Database Engineering

Lecture #9

Functional Dependencies and Normalization for Relational Databases

Presented By:

Dr. Suvasini Panigrahi

Associate Professor, Department of CSE,
VSSUT, Burla

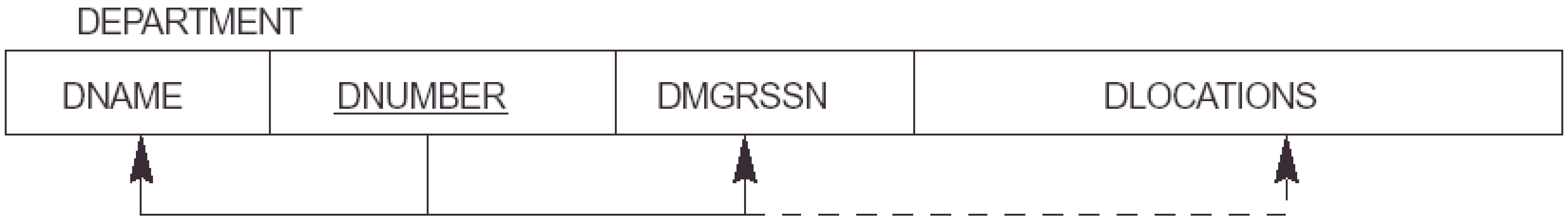
Introduction to Normalization

- **Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies
- This process decomposes unsatisfactory large relations by breaking up their attributes into smaller relations
- **Normal form**: Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form
  - 2NF, 3NF, BCNF based on keys and FDs of a relation schema
  - 4NF based on keys, multi-valued dependencies
  - 5NF based on keys, join dependencies

First Normal Form (1NF)

- Disallows composite attributes, multivalued attributes, and their combinations
- As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values (single valued attributes).
- It should hold only atomic values.

(a)



(b)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

(a) **EMP_PROJ**

SSN	ENAME	PROJS	
		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

(c) **EMP_PROJ1**

SSN	ENAME

EMP_PROJ2

SSN	PNUMBER	HOURS

Second Normal Form (2NF)

- Uses the concepts of FDs and primary key.
- **Important Definitions:**
  - **Prime Attributes -** Candidate keys are also referred to as primary keys, secondary keys or alternate keys. The constituent **attributes** are called **prime attributes**.
  - **Non-prime Attribute -** Conversely, an **attribute** that does not occur in any of the candidate keys is called a **non-prime attribute**
  - **Full Functional Dependency (FFD)** - A functional dependency Y → Z is FFD where removal of any attribute from Y means the FD does not hold any more
  - **Partial Functional Dependency -** Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key
- The 2nd Normal Form (2NF) eliminates the Partial Dependency.

Second Normal Form (2NF)

- A relation is said to be in 2NF if the following conditions hold:
    - Table is in 1NF (First Normal Form)
    - It does not have any non-prime attribute that is functionally dependent on any proper subset of any candidate key of the relation.

Examples - Second Normal Form

- {SSN, PNUMBER} → HOURS is a full FD since neither SSN → HOURS nor PNUMBER → HOURS hold
- {SSN, PNUMBER} → ENAME is *not* a full FD (it is called a *partial dependency*) since SSN → ENAME also holds
- **A relation schema R is in second normal form (2NF) if every non-prime attribute A in R is fully functionally dependent on the primary key**
- R can be decomposed into 2NF relations via the process of 2NF normalization

Examples - Second Normal Form

- **Example**: Suppose a school wants to store the data of teachers and the subjects they teach. They create a table that looks like this: Since a teacher can teach more than one subjects, the table can have multiple rows for a same teacher.

teacher_id	subject	teacher_age
111	Maths	38
111	Physics	38
222	Biology	38
333	Physics	40
333	Chemistry	40

Candidate Keys: {teacher_id, subject}
Non prime attribute: teacher_age

Examples - Second Normal Form

- The table is in 1 NF because each attribute has atomic values.
- However, it is not in 2NF because non prime attribute teacher_age is dependent on teacher_id alone which is a proper subset of candidate key.
- This violates the rule for 2NF as the rule says “**no non-prime attribute is dependent on the proper subset of any candidate key of the table**”.

Examples - Second Normal Form

- To make the table in 2NF, we can break it in two tables in the following manner:

teacher_details table:

teacher_id	teacher_age
111	38
222	38
333	40

teacher_subject table:

teacher_id	subject
111	Maths
111	Physics
222	Biology
333	Physics
333	Chemistry

Third Normal Form (3NF)

- A relation is in third normal form, if there is no **transitive dependency** for non-prime attributes as well as it is in second normal form.

Definition:

- **Transitive functional dependency** - A FD X -> Z that can be derived from two FDs X -> Y and Y -> Z

  Examples:

  SSN -> DMGRSSN is a *transitive* FD if

  SSN -> DNUMBER and DNUMBER -> DMGRSSN hold

  SSN -> ENAME is *non-transitive* if there is no set of attributes X where SSN -> X and X -> ENAME

Third Normal Form (3NF)

- A relation schema 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
- A relation schema R is in **third normal form** (**3NF**) if whenever a FD X -> A holds in R, then either:

    (a) X is a superkey of R, or

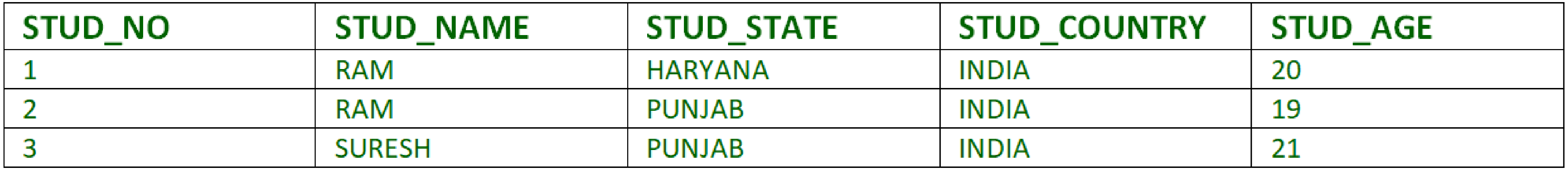
    (b) A is a prime attribute of R

Third Normal Form (3NF)

- R can be decomposed into 3NF relations via the process of 3NF normalization
- The normalization of 2NF relations to 3NF involves the removal of transitive dependencies.
- If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant.
- A determinant in a database table is any attribute that you can use to determine the values assigned to other attribute(s) in the same row.

Example of Third Normal Form

- In relation STUDENT given in the following table:



STUD_NO	STUD_NAME	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	HARYANA	INDIA	20
2	RAM	PUNJAB	INDIA	19
3	SURESH	PUNJAB	INDIA	21

- FD set:
  {STUD_NO -> STUD_NAME, STUD_NO -> STUD_STATE, STUD_STATE -> STUD_COUNTRY, STUD_NO -> STUD_AGE}
- Candidate Key:
  {STUD_NO}

Example of Third Normal Form

- For this STUDENT relation, STUD_NO -> STUD_STATE and STUD_STATE -> STUD_COUNTRY are true.
- Hence, STUD_COUNTRY is transitively dependent on STUD_NO. It violates the 3NF.
- To convert it in third normal form, we will decompose the relation STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_COUNTRY_STUD_AGE) as:

BCNF (Boyce-Codd Normal Form)

- BCNF is the advanced version of 3NF. It is stricter than 3NF.
- A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X -> A holds in R, then X is a super key of R
- A table is in BCNF the following condition holds:
  - **For every functional dependency X → Y, X is the super key of the relation.**
  - **The relation should be in 3NF.**
- Each normal form is strictly stronger than the previous one:
  - Every 2NF relation is in 1NF
  - Every 3NF relation is in 2NF
  - Every BCNF relation is in 3NF
- Every relation in BCNF is also in 3NF, but the reverse is not necessarily true. There exist relations that are in 3NF but not in BCNF

Example of a relation which is in 3NF but not in BCNF

- The simplest relation which violates BCNF but meets 3NF can be defined as shown below
- R(A, B, C), F = {AB -> C, C -> A}
- In this case, the super key is (A,B).
- It meets 3NF because the left-hand side of AB -> C is a super key and the right-hand side of C -> A is a primary attribute.
- It violates BCNF because in the FD C -> A, the left-hand-side is not a super key.

Example of Boyce-Codd Normal Form

Consider the following relationship : **R (A,B,C,D)**

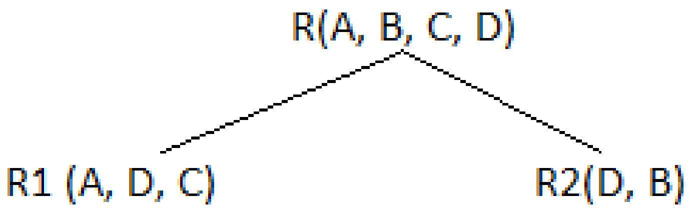
and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2.**



R(A, B, C, D)

R1 (A, D, C)

R2(D, B)

Breaking, table into two tables, one with A, D and C while the other with D and B.

Multivalued Dependency in DBMS

- Multivalued Dependency (MVD) means that for a single value of an attribute ‘a’, multiple values of attribute ‘b’ exist. We write it as follows:

  a -->--> b

- It is read as **‘a’ multidetermines ‘b’**.
- Multivalued Dependency would occur whenever two separate attributes in a given table are independent of each other, but both of these depend on another third attribute.
- The multivalued dependency contains at least two of the attributes dependent on the third attribute.
- This is the reason why it a relation with MVD consists of at least three attributes.

Conditions for MVD in a Relation

- A relation is said to have MVD, if the following conditions are true:
  1. For a dependency A → B, if for a single value of A, multiple value of B exists, then the relation is said to have multi-valued dependency.
  2. Also, a relation should have at least 3 columns in it in order to have a multi-valued dependency.
  3. For a relation schema R(A,B,C), if there is a multi-valued dependency between, A and B, and A and C then B and C should be independent of each other.
- If all these conditions are true for any relation (table), it is said to have multi-valued dependency.

Example1 of MVD

- Suppose that there is a car manufacturing company that produces two of the colours in the market, i.e., red and yellow for each of their models every year.

CAR_MODEL	MANUF_MONTH	COLOUR
S2011	JAN	Yellow
S2001	FEB	Red
S3001	MAR	Yellow
S3001	APR	Red
S4006	MAY	Yellow
S4006	JUN	Red

•In this case, the columns COLOUR and MANUF_MONTH are dependent on CAR_MODEL, and they are independent of each other.

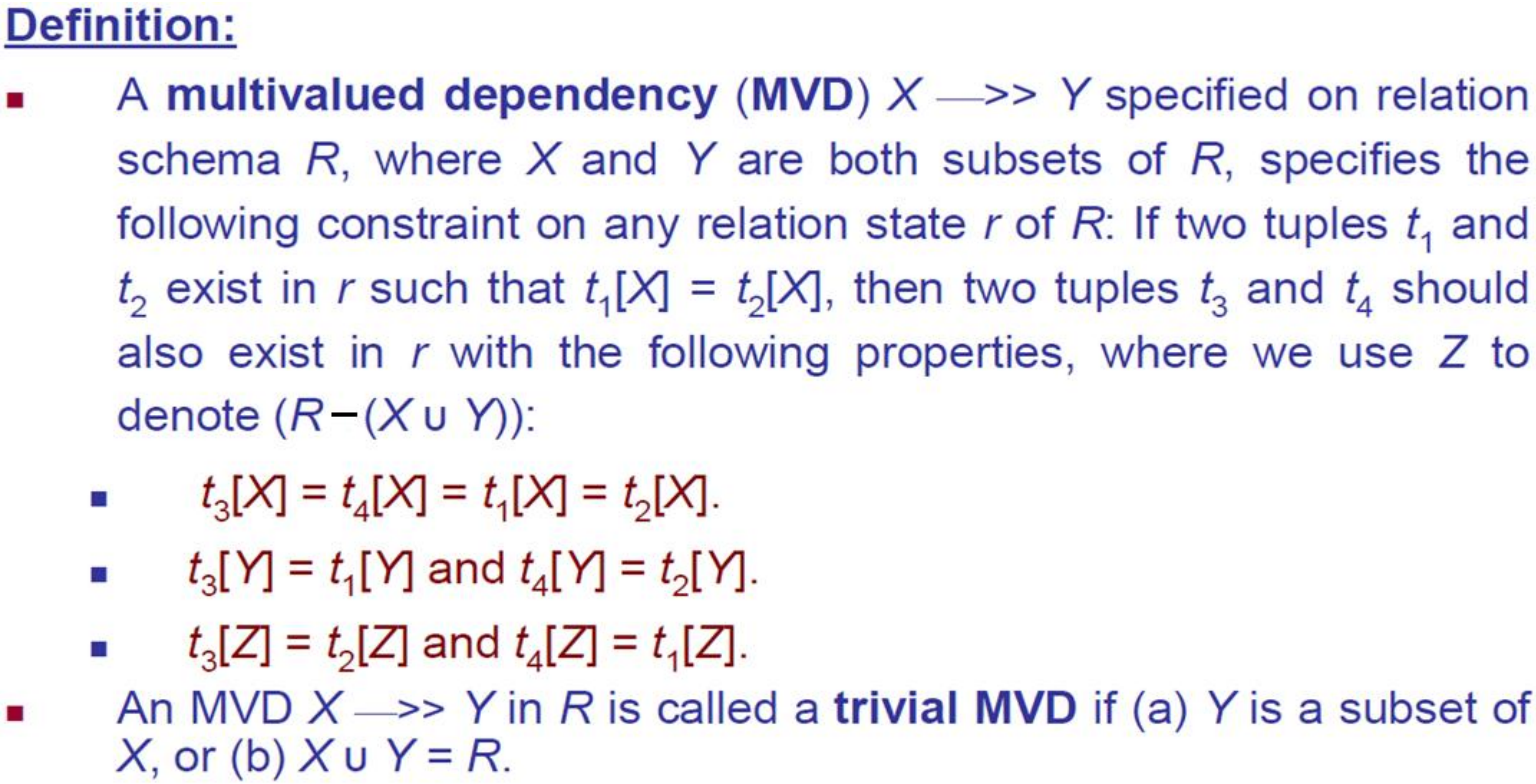
•But they are dependent on the attribute CAR_MODEL.

•The representation of the dependencies we discussed above is as follows:

CAR_MODEL →→ MANUF_MONTH
CAR_MODEL →→ COLOUR

•This can be read as:
“CAR_MODEL multidetermines MANUF_MONTH”
“CAR_MODEL multidetermines COLOUR”

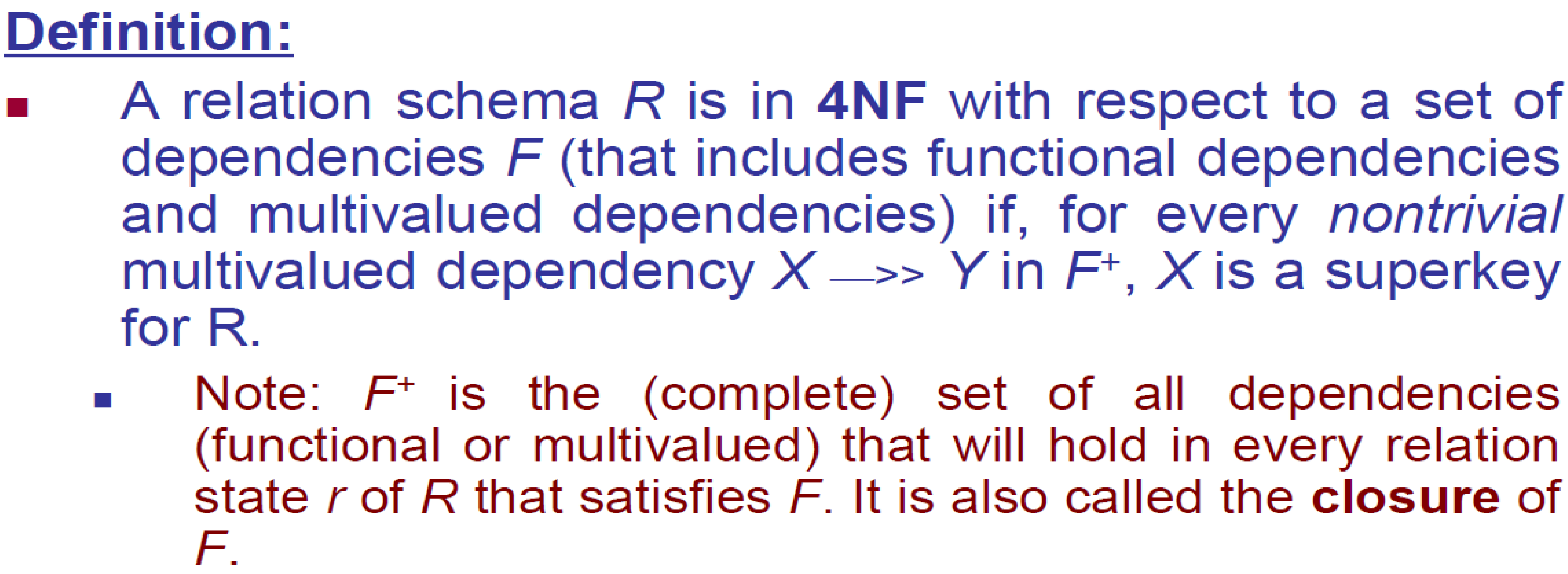
Multivalued Dependency and Fourth Normal Form



Definition:

- A **multivalued dependency** (**MVD**) X —>> Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation state r of R: If two tuples t_1 and t_2 exist in r such that $t_1[X] = t_2[X]$, then two tuples t_3 and t_4 should also exist in r with the following properties, where we use Z to denote $(R - (X \cup Y))$:
  - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$.
  - $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$.
  - $t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$.
- An MVD X —>> Y in R is called a **trivial MVD** if (a) Y is a subset of X, or (b) $X \cup Y = R$.

Multivalued Dependency and Fourth Normal Form

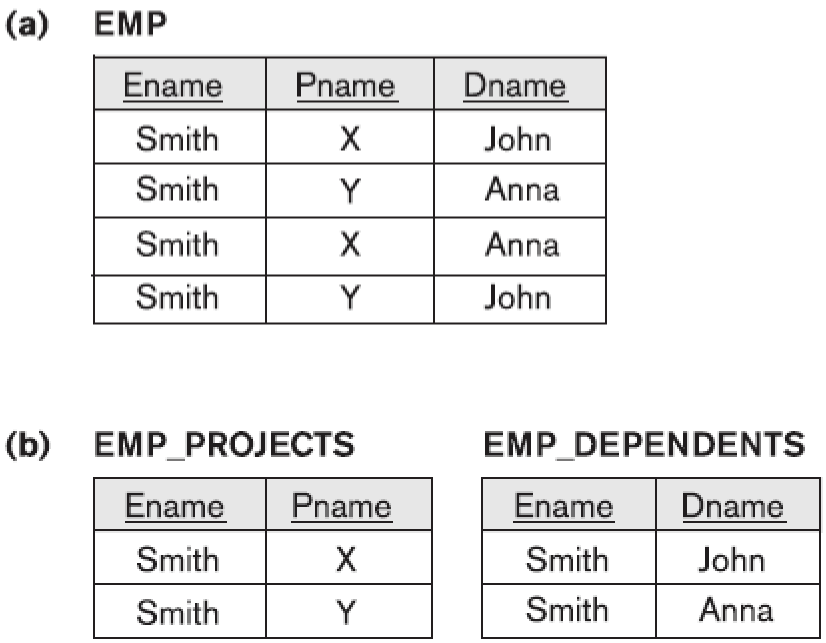


Definition:

- A relation schema *R* is in **4NF** with respect to a set of dependencies *F* (that includes functional dependencies and multivalued dependencies) if, for every *nontrivial* multivalued dependency $X \twoheadrightarrow Y$ in F^+, *X* is a superkey for R.
  - Note: F^+ is the (complete) set of all dependencies (functional or multivalued) that will hold in every relation state *r* of *R* that satisfies *F*. It is also called the **closure** of *F*.

- For a relation R to be in 4NF, it must meet two conditions –
  - It should be in Boyce-Codd Normal Form (BCNF).
  - It should not have any non-trivial multivalued dependencies.

Example of Multivalued Dependency and Fourth Normal Form



(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b) EMP_PROJECTS

Ename	Pname
Smith	X
Smith	Y

EMP_DEPENDENTS

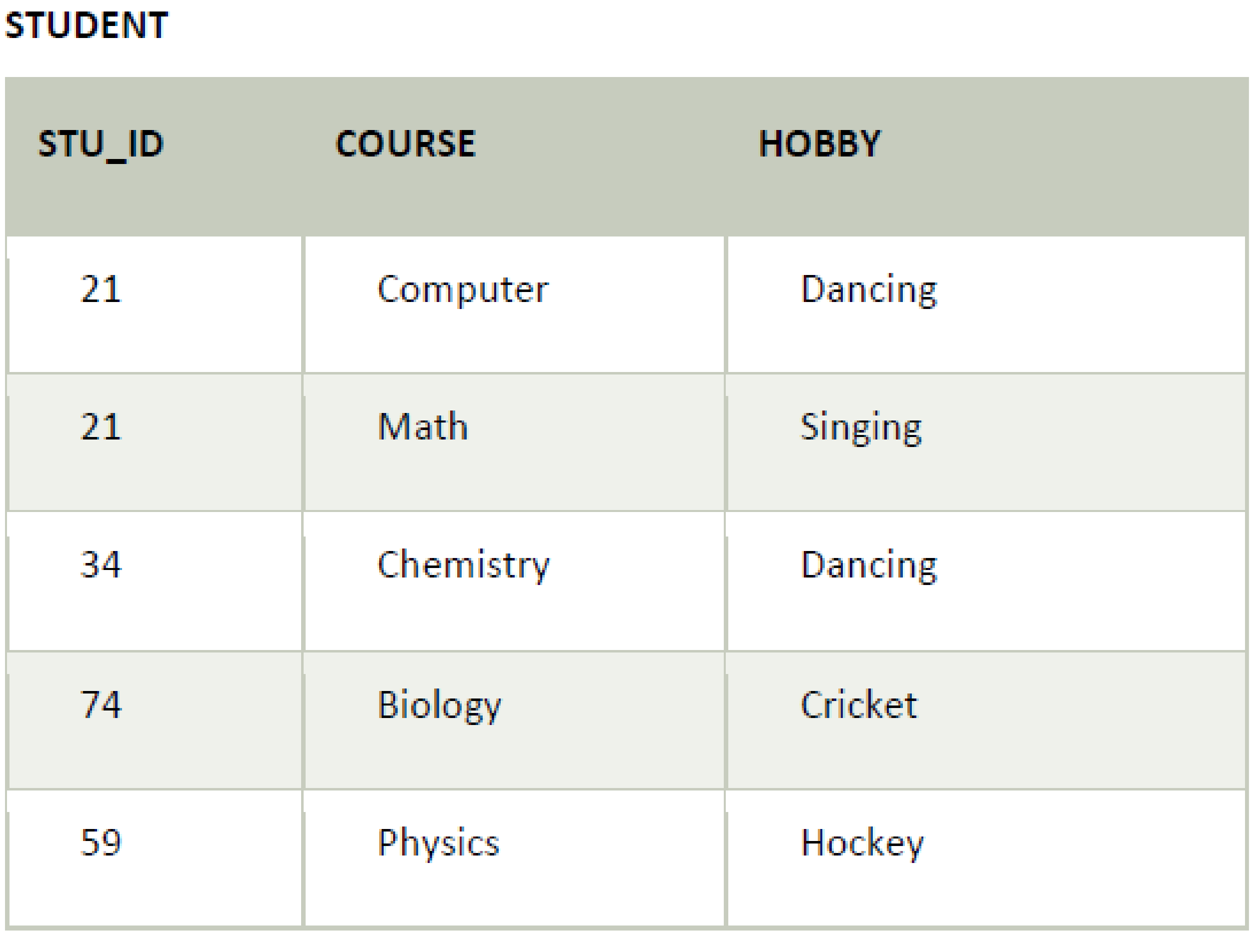
Ename	Dname
Smith	John
Smith	Anna

FOURTH NORMAL FORM

(a) The EMP relation with two MVDs: Ename $\twoheadrightarrow$ Pname and Ename $\twoheadrightarrow$ Dname.
(b) Decomposing the EMP relation into two 4NF relations EMP_PROJECTS and EMP_DEPENDENTS.

- A relation with only trivial MVDs and no non-trivial MVDs is in fourth normal form (4NF).
- Fourth Normal Form (4NF) is a level of database normalization that requires a relation to be in BCNF and have no non-trivial MVD, to eliminate redundant data and maintain data consistency.
- **Explanation:** A relation is in 4NF if all of its non-trivial MVDs are super keys, which means that the combination of all attributes in the MVD is a superset or candidate key.

Example of MVD and Fourth Normal Form



STUDENT

STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

- There is no relationship between “COURSE” and “HOBBY”. But both are dependent on “STU_ID”.
- In the STUDENT relation, a student with STU_ID = 21 is enrolled in two courses, “Computer” and “Math” and having two hobbies, “Dancing” and “Singing”.
- So, there are MVDs of “STU_ID” on “COURSE” and “HOBBY” attributes:

STU_ID →→ COURSE
STU_ID →→ HOBBY

Example of Multivalued Dependency and Fourth Normal Form

- The student with STU_ID = **21** has opted for two courses, **Computer** and **Math**, and has two hobbies, **Dancing** and **Singing**.
- **What problem this might lead to?**
  - The two records for student with STU_ID = 21 , will give rise to two more records, as shown below, because for this student, two hobbies exists. Hence, along with both the courses, these hobbies of the student should be specified in the relation.

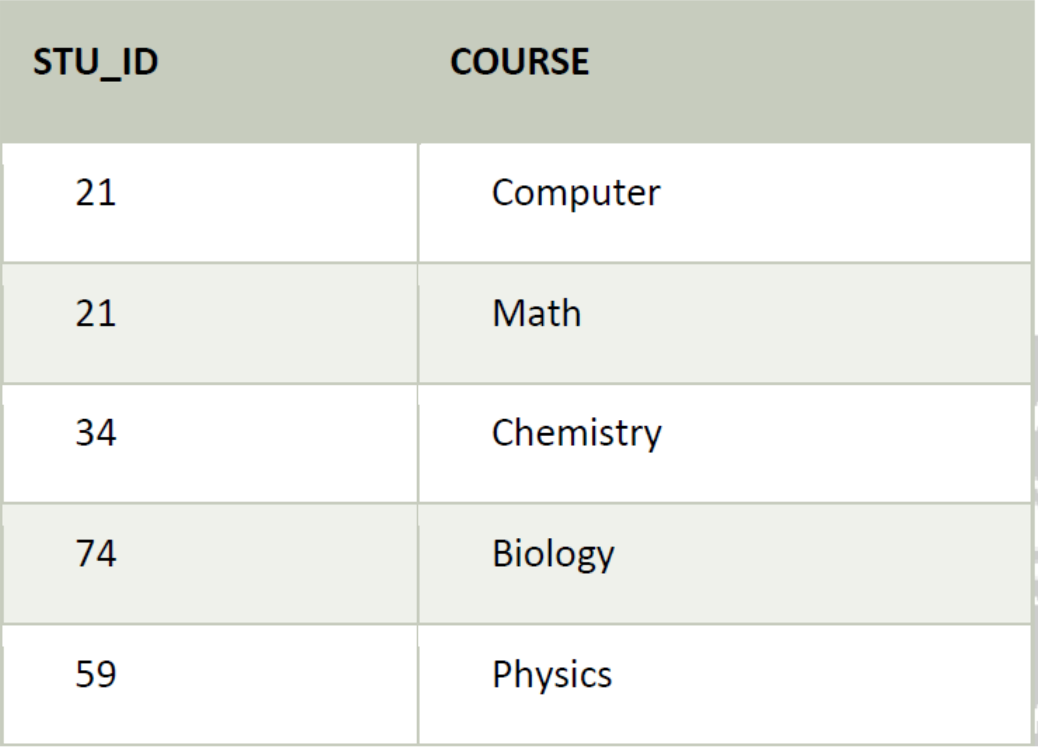
STU_ID	COURSE	HOBBY
21	Computer	Dancing
21	Computer	Singing
21	Math	Singing
21	Math	Dancing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

- The MVDs leads to unnecessary repetition or redundancy of data which may cause data inconsistency.

Example of Multivalued Dependency and Fourth Normal Form

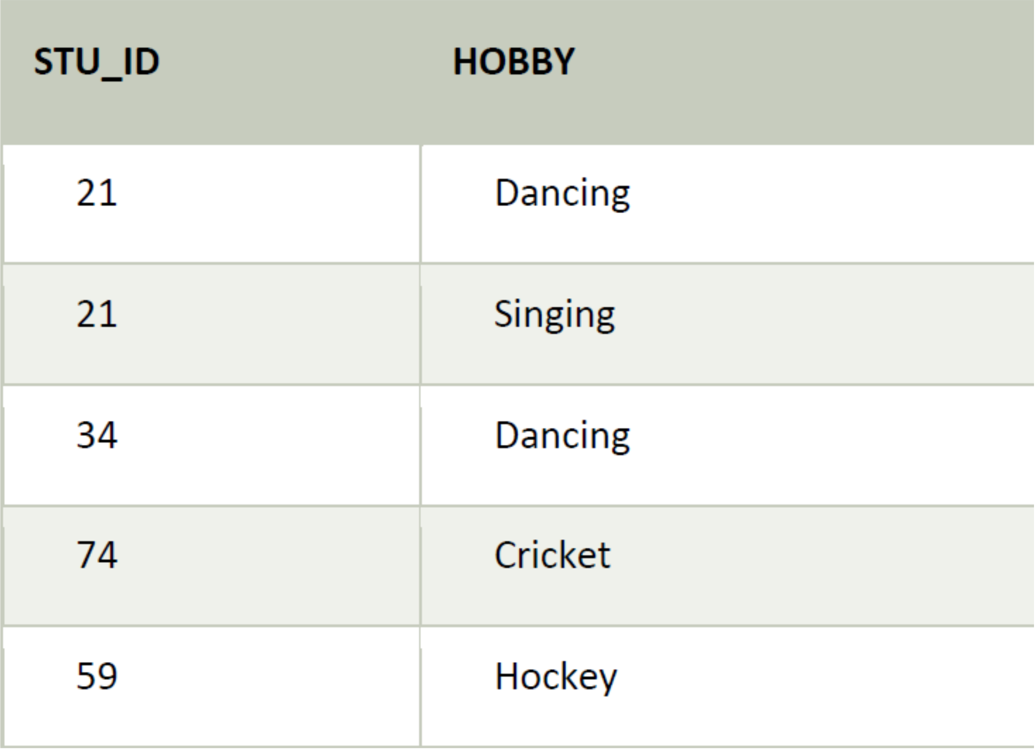
- To make the “STUDENT” relation satisfy the 4th normal form, we can decompose it into the following two tables:

STUDENT_COURSE



STU_ID	COURSE
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

STUDENT_HOBBY



STU_ID	HOBBY
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

Now the above relations satisfies the Fourth Normal Form.

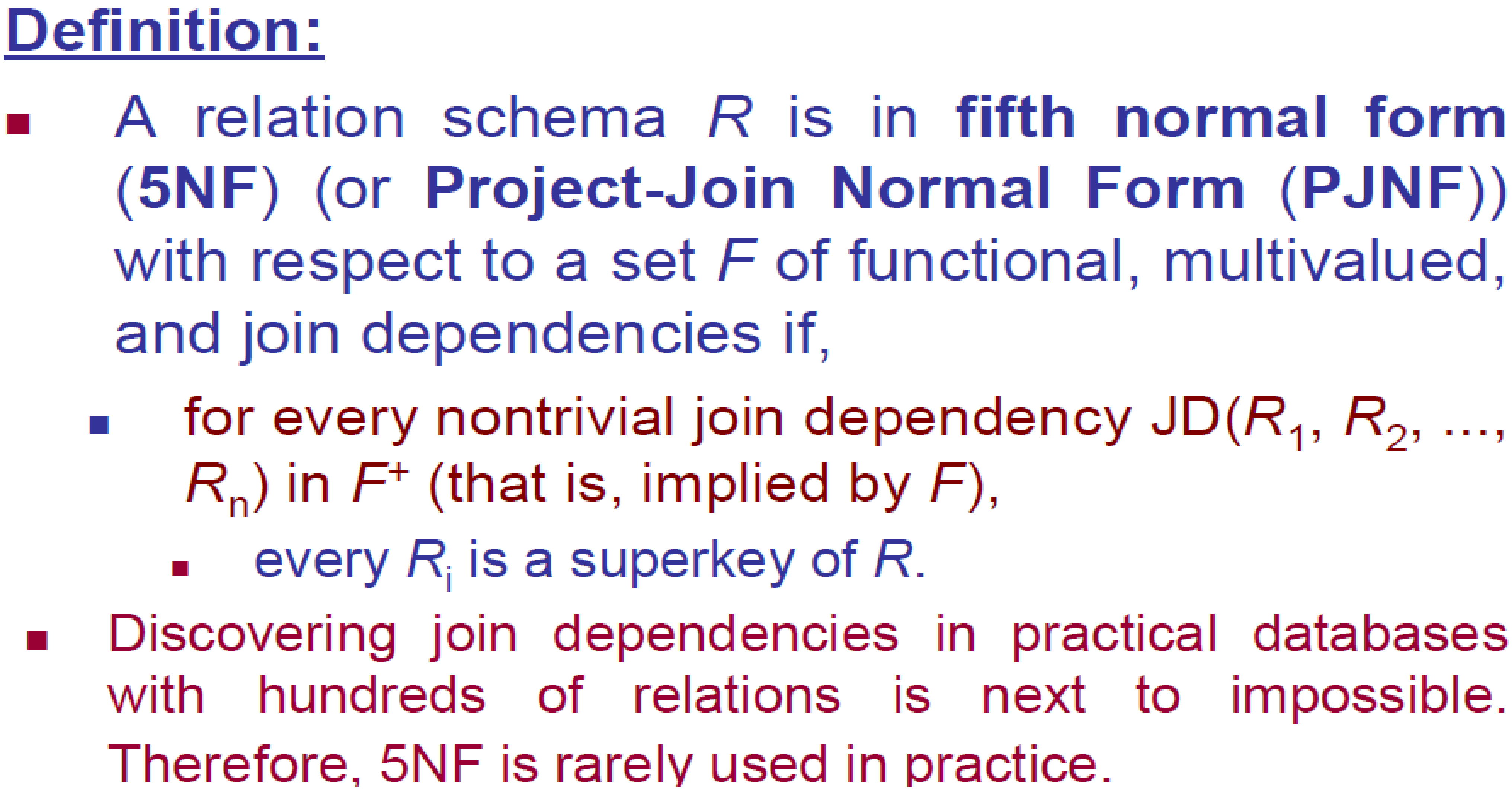
Join Dependencies and Fifth Normal Form

Definition:

- A **join dependency** (**JD**), denoted by JD(R_1, R_2, ..., R_n), specified on relation schema R, specifies a constraint on the states r of R.
  - The constraint states that every legal state r of R should have a non-additive join decomposition into R_1, R_2, ..., R_n; that is, for every such r we have
  - $* (\pi_{R1}(r), \pi_{R2}(r), \ldots, \pi_{Rn}(r)) = r$

  ***Note**: an MVD is a special case of a JD where n = 2.*
- A join dependency JD(R_1, R_2, ..., R_n), specified on relation schema R, is a **trivial JD** if one of the relation schemas R_i in JD(R_1, R_2, ..., R_n) is equal to R.

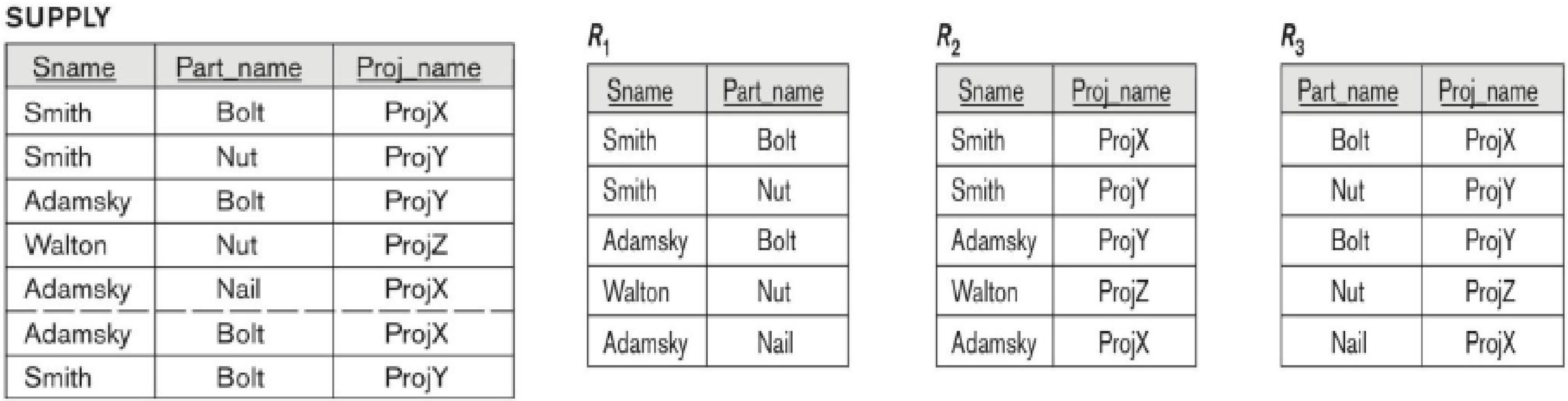
Join Dependencies and Fifth Normal Form



Definition:

- A relation schema *R* is in **fifth normal form (5NF)** (or **Project-Join Normal Form (PJNF)**) with respect to a set *F* of functional, multivalued, and join dependencies if,
  - for every nontrivial join dependency JD(R_1, R_2, ..., R_n) in F^+ (that is, implied by *F*),
    - every R_i is a superkey of *R*.
- Discovering join dependencies in practical databases with hundreds of relations is next to impossible. Therefore, 5NF is rarely used in practice.

Join Dependencies and Fifth Normal Form



SUPPLY

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

R_1

Sname	Part_name
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

Sname	Proj_name
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

Part_name	Proj_name
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

(c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R1, R2, R3). (d) Decomposing the relation SUPPLY into the 5NF relations R1, R2, R3.