WEEK 9

SECOND AND THIRD NORMAL FORM

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
    - Given the functional dependencies, put a table in second normal form
    - Given the functional dependencies, put a table in third normal form
    - Given the functional dependencies, identify the minimal prime attributes for a table.

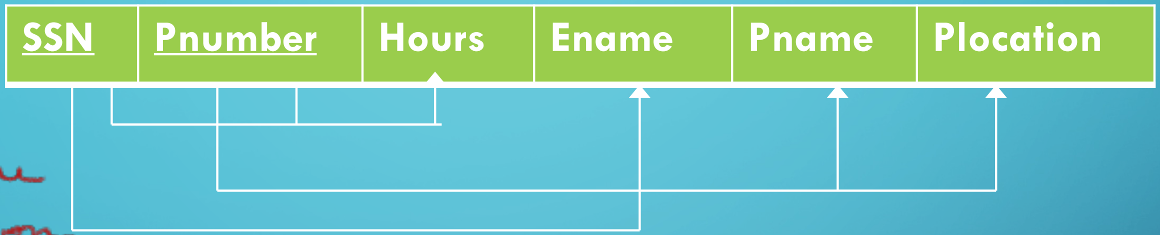
SECOND NORMAL FORM

- A relation R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R OR if every nonprime attribute A in R is not partially dependent on any key in R.

- To convert from 1NF to 2NF, start with the 1NF format and write the key components on separate line and write the original key on the last line. Then each of these components will become the key in a new table. Then write the dependent attributes after each of the new keys.

EXAMPLE:

- The following relation is in first normal form but not second normal form:



SSN	Pnumber	Hours	Ename	Pname	Plocation

the key is SSN+ Pnumber. But not any attribute is fully dependent on a single key.

- 2NF for this table would consist of the following 3 tables:

EMPLOYEE:

SSN	Ename

PROJECT:

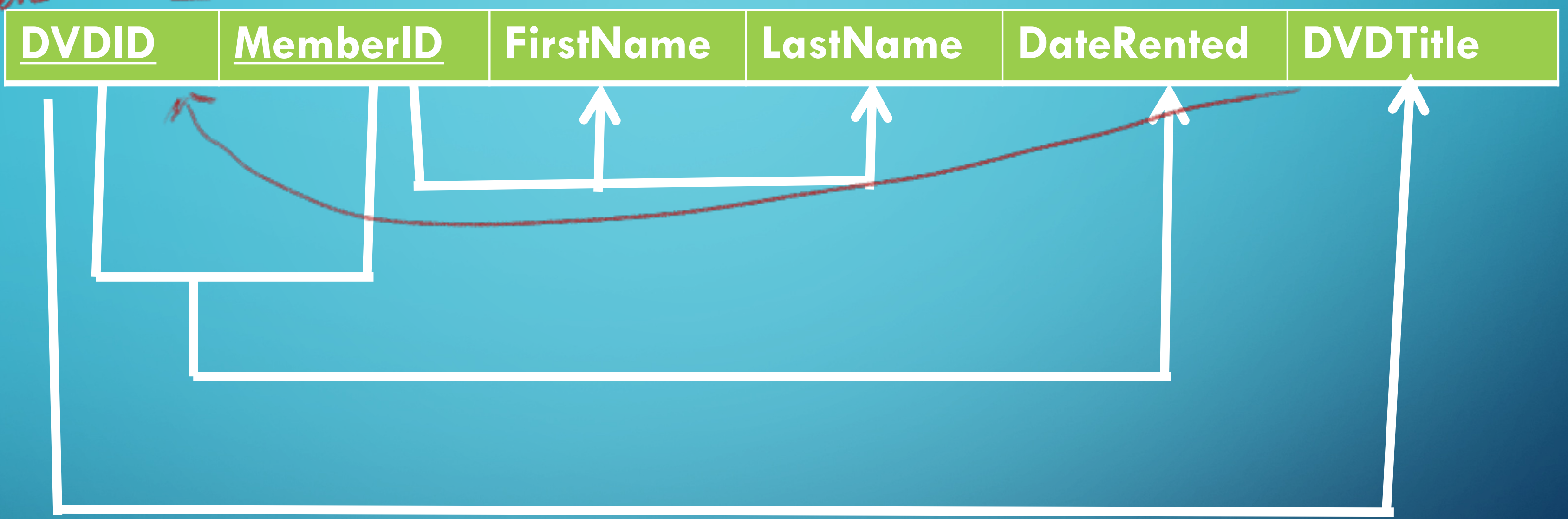
Pnumber	Pname	Plocation

PROJEMP:

SSN	Pnumber	Hours

QUESTION: Draw the functional dependencies on the following table and then put the table in 2NF

they are partial dependent on key.



DVDID	MemberID	FirstName	LastName	DateRented	DVDTitle

2NF

DVD

DVDID	DVDTitle

CUSTOMER

MemberID	FirstName	LastName

RENTED

DVDID	MemberID	DateRented

11/15/23

THIRD NORMAL FORM

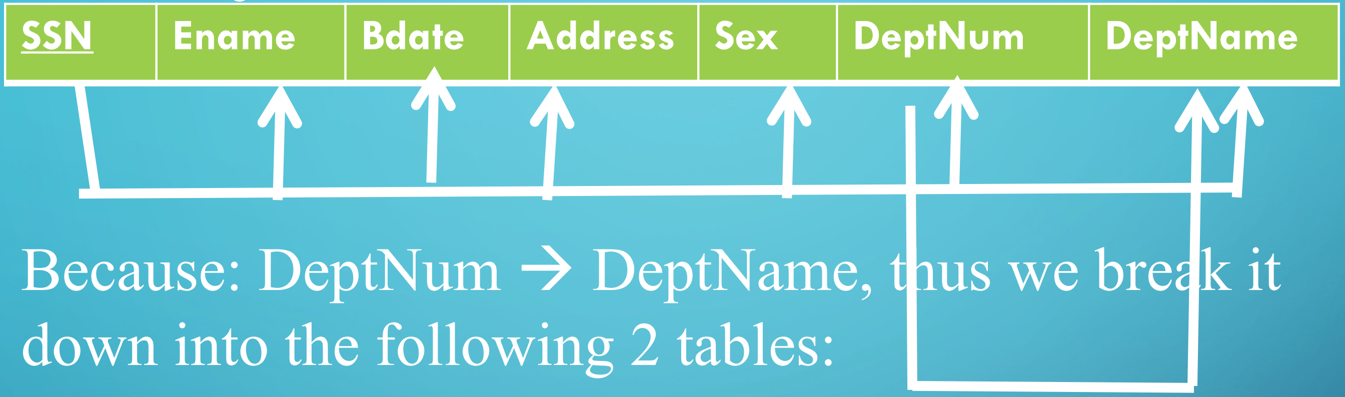
- A relation schema R is in 3NF if it is ① in 2NF and it ② contains no transitive dependencies (if you have a nonprime attribute functionally dependent on another nonprime attribute)

- To convert from 2NF to 3NF break off the piece(s) that are identified as transitive dependencies and store them in a separate table.

EXAMPLE:

there's a non-prime attribute determined by another non-prime attribute.

- The following table is in 2NF but not 3NF:



<u>SSN</u>	Ename	Bdate	Address	Sex	DeptNum	DeptName

Because: DeptNum → DeptName, thus we break it down into the following 2 tables:

Department:

DeptNum	DeptName

SSN -> DeptNum -> DeptName

Employee:

<u>SSN</u>	Ename	Bdate	Address	Sex	DeptNum

QUESTION: How would you break the following table into 3NF?

WorkerID	Ename	SkillType	BonusRate
2343	Skinner	Electric	3.00
3434	Arman	Plumbing	3.50
4545	Seymour	Electric	3.00
6767	Burns	Electric	3.00

Skill:

SkillType	Bonus

Worker:

WorkerID	Ename	SkillType

QUESTION: Assume you have the following relation R={A,B,C,D,E,F,G,H,I,J,K,L,M} and the set of functional dependencies: {{A,B,C}→{K},{A} →{D,F},{B}→{E}, {C}→{G,H}, {K}→{I,J}, {A,B}→{L,M}}

transitive dependency.

What is the key for R? **A,B,C**

Decompose R into 2NF

Table1:

A	B	C	K	I	J

Table2

A	B	L	M

Table3

A	D	F

Table4

C	G	H

Table5

B	E

Decompose R into 3NF

Table1:

A	B	C	K

Table2

A	B	L	M

Table6

K	I	J

Table3

A	D	F

Table4

C	G	H

Table5

B	E