2. ~~select max(balance) as max_balance~~
~~from balance;~~

select br_name, ~~maximum~~ max (sum_b)
from (select br_name, sum ~~max~~ (balance)
from account
group by br_name)
~~as br~~
as br_nam (br_name, sum_b ~~sum~~ ~~maximum~~));

~~OR~~

Another Method:

with max_balance(value)
as select max(balance)
from account
~~select as~~

select account_no from
account max_balance
where account.balance = max_balance.value;

To Create the virtual tables permanently —

create view tot_branch ~~balance~~ as
select br_name, sum(balance) as tot_balance
from account
group by br_name;

Functional dependency and Normalization

Relational Database

$\boxed{\text{Requirements}}$ R = { br_name, br_city, assets, cust_id, cust_name, cust_street, cust_city, loan_no, amount, account_no, balance }

branch (br_name, br_city, assets)

customer (cust_id, cust_name, cust_street, cust_city)

loan (loan_no, br_name, amount, cust_name)

account (account_no, br_name, balance, cust_name)

- Redundancy
- Modification anomaly
- Insertion ~~anomal~~ anomaly
- Deletion anomaly

Dependencies:

br_name → br_city

br_name → assets

cust_id → cust_name

loan_no → amount

- br_name determines br_city
  OR
  br_city depends on br_name

Functional Dependency: A functionally determines B $\boxed{A, B \subseteq R}$

Reflexivity (Trivial Dependency)

br_name → br_name

~~br_name → br_city~~

br_name, br_city → br_city

• Any set of attributes can determine its subsets.

br_name, asset → br_city, asset (Augmentation)

br_name → br_city
br_city → br_state, then br_name → br_state (Transitivity)

R

X, Y ⊆ R

Armstrong's Axioms

1. $X \to Y$ is trivial if $Y \subseteq X$ (Reflexivity)
2. $X \to Y$, then $XZ \to YZ \mid (Z \subseteq R)$. (Augmentation)
3. $X \to Y$, $Y \to Z$, then $X \to Z$ (Transitivity)

Secondary Rules

1) If $X \to Y$ and $X \to Z$, then $X \to YZ$ (Union Rule)
2) If $X \to YZ$, then $X \to Y$ and $X \to Z$ (Decomposition Rule)
3) If $X \to Y$ and $WY \to Z$, $WX \to Z$ (Pseudo Transitivity)

Ex: $R(A, B, C, D, E, F)$

$F = \{A \to BC,\ B \to E,\ CD \to EF\}$

Show that $AD \to F$.

~~∴ A → BC,~~

$F' \{A \to B,\ A \to C,\ B \to E,\ A \to E,\ A \to BC$
$CD \to E,\ CD \to F,\ CD \to EF\}$

• F and F' are equivalent

$A \to C$,

$CD \to F$

∴ $AD \to F$ (By Pseudo Transitivity)
[Proved]

F.D cover

- F' is a cover of F,
- F is a cover of F'

One satisfies means other also satisfies.

- We will be interested in finding the minimal cover.

Example:

R(A, B, C, D)

F = {A → BC, B → C, A → B, AB → C, AC → D}

Step I:

= {A → B, A → C, B → C, A → B, AB → C, AC → D} [Make attributes on the right side a single attribute]

Step II: Make attributes on the left a single attribute.

AC → D

A → C

AC → C

~~AC → CC~~

⇒ ~~AC → C~~

~~AC → A~~

A → AC

$\boxed{A \to D}$

Also, AB → C can be written as $\boxed{B \to C}$

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

= ~~{A → B, A → C, A → D}~~

= {A → B, B → C, A → D}

Minimal cover // Canonical Cover

- Minimal cover may not be unique.

Scanned by CamScanner

$X \rightarrow$ Set of attributes

$X^+ \rightarrow$ closure of X

$X \subseteq R$

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

$F = \{A \rightarrow BC, ~~BC~~ B \rightarrow C, A \rightarrow B, AB \rightarrow C, AC \rightarrow D\}$

$\{A\}^+$ $\quad \frac{A \rightarrow A}{X = \{A\}}$

~~$A \rightarrow A$~~

~~$X =$~~

- Find dependencies of the form $\alpha \rightarrow \beta$ where $\alpha \subseteq X$.
- If possible to find, change $X = X \cup \beta$

$\underline{A \rightarrow BC}$

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

$\underline{AC \rightarrow D}$

$X = \{A, B, C, D\} \leftarrow$ Closure of A $(\{A\}^+)$

$\therefore$ A is a superkey as its closure contains all the attributes. (i.e. $X = R$)

$\{B\}^+$ $\quad \underline{B \rightarrow B}$

$X = \{B\}$

$\underline{B \rightarrow C}$

$X = \{B, C\} \rightarrow$ Closure of B $(\{B\}^+)$

Candidate Key

1) $X^+ = R$

$X \rightarrow$ is a superkey

2) $\exists$ no $Y (Y \subset X)$ and $Y^+ = R$

1. Find the candidate keys —

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

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

$\{C\}^+$ $\quad \underline{C \to C}$

$X = \{C\}$

$\underline{C \to A}$

$X = \{C, A\}$

$\{D\}^+$ — $\underline{D \to D}$

$X = \{D\}$

$\underline{D \to B}$

$X = \{D, B\}$

$\{AB\}^+$ $\quad \underline{AB \to AB}$

$X = \{A, B\}$

$\underline{AB \to C}$

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

$X = \{A, B, C, D\}$

$\{AD\}^+$

$X = \{A, D, B, C\}$

$\{CD\}^+$

$X = \{C, D, A, B\}$

$\{BC\}^+$

$\{AB\}^+ = A^+ \cup B^+$ ✗ Not Necessarily True

~~Then AB~~

If $A^+ \cup B^+ = R$ ~~Then~~, then AB is a superkey.

$AB^+ = R$

<u>Fully Functional Dependent</u>

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

- $A \to B$

B fully depends on A

- $AC \to B$

B is not fully dependent on AC.

- $\alpha \to \beta$, ~~is not fully~~ β is fully dependent on α, if β is not dependent on any proper subset of α.

<u>Normal Form</u>

1) Good Form

2) Decomposed Relations

* good form
* lossless-join decomposition

[if we break a relation into smaller ones, then it should by lossless, ie. when we join them back, we should get back the original relation

R → R_1, R_2

① $R_1 \cup R_2 = R$

② $\forall r \in R$

$r_1 \in R_1$, $r_2 \in R_2$

$r_1 \bowtie r_2 = r$

$\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$

R_1
A
α
β

R_2
B
1
2

R	
A	B
α	1
α	2
β	1

Need for Normalization—
1) Reduce Redundancy
2) Lossless join decomposition
3) Dependency Preservation

③ Dependency Preservation

I NF

1) ~~Records~~ Attribute values should be atomic.

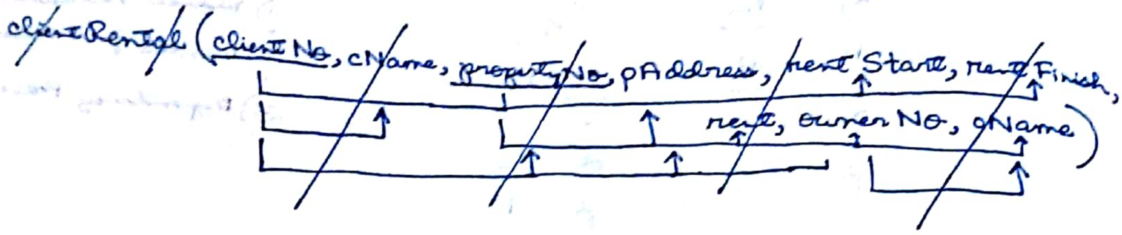
Roll	Subcode	Marks
r1	S1	m1
	S2	m2
	S3	m3
	S4	m4
	S5	m5

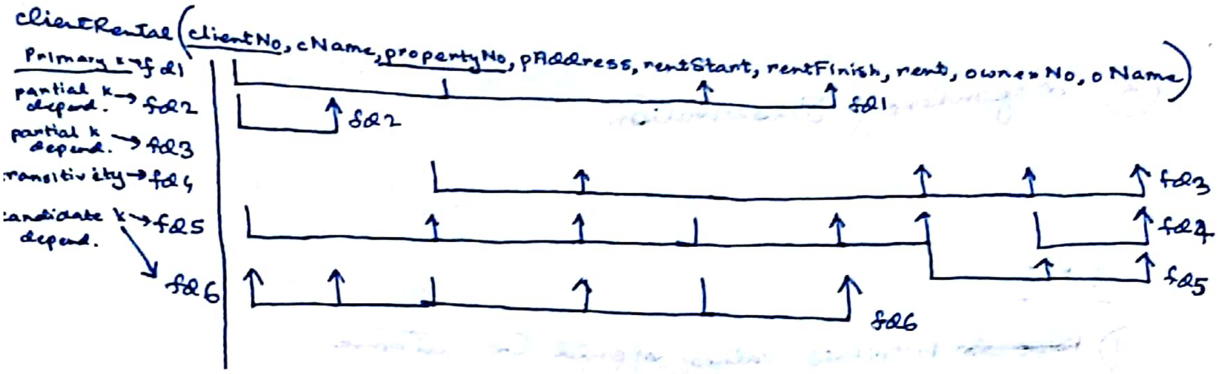
Not Allowed (Grouping Not Allowed)

UNF (Unnormalised form ~~Normal~~)

I NF

Roll	Subcode	marks
r1	S1	m1
r1	S2	m2
r1	S3	m3
r1	S4	m4
r1	S5	m5





fd1: clientNo, propertyNo → rentStart, rentFinish

fd2: clientNo → cName

fd3: propertyNo → pAddress, rent, ownerNo, oName

fd4: ownerNo → oName

fd5: clientNo, rentStart → propertyNo, pAddress, rentFinish, rent, ownerNo, oName

fd6: propertyNo, rentAddress → clientNo, pAddress, cName, rentFinish

2NF: A relation is in 1 NF and every non-~~key~~ primary key attribute is fully functionally dependent on the primary key, and candidate key.

Definition in Korth: A relation R is in 2 NF with respect to a set of functional dependencies F, if $\forall\, \alpha \to \beta \in F$,

1) $\alpha \to \beta$ is trivial, ie. $\alpha \supseteq \beta$

2) $\alpha \to \beta$, α cannot be a part of a primary key

Decomposition for 2NF:

client (clientNo, cName)

property (propertyNo, pAddress, rent, OwnerNo, oName)

clientrate (clientNo, propertyNo, rentStart, rentFinish)

3NF

The relation is in 1NF and 2NF, and ~~there is~~ no ~~transitive dependency~~ non-primary key attribute is transitively dependent on the primary key.

Decomposition to 3NF:

break the property table —

property (propertyNo, pAddress, rent, ownerNo)

owner (ownerNo, oName)

Definition in Korth:

A relation R is in 3NF with respect to a set of functional dependencies F, if $\forall \alpha \to \beta \in F$, one of the following is satisfied —

1) $\alpha \to \beta$ is trivial $\alpha \supseteq \beta$
2) $\alpha \to \beta$, α is ~~in~~ primary key / ~~super key~~
3) $\beta - \alpha$ is a part of candidate key.

BCNF:

R wrt F, $\forall \alpha \to \beta \in F$, one of the following is satisfied —

1) $\alpha \to \beta$ is trivial $\alpha \supseteq \beta$
2) $\alpha \to \beta$, α is a primary key / super key.

- The above tables are in BCNF as well.

Scanned by CamScanner

- A decomposition R to R_1, R_2 is lossless if either

  1) $R_1 \cap R_2 \rightarrow R_1$

  2) $R_1 \cap R_2 \rightarrow R_2$

ClientRental

(clientNo, propertyNo, cName,

Normalizing to 3 NF

Algo:

$$\underset{\alpha}{\text{ownerNo}} \rightarrow \underset{\beta}{\text{oName}}$$

Create a relation $\rightarrow R_1(\alpha \cup \beta)$

Original Relation $\rightarrow R = R - (\beta - \alpha)$

- A relation R is in BCNF with respect to F (a set of fds defined over R) is also in 3-NF.

Example of a relation in 3NF but not in BCNF —

~~A, B, C~~

This case arises generally when the primary key is a composite key

$\beta - \alpha$

$R(\underline{A}, \underline{B}, C, D)$

$F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow B\}$

Not in BCNF

$B - C = B$, which is a part of a candidate key.

1. $R(\underline{A}, \underline{B}, C, D, E)$

$F = \{AB \to CD,\ ABC \to E,\ C \to A\}$

Q1: Find ~~the~~ two candidate keys. AB, BC

Q2: What is the normal form of R? 3NF, Not in BCNF

$X = \{ABC\}$ Sup: ABC

~~AB → CD~~ AB

$X = \{AB\}$

ABCD

$\{D\}$ BCADE

BCAD

Simply: ~~F1: AB → CD~~

~~F2: ABC → E~~

~~F3: C → A~~

$AB \to C$
$AB \to D$
$ABC \to E$
$C \to A$

$F' = \{AB \to C,\ AB \to D,\ AB \to E,\ C \to A\}$

$F'' = \{AB \to CDE,\ C \to A\}$

$\begin{pmatrix} AB \to C \\ AB \to D \\ AB \to E \end{pmatrix}$

BD Scheme(br_name, loan_no, amount, account_no, balance, cust_name)

F = { loan_no → amount,
loan_no → br_name,
account_no → balance,
account_no → br_name }

R1 (loan_no, amount)

R2 (loan_no, br_name)

R3 (account_no, balance)

R5 (account_no, br_name) } This can be included, or may not be included. Both are okay.

R4 { loan_no, account_no, ~~balance~~ cust_name }

- While decomposing a relation in BCNF, it is not necessary that functional dependencies remain preserved.

Multivalued Dependency

α	β	$R-(\alpha+\beta)$
I1	CSE	Saltlake
I1	IT	Siliguri
I1	CSE	Siliguri
I1	IT	Saltlake

$\alpha \twoheadrightarrow \beta \Rightarrow \alpha \twoheadrightarrow R-\alpha-\beta$ → Replace all the β by $(R-\alpha-\beta)$.

If ① $t_1(\alpha) = t_2(\alpha) = t_3(\alpha) = t_4(\alpha)$, then

② $t_3(\beta) = t_1(\beta)$

③ $t_3(R-\alpha-\beta) = t_2(R-\alpha-\beta)$

④ $t_4(\beta) = t_2(\beta)$

⑤ $t_4(R-\alpha-\beta) = t_1(R-\alpha-\beta)$

	α	β	$R-\alpha-\beta$
t_1	$a_1, a_2, \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a\ a_2 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a\ a_2 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a\ a_2 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

- $\alpha \rightarrow \beta$, then $\alpha \twoheadrightarrow \beta$ → Always Holds

4 NF

R is in 4 NF wrt D (dependencies), if $\alpha \twoheadrightarrow \beta \in D^+$.

1) $\alpha \twoheadrightarrow \beta$ Trivial

2) α is a superkey

Conversion to 4NF:

$R(A, B, C, G, H, I)$

ABCG,H

$F = \{ A \twoheadrightarrow B,$

$B \twoheadrightarrow HI$

$CG \twoheadrightarrow H \}$

ACG is the primary key

a) $R_1 = (A, B) \longrightarrow$ in 4NF

$R_2 = (A, C, G, H, I) \longrightarrow$ not in 4NF

b) $R_3 = (C, G, H) \rightarrow$ in 4NF

$R_4 = (A, C, G, I) \rightarrow$ not in 4NF

$A \twoheadrightarrow B$

$B \twoheadrightarrow HI$

$\therefore A \twoheadrightarrow HI$

$\therefore A \twoheadrightarrow H$ and $A \twoheadrightarrow I$.

c) $R_5 = (A, I) \longrightarrow$ Not in 4NF

$R_6 = (A, C, G) \longrightarrow$ Not in 4NF

- For 4NF Conversion, it is not necessary that dependencies are preserved.