EXERCÍCIOS

1. Considere a relação `R(A, C, D, E, H)`. Calcule o conjunto de cobertura mínima de cada um dos conjuntos de dependências funcionais abaixo:

1.1. `F = {A → C, CA → D, E → AD, E → H}`

Find Minimal Cover

1.2. `G = {A → CD, E → AH}`

Find Minimal Cover

1.3. Os conjuntos de dependências funcionais `F` e `G` são equivalentes?

- Sim. São equivalentes pois os conjuntos de cobertura mínima de ambos são iguais.

2. Considere o esquema de relação

`EMP_DEPT(ename, ssn, bdate, address, dnumber, dname, dmgrssn)`

e o seguinte conjunto `G` de dependências funcionais em `EMP_DEPT`:

```
G = {{ssn} → {ename, bdate, address, dnumber},
     {dnumber} → {dname, dmgrssn}}
```

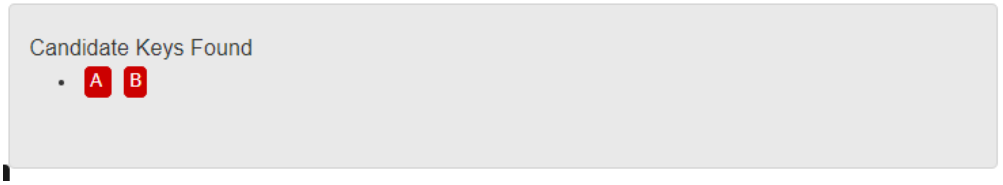
Calcule o conjunto de cobertura mínima de `G`.

Find Minimal Cover

3. Considere a relação R(A, B, C, D, E, F, G, H, I, J) e o conjunto de dependências funcionais F = {AB → C, A → DE, B → F, F → GH, D → IJ}.

3.1. Qual são as chaves candidatas de R?

- São candidatas A e B. Conforme imagem.

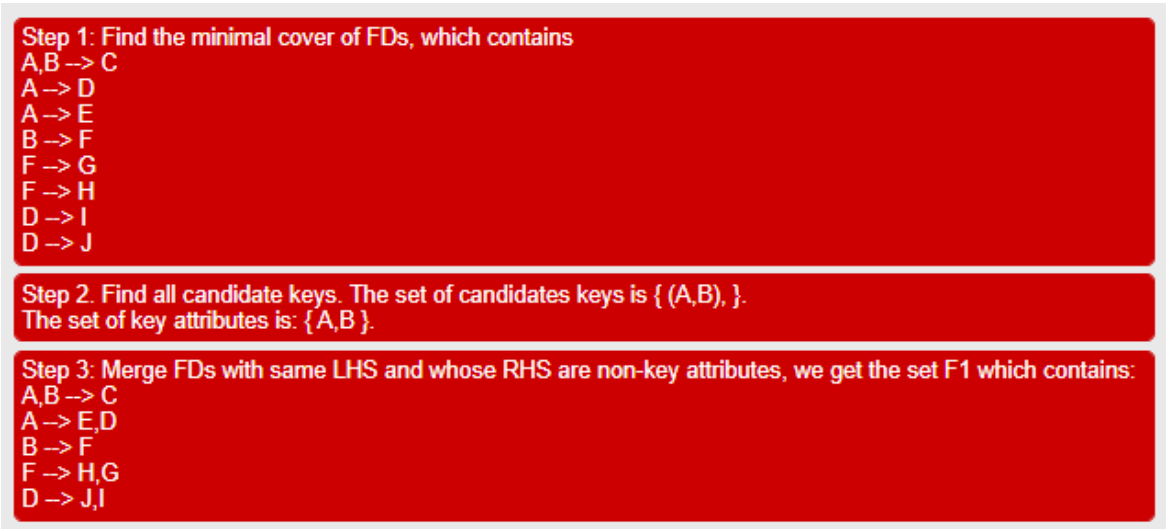


3.2. Ela está na 3NF?

- Não atende os critérios para 2NF, 3NF e nem BCNF. Conforme imagem.

3.3. Em caso negativo, decomponha R em relações 3NF.

- Seguindo o primeiro método:



Step 4: Check each FD in the set F1 for violation of 3NF, and split table accordingly.

Checking FD A,B --> C
FD does not violate 3NF
Checking FD A --> E,D
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

A,E,D
with FDs
A --> D,E

Checking FD B --> F
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,F
with FDs
B --> F

Checking FD F --> H,G
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

F,H,G
with FDs
F --> H,G

Checking FD D --> J,I
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

D,J,I
with FDs
D --> J,I

Step 5: Finally, add the following table into normalized 3NF table set (obtained by removing RHS attributes of FDs using which we produced a table):

A,B,C
with FDs
A,B --> C

4. Considere a relação `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, BD → EF, B → F, F → GH, D → IJ}`.
   4.1. Qual são as chaves candidatas de `R`?
   - São chaves candidatas A, B e D.

Candidate Keys Found
- A B D

4.2. Ela está na 3NF?

- Não atende os critérios para 2NF, 3NF e nem BCNF. Conforme imagem.

2NF

The table is not in 2NF.

3NF

The table is not in 3NF.

BCNF

The table is not in BCNF.

4.3. Em caso negativo, decomponha `R` em relações na 3NF.

- Seguindo o primeiro método:

Step 1: Find the minimal cover of FDs, which contains
A,B --> C
B,D --> E
B --> F
F --> G
F --> H
D --> I
D --> J

Step 2. Find all candidate keys. The set of candidates keys is { (A,B,D), }.
The set of key attributes is: { A,B,D }.

Step 3: Merge FDs with same LHS and whose RHS are non-key attributes, we get the set F1 which contains:
A,B --> C
B,D --> E
B --> F
F --> H,G
D --> J,I

Step 4: Check each FD in the set F1 for violation of 3NF, and split table accordingly.

Checking FD A,B --> C
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

A,B,C
with FDs
A,B --> C

Checking FD B,D --> E
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,D,E
with FDs
B,D –> E

Checking FD B –> F
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,F
with FDs
B –> F

Checking FD F –> H,G
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

F,H,G
with FDs
F –> H,G

Checking FD D –> J,I
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

D,J,I
with FDs
D –> J,I

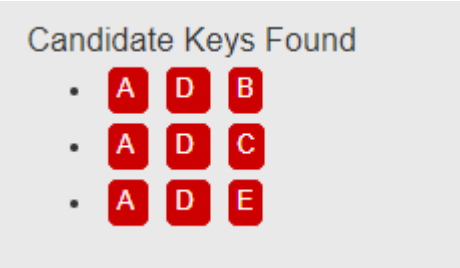
Step 5: Finally, add the following table into normalized 3NF table set (obtained by removing RHS attributes of FDs using which we produced a table):

A,B,D
with FDs

5. Considere uma relação `R(A, B, C, D, E)` e o conjunto de dependências funcionais `F = {AB → C, CD → E, DE → B}`

   `AB` é uma chave candidata dessa relação? Se não, `ABD` é?

   - Não. Tanto AB quanto ABD não são chaves candidatas da relação.



6. Considere a relação `R`, que possui atributos que guardam horários de cursos e seções em uma universidade:

```
R(courseno, secno, offeringdept, credithours, courselevel,
   instructorssn, semester, year, dayshours, roomno, noofstudents)
```

Suponha que as seguintes dependências funcionais em `R`:

```
{{courseno} → {offeringdept, credithours, courselevel},
 {courseno, secno, semester, year} →
    {dayshours, roomno, noofstudents, instructorssn}
 {roomno, dayshours, semester, year} →
    {instructorssn, courseno, secno}}
```

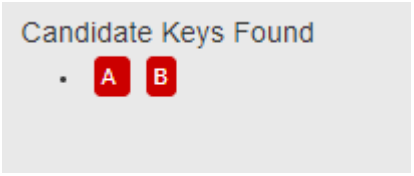
Encontre as chaves candidatas para `R` e decomponha a relação em relações na `3NF`.

- As chaves candidatas para a relação são as seguintes:

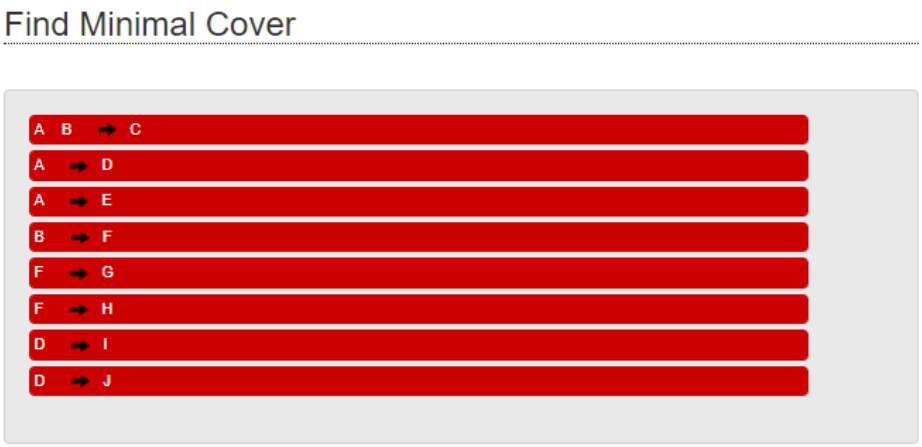
- Decompondo a relação para a 3NF, seguindo o primeiro método:

7. Considere a relação universal `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, A → DE, B → F, F → GH, D → IJ}`. Determine as chaves candidatas para `R`, encontre uma cobertura mínima para `F` e decomponha `R` em relações na 3NF.

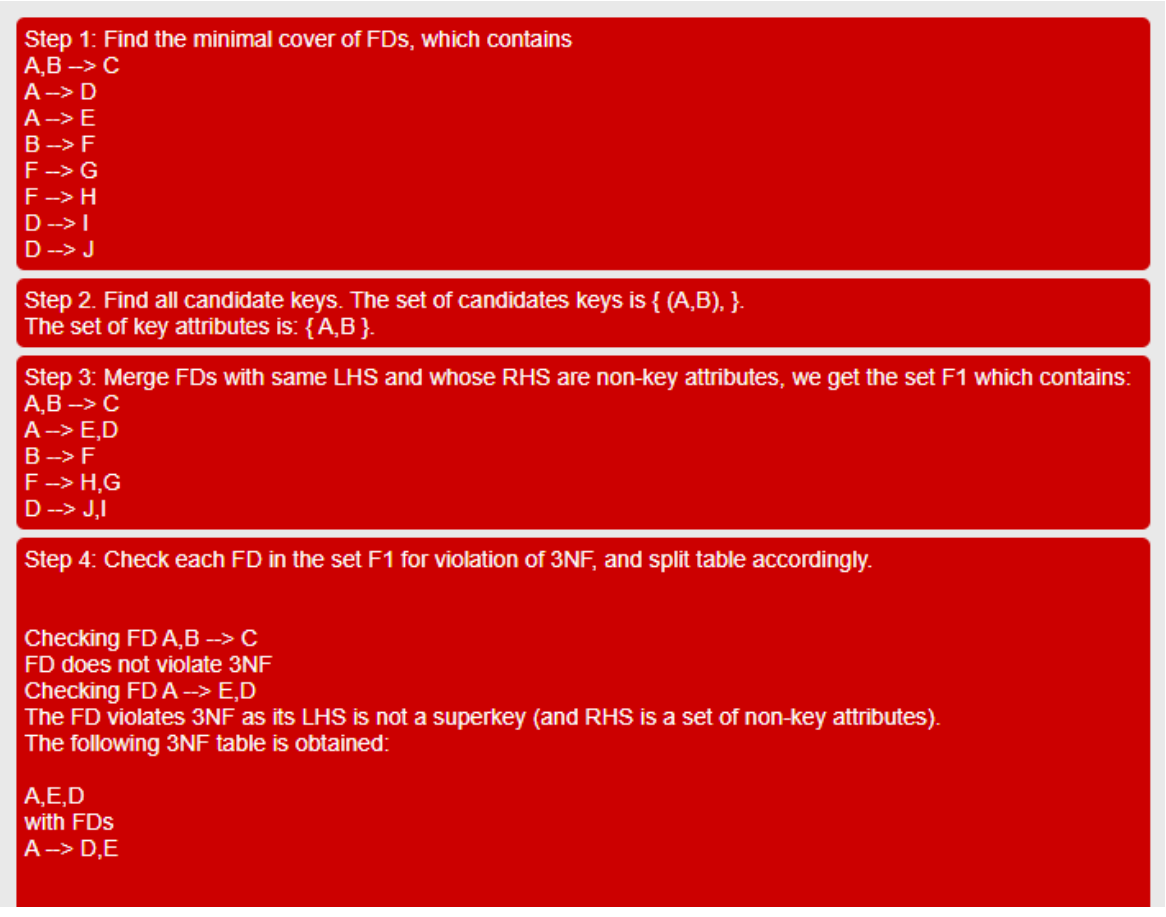
- As chaves candidatas para a relação são A e B. Conforme imagem:



- A cobertura mínima para F é:



- Decompondo a relação para a 3NF, seguindo o primeiro método:



Checking FD B --> F
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,F
with FDs
B --> F

Checking FD F --> H,G
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

F,H,G
with FDs
F --> H,G

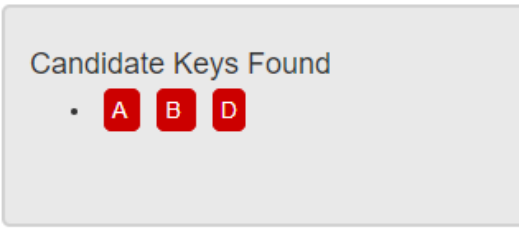
Checking FD D --> J,I
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

D,J,I
with FDs
D --> J,I

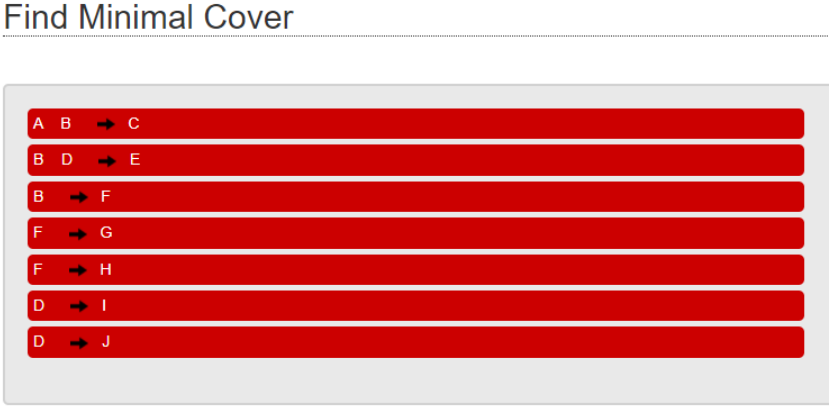
Step 5: Finally, add the following table into normalized 3NF table set (obtained by removing RHS attributes of FDs using which we produced a table):

A,B,C
with FDs
A,B --> C

8. Considere a relação universal `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, BD → EF, B → F, F → GH, D → IJ}`. Determine as chaves candidatas para `R`, encontre uma cobertura mínima para `F` e decomponha `R` em relações na 3NF.
   - As chaves candidatas para a relação são A, B e D. Conforme imagem:



   - A cobertura mínima para F é:



- Decompondo a relação para a 3NF, seguindo o primeiro método:

Step 1: Find the minimal cover of FDs, which contains
A,B --> C
B,D --> E
B --> F
F --> G
F --> H
D --> I
D --> J

Step 2. Find all candidate keys. The set of candidates keys is { (A,B,D), }.
The set of key attributes is: { A,B,D }.

Step 3: Merge FDs with same LHS and whose RHS are non-key attributes, we get the set F1 which contains:
A,B --> C
B,D --> E
B --> F
F --> H,G
D --> J,I

Step 4: Check each FD in the set F1 for violation of 3NF, and split table accordingly.

Checking FD A,B --> C
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

A,B,C
with FDs
A,B --> C

Checking FD B,D --> E
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,D,E
with FDs
B,D --> E

Checking FD B --> F
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

B,F
with FDs
B --> F

Checking FD F --> H,G
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

F,H,G
with FDs
F --> H,G

Checking FD D --> J,I
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

D,J,I
with FDs
D --> J,I

Step 5: Finally, add the following table into normalized 3NF table set (obtained by removing RHS attributes of FDs using which we produced a table):

A,B,D
with FDs

9. Considere a relação universal `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, A → DE, B → F, F → GH, D → IJ}`. Decomponha `R` em relações na BCNF.

- É possível verificar que a relação R não atendem nenhuma NF:

Check Normal Form

2NF
The table is not in 2NF.

3NF
The table is not in 3NF.

BCNF
The table is not in BCNF.

- Decompondo a relação para a BCFN, temos o seguinte:

Step 1. Find merged minimal cover of FDs, which contains:
A,B --> C
A --> D,E
B --> F
F --> G,H
D --> I,J

Initially rel[1] contains the original table, with the FDs above

Round1: Checking whether table rel[1] is in BCNF

The FD [A --> D,E] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[2]= (A,D,E,I,J)
With FDs:

rel[3]= (A,B,C,F,G,H)
With FDs:

Round2: Checking whether table rel[2] is in BCNF

The FD [D --> I,J] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[4]= (D,I,J)
With FDs:

rel[5]= (A,D,E)
With FDs:

Round3: Checking whether table rel[3] is in BCNF

The FD [B --> F] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[6]= (B,F,G,H)
With FDs:

rel[7]= (A,B,C)
With FDs:

Round4: Checking whether table rel[4] is in BCNF

*** Table rel[4] is in BCNF already, send it to output ***

Round5: Checking whether table rel[5] is in BCNF

*** Table rel[5] is in BCNF already, send it to output ***

Round6: Checking whether table rel[6] is in BCNF

The FD [F --> G,H] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[8]= (F,G,H)
With FDs:

rel[9]= (B,F)
With FDs:

Round7: Checking whether table rel[7] is in BCNF

*** Table rel[7] is in BCNF already, send it to output ***

Round8: Checking whether table rel[8] is in BCNF

*** Table rel[8] is in BCNF already, send it to output ***

Round9: Checking whether table rel[9] is in BCNF

*** Table rel[9] is in BCNF already, send it to output ***

Normalize to BCNF

Attributes

D I J

Functional Dependencies

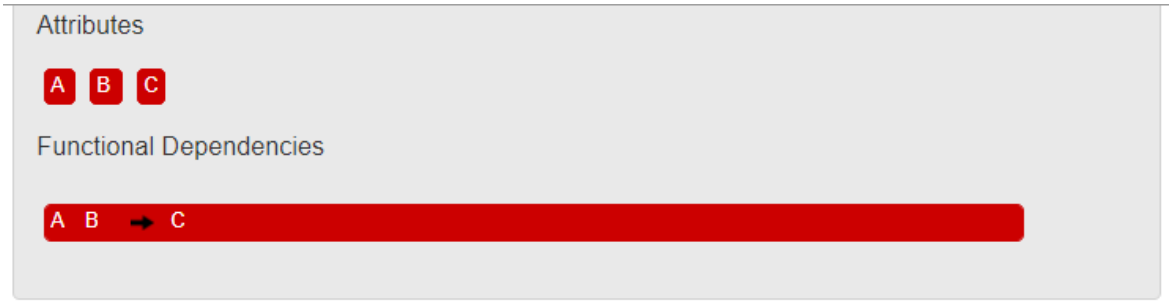
D → I J

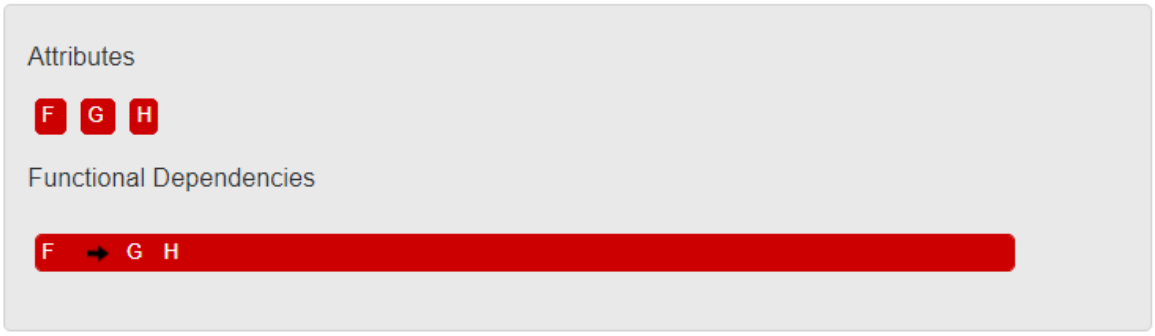
Attributes

A D E

Functional Dependencies

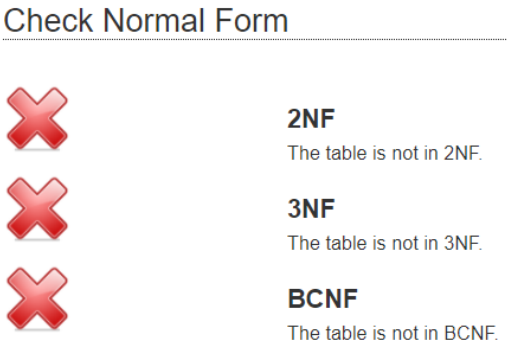
A → D E





10. Considere a relação universal `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, BD → EF, B → F, F → GH, D → IJ}`. Decomponha `R` em relações na BCNF.

- É possível verificar que a relação R não atendem nenhuma NF:



- Decompondo a relação para a BCFN, temos o seguinte:

Step 1. Find merged minimal cover of FDs, which contains:
A,B --> C
B,D --> E
B --> F
F --> G,H
D --> I,J

Initially rel[1] contains the original table, with the FDs above

Round1: Checking whether table rel[1] is in BCNF

The FD [A,B --> C] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[2]= (A,B,C,F,G,H)
With FDs:

rel[3]= (A,B,D,E,I,J)
With FDs:

Round2: Checking whether table rel[2] is in BCNF

The FD [B --> F] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[4]= (B,F,G,H)
With FDs:

rel[5]= (A,B,C)
With FDs:

Round3: Checking whether table rel[3] is in BCNF

The FD [B,D --> E] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[6]= (B,D,E,I,J)
With FDs:

rel[7]= (A,B,D)
With FDs:

Round4: Checking whether table rel[4] is in BCNF

The FD [F --> G,H] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[8]= (F,G,H)
With FDs:

rel[9]= (B,F)
With FDs:

Round5: Checking whether table rel[5] is in BCNF

*** Table rel[5] is in BCNF already, send it to output ***

Round6: Checking whether table rel[6] is in BCNF

The FD [D --> I,J] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[10]= (D,I,J)
With FDs:

rel[11]= (B,D,E)
With FDs:

Round7: Checking whether table rel[7] is in BCNF

*** Table rel[7] is in BCNF already, send it to output ***

Round8: Checking whether table rel[8] is in BCNF

*** Table rel[8] is in BCNF already, send it to output ***

Round9: Checking whether table rel[9] is in BCNF

*** Table rel[9] is in BCNF already, send it to output ***

Round10: Checking whether table rel[10] is in BCNF

*** Table rel[10] is in BCNF already, send it to output ***

Round11: Checking whether table rel[11] is in BCNF

*** Table rel[11] is in BCNF already, send it to output ***

Normalize to BCNF

Attributes

A B C

Functional Dependencies

A B → C

Attributes

A B D

Functional Dependencies

Attributes

F G H

Functional Dependencies

F → G H

Attributes

B F

Functional Dependencies

B → F

Attributes

D I J

Functional Dependencies

D → I J

Attributes

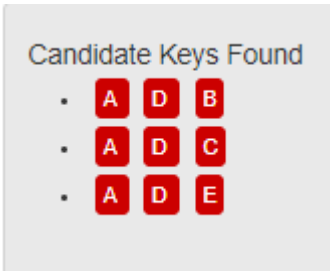
B D E

Functional Dependencies

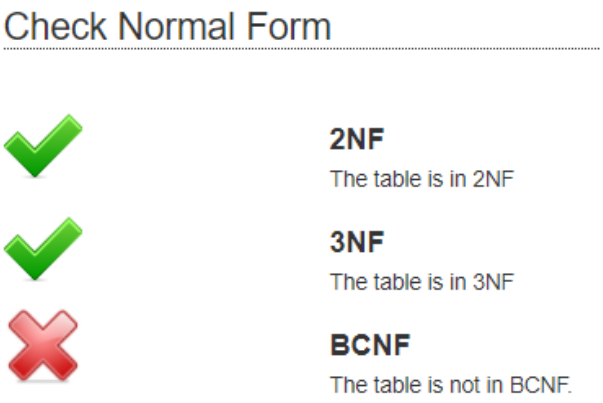
B D → E

11. Considere uma relação `R(A, B, C, D, E)` com as seguintes dependências funcionais: `F = {AB → C, CD → E, DE → B}`. Determine as chaves candidatas para `R` e decomponha `R` em relações na 3NF e em relações na BCNF.

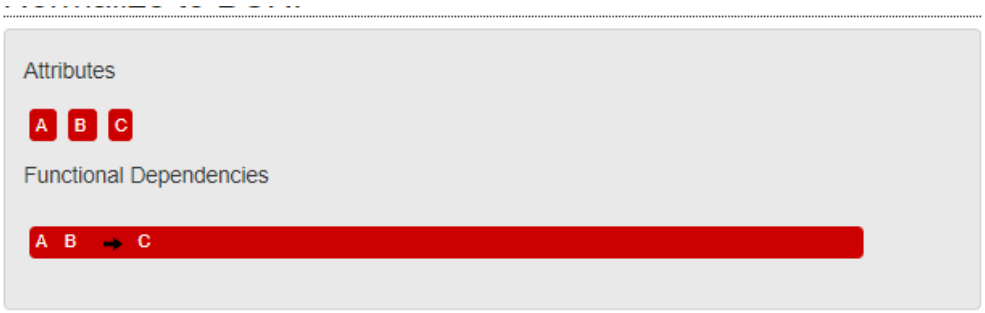
- As chaves candidatas para a relação são mostradas na imagem:

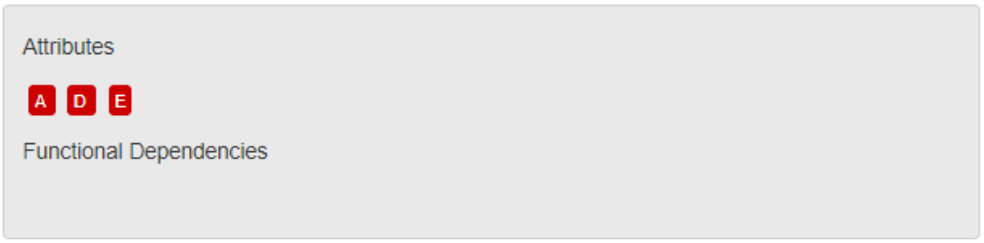


- É possível verificar que a relação R atende a 2NF e a 3NF:



- Tendo em vista que a Relação já se encontra na 3NF, segue a decomposição para a BCFN:





12. Considere a relação

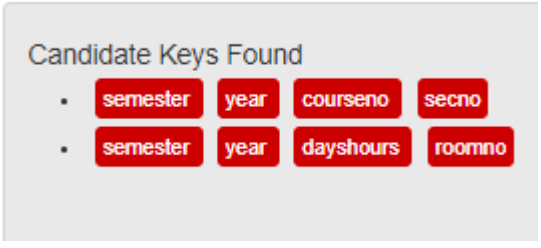
```
R = {courseno, secno, offeringdept, credithours, courselevel,
     instructorssn, semester, year, dayshours, roomno,
     noofstudents}
```

a as seguintes dependências funcionais:

```
{{courseno} → {offeringdept, credithours, courselevel} ,
 {courseno, secno, semester, year} →
      {dayshours, roomno, noofstudents, instructorssn} ,
 {roomno, dayshours, semester, year} →
      {instructorssn, courseno, secno}}
```

Determine as chaves candidatas para `R` e decomponha `R` em relações na 3NF e em relações na BCNF.

- As chaves candidatas para a relação são mostradas na imagem:



- É possível verificar que a relação R não atende a nenhuma NF:

Check Normal Form

2NF
The table is not in 2NF.

3NF
The table is not in 3NF.

BCNF
The table is not in BCNF.

- Decompondo a Relação para a 3NF, temos:

Step 1: Find the minimal cover of FDs, which contains
courseno --> offeringdept
courseno --> credithours
courseno --> courselevel
courseno,secno,semester,year --> dayshours
courseno,secno,semester,year --> roomno
courseno,secno,semester,year --> noofstudents
roomno,dayshours,semester,year --> instructorssn
roomno,dayshours,semester,year --> courseno
roomno,dayshours,semester,year --> secno

Step 2: Find all candidate keys. The set of candidates keys is { (semester,year,courseno,secno), (semester,year,dayshours,roomno), }.
The set of key attributes is: { semester,year,courseno,secno,dayshours,roomno }.

Step 3: Merge FDs with same LHS and whose RHS are non-key attributes, we get the set F1 which contains:
courseno --> courselevel,credithours,offeringdept
courseno,secno,semester,year --> noofstudents
roomno,dayshours,semester,year --> instructorssn

Step 4: Check each FD in the set F1 for violation of 3NF, and split table accordingly.

Checking FD courseno --> courselevel,credithours,offeringdept
The FD violates 3NF as its LHS is not a superkey (and RHS is a set of non-key attributes).
The following 3NF table is obtained:

courseno,courselevel,credithours,offeringdept
with FDs
courseno --> courselevel,credithours,offeringdept

Checking FD courseno,secno,semester,year --> noofstudents
FD does not violate 3NF
Checking FD roomno,dayshours,semester,year --> instructorssn
FD does not violate 3NF

Step 5: Finally, add the following table into normalized 3NF table set (obtained by removing RHS attributes of FDs using which we produced a table):

courseno,secno,instructorssn,semester,year,dayshours,roomno,noofstudents
with FDs
courseno,secno,semester,year --> roomno,dayshours
dayshours,roomno,semester,year --> secno,courseno,instructorssn,noofstudents

Attributes

courseno | courselevel | credithours | offeringdept

Functional Dependencies

courseno → courselevel credithours offeringdept

Attributes

courseno | secno | instructorssn | semester | year | dayshours | roomno | noofstudents

Functional Dependencies

courseno secno semester year → roomno dayshours

dayshours roomno semester year → secno courseno instructorssn noofstudents

- Decompondo a Relação para a BCNF, temos:

Step 1. Find merged minimal cover of FDs, which contains:
courseno --> offeringdept,credithours,courselevel
courseno,secno,semester,year --> dayshours,roomno,noofstudents
roomno,dayshours,semester,year --> instructorssn,courseno,secno

Initially rel[1] contains the original table, with the FDs above

Round1: Checking whether table rel[1] is in BCNF

The FD [courseno --> offeringdept,credithours,courselevel] violates BCNF as the LHS is not superkey. Table is split into the two below:

rel[2]= (courseno,offeringdept,credithours,courselevel)
With FDs:

rel[3]= (courseno,secno,instructorssn,semester,year,dayshours,roomno,noofstudents)
With FDs:

Round2: Checking whether table rel[2] is in BCNF

*** Table rel[2] is in BCNF already, send it to output ***

Round3: Checking whether table rel[3] is in BCNF

*** Table rel[3] is in BCNF already, send it to output ***

Attributes

courseno offeringdept credithours courselevel

Functional Dependencies

courseno → offeringdept credithours courselevel

Attributes

courseno secno instructorssn semester year dayshours roomno noofstudents

Functional Dependencies

courseno secno semester year → roomno dayshours

dayshours roomno semester year → secno courseno instructorssn noofstudents

13. Considere o seguinte esquema de relação

    `LOTS(propertyid, countyname, lotno, area, price, taxrate)`

    e conjunto associado de dependências funcionais abaixo:

    `{{propertyid} → {countyname, lotno, area, price, taxrate},`

    `{countyname, lotno} → {propertyid, area, price, taxrate},`

    `{countyname} → {taxrate},`

    `{area → {price}}`

    Determine se a decomposição de `LOTS` apresentada abaixo é uma decomposição de junção sem perdas.

    DICA: Para isso, você precisará usar a função `ljd` da ferramenta SWI-Prolog.

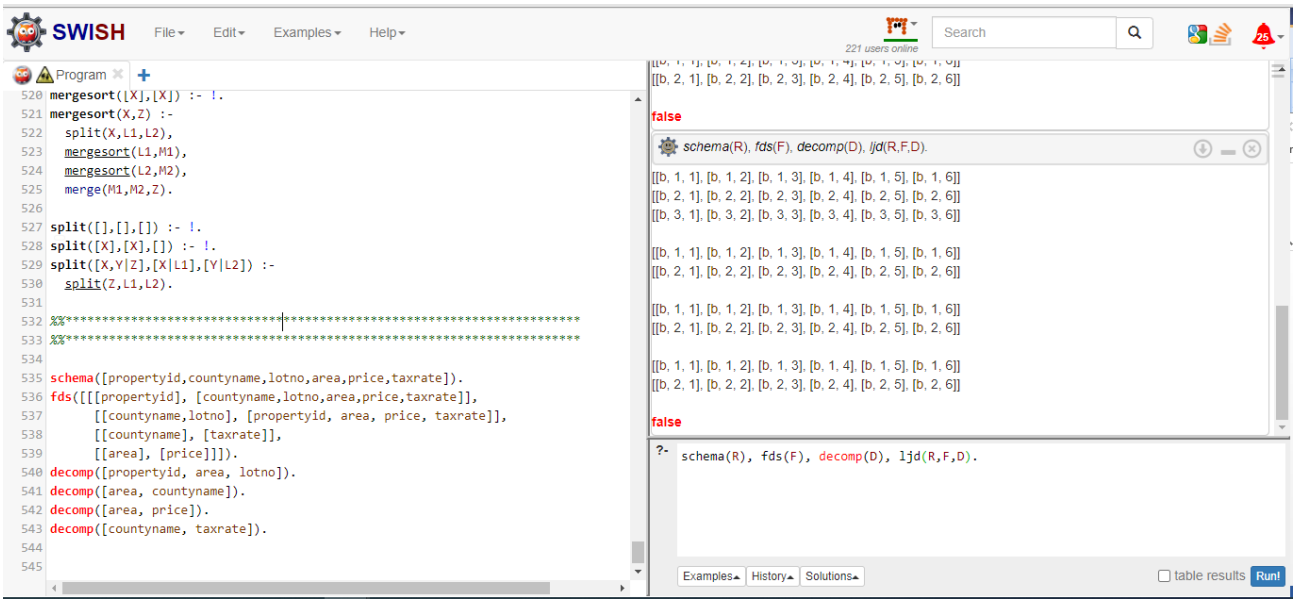
    `LOTS1AX (propertyid, area, lotno)`

    `LOTS1AY (area, countyname)`

    `LOTS1B (area, price)`

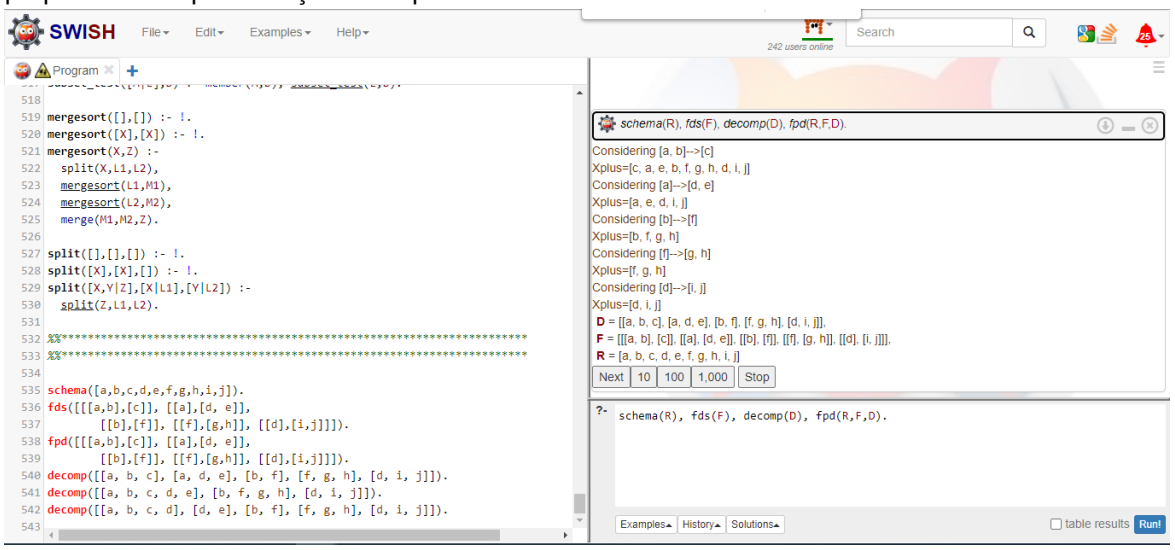
    `LOTS2 (countyname, taxrate)`

    - Conforme visto na imagem abaixo, as decomposições não apresentam junção sem perdas:

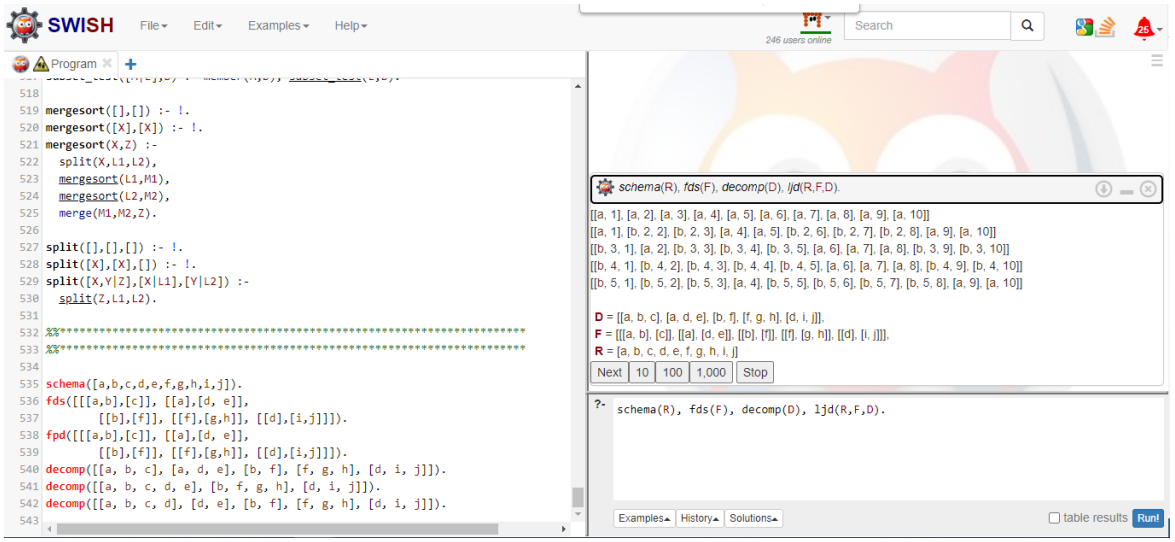


14. Considere a relação universal `R(A, B, C, D, E, F, G, H, I, J)` e o conjunto de dependências funcionais `F = {AB → C, A → DE, B → F, F → GH, D → IJ}`. Determine se cada uma das seguintes decomposições tem:

(a) propriedade de preservação de dependência



(b) propriedade de junção sem perdas em relação a `F`.



DICA: Para isso, você precisará usar as funções `ljd` e `fpd` da ferramenta SWI-Prolog.

14.1.

```
D = {R1,R2,R3,R4,R5};
R1 = {a,b,c},
R2 = {a,d,e},
R3 = {b,f},
R4 = {f,g,h},
R5 = {d,i,j}.
```

14.2.

```
D = {R1,R2,R3};
R1 = {a,b,c,d,e},
R2 = {b,f,g,h},
R3 = {d,i,j}.
```

14.3.

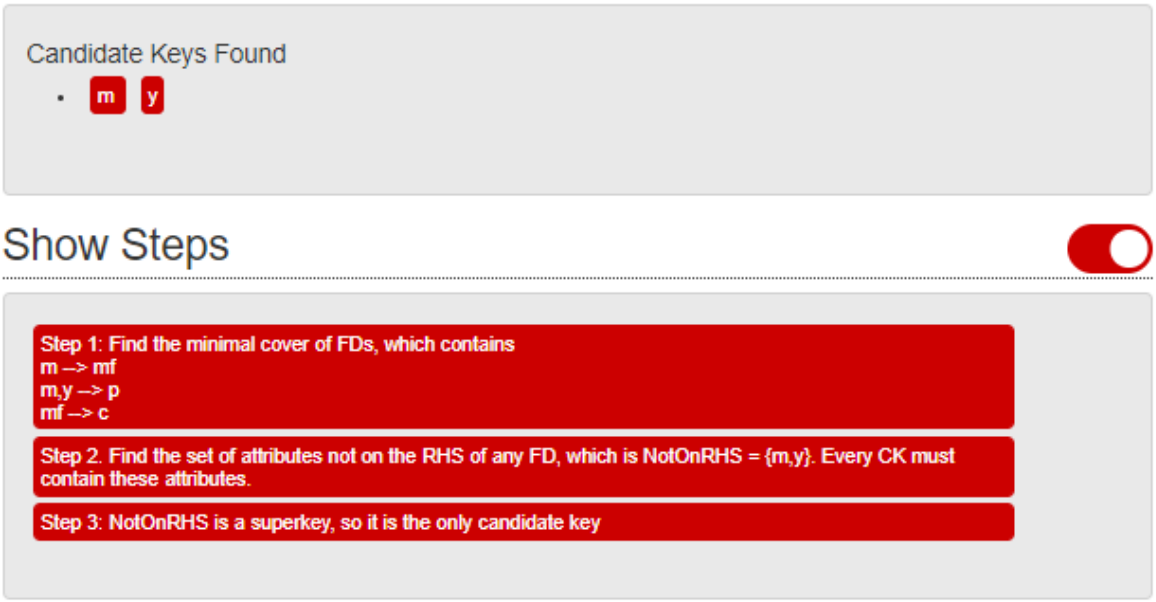
```
D = {R1,R2,R3,R4,R5};
R1 = {a,b,c,d},
R2 = {d,e},
R3 = {b,f},
R4 = {f,g,h},
R5 = {d,i,j}.
```

15. Considere a relação `REFRIG(modelno, year, price, manufacturer, color)` que é abreviada como `REFRIG(m, y, p, mf, c)` e o conjunto de dependências funcionais `F = {{m} → {mf}, {m, y} → {p}, {mf} → {c}}`

15.1. Avalie cada uma das seguintes opções como chaves candidatas:

- `{m}`
- `{m, y}`
- `{m, c}`

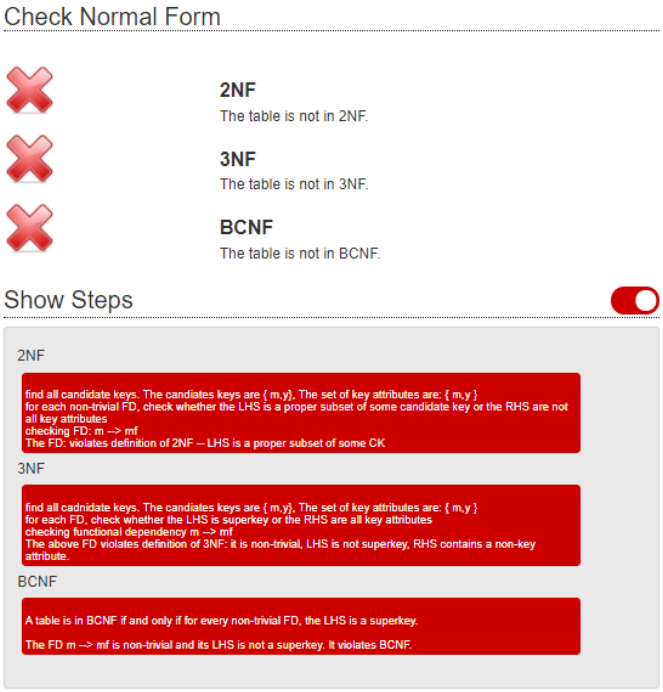
- Somente m e y são candidatas a chave:



15.2. Teste se `REFRIG` está na 3NF?

15.3. Teste se `REFRIG` está na BCNF?

- REFRIG não atende as regras de nenhuma NF:



15.4. Teste se a decomposição abaixo tem a propriedade de junção sem perdas.

```
D = {R1,R2};
R1 = {m,y,p},
R2 = {m,mp,c}.
```

- Para o teste de propriedade de junção sem perda tivemos um retorno FALSE: