UNIT – 3

DATABASE DESIGN

INFORMAL DESIGN GUIDELINES FOR RELATIONAL SCHEMA

- Semantics of attributes.
- Reducing the redundant values in tuples.
- Reducing the null values in tuples.
- Disallowing the possibility of generating spurious tuples.

Semantics of attributes

- Semantics – specifies how to interpret the attribute values stored in a tuple of the relation.
- Name of the attribute must have some meaning.
- Relationship among the relations must be clear.

Employee	<u>Person_id</u>	Person_name	Street	City
Company	<u>Company_id</u>	Company_name	C_City	
Works	<u>Person_id</u>	Company_id	Salary	
Manages	<u>Manager_id</u>	Manager_name	Person_id	

GUIDELINES:

- Design a relation schema so that it is easy to explain its meaning.
- Do not combine attributes from multiple entity types into a single relation.

Redundant Information in Tuples and Update Anomalies

- Mixing attributes of multiple entities may cause problems
- Information is stored redundantly wasting storage
- Problems with update anomalies
  - Insertion anomalies
  - Deletion anomalies
  - Modification anomalies

Person_id	Person_name	Street	City	Company_id	Company_name	C_City

Emp_com

Combining the 2 relations 'Employee' and 'Company' into one relation namely: 'Emp_com'

Insertion Anomalies

Insertion anomalies can be differentiated into two types:

- To insert a new tuple into emp_com, either include the attribute values for the company that the employee works for, or null if the employee does not work for a company yet.
- To insert a new company that has no employees as yet in the emp_com relation. Place the null values in the attributes for employee.

Deletion Anomalies

- If we delete from emp_com, an employee tuple that represent the last employee working for a particular company, the information concerning that company, is lost from the database.

Modification Anomalies

In Emp_com if the value of the attributes for a particular company is to be changed, the update is required for all tuples of 'emp_com' relation who work in that company, otherwise database will become inconsistent.

GUIDELINE : Design a schema that does not suffer from the insertion, deletion and update anomalies. If there are any present, then ensure that applications that update the database will operate correctly.

Null Values in Tuples

If many of the attributes do not apply to all tuples in the relation , we end up with many nulls in those tuples. Nulls can have multiple interpretations such as,

- Attribute not applicable or invalid
- Attribute value unknown (may exist)
- Value known to exist, but unavailable

GUIDELINE : Relations should be designed such that their tuples will have as few NULL values as possible

Generation of SpuriousTuples

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations

GUIDELINE : The relations should be designed to satisfy the lossless join condition. No spurious tuples should be generated by doing a natural-join of any relations.

FUNCTIONAL DEPENDENCIES

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations
- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes
- A functional dependency is a constraint between two sets of attributes from the database.
- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y

Functional Dependencies (contd…)

- X -> Y holds if whenever two tuples have the same value for X, they *must have* the same value for Y
- For any two tuples t1 and t2 in any relation instance r(R): *If* t1[X]=t2[X], *then* t1[Y]=t2[Y]
- X -> Y in R specifies a *constraint* on all relation instances r(R)
- Written as X -> Y; can be displayed graphically on a relation schema.
- FDs are derived from the real-world constraints on the attributes.

	X	Y	Z
t1	**Roll_No**	**N ame**	**Phone**
	1	XYZ	22235
	----	----	----
	----	----	----
t2	1	XYZ	22236

Therefore, X -> Y, i.e., Roll_No -> Name

But, X +> Y, Roll_No +> Phone

Inference Rules for FDs

- Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold

Armstrong's inference rules:

IR1. (**Reflexive**) If Y *subset-of* X, then X -> Y

IR2. (**Augmentation**) If X -> Y, then XZ -> YZ

(Notation: XZ stands for X U Z)

IR3. (**Transitive**) If X -> Y and Y -> Z, then X -> Z

- IR1, IR2, IR3 form a *sound* and *complete* set of inference rules

Inference Rules for FDs (contd…)

Some **additional inference rules** that are useful:

(**Decomposition**) If X -> YZ, then X -> Y and X -> Z

(**Union**) If X -> Y and X -> Z, then X -> YZ

(**Psuedotransitivity**) If X -> Y and WY -> Z, then WX -> Z

- The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3 (completeness property)

Proof for Inference Rules:

Rule 1:

{rollno, name} -> rollno

{rollno, name} -> name

	Rollno	Name	Age
t1	1	XYZ	21
t2	1	XYZ	21

t1[rollno, name] = t2[rollno, name]

{1,XYZ} = {1,XYZ}

1 = 1

XYZ = XYZ

Proof for Inference Rules:

Rule 2:

{rollno,} -> name

{rollno, age} -> {name, age}

1 -> XYZ

{1,21} -> {XYZ, 21}

t1[rollno] = t2[rollno]

t1[name] = t2[name]

t1[rollno,age] = t2[rollno, age]

Proof for Inference Rules:

Rule 3:

If X -> Y then,

t1[X] = t2[X]

⇒ t1[Y] = t2[Y]

If Y -> Z then,

t1[Y] = t2 [Y]

⇒ t1[Z] = t2[Z]

Hence,

X -> Z

Bhavana Vishwakarma

TRIVIAL AND NON-TRIVIAL FD

If X ->Y and X is a superset of Y, then this is called a Trivial Functional Dependency.

All FDs which are not trivial is called Non-trivial FD.

PRIME & NON-PRIME ATTRINUTES

If an attribute is a candidate key or a subset of candidate key then it is called a <u>prime attribute</u>.

For ex,

<u>**A**</u>	**B**	<u>**C**</u>	<u>**D**</u>

'A' is a candidate key.

Combination of C & D is a candidate key.

Hence; A, C, D are prime attributes and B is a non-prime attribute.

ATTRIBUTE CLOSURE

Def:

i. If a set of attributes 'X' appears on the left hand side of some FD in 'F', then the set X^+ of the attributes are those that are functionally dependent on X or any attribute of X. X^+ is called the closure of X under F where, F is the set of all FDs.

ii. For a given set of attributes 'X', the attribute closure 'X^+' with respect to FDs in 'F' is the set of attributes 'A' such that the FD X-> A can be inferred from F.

Ex:

a. If A -> B, then $A^+ = \{A,B\}$

b. If A -> BC, then $A^+ = \{A,B,C\}$

c. If A -> BC and C-> F, then $A^+ = \{A, B,C,F\}$

$AB^+ = \{A,B,BC,BF\}$

ALGORITHM FOR FINDING THE ATTRIBUTE CLOSURE

1. $X^+ := X$;
2. Repeat
   a. Old $X^+ := X^+$;
   b. For each FD Y -> Z in F do
   c. If Y is the subset of X, then $X^+ := X^+ \cup Z$;
3. Until ($X^+ :=$ old X^+);

Bhavana Vishwakarma

UN-NORMALIZED RELATION

A table is said to be un-normalized if each row contains multiple set of values for some of the columns, these multiple values in a single row are also called non-atomic values.

NORMALIZATION

It is a series of test performed on different relation schema so that the relation schema has all the properties of a good relation.

The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

First, Second and Third normal forms are based on functional dependencies whereas Fourth and Fifth normal forms are based on the concepts of multi-valued dependencies and join dependencies.

FIRST NORMAL FORM

Def:

A relation schema is said to be in first normal form (1NF) if the values in the domain of each attribute of the relation are atomic.

In other words, only one value is associated with each attribute and the value is not a set of values or a list of values.

Bhavana Vishwakarma

Ex: Course Preference Table

		Course Preferences	
Fac_Dept	Professor	Course	Course_Dept
Comp Sci	Smith	353	Comp Sci
		379	Comp Sci
		221	Decision Sci
	Clark	353	Comp Sci
		351	Comp Sci
		379	Comp Sci
		456	Mathematics
Chemistry	Turner	353	Comp Sci
		456	Mathematics
		272	Chemistry
Mathematics	Jamieson	353	Comp Sci
		379	Comp Sci
		221	Decision Sci
		456	Mathematics
		469	Mathematics

RELATION CRS_PREF IN 1NF

Professor	Course	Fac_Dept	Course_Dept
Smith	353	Comp Sci	Comp Sci
Smith	379	Comp Sci	Comp Sci
Smith	221	Comp Sci	Decision Sci
Clark	353	Comp Sci	Comp Sci
Clark	351	Comp Sci	Comp Sci
Clark	379	Comp Sci	Comp Sci
Clark	456	Comp Sci	Mathematics
Turner	353	Chemistry	Comp Sci
Turner	456	Chemistry	Mathematics
Turner	272	Chemistry	Chemistry
Jamieson	353	Mathematics	Comp Sci
Jamieson	379	Mathematics	Comp Sci
Jamieson	221	Mathematics	Decision Sci
Jamieson	456	Mathematics	Mathematics
Jamieson	469	Mathematics	Mathematics

RELATION CRS_PREF IN 1NF

Prof	Course1	Course2	Course3	Course4	Course5	Fac_Dept	Crs_Dept

RELATION CRS_PRF and COURSE IN 1NF

Prof	Course	Fac_Dept
Smith	353	Comp Sci
Smith	379	Comp Sci
Smith	221	Comp Sci
Clark	353	Comp Sci
Clark	351	Comp Sci
Clark	379	Comp Sci
Clark	456	Comp Sci
Turner	353	Chemistry
Turner	456	Chemistry
Turner	272	Chemistry
Jamieson	353	Mathematics
Jamieson	379	Mathematics
Jamieson	221	Mathematics
Jamieson	456	Mathematics
Jamieson	469	Mathematics

Course	Crs_Dept
353	Comp Sci
379	Comp Sci
221	Decision Sci
351	Comp Sci
456	Mathematics
272	Chemistry
469	Mathematics

Ex of 1NF

Member List

1	John Smith	Access, DB2, FoxPro
2	Dave Jones	dBASE, Clipper
3	Mike Beach	
4	Jerry Miller	DB2, Oracle
5	Ben Stuart	Oracle, Sybase
6	Fred Flint	Informix
7	Joe Blow	
8	Greg Brown	Access, MSSql Server
9	Doug Hope	

Member Table

MID	Name
1	John Smith
2	Dave Jones
3	Mike Beach
4	Jerry Miller
5	Ben Stuart
6	Fred Flint
7	Joe Blow
8	Greg Brown
9	Doug Hope

Database Table

DID	MID	Database
1	1	Access
2	1	DB2
3	1	FoxPro
4	2	dBASE
5	2	Clipper
6	4	DB2
7	4	Oracle
8	5	Oracle
9	5	Sybase
10	6	Informix
11	8	Access
12	8	MSSql Server

SECOND NORMAL FORM

Def:

A relation schema R is in second normal form (2NF) if it is in the 1NF and if all non-prime attributes are fully functionally dependent on the relation keys.

Full Functional Dependency

A given relation schema R and an FD $X \longrightarrow Y$, Y is fully functionally dependent on X if there is no Z, where Z is a proper subset of X such that $Z \longrightarrow Y$.

Ex:

if {Roll,Name} -> age then

For full FD

Roll $\not\longrightarrow$ age

Name $\not\longrightarrow$ age

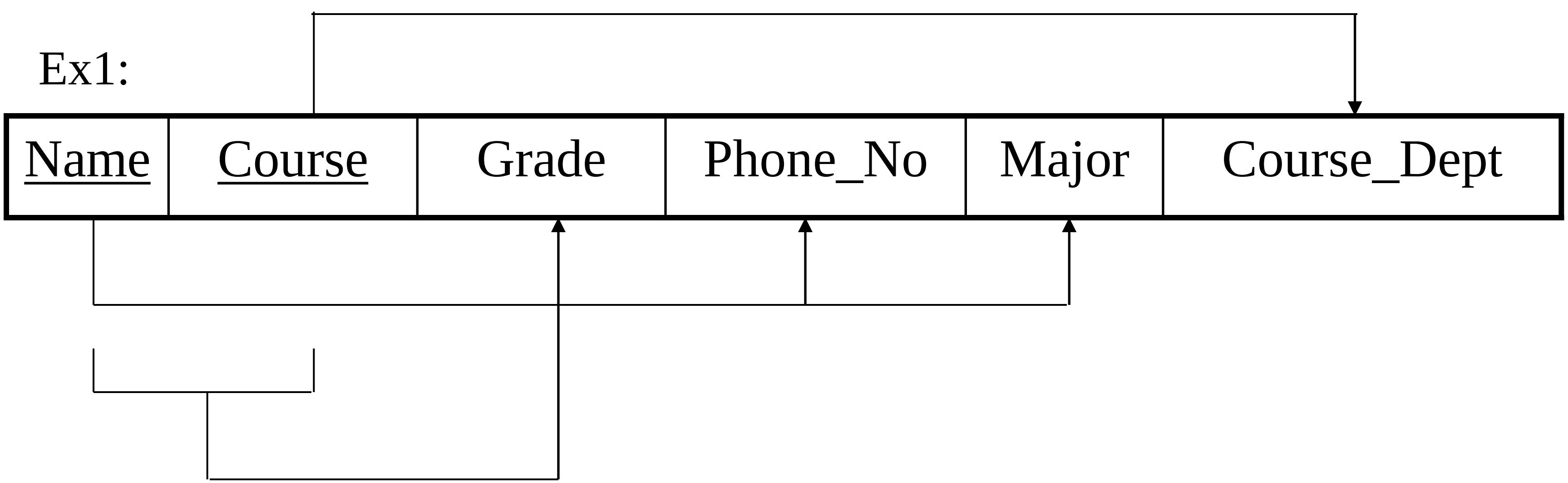
Ex:

In the relation schema R(ABCDEH) with FD's F = {A -> BC, CD -> E, E -> C, CD -> AH, ABH - > BD, DH -> BC }

Partial Dependency

Given a relation schema R with the functional dependencies F defined on the attributes of R and K is a candidate key, if X is a proper subset of K and if X -> A, then A is said to be partially dependent on K.

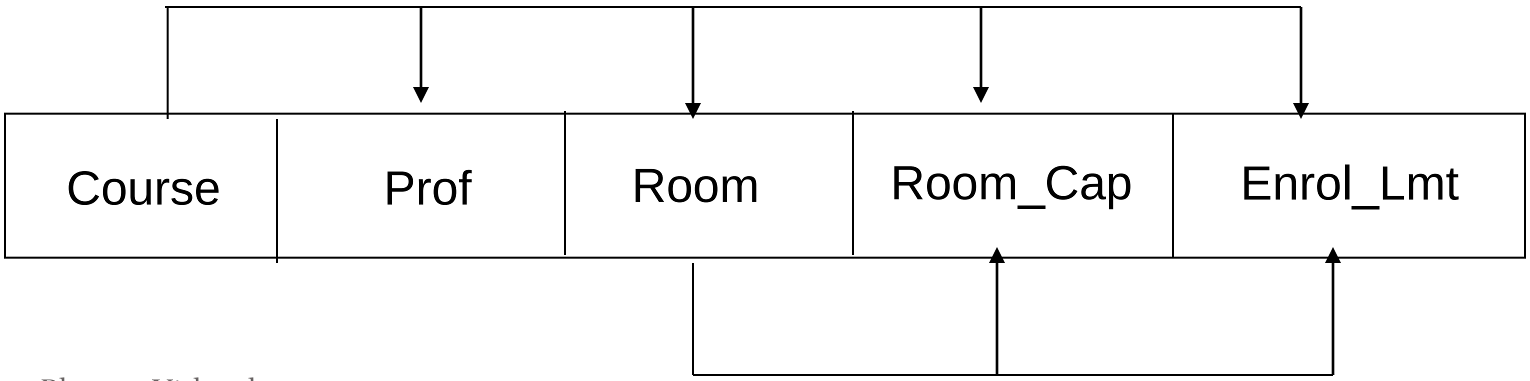
Ex1:



- In the relation schema STUDENT_COURSE_INFO(Name, Course, Grade, Phone_No, Major, Course_Dept) with the FDs F = {Name->Phone_NoMajor, Course->Course_Dept, NameCourse->Grade}; NameCourse is a candidate key, Name and Course are prime attributes. Grade is fully functionally dependent on the candidate key. Phone_No, Course_dept and Major are partially dependent on the candidate key.
- Ex2:

Given relation R(ABCD) and F = {AB->C, B->D}

EXAMPLE FOR 2NF – The TEACHES Relation



TEACHES Relation

Course	Prof	Room	Room_Cap	Enrol_Lmt
353	Smith	A532	45	40
351	Smith	C320	100	60
355	Clark	H940	400	300
456	Turner	B278	50	45
459	Jamieson	D110	50	45

FDs in this relation are:

Course->(Prof,Room,Room_Cap,Enrol_Lmt)

Room->Room_Cap

Room->Enrol_Lmt

(a) COURSE_DETAILS (b) ROOM_DETAILS

Course	Prof	Room	Enrol_ Lmt
353	Smith	A532	40
351	Smith	C320	60
355	Clark	H940	300
456	Turner	B278	45
459	Jamieson	D110	45

(a)

Room	Room_Cap
A532	45
C320	100
H940	400
B278	50
D110	50

(b)

Ex of 2NF

Database Table

DID	MID	Database
1	1	Access
2	1	DB2
3	1	FoxPro
4	2	dBASE
5	2	Clipper
6	4	DB2
7	4	Oracle
8	5	Oracle
9	5	Sybase
10	6	Informix
11	8	Access
12	8	MSSql Server

Member Table

MID	Name
1	John Smith
2	Dave Jones
3	Mike Beach
4	Jerry Miller
5	Ben Stuart
6	Fred Flint
7	Joe Blow
8	Greg Brown
9	Doug Hope

MbrDB Table

MID	DID
1	1
1	2
1	3
2	4
2	5
4	2
4	6
5	6
5	7
6	8
8	1
8	9

Database Table

DID	Database
1	Access
2	DB2
3	FoxPro
4	dBASE
5	Clipper
6	Oracle
7	Sybase
8	Informix
9	MSSql Server

THIRD NORMAL FORM

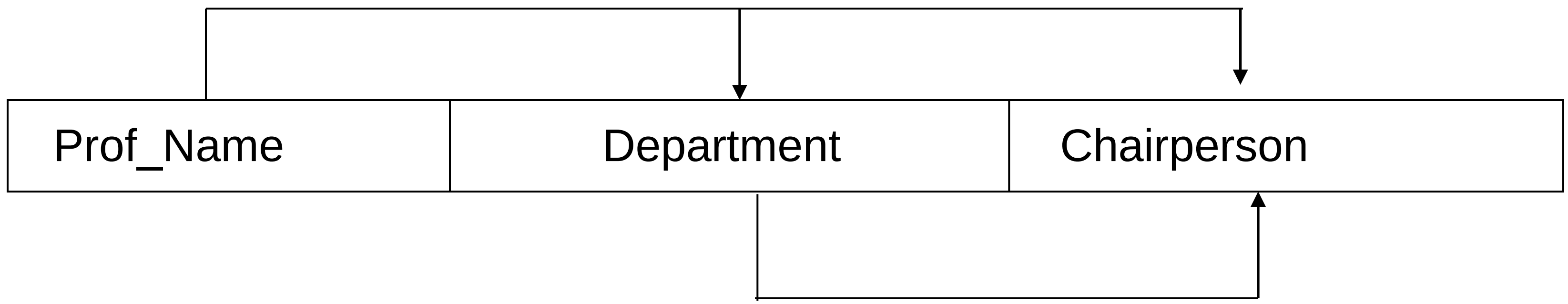
Def:

A relation schema R is in 3NF if no non-prime attribute of R is transitively dependent on the primary key of R.

Transitive Dependency

Given a relation R with the functional dependencies F defined on the attributes of R, let X and Y be the subsets of functional dependencies {X->Y, Y->A} is implied by F, then A is transitively dependent on X.

Ex:



In the relation schema PROF_INFO the FDs are;

Prof_Name -> Department,

Department -> Chairperson

Prof_Name is a key and Chairperson is transitively dependent on the key.

2.Given R(ABCDE) and the function dependencies

F = {AB->C, B->D, C->E}

Decomposition of COURSE_DETAILS to eliminate transitive dependency

Course	Prof	Enrol_Lmt
353	Smith	40
351	Smith	60
355	Clark	300
456	Turner	45
459	Jamieson	45

Course	Room
353	A532
351	C320
355	H940
456	B278
459	D110

Member Table

MID	Name	Company	CompLoc
1	John Smith	ABC	Alabama
2	Dave Jones	MCI	Florida
3	Mike Beach	IBM	Delaware
4	Jerry Miller	MCI	Florida
5	Ben Stuart	AIC	Nebraska
6	Fred Flint	ABC	Alabama
7	Joe Blow	RU Nuts	Iowa
8	Greg Brown	XYZ	New York
9	Doug Hope	IBM	Delaware

Member Table

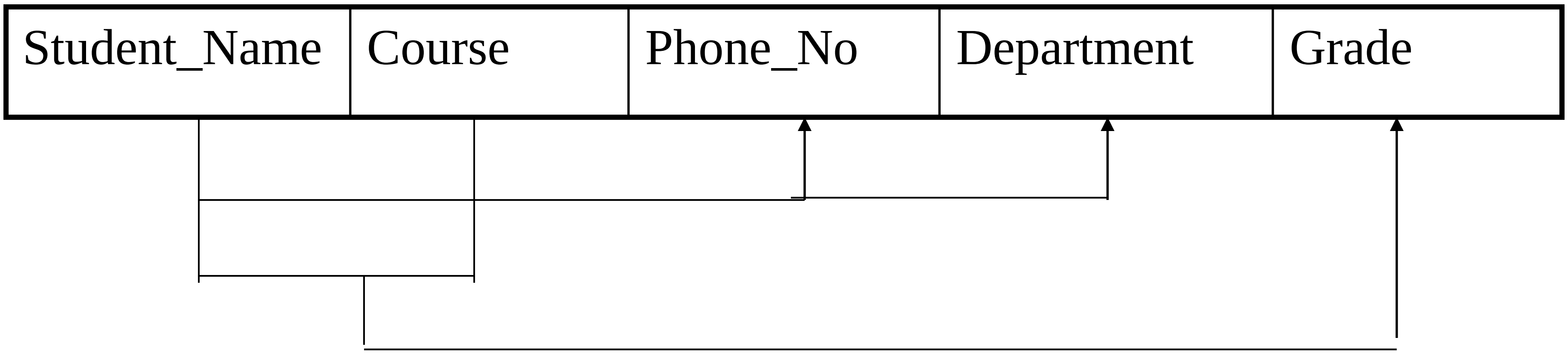
MID	Name	CID
1	John Smith	1
2	Dave Jones	2
3	Mike Beach	3
4	Jerry Miller	2
5	Ben Stuart	4
6	Fred Flint	1
7	Joe Blow	5
8	Greg Brown	6
9	Doug Hope	3

Company Table

CID	Name	Location
1	ABC	Alabama
2	MCI	Florida
3	IBM	Delaware
4	AIC	Nebraska
5	RU Nuts	Iowa
6	XYZ	New York

EXAMPLE- The ENROLLMENT relation

Key attribute – Student_Name, Course



DECOMPOSTION

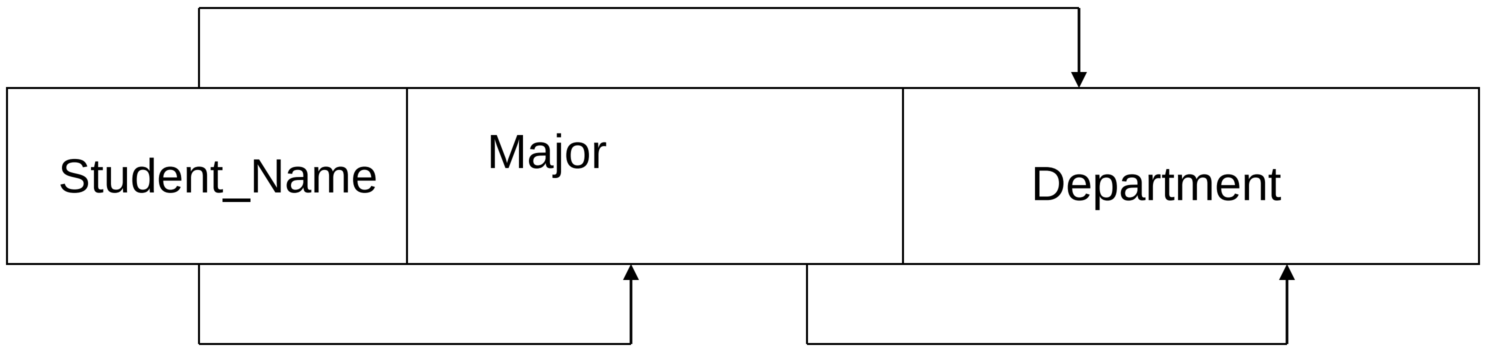
STUDENT

1	3	4

ENROL

1	2	5

MAJOR



STUDENT_MAJOR

Student_Name	Major

MAJOR_DEPARTMENT

Major	Department

Qu1.

A relation schema R = {A, B, C, D, E, F, G, H, I, J} & the set of FDs F = {AB->C, A->DE, B -> F, F->GH, D->IJ} holds on R. What is the key attribute of R. Covert the relation schema R into 2NF & then into 3NF.

Qu2.

Given relation R = {A,B,C,D,E,F,G,H,I,J} & set of FDs

F = {A->B,D->IJ,B->E,BC->I}. Also it is given that AB is a key attribute and D is also a key attribute. Convert the relation into 3NF.

Sol1.

$A^+ = \{A, D, E, I, J\}$

$B^+ = \{B, C, G, H\}$

$F^+ = \{F, G, H\}$

$D^+ = \{ I, J, D\}$

$AB^+ = \{A, B, C, D, E, F, G, H, I, J\}$

<u>A</u>	<u>B</u>	C	D	E	F	G	H	I	J

For the relation to be in 2NF C,D,E,F,G,H,I,J must be fully functionally dependent on AB. But D, E, F are partially dependent on AB.

A	B	C	G	H	I	J

Multivalued Dependencies

Functional dependencies rule out certain tuples from appearing in a relation. If *A* -> *B*, then we cannot have two tuples with the same *A* value but different *B* values.

Multivalued dependencies do not rule out the existence of certain tuples.

Instead, they **require** that other tuples of a certain form be present in the relation.

Let *R* be a relation schema, and let α is a subset of R and β is a subset of R .

The **multivalued dependency**

$\alpha \rightarrow\rightarrow \beta$

holds on *R* if in any legal relation *r*(*R*), for all pairs of tuples t_1 and t_2 in *r* such that $t_1[\alpha] = t_2[\beta]$, there exist tuples t_3 and t_4 in *r* such that:

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$

$$t_3[\beta] = t_1[\beta]$$

$$t_3[R- \beta] = t_2[R- \beta]$$

$$t_4[\beta] = t_2[\beta]$$

$$t_4[R- \beta] = t_1[R- \beta]$$

Tabular representation of $\alpha \twoheadrightarrow \beta$

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

name	address	car
Tom	North Rd.	Toyota
Tom	Oak St.	Honda
Tom	North Rd.	Honda
Tom	Oak St.	Toyota

(*name*, *address*, *car*) where $name \twoheadrightarrow address$ and $name \twoheadrightarrow car$

Course	Book	Lecturer
AHA	Silberschatz	John D
AHA	Nederpelt	John D
AHA	Silberschatz	William M
AHA	Nederpelt	William M
AHA	Silberschatz	Christian G
AHA	Nederpelt	Christian G
OSO	Silberschatz	John D
OSO	Silberschatz	William M

Teaching database

{course} ->> {book}

{course} ->> {lecturer}.

FOURTH NORMAL FORM

- **Fourth normal form** (**4NF**) is a normal form used in database normalization. 4NF ensures that independent multivalued facts are correctly and efficiently represented in a database design. 4NF is the next level of normalization after Boyce-Codd normal form (BCNF).
- The definition of 4NF relies on the notion of a multivalued dependency. A table with a multivalued dependency is one where the existence of two or more independent many-to-many relationships in a table causes redundancy; and it is this redundancy which is removed by fourth normal form.

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John
Brown	W	Jim
Brown	X	Jim
Brown	Y	Jim
Brown	Z	Jim
Brown	W	Joan
Brown	X	Joan
Brown	Y	Joan
Brown	Z	Joan
Brown	W	Bob
Brown	X	Bob
Brown	Y	Bob
Brown	Z	Bob

EMP

EMP_PROJECTS

ENAME	PNAME
Smith	X
Smith	Y
Brown	W
Brown	X
Brown	Y
Brown	Z

EMP_DEPENDENTS

ENAME	DNAME
Smith	Anna
Smith	John
Brown	Jim
Brown	Joan
Brown	Bob

Pizza Delivery Permutations

Restaurant	Pizza Variety	Delivery Area
Vincenzo's Pizza	Thick Crust	Springfield
Vincenzo's Pizza	Thick Crust	Shelbyville
Vincenzo's Pizza	Thin Crust	Springfield
Vincenzo's	ThinCrust	Shelbyville
Elite Pizza	Thin Crust	Capital City
Elite Pizza	Stuffed Crust	Capital City
A1 Pizza	Thick Crust	Springfield
A1 Pizza	Thick Crust	Shelbyville
A1 Pizza	Thick Crust	Capital City
A1 Pizza	Stuffed Crust	Springfield
A1 Pizza	Stuffed Crust	Shelbyville
A1 Pizza	Stuffed Crust	Capital City

To satisfy 4NF, we must place the facts about varieties offered into a different table from the facts about delivery areas:

Varieties By Restaurant

Restaurant	Pizza Variety
Vincenzo's Pizza	Thick Crust
Vincenzo's Pizza	Thin Crust
Elite Pizza	Thin Crust
Elite Pizza	Stuffed Crust
A1 Pizza	Thick Crust
A1 Pizza	Stuffed Crust

Delivery Areas By Restaurant

Restaurant	Delivery Area
Vincenzo's Pizza	Springfield
Vincenzo's Pizza	Shelbyville
Elite Pizza	Capital City
A1 Pizza	Springfield
A1 Pizza	Shelbyville
A1 Pizza	Capital City