Function Dependency

Emp
Eid | Ename | Eaddr | Eid → Ename, Eid → Eaddr

Date 06/03/19
Page 24

function dependency

Emp
Eid | Ename | Eaddr | Dname | D-loc
Eid → Ename, Eaddr, Dname
Dname → D-loc

Dname → Dloc (Dname identifies D-loc)
Eid → Ename
Eid → Eaddr (or) Eid → Ename, Eaddr

• Armstrong Axioms •

① **Reflexivity** - If A is a set of attr and B is a subset of A
A → B

② **Augmentation** - If A → B, C is any attr then
AC → BC

③ **Transitivity** - If A → B, B → C then A → C
D → B

④ **Pseudo-Transitivity** - If AB → C and [illegible] then
AD → C

⑤ **Union** - If A → B and A → C then A → BC

⑥ **Decomposition** - If A → BC then A → B and A → C

* F^+ Attribute Closure $[X^+]$

Let R be a relation R = { A, B, C, G, H, I } and the following set of FD's

F = A → B
A → C
CG → H
CG → I
B → H

F^+ = A → BC [Union A→B, A→C]
CG → HI [Union CG→H, CG→I]
A → H [Transitivity A→B, B→H]

CG → H
A → C
AG → H

Find A^+, CG^+, AB^+

~~$A^+ = \{A\}, \{AB\}, \{AC\}, \{AH\}$~~

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

Date / /
Page 25

$= \{ABCH\}$

$CG^+ = \{CGHI\}$

$AB^+ = \{AB\}$
$= \{ABCH\}$

$AG^+ = \{AG\}$ → Since we get whole R then AG will become the primary key of relation.
$= \{AGBCHI\}$

7/3/19

① Given a set of FD's = { A → B, ABCD → E, EF → G }
And we have ACDF → G

② F_1 : A → B, AB → C, D → AC, D → E
F_2 : A → BC, D → AE are the 2 sets equivalent

①
A → B
ABCD → E
EF → G

ABCDF → EF (Augmentation F)
ABCDF → G (Reflexivity)
AACDF → G (CR)
ACDF → G

ABCD → E
EF → G
ABCDF → G (Pseudo transitivity)
AACDF → G (Reflexivity)
ACDF → G.

(2) $A \to B$
$AB \to C$

Date 7/3/18
Page 26

$D \to AC$
$D \to E$

$AB \to C$ (~~Reflexivity $A \to B$~~)
~~$AB \to C$~~ $A \to C$
~~$AC \to BC$~~ $A \to B$ $A \to BC$
$D \to AE$

$D \to A$
$D \to E$

$D \to AC$ (Refl
~~$D \to AA$~~ $D \to A$
$D \to A$ $D \to C$
$D \to A$. $D \to AE$

$A \to B$ ✓
$A \to C$
$D \to A$
$D \to E$ ✓

$D \to C$

$D \to AC$

$AB \to C$

$AB \to C$

$AB^+ = \{AB\}$
$= \{ABC\}$

$A \to B$
$A \to C$
$D \to A$
$D \to E$
~~$AB \to C$~~
~~$AB \to BC$~~
~~$BC \to BC$~~

AB
$A \to BC$
$\to A \to B$
$A \to C$
$A \to B$ (Augment C)
$AC \to BC$
$AA \to BC$ (Reflexivity)
$A \to C$

Consider a relation R having attr. $R(A,B,C,D,E,F,G,H,I,J)$
FD's $\Rightarrow$ $AB \to C$
$A \to DE$
$B \to F$
$F \to GH$
$D \to IJ$

Date __/__/__
Page ~~27~~

(1) What is the key of R.
(2) For a given relation R key attr. will depends

$R(A,B,C,D,E)$
$AB \to C$
$CD \to E$ is AB a candidate key
$DE \to B$
is ABD is a candidate key

$AB^+ = \{AB\}$
$= \{ABDEIJ\}$

$ABE^+ = \{ABC\}$
$= \{ABDEIJ\}$

$ABD^+ = \{ABD\}$ ABD
$= \{ABDEIJ\}$

left	middle	right
L	M	R
A,B,E	DF	C,E,G,H,I,J
B		

✗ never contain a key

$AB^+ = \{AB$
~~ABD~~ $ABF^+ =$
$A^+ = \{ADIJ\}$
$B^+ = \{BFGH\}$
$D^+ = \{DIJ\}$
$F^+ = \{FGH\}$

$ABF^+ = \{ABF\}$
$= \{ABFDEGHIJ\}$

$AB^+ = \{$

$AB^+ = \{ABDEFGHIJC\}$

AB is the key of R

Date 1/1 Page 28

~~AD+ = {ADEIJ}~~ $AD^+ = \{ADEIJ\}$ $AF^+ = \{DEFGH\}$

$AB \to C$
$CD \to E$
$DE \to B$

L	M	R
A, BD	CE	ADE

$AB^+ = \{AB\}$
(No) $= \{ABC\}$

$ABD^+ = \{ABD\}$
(yes) $= \{ABCDE\}$
AB

$ABC^+ = \{ABC\}$
$= C$

AB {AB, AC, AD, AE}, {ABD}, ABE, ~~AB~~
AE

$ABE = \{ABE\}$

~~BC~~

$ADC = \{ADC\}$
$= \{ADCEB\}$

ABD, ADC, ADE,

8/8/19

Second Normal Form (2NF)

2NF deals with partial dependency.

A relation R is said to be in 2NF if it is INF and there is no partial dependency.

Partial Dependency

① ~~If~~ PD may exist whenever a relation R has a composite primary key. Date 8/3/19 Page 29
A composite key is a primary key having more than one attribute.

② If is required that every non-prime attribute in relation R is fully functionally dependent on ~~both~~ the primary key R.
for eg $XY \to Z$ & also $X \to Z$ exist. ie the role of y is exhaustive.

ORDER_NO	BOOK_TITLE	QUANTITY	UNIT_PRICE
1	~~DBMS~~	2	450/-
1	Software Engg	1	300/-
1	Theory cycle	3	490/-
2	DBMS	1	450/-
2	Human Values	1	150/-
3	CO	1	390/-
3	Theory cycle	2	490/-
3	Human values	1	150/-

ORDER_NO, BOOK_TITLE → Primary key (composite)

ORDER_NO, BOOK_TITLE → QUANTITY
BOOK_TITLE → UNIT_PRICE

Whenever there is composite primary key then all the non prime attribute must be fully dependent on composite primary key. Hence the above table violate 2NF for

BOOK_TITLE	UNIT_PRICE
DBMS	450/-
Software	300/-
Theory	490/-
Human	150/-
CO	390/-

Already in 2NF

Consider the Relay table (Decompose)

ORDER_NO	BOOK_TITLE	QUANTITY
1	DBMS	2
1	Python	1
1	OS	3
2	DMDS	1
2	UV	1
3	CO	1
3	OS	2
3	UV	1

Date 1/1
Page 30

2NF

R R(A, B, C, D, E, F, G, H, I, J)

AB → C
A → DE
AB → F
F → GH
D → IJ

→ | A | D | E | → | D | R |

2NF

Relay table

R(A D C G H I J)

→ | A | D | C | G | H | I | J |

1NF, remove all the multiple value
eg - phone no.

2NF, remove partial dependency
XY → Z
Y → K

Partial

*THIRD NORMAL FORM (3NF)

A relation R is in 3NF, if it is in 2NF and there are no transitive dependencies.

By transitive dependencies we mean that no non-prime attribute transitively depend upon the key of R.

In simple terms we can say that no non-prime attribute identifies other non-prime attribute present in R.

Date 9/3/19
Page 31

Empid	Ename	Addeen	Salary	Dept_name	D_loc

Empid → Ename, Addeen, Salary
Empid → Dept_name
Dept_name → D_loc [Voilate 3NF]
Empid → D_loc

Rollno	Name	Addeen	Project_id	Project_name

Rollno → Name, Addeen, Project_id
Projectid → Project_name [Voilate 3NF]
Rollno → Project_name

R1

Ename	Empid	Addeen	Salary	Dept_name

R2

Dept_name	Dloc

Table decomposition

R(A, B, C, D, E, F, G, H, I, J) AB is the key

AB → C
A → DE
F → GH
D → IJ Since they is AB and it is composite key then we check 2NF.
B → F

AB is the key of the relation and there is partial dependency since AB → C
A → DE
B → F

Now we decompose the given relation R

$A \to DE$ we find all the closure of the identifying key

$A^+ = \{ADEIJ\}$ — (1)

$B \to F$ we find all the closure of the identifying key

$B^+ = \{BFGH\}$ — (2)

Hence decomposed table for $A \to DE$ is

| A | D | E | I | J | R_1

and decomposed table for $B \to F$ is identifying terms we will be added in remaining table

| B | F | G | H | R_2

and the remaining table for R is

| A | B | C | R_3

R_1 | A | D | E | I | J | Since there exist transitive dependency

i.e. $A \to DE$ $D \to IJ$ $A \to IJ$

Hence the new table decomposed is

R_{11} | A | D | E | R_{12} | D | I | J | R_5

Consider R_2 | B | F | G | H | there exist transitive dependency i.e. $B \to F$ and $F \to GH$ hence $B \to GH$

∴ Decomposed table is

R_{21} | F | G | H | R_{22} | B | F |

Consider R_3 is already in 3NF as there is no transitive dependency of the non-prime attribute only

Hence

∴ $R_1, R_2, R_3, R_{11}, R_{12}, R_{21}, R_{22}$ are in 3NF.

13/03/19

Minimal Cover of a set of FD's

Step 1: Singleton RHS

Step 2: No extraneous LHS Attribute

Step 3: No redundant FD's.

Q For a given set of FD's

$F = \{ A \to D,\ BC \to AD,\ C \to B,\ C \to A,\ E \to D \}$

find the minimal cover of F

Affter step 1 is singleton RHS

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

Step 2: no extraneous attr

$BC \to A$

$BC^+ = \{BC\}$

$= \{B\}$

$C^+ = \{CABD\}$

B is extraneous

$C \to A$

$BC \to D$

$BC^+ = B^+ = \{B\}$

$C^+ = \{CABD\}$

B is extraneous

$C \to D$

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

Step 3: No redundant FD's

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

$A \to D$ $C \to B$ $E \to D$

$A^+ = \{A\}$ $C^+ = \{ADB\}$ $E^+ = \{ED\}$

$A \to D$ $C \to B$ $C \to D$ $E \to B$ $C \to D$

$\Rightarrow A^+ = \{A\}$ $C^+ = \{ADB\}$ $C^+ = \{BDA\}$ $E^+ = \{E\}$ $C^+ = \{BAD\}$

$A \to D$ is removed $C \to B$ removed $C \to A$ is removed $E \to B$ is removed $C \to D$ is removed

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

For a given set of FD's

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

Date / / Page 34

Step 1: Singleton RHS

$A \to BC$ i.e. $A \to B$ and $A \to C$

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

Step 2: No extraneous LHS attr

$AB \to C$

$A^+ = \{ ABC \}$

$B^+ = \{ BC \}$

$AB \to C$ i.e. $A \to C$ and $B \to C$

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

Step 3: No redundant FD's

Removing same FD: $\{A \to B\}$

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

$A \to B$ $\quad A^+ = \{ AC \}$ ∴ $A \to B$ is removed

$A \to C$ $\quad A^+ = \{ ABC \}$ since C is dependent on A in either way ∴ it will be removed $A \to C$

$B \to C$ $\quad B^+ = \{ B \}$ ∴ $B \to C$ is removed

Final Minimal Cover

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

Find the minimal cover of $F = \{ B \to A, D \to A, AB \to D \}$

Date / / Page 35

Step 1: Singleton RHS

Since all FD's are singleton FD's

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

Step 2: No extraneous LHS attr

$AB \to D$

$A^+ = \{ A \}$ → Since D is not dependent on A ∴ A is extraneous

$B^+ = \{ BAD \}$

$B \to D$

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

Step 3: No redundant FD's

$B \to A$ $\quad B^+ = \{ BDA \}$ ∴ $B \to A$ will be removed since A is dependent on A in either way

$D \to A$ $\quad D^+ = \{ D \}$ ∴ $D \to A$ is removed

$B \to D$ $\quad B^+ = \{ BA \}$ ∴ $B \to D$ is removed

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

TRANSACTIONS (Unit 4)

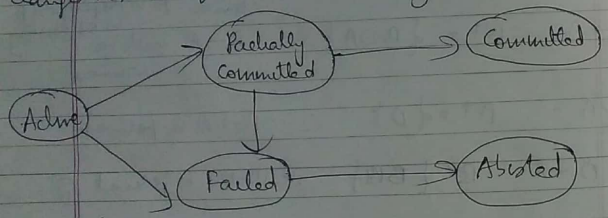
What do you understand by transaction and what are the properties of database transactions? / States of transaction / Life time of DB transaction.

Date 14/03/22 Page 36

Any unit of work that is being done in a DB is called transaction

Properties of transaction

① Atomicity - either all or none
② Consistency - Correctness of data
③ Isolation - Concurrent - whenever multiple transaction are occuring simultaneously i.e at the same time. Whenever concurrent transaction are occuring then medium should be used such that every concurrent transaction should act as individual transaction.
④ Durability - The database should be durable i.e the changes that are made should be long term



Active state - Whenever a transaction is under execution.

PC state - When the final statement of the transaction is under execution is about to execute

Failed state - A transaction due to any failures the transaction cannot be completed further

Committed state - The changes in the DB are made permanent

Aborted state - The transaction is stopped or abisted. No data values needs to be rollback.

kill state Whenever a transaction is stopped forcefully

Date / / Page 37

Then it is KILL.

Restart state - Restart a KILL or ABORTED transaction

Whenever a transaction is tasked again i.e the transaction which was killed or abisted

```
T1 = Read(A)      [Reading the value of data item]
     A = A - 100
     Write(A)     [Writing the value of data item]
     Read(B)
     B = B + 100
     Write(B)
```

* Schedule -

an order/way

A schedule is a way in which the transaction will execute

→ Serial schedule
→ Concurrent schedule

① Serial Schedule

T1	T2
read(A)	2000
A = A - 50	1950
write(A)	1950
Read(B)	1000
B = B + 50	1050
Write(B)	1050

Note - For n no. of transactions in a given schedule S°. n! are the total no. of serial schedules

Read(A)	1950	
temp = A * 0.1	195	1950 → 1755
A = A - temp	1755	
Write(A)	1755	A = 1755
Read(B)	1050	1050 → 1245
B = temp + B	1245	
Write(B)	1245	B = 1245

Date 1 1
Page 38

Concurrent Schedule

A = 2000
B = 1000

T_1	T_2
Read (A) – 2000	
A = A – 50 1950	
write (A) 1950	
	Read (A) 1950
	temp = A * 0.1 195
	A = A – temp 1755
	write (A) 1755
Read (B) 1000	
B = B + 50 1050	
write (B) 1050	
	Read (B) 1050
	B = B + temp 1245
	write (B) 1245

A = 1755
B = 1245

15/03/19

T_1	T_2
Read (A) x	
A = A – 50 x–50 1950	
	Read (A) 2000
	temp = A * 0.1 x*0.1 0.1x
	A = A – temp x – 0.1x
	write (A) 1800
	Read (B) 1000
write (A) 1950	
read (B) 1000	
B = B + 50 1050	
write (B) 1050	
	B = B + temp 1200
	write (B) 1200

$\frac{9}{10} \times 2000$ = 1800

A = 0.9x = 1800

B = y + 50 + 0.1x

1250 1050

1250
1950
. 50

Date 15/03/19
Page 39

Conflict Serializibility

Let us configure a schedule S in which there are two consecutive instruction I_i, I_j of T_i and I_j refer to the same same of data item Q then

There are four cases that needs to be considered

① I_i = Read (Q) and I_j = Read (Q) ∴ The order of read doesn't matter. The order of I_i and I_j doesn't matters since only read operation is being performed

② I_i = Read (Q) and I_j = Write (Q). ∴ If I_i comes before I_j then I_i is not reading the value of I(j). However if I_i comes after I_j then I_i will read the value of I_j. It means we cannot swap I_i and I_j.

③ I_i = Write (Q) and I_j = Read (Q) ∴ for reasons similar to ② these cannot be swap.

④ I_i = Write (Q) I_j = Write (Q) ∴ The order of instruction doesn't effect I_i and I_j. However the final value of Q is decided by the transaction i.e performing the final write. Hence we cannot swap I_i and I_j.

16/03/19

T_1	T_2
Read (A)	
Write (A)	
	Read (A)
	write (A)
Read (B)	
Write (B)	
	Read (B)
	Write (B)

⇒

T_1	T_2
Read (A)	
W (A)	
R (B)	
W (B)	
	R (A)
	W (A)
	R (B)
	W (B)

concurrent

By swapping the instruction we have transformed the schedule S_2 into serial schedule $\langle T_1, T_2 \rangle$. Hence the schedule S_2 is a conflict serializability schedule

→ Conflict equivalent

→ Conflict serializability

T_1	T_2
R(A)	
	R(A)
W(A)	
	W(A)
R(B)	
	R(B)
W(B)	

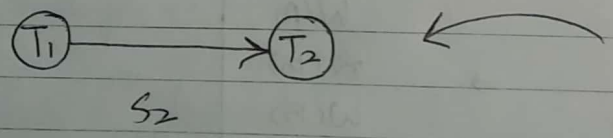
we cannot convert schedule S_3 into a serial schedule. Hence schedule S_3 is a non conflict serializable schedule. Hence schedule which are C.S give consistent data

Date 16/09/19
Page 40

Testing of Serializibility using Directed Graph

Consider a schedule 'S' we construct a directed graph also called a precedence graph from 'S'. This graph consist of a pair $G=(V, E)$ where V is a set of vertices and E is set of edges. The set of vertices constitute of all the transactions participating in 'S'. The set of edges consist of all edges $T_i \to T_j$ for which one of the following condition hold

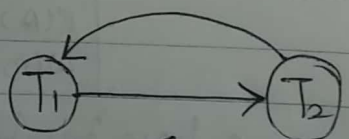
1. means T_i executes Read(Q) before Tj executes Write(Q)
2. Ti executes write(Q) before Tj executes Read(Q).
3. Ti executes write(Q) before Tj executes write(Q).



S_2

T_1	T_2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
	W(B)

T_1	T_2
R(A)	
	R(A)
	W(A)
	R(B)
W(A)	
R(B)	
W(B)	



S_3

Whenever a preceedence graph has a cycle then the corresponding schedule 'S' is a non-conflict serializible schedule and if there is no cycle then conflict serializible schedule.