UNIT - 3

The Relational Model and Relational Database Constraint

↓

Relation

↓

Mathematics

↓

Set Theory

$A = \{0, 1\}$

$B = \{a, b, c\}$

$\boxed{\text{Relation} \subseteq A \times B}$ → Relation is a subset of a cartesian product of two sets A and B.

$A \times B = \{(0,a)\ (0,b)\ (0,c)\ (1,a)\ (1,b)\ (1,c)\}$

B \ A	0	1
a	0,a	1,a
b	0,b	1,b
c	0,c	1,c

Relational Model Concepts

- Concepts based on the Relation from set theory (maths)
- Developed by Dr. E.F Codd in ~~IBM~~ IBM research in 1970's

The Relational Model represents the database as the collection of Relations.

DATE: __/__/____
PAGE ______

Relational schema

$\boxed{R(A_1, A_2, A_3, \ldots, A_n)}$

where R is the ~~relation~~ name of the relation and $A_1, A_2, A_3, \ldots, A_n$ are the attributes.

Eg: Student (Roll, name, age, marks)

(Arity) Degree of ~~a~~ Relation → no. of attributes in a relation

→ Degree (R) = n-degree
→ Degree (Student) = 4-degree

Informal ~~Relation~~ Way	Formal Way
1. Table	Relation
2. Row/Records	Tuples
3. Fieldname/column	Attribute
4. set of values of a column	Domain
5. Table definition	Relational Schema
6. Populated Table	Relational State

Scanned by CamScanner

Attributes

Student

	Roll No	Name	Age	Marks
t_1	1	X	18	95
t_2	2	Z	19	NULL
t_3	3	X	NULL	75
t_4	4	Y	20	80

Tuples

Domain of Roll No.

Relation = Student

Degree (Student) = 4

DOM (A_i)

where $i = 1$ to n

DOM (Roll No) = $\{1, 2, 3, 4\}$

DOM (Age) = $\{18, 19, \text{NULL}, 20\}$

* Values in a tuple should be in order
* Tuples in a relation may not be in order

Characterstics of a Relation

① → Ordering of tuples in a relation are not considered to be in order

② → Ordering of attributes in a relational schema $R(A_1, A_2, \ldots, A_n)$ should be in order with values $(V_1, V_2, \ldots\ldots V_n)$.

Otherwise, there is chances of data type mis-matching.

DATE: __/__/____
PAGE ______

(3) Values in a ~~tuple~~ tuple.

(3) - All values are considered to be atomic means single value, whether it can be NULL.
- No multivalued values are considered.
- No composite values are considered.

→ Relational Model is sometimes called Flat Relational Model.

Relational Model constraints and Relational Database Schema

constraints : rules or conditions applied to relation attribute, application, data model

Relational Model Constraints

- Implicit Constraint
  - → Applied to data model
- Explicit constraint
  - → Applied to relational Schema, DDL
- Application based constraint
  - → Also called semantic constraint / Business Rules
  - → Applied to particular database application

(1) Key constraints.

Keys

(A) Identifying Keys
(I) Primary Key
(II) Candidate Key
(III) Super ~~Key~~ Key

(B) Foreign Key

GOOD WRITE

Foreign Key → can contain a NULL value
Primary key → cannot contain a NULL value.

$t_1[P.K] \;!= t_2[P.K]$
$t_1[C.K] \;!= t_2[CK]$
$t_1[SK] \;!= t_2[SK]$

② Domain Constraint

↳ Domain is a set of values of a particular column
↳ always depends upon the datatypes of the attributes

Eg: DOM [cust_id] = $\underbrace{\{1, 2, 3, 4\}}_{\text{set of values}}$

character
numeric
date and time
} datatype

③ Relational Integrity Constraints

Entity ~~Constraint~~ Integrity

- Applied as Primary Key
- It states that no Primary Key can be NULL.
- It should be unique.
- Single relation / table

Referential Integrity

- Applied as Foreign Key
- It states that Foreign Key is a key in a relation which refers to some other relation in which it is a Primary Key
- Used for JOIN of two tables.

DATE: __/__/__
PAGE ______

Database Design

Mapping OR Constraints of ER to Relational Model

→ Algorithm / steps for converting ER Model to Relational Model

1. Mapping of Strong / Regular Entity Types.
2. Mapping of weak Entity Types

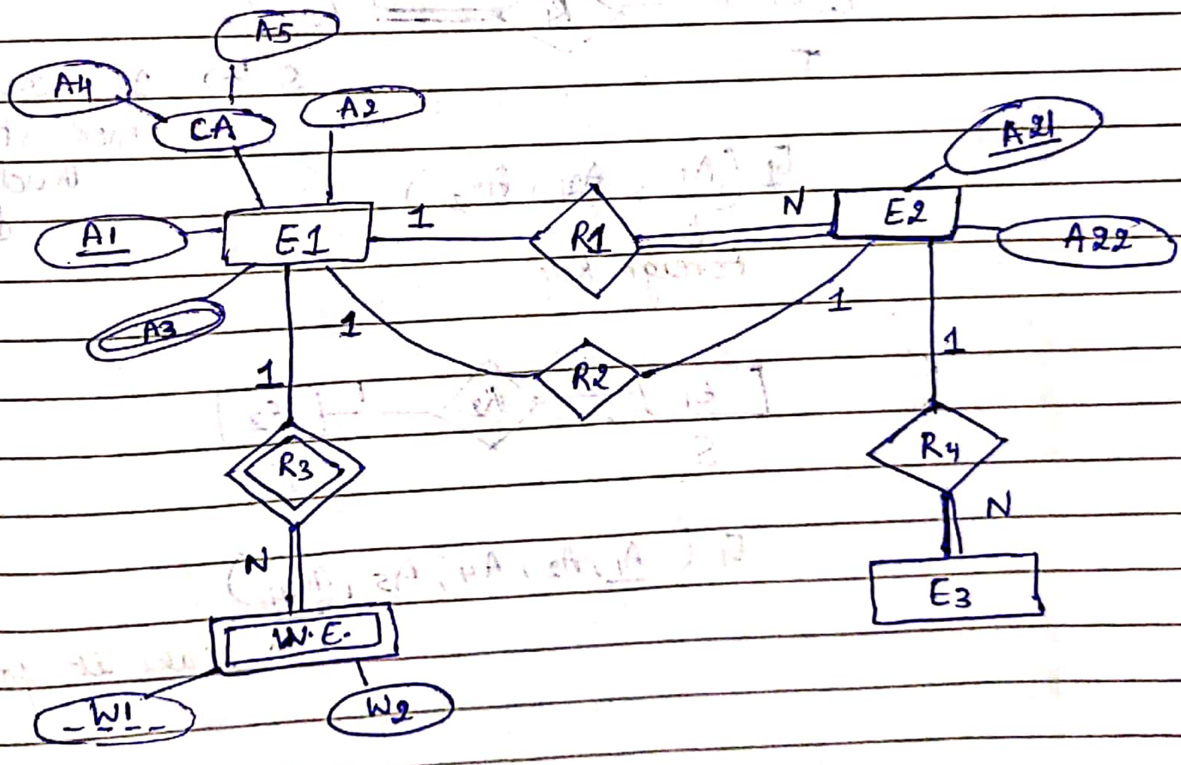
   } Related to entity

3. Mapping of Binary 1:1 Relationship
4. Mapping of Binary 1:N Relationship
5. Mapping of Binary M:N Relationship

   } Related to relationship type

6. Mapping of multivalued attribute
7. Mapping of N-ary relationship type where N >= 3.

Eg:



Step 1.

- Find out the strong entity in the given ER diagram
- Create a new relation R to each strong entity
- All the simple attributes became the field of the new relation R
- If there is multivalued, then goto Step 6.
- Make the key attribute to the primary key of R.

Conversion Steps.

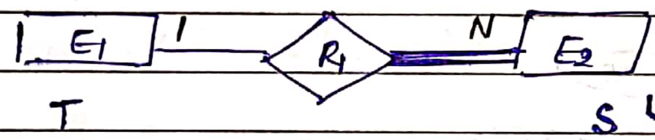
1: E_1 ($\underline{A_1}$, A_2, A_4, A_5)
 E_2 ($\underline{A_{21}}$, A_{22})

2: WE ($\underline{A_1}$, $\underline{W_1}$, W_2)

A_1 → Foreign Key
W_1 → ~~Existing Key~~ Partial Key

Primary ($A_1 + W_1$)

3: Identify Binary ~~=~~ Relationship Types

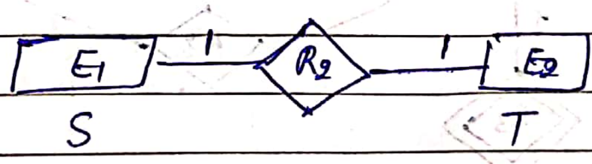


S ↳ It will be more stronger and will dominate

E_2 (A_1, $\underline{A_{21}}$, A_{22})

A_1 → Foreign Key

if

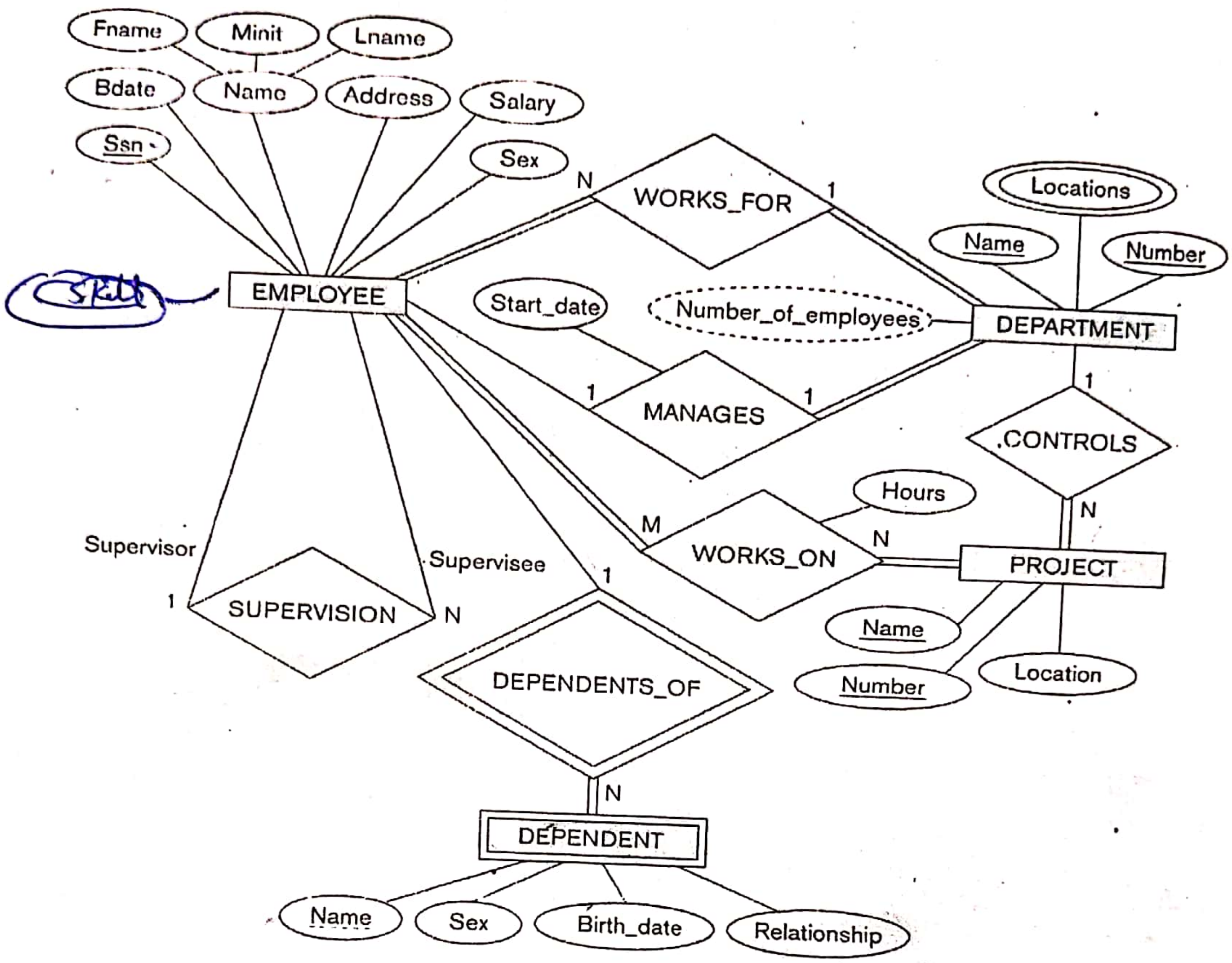


E_1 ($\underline{A_1}$, A_2, A_4, A_5, A_{21})

↳ Make it foreign key.

NOTE: if there is a composite attribute, then we will consider its simple fields only.

COMPANY ER-Diagram



Q. Convert the above COMPANY ER-Diagram into Relational Database Schema of COMPANY.

COMPANY ER-Diagram to Relational Database Schema

After step 1

Employee (Ssn , Bdate , Fname , Minit , Lname , Address , Salary , sex)
P.K = { Ssn }

Project (Name, Number , Location) P.K = { Name, Number }

Department (Name, Number) Location was omitted as it is a multi-valued attribute.

P.K = { Name, Number }

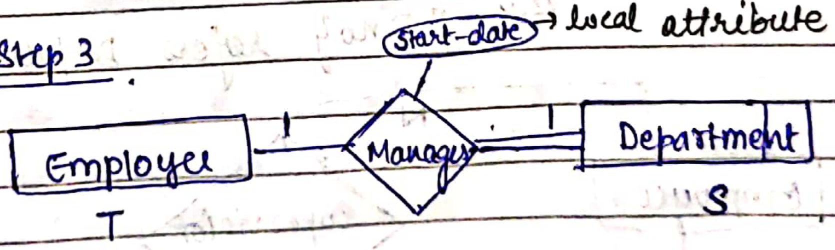
After Step 2

Dependent (Ssn, Name , Sex , Birthdate, Relationship)

P.K = { Ssn, Name }

F.K = { Ssn } refers employee

After Step 3.



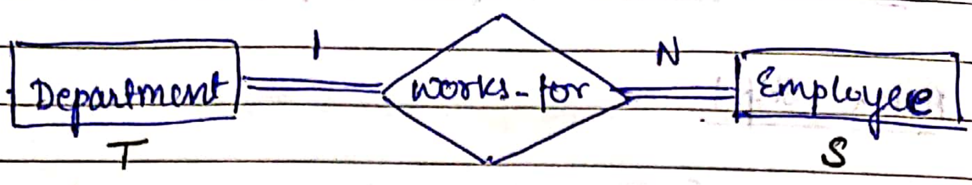
Department (Ssn , Name, Number, Start-date)

P.K = { Name, Number }
F.K = { Ssn } refers employee.

* The entity with total participation is made S.

After Step 4

a)



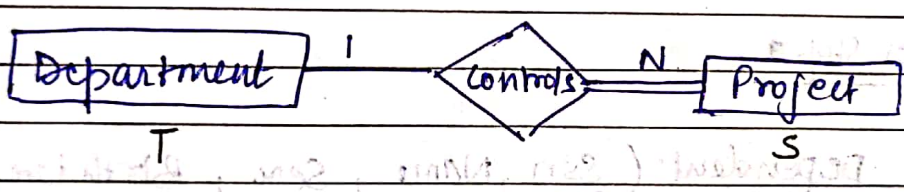
* The ~~attribute~~ ~~wit~~ entity having N will become S

Employee (<u>SSN</u> , Bdate , Fname , Minit , Lname , Address , Salary , Sex , Dno)

P.K = { SSN }

F.K = { Dno } refers Department

b)

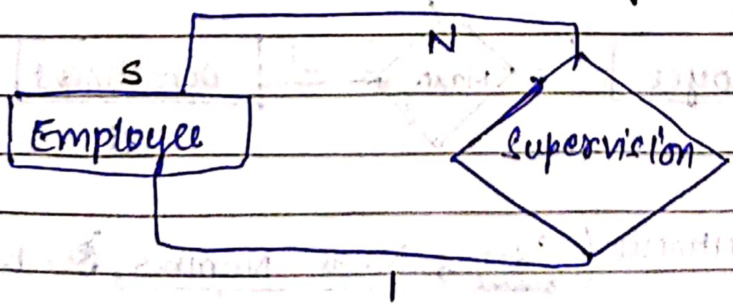


Project (<u>Name , Number</u> , Location , Dno)

P.K = { Name , Number }

F.K = { Dno } refers Department

c)



Employee (<u>SSN</u> , Bdate , Fname , Minit , Lname , Address , Salary , Sex , Dno , Super-SSN)

P.K = { SSN }

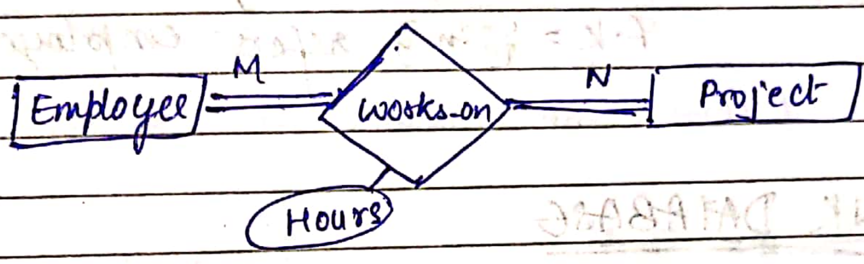
F.K = { Dno } refers Department

{ Super-SSN } refers employee

DATE: __/__/____
PAGE ______

Step 5 : Mapping of M:N Relationship

a) Identify the M:N Relationship from the diagram.
b) In this case, we need to create a new relation R with the same name as the relationship name
c) If, there is any local attribute in that relationship, then add it to the R.

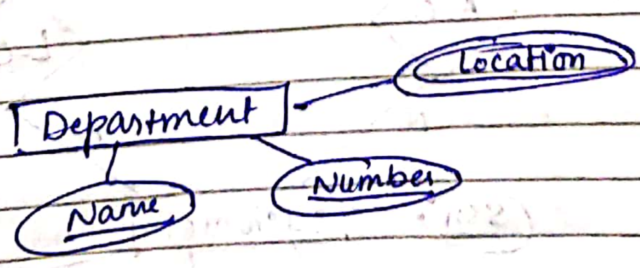


works_on (Ssn, Pno, Hours)

P.K = { Ssn, Pno }
F.K = { Ssn } refers employee
{ Pno } refers Project

Step 6 : Mapping of multivalued attribute.

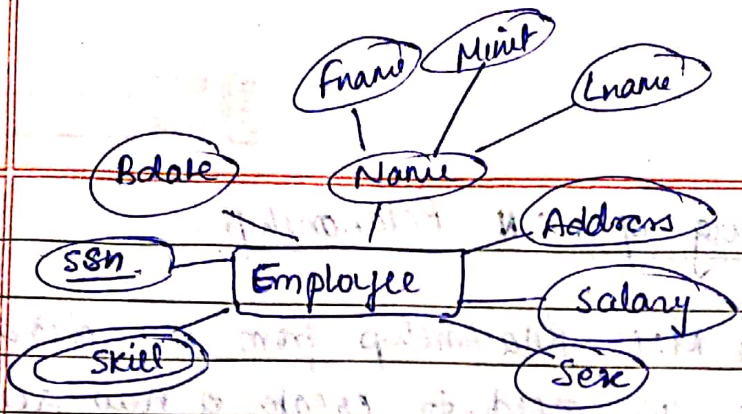
a) Identify the multivalued attributes
b) Create a new relation R that contains the associated attribute belong to the entity.



Dept_loc (Dno, Location)
P.K = { Dno, Location }
F.K = { Dno } refers Department

GOOD WRITE

Scanned by CamScanner

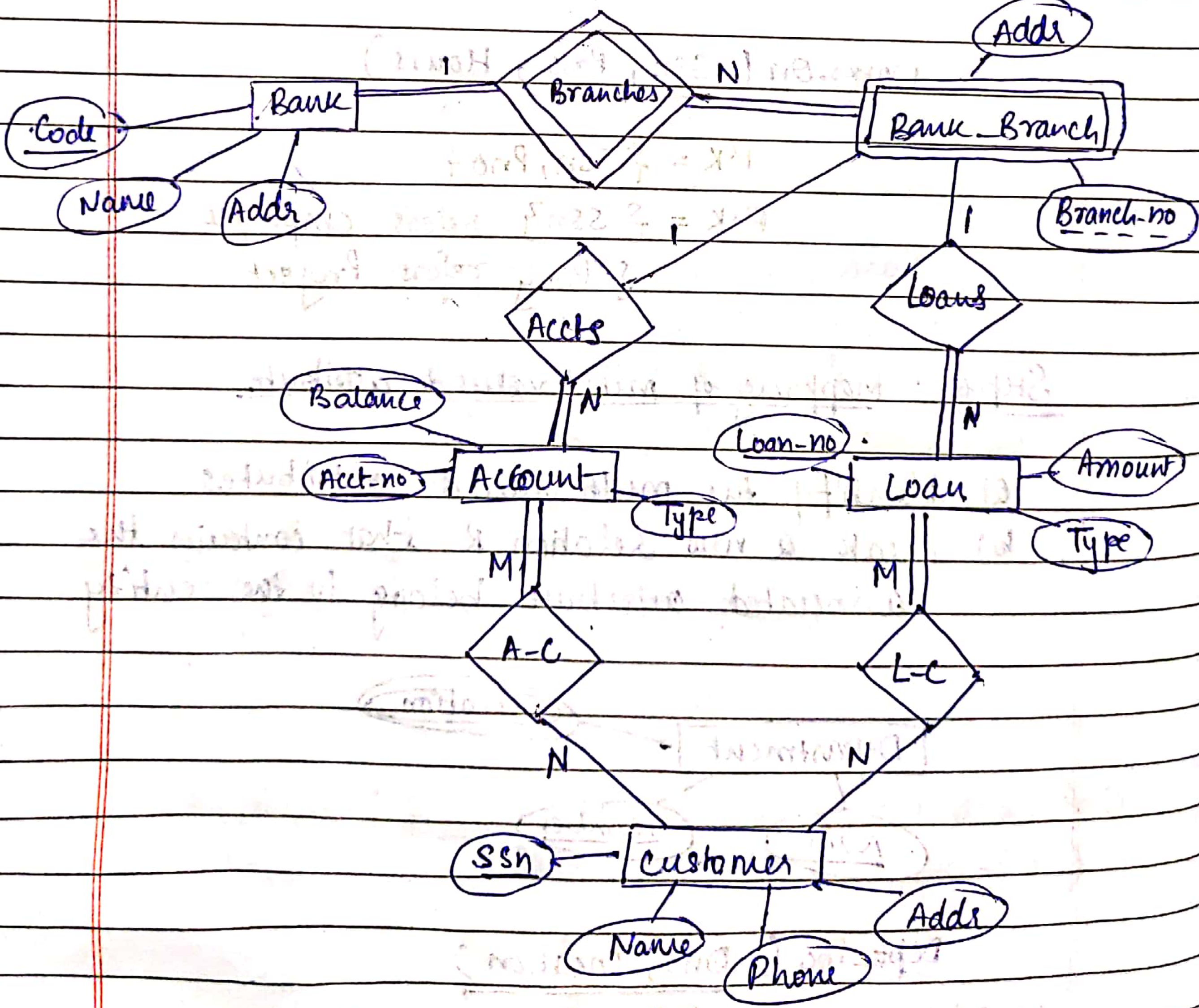


Emp_Skill (ssn, skill)

P.K. = { ssn, skill }

F.K = {ssn} refers employee

BANK DATABASE



After Step1.

Bank (Code, Name, Addr) P.K = {Code}

Account (Acct_no, Balance, Type) P.K = {Acct_no}

Loan (Loan_no, Amount, Type) P.K = {Loan_no}

Customer (SSn, Phone, Name, Addr) P.K = {SSn}

After Step 2.

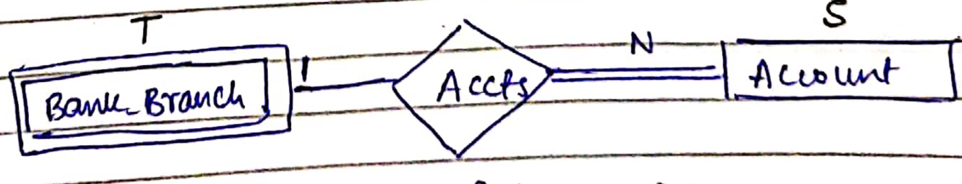
Bank_Branch (Code, Branch_no, Addr).

~~P.K = {Code, Branch_no}~~

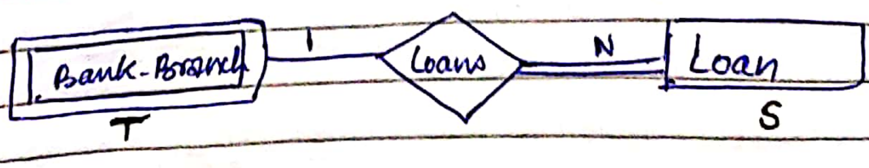
F.K = {Code} refers Bank

After Step3.
No 1:1 Binary Relationship

After Step 4

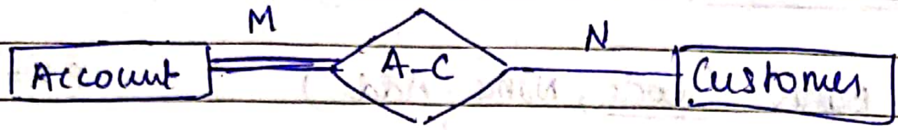


Account (Acc_no, Balance, Type



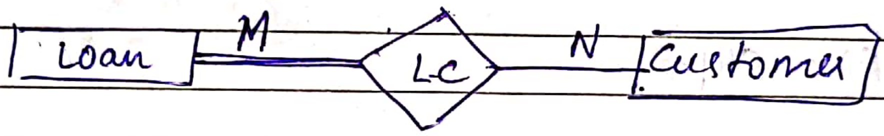
Loan (Loan_no, Amount, Type

After Step 5.



A-C (Ssn, Acct-no)

P.K = ~~{Ssn, Acct-no~~ { Ssn, Acct-no }
F.K = { Ssn } refers Customer
= { Acct-no } refers Account

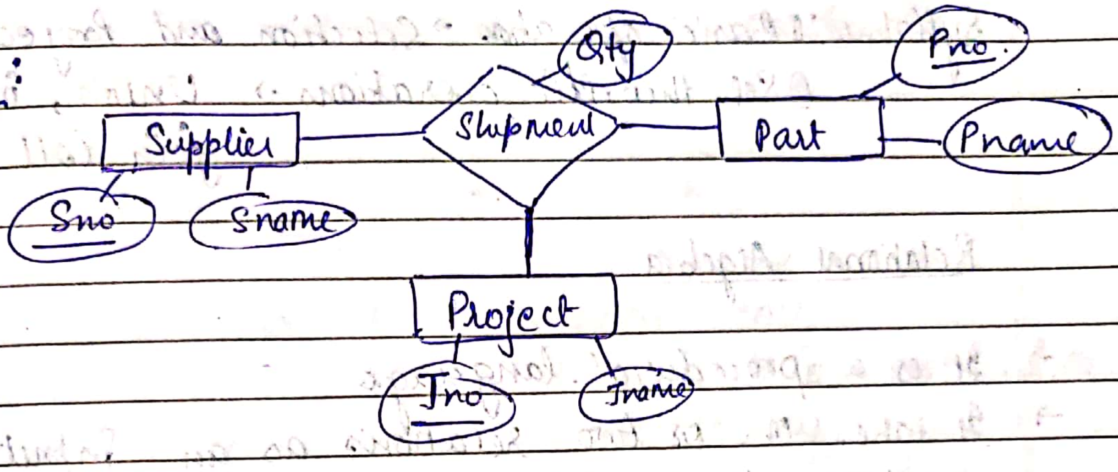


L-C (Ssn, Loan-no)

P.K = { Ssn, Loan-no }
F.K = { Ssn } refers Customer
= { Loan-no } refers Loan.

Q Consider the following N-ary ER diagram of SPJ database where N=3

step7 :



~~Ste~~ Supplier (Sno , Sname)
Part (Pno , Pname)
Project (Jno , Jname)
Shipment (Sno, Pno, Jno , Qty)

P.K = {Sno, Pno, Jno}

F.K = {Sno} refers Supplier
= {Pno} refers Part
= {Jno} refers Project

~~UNIT~~.

Relational Algebra (RA)

Syllabus: ① Basic operation → Selection and Projection
② Set theoretic operations → Union, intersection, join (all types of JOIN)

Relational Algebra

→ It is a procedural language
→ It takes one or two relations as an input and produces the result into a single relation.
→ It uses operations/operators to perform queries
→ Operators are : Unary and Binary

Unary → applied to single relation/table.

Binary → applied to two relations/tables.

Classification of Relational Algebra Operations

I. Unary Relational operation

Operation	operator
a) Selection	σ (sigma)
b) Projection	π (pi)

a) Selection operation / Select operator (σ)

↳ It is a kind of operation applied to a single relation/table R.

DATE: __/__/___
PAGE ______

↳ It's purpose is to select tuples from the relation based on some condition. This condition is called predicate.

Syntax

$$\boxed{Table \leftarrow \sigma_p (R)}$$

where R = Relation name
P = condition to be applied to R constitute diff type of operations along with the attribute

Q. Get all the details from table employee that belong to department No. 10.

Ans

$$T \leftarrow \sigma_{deptno = 10} (employee)$$

b) Projection Operation / π operator

↳ It is used to list the no. of attributes / columns from the relation R.

Syntax:

$$\boxed{Table \leftarrow \pi_{<attribute\_list>} (R)}$$

where R is the relation name and
<attribute_list> = column 1, column 2 ....., column n

DATE: __/__/__
PAGE ____

Q. 1. Select / get / retrieve name of all employees from the table employee.

Ans.

$$T \leftarrow \pi_{Name}(employee)$$

2. Select name of all employees where department no is 10 from table employee.

Ans. $T \leftarrow \pi_{Name}(\sigma_{deptno=10}(employee))$

OR

Step 1 : $T_1 \leftarrow \sigma_{deptno=10}(employee)$

Step 2 : : $Result \leftarrow \pi_{name}(T_1)$

II Relational Operators from Set Theory.

a) Union operation / Union Operator. ($\cup$)

↳ $\cup$ operator is binary because it is applied to two relations at a time.

Syntax :

$\boxed{Result \leftarrow R \cup S}$

where R and S are two input relations and Result is the single output relation.

GOOD WRITE

DATE: __/__/____
PAGE ______

Purpose of 'U' operation

The resultant tuples are coming from either relation R or S or both. If any duplicate tuples, they can be eliminated.

Most Imp → Union Type Compatability

→ No. of attributes in both the tables should be same
→ Datatypes of the corresponding columns should be same
→ Fieldname of both the tables can be different.

Eg:

R

ID	Name
101	Jones
103	Smith
104	Lal
107	Evan
110	Draw

S

ID1	Name 1
103	Smith
104	Lal
106	Baron
110	Draw

RUS

RUS → Result

ID	Name
101	Jones
103	Smith
104	Lal
107	Evan
110	Draw
106	Baron

→ By default, the fieldnames of table R will be shown in the result table.

b) Intersection operation ($\cap$)

Syntax : $\boxed{\text{Result} \leftarrow R \cap S}$

↳ It is used to select the common tuples in both the tables.

c) Set Difference ($-$)

↳ also called minus operation

Syntax : $\boxed{\text{Result} \leftarrow R - S}$

The result gives those tuples which are in R but not in S.

R−S

ID	Name
101	Jones
107	Evan

S−R

ID	Name
106	Baron

Properties

$R \cup S = S \cup R$

$R \cap S = S \cap R$

But, $R - S \neq S - R$

$R \cup (S \cup T) = (R \cup S) \cup T$

$R \cap (S \cap T) = (R \cap S) \cap T$

Cartesian Product (X)

$R \times S \rightarrow$ Result

where R and S are two relations having no common attribute. ~~where~~ If there is no common attribute, ~~they~~ then they can participate in cartesian product operation.

$R(A_1, A_2, A_3, \ldots, A_n)$

$S(B_1, B_2, B_3, \ldots, B_n)$

Rules

If R relation has K tuples and M attributes and S relation has L tuples and N attributes, then the resultant relation $(A_1, A_2, A_3, \ldots A_n, B_1, B_2, B_3 \ldots B_n)$

Therefore, the attributes of resultant relation = M+N and the tuples of resultant relation = L * K

Attributes are also called degree of a table

Eg:

A	B	C
a	a	a
b	b	b
c	c	c

P	S
x	x
y	y

Result

A	B	C	P	S
a	a	a	x	x
b	b	b	y	y
c	c	c	x	x
a	a	a	y	y
b	b	b	x	x
c	c	c	y	y

degree (Result) = 5

tuples (Result) = 6

DATE: __/__/__
PAGE ____

b) Binary Relational Operations

JOIN operation

→ If any relation R and S have at least one common attribute, then they can participate in JOIN operation.

→ It is denoted by symbol ⋈

Syntax: Result ← R ⋈<join condition> S

where <Join condition> is R.commonfield = S.commonfield.

Step 1 Perform cartesian product i.e R×S

$T_1 \leftarrow R \times S$

Step 2 Do selection operation (σ) for join condition.

Step 3 Optional

Do Project operation (π), if there are duplicate attributes any.

Eg: Q.

R

Name	Roll
X	101
Y	102
Z	301

S

Roll	Game
102	Cricket
302	Chess

Perform JOIN operation on the following tables.

DATE: __/__/____
PAGE ________

Steps

① $T_1 \leftarrow R \times S$

degree $(T_1) = 4$

Name	Roll	Roll	Game
X	101	102	Cricket
X	101	302	Chess
Y	102	102	Cricket
Y	102	302	Chess
Z	301	102	Cricket
Z	301	302	Chess

→ T_1

② $T_2 \leftarrow \sigma_{R.Roll = S.Roll} (T_1)$

Name	Roll	Roll	Game
Y	102	102	Cricket

→ T_2

③ $T_3 \leftarrow \pi_{Name, R.Roll, Game} (T_2)$

Name	Roll	Game
Y	102	Cricket

→ T_3

→ It is also called simple join or equi join

DATE: __/__/__
PAGE

OUTER JOIN (using relational algebra)

(I) Left Outer ⟕
(II) Right Outer ⟖
(III) Full Outer ⟗ → { (I) ⟕, (II) ⟖, (III) ⟗ }

R

Name	Roll
x	101
y	102
z	301

S

Roll	Game
102	Cricket
302	Chess

(1) Result

Left outer join

Name	Roll	Roll	Game
x	101	NULL	NULL
y	102	102	Cricket
z	301	NULL	NULL

(2) Result

Right outer join

Name	Roll	Roll	Game
y	102	102	Cricket
NULL	NULL	302	Chess

DATE: __/__/___ PAGE ____

(3) Full outer join

Result

Name	Roll	Roll	Game
x	101	NULL	NULL
y	102	102	Cricket
z	301	NULL	NULL
NULL	NULL	302	Chess

DBMS 2014 End Semester II

UNIT-3 Q6. (12.5 marks)

T_1

P	Q	R
10	a	5
15	b	8
25	a	6

T_2

A	B	C
10	b	6
25	c	3
10	a	5

Do the following operations using relational algebra.

(1) $T_1 \cup T_2$

Result

P	Q	R
10	a	5
15	b	8
25	a	6
10	b	6
25	c	3

GOOD WRITE

DATE : __/__/__
PAGE

(II) $T_1 \cap T_2$

Result

P	Q	R
10	a	5

(III) $T_1 - T_2$

Result

P	Q	R
15	b	8
25	a	6

(IV) $T_2 - T_1$

Result

P	Q	R
10	b	6
25	c	3

(V)

$T_1 \underset{T_1.P = T_2.A}{\bowtie} T_2$

Step 1 : Perform $T_1 \times T_2$

$T_1 \times T_2$

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	25	c	3
10	a	5	10	a	5
15	b	8	10	b	6
15	b	8	25	c	3
15	b	8	10	a	5

GOOD WRITE

DATE: __/__/____
PAGE ______

25	a	6	10	b	6
25	a	6	25	c	3
25	a	6	10	a	5

Step 2: $\sigma_{T1.P = T2.A} (T_1 \times T_2)$

Result

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	a	5
25	a	6	25	c	3

Step 3: Retrieve Q and B Condi

$\pi_{T1.Q,\, T2.B}$ (Result)

OR

$\pi_{T1.Q = T2.B} \left(\sigma_{T1.P = T2.A} (T1 \times T2) \right)$

Final Result

Q	B
a	b
a	a
a	c

Homework

1. $T_1 \bowtie_{T1.Q = T2.B} T2$
2. $T_1 \bowtie_{T1.R = T2.C} T2$

GOOD WRITE

Q: Perform $T1 ⟗_{T1.R = T2.C} T2$

Ans

Step 1 : $T_1 ⟕_{T1.R = T2.C} T2$

Step 2 : $T1 ⟖_{T1.R = T2.C} T2$

Step 3 : $T1 ⟗_{T1.R = T2.C} T2$

①

P	Q	R	A	B	C
10	a	5	10	a	5
15	b	8	NULL	NULL	NULL
25	a	6	10	b	6

→ left outer

②

P	Q	R	A	B	C
25	a	6	10	b	6
NULL	NULL	NULL	25	c	3
10	a	5	10	a	5

→ right outer

③

P	Q	R	A	B	C
10	a	5	10	a	5
15	b	8	NULL	NULL	NULL
25	a	6	10	b	6
NULL	NULL	NULL	25	c	3

→ full outer

Homework

③ $T1 ⟗_{T1.Q = T2.B} T2$

④ $T1 ⟗_{T1.P = T2.A} T2$

DATE: __/__/____
PAGE ______

HW: 2015 End Sem
Unit 1 Q-3

Homework Answers

① $\boxed{T1 \bowtie_{T1.Q = T2.B} T2}$

Step 1 $T1 \times T2 \rightarrow$

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	25	c	3
10	a	5	10	a	5
15	b	8	10	b	6
15	b	8	25	c	3
15	b	8	10	a	5
25	a	6	10	b	6
25	a	6	25	c	3
25	a	6	10	a	5

Step 2 $\sigma_{T1.Q = T2.B} (T1 \times T2)$

Result →

P	Q	R	A	B	C
10	a	5	10	a	5
15	b	8	10	b	6
25	a	6	10	a	5

② $\boxed{T1 \bowtie_{T1.R = T2.C} T2}$

Step 2 $\sigma_{T1.R = T2.C} (T1 \times T2)$

Result →

P	Q	R	A	B	C
10	a	5	10	a	5
25	a	6	10	b	6

GOOD WRITE

(8) $T1 \bowtie_{T1.Q = T2.B} T2$

Step 1.

P	Q	R	A	B	C
10	a	5	10	a	5
15	b	8	10	b	6
25	a	6	10	a	5

→ left outer join

Step 2

P	Q	R	A	B	C
15	b	8	10	b	6
NULL	NULL	NULL	25	c	3
10	a	5	10	a	5
25	a	6	10	a	5

→ Right outer join

Step 3

P	Q	R	A	B	C
10	a	5	10	a	5
15	b	8	10	b	6
25	a	6	10	a	5
NULL	NULL	NULL	25	c	3

→ Full outer join

(4) $T1 \bowtie_{T1.P = T2.A} T2$

Step 1.

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	a	5
15	b	8	NULL	NULL	NULL
25	a	6	25	c	3

→ left outer join

GOOD WRITE

DATE: __/__/____
PAGE ________

Step 2

P	Q	R	A	B	C
10	a	5	10	b	6
25	a	6	25	c	3
10	a	5	10	a	5

→ Right outer join

Step 3

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	a	5
15	b	8	NULL	NULL	NULL
25	a	6	25	c	3

2015 Paper UNIT-1 Q3.

Q3 a) $T \leftarrow \pi_{Pname}(\sigma_{Price < 200}(PARTS))$

b)

Relational Database Design and Normalisation

→ The goal/objective of relational database design is to generate a set of relational schemas/relations/tables that allows us to store information/data without redundancy and also retrieves the data easily.

Q. What constitute a bad database design?

OR

What are the factors that constitute a bad database design?

Some problems while storing data in database are :-

1. Redundancy
2. Various kinds of inconsistencies / anamolies
   a) Insertion Anamolies
   b) Deletion Anamolies
   c) Updation Anamolies.

Solutions for these problems is called a process Decomposition.

Decomposition → It is a process of decomposing a relation R into sub-relations in order ~~to remove of~~ reduce data redundancy.

DATE: __/__/__
PAGE ____

Supplier (Sname, Address, Item, Price)

Sname	Address	Item	Price
Havells	Civil Lines	Cable	400
Havells	Civil Lines	Fan	500
Bajaj	South-Ex	Fan	300
Bajaj	South-Ex	Greyser	500
XYZ	Rohini	X	X
X	X	Fan	500

data redundancy

Inconsistent data

Relation & Supplier

SA (Sname, Address)

Sname	Address
Havells	Civil Lines
Bajaj	South-Ex
XYZ	Rohini

P.K = {Sname}

SIP (Sname, Item, Price)

Sname	Item	Price
Havells	Cable	400
Havells	Fan	500
Bajaj	Fan	300
Bajaj	Greyser	500

P.K = {Sname, Item}
F.K = {Sname}
refer Supplier

Scanned by CamScanner

Relational Database Design

Functional Dependency (FD)

- It is a constraint / rule / relationship between two set of attributes X and Y of a relation R.
- It is the inter relationship among attributes.

FD is denoted by $\boxed{X \rightarrow Y}$

LHS (Determinant) — X

(Dependent) RHS — Y

Interpreted as : X functionally determines Y
X determines Y
Y is dependent on X.

FD holds only if

Case 1 : Every X attribute's value uniquely identifies Y attribute's value.

Case 2 : For any two tuples t_1 and t_2 of R, if we have

$$t_1[X] = t_2[X]$$

then we must have

$$t_1[Y] = t_2[Y]$$

Example 1 (Based on case 1)

A	B
1	2
2	3
3	4
4	9
5	6

unique values

Here, $A \rightarrow B$
(A determines B)

Example 2 (Based on case 2)

A	B
1	2
1	2
2	3
4	5
4	5
2	3

→ Is FD A→B holds?

Ans. Yes, A determines B

Example 3

A	B
1	2
1	3
2	3
4	5
6	7

Case 2 is violated

~~Is~~ A→B holds?

NO, A does not determine B.

Q Given R(X, Y, Z)

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Check if the FD holds for the following:

1) X → Y No
2) Y → Z Yes, FD holds
3) Z → Y No
4) XY → Z
5) YZ → X

1) X → Y

Y has diff values

X	Y
1	4
1	5
1	6

2) Y → Z

Y	Z
4	2
5	3
6	3
2	2 ✓

3) Z → Y

Z	Y
2	4
3	5
3	6

Y has diff values.

4) XY → Z

14 → 2
15 → 3
16 → 3
32 → 2 ✓

5) YZ → X

42 → 1
53 → 1
63 → 1
22 → 3 ✓

Q. S

Sno	Sname	Status	City
S1	Smith	20	London
S2	John	10	Paris
S3	Joe	30	Paris

Check for the following FD's:

1) Sno → Sname ✓
2) Sno → Status ✓
3) Sno → City ✓
4) City → Sno ✗

Q. R(A B C D)

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b2	c2	d3
a3	b3	c2	d4

DATE: __/__/__
PAGE ____

Check : 1) A → B ✗
2) A → C ✓
3) A → D ✗
4) B → C ✗
5) B → D ✗
6) C → D ✗
7) AB → D ✗

AB → D
a_1 b_1 → d_1
a_1 b_2 → d_2
a_2 b_2 → d_2] Different values of D.
a_2 b_2 → d_3]
a_3 b_3 → d_4

Therefore, No.

Inference Rules for Functional Dependency
Or
Armstrong's Axioms

Suppose, we have given a set of FD's
+
X, Y and Z attributes over a relation R.

Requirement :- R (X, Y, Z) and set of FD's

Rule 1 : Reflexibility Rule or Trivial Rule

If we have a FD X → Y and it is given that $Y \subseteq X$,
then X → Y is called Trivial FD.

GOOD WRITE

Eg: $X \to X$ ✓
$XY \to Y$ ✓
$XY \to Z$ ✗
$XXY \to Y$ ✓

Rule 2 : Augmentation Rule

If Given $X \to Y$ holds and Z is a set of attributes, then we have

$\boxed{XZ \to YZ}$ holds

Rule 3 : Transitivity Rule

If it is given that $X \to Y$ and $Y \to Z$, then we have $X \to Z$.

Rule 4 : Union Rule or Additive Rule

If $X \to Y$ and $X \to Z$, then $X \to YZ$.

OR

if $X \to Y$ and $X \to Z$, then $F \models X \to YZ$

↳ infer/conclude

Rule 5 : Decomposition Rule or Projective Rule

If $X \to YZ$ (Given)
then $X \to Y$ and $X \to Z$.

OR

if $X \to YZ$, then $F \models X \to Y$ and $X \to Z$

DATE: / /
PAGE

Rule 6 : Pseudo-transitivity Rule

If $X \rightarrow Y$ and $WY \rightarrow Z$, then $XW \rightarrow Z$

Q. Prove the following

1. It is given that $X \rightarrow Y$ and $Z \subseteq Y$, then prove $X \rightarrow Z$

Ans $Z \subseteq Y \Rightarrow Y \rightarrow Z$. (Acc to Rule 1)

If $X \rightarrow Y$ & $Y \rightarrow Z$.

then $X \rightarrow Z$. (Acc to Rule 3)

Hence, proved.

2. Given : $A \rightarrow B$, $C \rightarrow D$ with $C \subseteq B$. Show that $F \models A \rightarrow D$

Ans $C \subseteq B \Rightarrow B \rightarrow C$ (Rule 1)

If $A \rightarrow B$, $B \rightarrow C$, then $A \rightarrow C$ (Rule 3)

If $A \rightarrow C$, $C \rightarrow C$, then $A \rightarrow D$ (Rule 3)

Hence, proved.

3. $A \rightarrow B$, $C \rightarrow X$, $BX \rightarrow Z$ then prove that $AC \rightarrow Z$

Ans

If $C \rightarrow X$, $BX \rightarrow Z$, then $BC \rightarrow Z$ (Rule 6)

If $A \rightarrow B$, $BC \rightarrow Z$, then $AC \rightarrow Z$ (Rule 6)

Hence, proved

4. $W \rightarrow Y$ and $X \rightarrow Z$ then prove $F \models WX \rightarrow Y$

Ans $X \rightarrow Z$

$XW \rightarrow ZW$ (Rule 2)

$W \rightarrow Y$

$WZ \rightarrow YZ$. (Rule 2)

If $XW \rightarrow ZW$, $WZ \rightarrow YZ$, then $XW \rightarrow YZ$ (Rule 3)

DATE: __/__/__
PAGE

TYPES OF FD's

1. Trivial Functional Dependency
2. Fully Functional Dependency
3. Partial Functional Dependency
4. Transitive Functional Dependency.

Eg: Student (Rollno , Name, Marks)

Rollno → Name ⇒ Name is fully FD on Rollno.

Eg: Works (Eno, Cno , Salary)

Eno, Cno → prime attributes

Salary → Non-Prime attribute

Eno → Salary
Cno → Salary] Partial FD
Eno, Cno → Salary ⇒ Fully FD
Eno, Cno → Eno ⇒ Trivial FD

Q. Shipment (Sno, Pno, Jno , Qty)
Find the types of following FD's.

(I) Sno, Pno, Jno → Qty ⇒ Fully FD
(II) Sno, Pno → Qty
(III) Pno, Jno → Qty
(IV) Sno → Qty] Partial FD
(V) Sno, Pno, Jno → Sno ⇒ Trivial FD

Q. Given (R A,B,C,D,E) and FD's are A→C, B→C , B→D , C→E . Find out transitive FD's.

DATE: __/__/__
PAGE: ____

Ans. $A \to E$ (Since, $A \to C$ and $C \to E$ holds)
$B \to E$ (Since, $B \to C$ and $C \to E$ holds)

Normal Forms (VVI)

→ are used to ensure that the various types of anamolies and consistencies are not introduced into the DB.

→ A relation / relational schema R is said to be in particular normal forms if it satisfy a certain set of conditions / rules based on the FD's.

Types of Normal forms

syllabus
1. 1NF or First Normal Form
2. 2NF or Second Normal Form
3. 3NF or Third Normal Form
4. Boycee - Codd Normal Form (BCNF)
5. 4NF or Fourth Normal Form
6. 5NF or Fifth Normal Form

1NF

Consider an unnormalised relation P.

P

Prjno	Pname	Eno	Ename	category	Hrate
P001	E:AS	F001	X	A	700
		F002	Y	B	500
P002	EIS	E006	ABC	A	700
		E007	XYZ	A	500

Scanned by CamScanner

DATE: __/__/__
PAGE

Definition

↳ A relation R is in 1NF, if the values in the domain of each attribute can contain atomic value.

- Atomic means single value.
- The value cannot be multi-valued or set of values.

OR

In informal way, A table is 1NF if-

- Each table cell contains a single value.
- Each record need to be unique.

Eg: Student

Rollno	Name	Age	Marks
1	X	18	
2	Y	19	95
3		18	

Into 1NF →

Student

Rollno	Name	Age	Marks
1	X	18	NULL
2	Y	19	95
3	Z	18	98

P.K = {Rollno}

Eg: Emp

Empno	Ename	Sal	Phone
1	X	5K	989, 121
2	Y	6K	NULL
3	Z	7K	223, 777

→ multivalues not allowed

↓ Into 1NF

EmpNew

Empno	Ename	Sal	Phone
1	X	5K	989
1	X	5K	121
2	Y	6K	NULL
3	Z	7K	223
3	Z	7K	777

P.K = {Empno, Phone}

2NF

DATE: __/__/__
PAGE ______

→ It is based on the concept of Fully Functional dependency.

Definition:- A relation R is 2NF if
a) it should be in 1NF
b) Every non-prime attributes are fully functional dependent on primary key. (No partial FD should occur)

Eg① Student (Rollno, Name, Age, Marks)

Rollno → Name, Age, Marks

Is Student in 2NF?

Prime Attribute ~~PK~~ = {Rollno}
Non-Prime Attributes = {Name, Age, Marks}

Eg②

Emp-prj (Eno, Pno, hours, ename, pname, ploc)

~~Is the above t~~ Given FD's are:

FD1: eno, pno → hours
FD2: eno → ename
FD3: Pno → Pname, ploc

Q. Is the above table in 2NF? No
Q. If not, what is the higher normal form? 1NF
Q. How to decompose the above relation, such that the table is in 2NF?

Prime Attributes = {Eno, Pno}
Non Prime Attributes = {hours, ename, pname, ploc}

Scanned by CamScanner

In FD1, ~~all~~ non-prime attributes are fully dependent on primary key. ∴ It is in 2NF

In FD2, there exist partial FD. ∴ it is in 1NF

In FD3, there exist partial FD. ∴ it is in 1NF.

↳ If all three FD's are in 2NF, then higher normal form will be 2NF

→ But here, higher normal form is 1NF.

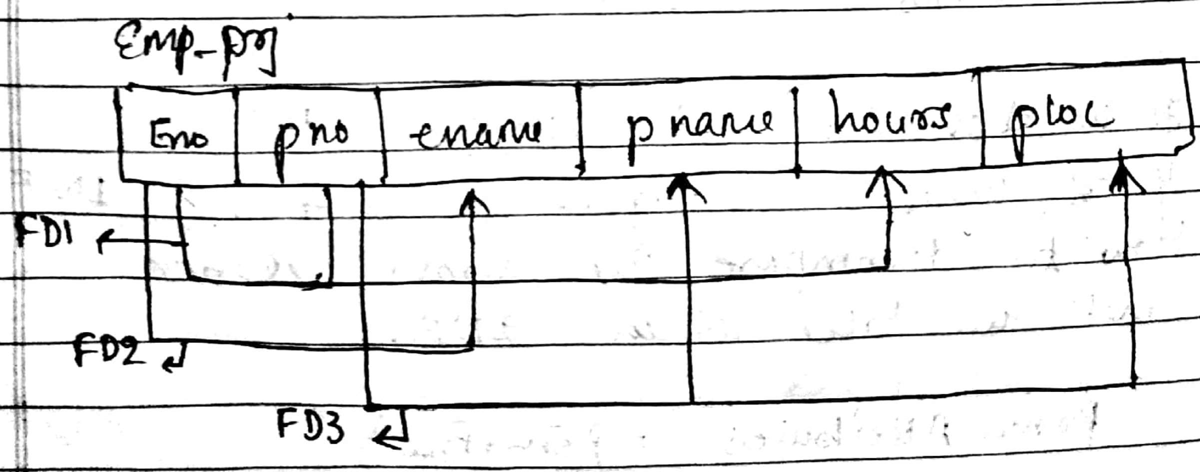
ep1 (eno, pno, hours) with FD eno, pno → hours
F.K F.K

ep2 (eno, ename) with FD eno → ename
P.K

ep3 (pno, pname, ploc) with FD pno → pname, ploc
P.K

Now all the three tables are in 2NF.

Schema Diagram (same question)



DATE: __/__/____
PAGE ______

3NF

↳ It is based on the concept of Transitive F.D.

Definition : A relation R is in 3NF if
a) it is in 2NF
b) it has no transitive FD i.e. no non-prime attribute of R is transitive functional dependent on the primary key.

Eg: R(X, Y, Z)
Given FD's : ① X → Y
② Y → Z.

In FD1, Y is fully dependent on X as X is the primary key

In FD2, Z is not fully dependent of Y.

R1 (X, Y) ↳ F.K.
R2 (Y, Z)
↓
P.K

These two tables are not transitive and are in 3NF

Alternative definition of 3NF

↳ Based on BNF
A relation R is in 3NF if for all FD X → Y in FD F
a) it should be in 2NF
b) and atleast one of the following 3 conditions hold
1. X → Y is trivial i.e Y ⊆ X
2. X is a super key of R
3. Y is a prime attribute of R.

Eg: R (A, B, C)

F = ~~FDs~~ { AB → C
C → B }

Super Key = { AB, AC }
Is this relation in 3NF ?

Prime Attributes = { A, B, C }
Non-prime attributes = { NULL }

FD1 : AB → C
X → Y

Here, X is ~~a~~ a super key

FD2 : C → B
X → Y

Here, X is not a super key but Y is a prime attribute.

Hence, relation R is in 3NF.

BCNF

A relation is in BCNF if it is
a) in 3NF
b) and atleast one of the following conditions hold
1. X→Y is trivial i.e $Y \subseteq X$.
2. X is a super key of R

Eg: R (A, B, C)
F = { AB → C
C → B }

DATE: __/__/__
PAGE ____

Super Key = {AB, AC}
Is this relation in BCNF?

FD1 : AB → C
~~FD2~~
X → Y

Here, X is a super key.
∴ relation is BCNF

FD2: C → B
X → Y

Here, X is not a super key. ∴ not in BCNF but in 3NF

⇒ Higher level = 3NF

Hence, Relation R is not in BCNF.

key

<u>2015 Paper</u>

<u>Q5</u> b) A table R (A, B, C, D, E) is with following FD's

A → BC
B → D
CD → E
E → A

1) What are the candidate keys?
2) Is this table in 2NF / 3NF / BCNF?

⇒ For finding out the candidate keys an algorithm is followed.

Scanned by CamScanner

Algorithm to find the closure of a set of attributes (X) i.e X^*

This algorithm is used for determining the superkey, candidate key from the relation/table R

Algo

Input: a) Relation / Table R with attributes
b) Set of FD's F $X \rightarrow Y$

Process: Steps

$X^* \rightarrow$ ~~col~~ closure of X

Eg: R(A, B, C, D)

FD's $A \rightarrow B$
$B \rightarrow D$
$C \rightarrow B$

Find the 1) closure of A i.e A^*
2) closure of B i.e B^*
3) closure of C i.e C^*

Ans $A^* = A$
using FD $A \rightarrow B$
$A^* = A \cup B = AB$
Using FD $B \rightarrow D$
$A^* = AB \cup D = ABD$

$\boxed{A^* = ABD}$

NOTE: closure of an attribute should contain all the attributes of the relation to become a super key

DATE: / /

PAGE

$B^* = B$

Using FD $B \to D$

$B^* = B \cup D = BD$

$\boxed{B^* = BD}$

$C^* = C$

Using FD $C \to B$

$C^* = C \cup B = CB$

Using FD $B \to D$

$C^* = CB \cup D = CBD$

$\boxed{C^* = CBD}$

No single attribute can become a super key

Now we have to consider two attributes together.

NOTE: A and C are not on the right hand side

∴ we take A and C together

$(AC)^* = AC$

Using FD $A \to B$, $C \to B$ and $B \to D$

$AC \cup B \cup D$

$\boxed{(AC)^* = \{A, B, C, D\}}$

Q. R (A, B, C, D, E, F)

FDs
$A \to C$
$C \to D$
$D \to B$
$E \to F$

Find out the keys.

Scanned by CamScanner

Ans. $A^* = ACDB$

$C^* = CDB$

$D^* = DB$

$E^* = EF$

Now, we will group A and E

$(AE)^* = AE$

Using FDs $A \to C$, $C \to D$, $D \to B$, $E \to F$

$(AE)^* = \{A, E, C, D, B, F\}$

2015 Paper

Ans 9 b

$A^* = \{A, B, C, D, E\}$

$B^* = \{A, B, C, D, E\}$

$(CD)^* = \{A, B, C, D, E\}$

$E^* = \{A, B, C, D, E\}$

Super Key = $\{A, B, CD, E\}$

FD 1 : $A \rightarrow BC$

$X \rightarrow Y$

X is in super key

∴ BCNF

FD2 : $B \rightarrow D$

$X \rightarrow Y$

X is a super key

∴ BCNF

FD3 : $CD \rightarrow E$

Since, CD is a super key

∴ BCNF

FD4 : $E \rightarrow A$

{E is a super key}

Since, all the FDS are in BCNF

∴ higher normal form is BCNF.

Correction in Madam Notes

UNIT 3 Pg 6

FD 4 is fully FD not partial FD