# Database Systems, CSCI 4380-01
# Homework # 2 Answers
# Due Thursday September 20, 2018 at 11:59:59 PM

**Database Description.** Suppose you are given the following database for keeping track of grades in this course. The data model from Homework #1 is significantly simplified where all gradable items (hw,quiz, exams) are combined into a single relation. Similarly all grades are also combined into a single relation).

```
students(rin, fname, lname, email, optin_date, optout_date)
gradables(gid, gtype, label, given_date, due_date, maxgrade, points, nextg_id)
grades(rin, gid, submission_date, grade)
```

**Question 1.** Write the following queries using relational algebra using any operator that you wish:

(a) Return the RIN of all students who missed a homework that was due during their opt-in period. Return the `gid` of the corresponding missed homeworks. Remember if there is no opt-out date, all homeworks after opt-in date are required.

**Answer.**

$$\begin{aligned} R1 &= \Pi_{rin,gid}(GRADABLES \bowtie_{C1} STUDENTS) \\ Result &= \Pi_{rin}\ (R1 - (\Pi_{rin,gid} GRADES)) \end{aligned}$$

where $C1 = optin\_date <> NULL$ and $due\_date >= optin\_date$ and $(optout\_date = NULL$ or $due\_date <= optout\_date)$.

(b) Find the RIN, first and last name of all students who had the highest grades for an exam (i.e. gtype `'exam'` or `'finalexam'`). Also return the `gid` and `label` of the exams they got the highest grades in.

**Answer.**

$$\begin{aligned} R0 &= \sigma_{gtype='exam' \text{ or } gtype='finalexam'}(GRADABLE) \\ R1 &= \Pi_{rin,gid,grade,label}(GRADES \bowtie R0)) \\ R2[rin1, gid1, grade1, label1] &= R1 \\ R3 &= \Pi_{rin,gid,label}(R1 \bowtie_{gid=gid1 \text{ and } grade1>grade} R2) \\ R4 &= (\Pi_{rin,gid,label} R1) - R3 \\ Result &= \Pi_{rin,gid,label,fname,lname}(R4 \bowtie STUDENTS) \end{aligned}$$

All rin, gid pairs in R3 have at least one exam for the same gid with a higher grade. Our question asks us for the remaining students, who have no assignment with higher grade which must be the max grade, which is returned in R4. The remainder is for finding the names of these students.

**Question 2.** For each of the following new relations:

(1) list all the relevant functional dependencies based on the explanations below,
(2) find all keys based on your functional dependencies,
(3) discuss whether the relation is in BCNF (Boyce-Codd Normal Form) or not, explain why or why not.
(4) discuss whether the relation is in 3NF (Boyce-Codd Normal Form) or not, explain why or why not.

(a) The system keeps track of multiple submissions for the same homework gradable like submitty in a relation called `submissions`:

`submissions(gid, rin, filename, attemptno, submission_datetime, isactive, totalruntime)`

Each student, gradable and specific attempt corresponds to a specific filename. Each filename corresponds to a specific student, gradable and attempt. For each filename, there is a specific submission_datetime, isactive value and totalruntime value.

**Answer.**

(1)

rin, gid, attemptno → filename
filename → rin, gid, attemptno
filename → submission_datetime, isactive, totalruntime

(2)

Keys: (rin, gid, attemptno) or filename

(3-4)

This is in BCNF and 3NF because all functional dependencies have a superkey on the left.

(b) Homeworks, quizzes and exams have individual questions. We will store the details of grades of each part separately using a relation called `grade_details`:

`grade_details(rin, gid, partno, topic, maxpoints, pointsearned)`

For each gradable (gid) and part, there is a maxpoints value. For each gradable, part and student, there is pointsearned. Each gradable part may have multiple topics.

**Answer.**

(1)

gid, partno → maxpoints
rin, gid, partno → pointsearned

(2)

Key: rin, gid, partno, topic

(3-4)

This is not in BCNF or 3NF because none of the functional dependencies have a superkey on the left. Neither maxpoints or pointsearned are prime attributes either (for 3NF).

**Question 3.** Given the following relation, functional dependencies and decomposition, answer the following questions:

Relation $R(A, B, C, D, E, F)$ with $\mathcal{F} = \{AB \to F, BD \to C, CE \to F, F \to D\}$

Decomposition: $R1(A, B, D)$, $R2(A, B, C, E)$, $R3(B, D, E, F)$

(a) Is this decomposition lossless? Show yes or no using Chase algorithm.

**Answer.** Construct the table for Chase algorithm.

| | | | | | |
|---|---|---|---|---|---|
| a | b | c1 | d | e1 | f1 |
| a | b | c | d2 | e | f2 |
| a3 | b | c2 | d | e | f |

Given A,B, we can get D. Hence, we can equalize d2 to d.

| | | | | | |
|---|---|---|---|---|---|
| a | b | c1 | d | e1 | f1 |
| a | b | c | d | e | f2 |
| a3 | b | c2 | d | e | f |

Given B,D, we can get C, and equalize c2 to c.

| | | | | | |
|---|---|---|---|---|---|
| a | b | c1 | d | e1 | f1 |
| a | b | c | d | e | f2 |
| a3 | b | c | d | e | f |

Finally, given C,E, we can get F, equalize f2 to f.

| | | | | | |
|---|---|---|---|---|---|
| a | b | c1 | d | e1 | f1 |
| a | b | c | d | e | f |
| a3 | b | c | d | e | f |

Given the second row has no subscript, this decomposition is lossless.

(b) Is this decomposition dependency preserving? Show your work.

Note: two show that two sets of functional dependencies, $F_1$ and $F_2$ are equivalent, it is sufficient to show that (1) all functional dependencies in $F_1$ are implied by $F_2$, and (2) all all functional dependencies in $F_2$ are implied by $F_1$.

**Answer.** First, we find the projection of the above functional dependencies to each decomposed relation.

Relation $R(A, B, C, D, E, F)$ with $\mathcal{F} = \{AB \to F, BD \to C, CE \to F, F \to D\}$

Decompositions:

$R1(A, B, D)$, $\mathcal{F}_1 = \{AB \to D\}$ (projection to R1)
$R2(A, B, C, E)$, $\mathcal{F}_2 = \{AB \to C\}$ (projection to R2)
$R3(B, D, E, F)$, $\mathcal{F}_3 = \{F \to D, BDE \to F\}$ (projection to R3)

We then look if $\mathcal{F}' = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 = \{AB \to D, AB \to C, F \to D, BDE \to F\}$ is equivalent to original set $\mathcal{F}$.

Given all functional dependencies in $\mathcal{F}'$ were implied by the original set, we only need to check if all the functional dependencies in the original set are implied by $\mathcal{F}'$.

$AB \to F$, given $\mathcal{F}'$, $AB^+ = \{A, B, C, D\}$ and since F is not in the closure, this f.d. is not implied by $\mathcal{F}'$. Hence, this is a lossly decomposition. It is sufficient to find one counterexample, so you can stop here. We test the rest for exercise.

$BD \to C$, given $\mathcal{F}'$, $BD^+ = \{B, D\}$, and $C$ is not in the closure, this f.d. is not implied by $\mathcal{F}'$.

$CE \to F$, given $\mathcal{F}'$, $CE^+ = \{C, E\}$. Given no $F$, this f.d. is not implied by $\mathcal{F}'$.

$F \to D$, already in $\mathcal{F}'$, so it is implied.

**Question 4.** Given the following relation, use BCNF decomposition to convert it to relations in BCNF.

$R(A, B, C, D, E)$ $\mathcal{F} = \{AB \rightarrow C, C \rightarrow E\}$

**Answer.**

Key: ABD. Not in BCNF. Both functional dependencies violate it.

Multiple ways to solve this depending on which functional dependency you choose to take out first.

**Solution 1:**

Decompose using $C \rightarrow E$, $C^+ = \{C, E\}$:

$R1(C, E)$ with $\{C \rightarrow E\}$. Key: C, in BCNF!

$R2(A, B, C, D)$ with $\{AB \rightarrow C\}$, Key: ABD. Still not in BCNF. We decompose this relation further.

$R21(A, B, C)$ with $\{AB \rightarrow C\}$, Key: AB. In BCNF.

$R22(A, B, D)$ with $\{\}$, Key: ABD. In BCNF.

Final solution: (C,E), (A,B,C), (A,B,D)

**Solution 2:**

Decompose using $AB \rightarrow C$, $AB^+ = \{A, B, C, E\}$

$R1(A, B, C, E)$ with $\{AB \rightarrow C, C \rightarrow E\}$, key: AB, not in BCNF as $C \rightarrow E$ violates it.

$R2(A, B, D)$ with $\{\}$, key: ABD. In BCNF.

We need to decompose R1 further, using $C \rightarrow E$, we get:

$R11(C, E)$ with $\{C \rightarrow E\}$. Key: C, in BCNF! $R12(A, B, C)$ with $\{AB \rightarrow C\}$, Key: AB. In BCNF.

Final relations are the same.

**Question 5.** Given the following relation, use 3NF decomposition to convert it to relations in BCNF. For each resulting relation, check if it is also in BCNF.

$R(A, B, C, D, E, F, G)$ $\mathcal{F} = \{AB \to C, CD \to EF, CF \to AG\}$

**Answer.**

Keys: ABD, BCD

Using 3NF decompositions, we get:

$R1(A, B, C)$ with $\{AB \to C\}$, key: AB, in 3NF and BCNF.

$R2(C, D, E, F)$ with $\{CD \to EF\}$, key:CD, in 3NF and BCNF.

$R3(C, F, A, G)$ with $\{CF \to AG\}$, key: CF, in 3NF and BCNF.

$R4(A, B, D)$ with $\{\}$, key: ABD, in 3NF and BCNF.

We need R4 because there is no relation with all the attributes of a key. The choice is key is arbitrary here, we could have used the other key as well. The application may help determine which key is more important to keep.