National University of Singapore
CS2102: Database System

Final Assessment AY2023/24 Sem 2

Instructions

1. Please read the **Instructions** carefully.
2. This assessment paper contains **12 (TWELVE)** questions and comprises of **8 (EIGHT)** printed pages.
3. The total mark for the assessment is **100 Points**.
4. **Answer ALL questions**.
5. All answers are to be written in the space provided in the answer sheet.
   - Any answer **NOT** in the space provided will **NOT** be graded.
6. This is a **CLOSED-BOOK** assessment
   - You are allowed **1 (ONE)** A4-sized double-sided cheatsheet.
   - Calculators are **NOT** allowed.
7. All codes are run on PostgreSQL 16.
8. We will use the *shorthand* notation for the functional dependencies.
   - Instead of writing $\{A, B\} \rightarrow \{C, D\}$, we will write $AB \rightarrow CD$.
   - You may use the same notation on your answer.
9. The duration of the assessment is **2 (TWO)** hours.

Marks

Section	Questions	Points
A	1 - 4	42
B	5 - 6	14
C	7 - 12	44
	Total	**100**

Good Luck!

A. Relational Model 42 Points

For the next **3 (THREE)** questions, consider the relations $R_1(A, B)$, $R_2(C, D)$, $R_3(A, C, E)$, and $R_4(A, C, F)$. Furthermore, consider the *natural join* of all the relations above. We have the following set of functional dependencies that holds on the ER diagram **AND** on the natural join of all the relations above.

$$\Sigma = \{A \rightarrow B,\ C \rightarrow D,\ A \rightarrow CE,\ C \rightarrow AF\}$$

You may assume that the following foreign key constraints are also satisfied:

$(R_3.A) \rightsquigarrow (R_1.A)$ $(R_4.A) \rightsquigarrow (R_1.A)$

$(R_3.C) \rightsquigarrow (R_2.C)$ $(R_4.C) \rightsquigarrow (R_2.C)$

1. (12 points) Complete the ER diagram that corresponds to the relations above and satisfies the functional dependencies as well as foreign key constraints.
2. (10 points) Complete the schema that corresponds to the relations above and satisfies the functional dependencies as well as foreign key constraints. Additionally, ensure that none of the attributes can be `NULL`.
3. (8 points) Select **ALL** queries that *may* produce duplicate, namely, multiple rows that have the same value on each attributes. We assume that the relational instance satisfies the constraints in the schema (*including the constraint that none of the attributes can be* `NULL`) as well as all the functional dependencies and foreign key constraints. You may assume that all queries are valid queries. We will use the `monospace` typeface `R1` instead of R_1 in our queries.

   (A) Query #1

```
SELECT R1.A, R2.C
FROM   R1, R2;
```

   (B) Query #2

```
SELECT R1.B, R2.D
FROM   R1 NATURAL JOIN R2;
```

   (C) Query #3

```
SELECT R2.C, R3.E
FROM   R2 NATURAL JOIN R3;
```

   (D) Query #4

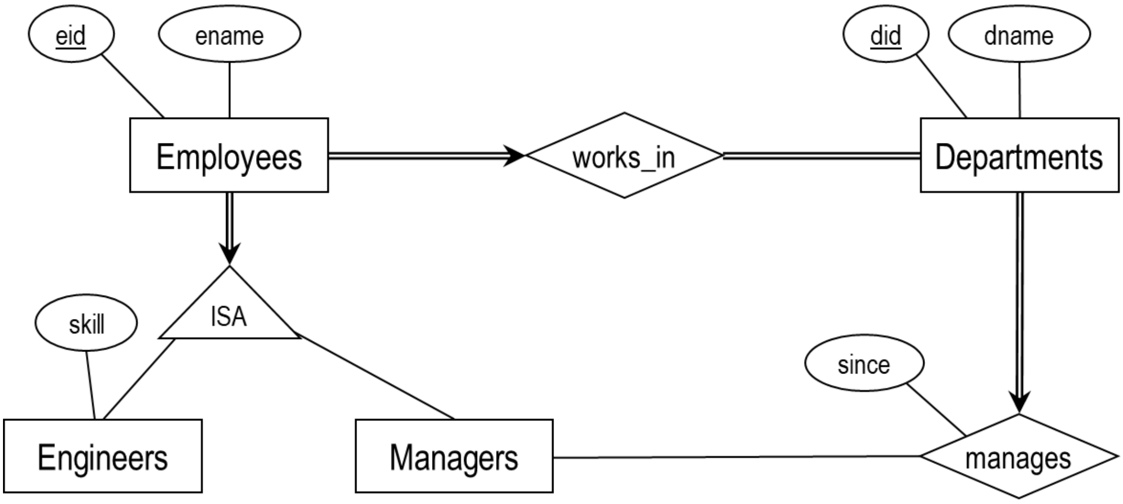
```
SELECT R2.C, R4.F
FROM   R2, R4
WHERE  R4.C = R2.C;
```

   (E) Query #5

```
SELECT R3.E, R4.F
FROM   R3 NATURAL JOIN R4;
```

Continue on the next page

4. (12 points) Consider the following ER diagram.



Further consider the following translation to schema. We will be using **VIEW**.

```
CREATE TABLE Departments (
  did   INT  PRIMARY KEY,
  dname TEXT NOT NULL
);

CREATE TABLE Engineers (
  eid    INT  PRIMARY KEY,
  ename  TEXT NOT NULL,
  skill  TEXT NOT NULL,
  did    INT  NOT NULL REFERENCES Departments (did)
);

CREATE TABLE Managers (
  eid    INT  PRIMARY KEY,
  ename  TEXT NOT NULL,
  did    INT  NOT NULL REFERENCES Departments (did)
);

CREATE VIEW Employees AS (
  SELECT eid, ename FROM Engineers
  UNION
  SELECT eid, ename FROM Managers
);

CREATE TABLE manages (
  did   INT  PRIMARY KEY REFERENCES Departments (did),
  eid   INT  NOT NULL REFERENCES Managers (eid),
  since DATE NOT NULL
);
```

Continue on the next page

Select **ALL** statements that are true about the relational schema.

(A) It enforces cardinality constraint on `Employees` with respect to `works_in`.

(B) It enforces total participation constraint on `Employees` with respect to `works_in`.

(C) It enforces cardinality constraint on `Departments` with respect to `manages`.

(D) It enforces total participation constraint on `Departments` with respect to `manages`.

(E) It enforces total participation constraint on `Departments` with respect to `works_in`.

(F) It enforces covering constraint of the ISA.

(G) It enforces overlap constraint of the ISA.

(H) It enforces the functional dependency $\{\texttt{eid}\} \rightarrow \{\texttt{ename}\}$.

Continue on the next page

B. Stored Procedures and Triggers 14 Points

sid	name	score
s1	Alice	50
s2	Bob	60
s3	Cathy	70
s4	David	80
s5	Eric	90

For the next **2 (TWO)** questions, we will consider the following table `Scores` on the right. The data types of `sid` and `name` are *TEXT*, while the data type of `score` is *INT*.

5. (6 points) Consider the `test_func` function below.

```
CREATE OR REPLACE FUNCTION test_func()
RETURNS INT AS $func$
DECLARE
  curs CURSOR for (SELECT * FROM Scores ORDER BY score desc);
  r RECORD;
  sum_score INT;
  cnt1 INT;
  cnt2 INT;
BEGIN
  sum_score := -1;
  cnt1 := 0;
  cnt2 := 0;

  OPEN curs;
  LOOP
    FETCH curs INTO r;
    EXIT WHEN NOT FOUND;
    cnt1 := cnt1 + 1;

    IF (sum_score = -1) THEN
      sum_score := r.score;
    ELSE
      sum_score := sum_score + r.score;
      IF (r.score < sum_score / cnt1) THEN
        cnt2 := cnt2 + 1;
      END IF;
    END IF;
  END LOOP;

  CLOSE curs;
  RETURN cnt2;
END;
$func$ LANGUAGE plpgsql;
```

Continue on the next page

Suppose that we execute the following query.

```
SELECT * FROM test_func();
```

What will be the result of the query? In your answer, you may omit the column name of the query result.

6. (8 points) Consider the `scores_check_func` function below.

```
CREATE OR REPLACE FUNCTION scores_check_func()
RETURNS TRIGGER AS $func$
DECLARE
  cnt1 INT;
  cnt2 INT;
BEGIN
  SELECT COUNT(*) INTO cnt1
  FROM   Scores;

  SELECT COUNT(*) INTO cnt2
  FROM   Scores
  WHERE  score >= 70;

  IF (cnt2 * 2 < cnt1) THEN
    RAISE EXCEPTION 'You are too mean!';
  END IF;

  RETURN OLD;
END;
$func$ LANGUAGE plpgsql;
```

Suppose that we create a trigger `scores_check_trigger` based on the function above, and then we execute the following transaction.

```
BEGIN TRANSACTION;
  DELETE FROM Scores WHERE sid = 's4';
  DELETE FROM Scores WHERE sid = 's5';
  INSERT INTO Scores VALUES ('s6', 'Fred', 95);
COMMIT;
```

After the execution of the transaction above, we find that the table `Scores` contains **4 (FOUR)** rows as shown on the right.

sid	name	score
s1	Alice	50
s2	Bob	60
s3	Cathy	70
s6	Fred	95

Provide **2 (TWO)** different **VALID** definitions of `scores_check_trigger` such that both definitions allow the above scenario. Note that some blank(s) in the answer sheet can be empty and each blank can be filled with multiple keywords. However, you are **NOT** allowed to use the `WHEN` keyword.

Continue on the next page

C. Functional Dependencies and Normal Form 44 Points

For the next **2 (TWO)** questions, consider the following relation $R(A, B, C, D, E, F)$ with the following set of functional dependencies. Note that we will use the *shorthand* notation for the functional dependencies. Instead of writing $\{A, B\} \to \{C, D\}$, we will write $AB \to CD$. You may use the same notation on your answer.

$$\Sigma = \{A \to D,\ A \to E,\ BC \to A,\ BE \to A,\ BF \to A,\ CD \to A,\ D \to B,\ D \to E,\ E \to C,\ F \to B,\ F \to C\}$$

7. (4 points) Suppose we decompose R into $R_1(A, B, C, D)$ and $R_2(C, D, E, F)$. Is this a lossless join decomposition? Briefly justify your answer.

8. (6 points) Suppose we decompose R into $R_1(A, B, C, D)$ and $R_2(C, D, E, F)$. Is this a dependency-preserving decomposition? If not, please identify **ALL** functional dependencies in Σ that are **NOT** preserved. If yes, please identify the projections of Σ on both R_1 and R_2 respectively.

For the remainder of the papers, each question will have their own set of functional dependencies Σ.

9. (4 points) Consider the following relation $R(A, B, C, D, E, F)$ with the following set of functional dependencies. Note that we will use the *shorthand* notation for the functional dependencies. Instead of writing $\{A, B\} \to \{C, D\}$, we will write $AB \to CD$. You may use the same notation on your answer.

$$\Sigma = \{A \to B,\ BC \to A,\ D \to E,\ E \to D,\ CF \to B,\ BF \to E,\ B \to C,\ EF \to A,\ DE \to C,\ EF \to BC\}$$

Identify **ALL** keys of R with respect to Σ. Briefly justify your answer.

10. (10 points) Consider the following relation $R(A, B, C, D, E, F)$ with the following set of functional dependencies. Note that we will use the *shorthand* notation for the functional dependencies. Instead of writing $\{A, B\} \to \{C, D\}$, we will write $AB \to CD$. You may use the same notation on your answer.

$$\Sigma = \{A \to F,\ AD \to E,\ AE \to C,\ AF \to B,\ B \to A,\ BC \to A,\ C \to DE,\ CD \to B,\ DE \to A,\ E \to F,\ F \to D,\ BF \to C\}$$

Is R in BCNF with respect to Σ? If yes, briefly justify your answer. If not, derive a BCNF decomposition of R using **ONLY** the decomposition algorithm introduced in the CS2102 lectures. Show your steps.

Continue on the next page

11. (12 points) Consider the following relation $R(A, B, C, D, E, F)$ with the following set of functional dependencies. Note that we will use the *shorthand* notation for the functional dependencies. Instead of writing $\{A, B\} \rightarrow \{C, D\}$, we will write $AB \rightarrow CD$. You may use the same notation on your answer.

$$\Sigma = \{A \rightarrow BCD,\ B \rightarrow C,\ E \rightarrow F,\ BD \rightarrow EF,\ EF \rightarrow D,\ AE \rightarrow DF\}$$

Is R in 3NF with respect to Σ? If yes, briefly justify your answer. If not, derive a 3NF decomposition of R using **only** the 3NF decomposition (*i.e.*, synthesis) algorithm introduced in the CS2102 lectures. Show your steps for deriving minimal basis.

12. (8 points) Consider the following relation $R(A, B, C, D)$. We do not know what the set of functional dependencies is yet, that is what we want to find out. We know that the set of functional dependencies Σ satisfies the following conditions simultaneously:

- R is in 3NF **BUT NOT** in BCNF with respect to Σ.
- If we derive a BCNF decomposition of R using **ONLY** the BCNF decomposition algorithm introduced in CS2102 lectures, we will obtain the following decomposition:

$$R_1(A, B),\quad R_2(B, C),\quad R_3(C, D)$$

Find Σ. Briefly justify your answer.

– End of Paper –