Advanced Database Management System

Lecture 11 – Sections 10.1-10.4
Schema Design and Normalization

Designing Good Schemas

- We know how to create schemas, but ...
  - how do we create *good* schemas?
  - what does *good* mean?
- Schema quality measurements:
  - semantics of the attributes
  - minimal redundancy
  - minimal frequency of null values

Clear Semantics

- Design schemas to be easy to understand
  - Each relation models a real-world entity type or relationship type (similar to a class)
  - Each tuple represents a real-world entity or a relationship between two entities
- Mixing unrelated information in the same relation is confusing
  - Avoid: “huh? why is this stored here?”
- Good names always improve clarity

Clear Semantics

- Each tuple in a relation should represent one entity or relationship instance.
  - Attributes of different entity types (i.e. 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.

- Bottom Line: Design a schema that can be explained easily *relation by relation*. The semantics of attributes should be easy to interpret.

> If you started with a good ER design, you should have a clear relational schema.

Reduce Redundancy

- If the same information is stored in multiple places, then extra work is required to ensure consistency
- We can reduce redundancy by splitting relations
  - keep the primary key in both new relations – join on PK to get back the original table when needed
- Redundancy leads to *update anomolies*
  - examples:
    - insertion: adding some info requires copying other info
    - deletion: removing some info cause unrelated info to be lost

Update Anomaly Example

- Consider the relation:
  - EMP_PROJ(<u>EmpNum</u>, <u>ProjNum</u>, Ename, Pname, NumHours)
- Update Anomaly:
  - Changing the name of project number P1 from “Billing” to “Customer-Accounting” requires an update for every employee who works on project P1.

Insert Anomaly Example

- Consider the relation:
  - EMP_PROJ(<u>EmpNum</u>, <u>ProjNum</u>, Ename, Pname, NumHours)

- Insert Anomalies:
  - Cannot insert a project
    unless an employee is assigned to that project.
  - Cannot insert an employee
    unless that employee is assigned to a project.

Delete Anomaly Example

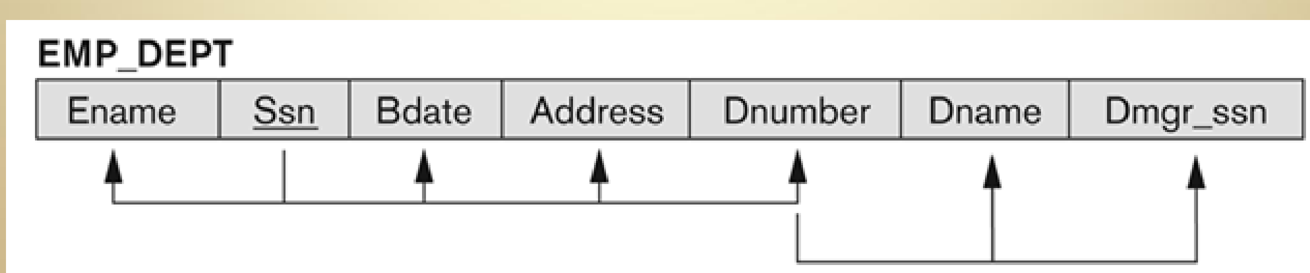
- Consider the relation:
  - EMP_PROJ(<u>EmpNum</u>, <u>ProjNum</u>, Ename, Pname, NumHours)
- Delete Anomaly:
  - When a project is deleted, it will result in deleting all the employees who work (only) on that project.
  - When an employee is the sole employee on a project, deleting that employee results in deleting the corresponding project.

Redundancy Example

- Where's the redundancy?

EMP_DEPT

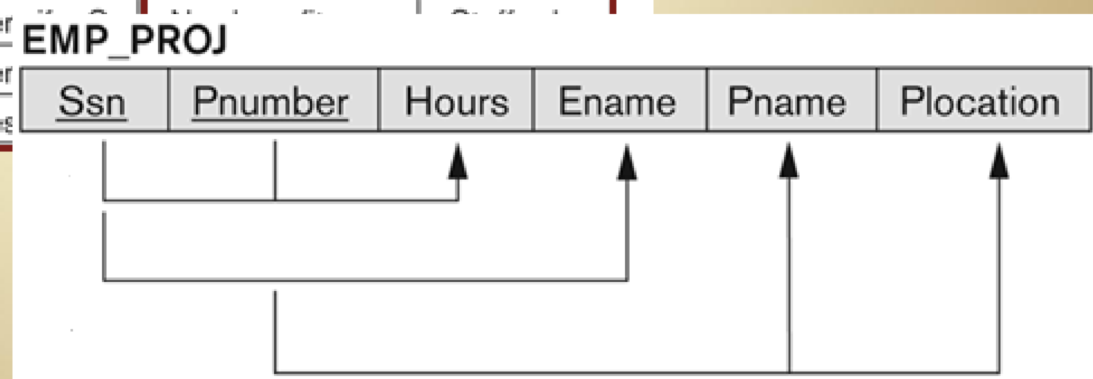
Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
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 Example

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, Jer [illegible]	[illegible]	[illegible]
987654321	20	15.0	Wallace, Jer [illegible]	[illegible]	[illegible]
888665555	20	Null	Borg, James [illegible]	[illegible]	[illegible]



Reduce Number of Null Values

- Null values should be used only for special cases
  - nulls should not be a "normal" value
- If some attributes are expected to frequently be null, move them to a separate relation
  - the new relation will only have tuples for the non-null cases
- Reasons for nulls:
  - Attribute not applicable or invalid
  - Attribute value unknown
  - Value known to exist, but it is currently unavailable

NORMALIZATION

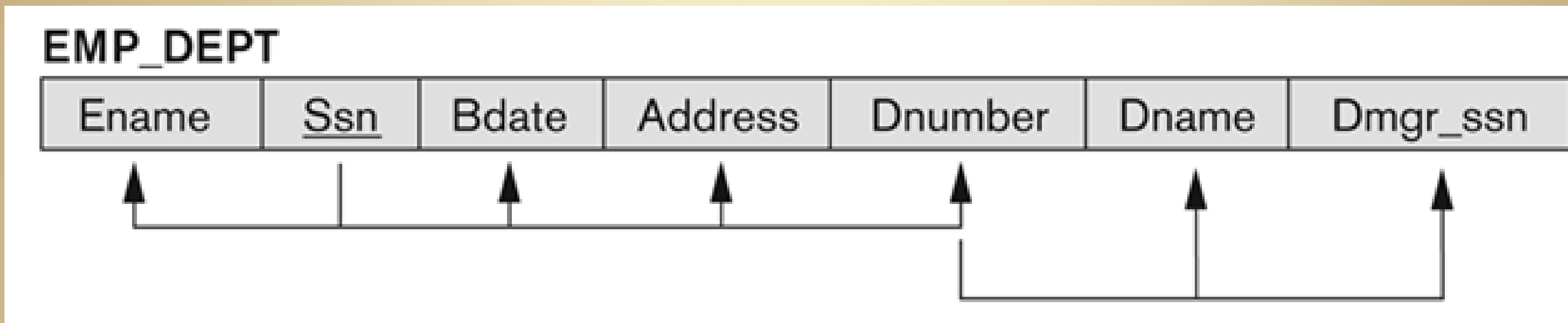
Theory for Good Design

- Previous discussion introduced
  informal guidelines for good schema design

- Remaining discussion introduces
  a formal theory for achieving good designs

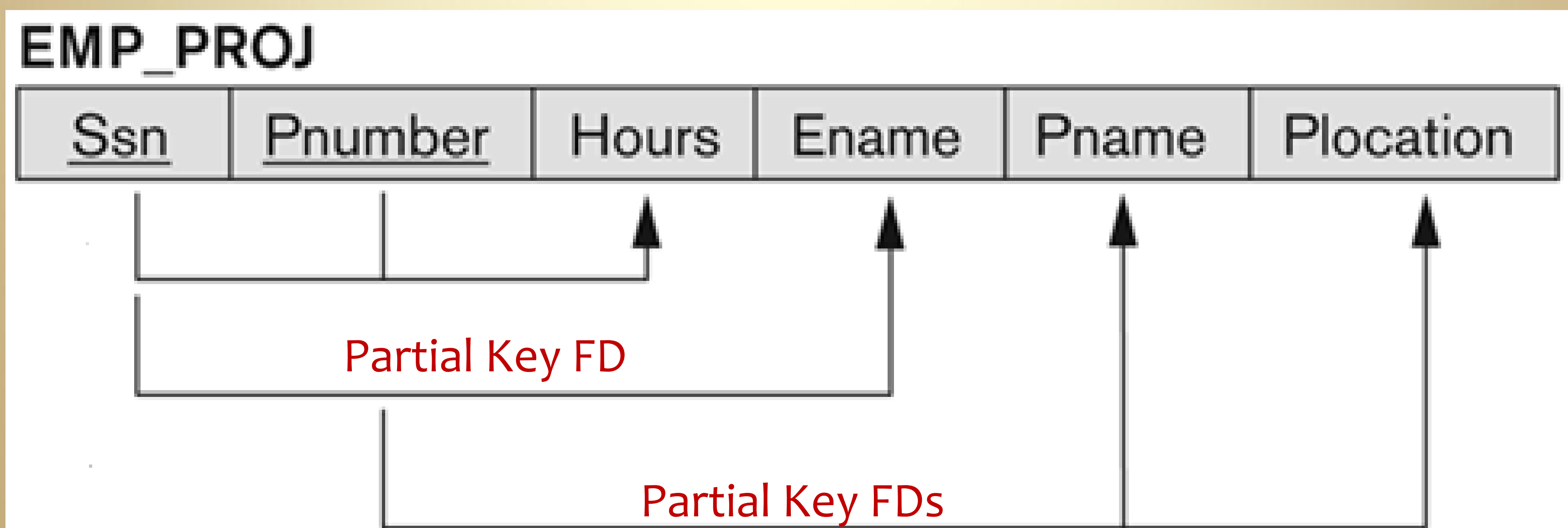
  - ***Functional Dependencies***:
    determine "true" relationships between attributes
  - ***Normal Forms***:
    define schemas in which the quality measurements are satisfied

Example FDs

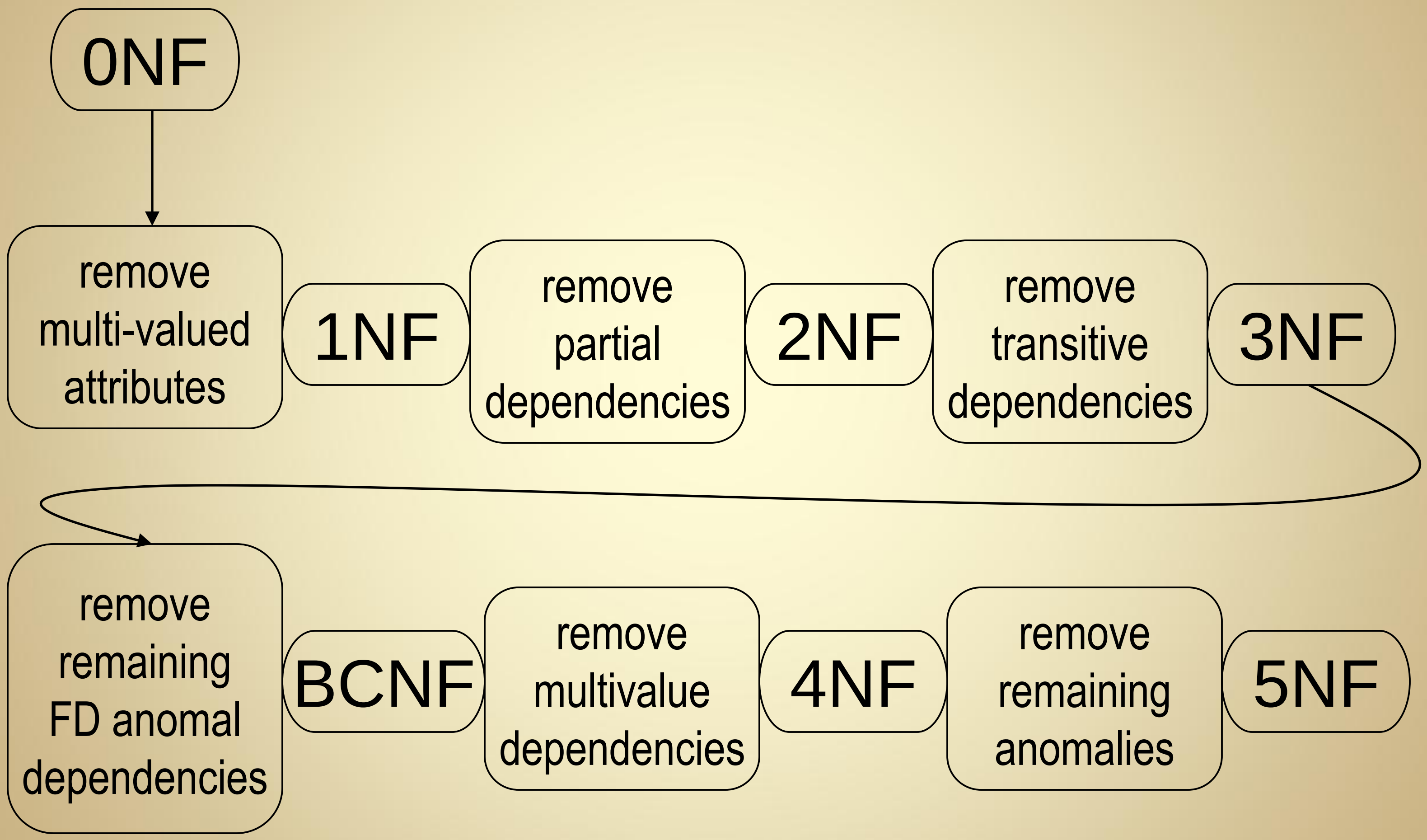


Proper FDs

Transitive FDs

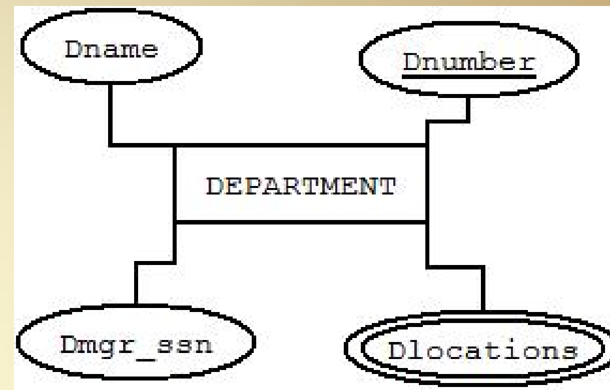


Normalization



1NF Normalization

DEPARTMENT



Proper translation from ER multi-value attributes will achieve 1NF.

DEPARTMENT

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

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

Still not a good solution, since we have redundancy in Dnumber and Dmgr_ssn. (This will be handled by 2NF.)

FD: Formal Definition

- A *functional dependency* (FD) is a constraint between two sets of attributes
- Let X, Y be two sets of attributes, such that

  if $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$

  - the value of X *determines* the value of Y
  - there exists a function Y = f(X)
- We denote FDs by

$$X \rightarrow Y$$

Example FDs

(BLDG, ROOM) → (CAPACITY, NUM_WINDOWS)

proper FD

(THEATER, SCREEN, TIME) → (MOVIE)

proper FD

partial key FD

(STORE_ID, ITEM_ID) → (QUANTITY_STOCKED, STORE_ADDRESS)

proper FD

transitive FD

(ISBN_NUMBER) → (PUBLISHER, PUB_PHONE, TITLE)

proper FD

proper FD

Partial Dependencies

- $X \rightarrow Y$ is a *partial dependency* (PD)

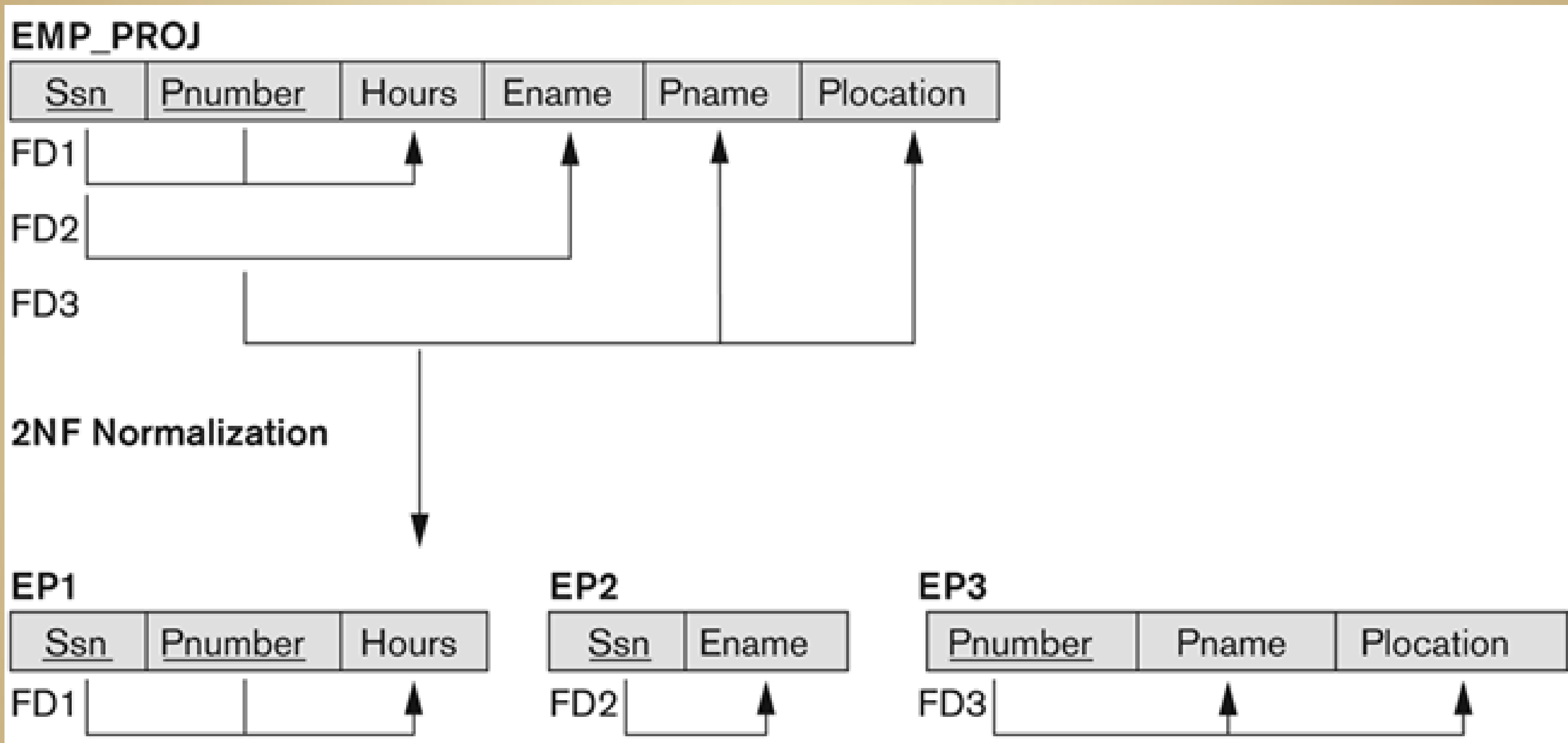
  if there exists $X' \subset X$

  such that $X' \rightarrow Y$

- PFDs lead to redundancy: there are multiple values of X that determine the same value of Y

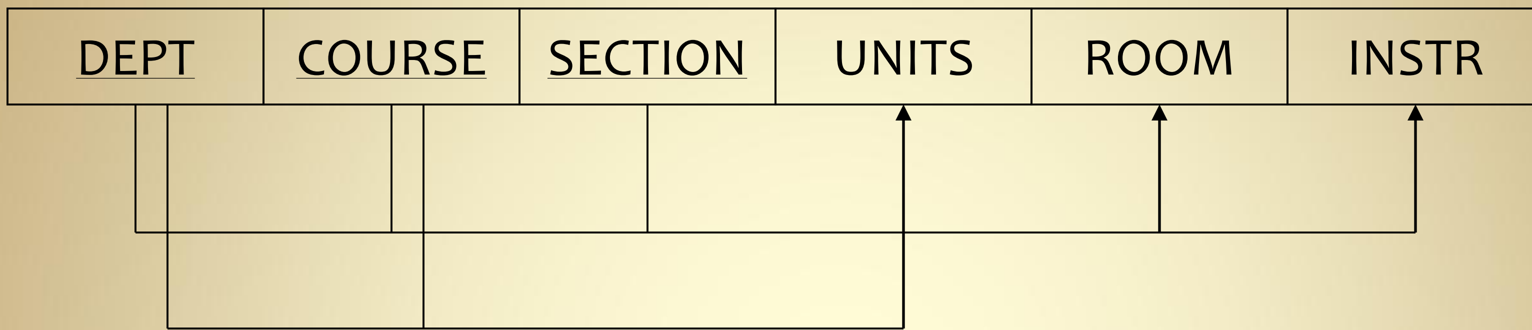
- 2NF normalization: move X' and Y to a new relation in which X' is the primary key

2NF Normalization



Move the partial key and dependent attributes to a new relation.

Partial Dependency Example



UNITS depends only on the partial key (DEPT,COURSE)

Following slides show update anomalies that will occur if we do not correct this problem.

Insert Anomaly

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

redundant information
must be copied

```
insert into COURSES values
("COMP",51,3,4,"KNOLES210","GAO")
```

Insert Anomaly

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

How do we add a course that currently has no sections?

```
insert into COURSES values
("COMP",179,NULL,3,NULL,NULL)
```

nulls are not allowed
in primary key

Delete Anomalies

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

What happens if the only section of a course is deleted?

```
DELETE FROM COURSES
WHERE DEPT = "COMP"
  AND COURSE = 163
  AND SECTION = 1
```

Course units information is also lost

Update Anomaly

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

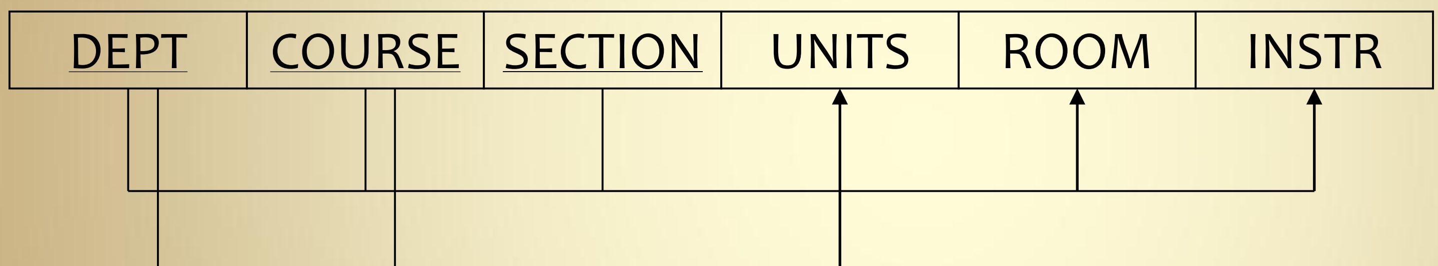
What if we change the units for a course?

```
UPDATE COURSES
SET UNITS = 3
WHERE DEPT = "COMP" AND COURSE = 51
```

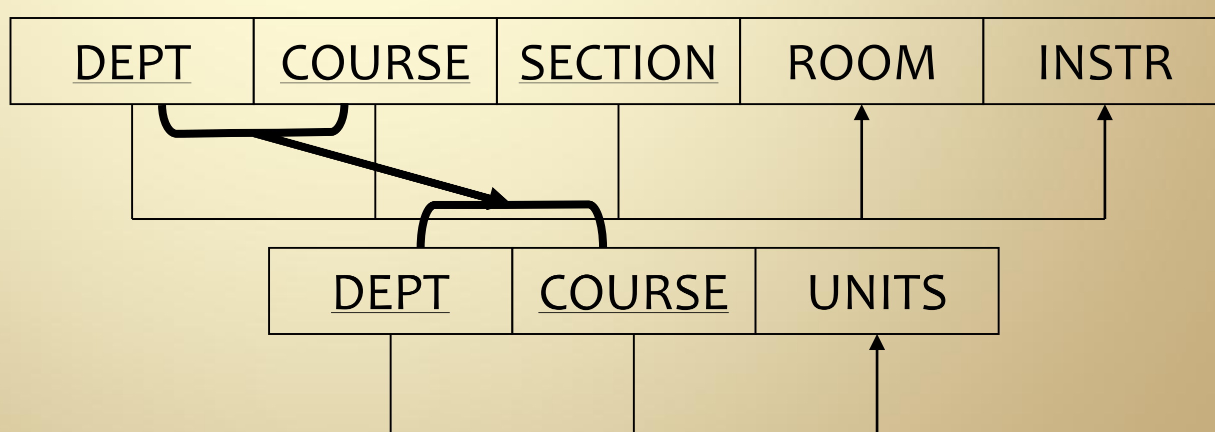
Must be careful to modify all sections

NF Decomposition: Foreign Keys

- NF decomposition results in new foreign keys



Decomposition:



2NF Decomposition

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

PDs are removed

anomalies are avoided

DEPT	COURSE	SECTION	ROOM	INSTR
COMP	51	1	WPC122	DOHERTY
COMP	51	2	WPC219	CLIBURN
COMP	163	1	WPC122	DOHERTY
COMP	53	1	WPC130	BOWRING
COMP	53	2	WPC130	CARMAN

DEPT	COURSE	UNITS
COMP	51	4
COMP	163	3
COMP	53	4

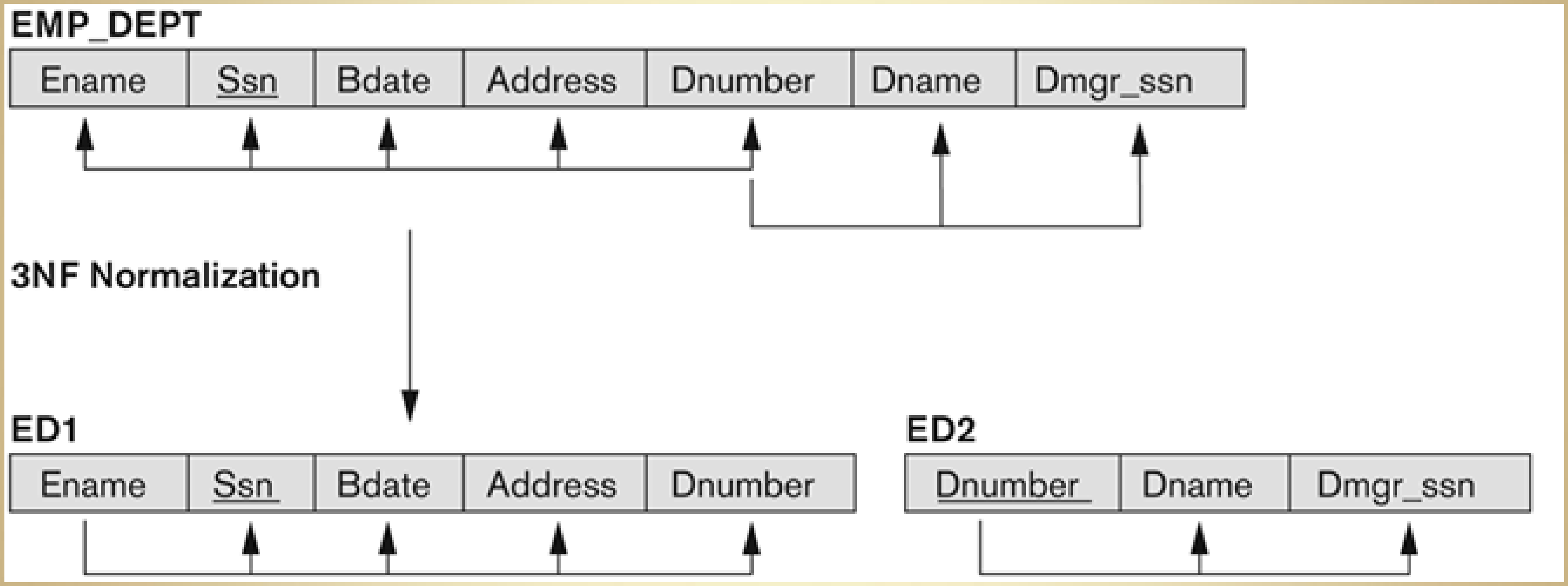
Transitive Dependencies

- $X \rightarrow Y$ is a *transitive dependency* (PD)

  if there exists $Z \nsubseteq$ any key

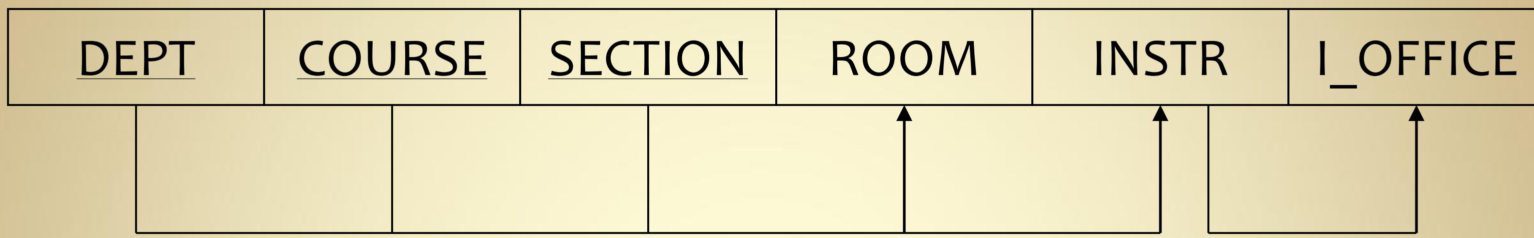
  such that $X \rightarrow Z \rightarrow Y$
- TDs can cause redundancy if there are multiple values of X that determine the same value of Z
  - the value of Y for that value of Z is stored multiple times
- 3NF normalization: move (Z,Y) to new relation in which Z is the primary key

3NF Normalization



- Create new relation to hold the attributes in the transitive FD.
- LHS of transitive FD becomes PK of new relation.

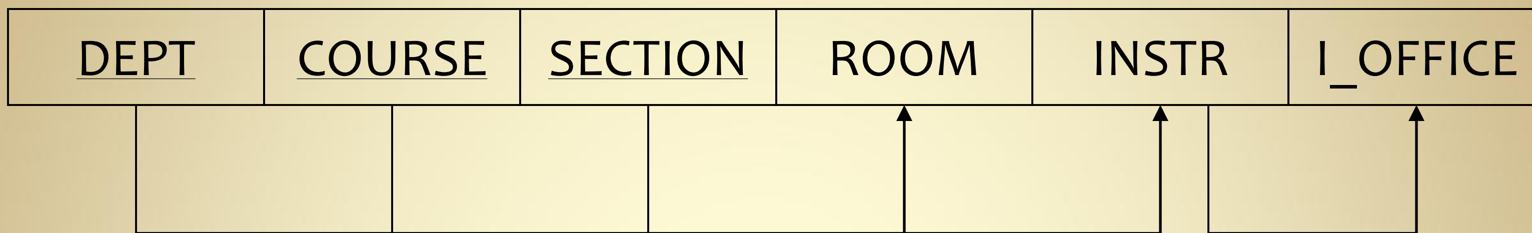
Transitive Dependency Example



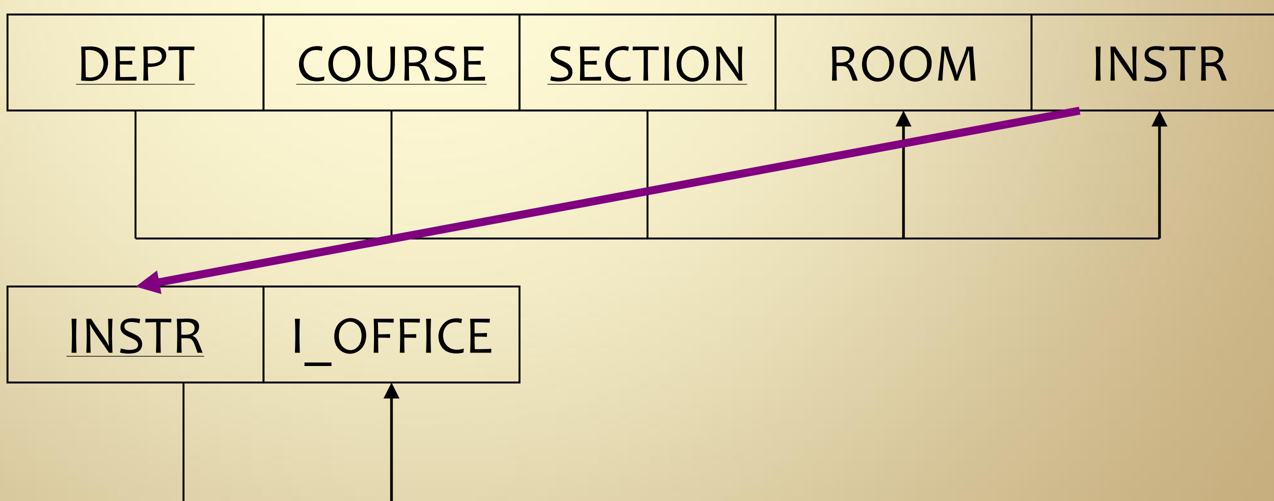
I_OFFICE (instructor's office) is determined
by the non-PK attribute INSTR

DEPT	COURSE	SECTION	ROOM	INSTR	I_OFFICE
COMP	51	1	WPC122	DOHERTY	CSB109
COMP	51	2	WPC219	CLIBURN	CSB107
COMP	163	1	WPC122	DOHERTY	CSB109
COMP	53	1	WPC130	BOWRING	CSB108
COMP	53	2	WPC130	CARMAN	CSB104

NF Decomposition: Foreign Keys



Decomposition:



3NF Decomposition

DEPT	COURSE	SECTION	ROOM	INSTR	I_OFFICE
COMP	51	1	WPC122	DOHERTY	CSB109
COMP	51	2	WPC219	CLIBURN	CSB107
COMP	163	1	WPC122	DOHERTY	CSB109
COMP	53	1	WPC130	BOWRING	CSB108
COMP	53	2	WPC130	CARMAN	CSB104

DEPT	COURSE	SECTION	ROOM	INSTR
COMP	51	1	WPC122	DOHERTY
COMP	51	2	WPC219	CLIBURN
COMP	163	1	WPC122	DOHERTY
COMP	53	1	WPC130	BOWRING
COMP	53	2	WPC130	CARMAN

INSTR	I_OFFICE
CLIBURN	CSB107
DOHERTY	CSB109
BOWRING	CSB108
CARMAN	CSB104

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**

Further Normalization

- BCNF
  - based on keys and FDs of a relation schema
- 4NF
  - based on keys, multi-valued dependencies
- 5NF
  - based on keys, join dependencies
- Additional properties may be needed
  - lossless join, dependency preservation
- *For this course*, we'll assume that 3NF is sufficient

Caution: False FDs

DEPT	COURSE	SECTION	UNITS	ROOM	INSTR
COMP	51	1	4	WPC122	DOHERTY
COMP	51	2	4	WPC219	CLIBURN
COMP	163	1	3	WPC122	DOHERTY
COMP	53	1	4	WPC130	BOWRING
COMP	53	2	4	WPC130	CARMAN

?

(INSTR) → (ROOM)?

No, it is only a coincidence that Doherty teaches both classes in the same room

Determining FDs

- Functional dependencies are determined by the *semantics of the schema*, not by a particular database state (an FD is a *constraint*)

- A database state can prove that something is not an FD

- A database state can only indicate that something might be an FD

Exercises

- Assume that the relation R(A, B, C, D)
  is in first normal form.
  Is it possible that it is not in second normal form?
  Explain your answer.

Exercises

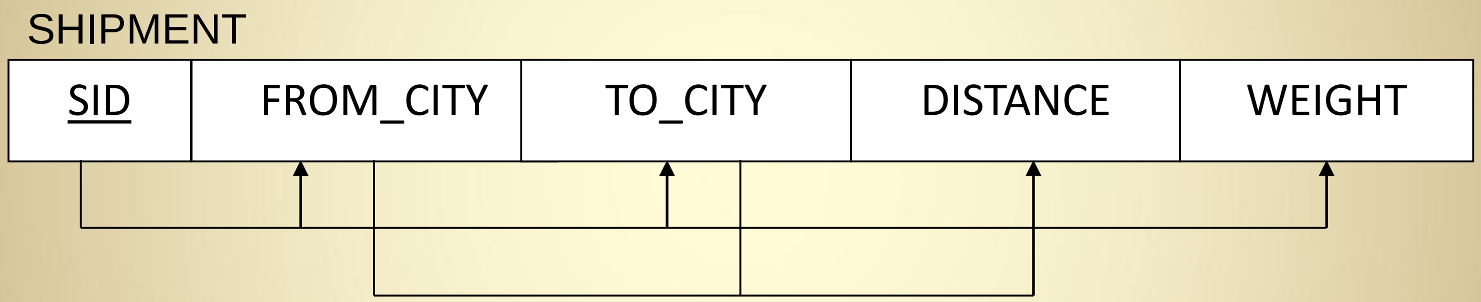
- The following instance of the relation T(U, V, W, X) indicates that there may be a functional dependency { U } -> { X }.

  Add a tuple to the relation that would show that this is not a true functional dependency.

U	V	W	X
12	34	21	65
29	52	56	34
13	34	78	72
12	45	42	65

Exercises

- The following relation schema is not in third normal form (3NF).



Is this an example of a *transitive dependency* or a *partial key dependency*?

Give an equivalent schema that is in 3NF.

Exercises

- This relation has been proposed
  to track Pacific alumni:

Alumni(SID, LastName, FirstName, Degree,
YearAwarded, Phone).

Pacific allows students to receive multiple degrees,
possibly in different years.
Identify all FDs.

Give a new schema that is in third normal form.

Exercises

- Consider the following relation schema:

Movie(title, genre, length, actor, sag_id, studio, studio_addr)

  - Every movie has a unique title.
  - A movie may have multiple actors.
  - Each actor has a unique sag_id.
  - An actor may appear in multiple movies.
  - A movie has exactly one studio,
    but a studio may produce more than one movie.
  - Each studio has exactly one address.

- Identify all functional dependencies.

- Normalize the schema to 3NF.