

Question 1 (10 marks)

Consider the relational schemas given below and the respective sets of functional dependencies valid in the schemas.

For each one of the relational schemas, determine the highest normal form, which is valid for a schema. Justify your answer. Justification must include the derivations of minimal keys from the functional dependencies and testing the validity of all normal forms (2NF, 3NF, BCNF) against the relational schemas, minimal keys, and functional dependencies.

If a schema is not in BCNF, then decompose it into a minimum number of schemas so that each one of them is in BCNF.

A correct guess without the comprehensive justifications scores no marks !

(1) 3 marks

$R = (A, B, C, D)$

$D \rightarrow C$

$C \rightarrow A$

$C \rightarrow B$

(2) 3 marks

$R = (A, B, C, D)$

$A \rightarrow B$

(3) 2 marks

$R = (A, B, C, D)$

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow D$

$D \rightarrow A$

(4) 1 mark

$R = (A, B, C, D)$

No nontrivial functional dependencies are valid in the schema R.

Question 2 (10 marks)

Assume, that SQL script with **CREATE TABLE** statements listed below has been used to create the relational tables on a database server located at a computer system **db1teach.adeis.uow.edu.au**

```
CREATE TABLE APPLICANT(
    anumber    DECIMAL(6)      NOT NULL, /* Applicant number */
    fname      VARCHAR(20)     NOT NULL, /* First name */
    lname      VARCHAR(30)     NOT NULL, /* Last name */
    dob        DATE            NOT NULL, /* Date of birth */
    email      VARCHAR(50)     , /* E-mail address */
    CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(
    anumber    DECIMAL(6) NOT NULL, /* Applicant number */
    sname      VARCHAR(30) NOT NULL, /* Skill name */
    slevel     DECIMAL(2) NOT NULL, /* Skill level */
    CONSTRAINT SPOSSESSED_pkey PRIMARY KEY ( anumber, sname ),
    CONSTRAINT SPOSSESSED_fkey1 FOREIGN KEY ( anumber )
        REFERENCES APPLICANT ( anumber ),
    CONSTRAINT SPOSSESSED_check1 CHECK ( slevel IN
        ( 1,2,3,4,5,6,7,8,9,10 ) ) );
```

For each one of the queries listed above write **CREATE INDEX** statement to create an index that can be used to speed up processing of a query. Each index must be created separately for each query.

(1) 2 marks

```
SELECT anumber, email
FROM APPLICANT
WHERE fname IN ('James', 'Kate', 'Paul') AND lname = 'BOND';
```

(2) 2 marks

```
SELECT sname, slevel
FROM SPOSSESSED
WHERE sname IN ('painting','singing')
ORDER BY slevel;
```

(3) 2 marks

```
SELECT fname, lname, count(*)
FROM APPLICANT
GROUP BY fname, lname;
```

(4) 2 marks
 SELECT AVG(slevel)
 FROM SPOSSESSED;

(5) 2 marks
 SELECT COUNT(DISTINCT anumber)
 FROM SPOSSESSED
 WHERE slevel > 5;

Question 3 (10 marks)

Assume, that SQL script with **CREATE TABLE** statements listed below has been used to create the relational tables in a sample database located at a computer system **db1teach.adeis.uow.edu.au**

```
CREATE TABLE APPLICANT(
anumber      DECIMAL(6)      NOT NULL, /* Applicants          */
fname        VARCHAR(20)     NOT NULL, /* Applicant number     */
lname        VARCHAR(30)     NOT NULL, /* First name           */
dob          DATE            NOT NULL, /* Last name            */
email        VARCHAR(50)     , /* Date of birth         */
              /* E-mail address       */
CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(
anumber      DECIMAL(6) NOT NULL, /* Skills possessed by applicants */
sname        VARCHAR(30) NOT NULL, /* Applicant number             */
slevel       DECIMAL(2) NOT NULL, /* Skill name                   */
CONSTRAINT SPOSSESSED_pkey PRIMARY KEY ( anumber, sname ),
CONSTRAINT SPOSSESSED_fkey1 FOREIGN KEY ( anumber )
REFERENCES APPLICANT ( anumber ),
CONSTRAINT SPOSSESSED_check1 CHECK ( slevel IN
( 1,2,3,4,5,6,7,8,9,10 ) ) );
```

(1) 1 mark

Write SQL statements that modify the structures of the sample database such after the modifications it would be possible to store in the database information about the total number of skills possessed by each applicant. Note, that after the structural modifications all relational schemas of the sample database must still be in BCNF.

(2) 2 marks

Write SQL statements that modify the contents of the sample database such after the modifications the sample database contains information about the total number of skills possessed by each applicant.

(2) 3 marks

Write a **statement database trigger** and a stored function to be executed by the trigger that automatically modifies the total number of

skills possessed by an applicant whenever information about a new skill possessed by an applicant is inserted into the sample database or information about a skill already possessed by an applicant is deleted from the sample database.

(2) 4 marks

Write a **row database trigger** and a stored function to be executed by the trigger that automatically modifies the total number of skills possessed by an applicant whenever information about a new skill possessed by an applicant is inserted into the sample database or information about a skill already possessed by an applicant is deleted from the sample database.

Question 4 (10 marks)

Assume, that SQL script with **CREATE TABLE** statements listed below has been used to create the relational tables in a sample database located at a computer system **db1teach.adeis.uow.edu.au**

```
CREATE TABLE APPLICANT(
/* Applicants */
anumber DECIMAL(6) NOT NULL, /* Applicant number */
fname VARCHAR(20) NOT NULL, /* First name */
lname VARCHAR(30) NOT NULL, /* Last name */
dob DATE NOT NULL, /* Date of birth */
email VARCHAR(50), /* E-mail address */
CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(
/* Skills possessed by applicants */
anumber DECIMAL(6) NOT NULL, /* Applicant number */
sname VARCHAR(30) NOT NULL, /* Skill name */
slevel DECIMAL(2) NOT NULL, /* Skill level */
CONSTRAINT SPOSSESSED_pkey PRIMARY KEY ( anumber, sname ),
CONSTRAINT SPOSSESSED_fkey1 FOREIGN KEY ( anumber )
REFERENCES APPLICANT ( anumber ),
CONSTRAINT SPOSSESSED_check1 CHECK ( slevel IN
( 1,2,3,4,5,6,7,8,9,10 ) ) );
```

A database developer implemented the following stored function.

```
CREATE OR REPLACE FUNCTION AVGTOTAL(skill1 IN VARCHAR, skill2 VARCHAR)
RETURNS VARCHAR AS '
DECLARE
total1 DECIMAL(7);
total2 DECIMAL(7);
BEGIN
SELECT COUNT(*)
INTO total1
FROM SPOSSESSED
WHERE sname = skill1;
```

```

SELECT COUNT(*)
INTO total2
FROM SPOSSESSED
WHERE sname = skill2;

RETURN (total1+total2)/2;
END;
' LANGUAGE plpgsql;

```

(1) 7 marks

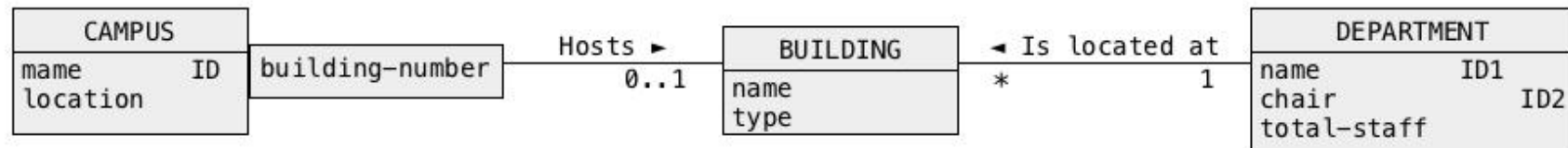
Show a sample concurrent processing of a stored function **AVGTOTAL** running at **READ COMMITTED** isolation level with another database transaction such that the result returned by the function is incorrect. The other transaction is up to you. Apply two-dimensional visualisation of concurrent processing of database transactions explained to you during the lecture classes.

(2) 3 marks

Explain how a stored function **AVGTOTAL** given above should be changed to always get the correct results when processed at **READ COMMITTED** isolation level.

Question 5 (10 marks)

Consider a conceptual schema given below. The schema represents a sample database domain where the university campuses host the buildings and the departments are located at the buildings.



(1) 7 marks

Transform a conceptual schema given above into a logical schema of BSON documents. For a sample logical schema of BSON documents see presentation 24 BSON Design, slides 53 or 54. Draw a logical schema and submit an image of your drawing. You can use UMLet to draw the logical schema or you can draw it on a piece of paper, take a photo or scan it and submit an image of your drawing.

(2) 3 marks

For a logical schema created in the previous step write the sample BSON documents whose contents are consistent with the logical schema. Your documents must contain information about at least one university campus, three buildings, and two departments. The values associated with the keys of BSON documents are up to you.

Question 6 (10 marks)

Consider a sample BSON document given below. Assume, that all documents in a collection **bank** have the same structure as the document listed below.

```
db.bank.insert(
  { "bank bsb":236,
    "bank name":"National Australia Bank",
    "bank symbol":"NAB",
    "address":{"street":"Bank Avenue",
              "bldg":1,
              "city":"Sydney",
              "country":"Australia"},
    "branches":[ {"branch name":"University",
                  "date opened":"12-DEC-2021",
                  "employees": [ {"employee number":007,
                                  "employee name":"James Bond",
                                  "years experience":85},
                                {"employee number":666,
                                  "employee name":"Harry Potter",
                                  "years experience":30} ]
                  },
                {"branch name":"City",
                  "date opened":"30-JAN-2022",
                  "employees": [ {"employee number":123
                                  "employee name":"Robin Hood",
                                  "years experience":550} ]
                }
    ]
  }
);
```

(1) 2 marks

Find the names of branches ("branch name" key) that belong to all banks located in Australia ("country" key) and in Sydney ("city" key).

(2) 2 marks

Find the bsbs and the names ("bank bsb" and "bank name" keys) of all banks that have no branches.

(2) 2 marks

For each bank find the total number of branches opened, list bank symbol ("bank symbol" key) and the total number of branches opened.

Use either a method **remove()** or a method **update()** to write the implementations of the following data manipulation operations.

(4) 2 marks

Delete from a collection **bank** the documents that contain information about a branch named University ("branch name" key).

(5) 2 marks

Delete from a collection **bank** information about all employees who have more than 10 years experience ("years experience" key).

Submit Query