Date: 15/3/23

RESULT $\leftarrow R \bowtie_{Ssn=Essn} R2$ } Previous ques using JOIN condition

So, don't have to use TEMP wala part in previous ques. → This way it will not take more space if we use JOIN condition in R2.

NOTE: In JOIN operation, the list of operators that are acceptible are =, >=, <=, >, <, ≠

These types of operations are called 'theta' JOIN.

When the operation is using = sign, it is called 'equi' join.

Otherwise, all operations are called 'θ' join, except 'equi' join.

Q Combine each project tuple with the department tuple.

Natural join, Rho

RESULT $\leftarrow$ PROJECT * $\rho_{(Dname, Dnum, Mgr\_ssn, Mgr\_start\_date)}$ (DEPARTMENT)

Pname	Pnumber	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date

→ The new Relation will look like this using Natural Join.

We have made Dnumber as Dnum using operation & then using Natural Join eliminates the duplicate column

Q Retrieve the name and address of all employees who work for Research Department.

Ans $R1 \leftarrow \sigma_{Dname = 'Research'}$ (DEPARTMENT)

$R2 \leftarrow R1 \bowtie_{Dnumber = Dno}$ (EMPLOYEE)

RESULT $\leftarrow \pi_{Fname, Address}(R2)$

Q Retrieve the name of all employees who have no dependent.

Ans $R1 \leftarrow \pi_{Ssn}$ (EMPLOYEE)

$R2 \leftarrow \pi_{Essn}$ (DEPENDENT)

$R3 \leftarrow R1 - R2$

$R4 \leftarrow R3 \bowtie_{Ssn = Ssn}$ (EMPLOYEE)

RESULT $\leftarrow \pi_{Fname}(R4)$

Date: 3/4/23

Division Operation (÷)

For a tuple (t) to appear in a result (T) of the division, the values in t must appear in R in combination with every tuple in S.

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a_1
a_2
a3

T (Result)

b1
b4

When you see 'all' word written in the ques, just use division operation.

Q Retrieve the Ssn of employees who works on all the projects that John Smith works on.

Ans

$R1 \leftarrow \sigma_{fname='John' \text{ AND } Lname='Smith'}$ (EMPLOYEE)

$R2 \leftarrow R1 \bowtie_{Ssn=ESsn}$ (WORKS_ON)

$S \leftarrow \pi_{Pno}$

$R \leftarrow \pi_{ESsn, Pno}$

$T \leftarrow R \div S$

Aggregate Functions & Grouping

1. Applied on numeric values.
   The operation includes SUM, AVERAGE, MAXIMUM, MINIMUM & COUNT

2. Grouping the tuples in a relation by the value of some of their grouping attributes and then applying aggregate functions independently on each group.

Date:
Page:

Prototype -

grouping attribute $\mathcal{F}$ function list

NOTE:
[∵ This F can be any alphabet or symbol like, G, A, B, etc. can be anything]

Q. Retrieve each department number, the no. of employees in the department and their average salary.

$$_{Dno}\mathcal{F}_{COUNT\ SSN,\ AVERAGE\ SALARY}(EMPLOYEE)$$

If we skip the rename operation on this query, then the default table name will look like -

Dno	COUNT_SSN	AVERAGE_SALARY
5	10	100000
6	15	800000
7	10	700000

If we rename, then

$$\rho_{R(Dno,\ no.\ of\ employee,\ Salary)}\left({}_{Dno}\mathcal{F}_{COUNT\ SSN,\ AVERAGE\ SALARY}(EMPLOYEE)\right)$$

Date: NOTE:
Page:

R → Table name → [If we don't want to name a table, just remove R from query]

Dno	no. of employee	Salary
5	10	100000
6	15	800000
7	10	700000

NOTE:

1. If no grouping attribute are specified, the functions are applied to all the ~~functions~~ tuples in the relation.

2. Duplicates are not eliminated when an aggregate functions are applied.

3. NULL values are discarded.

If there are 5 employee in the company, & one's emp. salary is NULL, then avg salary will be divided by 4 not by 5.

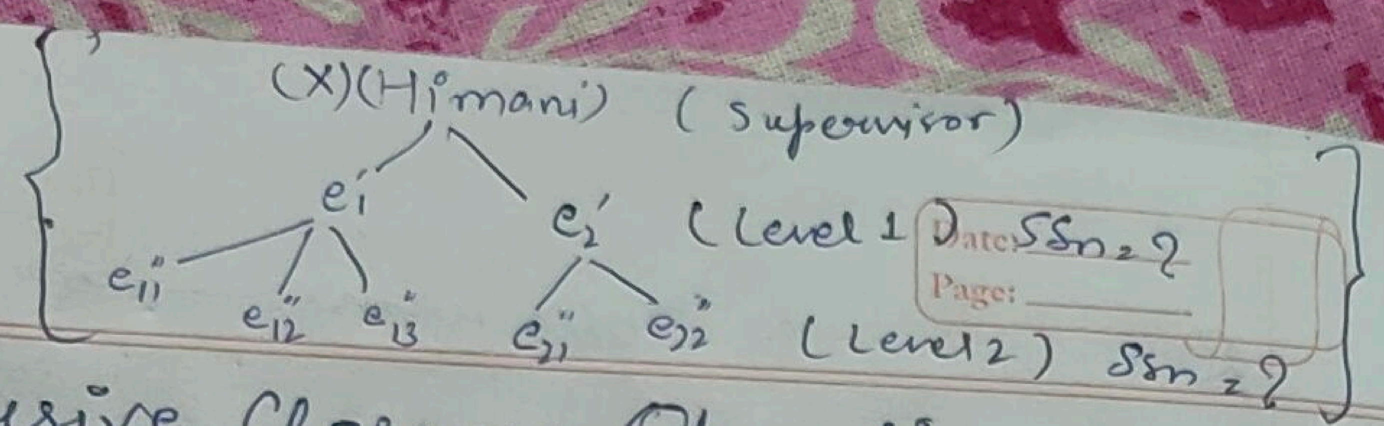
Outer Join Operations

Left Outer Join: $R ⟕_{\text{Join condition}} S$

Right Outer Join: $R ⟖_{\text{Join cond}} S$

Full Outer Join: $R ⟗_{\text{Join cond.}} S$

2023/05/05 22:57



Recursive Closure Operation

Level 1

Q Retrieve all employee Ssn who are directly supervised by James. Ssn of James is 888665555.

Date: 5/4/23

Normalization

In-formal guidelines for Relational Schema-

1. Making sure that the semantics (meaning) of the attributes is clear in the schema.

2. Using the redundant information in the tuples and update ~~ale~~ anomalies.

3. Reducing null values in the tuples.

4. Disallow the generation of ~~superious~~ ~~superious~~ spurious (fake) tuples.

Explanation of 1. Design a relation so that it is easy to explain.
Do not combine attributes ~~from~~ of multiple relations into a single relation

2.(i). REDUCING THE REDUNDANT INFORMATION

The goal of schema design is to minimize the storage space used by the base relation (i.e, which exists physically)

Date:
Page:

Emp-Dept

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr-ssn
~~ABC~~	~~0~~					
ABC	1	X42	X42	D1	Research	NULL
PQR	2	X42	"	D2	Admin	1
RST	3	X42	"	D2	Admin	1

Redundant values - repeated which will consume more space.

& this will not happen if we make two separate tables - Emp & Dept.

(ii) Update Anomalies

Insertion, Deletion, Modification

(a) Insertion.

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr-ssn
X42	1		~~D~~	D1	Research	1
X42	2			D2	finance	2
ABC	3			D2	finance	2
PQR	4			D3	finance	3

Insertion anomaly

This is the insertion mistake done by the Entry operator. He/she has to remember for which Dept. the manager has its Ssn or which Dept have which dep. number

Date:
Page:

Xamp server will not throw error but it must throw error as-

(i) Dept. number is not same as it should be.

(ii) the manager Ssn is also different for the same department - finance.

If tables would be different, then it will throw an error.

(b) Deletion

If we delete a Employee's data, it would lead to delete some ~~info~~ inf. information. Eg,

X42	D3	HR	1	X delete.
PQR	D1	finance	2	
ABC	D1	finance	2,	

Then, if we delete X42's employee row or information, the HR department will automatically gets deleted, because there is only one entry of HR. which is a ~~info~~ ~~big~~ error.

2023/05/05 22:57

Date:
Page:

(c) Modification

If we chang Dno. of finance, then all the rows need to be modified separately. This is modified anomaly.
If there will be separate tables, it wouldn't have happened.

Explanation Soln → Design ~~the~~ a base relation so that no update anomalies are present in the relation.

3. ~~En~~ Emp_office Ph

Ssn	Ename	Office Phone.

Only department heads will get their office Phone no's, so in this combined table, most of the values will be the NULL in the office Phone column.

And if there will separate tables, then no NULL values will be there.

Date:
Page:

As far as possible, reduce the NULL values in the relation.

4. EMP_LOCS

Ename	Plocation

EMP_PROJ1

Ssn	Pnumber	Hrs	Pname	Plocation

EMP_PROJ

Ssn	Pnumber	Hrs	Ename	Pname	Plocation

Design a relation schema so that they can be joined ~~on~~ attributes that are appropriately related (PK, FK) pairs in such a pay that guarantees no ~~super~~ spurious tuples

In these tables, Plocation is neither a ~~PK~~ PK nor a FK, it just a simple attribute. So, there is not uniqueness & fake tuples will get generated.

P.T.O

2023/05/05 22:57

Formal Guidelines Of Normalization

NF → Normal Forms.

In syllabus: 1 NF, 2 NF, 3 NF

1 NF	BCNF
2 NF	4 NF
3 NF	5 NF

→ Total NF which exits

- **Functional Dependency (F.D) or (f.d)**

F.D is denoted by $X \longrightarrow Y$.
Between two sets of attributes X & Y that are subset of R specifies a constraint on the possible tuples that can form a relation state (r) of R.

$X \rightarrow Y \rightarrow$
X goes to Y means there is a F.D from X to Y or Y is functionally dependent on X

E.g: Given a value of Ssn, we can uniquely determine an Ename or Ename is functionally determined by Ssn.

$Ssn \longrightarrow Ename$.

E.g:- Pnumber ⟶ {Pname, Plocation}
for multiple tuples

Read → Left → Right, Right → Left

Q Given a relation -

NOTE: It holds:-

A	B	C	D
a_1	b1	c1	d1
a_1	b2	c2	d2
a_2	b2	c2	d3
a_3	b3	c4	d3

Test whethere the following F.D holds or not?

(i) $B \longrightarrow C$
True as $(c_1 \& b_1)$, (c_2, b_2), (c_4, b_3).
So, it holds F.D.
If there will be suppose (c_4, b_3), $(c_4, b1)$, then FD not holds.

(ii) $C \longrightarrow B$
True as (b_1, c_1), (b_2, c_2), (b_3, c_4).
It holds

(iii) $\{A, B\} \longrightarrow D$

E.g:- a_1 b_2 d2
a_2 b2 d2
a_3 b_2 d2

This is a pattern. So, FD holds

Date:
Page:

FD

It holds.

because,

d1 & d2 are already unique. we'll focus on d3.

& d3 is following the same pattern

d3 → (a_2, b_2)

d3 → (a_3, b_3) → Same pattern.

If d3 → (a_4, b_2), then, it violates ~~to~~ FD.

It holds:-

(i) If there is some pattern which is being followed.

(ii) If there is same value-pair. or same data

e.g: $b_2 \rightarrow c_2$

$b_2 \rightarrow c_2$

~~End of topic~~

~~Basically same data को Point करना चाहिए।~~

(iv) $B \longrightarrow A$

It violates for a_1

(v) $D \longrightarrow C$

It violates for c_2

(vi) $\{A, B\} \longrightarrow C$

Date:
Page:

It holds.

For c_2, there is a pattern.

$$\begin{array}{cc|c} a_1 & b_2 & c_2 \\ a_2 & b_2 & c_2 \end{array} \rightarrow \text{Pattern.}$$

2023/05/05 22:57

Date: 12/4/23
Page:

Normalization of data

It can be considered as a process of analyzing the given relational schema based on their F.D and Keys to achieve the desirable properties of -

1. Minimizing the redundant data.
2. Minimizing the update anomalies.

Normal Form Test (NF)

It refers to the highest normal form conditions that a relational schema can achieve.

Prime Attributes

A Prime Attribute of R if it is a member of some candidate key of R.

Non-Prime attribute

If it is not the member of any candidate key of R.

Date:
Page:

E.g WORKS_ON

Ssn	Pno	Hrs

Ssn & Pno are the prime attributes of WORKS_ON &
Hrs is the non-prime attribute of WORKS_ON

1) 1 NF

- It states that the value of any attribute in the domain of that attribute must be atomic. E.g Non-atomic - Address (multi-valued attributes) Street City State
- It disallows multi-valued attributes & composite attributes

NOTE: We will check the RHS of the F.D

DEPARTMENT

Dname	Dnumber	Dmgrssn	Dlocation

→ Book Given

Ways of representing FD

Dnumber → {Dname, DmgrSsn, Dlocation}

↳ to solve ques, we use this reference.

2023/05/05 22:57

Since Dlocation is a multivalued attribute,
∴ The given relational schema is not in 1 NF.

There are three ways that can be used to decompose the relation into 1NF.

1st way → Best way

(1)

Dname	Dnumber	Dmgrssn

Dnumber	Dlocation

∴ Dnumber + Dlocation is PK

Dnumber	Dlocation
1	New Delhi
1	New York.

It's valid here that 1 is being repeated as it is in combination wise unique.

2nd way

(2)

Dname	Dnumber	Dmgr Ssn	Dlocation
Research	1	123	New Delhi
Research	1	123	LONDON
Admin	2	222	New Delhi
Admin	2	222	New York

Since the schema contains the
Disadvantage - A Redundant values, ~~which else not~~ it violates the normalization of data test.

Dnumber is PK, so we also can make Dlocation ~~as~~ a PK, so it combines to make PK

3rd way

(3)

Dname	Dnumber	Dmgrssn	Dlocation 1	Dloc 2	Dloc 3
Research	1	123	New Delhi	New York	NULL
Admin	2	222	London	NULL	NULL

This design suffers from introduction of NULL values.

2. 2NF

It is based on Primary Key, PK.

2NF test is based on full functional dependency.

Full Functional Dependency

A F.D $X \rightarrow Y$ is a full F.D if removal of any attribute A from X means that the dependency does not hold

Partial Dependency

A F.D $X \rightarrow Y$ is a partial dependency if some attribute A can be removed from X and the dependency still holds.

WORKS_ON

Ssn	Pno	Hrs

E.g of Full F.D

$\{Ssn, Pno\} \rightarrow Hrs$

It is a full F.D because

$\left.\begin{array}{l} Ssn \rightarrow Hrs \\ Pno \rightarrow Hrs \end{array}\right\} \rightarrow$ it does not holds.

as they separately can't determine the hrs.

E.g of Partial F.D

$\{Ssn, Pno\} \rightarrow Ename$

It is a Partial F.D because removal of any attribute from LHS doesn't affect the dependencies.

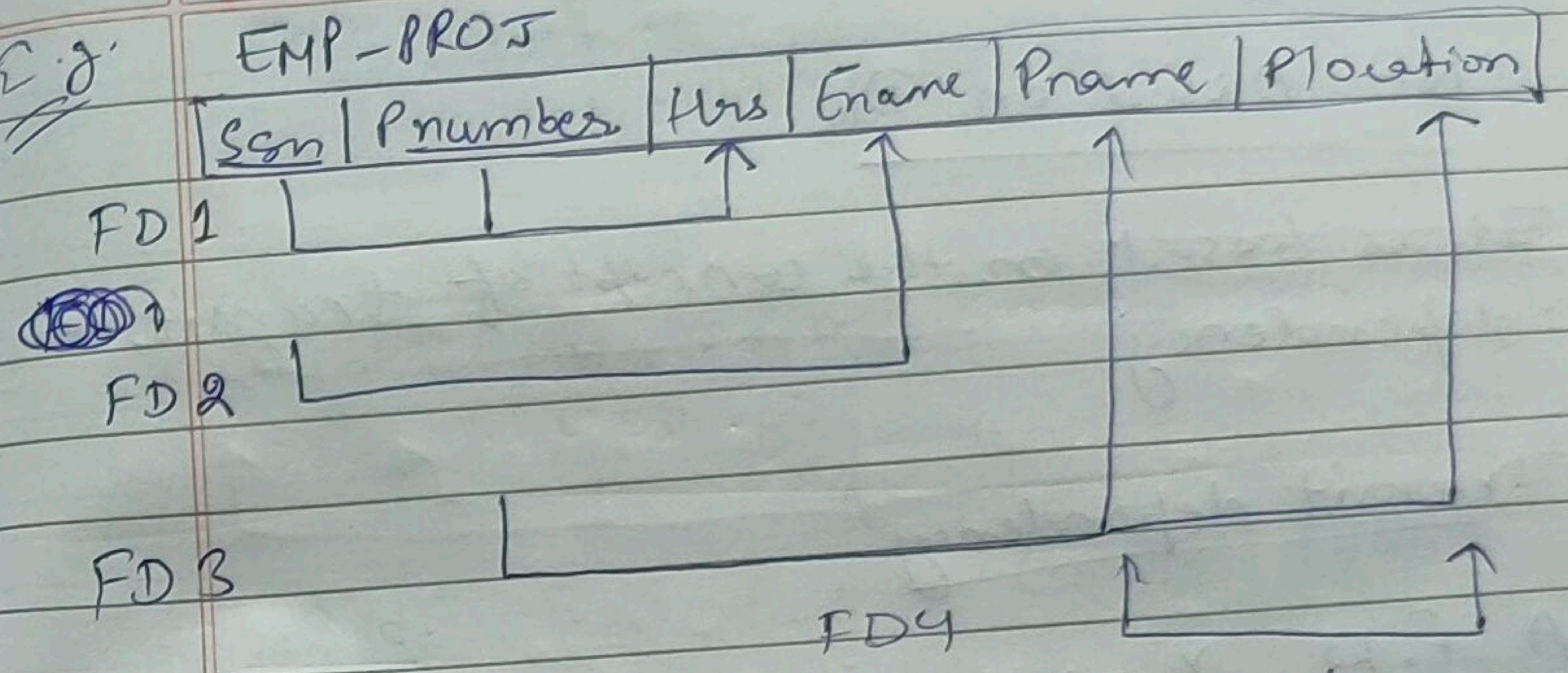
Seperately doesn't hold $\leftarrow \left\{\begin{array}{l} Ssn \rightarrow Ename \\ \cancel{Pnumber} \rightarrow Ename \end{array}\right\}$ These two dependencies holds.

This is correct $\leftarrow \{Ssn, Pnumber\} \rightarrow Ename$

① 2NF disallows Partial Dependency

② A Relation Schema, R is in 2NF if every non prime attribute, A in R is fully functionally dependent on the PK of R.

E.g.



↳ It is in 2NF.

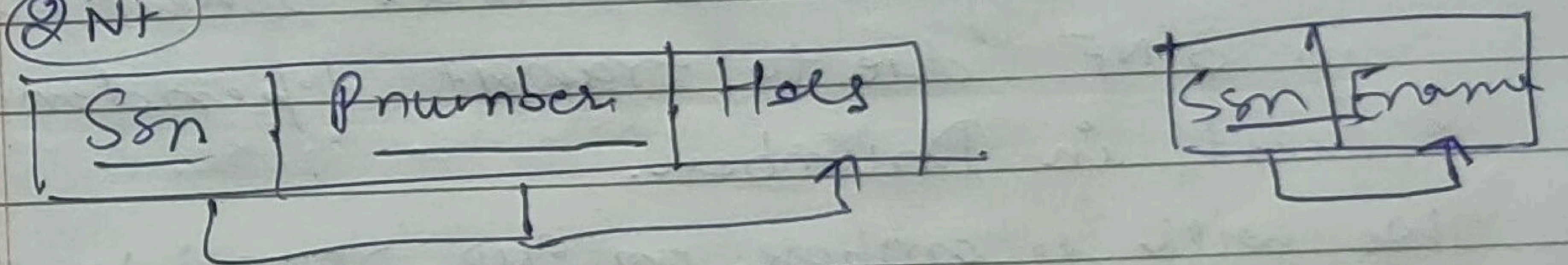
FD1 is in 2NF.

FD2 is a Partial Dependency.
∴ it is not in 2NF.

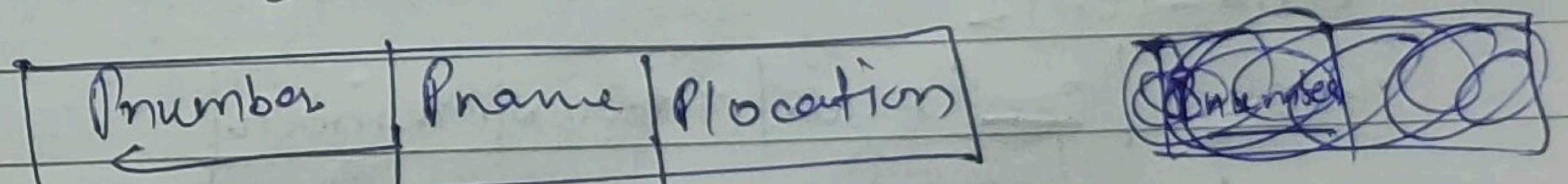
FD3 is not in 2NF because of Partial Dependency.

FD2 for FD2

2NF



FD3 → Making it to be suitable for 2NF.



Date: 15/4/23
Page:

3. 3NF Based on PK

1. It is based on the concept of transit dependency

Transit dependency

A F.D $X \rightarrow Y$ is transit if there exist a set of attributes Z that is neither a candidate Key nor a subset of any Key and both $X \rightarrow Z$ & $Z \rightarrow Y$ holds.

Eg. EMP_DEPT

It is Z (taken as z)

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgrssn

FD 1

FD 2

Since the given relational scheme is not in 3NF due to transit dependency exist in Dnumber.

We will decompose EMP-DEPT into 3NF

EMP-DEPT1

Ename	Ssn	Bdate	Address	Dnumber

Date:
Page:

EMP-DEPT2

Dnumber	Dname	Dmgrssn

2NF and 3NF Based on Candidate Key

- 2NF Based on Candidate Key

A relational schema R is in 2NF if every non-prime attribute A in R is fully functionally dependent on candidate Key of R (one of the C.K should be present on the LHS of the F.D).

CK (together can make a PK)

LOTS

Property_id#	Country-name	Lots #	Area	Price	Tax-rate

FD1

FD2

FD 3

FD4

In the given relation schema, FD3 is violating the test of 2NF. Decomposing
↳ because candidate Key is not in totality on LHS.

the relation LOTS# into 2NF

LOTS 1

Property-id #	Country-name	Lots #	Area	Price

FD 1

FD2

FD4

LOTS 2

Country_name	Tax-rate

FD3

• 3NF based on Candidate Key

A relation schema R is in 3NF if a F.D $X \to A$ holds in R then either

(a) X is a SK of R or

(b) A is a Prime attribute of R.

Since LOTS 1 is not in 3NF because of FD4.

Decomposing it into 3NF $\to$ LOTS 1 A & LOTS 1B

LOTS 1A

Property-id#	Country-name	Lots#	Area

FD1

FD2

LOTS 1 B

Area	Price

FD4

Now, LOTS1A, LOTS1B & LOTS 2 are in 3NF.

4. BC NF

A F.D $X \to A$ is in BC NF if X is a super Key of R.

Property-id#	Country-name	Lots#	Area

FD1

FD2

FD 3

FD 3 is violating the BCNF property.

Date:
Page:

hence decomposing the relational schema into BCNF.

Property_id #	Lots #	Area

FD1

Area	Country_name

FD3

FD2 gets deleted.

Closure :-

The ~~F~~ set of all dependencies that include F as well as all the dependencies that can be derived from F is called the closure of F and it is denoted by F^+.

Eg Q1. $F = \{Ssn \rightarrow \{Ename, Bdate, Address, Dnumber\}$

$Dnumber \rightarrow \{Dname, Dmgr\text{-}ssn\}\}$

Find closure of $\{Ssn\}^+$ & closure of $\{Dnumber\}^+$ and also find the Key.

Ans $\{Ssn\}^+ = \{Ssn, Ename, Bdate, Address, Dnumber, Dname, Dmgr\text{-}ssn\}$

$\{Dnumber\}^+ = \{Dnumber, Dname, Dmgr\text{-}Ssn\}$

Date:
Page:

Since closure of Ssn contains all the attributes. Therefore, Ssn is the Key.

Q2. $F = \{Ssn \rightarrow Ename,$

$Pnumber \rightarrow \{Pname, Plocation\}$

$\{Ssn, Pnumber\} \rightarrow Hrs\}$. find key.

Ans $\{Ssn, Pnumber\}^+ = \{Ssn, Pnumber, Hrs, Pname, Plocation, Ename\}$

Since closure of {Ssn & Pnumber} contains all the attributes. Therefore combination of Ssn & Pnumber is a ~~Pk~~ Key.

Q3 Consider the relation schema R(A, B, C, D, E) with the following dependencies ⇒

$AB \rightarrow C$

$CD \rightarrow E$

$DE \rightarrow B$

Is A B candidate Key of this Relation?

If not A B D is a candidate Key of this Relation?

2023/05/05 22:58

Date:
Page:

$\{A, B\}^+ = \{A, B, C\}$

AB's closure is not a key since it does not contains all the attributes of R.

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

Since closure of {ABD} has all the attributes. ∴ It is a key.

Q4. Find key of R (E, F, G, H, I, J, K, L, M, N)
Given the set of dependencies →
EF → G, F → IJ, EH → KL,
K → M, L → N

Ans {EFH} is the key.

$\{EFH\}^+ = \{E, F, G, H, K, L, I, J, M, N\}$

∴ It is a key because it's closure contains all the attribute.

Imp: 6 marks ques

Q5. Consider the relation R (A, B, C, D, E, F, G, H, I, J) and set of F.D's →

Date:
Page:

F = { AB → C
A → DE
B → F
F → GH
D → IJ }

What is the key of R? Decompose R into 2NF, then 3NF.

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

Since, AB's closure has all attributes.

∴ {A, B} is a key.

Since, we have only 1 key in the ques, therefore we'll find 2NF with taking it as PK. (Based on PK)

2NF

Partial dependencies that are violating 2NF.

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

{∵ A → DE
D → IJ
A → A (∵ trivial dependency)}

Date:
Page:

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

$B \rightarrow F$
$F \rightarrow GH$
$B \rightarrow B$ (trivial dependency)

Therefore, relations that are present in 2NF will be

$R1 = \{\underline{A}, D, E, I, J\}$

$R2 = \{\underline{B}, F, G, H\}$

$R3 = \{\underline{A}, \underline{B}, C\}$

R1 and R2 has transitive dependencies, therefore relations that are present in 3NF will be - because

$R11 = \{A, D, E\}$ $\quad \left[\begin{array}{l} A \rightarrow D \\ D \rightarrow IJ \end{array}\right]$
$R12 = \{D, I, J\}$

$R21 = \{\underline{B}, F\}$ $\quad \left[\begin{array}{l} B \rightarrow F \\ F \rightarrow GH \end{array}\right]$
$R22 = \{\underline{F}, G, H\}$

$\therefore$ Final set of Relations that are in 3NF are -

Date:
Page:

R11, R12, R21, R22 and R3.

6 marks ques CMA

Q. Consider the relational schema,
$R = (A, B, C, D, E, F, G, H, I, J)$ & FD's are -

$AB \rightarrow C$
$BD \rightarrow EF$
$A \rightarrow I$
$AD \rightarrow GH$
$H \rightarrow J$

Find the Key of R and then decompose the R into 2NF and then 3NF.

Ans $\{ABD\}$ is a key.

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

2NF

Partial dependencies that are violating 2NF.

- $\{A\}^+ = \{A, I\}$

$A \rightarrow I$ (given)
$A \rightarrow A$ (trivial dependency)

$R1 = \{\underline{A}, I\}$

2023/05/05 22:58

Note: Jiska closure nikalte hai, wohi PK bnta hai.

as A closure is finded already. $A \to I$

{∵ I will not be there, because it will become ~~redundant~~ data}.

- $\{AB\}^+ = \{A, B, C\}$

  A,B → C (given

  A, B → A B (trivial dependency)

  R2 = {<u>A</u>, <u>B</u>, C}

- $\{BD\}^+ = \{B, D, E, F\}$

  B D → BD (trivial dependency)

  BD → E F (Given)

  R3 = {<u>B</u>, <u>D</u>, E, F}

- $\{AD\}^+ = \{A, D, G, H, J\}$

  AD → AD (trivial dependency)

  H → J (given)

  AD → GH (given)

  R4 = {<u>A</u>, <u>D</u>, G, H, I}

Relations that are present in 2NF will be R1, R2, R3, R4.

Now, R4 has transitive dependencies,

Decomposing R4 into R41 & R42.

R41 = {A, D, G, H} because → [AD → GH, H → J]

R42 = {H, J}

~~∴ The Relations that are~~

∴ The final set of Relations that are in 3NF are R1, R2, R3, R41 & R42.

COVER

A ~~FD~~ set of F.D, F is said to cover another set of F.D, E if every F.D in E is also in F and vice-versa. (F.D's are equivalent).

4 marks ques (CMA)

Q7. For the following set of FD, check whether ~~F is a subset of G~~ F is a subset of G

$F \subseteq G$ (G covers F)

or $G \subseteq F$ (F covers G)

or $F = G$

or $F \neq G$.

2023/05/05 22:58

Date:
Page:

F = { A ⟶ C, AC ⟶ D,
E ⟶ AD, E ⟶ H }

G = { A ⟶ CD, E ⟶ AH }

Ans F =

F	G
A ⟶ C	A ⟶ CD
AC ⟶ D	E ⟶ AH
E ⟶ AD	
E ⟶ H	

1st Step
LHS → closure
& FD's G ki

F	G
$\{A\}^+ = \{A, C, D\}$	$\{A\}^+ = \{A, C, D\}$
$\{AC\}^+ = \{A, C, D\}$	$\{E\}^+ = \{E, A, D, H, C\}$
$\{E\}^+ = \{A, H, C, D, E\}$	

FD

IInd step → Check the dependencies of F in Step 1.

Since all F F.D's holds ∴ Work done by F can also be done by G. ∴ G covers F (F ⊆ G)	Now, G ⊆ F

Since, F cover G & G covers F, therefore it is equivalent.

Date:
Page:

Q. Find whether the two F.D's F & G are equivalent or not or whether one covers the other.

(i) F = { A → B,
B → C,
AB → D }

(ii) G = { A → B,
B → C,
A → C,
A → D }

Sol F = { A → B,
B → C,
A → C }

G = { A → B,
B → C,
A → D }

Ans (i) F =

F	G
A ⟶ B	A ⟶ B
B ⟶ C	B ⟶ C
AB ⟶ D	A ⟶ C
	A ⟶ D
$\{A\}^+ = \{A, C, D, B\}$	$\{A\}^+ = \{A, B, C\}$
$\{B\}^+ = \{B, C\}$	$\{B\}^+ = \{B, C\}$.
$\{AB\}^+ = \{A, B, C, D\}$	

Since, F covers G & G covers F, therefore it is equivalent.

2023/05/05 22:58

Date: 20/4/23
Page:

MINIMAL COVER

It is a set of functional dependencies, E is a minimal set of F.D's that is equivalent to E.

A minimal cover of a set of FD's, E is a minimal set of F.D that is equivalent to E.

We can always find out one minimal cover.

Q Find the minimal cover of F.

F: { A ⟶ D
BC ⟶ AD
C ⟶ B
E ⟶ A
E ⟶ D }

Step 1:- Singleton RHS

A ⟶ D
BC ⟶ A ——— (1)
BC ⟶ D
C ⟶ B
E ⟶ A
E ⟶ D

Date:
Page:

Step 2:- No extra elements on LHS (on step 1)

NOTE: Whenever you are making a modification, do it immediately before examine the other F.D's.

From (1),
{B}⁺ = {B}
{C}⁺ = {CBAD}

{C}⁺ = {CBD}
{C}⁺ = {CAD},
then also we can eliminate B from LHS.

Since (C closure) {C}⁺ has B, we can eliminate B from left hand side.

The set of F.D's are-

A ⟶ D
C ⟶ A
BC ⟶ D ——— (2)
C ⟶ B
E ⟶ A
E ⟶ D

[∵ C alone can found A & B, therefore no need of B]

{B}⁺ = {B}
{C}⁺ = {C, A, B, D}

Since {C}⁺ has B, we can eliminate B from LHS.

Date:
Page:

Therefore, the current set of F.D's are-

$A \rightarrow D$ —— (1)
$C \rightarrow A$ —— (2)
$C \rightarrow D$
$C \rightarrow B$
$E \rightarrow A$
$E \rightarrow D$

Step 3:- Remove redundant F.D's

NOTE: Same as in Step 2

From (1)

- $\{A\}^+ = \{A\}$

$\{A\}^+$ doesn't contain a D, therefore we need $A \rightarrow D$ in our final set.

- From (2), $\{C\}^+ = \{C, D, B\}$

Since $\{C\}^+$ doesn't contain A, therefore we need $C \rightarrow A$ in our final set.

- From (3), $\{C\}^+ = \{C, A, B, D\}$

Since $\{C\}^+$ contains a D, therefore we can eliminate $C \rightarrow D$.

Date:
Page:

After Modification has done -

Now, therefore the current set of FD's are-

$A \rightarrow D$
$C \rightarrow A$
$C \rightarrow B$ —— (4)
$E \rightarrow A$ —— (5)
$E \rightarrow D$ —— (6)

From (4), $\{C\}^+ = \{C, A\}$

Since $\{C\}^+$ doesn't contains B, therefore we need $C \rightarrow B$ in our final set.

From (5), $\{E\}^+ = \{E, D\}$

Since $\{E\}^+$ doesn't contains A, therefore we need $E \rightarrow A$ in our final set.

From (6), $\{E\}^+ = \{E, A, D\}$.

Since $\{E\}^+$ contains D, therefore we can eliminate $E \rightarrow D$.

The minimal cover of F is:-

$A \rightarrow D$
$C \rightarrow A$
$C \rightarrow B$
$E \rightarrow A$

2023/05/05 22:58

Date:
Page:

Q F : { $AB \to C$
$C \to A$
$BC \to D$
$ACD \to B$
$D \to EG$
$BE \to C$
$CE \to AG$
$CG \to BD$ }

Ans Step 1 :- Singleton RHS.

$A \to C$
$B \to C$
$C \to A$
$B \to D$
$C \to D$
$A \to B$
$C \to B$
$D \to B$
$D \to EG$
$B \to C$
$E \to C$
$C \to AG$
$E \to AG$
$C \to BD$
$G \to BD$

$AB \to C$ —— ①
$C \to A$
$BC \to D$ —— ②
$ACD \to B$ —— ③
$D \to E$
$D \to G$
$BE \to C$
$CE \to A$
$CE \to G$
$CG \to B$
$CG \to D$

Date:
Page:

Step 2 :- No extra elements on LHS

From ①,
$\{A\}^+ = \{A\}$
$\{B\}^+ = \{B\}$

From ②,
$\{B\}^+ = \{B\}$
$\{C\}^+ = \{C, A\}$

From ③,
$\{A\}^+ = \{A\}$
$\{C\}^+ = \{C, A\}$
$\{D\}^+ = \{D, E, G\}$

Since $\{C\}^+$ has A, we can eliminate A from LHS.
The set of FD's are-

$AB \to C$
$C \to A$
$BC \to D$
$CD \to B$
$D \to E$
$D \to G$

$BE \to C$ — ④
$CE \to A$ — ⑤
$CE \to G$
$CG \to B$
$CG \to D$

From ④

$\{B\}^+ = \{B\}$
$\{E\}^+ = \{E\}$
No changes

From ⑤

$\{C\}^+ = \{C, A\}$
$\{E\}^+ = \{E\}$
∴ Eliminate E

$AB \to C$
$C \to A$
$BC \to D$
$CD \to B$
$D \to E$
$D \to G$
$BE \to C$
$C \to A$
$CE \to G$ — ⑥
$CG \to B$ — ⑦
$CG \to D$ — ⑧

From ⑥

$\{C\}^+ = \{C, A\}$
$\{E\}^+ = \{E\}$
No change

From ⑦,

$\{C\}^+ = \{C, A\}$
$\{G\}^+ = \{G\}$
No change.

From ⑧,

$\{C\}^+ = \{C, A\}$
$\{G\}^+ = \{G\}$ No Change

Final set of FD's are-

$AB \to C$
$C \to A$
$BC \to D$
$CD \to B$
$D \to E$
$D \to G$
$BE \to C$
$CE \to G$
$CG \to B$
$CG \to D$

Step 3:- Remove redundant F.D's.

$\{AB\}^+ = \{A, B.$

Date:
Page:

~~Chapter-17 (Indexing)~~

Date:
Page:

Chapter-17 (Indexing)

① The file or table, EMPLOYEE, DEPARTMENT .... is referred to as Data File.

② Tuple will be referred to as Data Record.

③ 3 or 4 data records together is called Disc Block or Block.

④ There is another file called Index File that has 2 columns -

1st column name - Index Field

2nd colum name - Pointer.

⑤ Index ~~table~~ file Tuples are called Records.

Index File is of two types -

① Dense Index

② Sparse Index or (non-dense Index)

① Dense Index - It has an index entry for every search key-value (every record or data record in the data file).

② Sparse Index - It has index entry for only some of the search-key value. A Sparse

Date:
Page:

Index has fewer enteries than the no. of records in the data file.

Types of Indexes

① Primary Index
② Clustering Index
③ Secondary Index

① Primary Index

- It is based on Key-value. (hence unique) and the data file is ordered (i.e, Sorted).

- Total no. of enteries in the index file is equal to no. of disc blocks or Anchor blocks.

- Primary Index is a Sparse Index since it includes an entry for each block rather than for every record.

Advantages -

① It occupies less space.
② Binary Search is performed on Index File rather than Data File. (less time)

Date:
Page:

Disadvantages

① In terms of Insertion & Deletion of Data Records.

② Clustering Index

NOTE: It is based on non-key. (hence, duplicates) but the Data File is ordered.

- A Clustering Index is a Sparse Index because it has an entry for every distinct value which is non key by definition & hence has duplicates.

Advantage - Searching.

Disadvantage

① In terms of Insertion & Deletion of Data Records.

Date: 26/4/23

③ Secondary Index

Secondary Index is based on

(a) on the Key field　　(b) non-Key field

(a) The data file is unordered & it is based on Key-value, hence unique.

Index-File →

- It has a distinct value for every record.
- It is a dense index because it has a entry for each or every record of the data file.
- Pointer points either to the block in which the record is stored or to the record itself.

Advantages

- Insertion is easy as there is no order.
- searching is also easy.

Disadvantages

If entry is deleted from Data file, it also has to be deleted from Index File.

(b) Based on non-Key field

- The file is unordered and it is based on non-Key value, hence duplicate.
- It is a sparse index as far as an index file is concerned.

Advantages

~~One level of direction is there.~~

Disadvantages

Searching takes more time, because an extra level of indirection is there.

INTRODUCTION TO TRANSACTION PROCESSING

It is a transaction because there is one database & multiple users are accessing it. E.g:- booking of train, movie tickets, etc.

A transaction is an executing program that forms a logical unit of database processing.

A transaction can include one or more database access operations ~~with~~ (insertion, deletion, updation, searching retrieval).

The basic database access operation are-

(1) read_item(X)
It reads the database item named as 'X' into a program variable also named as 'X'.

(2) write_item(X)
Writes the value of program variable 'X' into the data item named 'X'.

Executing the read-item(X)

The command includes -

① Find out the disk block address where the X resides

disk block address $\boxed{X}$

② Copy the disk block content into the main memory

main memory ⟵ $\boxed{\text{disk block content}}$

③ Then copying the program variable to main memory

main memory ⟵ program variable 'X'.

④ The content in the main memory should be again stored to disk block (secondary)

disk block ⟵ main memory.

Why concurrency control is needed?

E.g:- Flight Reservation System

There are two transactions 'T_1' and 'T_2'.

T_1	T_2
read_item(X);	read_item(X);
X = X - N	X = X + M
write_item(X);	write_item(X);
read_item(Y);	
Y = Y + N	
write_item(Y);	

X is the no. of reserved seats in the first flight.
Y is the no. of reserved seats in the second flight.
T_1 transfers N reservation from first flight to second flight.
T_2 reserves M seats in the second flight.

2023/05/05 22:59

Date:
Page:

X = 500
Y = 700
N = 100
M = 200

T_1	T_2
read-item(500);	read-item(400)
X = 500-100	X = 400+200
write-item(400);	write-item (600);
read-item(700);	
Y = 700 + 100	
write-item (800);	

① The lost-update problem
The problem occurs when two transactions that access the same database have their operations interleaved in such a way that makes the value of some ~~operations~~ database items incorrect.

T_1	T_2
read-item(X);	
X = X − N	
	read-item(X);
	X = X+M
write-item (X);	
read-item (Y);	
Y = Y+N	
	write-item(X);
write-item (Y);	

X = 80 ; M = 4
N = 5 ; Y = 50

T_1	T_2
read-item(80);	
X = 80-5	
	read-item(80);
	X = 80 + 4
write-item(84);	
read-item(50);	
	write-item(84);
Y = 50+5	
write-item(55);	

② Temporary Update Problem (Dirty Read Problem)

This problem occurs when one transaction updates the database item and then fails. Meanwhile, the update value is read by other transaction before it is changed back to its original value.

2023/05/05 22:59

T_1 T_2

read_item (X);
X = X − N
Write_item (X);

read_item (X);
X = X + M
write_item (X);

read_item (Y);

The transaction 'T_1' fails and must change the value of X back to its old value meanwhile 'T_2' reads the updated value of 'X'.

X = 80 ; N = 5 ; M = 4 ; Y = 50.

T_1 T_2

read_item (80);
X = 80 − 5
write_item (75);

At this point of time, T_1 transaction fails

↓

read_item (75);
X = ~~X+M~~ 75 + 4
write_item (79);

read_item (50);

2023/05/05 22:59