Question 1

(i) {W -> Y, X -> Z} |= {WX -> Y}

Using the augmentation axiom, we add X on both sides the the FD W->Y resulting in WX🡪YX.

Next, we using the decomposition axiom will result in giving us

WX🡪Y

WX 🡪 X

Therefore, as we can see the FD WX 🡪 Y is possible.

ii) {X 🡪 Y, X 🡪 W, WY 🡪 Z} |= {X 🡪 Z}

the first step is using Union for the two given FD’s X 🡪 Y, X 🡪 W which will result in giving us the following fd X 🡪 WY.

Now with the resulting FD we can use the transitivity axiom with X 🡪 WY, WY 🡪 Z which will give us the FD that we have been looking for X 🡪 Z

iii) {XY 🡪 Z, Y 🡪 W} |= {XW 🡪 Z}

The following FD XW 🡪 Z is not possible. To prove this lets find the attribute set closure of {X, W}+

We first start with XY 🡪 Z but we do not possess W attribute in our set so nothing changes

We move to Y 🡪 W but again Y does not exist in out set therefore our set remain X,W and there is no way to access Z

iv) {X 🡪 Z, Y 🡪 Z} |= {X 🡪 Y}

The following FD X 🡪 Z is not possible

If we follow the Attribute set closure of X+ with the given dependancies X does not determine Y so it is not possible for X 🡪 Y to be valid

v) {X 🡪 Y, XY 🡪 Z} |= {X 🡪 Z}

Using the pseudo transitivity axiom for both the given attributes X 🡪 Y, XY 🡪 Z will result in us getting

XX 🡪 Z which is the same as X 🡪 Z so it does exist.

Another way to verify this is attribute closure of X🡪Z so we want to find {X}+

now using X 🡪 Y, {X} + += {Y} will result in {X,Y} now we move to the next FD

XY 🡪 Z, we do possess X and Y therefore Z will be added {X,Y,Z} therefore we see that X 🡪 Z does follow from this relation as the attribute Z is in {X}+

Question 2

R = {room#, size, type, no – of – beds, price, date, status} holds the following dependencies

room# 🡪 size, type

size, type,date 🡪 price, date

room#, date 🡪 status, price

room#, size 🡪 no – of – beds

a) to find all candidate keys we must find a combination of attributes that will alow us to access every other attributes.

If we look at room#, this attributes gives us the following set based on the given FD’s

{room#, size, type, no-of-beds} so the only attributes not accessed by this key are date, price and status so we need to find a combination that will allow us to get those as well.

Based on the FD’s above we can see that the status and price can be accessed if we have both room# and date therefore

{room#, date} is the only candidate key available

b) from the previous example we know the that the candidate keys are { room#,date}, The set of key attributes are: { room#,date }. To see which FD violates 3NF we check whether the LHS is a proper subset of some candidate key or the RHS are not all key attributes

*The FD (room# 🡪 size, type) violates 3NF because our LHS is not a superkey as well as the fact that the RHS is a set of non-key attributes*

c)

Step 1: Rewrite the FD into those with only one attribute on RHS by using decomposition axiom. We obtain:

room# 🡪 size

room# 🡪 type

size, type,date 🡪 price

size, type,date 🡪 date

room#, date 🡪 status

room#, date 🡪 price

room#, size 🡪 no – of – beds

Step 2 is to remove trivial FDs. In this case we will remove size, type,date 🡪 date since date already exists in our set so this FD is redundant. Now we get

room# 🡪 size

room# 🡪 type

size, type,date 🡪 price

room#, date 🡪 status

room#, date 🡪 price

room#, size 🡪 no – of – beds

the next step is to minimize LHS as much as possible but the only one we can do so is room#, size 🡪 no – of – beds where we can remove size due to the fact that room# can functionally determine size# so room# is enough to define both room# and size therefore we obtain the following

room# 🡪 size

room# 🡪 type

size, type,date 🡪 price

room#, date 🡪 status

room#, date 🡪 price

room# 🡪 no – of – beds

now the final step is to remove redundant FD’s.

in this case we can remove the FD (room#, date 🡪 price) due to the fact that the previous FD size, type,date 🡪 price is already defined and since room# functionally determines both attributes, both FD’s are practically the same so we can remove this FD. As a result we obtain

room# 🡪 size

room# 🡪 type

size, type,date 🡪 price

room#, date 🡪 status

room# 🡪 no – of – beds

d)

room# 🡪 size

room# 🡪 type

size, type,date 🡪 price

room#, date 🡪 status

room# 🡪 no – of – beds

lets create a table with all the attributes all the relations available are the fundamental dependancies above.

The first step is to fill out our tables with alpha symbols on each relations which can be taken from the FD dependancies above. This means we get the following

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room# | Size | Type | No-of-beds | Price | Date | status |
| (room#,size) | α | α |  |  |  |  |  |
| (room#, type) | α |  | α |  |  |  |  |
| (size, type, date) |  | α | α |  | α | α |  |
| room#,date, status | α |  |  |  |  | α | α |
| (room#, no-of-beds) | α |  |  | α |  |  |  |

This is our dependency preserving decomposition of R into 3NF

Now we will start with the FD room# 🡪 size. We need to find one row that contains alpha for corresponding LHS and RHS meaning both attributes have alpha which we in row1.

Now we find any row that contains room# and we can add alpha symbols to the size attributes. We can see that row 2, 4 and 5 satisfy the condition so we add alpha to size

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room# | Size | Type | No-of-beds | Price | Date | status |
| (room#,size) | α | α |  |  |  |  |  |
| (room#, type) | α | α | α |  |  |  |  |
| (size, type, date) |  | α | α |  | α | α |  |
| room#,date, status | α | α |  |  |  | α | α |
| (room#, no-of-beds) | α | α |  | α |  |  |  |

Now we repeat the process for the next FD room# 🡪 type. We will get the following table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room# | Size | Type | No-of-beds | Price | Date | status |
| (room#,size) | α | α | α |  |  |  |  |
| (room#, type) | α | α | α |  |  |  |  |
| (size, type, date) |  | α | α |  | α | α |  |
| room#,date, status | α | α | α |  |  | α | α |
| (room#, no-of-beds) | α | α | α | α |  |  |  |

Next is the FD size, type,date 🡪 price however, only a single row contains the LHS size, type and date so no chances are made.

Next is the FD room# 🡪 no-of-beds, so all the rows that contains room# can have alpha added to the no-of-beds columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room# | Size | Type | No-of-beds | Price | Date | status |
| (room#,size) | α | α | α | α |  |  |  |
| (room#, type) | α | α | α | α |  |  |  |
| (size, type, date) |  | α | α | α | α | α |  |
| room#,date, status | α | α | α | α |  | α | α |
| (room#, no-of-beds) | α | α | α | α |  |  |  |

Now we restart the process from the tp dependency. Room#-->size and room# 🡪 type so we move on to size, type,date 🡪 price

And from what we see from the above table the 4th row contains the LHS attributes so we can add alpha symbol in the price column so we get

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room# | Size | Type | No-of-beds | Price | Date | status |
| (room#,size) | α | α | α | α |  |  |  |
| (room#, type) | α | α | α | α |  |  |  |
| (size, type, date) |  | α | α | α | α | α |  |
| **room#,date, status** | **α** | **α** | **α** | **α** | **α** | **α** | **α** |
| (room#, no-of-beds) | α | α | α | α |  |  |  |

**Now because we have an entire row of that contains all the attributes of our dependency preserving decomposition is indeed loseless**

Question 10

a) {d | ∃ a,b ∈ DIRECTED( a[MName] = b[MName])

∧ a[AName] = “Prima Donna” ∧ b[AName] = “Me Mimi” ∧

ACTED\_IN(d[MName] = a[MName])}

b) {m | ∃ r ∈ REVIEW( r[Year] = 2006 ∧ r[Number\_of\_stars] = 5)

∧ m[MName] = r[MName]) ∧ m[Profit] = 0}

c) {t | ∃ d ∈ DIRECTED\_IN( t[DName] = d[DName] ∧

∃ a ∈ ACTED\_IN ( t[AName] = a[AName] ∧ a[MName] = d[MName] ∧

∃ m1 ∈ MOVIE ( ∃ m2 ∈ MOVIE (m1[Profit] > m2[Profit] ∧ m1[Year] = 2007)))}

d) {t | ∃ d ∈ DIRECTED\_IN( t[DName] = d[DName] ∧

∃ a ∈ ACTED\_IN ( t[AName] = a[AName] ∧ a[MName] = d[MName]))}