DBMS
unit - 4, 2nd half

Informal design guidelines for Relational Schema

imp **

This are the informal measures of Quality while design R/nal Schema.
they are 4

1) imparting clear Symmantics to attributes in R/n.

whenever we group attribute to form a R/nal schema. we assume that attribute belonging to one R/n having certain real world meaning and proper interpretation associated with them the interpretation a Symmantics of a R/n refers to the interpretation of the attribute value in a tuples.

Guideline 1 : words

* design a R/nal schema so that it is easy to Explain its meaning. Do not combine attributes from multiple entity type and R/s types into a single R/ns. hence if a R/nal schema belongs to one entity type (or) 1 R/s type it is Straightforward to interpret and do Explain its meaning. Otherwise if the R/s corresponds to a R/n of mixture multiple entities and R/s symmantic eambiguity will result and the R/n cannot be easly explained

Gxi 2) Redundency infn in tuples + update anomalies ; of 1 of the Goals of schema design is to minimise the Storage Space used by base R/n. Because Grouping of attributes into R/ns Schema's as a Significant effect on the Storage Space.

for Example ; Emp-Dept

* another problem with The R/ns is the problem of update anamalies **

catagoria into 3 parts

1) Insert anamalie
2) Delete —"—
3) update / modification.

1) Insertion anamalie :- can be differentiate into 2 types which can be illustrated by The Example of Employee department R/n

* To insert a new employee tuple into Emp-dept we must include either the attribute values for the department that the employee works for (or) Nulls, if the employee does not work for the department yet and also while inserting the new tuple into the R/n. we should make sure that all the entired tuple is consistency with the existing data.

* It in difficult to insert a new department that as no employees are yet in emp-dept as R/n.

2) <u>Deletion anamalie</u> :- if we delete from Emp-dept an employee tuple that happens to represent the last employee working for a particular department the infn consering the department is also lost from the db.

3) <u>Modification anamolie</u> :- In Emp-dept if we change the value of one of the attribute of a particular department say the manager of dept no 5 we must update the tuples of all employee who work in that department, otherwise the db will become inconsistancy.

<u>Guideline 2</u>

* design the ~~~~ base R/n schema so that no insertion, deletion / modification anamalies are present in the R/ns. if any anamalie are present note them clearly and make sure that the programs that update the db will operate correctly.

3) <u>Null values in tuples</u> :- A Null value can have multiple interpretation such as * The attribute does not apply to this tuples.

* The attribute value for this tuple is unknown
* The value is known ~~~~ but it is missing.

because of the above interpreation when null values are present the result of join and aggregate operation are unpredecatable.

Guideline 3

* as far as possible avoid placing attribute in a base R/n whose values may frequently be null if null's are unavoidable. make sure that they applying exceptional case only and donot applying to a majority of tuples in the R/ns.

4) Generation of Spurious tuples

works_on

ssn	pno
1	1
1	2
2	1

proj

pno	pname	ploc
1	x	abc
2	y	bca

Emp

ssn	name
1	a
2	b

Emp-loc

name	plocal
a	abc
a	bca
b	abc

Emp-Proj

ssn	Pno	Pname	ploc
1	1	x	abc
1	2	y	bca
2	1	x	abc

Natural join of E * w

ssn	pno	pname
1	1	a
1	2	a
2	1	b

(E * w) * proj

ssn	pno	pname	ploc	name
1	1	x	abc	a
1	2	y	bca	a
2	1	x	abc	b

ssn	pno	name	loc	name
1	1	x	abc	a
1	2	y	bca	a
2	1	x	abc	b
2	1	x	abc	a
1	1	x	abc	b

Guidline 4

Design R/nal schema's so that they can be join with equality conditions on attributes that are either primary key's (or) foregin key's in a way that guaranteeu's that no spurious tuples are generated avoid R/ns that contain matching attributes that are not (PK, FK) combinations, because joining on the such attributes may produced spurious tuples.

Functional Dependency (FD (or) fd)

A FD denoted by $X \rightarrow Y$ b/w 2 set of attributes x and y that are subset R [R is an universal R/nal schema] where ($R = (A_1, A_2 \ldots A_n)$) specify the contraint on the possible tuples that can form a R/nal state $r(R)$. The contraints says that for any 2 tuples T_1 and T_2 in r that have $T_1(x) = T_2(x)$ they must also have $T_1(y) = T_2(y)$

$x \rightarrow y$ (x fd y or y fd x)

* R/n Extension $r(R)$ that satisfy the fu^n^al dependency contraints are called as legal R/nal states.

* The fu^n^al dependency is a properties of symmatic of meaning of attributes.

* if X determine y vice versa may not be true.

Emp
ssn name age d

Inference rules for FD

Set of all dependency that include a given set FD F as well as all dependencies that can be inferred from f is called closure closure of F and is denoted by F^+

We use the notation $F \models X \rightarrow Y$ to denote the final dependency x determine y is infered in the set of fu^n^al dependency f.

Rules.

1> Inference rule 1 [IR1] : Reflexive Rule.

if $X \supseteq Y$, then $X \rightarrow Y$

Proof : suppose that $Y \subseteq$ of X then there exits 2 tuples t_1, t_2 for some R/s $r(R)$ such that $t_1[X] = t_2[X]$. then $t_1[Y] = t_2[Y]$ hence $X \rightarrow Y$.

2> IR2 : Augmentation Rule

$\{X \rightarrow Y\} \models \{XZ \rightarrow YZ\}$

Proof :- Assumes that $X \rightarrow Y$ holds in a R/n instance $r(R)$ but $XZ \rightarrow YZ$ does not holds. then there must exits 2 tuples t_1 and t_2 such that

$t_1[X] = t_2[X]$ .... (1)

$t_1[Y] = t_2[Y]$ .... (2)

$t_1[XZ] = t_2[XZ]$ .... (3)

$t_1[YZ] \neq t_2[YZ]$ .... (4) from (1) + (3)

$t_1[Z] = t_2[Z]$ ..... (5) from (2) + (5)

$t_1[YZ] = t_2[YZ]$ this contradic the assumption.

$\therefore XZ \rightarrow YZ$

3> IR3 : Trasative rule

$\{X \rightarrow Y, Y \rightarrow Z\} \models \{X \rightarrow Z\}$

Proof :- Assume that $X \rightarrow Y$ and $Y \rightarrow Z$ both hold in a R/n state $r(R)$. then for any 2 tuples t_1 and t_2 in $r(R)$ such that $t_1[X] = t_2[X]$ we must also have $t_1[Y] = t_2[Y]$, then we must also have $t_1[Z] = t_2[Z]$

$\therefore X \rightarrow Z$

4> IR4 : Decomposition of projective rule.

this says $\{X \rightarrow YZ\}$ then $\models \left\{\begin{matrix} X \rightarrow Y \\ X \rightarrow Z \end{matrix}\right\}$

Proof :- $X \rightarrow YZ$ is given, $YZ \rightarrow Y$, from the IR1 rule, become $X \rightarrow Y$ from IR3,

Similarly $YZ \rightarrow Z$

$\therefore X \rightarrow Z$.

5> IR5 : union / additive rules

$\{X \to Y, X \to Z\} \models \{X \to YZ\}$

Proof :- $X \to Y$, $X \to Z$ is given. Now consider

$X \to Y$ ------ (1)

$X \to Z$ ------ (2) now agment X on both the sides of 1 $X \to XY$ ------ (3)

$XY \to YZ$ ---- (4)

from (3) & (4)

$X \to YZ$ from (IR3)

6> IR6 : pseudo transitive rule

$\{X \to Y, WY \to Z\} \models \{WX \to Z\}$

Proof :- $X \to Y$, $WY \to Z$ is given

$X \to Y$ ------ (1)

$WY \to Z$ ------ (2) then agment W on both the sides of 1 $WX \to WY$ ---- (3)

from (3) & (2)

$WX \to Z$

Armstrong's inference rule. imp

i) IR1
2) IR2 } basic rule.
3) IR3.

$F = \{A \to B, B \to C\}$

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

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

$F^+ = \{A \to B, A \to C, B \to C\}$.

$XY \to Z$
$X, Y \to Z$ } both are same.

Algorithm for determining X^+ that ie the closure of X under F.

$X^+ := X$
repeat
  old $X^+ := X^+$
  for each fd $Y \rightarrow Z$ in F
  do
    if $X^+ \supseteq Y$ (covers)
    then
      $X^+ := X^+ \cup Z$
until
  ($X^+ =$ old X^+)

Example:

F = { ssn → ename, pno.
  pno → {pname, ploca}
  {ssn, pno} → hours calculate F^+

soln:- $\{ssn\}^+ = \{ssn, ename\}$

* ssn → ename.

simly $\{pno\}^+ = \{pname, pno, ploc\}$

* pno → pname, pno → ploc.

$\{ssn, pno\}^+ = \{ssn, pno, ename, pname, ploc, hours\}$

$F^+ = \{ F,$
  $\{ssn, pno\} \rightarrow ename$
  $\{ssn, pno\} \rightarrow \{pname, ploc\}$

~~Equivalent~~ Equivalence of sets of FD : imp *

A set of FD F is set to cover another set of functional dependencies E, if every func^n^al dependency in E is also in F^+ in other way if every dependency in E can be inforced from F then we call F covers E (or) E is covered by F

2 set of FD E and F are equivalent if $F^+ = E^+$

Minimal set (form) (cover) of FDependency / canonical set of FD

Minimal cover of a set of FD E is a set of FD F that satisfy the property that every dependency in E is in F^+. In addition if any dependency from set F is removed further F will not cover E.

* we can formally define a set of FDs F to be minimal if it satisfy the following conditions.

1. Every dependency in F has a single attribute for its right hand side.
2. we cannot replace any dependency X determines A in F with a dependency Y determines A, where Y is proper subset of X. and still have a set of dependency that is equivalant to F.
3. We cannot remove any dependecny from F + still have a set of dependency that is equivalent to F.
4. A minimal cover of a set of FD E is a minimal set of dependencies that is equivalent to E.

Algorithm for finding minimal cover F for a set of FD E.

Step 1 : set F := E

Step 2 : replaces each FD $X \rightarrow \{A_1, A_2 \ldots\ldots A_n\}$ in F by n FDs $X \rightarrow A_1$, $X \rightarrow A_2$ ...... $X \rightarrow A_n$.

Step 3 : For each fd $\alpha \rightarrow A$ in F
for each attribute B, which is an elmt of α, ~~belong to~~ i.e. ($B \in \alpha$)

if $\left\{ \begin{array}{l} F - \{\alpha \rightarrow A\} \cup \\ \{\alpha - B\} \rightarrow A \end{array} \right\}$ is equivalent to F.

then replace $\alpha \rightarrow A$ with $\{\alpha - B\} \rightarrow A$ in F

Step 4 : for each remaining fd $\alpha \rightarrow A$ in F
if $\{F - \{\alpha \rightarrow A\}\} \Longleftrightarrow F$ then remove $\alpha \rightarrow A$ from F.

Example ① compute the closure of following set F(fd) for R/nal schema $R = (A \ldots E)$ where $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ find the F^+.

Question: find out key attribute.

soln :- F^+: $A^+ = \{ABCDE\}$

$$\left. \begin{array}{l} A \rightarrow BC \\ A \rightarrow CD \\ A \rightarrow E \end{array} \right\} A \rightarrow BCDE$$

key attribute

candidate keys

Ily.

* $CD^+ = \{CDEBA\}$

$CD \rightarrow E$
$CD \rightarrow AB$

* $B^+ = \{BD\}$

* $E^+ = \{EABCD\}$

$E \rightarrow A$
$E \rightarrow BCD$

(or)

$B \rightarrow D$
$BC \rightarrow CD$
$BC \rightarrow E$
$BC \rightarrow ADE$

$F^+ = \{F \cup A \rightarrow DE, CD \rightarrow AB, E \rightarrow BCD\}$

② $R = (A \ldots \ldots F)$
$F_1 = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow F\}$
$F_2 = \{A \rightarrow CD, E \rightarrow AF\}$ check whether F_1 & F_2 are equivalent.

$F_1^+ = \{A^+ = \{A, C, D\}, AC^+ = \{A, C, D\}, E^+ = \{E, A, D, C, F\}\}$
$\therefore F_1^+$ includes $\{A \rightarrow C, AC \rightarrow D, A \rightarrow CD, E \rightarrow AD, AC, F\}$.

soln :- F_1 covers F_2

$A \rightarrow CD \}$ f2

$A \rightarrow C \}$ $A \rightarrow AC$
$A \rightarrow DC$

$A \rightarrow CD$

$E \rightarrow AF$

$E \rightarrow F$
$E \rightarrow AD$
$E \rightarrow A$, $E \rightarrow D$
$E \rightarrow AF$

(transitive rule)

F_2 covers F_1

* $A \to C$ } F_1
  $A \to C,\ A \to D$

* $AC \to D$ } F_1
  $A \to CD$ } $A \to C,\ A \to D$ (argument c)
  $AC \to CD$
  $AC \to C,\ AC \to D$

* $E \to AD$ } F_1
  $E \to A,\ E \to D$

$E \to AF$ }
$E \to A,\ E \to F$ (transitive rule)
$E \to D$ } $A \to CD$,
$A \to C,\ A \to D$

$E \to AD$.

$F_2^+ = \{A \to CD\}$ $A^+ = \{A, C, D\}$
$E \to AF\}$ $E^+ = \{E, A, C, D, F\}$

---

* $R = (A \ldots\ldots F)$
$F_1 = \{A \to C,\ AC \to D,\ E \to AD,\ E \to F\}$ find the minimal cover F_1

1) $A \to C$
$AC \to D$
$E \to A$
$E \to D$
$E \to F$

2) $A \to C$, $A \to A.C$ } $AC \to D$, $A \to D$

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

$A \to C$
$A \to D$
$G \to A$
$G \to F$

$A \to CD$
$E \to AF$

4) $A \to BC$
$B \to C$
$A \to B$
$AB \to C$

$A \to B$
$A \to C$
$B \to C$
$A \to B$
$AB \to C$

$AB \to C$
$A \to C$
$B \to C$

$A \to B$
$\boxed{A \to C}$
$B \to C$
$\boxed{A \to B}$
$\boxed{A \to C}$

$A \to B$
$A \to C$.

F_1 covers F_2.

$A \to B,\ B \to C,\ D \to ABC$
$AC \to D$

F_2 covers F_1 ?

$A \to BC$ } $A \to BC$ (F_2 : transitive)
$AB \to C$ } $B \to C$
$AB \to AC$
$AB \to C$

Normalization of Relations

Normalization of Data is considered as the process of analysing the given R/nal schemas based on there func'nal dependencies and primary keys to acheive the following properties.

1) Minimizing redundancy:
2) Minimizing insertion, deletion and updation anomalie
3) To garauent lossless Join (or) non - ~~negative~~ additive Join properties.
4) To guarantee dependency preservation property.

unsatisfactory R/nal schema that donot meet certain conditions (or) normal form tests (NF) are decomposited into smaller R/nal schemas that meets the Normal forms and hence satisfying the above set properties.

1) **First Normal form (1NF):** 1st NF takes that the domain of an attribute must includes only atomic values and the values of any attribute in a tuple must be a single valued from the domain of that attribute. In other words 1st NF disallow multi valued attributes, composite attribute and there combinations. The only attribute value permitted by 1NF (or) single atomic values.

Ex 1: Department

multi value

Dno	Dname	Dmgr_ssn	Dlocation
1	x	10	{a, b, c}

Dno	Dname	Dmgr_ssn

Dno	Dlocation

} 1NF.

Example 2: Emp_project

composite

ssn	name	Project	
		Pno	Pname
1		1	x

x

ssn	name

ssn	pno	Pname

1NF
will not allow
* composit
* multi value
* combination

2) Second Normal form (2^{nd} NF) :- an attribute of R/nal schema is called as Prime attribute if it is a member of some candidate key of R. if it not a "prime attribute" it is "nonprime."

* 2^{nd} NF is based on concept of full funal dependency a funal dependency $X \rightarrow Y$ is a full funal dependency if removal of any attribute A from X means that the dependency does not hold any more in such case otherwise the dependency $X \rightarrow Y$ is called as "partial dependency".

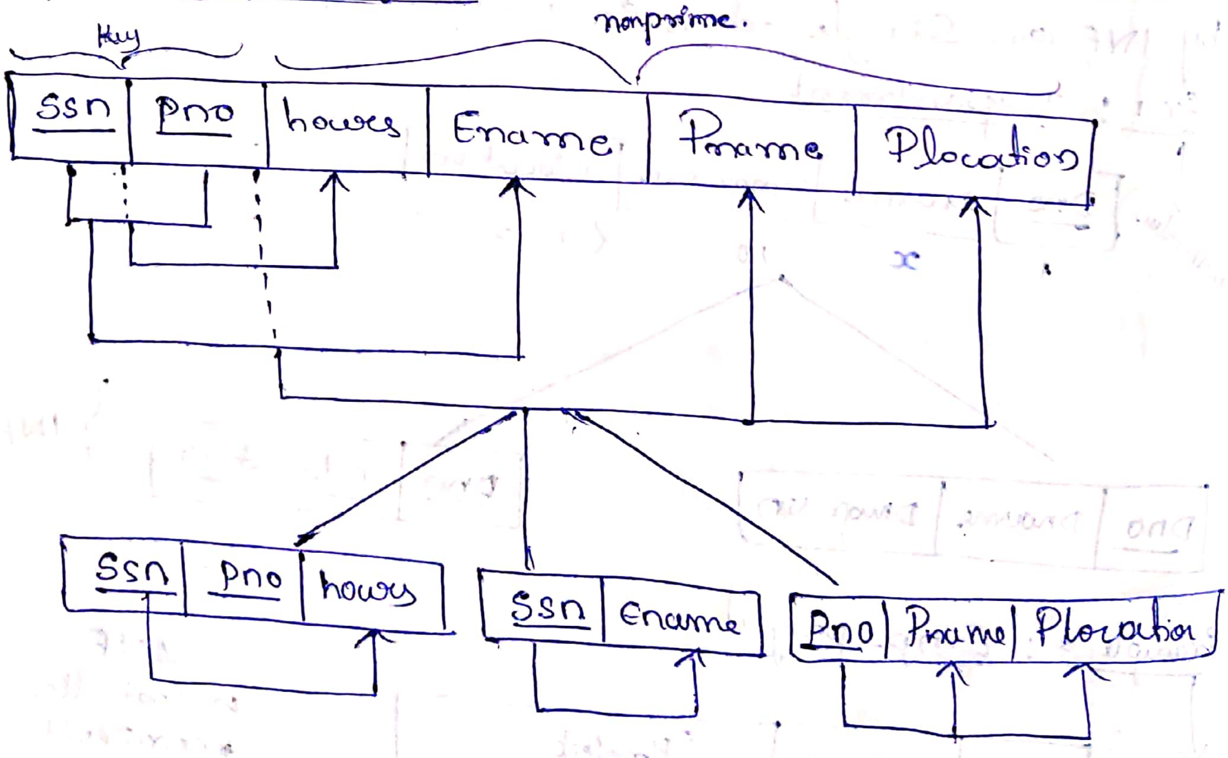
2^{nd} NF :- A R/nal schema have 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

Prime attribute → part of candidate key.

(or)

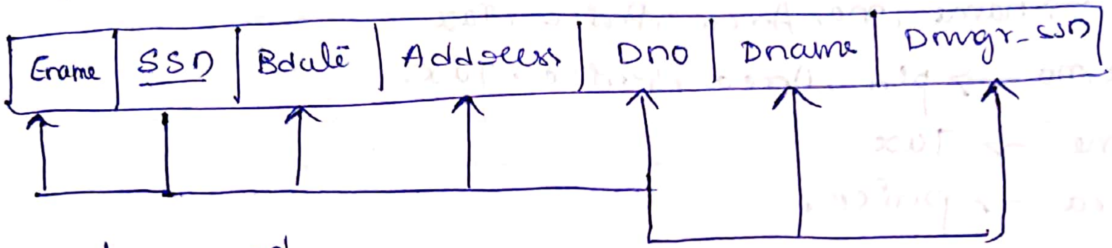
A R/nal schema are is in 2^{nd} NF if every non prime attribute A is R, is not partially dependent of any key of R.

Example : Emp_Project



3> Third NF (3rd) :- Based on the concept of transitive dependency. a functional dependency $X \to Y$ a R/n Schema R is a transitive dependency if there is a set of attribute Z that is a neither candidate key Nor a ~~subset of~~ any prime and both $X \to Z$, and $Z \to Y$ hold

Example : Emp-DEPT.

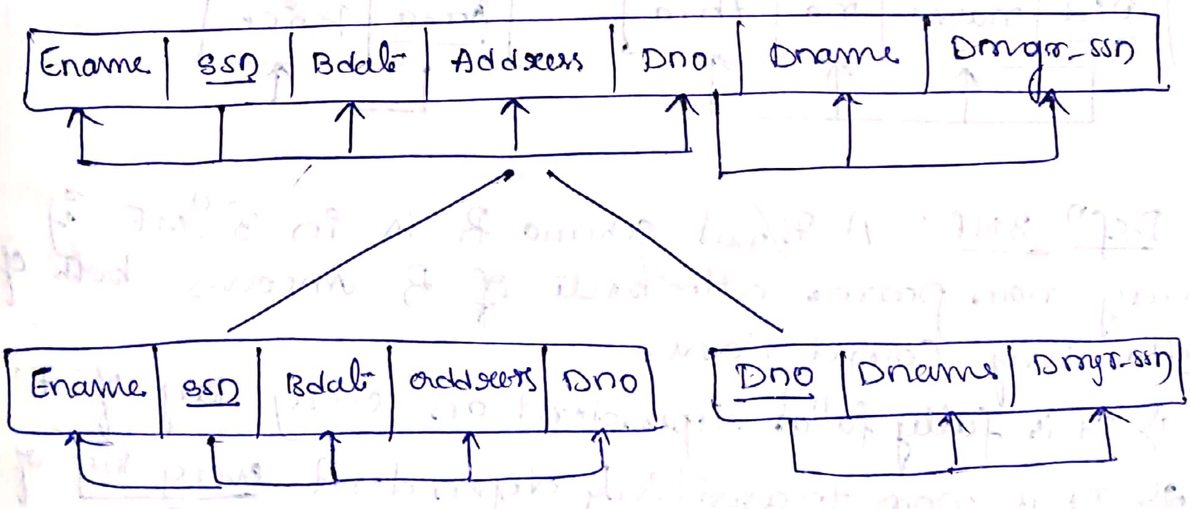


Statement for 3rd NF

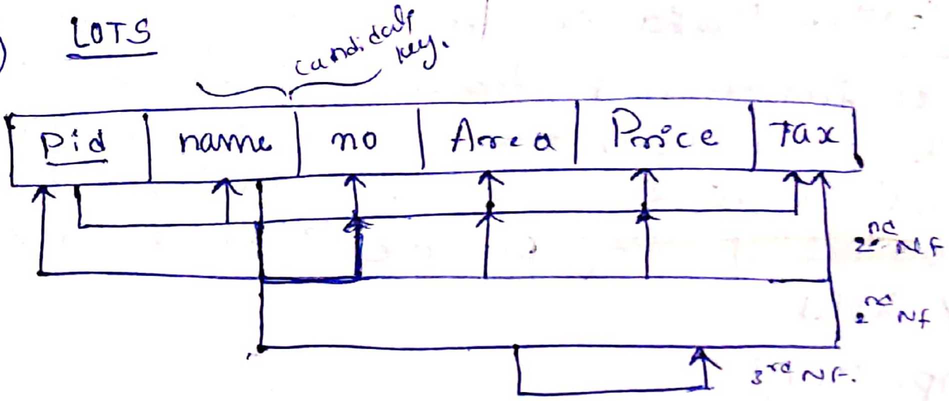
A R/n schema R is in 3NF if it Sastisfy 2NF and No non prime attribute of R is transitive-ly dependent on primary key (candidate key)

* A R/n schema R is in 3NF whenever a non trivil dependency $X \to A$ holds in R. either.

1> X is super key of R. (or)
2> A is prime attribute R.



(*) LOTS



Pid → name, no, Area, Price, Tax

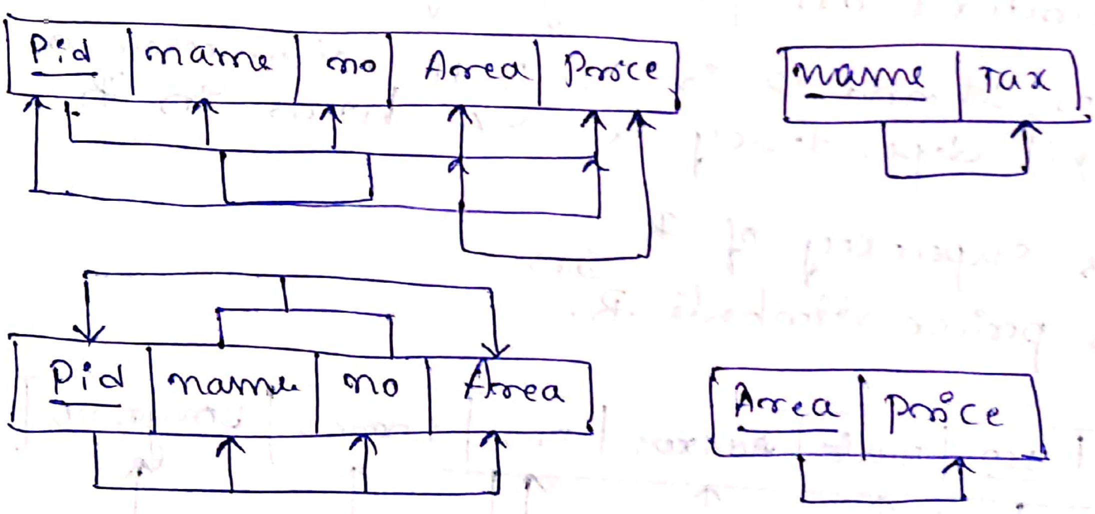
name, no → pid, Area, Price, Tax.

name → Tax

Area → price.

$name^+ = \{name, Tax\}$

$Area^+ = \{Area, Price\}$



* Defn 3NF :- A R/nal schema R is in 3rd NF if every non prime attribute of R meaning both of following conditions.

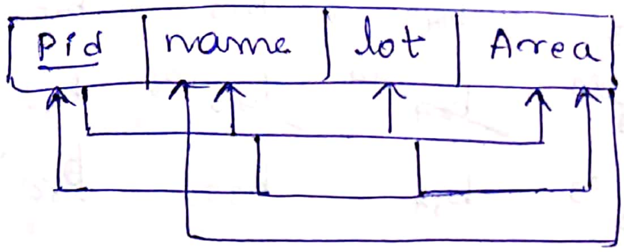
1) It is fully fun'al dependent on every key of R

2) It is non transitively dependent every key of R.

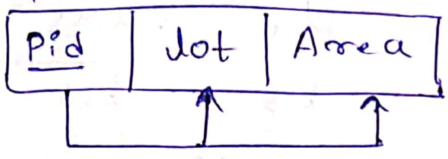
Boyce - codd Normal formal [BCNF]

A R/n schema R is BCNF if ** whenever a non trivial functional dependency $x \rightarrow A$ hold in R then α is super key of R. holds $x \rightarrow y$.

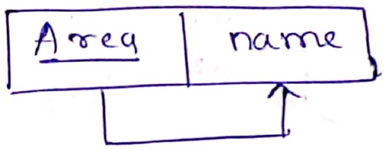
Example:



R_1



R_2



Testing binary decomposition for non-negative join property

Properties of R/nal decomposition

1> attribute preservation property: Given a R/nal schema $R = \{A_1, A_2 \ldots A_n\}$ given a universal R/nal schem given a set of dependencies (F) and $D = \{R_1, R_2 \ldots R_m\}$ attribute property say that

$\bigcup_{i=1}^{m} R_i = R$ this is called as attribute preservation prop

2> Dependency preservation property: given a set of F on R_i dependencies F on R projection of dependency $\Pi_{R_i}(F)$ where $R_i \subseteq R$ is the set of dependency $x \rightarrow y \in F^+$ such that $X \cup Y$ are in R_i then if D is said to dependency preserving

$\{\Pi_{R_1}(F) \cup \Pi_{R_2}(F) \ldots\ldots \Pi_{Rm}(F)\}^+ = F^+$

Non-negative / loss less join property

A decomposition $D = \{R_1, R_2 \ldots\ldots R_m\}$ are loss less join property with respect to set of F^n dependency F on R. if $*(\Pi_{R1}(r), \ldots\ldots \Pi_{Rm}(r)) = r$ where r is any valid R/nal state.

Given R = {ssn, ename, Pno, Pname, Ploc, hours}

R_1 = {ssn, ename}

R_2 = {Pno, Pname, ploc}

R_3 = {ssn, Pno, hours}.

F = { ssn → ename, Pno → Pname, ploc
ssn → pno } → Hours }

b_{ij}

	ssn	ename	Pno	Pname	Ploc	hours
R_1	$\cancel{b_{11}}$ a_1	$\cancel{b_{12}}$ a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	$\cancel{b_{23}}$ a_3	$\cancel{b_{24}}$ a_4	$\cancel{b_{25}}$ a_5	b_{26}
R_3	$\cancel{b_{31}}$ a_1	b_{32}	$\cancel{b_{33}}$ a_3	b_{34}	b_{35}	$\cancel{b_{36}}$ a_6

	ssn	ename	pno	Pname	ploc	hours
R_1	$\cancel{b_{11}}$ a_1	$\cancel{b_{12}}$ a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	$\cancel{b_{23}}$ a_3	$\cancel{b_{24}}$ a_4	$\cancel{b_{25}}$ a_5	b_{26}
R_3	$\cancel{b_{31}}$ a_1	$\cancel{b_{32}}$ a_2	$\cancel{b_{33}}$ a_3	$\cancel{b_{34}}$ a_4	$\cancel{b_{35}}$ a_5	$\cancel{b_{36}}$ a_6

R = {ssn, ename, Pno, Pname, ploc, hours}

R_1 = {ename, ploc}

R_2 = {ssn, Pno, hours, Pname, ploc}

F = { ssn → ename, Pno → Pname, ploc
ssn → Pno } → Hours }.

	ssn	ename	Pno	Pname	ploc	hours
R_1	b_{11}	$\cancel{b_{12}}$ a_2	b_{13}	b_{14}	$\cancel{b_{15}}$ a_5	b_{16}
R_2	$\cancel{b_{21}}$ a_1	b_{22}	$\cancel{b_{23}}$ a_3	$\cancel{b_{24}}$ a_4	$\cancel{b_{25}}$ a_5	$\cancel{b_{26}}$ a_6
~~R_3~~	~~b_{31}~~	~~b_{32}~~	~~b_{33}~~	~~b_{34}~~	~~b_{35}~~	~~b_{36}~~

END unit 4

Equivalence of FD

Eg (1)

$X = \{A \to B,\ B \to C\}$ | $Y = \{A \to B,\ B \to C,\ A \to C\}$

check

$X \supseteq Y$

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

$B^+ = \{BC\}$

$X \supseteq Y$

$Y \supseteq X$

hence $X \equiv Y$

check

$Y \supseteq X$

$A^+ = \{ABC\}$

$B^+ = \{BC\}$

Eg (2)

$X = \{AB \to CD,\ B \to C,\ C \to D\}$

$Y = \{AB \to C,\ AB \to D,\ C \to D\}$

Check $X \supseteq Y$

$(AB)^+ = \{ABCD\}$

$B^+ = \{BCD\}$

$C^+ = \{CD\}$

check $Y \supseteq X$

$(AB)^+ = \{ABCD\}$

$B^+ = \cancel{\{BCD\}}\ \{B\}$

$C^+ = \{CD$

$X \supseteq Y$ but $Y \not\supseteq X$

$\therefore X \not\equiv Y$

Scanned with OKEN Scanner

Trivial FD.

In trivial FD, a dependent is always a subset of the determinant.

Eg: if $X \rightarrow Y$ is the subset of X, then it is called trivial functional dependency.

Eg:

roll_no	name	age
42	abc	17
43	pqr	18
44	xyz	18

Here {roll_no, name} → name is a trivial FD.

IIIly - roll_no → roll_no is also trivial FD.

Non-Trivial FD

In non-trivial functional dependency, the dependent is strictly not a subset of the determinant.

i.e if $X \rightarrow Y$ then Y is not a subset of X.

(roll_no) → name. is non-trivial FD.

{roll_no, name} → age is non-trivial FD

3. Multivalued Functional Dependency.

In multivalued functional dependency, entities of the dependent set are not dependent on each other.

ie if $a \rightarrow \{b, c\}$ exists then there is not FD between b & c.

ie → $b \rightarrow c$ or $c \rightarrow b$. doesnot exits

Scanned with OKEN Scanner

roll_no → {name, age} is MFD because
name → age or age → name doesnot exist.

4. Transitive FD
In transitive FD, dependent is indirectly dependent on determinant.

ie if a → b & ~~b~~ b → c then a → c.

if roll-no → name & name → age.
then roll_no → age.

5. Fully ~~functional~~ FD
In fully functional dependency an attribute or a set of uniquely determines another attribute or set of attributes.
If a relation R has attributes X, Y, Z with the dependencies X → Y and Y → Z, then it states that these dependencies are fully functional

6. Partial ~~fun~~ FD
In partial FD a non key attribute depends on a part of the composite key, rather than the whole key. If a relation R has attributes X, Y, Z where X and Y are the composite key and Z is non key attribute, then X → Z is a partial FD.

ie if XY → Z then X → Z is PFD

2NF.

customer_id	store_id	location
1	1	Delhi
1	3	Mumbai
2	1	Delhi
3	2	Bangalore
4	3	Mumbai

in 1 NF but not in 2NF.

because store_id → location.

while candidate key = (customer_id, store_id)

→ location becomes non prime attribute

customer_id	store_id
1	1
1	3
2	1
3	2
4	3

store_id	location
1	Delhi
2	Bangalore
3	Mumbai

Table to be 2 NF.

① It has to be in 1 NF.

② All the non-prime attributes should be fully functionally dependent on candidate key.

Closure functional dependency

① R(ABCD)

Prime attribute = A
Non prime attributes = B, C, D

R = {A → B, B → C, C → D}

A^+ = BCDA → A → B, B → C, C → D — transitive rule

B^+ = BCD → B → C, C → D

C^+ = CD → C → D

D^+ = D

candidate key {A}

B → A, C → B, C → A — canot dependent

A is a candidate key

B is dependent totally

Attributes used in making primary is called prime attribute

② R(ABCD)

FD = {A → B, B → C, C → D, D → A}

A^+ = ABCD

B^+ = BCDA

C^+ = CDAB

D^+ = DABC

Prime attributes = ABCD

Non prime attributes = φ

(AB^+) = ABCD // But cannot be a candidate key because candidate key has to be minimal

③ R(ABCDE)

FD = {A → B, BC → D, E → C, D → A}

E^+ = {EC} B^+ = B × D^+ = {DAB}

AE = AEBCD // if A or E is present on right side then LHS becomes the prime attribute

↓

DE = DEABC

↓

BE^+ = {BECDA}

CE^+ = {CE·}

1 1 1 1 1 - 15
1 0 1 1 - 11
1 1 0 1 1 - 27
1 0 0 0 - 8
1 1 0 0 - 12

0 — 8 - 12
1 — 15, 11, 27

② R(A, B, C, D, E, F).

FD = {A → B, B → C, C → D, D → E}

AF^+ = {ABCDEF}

* A & F are not present on right hand side of any FD, hence there cannot be anymore CD's

∴ Non prime attributes = BCDE

Now A → B A → C
F → F.

because ~~part~~ proper subset of CD can determine the non prime the given Relation is not in 2NF

③ R(A, B, C, D)

FD = {AB → CD, C → A, D → B}.

$\underbrace{ABCD}^+$ = {ABCD}
superkey.

AB^+ = {ABCD}

A^+ = {A}

B^+ = {B}

∴ AB → candidate key.

CB^+ → {

AD^+ = {

B^+ = {B}

C^+ {CA} ∴ CB is also candidate key

D^+ = {DB} ∴ AD is also candidate key

B^+ = {A}

// No non prime attributes ∴ the given relation is in 2NF.