5/9/1

Q Find the names of sailors who have reserved both a red & a ~~green~~ ~~blue~~ green boat.

~~Select s.name from sailor S where~~
~~s.sid EXISTS (select ~~ R.bid~~
~~from Reserves R where R.bid~~
~~EXISTS (select b.bid from~~
~~Reserves R where~~

~~Select S.name from sailor S where~~
~~exists (select * from boat~~
~~B, reserves R where r.sid = s.sid &~~
~~b.color = red and green)~~

Select S.name, s.sid from sailor S,
boat B, reserves R where r.sid =
s.sid & b.bid = r.bid & b.color = red
EXISTS (select S.sid from sailor S,
where r.sid = s.sid & b.bid =
r.bid & b.color = ~~blue~~ green)

⇓

22
31
64
95

22
31
74

⇒

22
31

output

22 → 101
→ 102 ~~R~~ b.bid ≠

Q Find the names of sailors who have reserved all boats.

Select S.~~name~~ id from sailor S where ~~S.sid~~ NOT EXIST (Select b.bid from boats B where NOT Exists (Select R.bid from reserves R where r.bid = b.bid & R.sid = S.sid)).

7/9/18

Emp-Dept (Eid, Name, DOB, Phno, Dno, Dname, Mgid)

When we combine two tables:-
There will be anomalies.
1. Insertion anomaly.
2. Deletion anomaly.
3. Updation anomaly

} Due to the redundant information.

→ When we are doing join on table having no primary key, then it will give false ~~rows~~ tupples.

Normalization (To remove false tupples).

Now to study normalization, we should know functional dependency.

Functional dependency:

If we have a relation R having attribute X and Y, and two tupples t_1 and t_2 then functional dependency is written as $X \longrightarrow Y$ if $t_1 \cdot X = t_2 \cdot X$ then $t_1 \cdot Y = t_2 \cdot Y$. (This means functional dependency holds).

If $t_1 \cdot X = t_2 \cdot X$ and then $t_1 \cdot Y \neq t_2 \cdot Y$ (This means functional dependency doesn't hold.

Eg:-

Eid $\longrightarrow$ Ename

Pno $\longrightarrow$ Pname, Ploc

Eid, Pno $\longrightarrow$ Hours.

Armstrong Rules

① If $Y \subseteq X$

$X \longrightarrow Y$ holds

(Reflex rules).

② $X \longrightarrow X$

(trivial dependency)

We have to find non-trivial dependency.

③ Transitive rule:

$X \rightarrow Y$
$Y \rightarrow Z$
then $X \rightarrow Z$

eg:- If every dept. has a unique manager with mgid.
And every manager has a unique phon no, then we can uniquely identify the Dept with Mg Phno

④ If $X \rightarrow YZ$
then $X \rightarrow Y$
$X \rightarrow Z$ } holds.

Decomposition rule

⑤ Union Rule.

$X \rightarrow Y$
$X \rightarrow Z$

$X \rightarrow YZ$ holds.

⑥ Augmented Rule

If $X \rightarrow Y$

then $XZ \rightarrow YZ$

⑦ Pseudo Transitive rule

$$\frac{\begin{array}{c} X \rightarrow Y \\ WY \rightarrow Z \end{array}}{WX \rightarrow Z}$$

Closure

Set of f^n dependencies, that contains all the given functional dependency and the inferred functional dependency.

[Eid]$^+$ = This means what is the closure of Eid.

[Eid]$^+$ = { Eid, Ename }

[Pno]$^+$ = { Pno, Pname, Ploc }

$[Eid, Pno]^+ = \{ all \}$.

$= \{ Ename, Eid, Pno, Pname, Ploc, Hours \}$.

Super Key is that in whose closure all the attributes will come

Q (i) R(ABCD)

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow D$.

$[A]^+ = \{A, B, C, D\}$ { According to Transitive Rule }

$[B]^+ = \{B, C, D\}$

$[C]^+ = \{C, D\}$.

So A becomes the super key.

Q. (ii) $AB \rightarrow C$

$C \rightarrow D$

$B \rightarrow E$.

A and B are the subset of superkey. If these two subsets doesn't contain all the attributes then $[AB]^+$ will become the Candidate key.

$[A]^+ \rightarrow \{A\}$

$[AB]^+ \rightarrow \{A, B, C, D, E\}$ → Super key

$[B]^+ \rightarrow \{B, E$

$[C]^+ \rightarrow \{C, D$

Candidate Key → The single attributes of the super key (subsets of the super key). Then if all the attributes does not comes ~~in the candidate key then~~ in the subset then it is the candidate key.

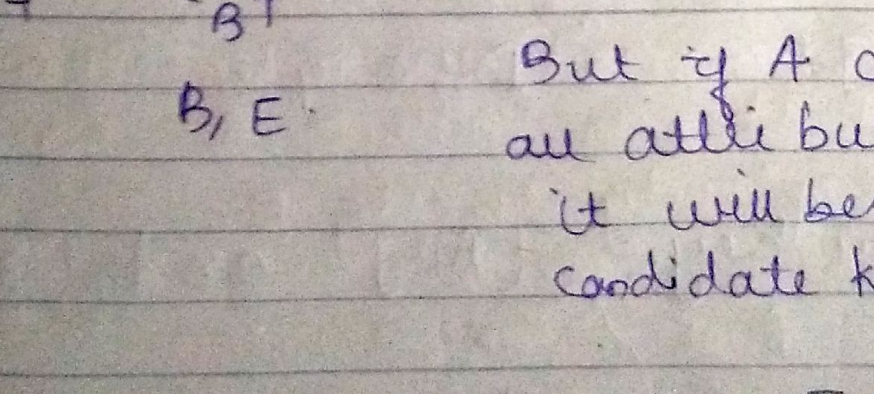
$[AB]^+$

A^+ : A

B^+ : B, E

So AB is the candidate key. But if A contains all attributes then it will become the candidate key.

Q



$AB \rightarrow C$

$C \rightarrow D$

$B \rightarrow AE$.

$\Downarrow$

$AB \rightarrow C$

$C \rightarrow D$

$\left.\begin{array}{l} B \rightarrow A \\ B \rightarrow E \end{array}\right\}$ Decomposition rule.

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

$[A]^+ = \{A\}$

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

$[C]^+ = \{D, C\}$.

Super Key = $[AB]^+$ and $[B]^+$.

$[AB]^+$ — Super Key

$[A]^+$ $[B]^+$

$\downarrow$

Candidate Key

Q $R(ABCDEF)$.

$BC \longrightarrow ADEF$.
$A \longrightarrow BCDEF$.
$B \longrightarrow F$.
$D \longrightarrow E$.

$[A]^+ = \{A, B, C, D, E, F\}$ = Super Key & candidate Key

$[B]^+ = \{B, F\}$

$[C]^+ = \{C\}$

$[BC]^+ = \{A, B, C, D, E, F\}$ = Super Key & Candidate Key

$[D]^+ = \{D, E\}$

$[BC]^+$ (Candidate Key).

$[B]^+$ $[C]^+$

$\{B, F\}$ $\{C\}$.

Q $R(ABCD)$
$AB \to C$
$C \to DE$
$BE \to F$ (x)
$F \to G$ (x)

$C \to D$
$C \to E$ (x)

All these which contain E, F, G will be cancelled because this are not our att.)

$AB \rightarrow C$

$C \rightarrow D$.

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

$[AB]^+ = \{C, D, A, B\}$ (candidate of super key)

$[B]^+ = \{B\}$.

$[C]^+ = \{C, D\}$.

Q. $R(ABCDE)$

$AB \rightarrow C$

$C \rightarrow D$

$D \rightarrow A$

As E is not given in dependency but it is our attribute, so we will explicity write E.

$ABE \rightarrow C$

$CE \rightarrow D$

$DE \rightarrow A$

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

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

$[BE]^+ = \{B, E\}$

$[CE]^+ = \{C, E, D, A\}$

$\underline{[BCE]^+} = \{B, C, E, D, A\}$

$[DE]^+ = \{D, E, A\}$.

We have to calculate closure for all the subsets of the ABCDE to find the candidate key.
That is $2^5 = 32$ closures we have to find. (But we don't have to write all.)

This is applicable for all the previous questions.