Exercises 2 – Functional dependencies and Normalisation

Bases de Dados, FCT-NOVA

Ano letivo 2015/16

Group 1. Consider the following relation with schema $R(A, B, C, D, E)$:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_2	c_1	d_1	e_1
a_2	b_2	c_1	d_2	e_3
a_2	b_3	c_3	d_2	e_2

Check whether each of the following functional dependencies is satisfied by the relation above.

1. $A \rightarrow C$
2. $B \rightarrow C$
3. $D \rightarrow E$
4. $CD \rightarrow E$

Group 2. Consider schema $R(A, B, C, D, E)$ and the following set of functional dependencies define over R:

$$F = \{A \rightarrow C, AC \rightarrow E, AD \rightarrow B, B \rightarrow ADE, D \rightarrow E\}$$

Using the Armstrong's Axioms (reflexivity, augment, and transitivity), together with the derived rules of union, decomposition and pseudo-transitivity, show that each of the following dependencies belongs to the closure of F. F^+, o fecho de F. Proceed step-by-step, applying only one axiom or rule at a time.

1. $A \rightarrow E$
2. $B \rightarrow C$
3. $ABC \rightarrow A$

4. $AD \rightarrow C$

Group 3. With the set of dependencies F from Group 2, compute the closure of each the following sets of attributes. Proceed step-by-step, using just one functional dependency at a time.

1. D
2. E
3. AD

Group 4. Consider the following set of functional dependencies define over $R(A, B, C, D, E)$:

$$F = \{AB \rightarrow C, CE \rightarrow D, A \rightarrow E\}$$

Check whether each of the following functional dependencies belongs to F^+.

1. $AB \rightarrow D$
2. $AC \rightarrow D$
3. $A \rightarrow C$
4. $A \rightarrow B$
5. $BE \rightarrow D$

Group 5. Compute a canonical cover F_c, for each of the pair schema R and set of dependencies F:

1. $R(A, B, C, D, E)$ and
   $F = \{AB \rightarrow CD, A \rightarrow B, BE \rightarrow DA, E \rightarrow D, C \rightarrow D\}$
2. $R(A, B, C, D, E, G)$ and
   $F = \{ABD \rightarrow CE, BC \rightarrow D, CD \rightarrow E, DE \rightarrow G, A \rightarrow B\}$
3. $R(A, B, C, D, E, F, G, H)$ and
   $F = \{AC \rightarrow G, D \rightarrow EG, BC \rightarrow D, CG \rightarrow BD, AC \rightarrow B, CE \rightarrow AG\}$

Group 6. Consider the schema below, for storing information in a library:

$$Library(book, title, ISBN, copy, branch, time, reader, deliverDate)$$

With the functional dependencies of Table 1:

1. What is the meaning, in plain english, of each of these dependencies?
2. What are the candidate keys of *Library*
3. Single out the dependencies that may cause violation of the Boyce-Codd normal form (BCNF), if any.

Table 1: Functional dependences over *Library*

a.	$book \to title$
b.	$book \to ISBN$
c.	$ISBN \to book$
d.	$copy \to book$
e.	$copy \to branch$
f.	$branch, book \to copy$
g.	$copy, time \to reader$
h.	$branch, book \to deliverDate$

4. Single out the dependencies that may cause violation of the 3rd normal form (3NF).

5. If the schema is not in the BCNF, decompose it into a set of schemas in the BCNF.

6. Decompose *Library*, into a set of schema in the 3NF that preserves dependencies.

Group 7. To store the information of list of candidates to the national election, a table with the following schema was created: Para armazenar informação relativa a listas de candidatos criou-se uma tabela com o seguinte esquema:

$$Lists = \{candidate, party, constituency, ordNum, numDeputies\}$$

As usual, by a list we mean the ordered set of candidates of a political party in a constituency, where *candidate* is the id number of a candidate, *party* and *constituency* is the political party and the constituency in which this person is a candidate, *ordNum* is the order number of that candidate in the list of that party in that constituency, and *numDeputies* is the total number of candidates elected in that constituency. Moreover, ones has to consider the restriction described in Table 2.

Table 2: Restrictions over *Lists*

a.	The total number of candidates is a function of the constituency.
b.	Each candidate cannot run for two different political parties.
c.	Each candidate cannot belong to more than one list of a political party.
d.	In any given list, a candidate can only have one order number.
e.	No two candidates can have the same order number in any given list.

1. Formalise each of the restrictions in Table 2 with a functional dependencies.

Table 3: Functional Dependencies over R

a.	$\{AC \to E, A \to CD, B \to C, BC \to D\}$;
b.	$\{D \to B, CE \to A\}$;
c.	$\{A \to E, BC \to A, DE \to B\}$;
d.	$\{A \to C, BD \to C, AD \to E\}$;
e.	$\{AB \to C, AE \to B, C \to E, E \to DA\}$;

2. With those functional dependencies, compute the candidate keys of *Lists*.
3. Show, with the help of an example, that the schema *Lists* does not prevent redundancies.
4. Propose a decomposition of *Lists* in the Boyce-Codd normal form.

Group 8. For each of the sets of functional dependencies of Table 3, defined over $R(A, B, C, D, E)$:

1. What are the candidate keys?
2. What are the dependencies that violate the BCNF, if any?
3. What are the dependencies that violate the 3NF, if any?
4. If the schema is not in the BCNF, decompose it into one that it is.
5. Decompose it into a set of schema in the 3NF that preserves dependencies, if needed.
6. Is the decomposition into $R1(A, B, C)$ e $R2(C, D, E)$ lossless?