Date: 18/7/17

Structure of DBMS –

- DBMS is a layered architecture
- layers must consider concurrency control and recovery.

Query
Optimization &
Execution Relational.

Transaction Management
Storage Management.

Database Application Architecture –

① two-tier –
user interacts with db directly in a n/w.

② three-tier –
user interacts with db indirectly via an applⁿ server in a n/w.

Standalone applⁿ –
user interacts with db directly without n/w.

Database users –
① Application programmers
② Sophisticated User
③ Specialized User
④ Naive User

DBA –
Responsibilities –
① design logical / physical schema.
② storage structure & access specificatⁿ
③ schema modification.
④ granting user authority.
⑤ specifying integrity constraints
⑥ routine maintenance.

DBMS –
① H/w
② S/w
③ Data
④ Users
⑤ Procedures.

Unit - 1.

① Comparing DBMS with File System –

② Schema (it is abstraction)
↳ i) physical
ii) logical
iii) view

③ Data Models i) Hierarchical ii) Network
iii) ER

iv) RDBMS
v) OODBMS
vi) ORDBMS
vii) XML

4. DBMS Architecture
5. Db Appl^n Architecture
6. DBMS users
7. 5 Components of DBMS

Unit 2 - RDBMS

Primary Key and Foreign Key —

1. Primary Key —

It's a column in a table which has a capability to uniquely identify every single row of the table.

Primary Key = unique + Not NULL

- Duplication and NULL values are not allowed.

Company

CCode	CName
A101	ABC
A102	XYZ

2. Foreign Key —

It's a column in a table which always refer to the primary key in another table for it's existence. making it called a foreign key.

- A foreign key cannot contain those values which are not present in the primary key column.

Employee

EId	EName	CCode
1		
2		
3		

In Employee table, CCode is playing a role of foreign key and it is referring to CCode column in Company Table.

- In foreign key column the values could be duplicated.

Composite Key –

Example –

Supplier Table

SCode	PCode	Quantity
S1	P1	10
S1	P2	20
S2	P2	10
S2	P1	25
S3	P3	15
S3	P1	10

In the above db, none of the column individually would play the role of primary key but the combination of SCode and PCode has a capability to uniquely identify every single row of the table. So when more than one column plays the role of primary key in combination is called Composite Key.

20-07-2017

Entity Relationship Model –

① Design Process
- Requirement Analysis
- Conceptual database design.
- logical database design.

} Relevant to E-R model.

- Schema refinement
- Physical database design
- Application security design.

② Modeling
- An entity is an object that exists and it is distinguishable from other objects.

Example – specific person, company, event, plant.

- Entities have set of properties are called attributes.

Example – people have names and address.

- An relationship is an association among several entities.
- A relationship can be binary (2 entities in relationship) or ternary (3 entities in relationship),

* Degree of Relationship set
  - refers to number of entities.

* Attribute Types :
  1. Simple and Composite attributes.
  2. Single-valued and multi-valued attributes.
  3. Derived attributes e.g. age.

(3) Constraints : Cardinality

* Cardinality
  - One-to-one
  - Many-to-many
  - One-to-many

* Mapping Cardinality [Entity key, Candidate key, Super key]

- Primary Key

  The primary key of an entity is an attribute or a set of attributes that uniquely identifies a specific instance of an entity.

- Candidate Key

  Candidate key of an entity set is a minimal super key.

  Example - customer_id is candidate key of customer.

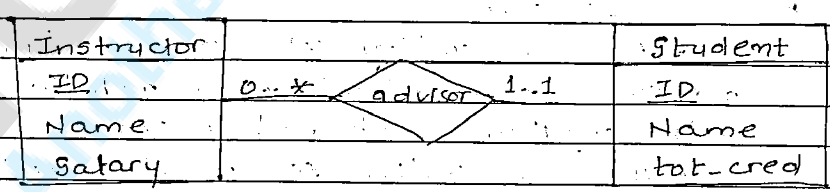
- From several candidate, one of them can be a primary key.

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  Example - (customer_id, acc_number) is super key of depositor.

* Redundant Attributes

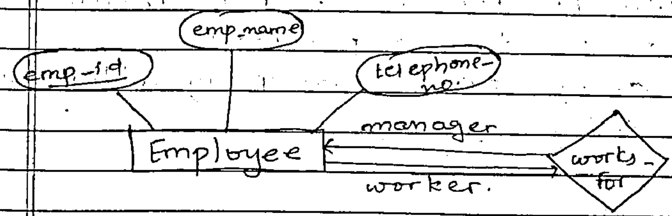
* Participation Constraints
  - Total Participation
  - Partial Participation

Cardinality limits can also express participation constraints.



* Self Relationship -

  Sometimes entities in a entity set may relate to other entities in the same set, thus self relationship.



The labels "manager" and "worker" are called roles the self relationship.

④ Weak Entity Set

Weak entity – not having primary key.
Weak entities are those who depend on other strong entities for its existance.
Ex. section depends on course.
- An entity set may not have sufficient attributes to form a primary key and is termed as a weak entity set.

Strong Entity Set
- An entity set that has a primary key is termed as strong entity set.

* Characteristics –
- A weak entity set may participate as owner in an identifying relationship with another weak entity set.
- A weak entity set must have total participation in identifying relationship set.
- A weak entity set and its owner entity set must participate in many to one relationship set.

* Extended ER Features : Specialization
- Top-down design process –
- designate subgrouping within an entity sets.

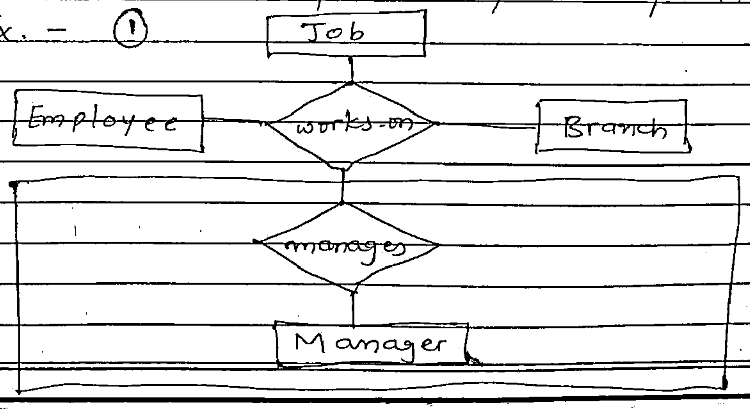
- Design constraints on a specialization/generalization.
- Constraint on which entity can be members of a given lower level entity set.
- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.
  - Design – belong to only one lower-level set.
  - Overlapping – belong to more than one lower-level set.

- Completeness/ Covering Constraint –
  - total – belong to one of the lower-level entity set.
  - partial – need not belong to one of the lower-level entity set.

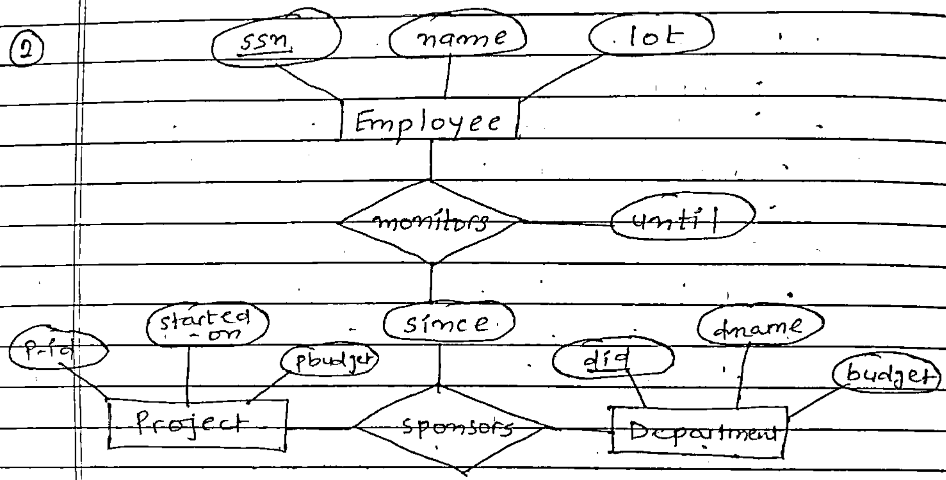
* bottom-up design process –

* Aggregation
- Relationship between 2 relations.
- Can be replaced by ternary relation.

Ex. – ①



(2)



Design Issues:

- Use of entity sets vs. attributes.
- Use of entity sets vs. relationship sets.
- Binary versus n-ary relationship sets.
- Placement of relationship attributes

Relational Data Model –

Data Model is link betⁿ user's view of the world and bits stored in computer. DBMS models the real world objects.

* Introduction to Relational model –

The main construct for representing data in the relational model is a relation. A relation consist of a relational schema = relation name and attribute list.

Ex: student (name, address) or

student (name String, address string)

read as → String is the domain of attribute 'name'

db = collection of relations.

db schema = set of all relational schema.

	name	age
touple (rows)	xyz	20
	abc	15

Relation consists of two –
① Relation schema (column)
② Relation instance (row)

• no. of rows in a relation – cardinality
• no. of columns in a relation – degree

① DDL –
create, alter, drop, rename, truncate

② DML –
update, insert, delete

③ DCL –
grant, revoke

④ TCL –
commit, rollback.

Relational Model Properties

Relational Model Terminology

row = tuple
column = attribute = field
table = relation
type of data = domain

Creating relation in SQL –

Modifying attributes –

Integrity Constraints –
condⁿ specified on a db schema and restricts the data that can be stored in an instance of db.

① Domain Constraints
② Not-null Constraints
③ Entity integrity constraints – primary key.
④ Referential integrity constraints – foreign key.

Candidate Key – It uniquely identifies a tuple.
Primary Key –
Super Key – any key in combination with primary key is known as Super Key.

create table tablename (<column> <domain>
constraint <constraint_name>
primary key (column));

Date: 25/7/17 Page No.:

Foreign Key -

Key Constraints -

A set of field that uniquely identifies a tuple is called a Candidate key.

- Out of the available candidate key a DB can identify primary key.
- A primary key should be unique and NOT NULL

A candidate key should satisfy two properties:

1. Two distinct tuple cannot have identical values in the field of key.
2. No subset of a field is key.

Specifying Key Constraints in SQL -

```
SQL> create table student
    (sid varchar(20),
     name varchar(20),
     login varchar(20),
     age integer,
     marks integer,
     constraint student_key
```

Date: / / Page No.:

Foreign Key -

- Keys are a way to associate tuples in diff. relation.

25-07-2017

- Foreign Key is like a logical pointer.
- If all foreign key constraints are enforced then referential integrity is achieved.

Add Foreign Key -

```
create table enrol (sid char(20)
                    cid char(20)
                    grade char(10)
                    PRIMARY KEY (sid, cid)
                    FOREIGN KEY (sid)
                    REFERENCES student
                    ON DELETE CASCADE
                    ON UPDATE NO ACTION);
```

1. No Action
2. Cascading - deletes from both table
3. Default - sets default value
4. NULL - sets NULL value

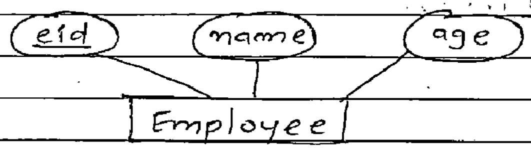
Adding Unique Field

Adding check constraint

ER to Rational –

- Entity – relation
- Attributes – attributes

Entity Sets to Tables –



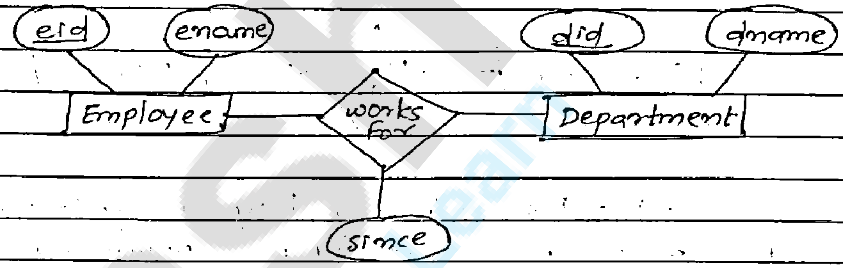
Primary Key

Eid	Name	Age

← Employee

Mapping attributes by composite

Relationship sets to Tables (without constraints) –



Employee

eid	ename

Department

did	dname

foreign key for emp. ← works-for → foreign key for dept.

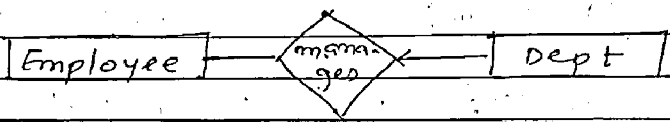
eid	did	since

→ primary key.

Ternary Relationship Set –

Relationship sets to Tables (with key constraints) –

The table having key constraints then that table's primary key is considered as new table's primary key and other is just foreign key.



Dept's primary key is primary key for manages and emp's primary key is just foreign key for manages.

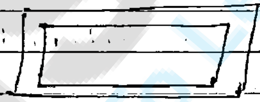
Relationship sets to tables (with participation constraints) —

For relationship, use NOT NULL constraint for showing participation constraint.

Weak Entity —

It does not have a primary key.

It is shown as —



ISA Relationship —

Aggregation —

Relationship between two relationships.

Date: 28/7/17 Page No.:

3. Normalization.

Functional Dependency –

F $\alpha \rightarrow \beta$

Attribute α determines attribute β in a relation.

1NF
2NF
3NF
BCNF
} depends on F.D. for removing redundancy.

4NF
5NF
} does not depend on F.D.

α or β can be group of attributes.

If $\alpha = a$ then we can find corresponding β value but value of α cannot be same for two different β values.

R

α	β
t_1	t_2
$t_1[\alpha]$	$= t_2[\alpha]$
$\Rightarrow t_1[\beta]$	$= t_2[\beta]$

	α	β
if this is true then $\alpha \rightarrow \beta$ cannot be true.	a	1
	a	2
	b	1
	d	4

To $\alpha \rightarrow \beta$ be true values of same α cannot be same for two different values of β.

Date: / / Page No.:

If $\alpha \rightarrow \beta$ is true,

α	β
a	1
b	1
d	4

To find closure of Functional Dependency –

~~R(A B C D E F G)~~

F.D. 2 types –

① Trivial $AB \rightarrow B$ (~~It gives the known value~~)
② Non-Trivial $AB \rightarrow ABC$ (It gives new value)

① → Given a value of attributes of A and B and you uniquely find the known attribute B.

② → Given a value of attributes A and B and you find out new attribute C.

Q. R (A B C D E F G) — Relation; Attributes

$A \rightarrow B$

$BC \rightarrow DE$

$AEG \rightarrow G$

Find closure of A C

$(AC)^+ = AC$
$= ABC$
$= ABCDE$

Q. R(ABCDE) closure means what is given.

A → BC
CD → E
B → D
E → A

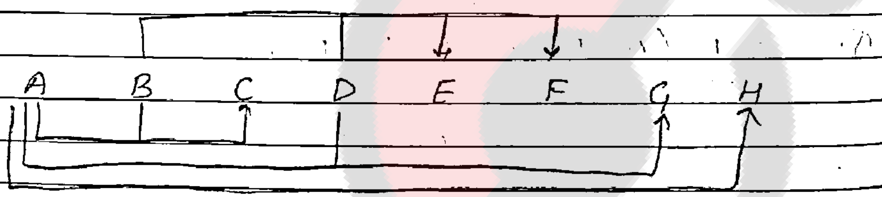
find the closure of B.

→ $(B)^+$ = BD

Finding the Candidate Key –

Q. A relation has foll. attributes
A B C D E F G H
with Functional dependencies
AB → C
BD → EF
AD → G
A → H
Identify the candidate key for the relation from the given F.D.

→



A, B, D doesn't have any inward arrow so given any attribute in relation we cannot identify them.

So, A,B,D can be candidate key.
or
any other ~~ab~~ attribute ~~with~~ in combination with them can be candidate key.

(ABD) ← part of candidate key.

$(ABD)^+$ = ABD
= ABCD
= ABCDEF
= ABCDEFG
= ABCDEFGH
= R

As we can identify all the attributes in a relation R using ABD
So ABD is a candidate key.

Any combination with ABD gives super key.

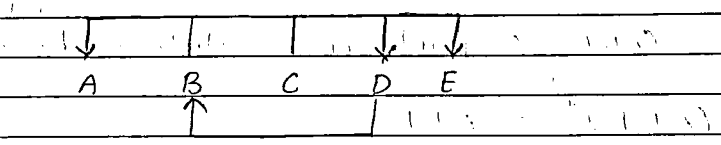
– Taking a closure of candidate ~~sho~~ key should give complete relation.

Q. R (A B C D E)

F.D —

BC → AD E

D → B



C → no incoming arrow

so take $(C)^+$

$(C)^+ = C \neq R$

∴ So C is not, so try possible combⁿ.

$(AC)^+ = AC$

$(BC)^+ = BCADE$

$(CD)^+ = CDBAE$

$(CE)^+ = CE$

So ~~ke~~ candidate key can be BC or CD.

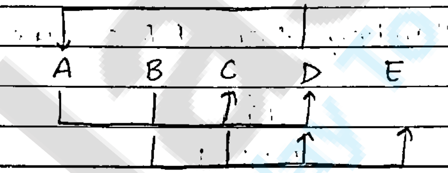
Q. R (A B C D E)

F.D —

AB → CD

D → A

BC → DE



B → no inside arrow

$(B)^+ = B \neq R$

$(AB)^+ = ABCDE = R$

$(BC)^+ = \cancel{BCA}\ ABCDEA = R$

$(BD)^+ = BDACD = R$

$(BE)^+ = BE$

So candidate key can be AB or BC or BD

Canonical Form of F.D. means minimal form of F.D. i.e. no redundancy in F.D.

Equivalence of F.D. —

Q. Whether the following two F.D.s are equivalent —

F	G
$A \to C$	$A \to CD$
$AC \to D$	$E \to AH$
$E \to AD$	
$E \to H$	

* To check equivalence there are 4 ways —

1. $F \subseteq G$
2. $G \subseteq F$
3. $F = G$
4. $F \neq G$

→ Find closure of F using G.

depending on G	depending on F
$(A)^+ \to ACD$	$(A)^+ \to ACD$
$(AC)^+ \to ACD$	$(E)^+ \to EADHC$
$(E)^+ \to EAHCD$	All G found by F so
All F found by G so	
$F \subseteq G$	$G \subseteq F$

$F = G$.

Canonical Form —

Q. $R(WXYZ)$

$X \to W$

$WZ \to XY$

$Y \to WXZ$

→ Decompose only β

✓ $X \to W$ ①

$WZ \to X$ ② *

✓ $WZ \to Y$ ③

$Y \to W$ ④

✓ $Y \to X$ ⑤

✓ $Y \to Z$ ⑥

with all $(X)^+ \to XW$ → depending on all F.D.

without ① $(X)^+ \to X$ → eliminating 'X' F.D. so it is necessary in the relation as it is not redundant.

with all $(WZ)^+ \to WZXY$

without ② $(WZ)^+ \to WZYX$ → it is redundant eliminate ② as redundant. For further do not use ②.

with all $(WZ)^+ \to WZYX$

without ③ $(WZ)^+ \to WZ$ → not redundant.

with all $(Y)^+ = YWxZ$
without (4) $(Y)^+ = YxZW$ → redundant

with all $(Y)^+ = YWxZ$
without (5) $(Y)^+ = YZ$ → not redundant

with all $(Y)^+ = YWxZ$
without (6) $(Y)^+ = YxW$ → not redundant.

$x \to W$	$x \to W$
$WZ \to Y$	$WZ \to Y$
$Y \to x$	$Y \to xZ$
$Y \to Z$	→ Canonical form as there is no redundancy.

- There can be multiple canonical form for given functional dependency.

1NF –

The intersection of a row and a column should be atomic.

- If table is in BCNF then the table is in 3NF, 2NF and 1NF and so on.

2NF –

There should be no partial dependency in a relation.

R (A B C D)

$\alpha \to \beta$

prime/non-prime → non-prime

AB → primary key.

AB → C
B → D

It is not 2-NF.

As B is partial key of primary key it is not valid.

Identifying other attri part of primary key is partial dependency.

- 2NF comes into picture only when primary key is composite.
- In normalization, no. of tables increases.

To remove partial dependency

R (A B C D)

R → R_1 (A B C), R_2 (B D)

Now these two tables are in 2NF.

Date: 1/8/17

3NF-

Transitive Dependency-
let A, B, C are 3 attributes of relation R
If A → B and B → C, then A transitively determines C through B, provided B and C do not determine A.

A non-prime attr. identifies non-prime then this is transitive dependency.

If a part of prime attr. identifies non-prime then this is partial dependency.

$\alpha \longrightarrow \beta$

then α should be prime and β should be non-prime.

R (A B C D)

This is transitive dependency.

If C is NULL then we cannot find D.

∴ there should not be transitive dependency.

R (A B C D)

R → R_1 (A B C)
R → R_2 (C D)

So, now transitive dependency is removed.

2NF $\xrightarrow[\text{Transitive Dependencies}]{\text{Remove}}$ 3NF

Prime attr. → which are used to find other attr.

BCNF -

$\alpha \longrightarrow \beta$
prime/non-prime → prime

A prime attribute is being identified by other. This cannot be done. This wasn't taken care in any of the above normal form.

A prime attribute must not be identified by other attribute. So this is to be removed in BCNF

$\alpha \longrightarrow \beta$

super key

If α is super key always then β cannot be prime key.

Super Key → comb^n of primary & candidate

R(A B C D)

This ~~com~~ should not be done.

Q. R (A B C D E F G H I J)

Decompose to BCNF.

① Identify Candidate Key – AB

which does not have incoming edges.

$(AB)^+$ = ABCDEFGHIJ

Note Functional Dependencies: –

AB → C
B → F
A → DE
F → GH
D → IJ

It is in 1NF.

For 2NF no partial dependency should be there.

AB is candidate key.

A is also partial and ~~so~~ B is also.
So decompose R into 3.

R_1 (ABC)

R_2 (ADEIJ)

R_3 (BFGH)

It is in 2NF as there is no partial dependency.

R_1 (A B C).

R_2 (A D E I J)

R_3 (B F G H)

As there is transitive dependency as non-prime is identifying non-prime so R_2 and R_3 are not in 3NF.

R_2 <
R_{21} (ADE)
R_{22} (DIJ)

R_3 <
R_{31} (BF)
R_{32} (FGH)

Now there is no transitive dependency.
So these relations are in 3NF now.

No F.D. is identifying prime
So this is in BCNF.

Date: 3/8/17 Page No.:

Q. R(ABCD)

$AB \rightarrow C$
$B \rightarrow D$
$BC \rightarrow A$

Identify the normal form of this relation.

$\longrightarrow$

A B C D

$(B)^+ = BD$

$(BA)^+ = ABCD$
$(BC)^+ = BCAD$
$(BD)^+ = BD$

$\therefore$ AB, BC are candidate keys.

A, B, C are prime attributes.

① $AB \rightarrow C$
C is not the subset of AB.
So, it is non-trival.

F.D. is in BCNF.

② $B \rightarrow D$ $D \not\subset B$, so non-trival.
Part of primary key. So it is partial Dependent. and so no 2NF.
It is only in 1NF.

③ $BC \rightarrow A$

~~BC ⊄ A~~ $A \not\subseteq BC$ so non-trivial.

Consider BC as candidate key.

∴ This F.D. is in BCNF.

	1NF	2NF	3NF	BCNF
$AB \rightarrow C$				✓
$B \rightarrow D$	✓			
$BC \rightarrow A$				✓

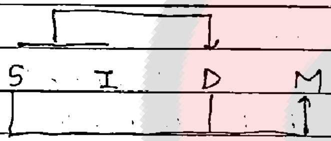
Q. $R(S\ I\ D\ M)$

$SI \rightarrow D$

$SD \rightarrow M$

Identify the normal form of this F.D.

→



$(SI)^+ = SIDM$

∴ SI is candidate key.

① $SI \rightarrow D$

As $D \not\subseteq SI$

∴ It is non-trivial.

It is in BCNF.

② $SD \rightarrow M$

As $M \not\subseteq SD$

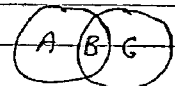
∴ It is non-trivial.

It is in 1NF.

- If a relationship is in 3NF and has only one candidate key then relationship is in BCNF.

- If in 3NF there are multiple candidate keys which are composite and are not disjoint then it is not in BCNF.

composite $\left\{ \begin{matrix} AB \\ BC \end{matrix} \right\}$ multiple &



non-disjoint

so not BCNF.

Date: 4/8/17

Desirable Properties of Decomposition –

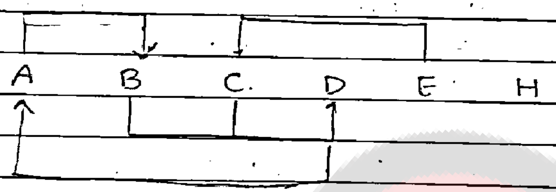
- Attribute preservation
- Loossless – Join decomposing
- Dependency Preservation
- Lack of redundancy

Q. Identify the candidate key –

R (A B C D E ~~F~~ H)

A → B
BC → D
E → C
D → A

→



$(EH)^+ = EHC$

$(AEH)^+ = AEHBCD = R$
$(BEH)^+ = BEHCDA = R$
$(CEH)^+ = CEH$
$(DEH)^+ = DEHCAB = R$

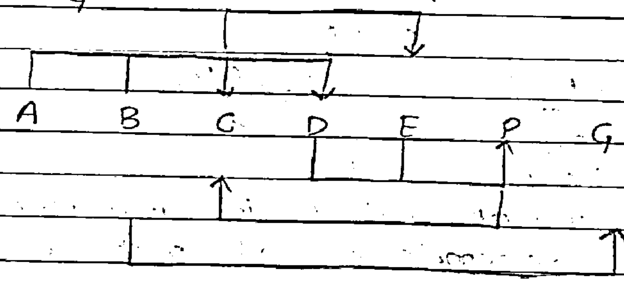
① A → B
Not in BCNF, 3NF, 2NF

② BC → D
Only in 1NF

③ E → C
Only in 1NF

Q. Find Candidate Key –

R (A B C D E P G)
AB → CD
DE → P
C → ~~E~~
P → C
B → G



$(AB)^+ = ABCDEPG = R$

Desirable Properties of Decomposition –

(1) Attribute ~~Decomposition~~ Preservation –
While decomposing no attributes must be lost, we should preserve all of them.

(2) lossless - Join decomposing –

R

R_1 R_2

$R_1 \bowtie R_2 = R$.

(3) Dependency Preservation. –

(ABCD)
$AB \rightarrow C$
$C \rightarrow D$

$R_1(ABC)$ $R_2(CD)$

F.D must be preserved

We cannot put C in any one of them then dependency will not be preserved.

(4) No lack of redundancy.

$\bowtie$ = Natural Join.

R

R_1 R_2

$R' = R_1 \bowtie R_2$

If $R' = R$ then it's lossless join
If $R' \neq R$ then it's lossy join.

- lossless is exact what we need to perform.
- In lossy there may be more information then the original table, this should not happen.

Decomposition –
(1) Lossless
(2) Lossy.

Date: 7/8/17 Page No.:

Inference Rule of Functional Dependency

A F.D. $x \to y$ is infered from a set of dependencies. The closure F^* of F is the set of all F.D. that can be infered from F.

- The set of ~~Inb~~ inference rule can be used to infer to new dependency from the set of given dependency.

Rules –

① IR1 (reflexive rule)
if $x \supseteq y$ then $x \to y$

② IR2 (argumentation rule)
$\{x \to y\} \Rightarrow xz \to yz$

③ IR3 (transitive rule)
$\{x \to y, y \to z\}$ $x \to z$

④ IR4 (decomposition rule)
$\{x \to yz\} \Rightarrow x \to y$

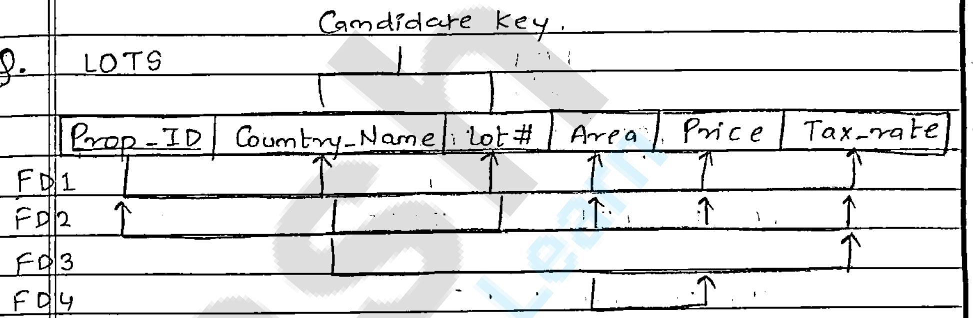
⑤ IR5 (union)
$\{x \to y, x \to z\} \Rightarrow x \to yz$

⑥ IR6 (psuedo transitive)
$\{x \to y, wy \to z\} \Rightarrow wx \to z$

* – Amstrong's inference rule

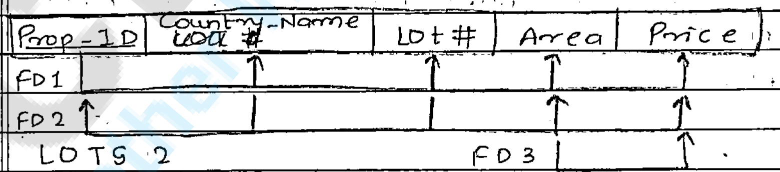
Date: / / Page No.:

Q. LOTS

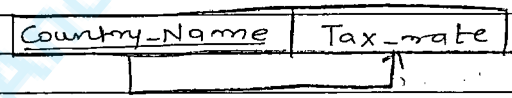


→ Step 1: Reduce the table to 2NF.

LOTS 1

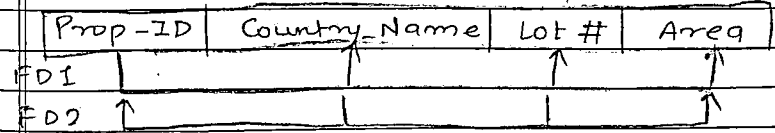


– LOTS 2

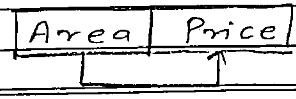


Step 2: Reduce the table to 3NF.

$LOTS1_1$



– $LOTS1_2$



LOTS — 1NF

LOTS1   LOTS2 — 2NF

$LOTS1_1$ $LOTS1_2$ LOTS2 — 3NF

It is in BCNF.

Now adding one more F.D.

$LOTS1_1$ – Now this is not in BCNF.

Prop_ID	Country_Name	LOT#	Area

FD1

FD2

FD5

$LOTS1_2$

Area	Price

FD3

LOTS 2

Country_Name	Tax_rate

FD4

$LOTS1_{1A}$

Prop_ID	LOTS#	Area

$LOTS1_{1B}$

Area	Country_Name

LOTS — 1NF

LOTS1   LOTS2 — 2NF

$LOTS1_1$ $LOTS1_2$ LOTS2 — 3NF

$LOTS1_{1A}$ $LOTS1_{1B}$ — BCNF

Date: 8/4/17

Q. Given are 2 FDS for R

R (A, B, C, D, E)

F1 – $A \to B$
$AB \to C$
$D \to AC$
$D \to E$

F2 → $A \to BC$
$D \to AE$

Find closure of F1 using F2 and vice versa.

$(A)^+ = ABC$
$(AB)^+ = ABC$
$(D)^+ = DAEBC$

$(A)^+ = ABC$
$(D)^+ = DACEB$

Q.1. Consider R(A, B, C, D, E, F, G, H, I, J) and set of FDs is

F = { $(A, B) \to C$
$A \to D, E$
$B \to F$
$F \to G, H$
$D \to I, J$

What is the key of R, decompose the relation in 2NF and 3NF.

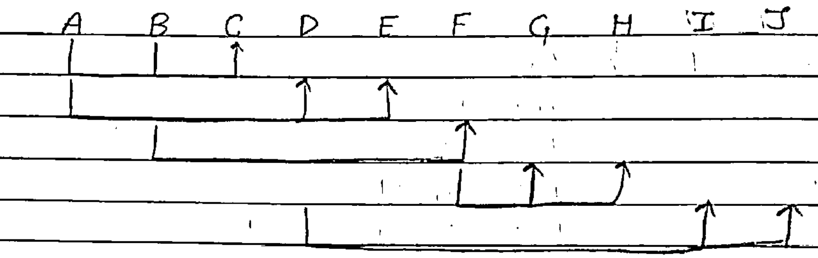
Q.2. F.D. of R(A, B, C, D, E, F) is –
$AB \to C$
$C \to A$
$BC \to D$
$ACD \to B$
$BE \to C$
$EC \to FA$
$CF \to BD$
$D \to E$

Find the minimum cover of set of F.D

Q.3. R(A, B, C, D, E, H) FDs –
$A \to BC$
$CD \to E$
$E \to C$
$D \to AEH$
$ABH \to BD$
$DH \to BC$

Find the key of the relation.

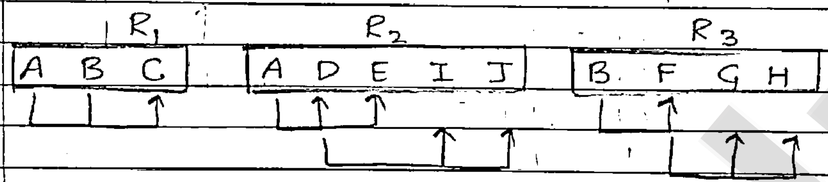
1.



$(AB)^+ = ABCDEFGHIJ$

① Reduce dependency to 2NF

R_1: ~~A B C F G H I J~~ R_2: ~~A D E~~

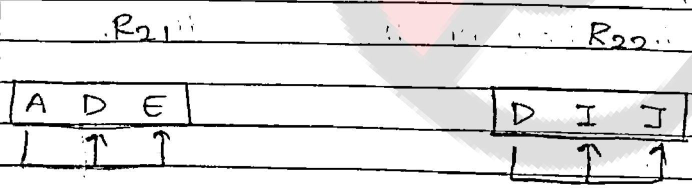


Now this is in 2NF.

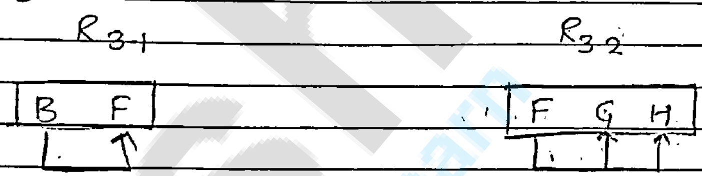
② Reduce dependency to 3NF.

R_1 is in 3NF.

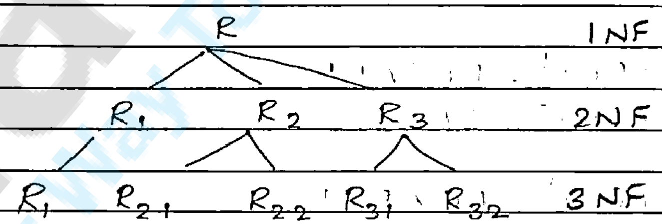
R_2 —



R_3 —



Now it is 3NF.



2.

~~$(AB)^+ = ABCDE$~~

① $AB \rightarrow C$ ✓
② $C \rightarrow A$ ✓
③ $BC \rightarrow D$ ✓
④ $ACD \rightarrow B$
⑤ $BE \rightarrow C$ ✓
⑥ $EC \rightarrow F$ ✓
⑦ $EC \rightarrow A$
⑧ $CF \rightarrow B$ ✓
⑨ $CF \rightarrow D$
⑩ $D \rightarrow E$

$(AB)^+ = ABCDEF$

without ① $(AB)^+ = AB$

$(C)^+ = CA$

without (2) $(C)^+ = C$

$(BC)^+ = BCDEFA$

without (3) $(BC)^+ = BCA$

$(ACD)^+ = ACDEFB$

without (4) $(ACD)^+ = ACDEFB$

$(BE)^+ = BECADF$

without (5) $(BE)^+ = BE$

$(EC)^+ = ECAFBD$

without (6) $(EC)^+ = ECA$

$(EC)^+ = ECAFBD$

without (7) $(EC)^+ = ECFBDA$

$(CF)^+ = CFABDE$

without (8) $(CF)^+ = CFA\cancel{B}DE$

$(CF)^+ = CFABDE$

without (9) $(CF)^+ = CFABDE$

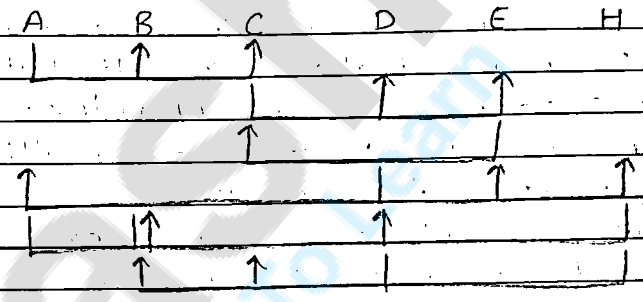
$(D)^+ = DE$

without (10) $(D)^+ = D$

$AB \rightarrow C$ $CF \rightarrow B$
$C \rightarrow A$ $D \rightarrow E$
$BC \rightarrow D$
$BE \rightarrow C$
$EC \rightarrow F$

Hence, this is canonical form.

3.



No attr. without arrow head. So do with all incoming wala comb^n as follows :-

$(AB)^+ = ABC$
$(AC)^+ = ACB$
$(AD)^+ = ADEHCB = R$
$(AE)^+ = AEBC$
$(AH)^+ = AHBCDE = R$

Date: 10/8/17

(MVD) Multi-Valued Dependency —

These are the consequences of 1NF which disallows an attribute in a tuple to have a set of values

- ~~In~~ If we have 2 or more multi-value independent attribute in a same relation we get into a problem of having to repeat every value of one of the attribute with every value of the other attribute,

* To keep a relation in a consistent state and maintain the independency among the attributes

4NF —

Table is in BCNF and has no multi-valued dependency. ($x \to\to y$)
x has multiple values for Y.

$x \to\to Y | Z$

① x has multiple values for Y and Z.
and
② Y and Z has no dependency on one another.

If these two are satisfied then table is not in 4NF.

then it is decomposed as

$x \to\to Y$ $\quad\quad$ $x \to\to Z$.

Trival —
$A \to A$

Non-trival —
$A \to AB$

Completely Non-Trival —
$A \to B$

4NF Points to Note :

- If $(t1, x1)$; $(t2, x2)$ both appear, then $(t1, x2)$, $(t2, x1)$ will also appear.
- Teachers and Texts are completely independent of one another.
- CTX ~~no~~ has no FDs at all
- But still there is a need to normalize CTX.

A multi-valued dependency $(X \to\to Y)$ in a relation R: if two tuples t1 and t2 exists such that ~~the~~
$t1.x == t2.x$
then there exists two tuples t3 and t4 such that
$t3.x == t4.x == t1.x == t2.x$

t3.y == t1.y
t4.y == t2.y

Q. Emp Table -

e_name	p_name	d_name
Smith	x	John
Smith	y	Anna
Smith	x	Anna
Smith	y	John

In this, there is multivalue dependency and it is decomposed as -

Emp1		Emp2	
e_name	p_name	e_name	d_name
Smith	x	Smith	John
Smith	y	Smith	Anna

- F.D.s are not considereds in 4NF.

5NF -

Join Dependency and the 5NF -

- 5NF deals with join dependencies which is a generalization of MVD.
- The aim of 5NF is to have tables that cannot be decomposed further.
- A table in 5NF cannot be constructed from several smaller tables.

$$R \rightarrow R_1,\ R_2,\ R_3$$

$$R = \Pi(R_1) \bowtie \Pi(R_2) \bowtie \Pi(R_3)$$

- 5NF is not practically possible.

A Join Dependency (JD) specified on a relation R is stated as for every legal state a non-additive join decomposition $R_1, R_2 \ldots R_n$ such that,

$$\Pi(R_1)\ \Pi(R_2) \ldots \Pi(R_n) = R$$

The 5NF is also known as Project Join Normal Form (PJNF).

Date: 11/8/17

Q. A supply table –

R

S_name	Part_name	Project_name
Smith	Bolt	Proj X
Smith	Nut	Proj Y
Alice	Bolt	Proj Y
Nelsom	Nut	Proj Z
Alice	Nail	Proj X
Alice	Bolt	Proj X
Smith	Bolt	Proj Y

R → R_1, R_2, R_3

R_1

S_Name	Part_Name
Smith	Bolt
Smith	Nut
Alice ~~Smith~~	Bolt
~~Alice~~	~~Nail~~
Nelsom	Nut
Alice	Nail

R_2

S_Name	Proj_Name
Smith	Proj X
Smith	Proj Y
Alice	Proj Y
Nelsom	Proj Z
Alice	Proj X

Date: / /

R_3

Part_Name	Proj_Name
Bolt	Proj X
Nut	Proj Y
Bolt	Proj Y
Nut	Proj Z
Nail	Proj X

No further decomposition possible and we get R after joining R_1, R_2, R_3. So it is in 5NF.

Q. A Emp table –

E_Name	P_Name	D_Name
Smith	x	John
Smith	y	Anna
Smith	x	Anna
Smith	y	John
Alice	W	Jim
Alice	x	Jim
Alice	y	Jim
Alice	z	Jim
Alice	W	Eve
Alice	x	Eve
Alice	y	Eve
Alice	z	Eve
Alice	W	Bob
Alice	x	Bob
Alice	y	Bob
Alice	z	Bob

Emp1		Emp2	
E_Name	P_Name	E_Name	D_Name
Smith	x	Smith	John
Smith	y	Smith	Anna
Alice	w	Alice	Jim
Alice	x	Alice	Eve
Alice	y	Alice	Bob
Alice	z		

Q. Car_Sale Table –
Attributes: Car, Date, Salesman, Commission, Discount_Amount.

Assume that car is sold by the multiple salesmen and hence car, Salesman is primary key. The other FDS are –

Date → Discount_Amount

Salesman → Commission

Based on the primary key is this relation in 1NF, 2NF or 3NF. Justify the answer.
How could you successfully normalize it completely?

→

R → Car Date Salesman Commission Discount_Amount

It is in 1NF

It is not in 2NF as it is partial dependent.

R → R_1, R_2

R_1 → Car Date Salesman Discount_Amount

R_2 → Salesman Commission ~~Discount Amount~~

It is in 2NF now as there is no partial dependency.

Now, ~~there~~ R_1 is not in 3NF as non-prime 'Date' is identifying non-prime 'Discount Amount'.

R_2 is in 3NF.

	R		1NF
R_1		R_2	2NF
R_{1A}	R_{1B}	R_2	3NF

R_{1A} – Car Date Salesman

R_{1B} – Date Discount_Amount

Now it is in 3NF.