CS262

LAB ASSIGNMENT 8

Hostel Management System

TA: Mr. Atul Sharma

GROUP 4

GROUP MEMBERS:

KAMLESH KUMAR : 202051099

KANANI DARPAN ASHOKBHAI : 202051100

KOMMULA CHIRANJEEVI SAGAR : 202051103

LINGAREDDY SAI CHARAN PREET REDDY : 202051111

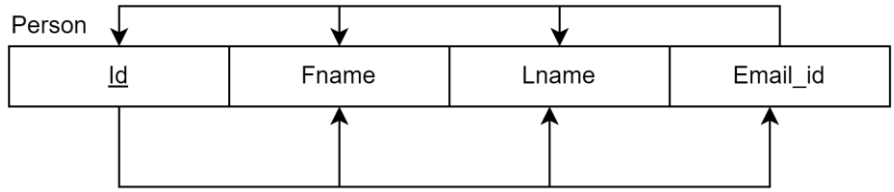
RUSHIKESH JADHAV : 202051164

Task 1-

Find closure and minimal cover of the functional dependencies designed in Assignment 7 Task 1.

→

Person :



FD:{ { Id → Fname,Lname,Email_Id}, { Email_Id → Id,Fname,Lname} }

PK: Id

CK: Id, Email_Id

Closure:

Id^+ = {Id, Fname, Lname, Email_Id}

$Fname^+$ = {Fname}

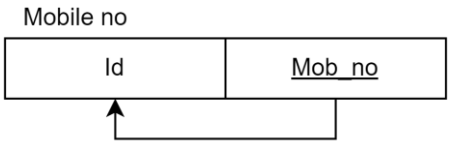
$Lname^+$ = {Lname}

$Email\_Id^+$ = {Id, Fname, Lname, Email_Id}

Minimal Cover:

{ Id → Fname, Id → Lname, Id → Email_Id, Email_Id → Id }

Mobile_no



FD: {Mob_no → Id }

PK: Mob_no

CK: Mob_no

Closure:

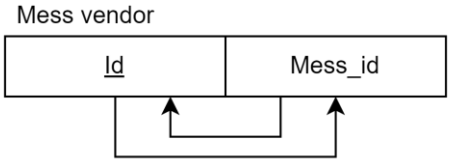
Mob_no^{+} = { Mob_no, Id}

Id^{+} = { Id }

Minimal Cover:

{Mob_no → Id }

Mess Vendor



FD: { Id → Mess_Id, Mess_Id → Id }

PK: Id

CK: Id, Mess_Id

Closure:

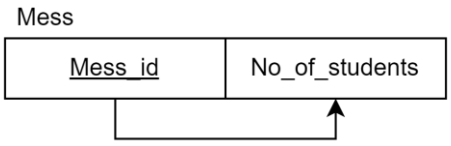
Id^{+} = { Id, Mess_Id }

$Mess\_Id^{+}$ = { Id, Mess_Id }

Minimal Cover:

{ Id → Mess_Id, Mess_Id → Id }

Mess



FD: { Mess_Id → No_of_Students }

PK: Mess_Id

CK: Mess_Id

Closure:

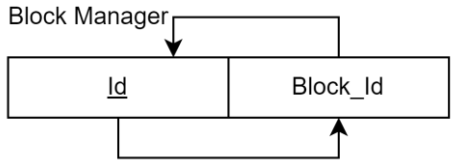
$Mess\_Id^{+}$ = { Mess_Id, No_of_Students }

$No\_of\_Students^{+}$ = { No_of_Students }

Minimal Cover:

{ Mess_Id → No_of_Students }

Block Manager:



FD: { Id → Block_Id, Block_Id → Id }

PK: Id

CK: Id, Block_Id

Closure:

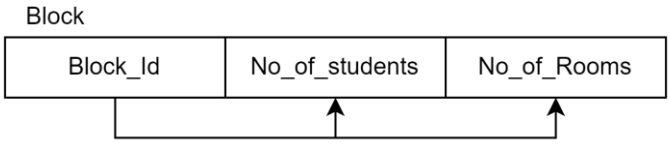
Id^{+} = { Id, Block_Id }

$Block\_Id^{+}$ = { Id, Block_Id }

Minimal Cover:

{ Id → Block_Id, Block_Id → Id }

Block



FD: { Block_Id → No_of_Students, No_of_Rooms }

PK: Block_Id

CK: Block_Id

Closure:

$Block\_Id^{+}$ = { Block_Id, No_of_Students, No_of_Rooms }

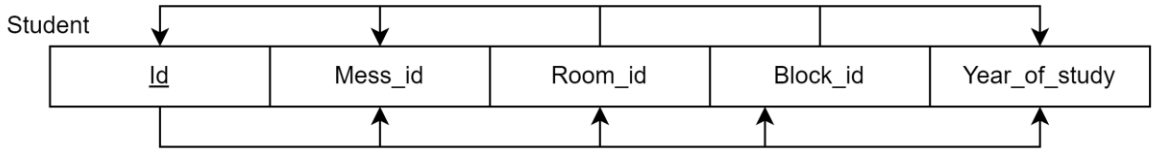
$No\_of\_Students^{+}$ = { No_of_Students }

$No\_of\_Rooms^{+}$ = { No_of_Rooms }

Minimal Cover:

{ Block_Id → No_of_Students, Block_Id → No_of_Rooms }

Student



FD: { {Id → Mess_Id, Block_Id, Room_Id, Year_of_Study},

{Block_Id, Room_Id → Id, Mess_Id, Year_of_Study } }

PK: Id

CK: Id, {Block_Id, Room_Id}

Closure:

Id^{+} = {Id, Mess_Id, Block_Id, Room_Id, Year_of_Study}

$Mess\_Id^{+}$ = { Mess_Id }

$Block\_Id^{+}$ = { Block_Id }

$Room\_Id^{+}$ = {Room_Id }

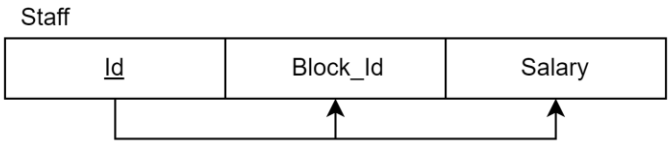
Year_of_Study = { Year_of_Study }

Minimal Cover:

{ Id → Block_Id, Id → Room_Id, Block_Id,Room_Id → Id,

Block_Id,Room_Id → Mess_Id, Block_Id,Room_Id → Year_of_Study }

Staff



FD: { Id → Block_Id, Id → Salary }

PK: Id

CK: Id

Closure:

Id^+ = { Id, Block_Id, Salary }

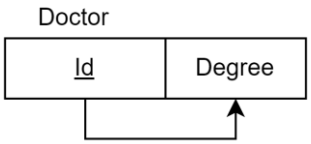
$Block\_Id^+$ = { Block_Id }

$Salary^+$ = { Salary }

Minimal Cover:

{ Id → Block_Id, Id → Salary }

Doctor



FD: { Id → Degree }

PK: Id

CK: Id

Closure:

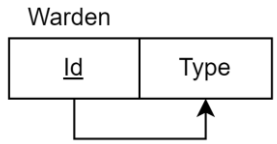
Id^+ = { Id, Degree }

$Degree^+$ = { Degree }

Minimal Cover:

{ Id → Degree }

Warden



FD: { Id → Type }

PK: Id

CK: Id

Closure:

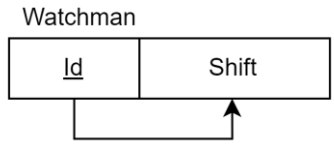
Id^+ = { Id, Type }

$Type^+$ = { Type }

Minimal Cover:

{ Id → Type }

Watchman



FD: { Id → Shift }

PK: Id

CK: Id

Closure:

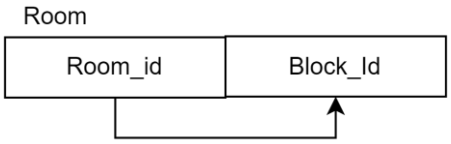
Id^+ = { Id, Shift }

$Shift^+$ = { Shift }

Minimal Cover:

{ Id → Shift}

Room



FD: { Room_Id → Block_Id }

PK: Id

CK: Id

Closure:

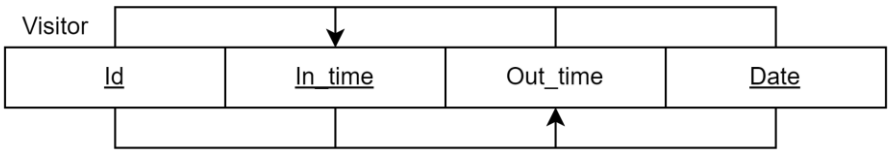
$Room\_Id^{+}$ = { Room_Id, Block_Id }

$Block\_Id^{+}$ = { Block_Id }

Minimal Cover:

{ Room_Id → Block_Id }

Visitor



FD: { { Id,Date,In_time → Out_time },

{ Id,Date,Out_time → In_time } }

PK: { Id, Date, In_time}

CK: { Id, Date, In_time}, { Id, Date, Out_time}

Closure:

Id^{+} = { Id }

$Date^{+}$ = { Date }

In_time^{+} = { In_time }

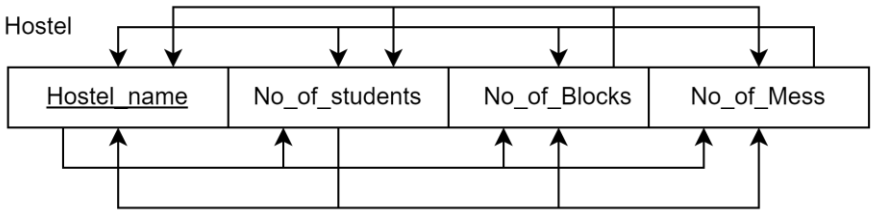
Out_time^{+} = { Out_time }

Minimal Cover:

{ { Id,Date,In_time → Out_time },

{ Id,Date,Out_time → In_time } }

Hostel



FD: { { Name → No_of_Blocks, No_of_Students, No_of_Mess }

{ No_of_Blocks → Name, No_of_Students, No_of_Mess }

{ No_of_Students → No_of_Blocks, Name, No_of_Mess }

{ No_of_Mess → No_of_Blocks, No_of_Students, Name } }

PK: Name

CK: Name, No_of_Blocks, No_of_Students, No_of_Mess }

Closure:

$Name^{+}$ = { Name, No_of_Blocks, No_of_Students, No_of_Mess }

$No\_of\_Blocks^{+}$ = { Name, No_of_Blocks, No_of_Students, No_of_Mess }

$No\_of\_Students^{+}$ = { Name, No_of_Blocks, No_of_Students, No_of_Mess }

$No\_of\_Mess^{+}$ = { Name, No_of_Blocks, No_of_Students, No_of_Mess }

Minimal Cover:

{ Name → No_of_Blocks, No_of_Blocks → Name,

No_of_Blocks → No_of_Students, No_of_Students → Name,

No_of_Students → No_of_Mess, No_of_Mess → No_of_Students }

Task 2- Show that the minimal cover of the functional dependencies designed in Task 1 is equivalent to the original set of functional dependencies.

→

Person :

X: { { Id → Fname,Lname,Email_Id}, { Email_Id → Id,Fname,Lname} }

Y: { Id → Fname, Id → Lname, Id → Email_Id, Email_Id → Id }

To check if X covers Y:

Id^+ = Id,Fname,Lname,Email_Id

$Email\_Id^+$ = Id,Fname,Lname,Email_Id

So X covers Y

To check if Y covers X:

Id^+ = Id,Fname,Lname,Email_Id

$Email\_Id^+$ = Id,Fname,Lname,Email_Id

So Y covers X

∴ X and Y are equivalent.

Mobile_no

X : {Mob_no → Id }

Y : {Mob_no → Id }

To check if X covers Y:

Mob_No^+ = Id, Mob_no

So X covers Y

To check if Y covers X:

Mob_No^+ = Id, Mob_no

So Y covers X

∴ X and Y are equivalent.

Mess Vendor

X: { Id → Mess_Id, Mess_Id → Id }

Y: { Id → Mess_Id, Mess_Id → Id }

To check if X covers Y:

Id^+ = Id, Mess_Id

$Mess\_Id^+$ = Id, Mess_Id

So X covers Y

To check if Y covers X:

Id^+ = Id, Mess_Id

$Mess\_Id^+$ = Id, Mess_Id

So Y covers X

∴ X and Y are equivalent.

Mess

X: { Mess_Id → No_of_Students }

Y: { Mess_Id → No_of_Students }

Both X and Y are same

∴ X and Y are equivalent.

Block Manager:

X: { Id → Block_Id, Block_Id → Id }

Y: { Id → Block_Id, Block_Id → Id }

Both X and Y are same

∴ X and Y are equivalent.

Block

X: { Block_Id → No_of_Students, No_of_Rooms }

Y: { Block_Id → No_of_Students, Block_Id → No_of_Rooms }

To check if X covers Y:

$Block\_Id^{+}$ = Block_Id, No_of_Students, No_of_Rooms

So X covers Y

To check if Y covers X:

$Block\_Id^{+}$ = Block_Id, No_of_Students, No_of_Rooms

So Y covers X

∴ X and Y are equivalent.

Student

X: { {Id → Mess_Id, Block_Id, Room_Id, Year_of_Study},

{Block_Id, Room_Id → Id, Mess_Id, Year_of_Study } }

Y: { Id → Block_Id, Id → Room_Id, Block_Id,Room_Id → Id,

Block_Id,Room_Id → Mess_Id, Block_Id,Room_Id → Year_of_Study }

To check if X covers Y:

Id^{+} = Id, Mess_Id, Block_Id, Room_Id, Year_of_Study

$\{Block\_Id, Room\_Id\}^{+}$ = Id, Mess_Id, Block_Id, Room_Id, Year_of_Study

So X covers Y

To check if Y covers X:

Id^{+} = Id, Mess_Id, Block_Id, Room_Id, Year_of_Study

$\{Block\_Id, Room\_Id\}^{+}$ = Id, Mess_Id, Block_Id, Room_Id, Year_of_Study

So Y covers X

∴ X and Y are equivalent.

Staff

X: { Id → Block_Id, Id → Salary }

Y: { Id → Block_Id, Id → Salary }

Both X and Y are same

∴ X and Y are equivalent.

Doctor

X: { Id → Degree }

Y: { Id → Degree }

Both X and Y are same

∴ X and Y are equivalent.

Warden

X: { Id → Type }

Y: { Id → Type }

Both X and Y are same

∴ X and Y are equivalent.

Watchman

X: { Id → Shift }

Y: { Id → Shift}

Both X and Y are same

∴ X and Y are equivalent.

Room

X: { Room_Id → Block_Id }

Y: { Room_Id → Block_Id }

Both X and Y are same

∴ X and Y are equivalent.

Visitor

X: { { Id,Date,In_time → Out_time },

{ Id,Date,Out_time → In_time } }

Y: { { Id,Date,In_time → Out_time },

{ Id,Date,Out_time → In_time } }

Both X and Y are same

∴ X and Y are equivalent.

Hostel

X: { { Name → No_of_Blocks, No_of_Students, No_of_Mess }

{ No_of_Blocks → Name, No_of_Students, No_of_Mess }

{ No_of_Students → No_of_Blocks, Name, No_of_Mess }

{ No_of_Mess → No_of_Blocks, No_of_Students, Name } }

Y: { Name → No_of_Blocks, No_of_Blocks → Name,

No_of_Blocks → No_of_Students, No_of_Students → Name,

No_of_Students → No_of_Mess, No_of_Mess → No_of_Students }

To check if X covers Y:

$Name^{+}$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

$No\_of\_Blocks^{+}$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

No_of_Students$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

No_of_Mess$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

So X covers Y

To check if Y covers X:

Name$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

No_of_Blocks$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

No_of_Students$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

No_of_Mess$^+$ = Name, No_of_Blocks, No_of_Students, No_of_Mess

So Y covers X

∴ X and Y are equivalent.

Task 3- Using the Matrix method, show that the relations designed in Task 3 of Assignment 7 satisfies the Lossless join property.

→

In task 3 of Assignment 7 we divided the table

Student(USER_ID, HOSTEL_ID, MESS_ID, ROOM_ID, YEAR_OF_STUDY)

in two

R1(USER_ID,MESS_ID, ROOM_ID),YEAR_OF_STUDY) AND

R2(ROOM_ID, HOSTEL_ID)

FD: { {USER_ ID -> HOSTEL_ID ,MESS_ID ,ROOM_ID,YEAR_OF_STUDY},

{ROOM_ID->HOSTEL_ID}}

	USER_ID	HOSTEL_ID	MESS_ID	ROOM_ID	YEAR_OF_STUDY
R1	α 11	β 12	α 13	α 14	α 15
R2	β 21	α 22	β 23	α 24	β 25

For ROOM_ID->HOSTEL_ID

we get α in both R1 and R2 for ROOM_ID so HOSTEL_ID in R1 becomes α.

	USER_ID	HOSTEL_ID	MESS_ID	ROOM_ID	YEAR_OF_STUDY
R1	α 11	α 22	α 13	α 14	α 15
R2	β 21	α 22	β 23	α 24	β 25

Now we have alpha in full R1. So, it satisfies the lossless join property.

---

*Note: We have updated our relational model. So some of the tables in assignment 8 are different than assignment 7.

-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-

THANK YOU