COMP3013 2025 Fall

Assignment 3

Student are expected to submit two files.

- "COMP3013_25F_A3_XXX.sql", where "XXX" is your student ID. This SQL file contains all the SQL queries for Q1. First line of the file should contain your name and ID as a comment. And question number for each question is also included as a comment. SQL comments are quoted by the sign /*…*/.
- "COMP3013_25F_A3_XXX.pdf", for the rest of the questions.

Submissions which do not following the guideline may **not be marked**.

Q1. The schema of a database is given as follows. Keys are underlined.

- `student=(sID,sname,gender,age,gpa,pname)`
  // `gender` is either `male` or `female`.
- `program=(pname,division)`
  // `pgname` is the program name
- `course=(cID,cname,pname,credit)`
  // `credit` is an integer and 1 ≤`credit`≤ 3.
- `enroll=(sID,cID,grade)`
  //`grade` is one of `A`, `B`, `C`, `D`, or `F`.

Write a query for each following question. (8 marks for each)

a) Find the names of students who have enrolled all courses.
Ans.

```
SELECT sname
FROM student JOIN enroll as e1 USING (sID)
WHERE NOT EXISTS (
      (SELECT cID
      FROM course)
      EXCEPT
      (SELECT cID
      FROM enroll AS e2
      WHERE e1.sID = e2.sID)
)

SELECT sname
FROM student
JOIN enroll as e1 USING (sID)
WHERE NOT EXISTS (
      SELECT cID
      FROM course
      WHERE cID NOT IN (
            SELECT cID
```

```
                FROM enroll AS e2
                WHERE e1.sID = e2.sID
        )
)
```

b) Find the names of students who have enrolled some courses from every program.
Ans.

```
SELECT sname
FROM student
WHERE NOT EXISTS (
    SELECT pname
    FROM program
    WHERE pname NOT IN (
        SELECT pname
        FROM course JOIN enroll ON course.cID = enroll.cID
        WHERE enroll.sID = student.sID
    )
);
```

c) Find the names of students who has not received an "F" from any course.
Ans.

```
SELECT sname
FROM student AS s LEFT OUTER JOIN enroll AS e
      ON s.sID = e.sID
WHERE s.sID NOT IN (
      SELECT sID
      FROM enroll
      WHERE grade= 'F'
)
```

d) Find the name of the student who has enrolled more courses than other students.
Ans.

```
SELECT student.sname
FROM student
JOIN enroll AS e1 ON student.sID = e1.sID
GROUP BY student.sID
HAVING COUNT(e1.cID) >= ALL(
      SELECT COUNT(e2.cID)
      FROM enroll AS e2
      GROUP BY e2.sID
)

SELECT student.sname
FROM student
JOIN enroll ON student.sID = enroll.sID
GROUP BY student.sID
HAVING COUNT(cID) = (
      SELECT MAX(cnt)
      FROM (
            SELECT COUNT(cID) AS cnt
```

```
                FROM enroll
                GROUP BY Sid
          ) AS count
    )
```

e) Create a constraint to guarantee the credit is in the correct range.
Ans.

```
ALTER TABLE course
ADD CONSTRAINT credit_range
CHECK (credit >= 1.0 AND credit <= 3.0)
```

Q2. Given an instance of a relational schema $R = \{A, B, C\}$ and a list of functional dependencies.

A	B	C
a	α	T
a	β	T
a	γ	T
b	ε	F

Decide whether the functional dependencies are satisfied by the instance. (9 pt)

a) $A \to B$
b) $A \to C$
c) $B \to A$
d) $B \to C$
e) $C \to A$
f) $C \to B$
g) $AB \to C$
h) $AC \to B$
i) $BC \to A$

Ans.

a)	b)	c)	d)	e)	f)	g)	h)	i)
no	yes	yes	yes	yes	no	yes	no	yes

Q3. Let A, B, C be three arbitrary attributes. Assume the functional dependency $AB \to C$ holds. Can we prove $A \to C$? If yes, then prove it by Armstrong's Axiom; if no, then disprove it by a counter example. (10pt)

Ans.

No. Consider the following instance.

	A	B	C
t_1	a_1	b_1	c_1
t_2	a_1	b_2	c_2

"if $t_1[A, B] = t_2[A, B]$, then $t_1[C] = t_2[C]$" holds. But, "if $t_1[A] = t_2[A]$, then $t_1[C] = t_2[C]$" does not hold.
More intuitively, "$AB \to C$" is understood as "if we know A and B, then we know C"; and "$A \to C$" is understood as "if we know A, then we know C". Then, the answer becomes obvious.

Q4. Given a relational schema and a set of functional dependencies

- $R = \{A, B, C, D, E\}$
- $F = \{AC \to B, BD \to C, CE \to D, DA \to E\}$

a) Find all candidate keys of R. (6 pt)
b) Decompose R into BCNF. Show the steps. (15 pt)
c) Does the BCNF decomposition in part b) preserve all functional dependencies? Why? (5 pt)
d) Decompose R into 3NF. Show the steps. (15 pt)

Ans.

a) $\{B, C, D, E\}^+ = \{B, C, D, E\}$ Thus, A is an attribute in an arbitrary key. Then, we can enumerate all the cases.
One attribute:
$\{A\}^+ = \{A\}$ is not a key.
Two attributes:
$\{AC\}^+ = \{ACB\}$ is not a key.
$\{BD\}^+ = \{BDC\}$ is not a key.
$\{CE\}^+ = \{CED\}$ is not a key.
$\{DA\}^+ = \{DAE\}$ is not a key.
Three attributes:
$\{A, B, C\}^+ = \{A, B, C\}$ is not a key.
$\{A, B, D\}^+ = \{A, B, C, D, E\}$ is a key.
$\{A, B, E\}^+ = \{A, B, E\}$ is not a key.
$\{A, C, D\}^+ = \{A, B, C, D, E\}$ is a key.
$\{A, C, E\}^+ = \{A, B, C, D, E\}$ is a key.
$\{A, D, E\}^+ = \{A, D, E\}$ is not a key.
Candidate keys: $\{ACD\}, \{ABD\}, \{ACE\}$

b) All FDs in F^+ violates BCNF.
$result = R_0 = \{ABCDE\}$
For $AC \to B$,
$R_1 = \{ABC\}, F_1 = \{AC \to ABC\} = \{AC \to B\}$
$R_2 = R_0 \backslash (\{B\} \backslash \{\mathrm{AC}\}) = \{ACDE\}, F_2 = \{AD \to ADE, CE \to CDE\} = \{AD \to E, CE \to D\}$
$result = (\{result \backslash R_0\}) \cup \{R_1\} \cup \{R_2\} = \{\{ABC\}, \{ACDE\}\}$

Situation1:
For $AD \to E$,
$R_3 = \{ADE\}, F_3 = \{AD \to ADE\} = \{AD \to E\}$
$R_4 = \{ACDE\} \backslash (\{E\} \backslash \{\mathrm{AD}\}) = \{ACD\}, F_4 = \{\}$
$result = (\{result \backslash \{ACDE\}\}) \cup \{R_3\} \cup \{R_4\} = \{\{ABC\}, \{ADE\}, \{ACD\}\}$
Note: the result above fails to preserve $BD \to C$ and $CE \to D$.

Situation2:
For $CE \to D$,
$R_3 = \{CDE\}, F_3 = \{CE \to D\}$
$R_4 = \{ACDE\} \backslash (\{D\} \backslash \{CE\}) = \{ACE\}, F_4 = \{\}$
$result = (\{result \backslash \{ACDE\}\}) \cup \{R_3\} \cup \{R_4\} = \{\{ABC\}, \{CDE\}, \{ACE\}\}$
Note: the result above fails to preserve $AD \to E$ and $BD \to C$.

c) The result in b) fails to preserve all functional dependencies in F.

d) F_c=F, there's no redundant dependencies or extraneous attributes in any one of the dependencies. So the 3NF decomposition of R is $R_1 = \{ABC\}, R_2 = \{ADE\}, R_3 = \{BCD\}, R_4 = \{CDE\}, R_5 = \{ACD\}, or\ R_5 = \{ABD\}, or\ R_5 = \{ACE\}$.