Solutions for Exercise No. 7

25.04.2017

1 Anomalies

Consider the following instance **gameRelease** with relation schema GameRelease(Developer, Game, Platform, ReleaseYear). The primary key is (Game, Platform). Assume that the relation schema and thus, the instance are in 1NF, but not in 2NF.

gameRelease

	Developer	Game	Platform	ReleaseYear
g_1	'Zenimax Online Studios'	'Elder Scrolls Online'	'PC/Mac'	2014
g_2	'Zenimax Online Studios'	'Elder Scrolls Online'	'Xbox'	2015
g_3	'Arkane Studios'	'Dishonored 2'	'PC/Mac'	2016
g_4	'Lucas Arts'	'Monkey Island'	'DOS'	1990

For each of the following tasks, use instance **gameRelease** as given above.

1. Insert anomaly

   (a) Give a DML statement that leads to an *insert* anomaly in instance **gameRelease**.

   ```
   INSERT INTO gameRelease(Game, Platform, ReleaseYear)
   VALUES ('Elder␣Scrolls␣Online', 'PS4', 2015);
   ```

   (b) Explain shortly, but precisely why your statement leads to an *insert* anomaly. State all your assumptions.

   For the game 'Elder Scrolls Online' it is only known that it will be published on platform 'PS4' in 2015, but it is not known who is the developer. The relation schema forces us to specify a *Developer* when inserting a release for a new platform.

2. Update anomaly

   (a) Give a DML statement that leads to an *update* anomaly in instance **gameRelease**.

   ```
   UPDATE gameRelease
   SET Developer = 'Bethesda'
   WHERE Game = 'Elder␣Scrolls␣Online' AND Platform = 'Xbox';
   ```

   (b) Explain shortly, but precisely why your statement leads to an *update* anomaly. State all your assumptions.

   Assumption: functional dependency Game $\rightarrow$ Developer.

   If the developer stored in attribute *Developer* is updated for the game 'Elder Scrolls Online' on platform 'Xbox', the developer for the same game on platform 'PC/Mac' must also be changed in order to not violate the functional dependency.

3. Delete anomaly

   (a) Give a DML statement that leads to a *delete* anomaly in instance **gameRelease**.

   ```
   DELETE FROM gameRelease
   WHERE Platform = 'DOS';
   ```

   (b) Explain shortly, but precisely why your statement leads to a *delete* anomaly. State all your assumptions.

   This delete statement deletes tuple g_4 from instance **gameRelease**. Then, the information that the game 'Monkey Island' was published by 'Lucas Arts' is lost.

2 Relational Database Design

1. Consider the following instance **exhibition**. A painting's exhibition location (attribute *Location*) consists of the museum (attribute *Museum*) and the year of the exhibition (attribute *ExhibitionYear*).

exhibition

	Painter	Painting	Location
e_1	'Miro'	'The Singing Fish'	{('Kunsthaus Zurich', 2015), ('Guggenheim', 2016)}
e_2	'Miro'	'Constellation'	{('Kunsthaus Zurich', 2016)}

Relation schema *Exhibition* has the following functional dependencies:

$$
\begin{aligned}
F_e \;=\; \{ & \\
& \mathit{Painting} \rightarrow \mathit{Painter}, \\
& \mathit{Painting}, \mathit{ExhibitionYear} \rightarrow \mathit{Museum} \\
\} &
\end{aligned}
$$

   (a) Determine the set S_e of all candidate keys for relation schema *Exhibition*.

$$
\begin{aligned}
S_e \;=\; \{ & \\
& \{\mathit{Painting}, \mathit{ExhibitionYear}\} \\
\} &
\end{aligned}
$$

   (b) Determine the highest normal form (no NF, 1NF, 2NF, 3NF) that relation schema *Exhibition* is in. Explain your answer.

   - As the *Location* attribute is a multi-valued attribute, the relation schema is not in 1NF.
   - Relation schema *Exhibition* is not in 2NF. *Painter* is an attribute that is not contained in any candidate key. As $\{\mathit{Painting}, \mathit{ExhibitionYear}\}$ is a candidate key, $\mathit{Painting} \rightarrow \mathit{Painter}$ is a functional dependency from a strict subset of a candidate key,

$\{Painting\} \subset \{Painting, ExhibitionYear\}$, to an attribute, *Painter*, that is not contained in any candidate key. *Painter* is partially dependent on a candidate key and therefore *Exhibition* is not in 2NF.

- As 2NF is a prerequisite for a schema to be in 3NF, *Exhibition* is also not in 3NF.
- The relation schema is not in any NF.

(c) Transform instance **exhibition** into a new instance that satisfies the next higher normal form. Preserve all functional dependencies.

exhibition′

	Painter	Painting	Museum	ExhibitionYear
e'_1	'Miro'	'The Singing Fish'	'Kunsthaus Zurich'	2015
e''_1	'Miro'	'The Singing Fish'	'Guggenheim'	2016
e'_2	'Miro'	'Constellation'	'Kunsthaus Zurich'	2016

(d) Show the minimal set of functional dependencies for the transformed instance.

Same as F_e.

(e) List the candidate keys of the transformed instance.

Same as S_e.

2. Consider the following instance **r** with relation schema R = (A, B, C, D). Assume that all attributes of the relation schema are not multivalued, are not composite, and do not contain nested relations.

r

	A	B	C	D
r_1	'x'	1	'a'	5
r_2	'y'	2	'b'	6
r_3	'x'	3	'a'	5

Relation schema **R** has the following functional dependencies:

$$
\begin{aligned}
F_r \;=\; \{ & \\
& D \;\rightarrow\; C, \\
& C \;\rightarrow\; AD, \\
& B\,D \;\rightarrow\; A, \\
\} &
\end{aligned}
$$

(a) Determine the set S_r of all candidate keys for relation schema R.

$$
\begin{aligned}
S_r \;=\; \{ & \\
& \{B, D\} \\
& \{B, C\} \\
\} &
\end{aligned}
$$

(b) Determine the highest normal form (no NF, 1NF, 2NF, 3NF) that relation schema R is in. Explain your answer.

- Relation schema R is in 1NF because all attributes are atomic (as given in the task description).
- Relation schema R is not in 2NF. A is an attribute that is not contained in any candidate key. As $\{B, C\}$ is a candidate key, $C \rightarrow AD$ is a functional dependency from a strict subset of a candidate key, $\{C\} \subset \{B, C\}$, to an attribute, A, that is not contained in any candidate key. A is partially dependent on a candidate key and therefore R is not in 2NF.
- As 2NF is a prerequisite for a schema to be in 3NF, R is also not in 3NF.
- The highest normal form is 1NF.

(c) Transform instance **r** into new instances that satisfy the next higher normal form. Preserve all functional dependencies.

r′

	B	D
r'_1	1	5
r'_2	2	6
r'_3	3	5

r″

	A	C	D
r''_1	'x'	'a'	5
r''_2	'y'	'b'	6

Note, the resulting normalization depends on the order in which the violating functional dependencies are chosen when performing the normalization algorithm. As the functional dependency $C \rightarrow AD$ also violates 2NF, another possible normalization is $R'(B, C)$ and $R''(A, C, D)$.

(d) Show the minimal set of functional dependencies for the transformed instances.

$$F_{r'} = \{\}$$

$$F_{r''} = \{ D \rightarrow C, \; C \rightarrow AD \}$$

(e) List the candidate keys of the transformed instances.

$$S_{r'} = \{ \{B, D\} \}$$

$$S_{r''} = \{ \{C\}, \{D\} \}$$