Questão 01 - Parte 01

A → C
A → D
E → A
E → H

Show Steps

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

A → C
C A → D
E → A
E → D
E → H

Step 2: Remove trivial FDs (those where the RHS is also in the LHS). We obtain:

A → C
C A → D
E → A
E → D
E → H

Step 3: Minimize LHS of each FD. We obtain:

A → C
A → D
E → A
E → D
E → H

Step 4: Remove redundant FDs (those that are implied by others). We obtain:

Step 4: Remove redundant FDs (those that are implied by others). We obtain:

A → C

A → D

E → A

E → H

Questão 01 - parte2

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

A → C

A → D

E → A

E → H

Step 2: Remove trivial FDs (those where the RHS is also in the LHS). We obtain:

A → C

A → D

E → A

E → H

Step 3: Minimize LHS of each FD. We obtain:

A → C

A → D

E → A

E → H

Step 4: Remove redundant FDs (those that are implied by others). We obtain:

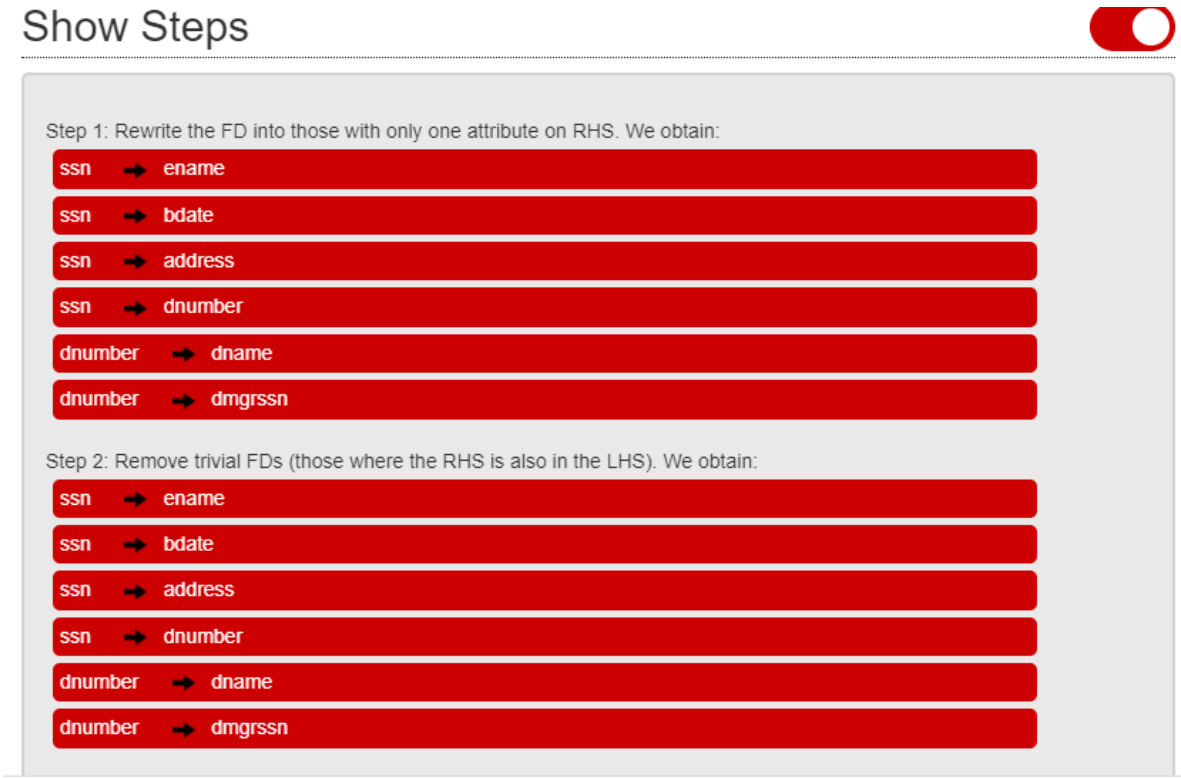
A → C

A → D

E → A

E → H

Questão 02 -


Show Steps
Step 1: Rewrite the FD into those with only one attribute on RHS. We obtain:
ssn → ename
ssn → bdate
ssn → address
ssn → dnumber
dnumber → dname
dnumber → dmgrssn
Step 2: Remove trivial FDs (those where the RHS is also in the LHS). We obtain:
ssn → ename
ssn → bdate
ssn → address
ssn → dnumber
dnumber → dname
dnumber → dmgrssn

Step 3: Minimize LHS of each FD. We obtain:

- ssn → ename
- ssn → bdate
- ssn → address
- ssn → dnumber
- dnumber → dname
- dnumber → dmgrssn

Step 4: Remove redundant FDs (those that are implied by others). We obtain:

- ssn → ename
- ssn → bdate
- ssn → address
- ssn → dnumber
- dnumber → dname
- dnumber → dmgrssn

Questão 3 -

3.1 -

Candidate Keys Found

- A B

Show Steps

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

Step 2. Find the set of attributes not on the RHS of any FD, which is NotOnRHS = {A,B}. Every CK must contain these attributes.

Step 3: NotOnRHS is a superkey, so it is the only candidate key

3.2 -

3.3 -

Step 1: Find the minimal cover of FDs, which contains
A,B --> C
A --> D
A --> 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), }.
The set of key attributes is: { A,B }.

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

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

Questão 04 -

4.1

Candidate Keys Found

- A B D

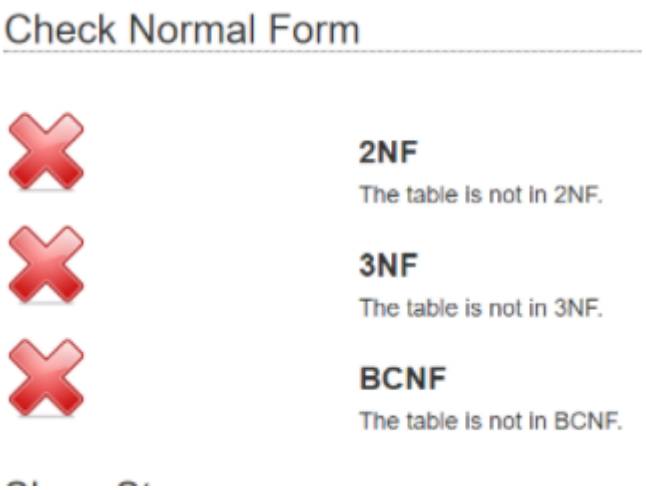
Show Steps

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

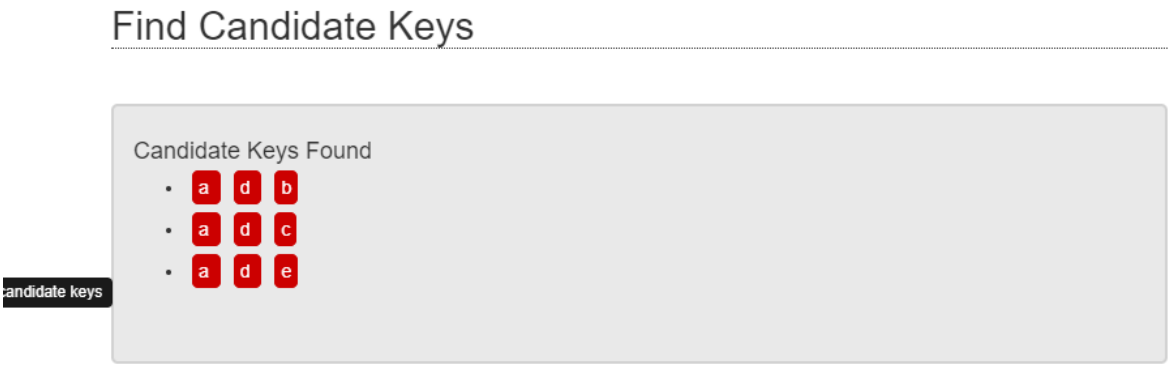
Step 2. Find the set of attributes not on the RHS of any FD, which is NotOnRHS = {A,B,D}. Every CK must contain these attributes.

Step 3: NotOnRHS is a superkey, so it is the only candidate key

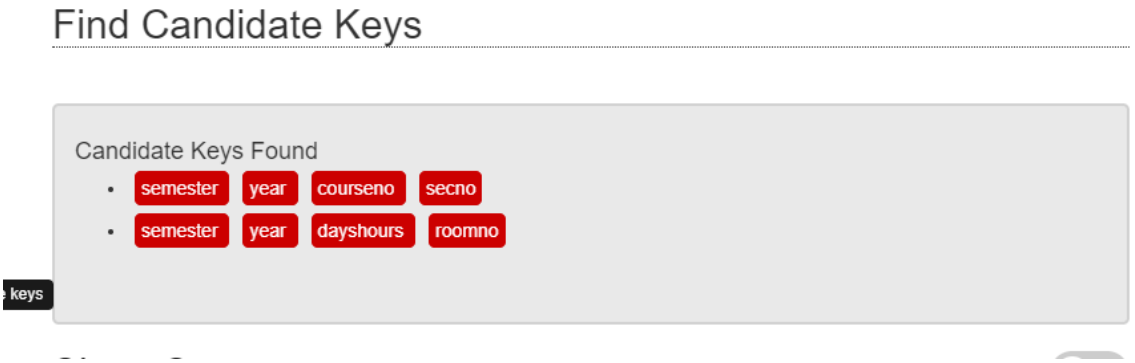
4.2 -



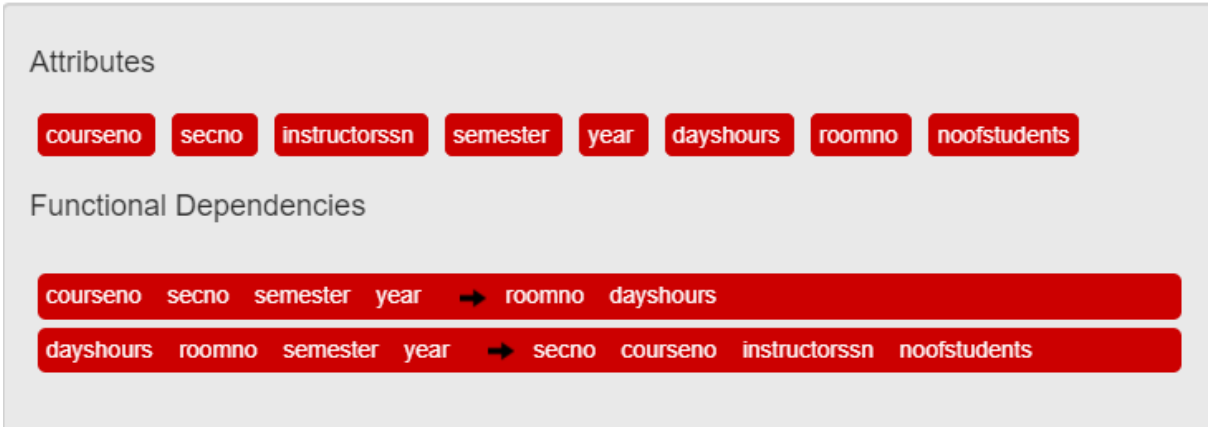
Questão 5



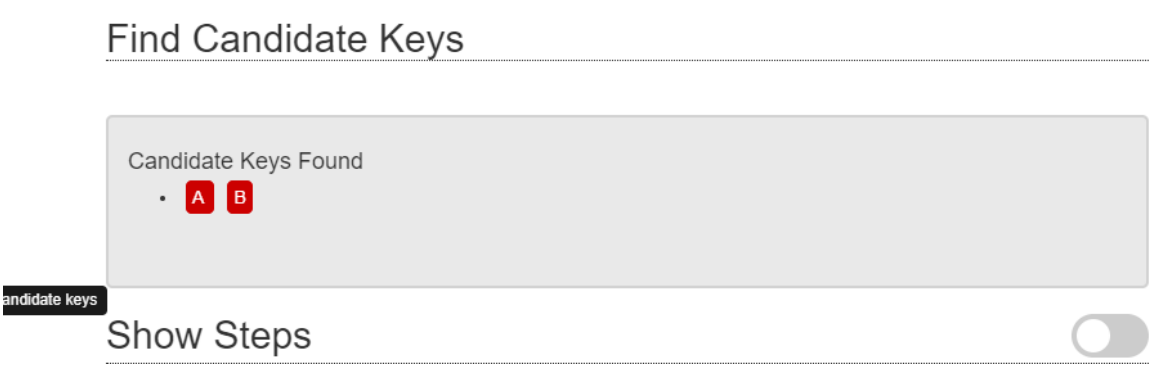
Questão 06 -



Normalização 3NF



Questão 07-



Normalização 3NF

Attributes

A E D

Functional Dependencies

A → D E

ble to 3NF
FDs tributes

B F

Functional Dependencies

B → F

Attributes

F H G

g FDs

F H G

Functional Dependencies

F → H G

Attributes

D J I

Functional Dependencies

D → J I

Attributes

A B C

Attributes

D J I

Functional Dependencies

D → J I

Attributes

A B C

Functional Dependencies

A B → C

Questão 08-

Find Minimal Cover

A B → C

A → D

A → E

B → F

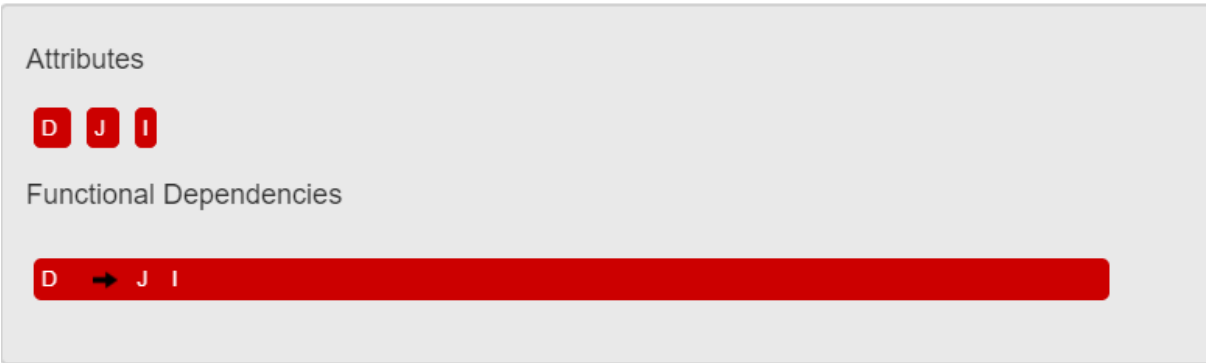
F → G

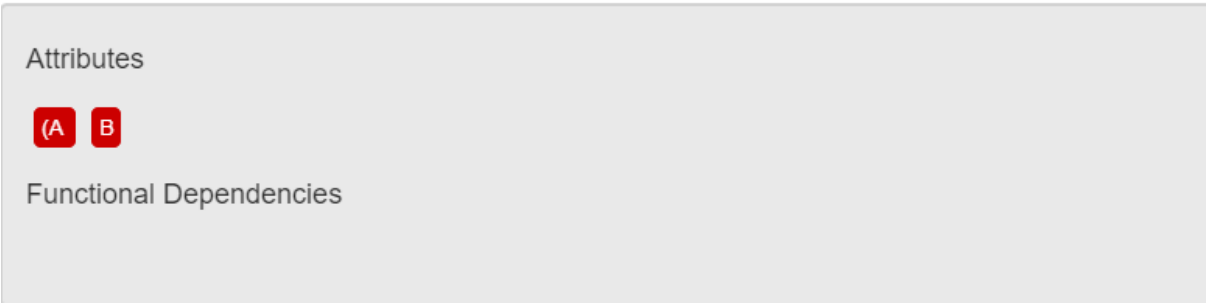
F → H

D → I

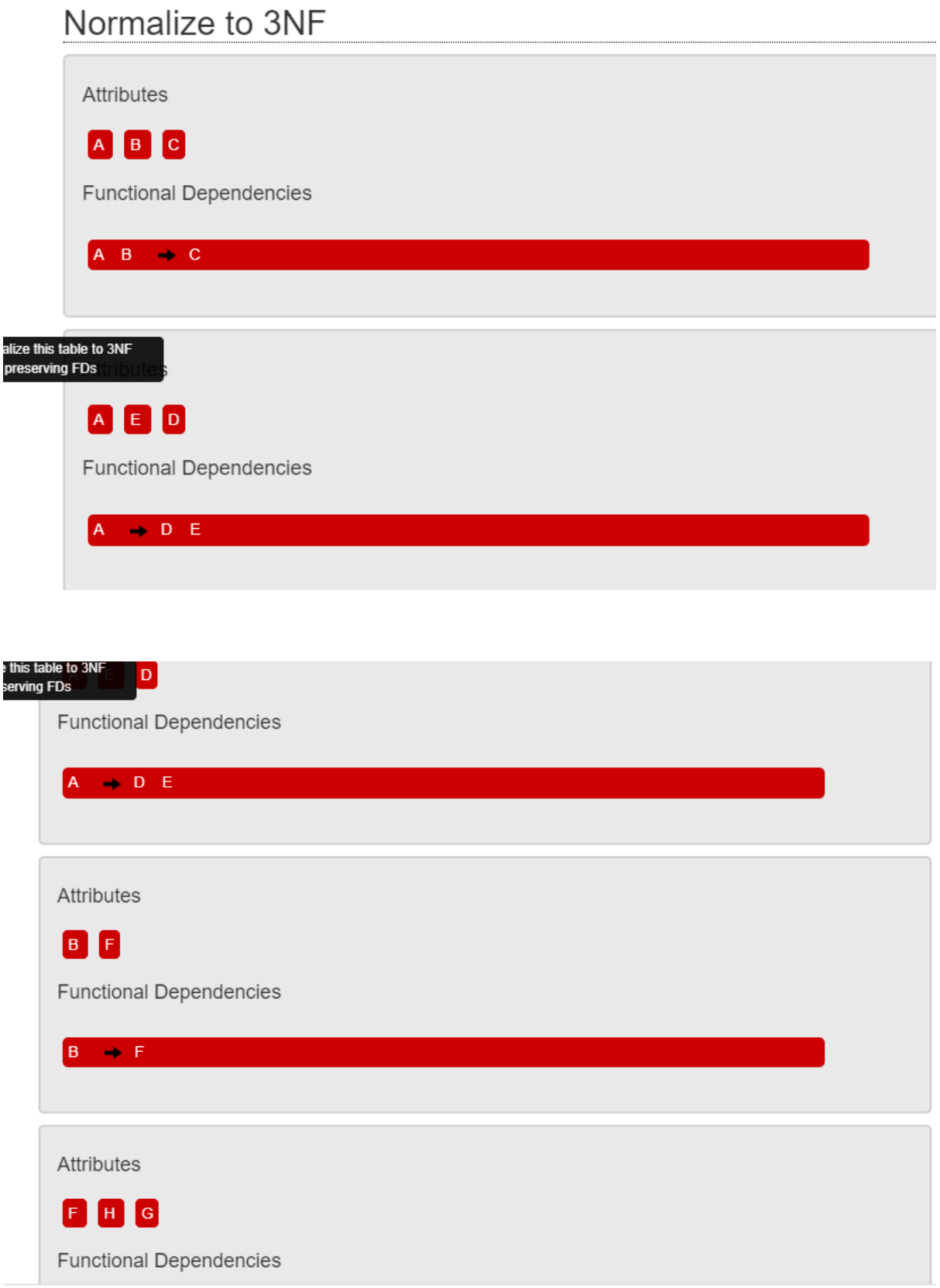
D → J

Normalização 3NF





Questão 08 -


Normalize to 3NF
Attributes
A B C
Functional Dependencies
A B → C
alize this table to 3NF
preserving FDs
A E D
Functional Dependencies
A → D E
this table to 3NF
serving FDs
D
Functional Dependencies
A → D E
Attributes
B F
Functional Dependencies
B → F
Attributes
F H G
Functional Dependencies

B F

Functional Dependencies

B → F

Attributes

F H G

Functional Dependencies

F → H G

Attributes

D J I

Attributes

D J I

Functional Dependencies

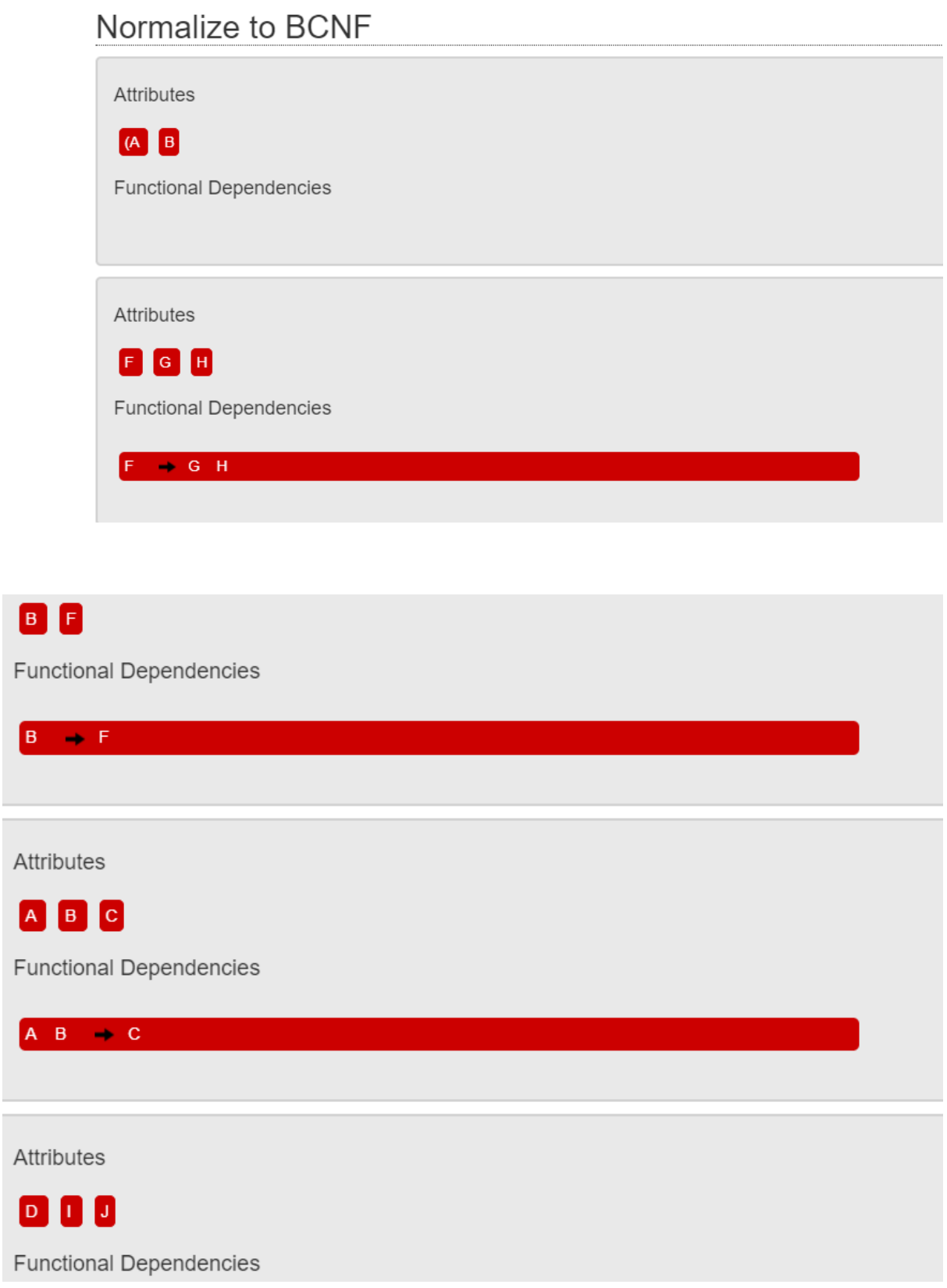
D → J I

Attributes

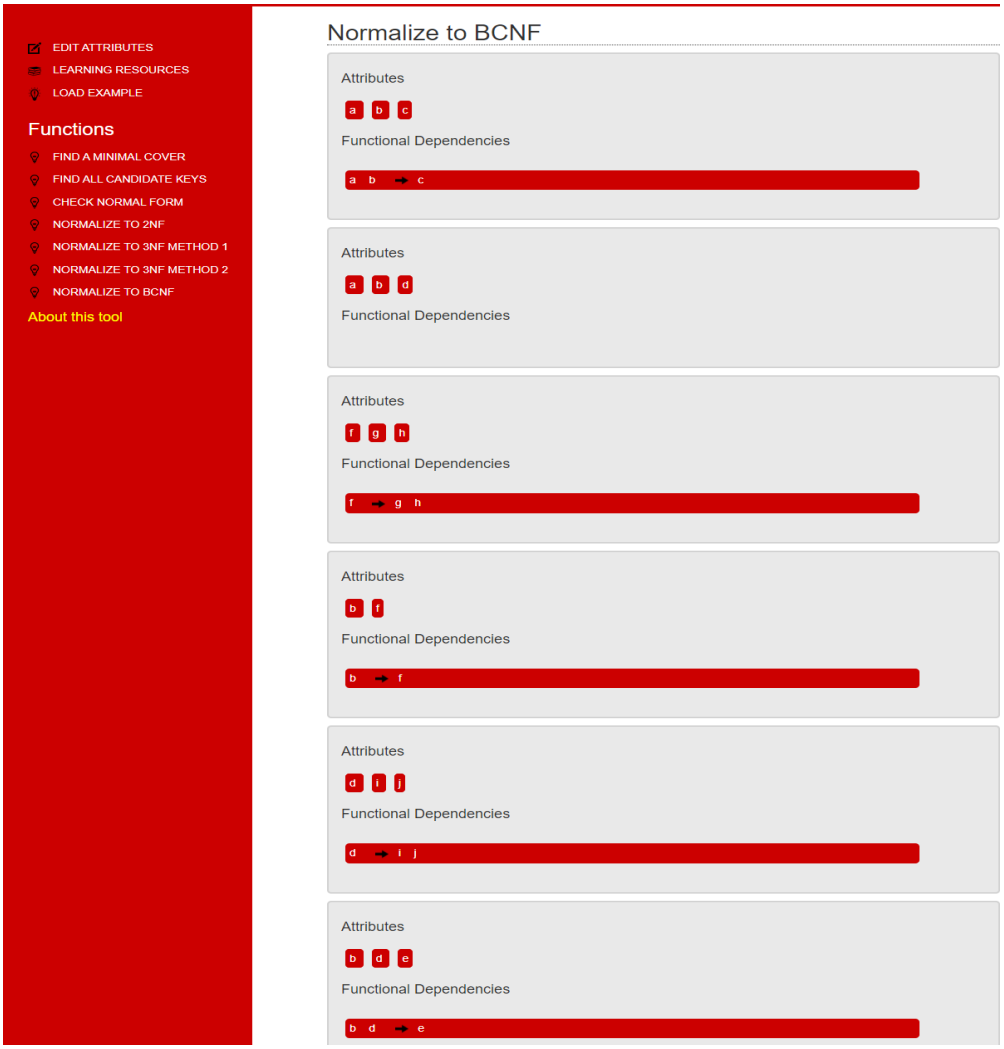
(A B

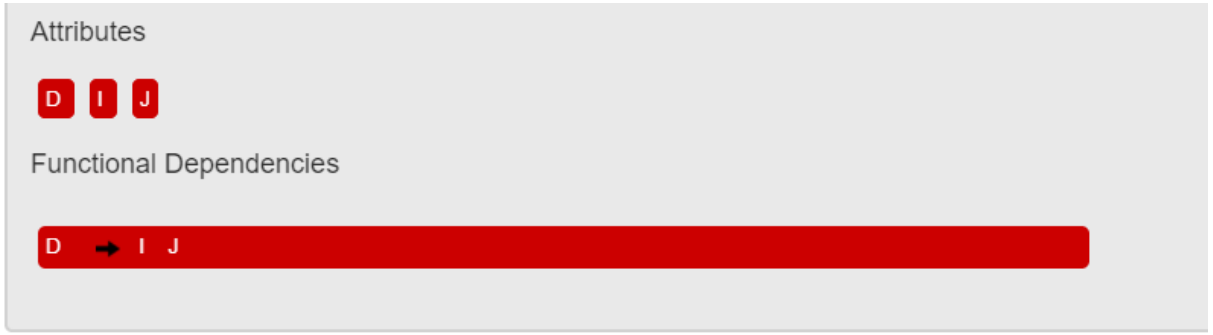
Functional Dependencies

Questão 09 -


Normalize to BCNF
Attributes
(A B
Functional Dependencies
Attributes
F G H
Functional Dependencies
F → G H
B F
Functional Dependencies
B → F
Attributes
A B C
Functional Dependencies
A B → C
Attributes
D I J
Functional Dependencies

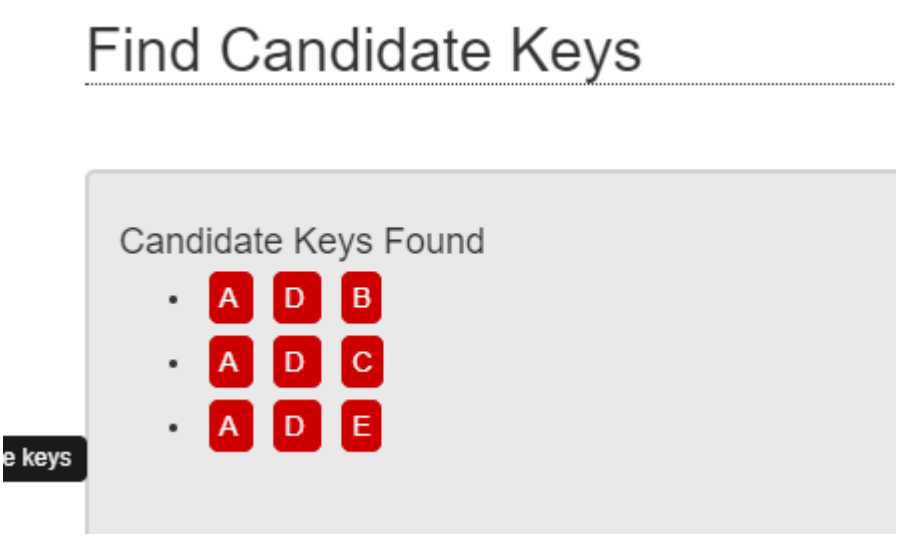
Questão 10 -


EDIT ATTRIBUTES
LEARNING RESOURCES
LOAD EXAMPLE
Functions
FIND A MINIMAL COVER
FIND ALL CANDIDATE KEYS
CHECK NORMAL FORM
NORMALIZE TO 2NF
NORMALIZE TO 3NF METHOD 1
NORMALIZE TO 3NF METHOD 2
NORMALIZE TO BCNF
About this tool
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



Questão 11

Chaves

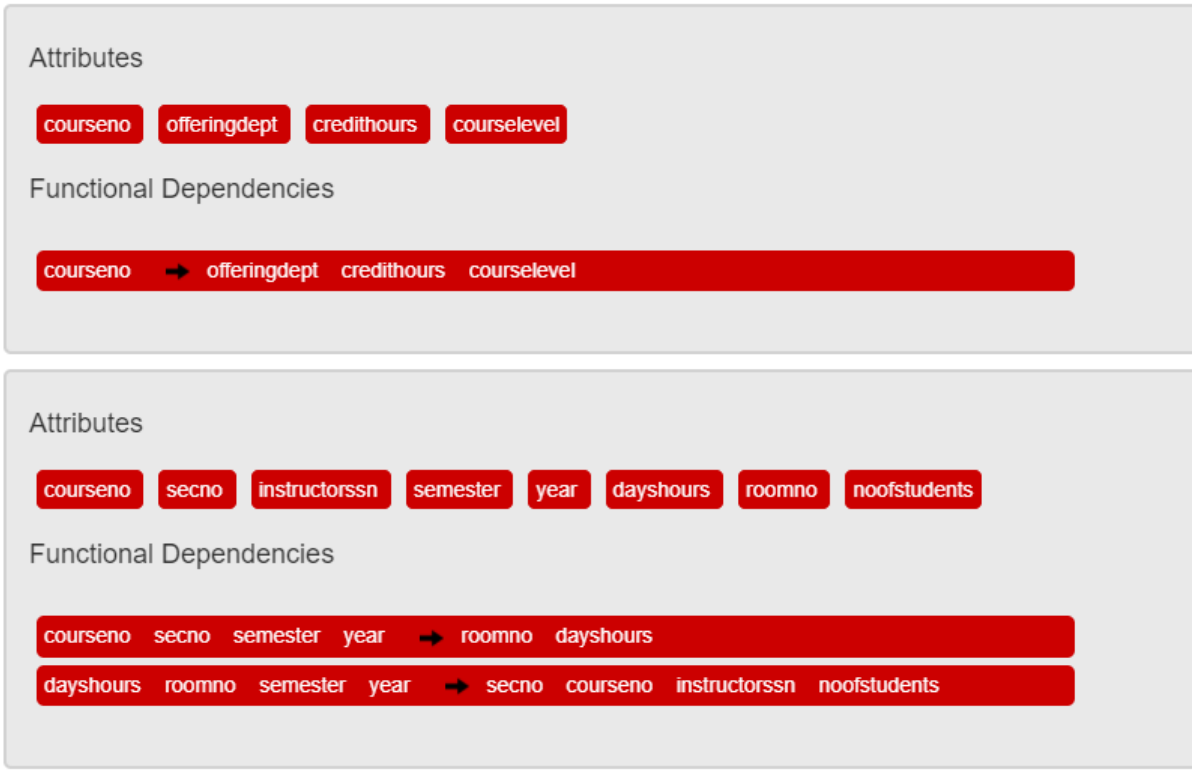


Normalização 3NF

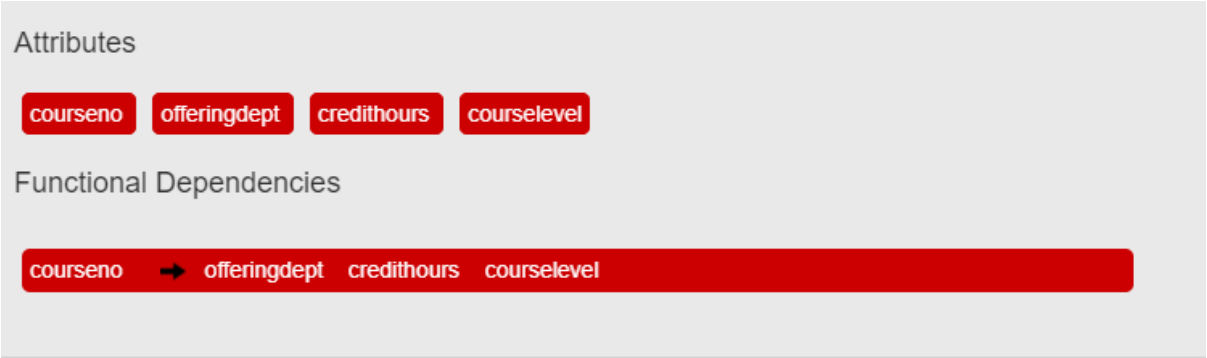
Questão 12 -

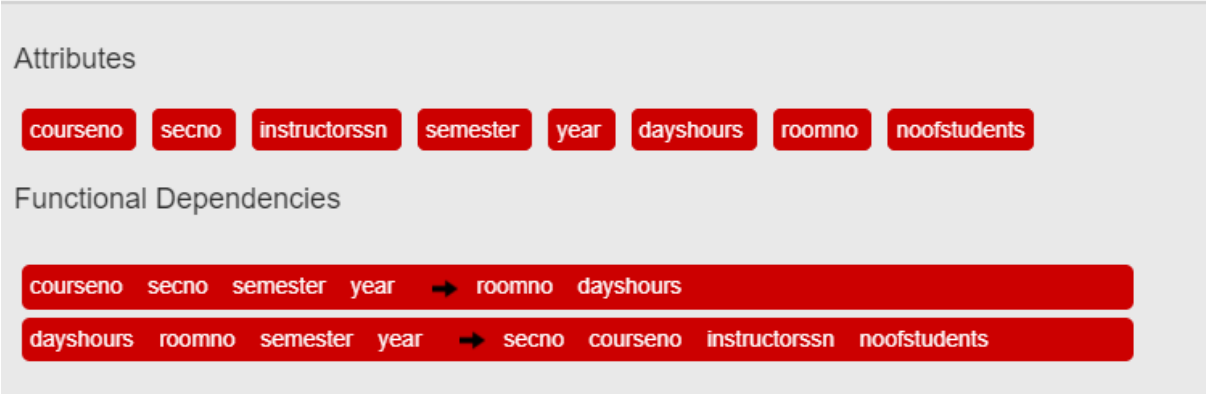
Chaves Principais

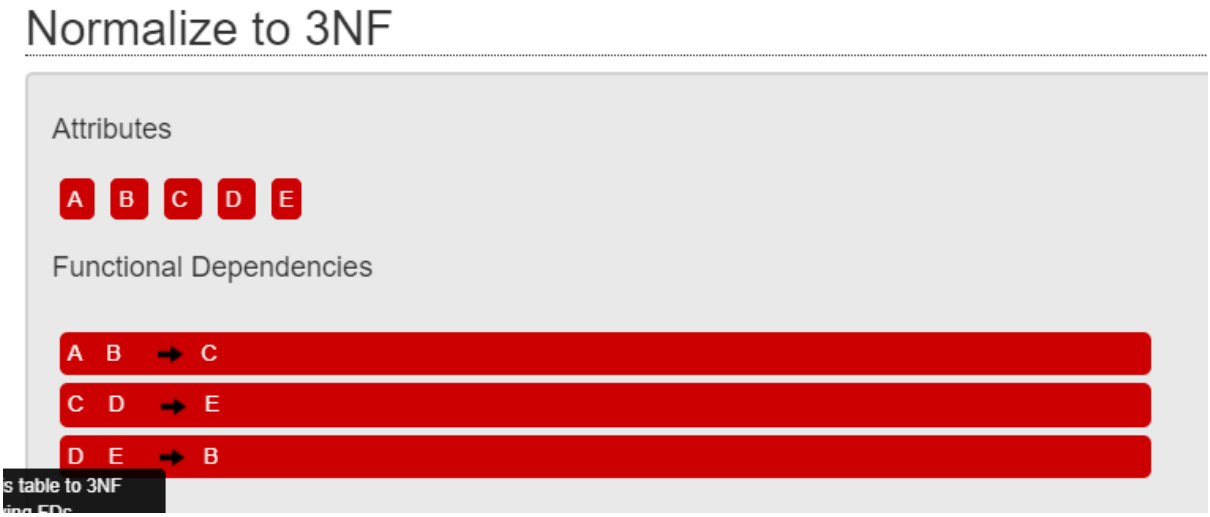
Relação 3NF Método 2



E relação BCNF

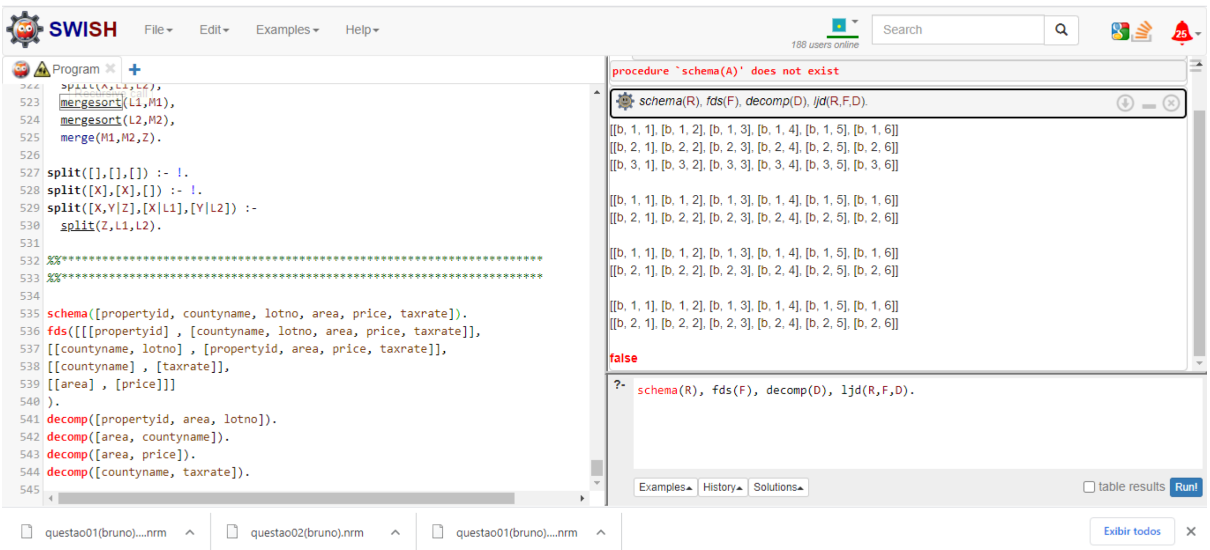




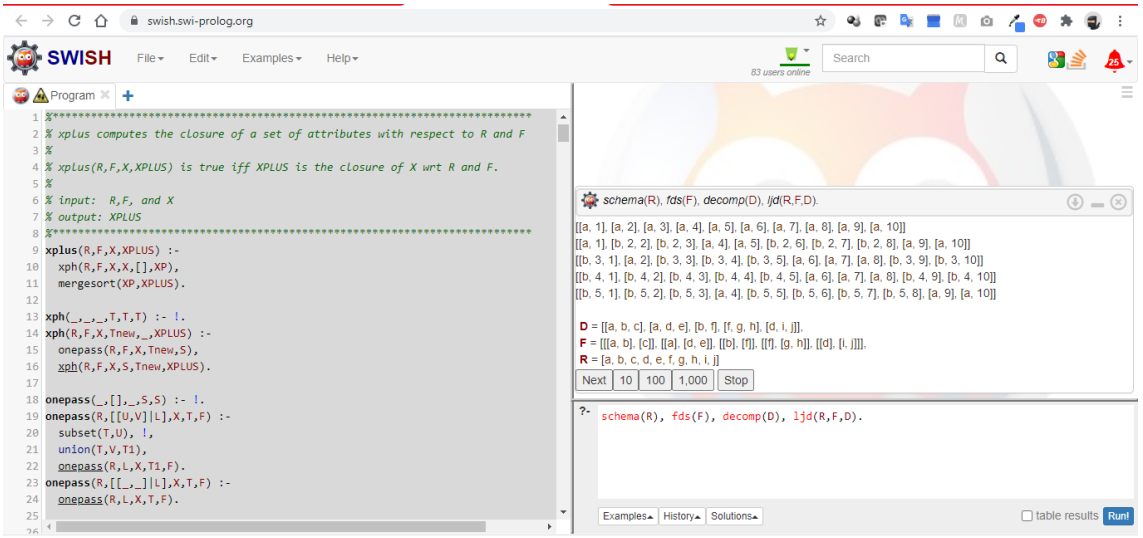


Relações BNCF

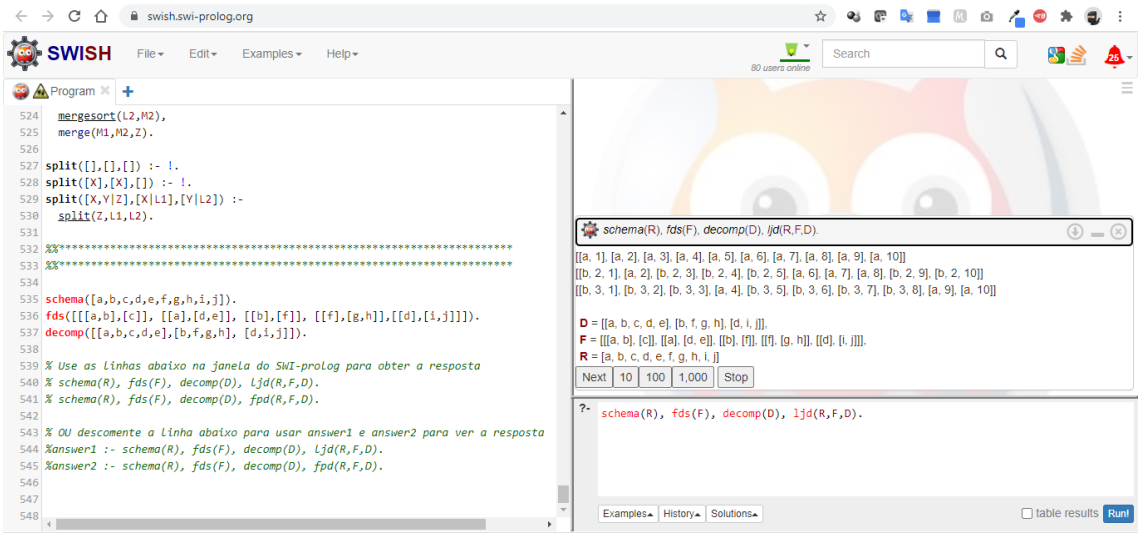
Questão 13 - Falso



Questão 14
parte 1

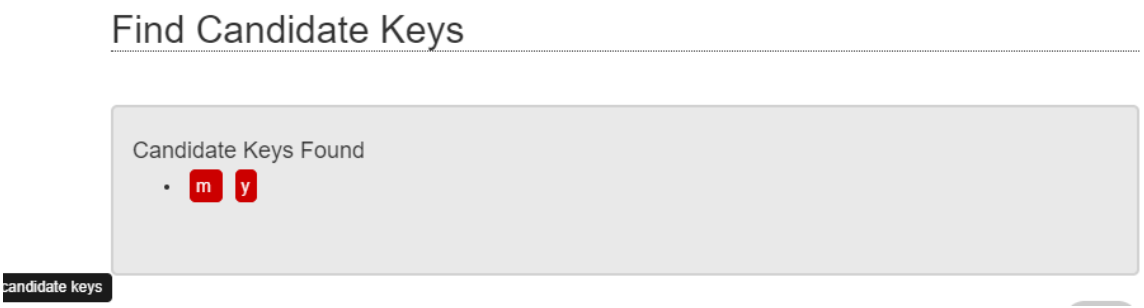


parte 2



Questão 15

chaves Principais

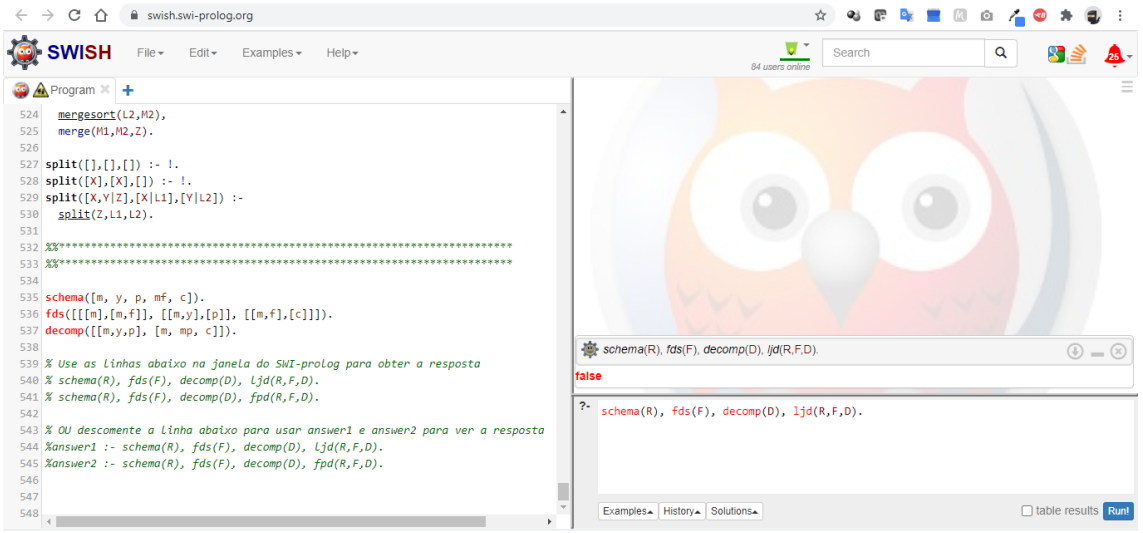


Checar Formas Normalizações

Checar relação BCNF

NF

Teste de Decomposição de Perdas


swish.swi-prolog.org
SWISH
File
Edit
Examples
Help
84 users online
Search
Program
mergesort(L2,M2),
merge(M1,M2,Z).
split([],[],[]) :- !.
split([X],[X],[]) :- !.
split([X,Y|Z],[X|L1],[Y|L2]) :-
split(Z,L1,L2).
%%%
%%%
schema([m, y, p, mf, c]).
fds([[[m],[m,f]], [[m,y],[p]], [[m,f],[c]]]).
decomp([[m,y,p], [m, mp, c]]).
% Use as linhas abaixo na janela do SWI-prolog para obter a resposta
% schema(R), fds(F), decomp(D), ljd(R,F,D).
% schema(R), fds(F), decomp(D), fpd(R,F,D).
% OU descomente a linha abaixo para usar answer1 e answer2 para ver a resposta
%answer1 :- schema(R), fds(F), decomp(D), ljd(R,F,D).
%answer2 :- schema(R), fds(F), decomp(D), fpd(R,F,D).
schema(R), fds(F), decomp(D), ljd(R,F,D).
false
?-
schema(R), fds(F), decomp(D), ljd(R,F,D).
Examples
History
Solutions
table results
Run!