Assignment 3

Jens Kristian R. Nielsen & Thomas D. Vinther

4. marts 2019

Consider the relation $\mathcal{R}$, which has attributes that hold schedules of courses and sections at a university (shorthand notation written in parenthesis): $\mathcal{R}$ = Course Number (C), Section Number (SN), Offering Department (D), Course Type (CT), Credit Hours (CH), Course Level (CL), Instructor (I), Semester (S), Year (Y), Day and Hours (DH), Room Number (R), Number of Students Enrolled (NS).

1 Compute the minimal cover of F

$$
\begin{aligned}
F = \{\ \{C\} &\rightarrow \{D, CT, CH, CL, I\},\\
\{C, SN, S, Y\} &\rightarrow \{R, DH, NS, I\},\\
\{R, DH, S, Y\} &\rightarrow \{I, C, SN\},\\
\{I\} &\rightarrow \{D\},\\
\{CT\} &\rightarrow \{CH, CL\},\\
\{D, CT\} &\rightarrow \{I\},\\
\{I, C\} &\rightarrow \{D, CT\}\}
\end{aligned}
$$

We compute the minimal cover of F, using the technique described in the lectures. First we reduce right hand sides and get

Step 1 and 2 single right hand attributes and reducing left hand sides

$$
\begin{aligned}
F' = \{\ \{C\} &\to \{D\},\\
\{C\} &\to \{CT\},\\
\{C\} &\to \{CH\},\\
\{C\} &\to \{CL\},\\
\{C\} &\to \{I\},\\
\{C, SN, S, Y\} &\to \{R\},\\
\{C, SN, S, Y\} &\to \{DH\},\\
\{C, SN, S, Y\} &\to \{NS\},\\
\{C, SN, S, Y\} &\to \{I\},\\
\{R, DH, S, Y\} &\to \{I\},\\
\{R, DH, S, Y\} &\to \{C\},\\
\{R, DH, S, Y\} &\to \{SN\},\\
\{I\} &\to \{D\},\\
\{CT\} &\to \{CH\},\\
\{CT\} &\to \{CL\},\\
\{D, CT\} &\to \{I\},\\
\{I, C\} &\to \{CT\},\\
\{I, C\} &\to \{D\}\ \}
\end{aligned}
$$

We remove the functional dependency with left hand side in a color, due to the FD with the right hand side in the same color.

$$
\begin{aligned}
F'' = \{\ \{C\} &\to \{D\},\\
\{C\} &\to \{CT\},\\
\{C\} &\to \{CH\},\\
\{C\} &\to \{CL\},\\
\{C\} &\to \{I\},\\
\{C, SN, S, Y\} &\to \{R\},\\
\{C, SN, S, Y\} &\to \{DH\},\\
\{C, SN, S, Y\} &\to \{NS\},\\
\{R, DH, S, Y\} &\to \{C\},\\
\{R, DH, S, Y\} &\to \{SN\},\\
\{I\} &\to \{D\},\\
\{CT\} &\to \{CH\},\\
\{CT\} &\to \{CL\},\\
\{D, CT\} &\to \{I\}\}
\end{aligned}
$$

For step 3 consider

$$
\begin{aligned}
\{C\} \to \{I\} \to \{D\} &\overset{\text{transitivity}}{\Rightarrow} \{C\} \to \{D\}\\
\{C\} \to \{CT\} \to \{CL, CH\} &\overset{\text{transitivity}}{\Rightarrow} \{C\} \to \{CL, CH\}
\end{aligned}
$$

In conclusion $\{C\} \to \{D\}$ and $\{C\} \to \{CL\}$ and $\{C\} \to \{CH\}$ are redundant. Which leaves us with the functional dependencies

$$\begin{aligned} F''' = \{\ & \{C\} \to \{CT\}, \\ & \{C\} \to \{I\}, \\ & \{C, SN, S, Y\} \to \{R\}, \\ & \{C, SN, S, Y\} \to \{DH\}, \\ & \{C, SN, S, Y\} \to \{NS\}, \\ & \{R, DH, S, Y\} \to \{C\}, \\ & \{R, DH, S, Y\} \to \{SN\}, \\ & \{I\} \to \{D\}, \\ & \{CT\} \to \{CH\}, \\ & \{CT\} \to \{CL\}, \\ & \{D, CT\} \to \{I\}\} \end{aligned}$$

2

Based on your answer to (1), find the candidate keys of R based on F. Note that C determines $\{CT, I, D, CH, CL\}$ wrt. F''' and the set that C determines is all the "small"functional In accordance with slide 3 we find a candidate key

$$\begin{aligned} C+ &= C, D, CT, CH, CL, I \\ SN+ &= SN \\ D+ &= D \\ CT+ &= CT, CH, CL \\ CH+ &= CH \\ CL+ &= CL \\ I+ &= I, D \\ S+ &= S \\ Y+ &= Y \\ DH+ &= DH \\ R+ &= R \\ NS+ &= NS \end{aligned}$$

We conclude that no singleton can be a candidate key. Consider the only interesting doubles

$$\begin{aligned} D, CT+ &= D, CT, CH, CL, I, \\ C, SN+ &= C, D, CT, CH, CL, I, SN \end{aligned}$$

Note that we cannot achieve S and Y without having them as a part of the key, as such we jump to the 4 t-uples

$$\begin{aligned} C, SN, S, Y+ &= C, D, CT, CH, CL, I, SN, S, Y, R, DH, NS = \mathcal{R} \\ R, DH, S, Y+ &= C, D, CT, CH, CL, I, SN, S, Y, R, DH, NS = \mathcal{R} \end{aligned}$$

In conclusion $\{C, SN, S, Y\}$ and $\{R, DH, S, Y\}$ are our candidate keys.

3

Normalize R to a lossless BCNF decomposition with respect to F. Note that the FD's

$$\{C, SN, S, Y\} \to \{R\}$$
$$\{C, SN, S, Y\} \to \{DH\}$$
$$\{C, SN, S, Y\} \to \{NS\}$$
$$\{R, DH, S, Y\} \to \{C\}$$
$$\{R, DH, S, Y\} \to \{SN\}$$

Are all part of one of the candidate keys, however the following FD's need to be decomposed

$$\{C\} \to \{CT\} \qquad C+ = C, D, CT, CH, CL, I$$
$$\{C\} \to \{I\}$$
$$\{I\} \to \{D\}$$
$$\{CT\} \to \{CH\}$$
$$\{CT\} \to \{CL\}$$
$$\{D, CT\} \to \{I\}$$

$\mathcal{R}_1 = \{C, SN, S, Y, R, DH, NS\}$ Candidate keys : C, SN, S, Y and R, DH, S, Y
$\mathcal{R}_2 = \{C, D, CT, CH, CL, I\}$ Candidate key : C

This decomposition is lossless because $\mathcal{R}_1 \cap \mathcal{R}_2 = \{C\}$ and C is a candidate key in $\mathcal{R}_2$.
Now in $\mathcal{R}_2$ we have the BCNF violations:

$$\{I\} \to \{D\} \qquad I+ = I, D$$
$$\{CT\} \to \{CH\} \qquad CT+ = CT, CH, CL$$
$$\{CT\} \to \{CL\}$$
$$\{D, CT\} \to \{I\} \qquad D, CT+ = I, D, CT, CH, CL$$

So we decompose $\mathcal{R}_2$ into $\mathcal{R}_3 \times \mathcal{R}_4$ with regards to $\{I\} \to \{D\}$

$\mathcal{R}_3 = \{C, CT, CH, CL, I\}$ Candidate key : C
$\mathcal{R}_4 = \{I, D\}$ Candidate key : I

This decomposition is lossless because $\mathcal{R}_3 \cap \mathcal{R}_4 = \{I\}$ and I is a candidate key in $\mathcal{R}_4$.
Now in $\mathcal{R}_2$ we have the BCNF violations:

$$\{CT\} \to \{CH\} \qquad CT+ = CT, CH, CL$$
$$\{CT\} \to \{CL\}$$

Decomposing $\mathcal{R}_3$ w.r.t. $\{CT\} \to \{CH\}$ now yields

$\mathcal{R}_5 = \{C, I, CT\}$ Candidate key : C
$\mathcal{R}_6 = \{CT, CH, CL\}$ Candidate key : CT

This decomposition is lossless because $\mathcal{R}_3 \cap \mathcal{R}_4 = \{CT\}$ and CT is a candidate key in $\mathcal{R}_4$. In total we get

$$\begin{aligned}\mathcal{R} &= \mathcal{R}_1 \times \mathcal{R}_4 \times \mathcal{R}_5 \times \mathcal{R}_6 \\ &= \{C, SN, S, Y, R, DH, NS\} \times \{I, D\} \times \{C, I, CT\} \times \{CT, CH, CL\}\end{aligned}$$

$$\begin{aligned}\mathcal{F}_1 = \{&\{C, SN, S, Y\} \to \{R\}, \\ &\{C, SN, S, Y\} \to \{DH\}, \\ &\{C, SN, S, Y\} \to \{NS\}, \\ &\{R, DH, S, Y\} \to \{C\}, \\ &\{R, DH, S, Y\} \to \{SN\}\} \\ \mathcal{F}_4 = \{&\{I\} \to \{D\}\} \\ \mathcal{F}_5 = \{&\{C\} \to \{I\}, \\ &\{C\} \to \{CT\}\} \\ \mathcal{F}_6 = \{&\{CT\} \to \{CH\}, \\ &\{CT\} \to \{CL\}\}\end{aligned}$$

The decomposition is lossless because each sub decomposition was.

4

Is your BCNF decomposition dependency-preserving? Why?
We lost the FD $\{D, CT\} \to \{I\}$ when we decomposed $\mathcal{R}_2$, so the decomposition is not dependency preserving.

5

Can you produce a dependency-preserving BCNF decomposition of R? Why?
Let R be a BCNF decomposition. Assume that $\mathcal{R}_{contradiction} = \{D, CT, I\}$ is a subset of one of our relations $\mathcal{R}_i$ in a BCNF decomposition, then this relation has functional dependencies $\{D, CT\} \to \{I\}$ and $\{I\} \to \{D\}$ however if D and CT are part of the key of this relation then $\{I\} \to \{D\}$ is a BCNF violation, so I must also be part of the key of $\mathcal{R}_i$ but then $\{D, CT\} \to \{I\}$ is a BCNF violation, which is a contradiction with R being on BCNF. So D,CT and I cannot be in the same relation in a BCNF decomposition, and as such we will loose the functional dependency $\{D, CT\} \to \{I\}$.

6

Synthesize a lossless and dependency-preserving decomposition of R in 3NF. Explain what foreign key constraints you will use to guarantee losslessness.
A FD is in violation of 3NF if the LHS is not a superkey and the RHS is not part of a candidate

key. We use the minimal cover we previously computed.

$\{C\} \to \{CT\}$	3NF violation
$\{C\} \to \{I\}$	3NF violation
$\{C, SN, S, Y\} \to \{R\}$	LHS super key
$\{C, SN, S, Y\} \to \{DH\}$	LHS Super key
$\{C, SN, S, Y\} \to \{NS\}$	LHS Super key
$\{R, DH, S, Y\} \to \{C\}$	LHS Super key
$\{R, DH, S, Y\} \to \{SN\}$	LHS Super key
$\{I\} \to \{D\}$	3NF violation
$\{CT\} \to \{CH\}$	3NF violation
$\{CT\} \to \{CL\}$	3NF violation
$\{D, CT\} \to \{I\}\}$	3NF violation

We decompose according to the 3NF synthesis procedure listed on slide 16 from the feb 26. lecture.

$\mathcal{R}_1 = \{C, CT\}$	Foreign key: C from $\mathcal{R}_3$
$\mathcal{R}_2 = \{C, I\}$	Foreign key: C from $\mathcal{R}_3$
$\mathcal{R}_3 = \{C, SN, S, Y, R\}$	Includes the candidate key
$\mathcal{R}_4 = \{C, SN, S, Y, DH\}$	Foreign key: C, SN, S, Y from $\mathcal{R}_3$
$\mathcal{R}_5 = \{C, SN, S, Y, NS\}$	Foreign key: C, SN, S, Y from $\mathcal{R}_3$
$\mathcal{R}_6 = \{R, DH, S, Y, C\}$	Foreign key: S, Y, R from $\mathcal{R}_3, DH \in \mathcal{R}_4$
$\mathcal{R}_7 = \{R, DH, S, Y, SN\}$	Foreign key: S, Y, R from $\mathcal{R}_3 DH \in \mathcal{R}_4$
$\mathcal{R}_8 = \{I, D\}$	Foreign key: I from $\mathcal{R}_2$
$\mathcal{R}_9 = \{CT, CH\}$	Foreign key: CT from $\mathcal{R}_1$
$\mathcal{R}_{10} = \{CT, CL\}$	Foreign key: CT from $\mathcal{R}_1$
$\mathcal{R}_{11} = \{D, CT, I\}$	Foreign key: CT from $\mathcal{R}_1$, and D from $\mathcal{R}_8$

This decomposition is clearly lossless and functional dependency preserving because each FD gets its own relation to live in, and every relation has a foreign key pointing back to $\mathcal{R}_3$ that is our candidate key, so determines everything.

7

Suggest a way to enforce all dependencies in F via integrity constraints in SQL DDL (e.g., PRIMARY KEY, FOREIGN KEY, UNIQUE). Specifically, go through each dependency in the minimal cover of F and explain how it is now enforced.

```
CREATE TABLE R3 (
Course_number INT,
Section_number INT,
Semester INT,
Year YEAR,
Room_number INT,
```

```
PRIMARY KEY (Course_number,Section_number,Semester,Year) );

CREATE TABLE R1 (
Course_number INT,
Course_type VARCHAR(32),
FOREIGN KEY (Course_number)
    REFERENCES R3 (Course_number),
PRIMARY KEY (Course_number) );

CREATE TABLE R2 (
Course_number INT,
Instructor VARCHAR(32),
FOREIGN KEY (Course_number)
    REFERENCES R3 (Course_number),
PRIMARY KEY (Course_number) );

CREATE TABLE R4 (
Course_number INT,
Section_number INT,
Semester INT,
Year YEAR,
Day_and_hours VARCHAR(32),
FOREIGN KEY (Course_number, Section_number, Semester, Year)
    REFERENCES R3 (Course_number, Section_number, Semester, Year),
PRIMARY KEY (Course_number) );

CREATE TABLE R5 (
Course_number INT,
Section_number INT,
Semester INT,
Year YEAR,
Number_of_students_enrolled VARCHAR(32),
FOREIGN KEY (Course_number, Section_number, Semester, Year)
    REFERENCES R3 (Course_number, Section_number, Semester, Year),
PRIMARY KEY (Course_number) );

CREATE TABLE R6(
Room_number INT,
Day_and_hours VARCHAR(32),
Semester INT,
Year YEAR,
Course_number INT,
FOREIGN KEY(Day_and_hours)
    REFERENCES R4 (Day_and_hours),
FOREIGN KEY(Semester, Year, Room_number)
    REFERENCES R3 (Semester,Year, Room_number),
PRIMARY KEY (Room_number,Day_and_hours,Semester,Year)
);
```

```
CREATE TABLE R7(
Room_number INT,
Day_and_hours VARCHAR(32),
Semester INT,
Year YEAR,
Section_number INT,
FOREIGN KEY(Day_and_hours)
    REFERENCES R4 (Day_and_hours),
FOREIGN KEY(Semester, Year, Room_number)
    REFERENCES R3 (Semester,Year,Room_number),
PRIMARY KEY (Room_number,Day_and_hours,Semester,Year)
);

CREATE TABLE R8(
Instructor VARCHAR(32),
Department VARCHAR(32),
FOREIGN KEY (Instructor)
    REFERENCES R2 (Instructor),
PRIMARY KEY (Instructor)
);

CREATE TABLE R9(
Course_type VARCHAR(32),
Credit_hours INT,
FOREIGN KEY (Course_type)
    REFERENCES R1 (Course_type),
PRIMARY KEY (Course_type)
);

CREATE TABLE R10(
Course_type VARCHAR(32),
Course_level VARCHAR(32),
FOREIGN KEY (Course_type)
    REFERENCES R1 (Course_type),
PRIMARY KEY (Course_type)
);

CREATE TABLE R11(
Department VARCHAR(32),
Course_type VARCHAR(32),
Instructor VARCHAR(32),
FOREIGN KEY (Course_type)
    REFERENCES R1 (Course_type),
FOREIGN KEY (Instructor)
    REFERENCES R8 (Instructor),
PRIMARY KEY (Department)
);
```

Explain how the dependencies from the minimal cover of F is enforced. From the way we have constructed our SQL DDL with constraints. It is fairly obvious that we uphold all the

functional dependencies of our minimal cover. Since we made each left hand side a primary key in the table dedicated to each functional dependency, and a foreign key chain that leads back to the candidate key table R3. F.ex. R8 that represents $\{I\} \rightarrow \{D\}$ has I as the primary key and collects I from R2 via a foreign key constraint, which in turn represents $\{C\} \rightarrow \{I\}$ and collects C from the candidate key R3, via a foreign key constraint as well.

8

Assuming you have answered Question 7 correctly, some dependency among attributes within a table, which we call child table, must be enforced using a foreign key constraint to a parent table (if not, you might consider this as a hint). Write a trigger that enforces that same dependency, in case of an insertion to the child table.

We wish to make a trigger that takes care of the foreign key constraint when inserting into a child table. As an example we enforce this on table R1 with the following trigger:

```
delimiter $$
CREATE TRIGGER opgave8
    BEFORE INSERT ON R1
    FOR EACH ROW
    BEGIN
        If new.Course_number NOT IN (SELECT Course_number FROM R3)
        THEN SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot insert into table, foreign key does not exist';
    END IF;
END;
$$
delimiter ;
```

This trigger simply checks if the foreign key already exists, if it doesn't SQL throws an error when trying to insert into the child table.