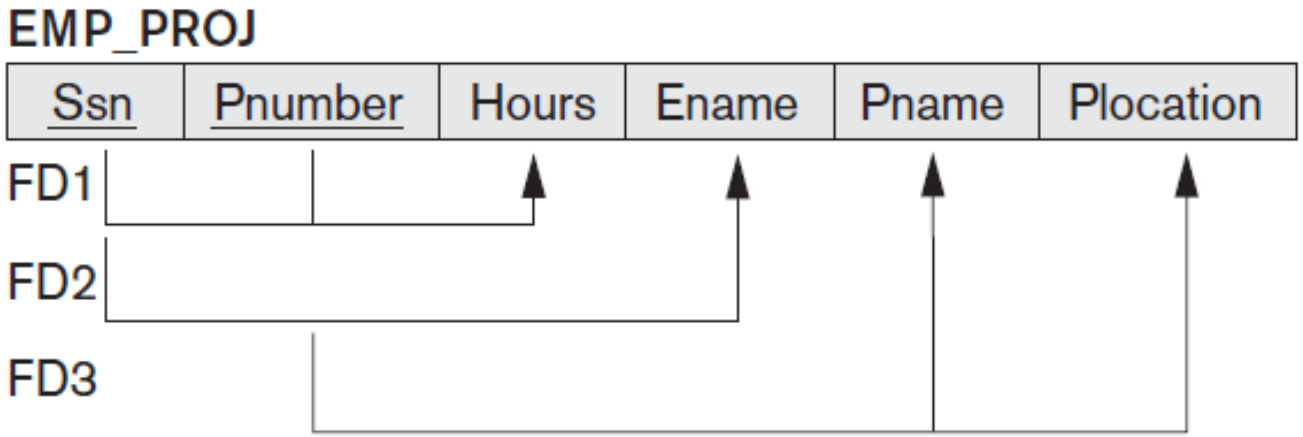
- From the 3 techniques, the first considered Best.

Second Normal Form (2NF)

- Second normal form (2NF) is based on the concept of full functional dependency.
  - A functional dependency X -> Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.
- **Definition**: A relation R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.
  - For relations where primary key contains multiple attributes.

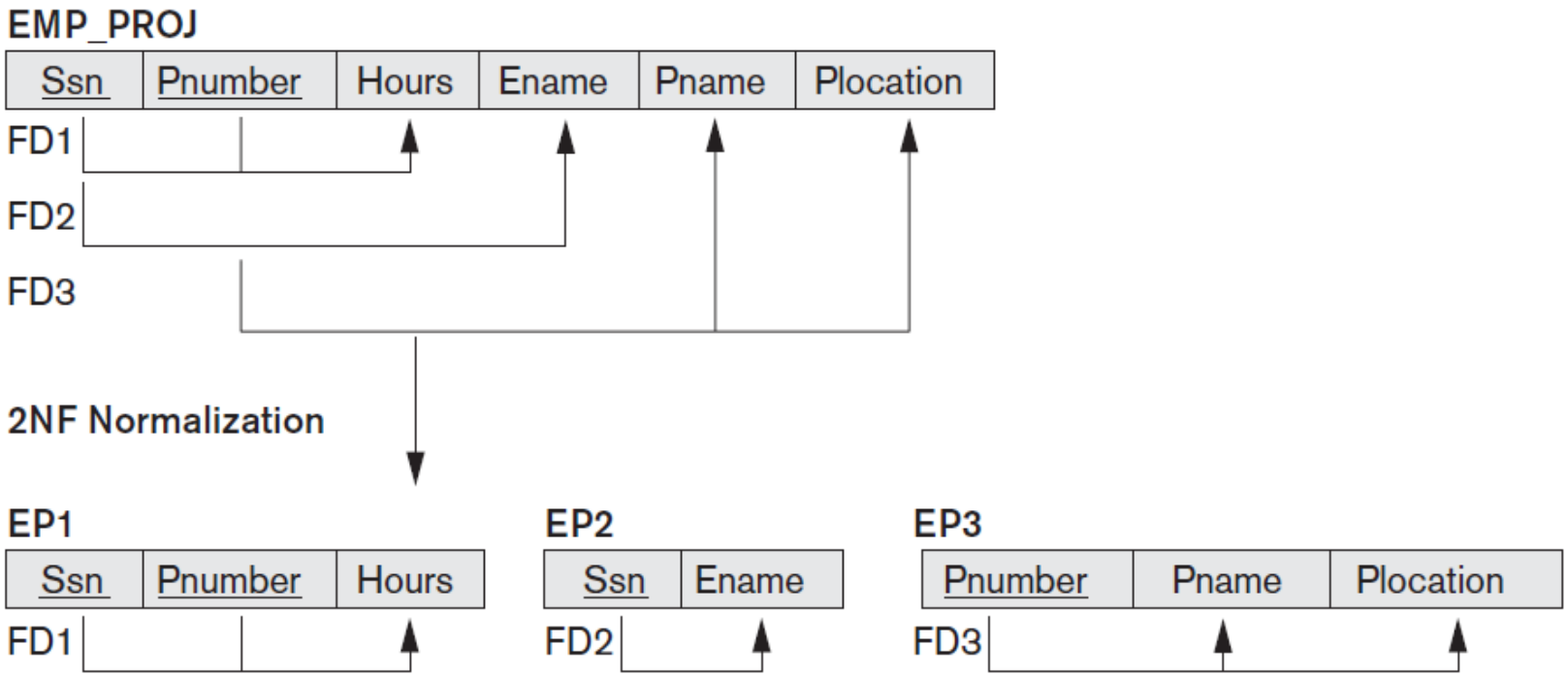
Second Normal Form (2NF) Cont…

- The EMP_PROJ relation is in 1NF but is not in 2NF.
  - The attribute Ename violates 2NF because of FD2, as do the attributes Pname and Plocation because of FD3.
  - The functional dependencies FD2 and FD3 make Ename, Pname, and Plocation *partially dependent* on the primary key {Ssn, Pnumber} of EMP_PROJ, thus violating the 2NF test.



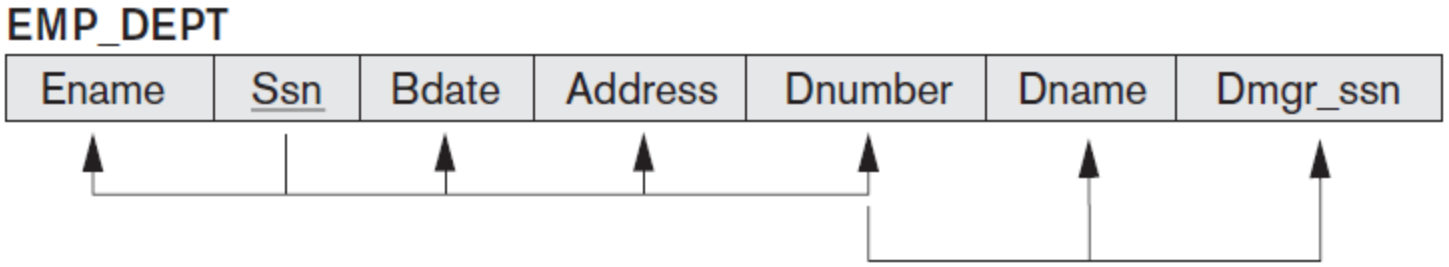
Second Normal Form (2NF) Cont…

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



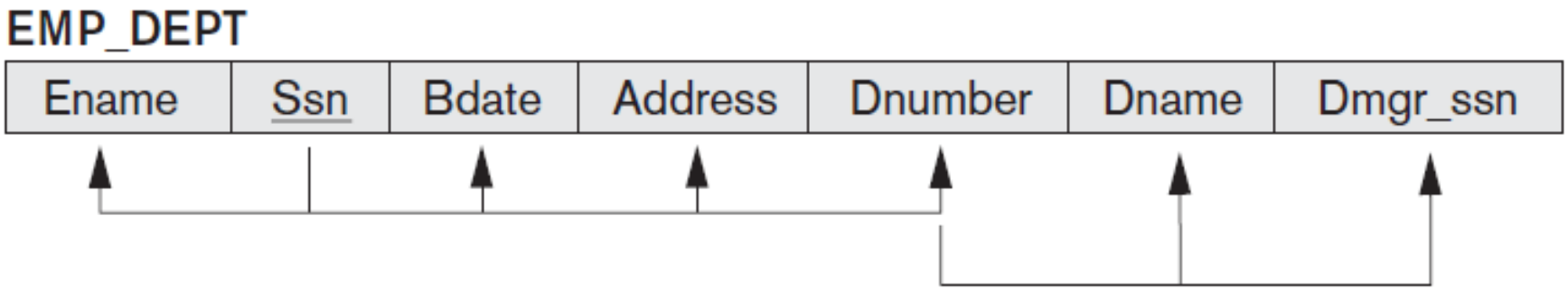
Third Normal Form (3NF)

- Third normal form (3NF) is based on the concept of transitive dependency.
  - A functional dependency X->Y in a relation R is a transitive dependency if that can be derived from two FDs X->Z and Z->Y.
- A relation 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.



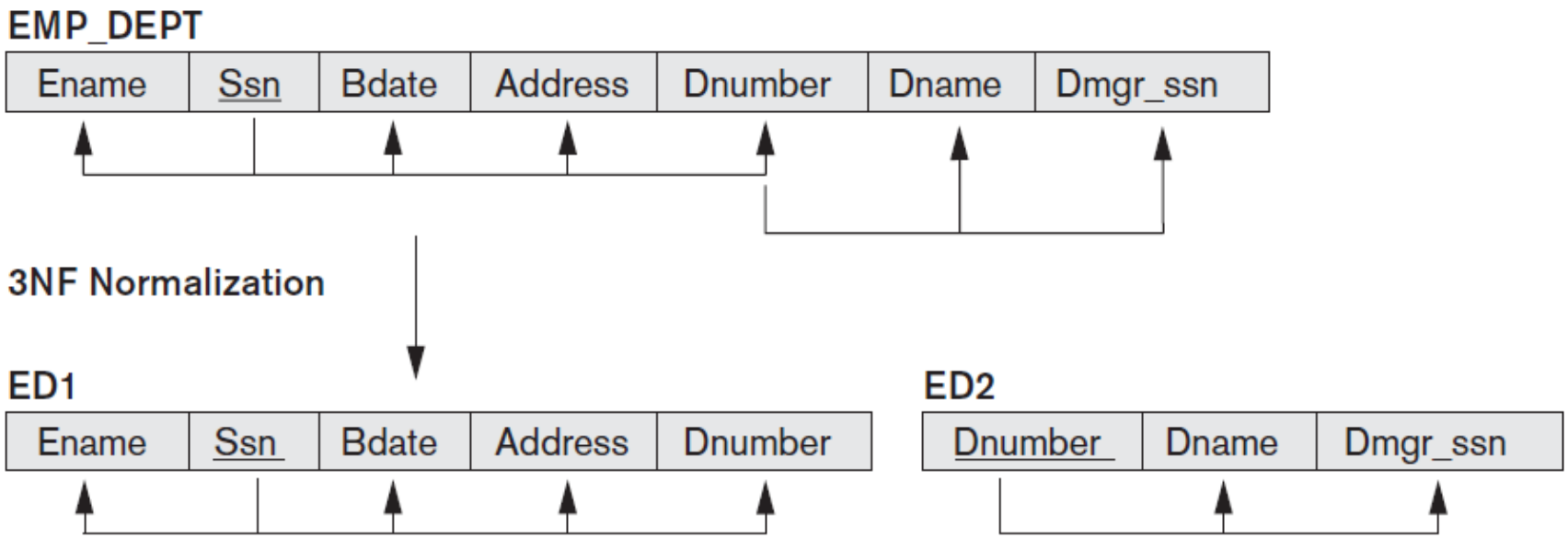
Third Normal Form (3NF) Cont…

- The dependency Ssn → Dmgr_ssn is transitive through Dnumber in EMP_DEPT, because both the dependencies Ssn → Dnumber and Dnumber → Dmgr_ssn hold.
  - The relation EMP_DEPT is in 2NF, since no partial dependencies on a key exist.
  - However, EMP_DEPT is not in 3NF because of the transitive dependency of Dmgr_ssn (and also Dname) on Ssn via Dnumber



Third Normal Form (3NF) Cont…

- We can normalize EMP_DEPT by decomposing it into the two 3NF relation schemas ED1 and ED2.
  - ED1 and ED2 represent independent entity facts about employees and departments.
  - A NATURAL JOIN on ED1 and ED2 will recover the original relation.



Let's Sum Up...!

Dependencies: Definitions

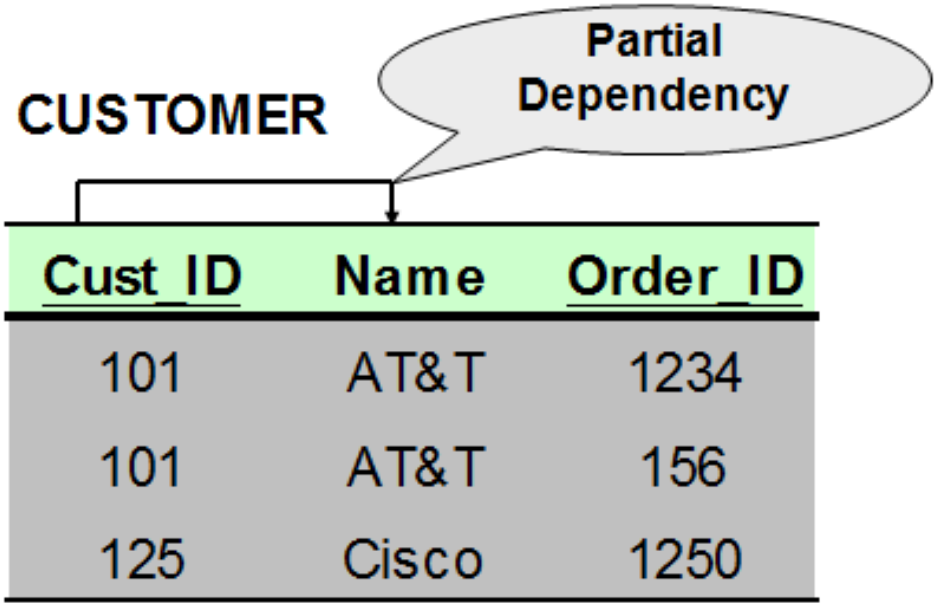
- ***Multivalued Attributes:***– non-key attributes the values of them are not uniquely identified by (not functionally dependent on) the Primary Key.

STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

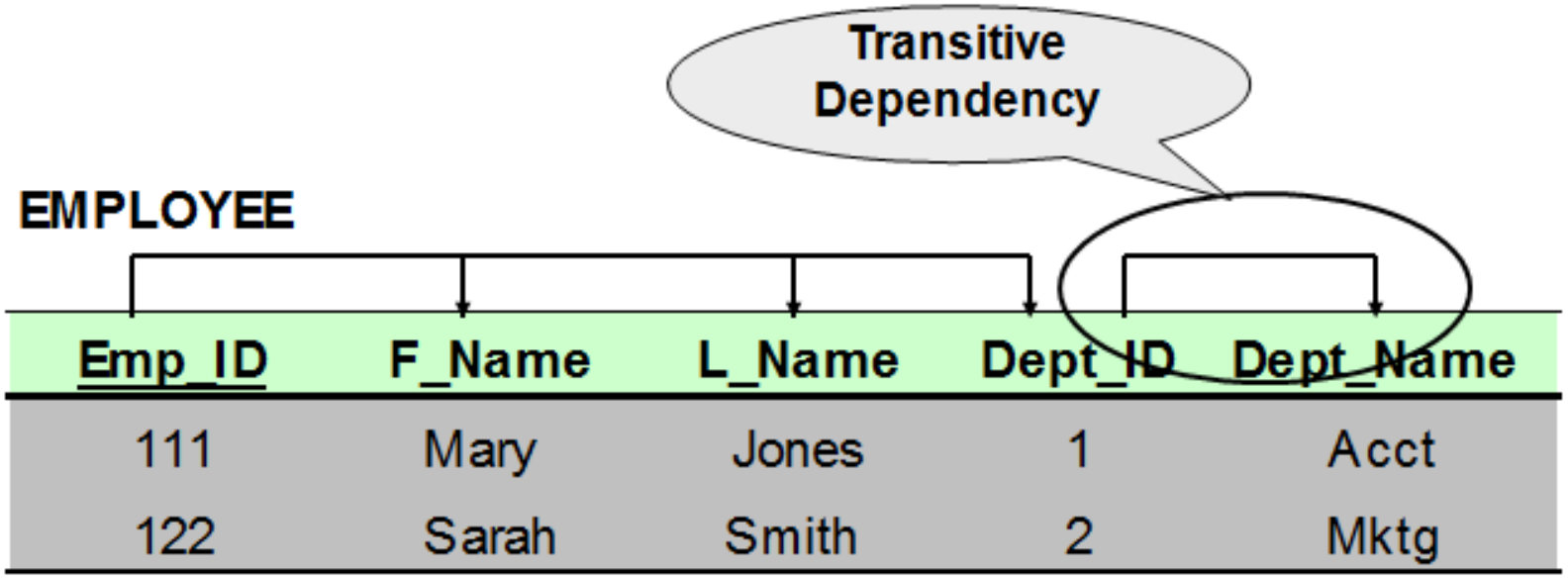
Dependencies: Definitions

- ***Partial Dependency:***– when an non-key attribute is determined by a part, but not the whole, of a COMPOSITE primary key.



Dependencies: Definitions

- ***Transitive Dependency:***– when a non-key attribute determines another non-key attribute.



What is Normalization

- Normalization allows us to organize data so that it:
  - Allows faster access (dependencies make sense)
  - Reduced space (less redundancy)

Normal Forms: Review

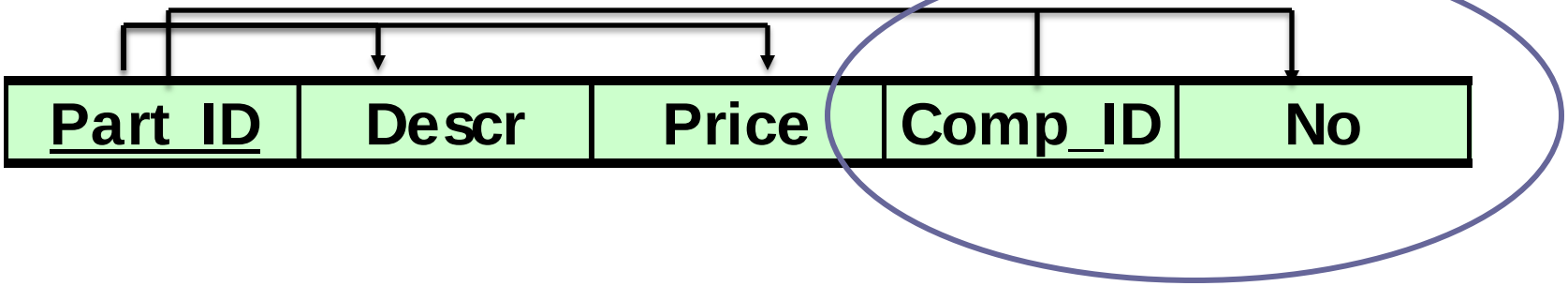
- Unnormalized – There are multivalued attributes or repeating groups
- 1 NF – No multivalued attributes or repeating groups.
- 2 NF – 1 NF plus no partial dependencies
- 3 NF – 2 NF plus no transitive dependencies

Example 1: Determine NF

- Part_ID → Description
- Part_ID → Price
- Part_ID, Comp_ID → N

Comp_ID and No are not determined by the primary key; therefore, the relation is NOT in 1 NF. No sense in looking at partial or transitive dependencies.

PART

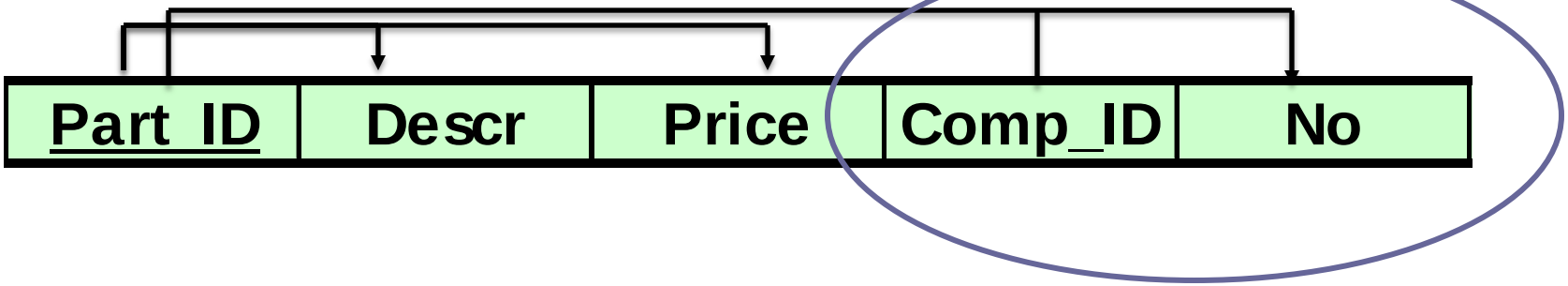


Example 1: Determine NF

- Part_ID → Description
- Part_ID → Price
- Part_ID, Comp_ID → N

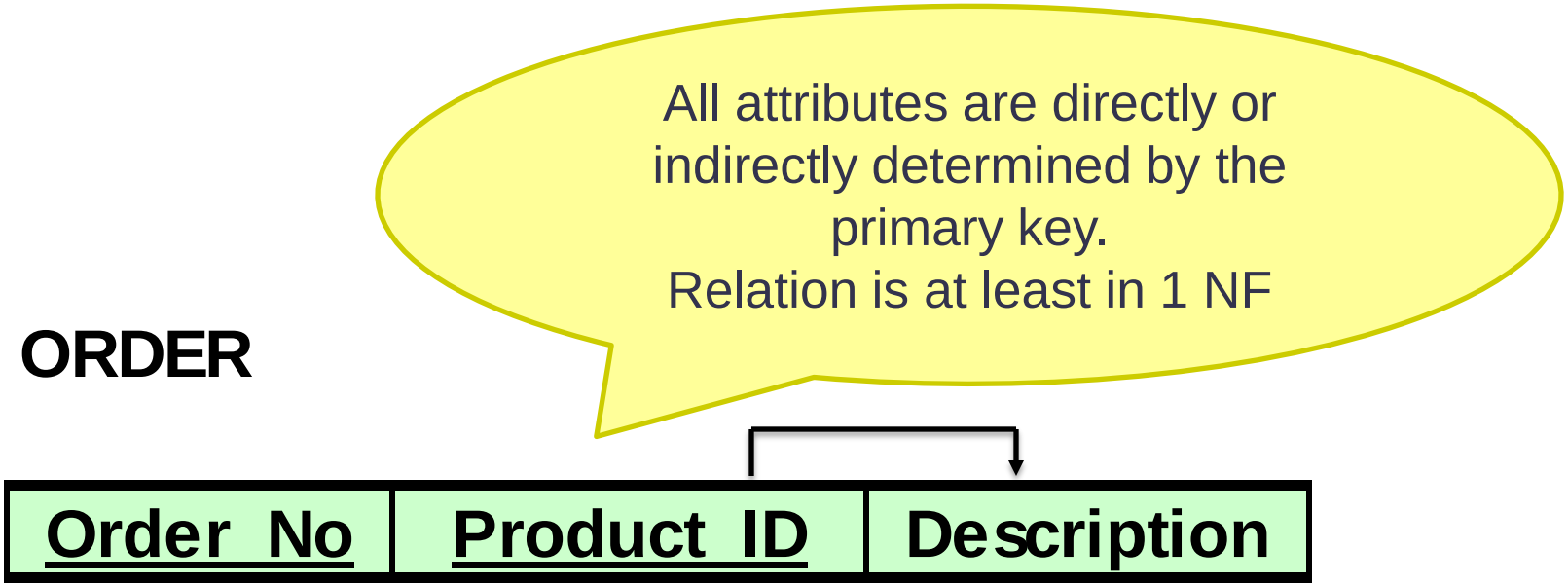
In your solution you will write the following :
There are M/V attributes; so, not 1NF
Conclusion: The relation is not normalized.

PART



Example 2: Determine NF

- Product_ID → Description



Example 2: Determine NF

- Product_ID → Description

The relation is at least in 1NF.
There is a COMPOSITE Primary Key (Order_No, Product_ID), so there can be partial dependencies.
Product_ID, which is a part of PK, determines Description; hence, there is a partial dependency.
So, the relation is not 2NF.
No need to check for transitive dependencies (3NF)!

ORDER

Example 2: Determine NF

- Product_ID → Description

We know that the relation is at least in 1NF, and it is not in 2 NF. Therefore, we conclude that the relation is in 1 NF.

ORDER

Order_No	Product_ID	Description

Example 2: Determine NF

- Product_ID → Description

In your solution you will write the following:
1) No M/V attributes, therefore at least 1NF
2) There is a partial dependency (Product_ID → Description), therefore not in 2NF
Conclusion: The relation is in 1NF

ORDER

Example 3: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

All attributes are directly or indirectly determined by the primary key.
Relation is at least in 1 NF

BOOK

Example 3: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

The relation is at least in 1NF.
There is no COMPOSITE primary key, so there can't be partial dependencies.
Relation is at least in 2NF

BOOK

Example 3: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

Publisher is a non-key attribute, and it determines Address, another non-key attribute. There is a transitive dependency, which means that the relation is NOT in 3 NF.

BOOK

Example 3: Determine NF

- ISBN → Title
- ISBN → Publisher
- Publisher → Address

We know that the relation is at least in 2NF, and it is not in 3 NF. Therefore, we conclude that the relation is in 2NF.

BOOK

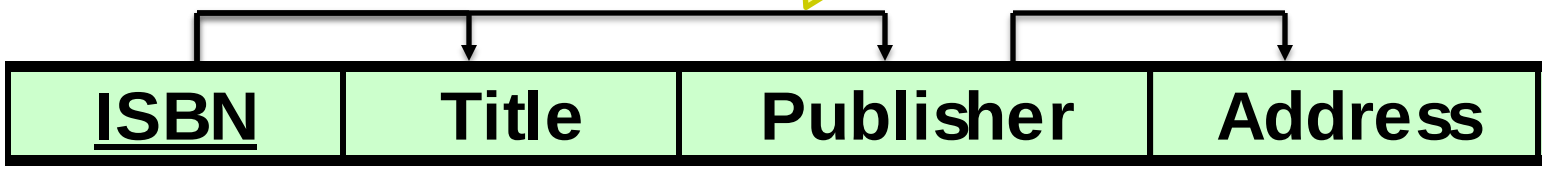
- ISBN → Title
- ISBN → Publisher
- Publisher → Address

In your solution you will write the following :

1) No M/V attributes, so at least 1NF
2) No partial dependencies, at least 2NF
3) There is a transitive dependency (Publisher → Address), so , not 3NF

Conclusion: The relation is in 2NF

BOOK



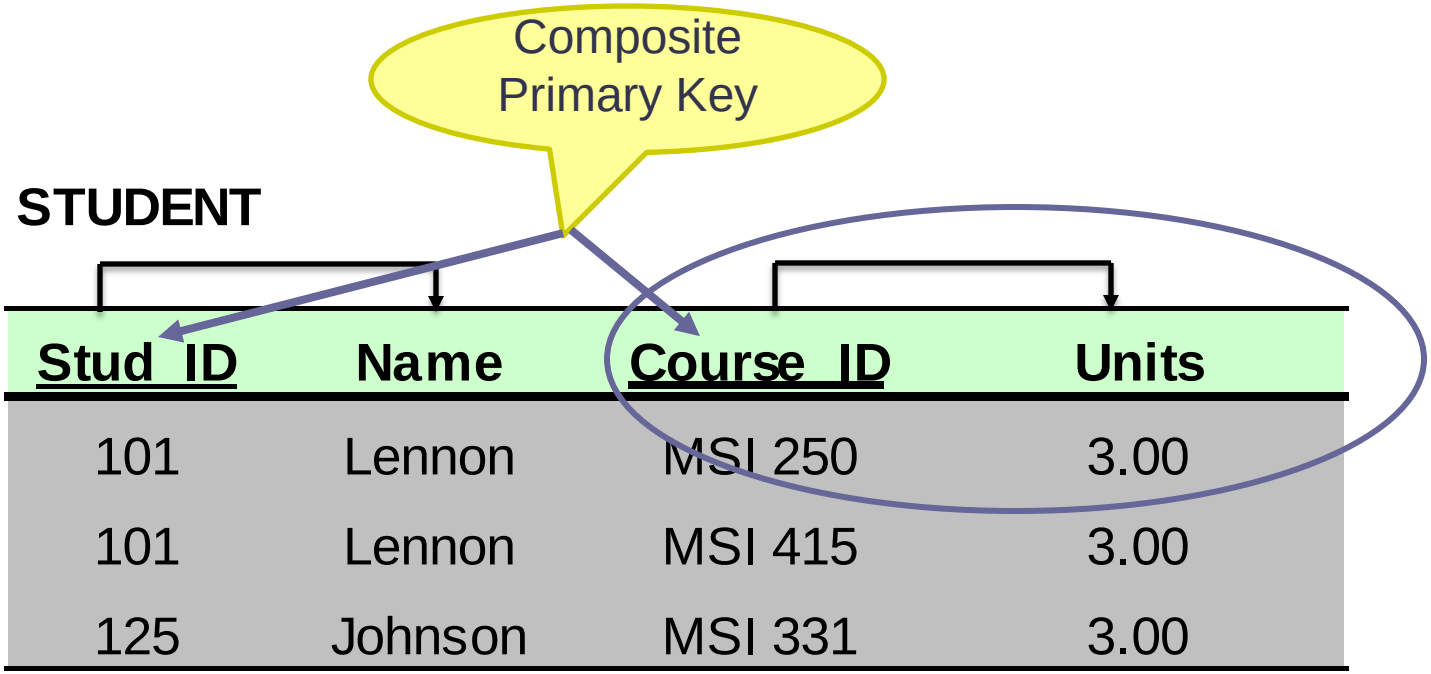
Bringing a Relation to *1NF*

STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to *1NF*

- Option 1: Make a determinant of the repeating group (or the multivalued attribute) a part of the primary key.



Bringing a Relation to *1NF*

- Option 2: Remove the entire repeating group from the relation.
  - Create another relation which would contain all the attributes of the repeating group, plus the primary key from the first relation.
  - In this new relation, the primary key from the original relation and the determinant of the repeating group will comprise a primary key.

STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to *1NF*

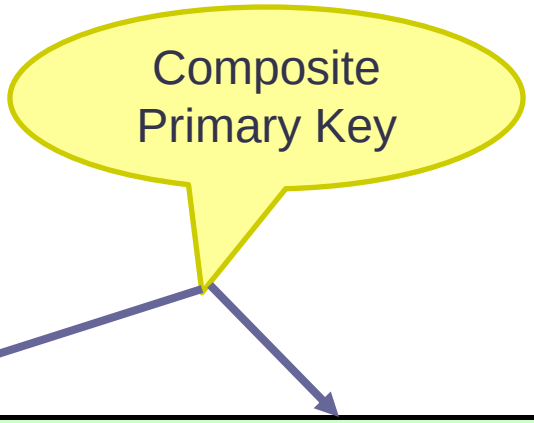
STUDENT

Stud_ID	Name
101	Lennon
125	Jonson

STUDENT_COURSE

Stud_ID	Course	Units
101	MSI 250	3
101	MSI 415	3
125	MSI 331	3

Bringing a Relation to *2NF*

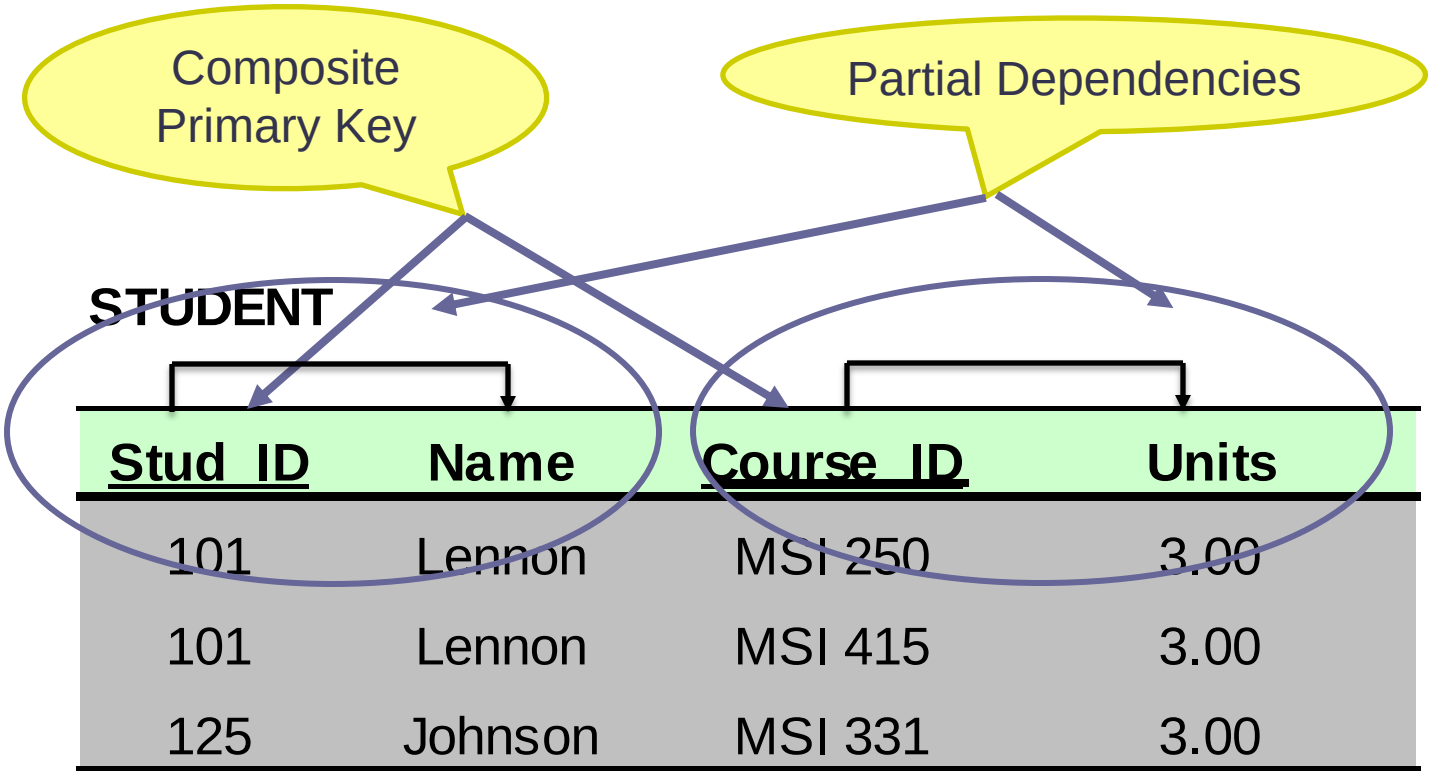


STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to *2NF*

- Goal: Remove *Partial Dependencies*



Bringing a Relation to *2NF*

- Remove attributes that are dependent from the part but not the whole of the primary key.
  - For each partial dependency, create a new relation, with the corresponding part of the primary key from the original as the primary key.

STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

Bringing a Relation to *2NF*

STUDENT

Stud_ID	Name	Course_ID	Units
101	Lennon	MSI 250	3.00
101	Lennon	MSI 415	3.00
125	Johnson	MSI 331	3.00

STUDENT_COURSE

Stud_ID	Course_ID
101	MSI 250
101	MSI 415
125	MSI 331

STUDENT

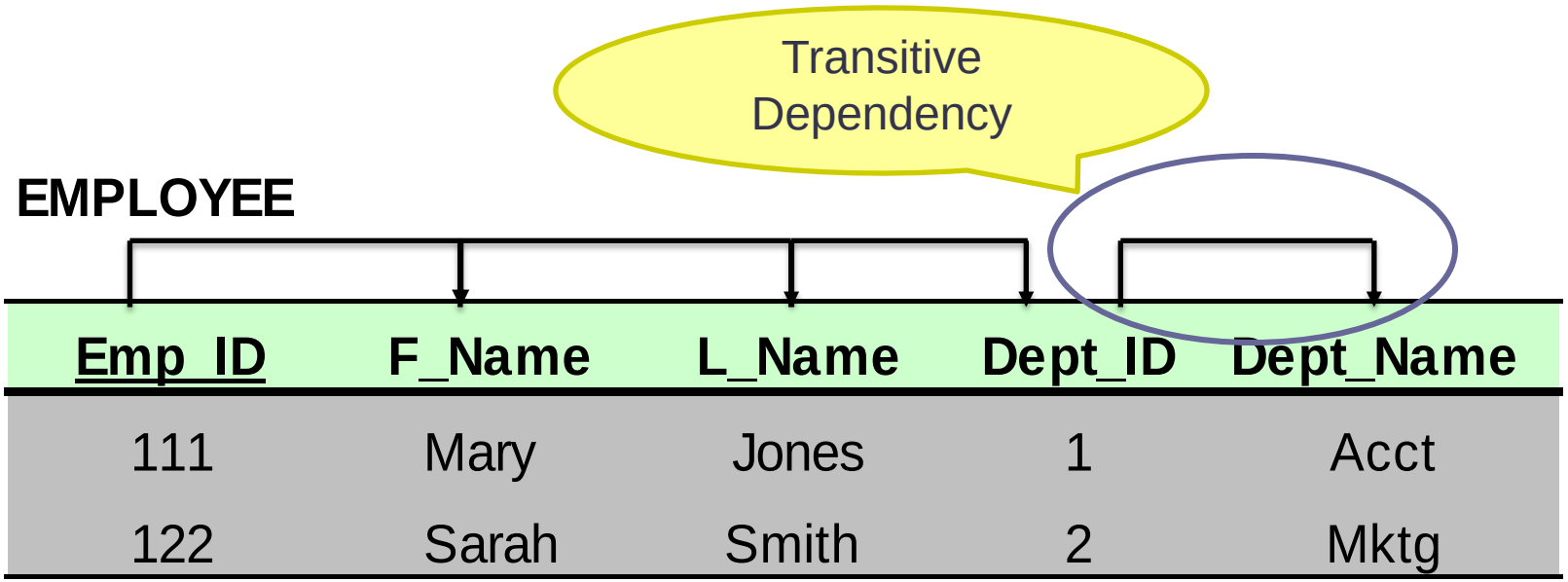
Stud_ID	Name
101	Lennon
101	Lennon
125	Johnson

COURSE

Course_ID	Units
MSI 250	3.00
MSI 415	3.00
MSI 331	3.00

Bringing a Relation to *3NF*

- Goal: Get rid of *transitive dependencies*.



Bringing a Relation to *3NF*

- Remove the attributes, which are dependent on a non-key attribute, from the original relation.
  - For each transitive dependency, create a new relation with the non-key attribute which is a determinant in the transitive dependency as a primary key, and the dependent non-key attribute as a dependent.

EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

Bringing a Relation to *3NF*

EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID	Dept_Name
111	Mary	Jones	1	Acct
122	Sarah	Smith	2	Mktg

EMPLOYEE

Emp_ID	F_Name	L_Name	Dept_ID
111	Mary	Jones	1
122	Sarah	Smith	2

DEPARTMENT

Dept_ID	Dept_Name
1	Acct
2	Mktg

All the Best...!