Normalization

Normalization of a relation schema refers to the process of

1. Identifying those data dependencies in the schema, which would cause anomalies (situation or thing which will defer from normal or expected thing) during insert, update and deletion of data?
2. Decomposing the schema into a set of sub schemas, on the basis of dependencies causing anomalies.

Functional Dependency:

Let there be a relation schema R_1 comprising of attributes α ,β , γ i.e α ⊆ R , β ⊆ R , γ ⊆ R . A Functional dependency α →β (read as "α determines β") is said to be holding on the schema R , if for every legal relation γ(R) and for every tuple pair {t_1,t_2}€ r if $t_1[\alpha] = t_2[\alpha]$ then it must satisfy $t_1[\beta]=t_2[\beta]$. This means that if any two tuples relation r(R) agrees (equal) on the values of α, then the two tuples must agree on the values of β.

α, β , γ are the sets of attributes .

Ex :

α={Rollno , name}

β = {sname, DOB , Branch}

α the left side of FD α -> β is called "determinant" and the right side of FD α->β is called "Dependent".

Ex: Registration_NO -> Name

↓ ↓

Determinant dependent

This example is given by considering a relation schema

Student(Roll_NO,Regestration_NO,Branch,Section,Name,father_name,address,DOB).

KEY:

It is the set of attributes that uniquely identify an entire tuple.

FUNCTIONAL DEPENDENCY:

It allows us to express constraints that uniquely identify the values of certain attributes.

Most generally all keys are Functional Dependencies.

LEGAL RELATION:

Relation which satisfies all the constraints is called legal relation.

A set of FD's F is said to be holding on a schema R if all FD's in F are satisfied by every legal relation r(R).

NORMAL FORMS

1st NORMAL FORM:

A schema R is said to be in first Normal Form if all its attributes are single valued or atomic.

Normalized Table(emp)

Eid	Ename	Salary	Tel no
001	Ajay	20000	9801222777, 2449227, 9422230230
002	Vijay	5000	Null
003	Ram	10000	980345567

The above table is not in the first normal form because Telno column is containing multiple values so it is not a single valued attribute it is multi valued attribute.

1 st Normal Form Table(emp)

Eid	Ename	Salary	Tel no
001	Ajay	20000	9801222777
001	Ajay	20000	2449227
001	Ajay	20000	9422230230
002	Vijay	5000	Null
003	Ram	10000	980345567

The above table is having only atomic values so it is first normal form

TRIVIAL FUNCTIONAL DEPENDENCY:-

A FD ($\alpha \rightarrow \beta$) is said to be trivial, if($\beta \subseteq \alpha$) such a FD would be satisfied by any relation r(R). if two tuples of a relation agree on the values of α ,then the tuples definitely will agree on the values of B.

EX:-

{ Roll no, Name}→Name

EXTRANOUS ATTRIBUTE:-

An attribute A(A ⊆α) is said to be extraneous if FD(α -A)→b also holds on R. This means that attribute.

A is not required in α to determine the values of β. the subset (α-A) is sufficient to determine the values of β.

LEFT-IRREDUCIBLE FD:-

A FD α→β, holding on a schema R, is said to be left irreducible if there exists no proper subset of α, which can determine the value of β.

SUPER KEY:-

It is said to be attribute on the left side of FD to determine the values of attributes.

In student schema the values of {Roll No} is distinct and also the values of registration_ No is also distinct.

{Roll No, Class} &{ registration_ No, Name},{name, fathers_ name ,Address, DOB} will from super keys for student.

There may be any no. of super keys for a relation.

The subset super key may also from super key.

CANDIDATE KEY:-

A super key may contain some extraneous attributes. Suppose K is super key of R and E is the set of total extraneous attributes in K then (K-E) will from a minimal super key of R.

The minimal super key is called a candidate key of R.

→Alternatively it can be stated that if K→R forms a left_ irreducible FD holding on R than K is said to be candidate key of R.

EX:-

{Roll No}, {registration no} , {name, fathers_ name ,Address, DOB} will form candidate keys of student

There may be any no. of candidate keys for a relation.

PRIMARY KEY:-

A Relation schema R may have more than one candidate keys. One of the candidate key is choose as primary to identity the tuples uniquely in a relation. Rest of the candidate keys are called **secondary keys**

Ex: if {Roll No} is chosen as primary key then {registration no} , {name, fathers_ name ,Address, DOB} are secondary keys.

Logically Implied FDs:-

Suppose F is the set of FDs holding on a Relational Schema R. There may be some FDs that can be logically inferred (implied) from F. These implied FDs also hold on every legal relation r(R) such type FDs are called logically implied FD's by F.

Armstrong's Axioms (rules):-

These rules are used to find the logically implied FDs

1. **Reflexivity Rule**:- If α is a set of attributes and $\beta \subseteq \alpha$ then $\alpha \rightarrow \beta$ holds on 'R'.

Proof:-

Consider a relation γ(R) and a tuple pair $\{t_1,t_2\} \in r$ such that

$t_1[\alpha]=t_2[\alpha]$ –---- (1)

since$\beta \subseteq \alpha$

then t_1 and t_2 will also satisfy

$t_1[\beta]=t_2[\beta]$ – -------(2)

from (1) & (2)

$\alpha \rightarrow \beta$holds on R

2. **Augmentation rule:-** If $\alpha \rightarrow \beta$ holds on R, then $\alpha\gamma \rightarrow \beta$ also holds on R.

Proof:- This rule will be proved by Contradiction.

Suppose $\alpha \rightarrow \beta$ holds on R

Consider a relation γ(R) and tuple pair $\{t_1,t_2\} \in r$

such that

$t_1[\alpha]=t_2[\alpha]$ –------ (1)

since$\alpha \rightarrow \beta$ holds on R, then t_1,t_2 will also satisfy

$t_1[\beta]=t_2[\beta]$ --------– (2)

We make an assumption that $\alpha\gamma \rightarrow \beta\gamma$ does not hold on it.

So $t_1[\alpha\gamma]=t_2[\alpha\gamma]$ –------- (3)

Then

$t_1[\beta\gamma]$!=$t_2[\beta\gamma]$ –----- (4)

from (1) & (3)

$t_1[\gamma]=t_2[\gamma]$ –---------- (5)

from (2) &(5)

$t_1[\beta\gamma]=t_2[\beta\gamma]$ –-------- (6)

(4) and (6) contradicts each other so our assumption is wrong

So, if $\alpha \rightarrow \beta$ holds on R then $\alpha\gamma \rightarrow \beta\gamma$ also holds on R.

3. Transitivity Rule:- If $\alpha \rightarrow \beta \ and \ \beta \rightarrow \gamma$ holds on R then $\alpha \rightarrow \gamma$ holds on R.

Proof:- since $\alpha \rightarrow \beta$ holds on γ(R) & for a tuple pair $\{t_1,t_2\}\in r$ then

$t_1[\alpha]=t_2[\alpha]$ –---------- (1)

$t_1[\beta]=t_2\beta]$ ----------– (2)

since$\beta \rightarrow \gamma$ holds on R

then $t_1[\beta]=t_2[\beta]$ ---------– (3)

$t_1[\gamma]=t_2[\gamma]$ –------(4)

from (1) & (4)

α -> γ holds on R.

<u>Additional Rules:</u>

1. **Union Rule :** If α ->β and α ->γ holds , then α ->βγ also holds on R

Proof:

If α ->β ,α ->γ holds on R

By using augmentation rule

α ->αβ ------- (1) augmented by α

αβ -> βγ ----------(2) augmented by β

from (1) & (2) by transitivity rule

α -> βγ

2. **Decomposition Rule:**If α -> βγ holds on R , then α ->β and α ->γ also holds on R.
   Suppose α -> βγ ---------- (1) holds on R then by reflexivity rule
   βγ -> β ---------- (2) by reflexivity
   From (1) and (2) by transitivity
   Then α ->β holds on R
   Similarly
   βγ -> γ -------- (3) by reflexivity
   from (1) and (3) by transitivity
   Then α -> γ holds on R.
3. **Pseudo – Transitivity rule:** If α ->β and βγ ->δ holds on R then αγ -> δ also holds on R.

Proof: Given

α ->β ---------(1)

βγ -> δ ----------(2)

Applying augmentation rule to (1) , then

αγ -> βγ ----------(3)

Applying transitivity rule to (1) &(3)

We get αγ -> δ

CLOSURE OF FD SET:-

Suppose F is said to be the set of FD's that holds on a schema R, the closure of F. denoted by $F^{+.}$ is the complete set of FDs that includes all the given FDs and FDs that are logically implied by F. any logical relation in r(R) that satisfy F will also satisfy $F^{+.}$

LOSS-LESS- DECOMPOSITION OF A RELATION SCHEMA R

Decomposition of a relation r(R) into $r_1(R_1)$ and $r_2(R_2)$ such that $R_1 \cup R_2 = R$ is said to be a loss-less-join decomposition ,if it satisfy $r_1 * r_2 = r$ i.e natural join to the r1 and r2 should generate r with no tuples eliminated and no tuples added. Such decomposition is also called a Non-additive Decomposition.

EX:-

Consider the following relation r on schema R(A,B,C) if decompose into r_1 and r_2

Case 1

r(R) =

A	B	C
A_1	B_1	C_1
A_2	B_2	C_1
A_1	B_1	C_2
A_3	B_2	C_3
A_1	B_1	C_3
A_2	B_2	C_4

$r_1(R_1)$ =

A	B
A_1	B_2
A_2	B_2
A_3	B_2

$r_2(R_2)$=

A	C
A_1	C_1
A_2	C_1
A_1	C_2
A_3	C_3
A_1	C_3
A_2	C_4

$r_1 \times r_2$ =

A	B	A	C
A_1	B_1	A_1	C_1
A_1	B_1	A_2	C_1
A_1	B_1	A_1	C_2
A_1	B_1	A_3	C_3
A_1	B_1	A_1	C_3
A_1	B_1	A_2	C_4
A_2	B_2	A_1	C_1
A_2	B_2	A_2	C_1
A_2	B_2	A_1	C_2
A_2	B_2	A_3	C_3
A_2	B_2	A_1	C_3
A_2	B_2	A_2	C_4
A_3	B_2	A_1	C_1
A_3	B_2	A_2	C_1

A_3	B_2	A_1	C_2
A_3	B_2	A_3	C_3
A_3	B_2	A_1	C_3
A_3	B_2	A_2	C_4

So $r_1 * r_2$

A	B	C
A_1	B_1	**C_1**
A_1	B_1	C_2
A_1	B_1	C_3
A_2	B_2	C_1
A_2	B_2	C_4
A_3	B_2	C_3

It is loss less join decomposition since $r_1 * r_2 = r$.

Case 2

r(R)=

A	**B**	**C**
A_1	B_1	C_1
A_2	B_2	C_1
A_1	B_2	C_2
A_3	B_2	C_3
A_1	B_1	C_3
A_2	B_1	C_4

$r_1(R_1)$

A	**B**
A_1	B_1
A_2	B_2
A_1	B_2
A_3	B_2
A_2	B_1

$r_2(R_2)$

A	**C**
A_1	C_1
A_2	C_1
A_1	C_2
A_3	C_3
A_1	C_3
A_2	C_4

$r_1 \times r_2$

A	**B**	**A**	**C**
A_1	B_1	A_1	C_1
A_1	B_1	A_2	C_1
A_1	B_1	A_1	C_2
A_1	B_1	A_3	C_3
A_1	B_1	A_1	C_3
A_1	B_1	A_2	C_4
A_2	B_2	A_1	C_1
A_2	B_2	A_2	C_1
A_2	B_2	A_1	C_2
A_2	B_2	A_3	C_3
A_2	B_2	A_1	C_3
A_2	B_2	A_2	C_4
A_1	B_2	A_1	C_1
A_1	B_2	A_2	C_1
A_1	B_2	A_1	C_2
A_1	B_2	A_3	C_3
A_1	B_2	A_1	C_3
A_1	B_2	A_2	C_4
A_3	B_2	A_1	C_1

A_3	B_2	A_2	C_1
A_3	B_2	A_1	C_2
A_3	B_2	A_3	C_3
A_3	B_2	A_1	C_3
A_3	B_2	A_2	C_4
A_2	B_1	A_1	C_1
A_2	B_1	A_2	C_1
A_2	B_1	A_1	C_2
A_2	B_1	A_3	C_3
A_2	B_1	A_1	C_3
A_2	B_1	A_2	C_4

$r_1 * r_2$ =

A	**B**	**C**
A_1	B_1	C_1
A_1	B_1	C_2
A_1	B_1	C_3
A_2	B_2	C_1
A_2	B_2	C_4
A_1	B_2	C_1
A_1	B_2	C_2
A_1	B_2	C_3
A_3	B_2	C_3
A_2	B_1	C_1
A_2	B_1	C_4

It is not a loss less join decomposition

Since $r_1 * r_2 \neq r$

Here $r_1 * r_2$ contains five additional tuples which are not in r.

Necessary Condition for a loss-less-join decomposition

A decomposition of relational schema R into R_1 and R_2 (such that $R_1 \cup R_2 = R$) will be a loss-less-join decomposition (No-additive), if the common attributes of R_1 and R_2 i.e., form candidate key of either of R_1 or R_2 or both.

i.e $R_1 \cap R_2 \rightarrow R_1$

(or)

$R_1 \cap R_2 \rightarrow R_2$

Heafh's Theorem:-If a relation schema R(α, β, γ) FD $\alpha \rightarrow \beta$ holding on it, then it can have a loss less decomposition as follows.

$R_1(\alpha, \beta)$ and $R_2(\alpha, \gamma)$

since $R_1 \cap R_2 = \{ \alpha \} \rightarrow R_1$

N-Ary Loss-Less-Join Decomposition:-

An N-ary decomposition of R into $R_1, R_2, R_3 \ldots\ldots R_n$ (such that $R_1 \cup R_2 \cup R_3 \ldots\ldots \cup R_m = f$) is said to be loss-less-join decomposition (Non-Additive) if each of the

Decomposition R_i satisfy the following.

$R_i \cap R_k \rightarrow R_i$

$R_i \cap R_k \rightarrow R_k$

Where R_k is the union of all decompositions

$R_1, R_2, R_3 \ldots\ldots R_m$ except R_i

Restrictions of FDs to a Desomposition

Let F be the set of FDs holding on a Relational schema R then the restriction of F to decompose R_i (denoted by F_i) is defined as follows.

$F_i = \{\alpha \rightarrow \beta \ / \ (\alpha \rightarrow \beta) \in F^+ , \alpha \subseteq R_i , \beta \subseteq R_i \}$

i.e F_i is the set of FDs that belonging to $F^+ \& \alpha \cup \beta \subseteq R_i$

Dependency-Preserving Decomposition:-

Let F be the set of FDs holding on a schema R, having decomposition (R_1, R_2 …..R_m) such that $R_1 \cup R_2$ …..$\cup R_m$ = R.

Let { F_1 ,F_2 …..F_m } be the restrictions of F to R_1 ,R_2 ….. respectively.

Let $F^1 = F_1 \cup F_2 \cup F_3$ …..$\cup F_m$

The decomposition is said to be dependency preserving if $\mathbf{F^{1+} = F^+}$ i.e each of FD of F^+ must be preserved in at least one of the decompositions, else the decomposition's non-dependency preserving.

Ex:- Consider a relation R(A, B, C) and set of FDs

F: {A→B , B→C }

F^+={ A→B , B→C, A→C}

Let us consider the decomposition R_1(A,B) and R_2(B,C).

The restriction of F to R_1 is

F_1 = {A→B}

The restriction of F to R_2 is

F_2 = {B→C}

$F^1 = F_1 \cup F_2$

= { A→B , B→C }

F^{1+}= { A→B , B→C, A→C }

So

$F^{1+} = F^+$

So it is dependency preserving decomposition and also $R_1 \cap R_2$ = {B} which is primary/candidate key of R_2

So it is loss-less-join decomposition

Now consider the decomposition

R_1(A,B) and R_2(A,C)

The restriction of F to R_1 is

F_1 = {A→B}

The restriction of F to R_2 is

$F_2 = \{A \rightarrow C\}$

$F^1 = F_1 \cup F_2$

$= \{A \rightarrow B , A \rightarrow C\}$

$F^{1+} = \{A \rightarrow B , A \rightarrow C, A \rightarrow BC \}$

So

$F^{1+} \neq F^+$

So it is not dependency preserving decomposition but $R_1 \cap R_2 = \{A\}$ is the primary/candidate key of both the relations R_1 and R_2 . So it is loss-less-join decomposition.

For any decomposition, it is mandatory that it should be a loss-less-join decomposition to ensure database consistency and also desirable (not mandatory) that it should be dependency preserving decomposition. The reason is if each FD in F^+ is preserved in at least one of the decompositions, then satisfaction of FD can be verified a single relation itself ; else it would require matter join of more than one relation to verify some FDs. A natural join operation would be too costly in turn of execution of time and memory.

Full Functional Dependency:-

Let there be a relational scheme R and with candidate key K and a non-prime attribute A ($K \subseteq R$, $A \in R$) . The FD $K \rightarrow A$ is said to be Full Functional Dependency, if the non-prime attribute A cannot be determined by any proper subset of K .i.e there does not exists $K_1 \subseteq K$ for which $K_1 \rightarrow A$ holds.

Partial Functional Dependency:-

Let there be a relational schema R with a candidate key k and a non-prime attribute A ($K \subseteq R$, $A \in R$) . The FD $K \rightarrow A$ is said to be a Partial Functional Dependency, if the non-prime attribute A can be determined by a proper subset of K i.e there exists $K_1 \subseteq K$ for which $K_1 \rightarrow A$ holds.

Ex:- Consider a Relation R (A,B,C,D,E) having atomic domains from A to E and Let

F : $\{AB \rightarrow C, A \rightarrow D, D \rightarrow E\}$ be the set of FDs holding on it.

Since all attributes of R have only atomic domains, it is in 1NF and {A,B} forms candidate key

- The non-prime attribute C (science it is not part of candidate key) is determined only by full candidate key so $AB \rightarrow C$ is called Full Functional Dependency.

- The non-prime attributes D,E can be obtained by the proper subset of candidate key [A→D, D→E, A→E] so A→D, A→E are the Partial Functional Dependencies.

Example:-

Consider a schema SP1 (sid, pid, sname, scity, status, pname, Qty)

where Sid = Supplier Id number (unique)

Pid = part Id number (unique)

Sname = Supplier Name

Scity = Supplier City

Status = Supplier status which depends on supplier city 'scity'

Pname = Part Name

Qty = Quantity of part (PId) to be supplied by a supplier (Sid)

Suppose the following FDS holding on the schema SP1

Sid →sname,scity

Scity→status

Pid→pname

{sid,pid} →Qty

Instance of a Relation Defined on SP1

Sid	Pid	Sname	Scity	Status	Pname	Qty
S_1	P_1	Sone	Hyd	10	Engine	5
S_1	P_2	Sone	Hyd	10	Generator	5
S_2	P_1	Stwo	Pune	20	Engine	2
S_2	P_3	Stwo	Pune	20	Altimeter	5
S_3	P_2	Sthr	Hyd	10	Generator	10
S_3	P_3	Sthr	Hyd	10	Altimeter	20

Insertion Anomalies:

(i) Information about a supplier like sname, scity can be inserted only when the supplier is supplying at least one part

(ii) Information about a part like pname can be inserted only when the part is being supplied by at least one supplier.

(iii) information about city like status can be inserted only when there is at least one supplies that city and the supplier is supplying at least one part.

Deletion Anomalies:-

(i) If the supplier is supplying only one part and that part supply is concluded, then the tuple relating to that supply will be deleted. With the deletion of that tuple, we would lose complete into about supplier i.e. its name & city.

(ii) Suppose a part is being supplied by only one supplier on completion of that supply, when the related tuple is deleted, we would lose the information about the name of the part.

(iii) Suppose a city has only one supplier and that supplier is supplying only one part. On deletion of the tuple of that particular supply, we would lose the info about the status of that city.

Update Anomalies:-

There is data redundancy like

(i) Info about sname and scity of a particular supplier will be appearing many times as the no. of parts supplied by that supplier.

(ii) Information about the name of a particular part will appear as many times as the no. of supplies relating to that part.

(iii) Info about the status of city will be appearing as the no. of supplies from the supplier of the city.

2nd Normal Form

A relation schema R is said to be in 2nd NF if and only if

(i) It is in 1st NF and

(ii) Each non-prime attribute of R is fully function dependency by the candidate keys of R i.e.

R: does not involve any partial functional dependencies.

Ex:

Consider a Relation R (A,B,C,D,E) having atomic domains from A to E and Let

F_R : {AB→C, A→D, D→E} be the set of FDs holding on it.

Since all attributes of R have only atomic domains, it is in 1NF and {A,B} forms candidate key

The above example relation schema R (A,B,C,D,E) is not in 2nd NF because it involves partial functional dependencies A->D,A->E.

Decomposing 1st NF schema into 2nd schemas

Using heafh's theorem, R can be lose-less decomposed into

R_{11} (A, D,E) and R_{22} (A,B,C)

So

R_{11} (A, D,E) → primary key{A}

F_{R11}={A->D,D->E},By transitivity A->E also holds on R_{11}

R_{22} (A,B, C)→primary key{A,B},foreign key{A} references

F_{R22} ={AB→C}

since R11∩R_{22}→R_{11}→{A},the decomposition of R into r_{11} and r_{22} is loss-less –join decomposition.

Now let us consider schema SP1

{Sid, Pid } is a candidate key of SP1&this is candidate key of SP1.

so ,Sid, Pid are the prime attributes and all other are non-prime attributes.

Since Sid →Scity and Scity→status, Sid→status,holds,SP1 involves the following FDs.

F_{sp1}={Sid→{Sname, Scity}, scity→status, Pid→Pname,{sid, Pid}→Qty}

Sp1 can be decomposed into S and SP_{11} on the basis of partial FD Sid→{Sname, Scity}

S(Sid, Sname, Scity, status)

primary key {Sid}

F_S={Sid→Sname, Sid→scity, Scity→status}

By transitivity Sid→status

Sp_{11} (Sid, pid, Pname, Qty)

primary key {Sid, Pid}

Pid→pname

{Sid ,Pid}→Qty

The above decomposition is loss less join decomposition.

Since

$S \cap SP_{11} = \{Sid\} \rightarrow S$

The schema S has no partial dependency so it is in 2NF however SP_{11} has a partial FD Pid→Pname and it is still not in 2NF.

Decompose SP_{11} into SP_2 and P on the basis of partial FD

Pid→P name

P (Pid,Pname)

primary key { Pid }

F_P= {Pid→Panme}

SP_2

SP_2(Sid, Pid, Qty)

primary key {Sid, Pid}

Foreign key {Sid} references S

F_{SP2}={{Sid, Pid}→Qty}

Foreign key { Pid} references p

So,

The projections of SP_1 over S, P and SP_2 are

S

Sid	Sname	Scity	status
S1	Sone	Hyd	10
S2	Stwo	Pune	20
S3	Sthree	Hyd	10

P

Pid	Pname
P1	Engine
P2	Generator
P3	altimiter

SP_2

Sid	Pid	Qty
S1	P1	5
S1	P2	5
S2	P1	2
S2	P3	5
S3	P2	10
S3	P3	20

THIRD NORMAL FORM (3NF)

A relation schema r is said to be in 3NF if

a) If is in 2^{nd} NF
b) Each non-prime attribute of R is non-transitive dependent on its candidate keys i.e R does not involves any transitive decencies.

Thus R_{22} is in 3NF but $R_{11}$is not in 3NF, since it involves transitive dependencies i.e A→D, D→E, therefore A→E.

DECOMPOSITION OF 2NF SCHEMA INTO 3 NF SCHEMAS

R_{11} can be decomposed into R_{31} and R_{32}

R_{31} (D,E)

primary key {D}

F_{R31}={D→E}

R_{32}(A,D)→ primary key {A}

F_{R32}={A→D} foreign key{D} references R_{31}

The decomposition of R_{11} into R_{31} and R_{32} is loss less join decomposition

since

$R_{31} \cap R_{32} \rightarrow R_{31} \rightarrow \{D\}$

R_{31}&R_{32} do not involve any transitive dependency .so they are in 3NF.

Let us consider the decomposition of SP_1

The schema P and SP_2 do not have any transitive dependencies so they are in 3NF. But S has a Transitive dependency i.e

Sid→Scity,Scity→status :-Sid→status

Decomposing of S on the basis of FD Scity→status into STS and SUPP.

STS(Scity, status)

primary key{Scity}

F_{STS}={Scity→status}

SUPP(Sid,Sname,Scity)

primary key{Sid}

Foreign key{Sity} references STS

F_{SUPP}={ Sid→{Sname, Scity} }

Now STS and supp do not have any Transitive Dependency; so both are in 3NF.

The decomposition is loss less, since

STS∩SUPP = {scity} → STS

The projections of S over STS and supp are

Supp

Sid	Sname	Scity
S1	Sone	Hyd
S2	Stwo	Pune
S3	Sthr	Hyd

STS

Scity	Status
Hyd	10
Pune	20

MVD (Multi Valued Dependencies):-

A relational schema R(α, β, γ) is said to have multivalued dependencies $\alpha \rightarrow\rightarrow \beta$

(α multi determined β) , $\alpha \rightarrow\rightarrow \gamma$ (α multi determined γ), if and only if for each and every legal $\gamma(R)$ and for tuple pair $\{t_1 , t_2\} \in$ γ or that satisfies $t_1[\alpha] = t_2[\alpha]$,

$\gamma_1[\beta] \neq \gamma_2[\beta]$, $t_1[\gamma] \neq t_2[\gamma]$, there exists a tuple pair $\{t_3 , t_4\} \in \gamma$ that satisfies

$t_3[\alpha]=t_4[\alpha] =t_1[\alpha]=t_2[\alpha]$ and $t_3[\beta]=t_1[\beta]$, $t_4[\beta]=t_2[\beta]$ and $t_3[\gamma]=t_2[\gamma]$, $t_4[\gamma]=t_1[\gamma]$. The MVDs always occur is pairs like $\alpha \rightarrow\rightarrow \beta$ and $\gamma \rightarrow\rightarrow \gamma$ both can be jointly denoted as $\rightarrow\rightarrow {}^{\beta}/_{\gamma}$.

Trivial MVD:-

An MVD $\alpha \rightarrow\rightarrow \beta$ holding on a schema R is said to be trivial iff

a) $\beta \subseteq \alpha$

(or)

b) $\alpha \cup \beta = R$

Fagin's Theorem:-If A relational schema R(α, β, γ) has a MVD $\alpha \rightarrow\rightarrow \beta/\gamma$ holding on it , then it can be loss-less-join decomposition into $R_1(\alpha, \beta)$ and $R_2(\alpha, \gamma)$

i.e r = $\pi_{(\alpha,\beta)}(r)$ * $\pi_{\alpha,\gamma}$(r)

Consider a relational schema CTX (course,Teacher,Text) with the following

Constraints

a) A course can be taught by more than one teacher
b) Number of text books can be followed for teaching a course
c) The teacher and text book columns are completely independent of each other.

There exists a one-to-many relationship from course to Teacher and course to Text . But there is no relationship between teacher and text . This situation represents a MVD course →→ Teacher/Text .

CTX

Course	Teacher	Text

OS	Ravi	text 1
OS	Vijay	text 2
OS	Ravi	text 2
OS	Vijay	text 1
CO	Ram	text 3
CO	Syam	text 4
CO	Ram	text 4
CO	Syam	text 3

The schema CTX is "all key" schema .

The CTX relation has following Anomalies:-

a) In info of particular teacher is repeated as many times as the no. Of text-books.
b) Info about particular text book is repeated as many times as the no. Of teachers.

Fourth Normal Form (4NF):-

A relational schema R is said to be in 4NF , if and only if every MVD $\alpha \rightarrow\rightarrow \beta$ holding on R satisfies either of the following two conditions .

a) It is trivial MVD or
b) α is a super key of R

Now the relation CTX has non-trivial MVDs course → Teacher and course →Text .

These are non trivial MVDs also course is not super key of CTX . So CTX is not in 4NF .

Loss-Less-Join Decomposition Based on MVDs:-

As per Fagin's theorem , A relational schema R(α, β, γ) satisfying $\alpha \rightarrow\rightarrow {}^{\beta}/_{\gamma}$ can be loss-less-join decomposition into $R_1(\alpha, \beta)$ & $R_2(\alpha, \gamma)$. So CTX can be divided into CT and CX

CT

Course	Teacher
OS	Ravi
OS	Vijay
CO	Ram
CO	Syam

CX

Corse	Text
OS	Text1
OS	Text2
CO	Text3
CO	Text4

CT × CX

Course	Teacher	Course	Text
OS	Ravi	OS	Text 1
OS	Ravi	OS	Text2
OS	Ravi	CO	Text3
OS	Ravi	CO	Text4
OS	Vijay	OS	Text1
OS	Vijay	OS	Text2
OS	Vijay	CO	Text3
OS	Vijay	CO	Text4
CO	Ram	OS	Text1
CO	Ram	OS	Text2
CO	Ram	CO	Text3
CO	Ram	CO	Text4
CO	Syam	OS	Text1
CO	Syam	OS	Text2
CO	Syam	CO	Text3
CO	Syam	CO	Text4

CT * CX

Course	Teacher	Text

OS	Ravi	Text1
OS	Ravi	Text2
OS	Vijay	Text1
OS	Vijay	Text2
CO	Ram	Text3
CO	Ram	Text4
CO	Syam	Text3
CO	Syam	Text4

So,CT * CX =CTX so the decomposition is loss less join decomposition and

The relations CT and CX are in the 4NF because the first condition of 4NF is satisfied with both the relations CT and CX.

ALTERNATIVE DEFINITION FOR 3NF

A relation schema R is sad to be in 3NF , if each FD $\alpha \rightarrow \beta$ holding on R satisfies one of the following three conditions.

(a) It is trivial FD

Or (b) α is super key of R

Or(c) each attribute in the set $\beta \rightarrow \alpha$ is prime attribute.

For example consider the relation

SP (Sid, Sname, Pid, Qty)

F_{sp}={{Sid,Pid}→Qty}

{Sname,Pid}→Qty,

Sid→Sname,

Sname→Sid} holding on it

Candidate keys are {Sid,Pid},{Sname,Pid}

Both the candidate keys are composite and have common attribute Pid.

Set of prime attribute-{Sid, Pid, Sname}

Set of non-prime attribute ={Qty}

The schema SP is free of any partial functional dependencies or a transitivity dependency, so it is 3NF.

Inspite of being in 3NF ,SP may contain some data redundancies like the name of particular supplier i.e Same will be repeated as many times as the number of supplier being mode by that supplier. That, there is need to have normal form, stronger than 3NF. The necessary solution is providing by BCNF.

BOYCE CODD NORMAL FORM(BNF):- A relation schema R is said to be in BCNF, if all non-trivial left-irreducible FD's that hold on R have candidate key as determinants. Alternatively, we can state that a relation schema R would be in BCNF if each FD $\alpha \rightarrow \beta$ holding on R satisfies one of the following conditions:-

(a) It is a trivial FD (or)
(b) α is a super key of R

The above two conditions for BCNF are same as the first two conditions of 3NF thus, if a schema is in BCNF, if a schema is in BCNF, it must also be in 3NF. The third condition of 3NF is missing against BCNF so it give more restrictive as compared to 3NF.BCNF is more strong than 3NF.

From the definition of BCNF SP is in 3NF but not in BCNF science the two FD i.e Sid→Sname, Sname→Sid are neither trivial nor super keys having on left side.

Alternatively if can be stated that a Relational schema R will be in BCNF if each non-trivial left-irreducible FD $\alpha \rightarrow \beta$ holding on R ,has only candidate keys on its left side i.e α must be a candidate key of R.

Decomposition of SP into a BCNF schema

SP can be decomposed into BCNF schemas, on the basis of the FD's that violate BCNF i.e Sid→Sname and Sname→Sid. The resulting BCNF decomposition of SP will be

S(Sid,Sname) primary key {Sid} or {Sname}

Fs={Sid→Sname,Sname→Sid}

SP1(Sid,Pid,qty) primary key {Sid,Pid}

Foreign key {Sid} reference S

FSP1={Sid,Pid}→Qty.

(OR)

S(Sid,Sname) primary key {Sid} or{Sname}

F_s={Sid→Sname,Sname→Sid}

SP_2(Sname,Pid,Qty) primary key{Sname,Pid}

Foreign key {Sname } references S

F_{SP2}={Sname,Pid }→Qty

Surrogate Key:

A surrogate key is an artificially generated key. They're useful when your records essentially have no natural key

Most often you'll see these implemented as integers in an automatically incrementing field, or as GUIDs that are generated automatically for each record. ID numbers are almost always surrogate keys.

A Surrogate Key should have the following characteristics:

- Unique Value
- The key is generated by the system, in other words automatically generated
- The key is not visible to the user (not a part of the application)
- It is not composed of multiple keys
- There is no semantic meaning of the key