NORMALIZATION :-

⇒ A large database defined as a single relation may result in data duplication. This repetition of data may result in :

i) Making relation very large.

ii) It isn't easy to maintain and update data as it would involve searching many records in relation.

iii) Wastage and poor utilization of the disk space and resources.

iv) The likelihood of errors and inconsistencies

⇒ So to handle these problems, we should analyze and ~~decom~~- decompose the relation with redundant data into smaller, simpler and well-structured relations that are satisfy desirable properties.

⇒ Normalization is a process of decomposing the relation into relation with fewer attributes.

What is Normalization? :-

⇒ Normalization is the process of organizing the data in the database.

⇒ Normalization is used to minimize the redundancy from a relation or set of relations.

⇒ It is also used to eliminate undesirable characteristics like Insertion, Updation and Deletion Anomalies.

⇒ Normalization divides the larger table into smaller and link them using relationships.

⇒ The normal form is used to reduce redundancy from the database ~~ass~~ model.

Scanned with OKEN Scanner

DATA MODIFICATION ANOMALIES :-

▶ **Insertion Anomaly –**

Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.

▶ **Deletion Anomaly –**

The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.

▶ **Updation Anomaly –**

The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

Example:-

Scanned with OKEN Scanner

Why Normalization is required ? :-

⇒ The main reason for normalizing the relations is removing these anomalies.

⇒ Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows.

⇒ Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.

Types of Normal forms :-

i) 1st
ii) 2nd
iii) 3rd
iv) BCNF
v) 4th
vi) 5th

Advantages of Normalization :-

⇒ Normalization helps to minimize data redundancy.

⇒ Data consistency within the database.

⇒ Much more flexible database design.

⇒ Enforces the concept of relational integrity.

Disadvantages Of Normalization :-

⇒ You cannot start building the database before knowing what the user needs.

⇒ The performance degrades when normalizing the relations to higher normal forms i.e. 4NF, 5NF.

⇒ It is very time consuming and difficult to normalize relation of a higher degree.

⇒ Careless decomposition may lead to a bad database design

Scanned with OKEN Scanner

First Normal Form :-

⇒ A relation will be in 1NF if it contains atomic values.

⇒ It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.

⇒ First Normal form disallows the multi-valued attribute, composite attribute and their combinations.

⇒ Consider the following table :

Roll No	Name	Course
1	Aakash	C, DBMS.
2	Ajit	Python
3	Ashok	PHP, Python.

⇒ It is not in 1st NF as it contains multiple values.

Option 1 :- X

Roll No	Name	Course 1	Course 2
1	Aakash	C	DBMS
2	Ajit	Python	NULL
3	Ashok	PHP	Python

Option 2 - X

Roll No	Name	Course
1	Aakash	C
1	Aakash	DBMS
2	Ajit	Python
3	Ashok	PHP
3	Ashok	Python

Solution —

→ Primary Key (P.K)

Roll No	Name
1	Aakash
2	Ajit
3	Ashok

→ (P.K).

Roll No	Course
1	C
1	DBMS
2	Python
3	PHP
3	Python.

Scanned with OKEN Scanner

SECOND NORMAL FORM :-

⇒ In the 2NF, relational must be in 1NF.

⇒ In the second normal form, all non-key attributes are fully functional dependent on the candidate key.

⇒ There should not be any partial dependency.

NOTE :-

When all the non-key attribute fully depend on candidate key then it is known full dependency else partial dependency.

Q $R(ABCDEF)$

FD : $\{C \rightarrow F, E \rightarrow A, EC \rightarrow D, A \rightarrow B\}$.

$(CE)^+ = CFAEBD$

$(ABCDE\cancel{F})^+ = \{A, B, C, D, E, F\}$.

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

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

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

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

$\{C\} = \{F\}$

$\{E\} = \{A\}$.

Candidate Key – $\{CE\}$

Prime Attribute – $\{C, E\}$

Non-Prime Attribute – $\{A, B, D, E\}$.

No, the table is not in 2nd Normal form

P.K

$R_1 (E\ C\ D)$

$R_2 (C\ F)$

$R_3 (E\ A\ B)$

PK

Scanned with OKEN Scanner

Q2 R(ABCDE)

F.D: {AB → C , D → E}

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

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

$ABD^+ = \{A, B, C, D, E\}$. → CK

A = {A}.

B = {B}

D = {D}

Prime = {A, B, D}

Non Prime = {C, E}.

it is not in 2nd normal form.

R1 (ABD) PK

R2 (AB C) PK

R3 (D E) PK

Q3 R(ABCDE)

FD: {A → B, B → E, C → D} (P.D., P.D)

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

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

$AC^+ = \{A, B, E, C, D\}$ → CK

{A}

{C}

Prime Attribute ⟨A, C⟩

Non Prime Attribute ⟨B, D, E⟩

No, it is not in 2nd NF.

R1 (AC)

R2 (A → B → E)

R3 (C → D)

Scanned with OKEN Scanner

Q R (ABCDEFGHIJ) AP.D.

FD: { $AB \xrightarrow{P.D} C$, AD → GH , BD → EF , A → I , H → J }

{ AD → G , AD → H
BD → E , BD → F .

ABCDEFGH~~IJ~~$^+$ = {A, B, C, D, E, F, G, H, I, J}

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

ABDEF~~G~~H$^+$ = {A, B, C, D, GH, E, F, I, J}.

ABD~~EF~~$^+$ = {A, B, C, D, E, F, I, G, H, J}

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

{A} = {I, A}
{B} = {B}
{D} = {D}.

Prime ⟨A, B, D⟩

Non Prime ⟨C, E, F, G, H, I, J⟩.

No, it is not in 2nd Normal form

R1 (ABD)

R2 (AB → C)

R3 (AD → GH → J)

R4 (A → I)

R5 (BD → EF)

~~R6~~

Q R (ABCDE)

FD: {AB → CD, C → A, D → B}.

A~~B~~CD = {AB, C, D}

Q R (ABCDE)

FD: {A → B, B → C, C → D, D → E}

AB~~C~~DE$^+$ = {A, B, C, D, E}

A~~BD~~E$^+$ = {A, B, C, D, E

⇒ AE$^+$ = {A, B, C, D, E}

A = {A, B, D, C, E}

E = {E}

R1 (AE)

R2 [A → B → C → D → E]

~~R3~~

THIRD NORMAL FORM :-

= In 3NF, the table must be in 2NF and doesn't contain any transitive functional dependency.

= 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.

= If there is no transitive dependency for non-prime attributes, then the relation must be in 3NF.

Q $R(ABCD)$

FD: $\{AB \rightarrow \underline{C},\ \underline{C} \rightarrow D\}$

Candidate Key – AB

Prime Attribute – $\{A, B\}$

Non-Prime Attribute – $\{C, D\}$

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

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

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

$\{A\} = \{A\}$

$\{B\} = \{B\}$

Not in 3NF.

$R_1\ (AB \rightarrow C)$

$R_2\ (\cancel{AB} \rightarrow C \rightarrow D)$.

Q $R(ABCDE)$

FD: $\{A \rightarrow B,\ \cancel{B} \rightarrow E,\ C \rightarrow D\}$

Candidate Key = AC

Prime Attribute – $\{A, C\}$

Non-Prime Attribute – $\{B, D, E\}$

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

$ABC\cancel{D} = \{A, B, C, D, E\}$

$A\cancel{B}C = \{A, B, C, D, E\}$.

$AC = \{A, B, C, D, E\}$.

Not in 2nd NF.

~~Not in~~ 3nd NF

2NF

$R_1\ (AC)$

$R_2\ (A \rightarrow B \rightarrow E)$

$R_3\ (C \rightarrow D)$.

~~3NF~~

~~R_1 (AC)~~

$R_{21}\ (A \rightarrow B)$.

$R_{22}\ (B \rightarrow E)$ ~~$R_4\ (C \rightarrow D)$~~

Scanned with OKEN Scanner

Q. $R(ABCDEF\cancel{G}\cancel{H}IJ)$ 2nd NF.

FD: $\{AB \to C,\ A \to DE,\ B \to F,\ F \to GH,\ D \to IJ\}$

$\left\{ \begin{array}{ll} A \to D, & A \to E \\ F \to G, & F \to H \\ D \to I, & D \to J \end{array} \right.$

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

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

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

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

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

Candidate Key = AB.

Prime Attribute = {A, B}.

Non-Prime Attribute = {C, D, E, F, G, H, I, J}

~~R1 (AB)~~

$R_1\ (AB \to C)$

$R_2\ (A \to DE \to IJ)$ — R_{21} — $(A \to DE)$, R_{22} — $(D \to IJ)$

$R_3\ (B \to F \to GH)$ — R_{31} — $(B \to F)$, R_{32} — $(F \to GH)$

Q. R(ABCDE)

FD: $\{AB \to C,\ B \to D,\ D \to E\}$.

$ABCD\cancel{E}^+ = \{ABCDE\}$

$AB\cancel{D}\cancel{E}$ = CK.

Candidate key –

Scanned with OKEN Scanner

BCNF (Boyce Codd Normal Form) –

= BCNF is the advance version of 3NF. It is stricter than 3NF.

= A table is in BCNF if every functional dependency $X \to Y$, X is the super key of the table.

= For BCNF, the table should be in 3NF and for every FD, LHS is super key / candidate key.

Q. R(ABC)

FD: {AB → C , C → B}

Candidate 'Key = AB'

Prime Attribute ⟨A, B, C⟩.

$AB\cancel{C} = \{A, B, C\}$.

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

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

$AC^+ = \{A, C, B\} \to CK$

~~Yes the ta~~

~~R_1 (AB → C → B)~~

~~R_2 (C → B)~~

~~R_2 (AB → AB → C)~~

~~R_3 (AC)~~

- R_1 (C → B)
- R_2 (AB)
- R_3 (AC)

R_1 (C → B)

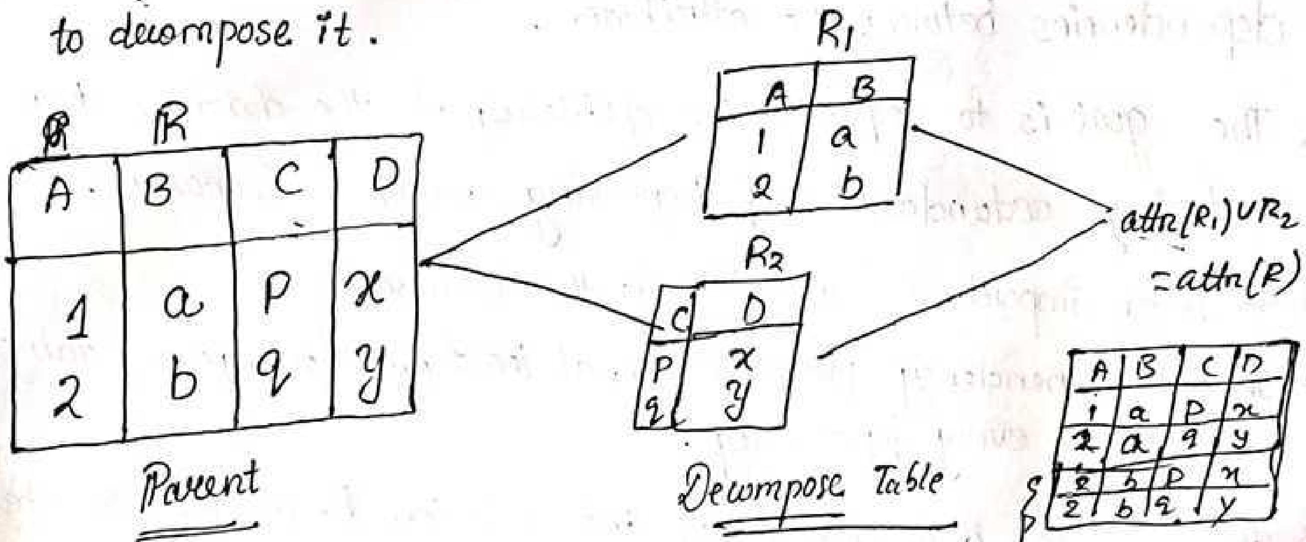
R_2 (AB)

Scanned with OKEN Scanner

Decomposition :-

= The process of breaking up or dividing a single relation into two or more sub relations is called as decomposition of a relation

= Whenever the relation is not, in proper Normal form we need to decompose it.



The tuple which we get extra are known as Spurious Tuple.

Rule 1 : Attribute of $R_1 \cup R_2$ = Attribute of R

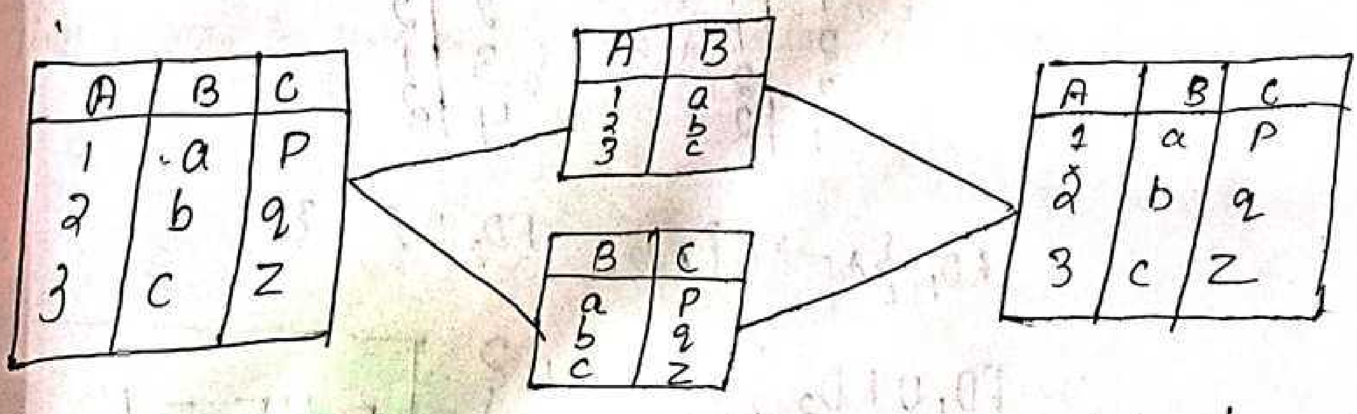
i.e $attr(R_1) \cup attr(R_2) = attr(R)$

Rule 2 : Attribute of $R_1 \cap$ Attribute of $R_2 \neq \phi$

i.e. $attr(R_1) \cap attr(R_2) \neq \phi$

Rule 3 : Attribute of $R_1 \cap$ Attribute of $R_2 \longrightarrow$ Attribute(R_1)

Attribute (R_1) $\cap$ Attribute (R_2) $\longrightarrow$ Attribute (R_2)



Here, table is following lossless decomposition

Dependency Preserving :-

⇒ Dependency Preserving Decomposition is a technique used in Database Management System (DBMS) to decompose a relation into smaller relations while preserving the functional dependencies between the attributes.

⇒ The goal is to improve the efficiency of the database by reducing redundancy and improving query performance.

⇒ It is an important constraint of the database.

⇒ In the dependency preservation, at least one decomposed table must satisf every dependency.

⇒ If a relation R is decomposed into relation R_1 and R_2, then the dependencies of R either must be a part of R_1 or R_2 or must be derivable from the combination of functional dependencies of R_1 and R_2.

$\boxed{R_1 \cup R_2 \equiv R}$ ⟶ Rule.

⇒ Example :-

R

A	B	C
1	1	1
2	1	2
3	2	1
4	2	2

FD: $\{A \rightarrow BC, BC \rightarrow A\}$

$(A \rightarrow C, A \rightarrow B)$

R_1

A	B
1	1
2	1
3	2
4	2

R_2

A	C
1	1
2	2
3	1
4	2

$FD_1: \{A \rightarrow B\}$ $FD_2: \{ \quad \}$

$FD_1 \cup FD_2 \{A \longrightarrow B\}$

$F_1 \cup F_2 \equiv F$

Scanned with OKEN Scanner

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

$FD: \{A \to B, B \to C, C \to D, D \to A\}$

$R_1 (A, B, C)$ $R_2 (C, D, E)$

$FD_1: \{$

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

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

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

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

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

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

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

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

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

$E^+ = \{E\}$

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

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

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

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

$FD_1: \{A \to BC, B \to AC, C \to AB, AB \xrightarrow{\times} C, AC \xrightarrow{\times} B, BC \xrightarrow{\times} A\}$

$\Rightarrow FD_1: \{A \to BC, B \to AC, C \to AB, \cancel{AC \to B}, \cancel{BC \to A}\}$

$FD_2: \{C \to D, D \to C, CE \to D, DE \xrightarrow{\times} C\}$

$\Rightarrow FD_2: \{C \to D, D \to C, CE \to D\}$

$G \{A \to BC, B \to AC, C \to AB\}, CE \to D\}$

This, relation is preserving the function dependency.

Scanned with OKEN Scanner

Multivalued Dependency :-

⇒ It occurs in between two attributes, that they are independent of each other and dependent on the third attribute.

FOURTH NORMAL FORM :-

⇒ A relation will be in 4NF if it's is in BCNF and has no multivalued dependency.

⇒ For a dependency A → B, if for a single value of A multiple values of B exists, then the relation will be a multi-valued dependency.

Scanned with OKEN Scanner

Example :-

Std_ID	Course	Hobby
21	Computer	Dancing
21	Math	Singing
34	Chemistry	Dancing
74	Biology	Cricket
59	Physics	Hockey

Independent

Here Course and Hobby are independent but both depend on Std_Id.

∴ Student_Course

Std_Id	Course
21	Computer
21	Math
34	Chemistry
74	Biology
59	Physics

Student_Hobby

Std_Id	Hobby
21	Dancing
21	Singing
34	Dancing
74	Cricket
59	Hockey

Join Dependency :-

⇒ If the join of R_1 and R_2 over C is equal to relation R, then we can that a join dependency (JD) exists.

⇒ Where R_1 and R_2 are the decompositions $R_1(A,B,C)$ and $R_2(C,D)$ of a given relations $R(A,B,C,D)$.

⇒ Alternatively, R_1 and R_2 are a lossless decomposition of R.

⇒ A join dependency $\bowtie \{R1, R2, R3, \ldots Rn\}$ is said to hold over a relation R if $R_1, R_2, \ldots\ldots R_n$ is a lossless join decomposition.

Scanned with OKEN Scanner

FIFTH NORMAL FORM :-

⇒ A relation is in 5NF if it is in 4NF and no join dependency exists. i.e lossless decomposition should be there.

⇒ It can't be further non loss decomposed.

⇒ 5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.

⇒ 5NF is also known as Project-Join Normal form.

⇒ Example:

Agent	Company	Product
A_1	PQR	Nut
A_1	PQR	Bolt
A_1	XYZ	Nut
A_1	XYZ	Bolt
A_2	PQR	Nut

Agent	Company
A_1	PQR
A_1	XYZ
A_2	PQR

Agent	Product
A_1	Nut
A_1	Bolt
A_2	Nut

Company	Product
PQR	Nut
PQR	Bolt
XYZ	Nut
XYZ	Bolt

Scanned with OKEN Scanner