

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

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

$A \rightarrow B$

$B \rightarrow A$

$B \rightarrow D$

(2) 3 marks

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

$A, B \rightarrow C$

$A, B \rightarrow D$

$C \rightarrow D$

(3) 2 marks

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

$B \rightarrow C, D$

(4) 1 mark

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

No nontrivial functional dependencies are valid in the schema R.

Question 2 (10 marks)

This question is related to the relational tables created through processing of **CREATE TABLE** statements listed below.

```
CREATE TABLE APPLICANT(                                /* Applicants */
anumber          NUMBER(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
*/
city         VARCHAR(30)   NOT NULL, /* City
CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(          /* Skills possessed */
anumber      NUMBER(6)      NOT NULL, /* Applicant number */
sname        VARCHAR(30)    NOT NULL, /* Skill name */
slevel       NUMBER(1)      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) ) );

CREATE TABLE POSITION(          /* Advertised positions */
pnumber      NUMBER(8)      NOT NULL, /* Position number */
title        VARCHAR(30)    NOT NULL, /* Position title */
salary       NUMBER(9,2)    NOT NULL, /* Salary */
CONSTRAINT POSITION_pkey PRIMARY KEY ( pnumber ) );

CREATE TABLE APPLIES(          /* Positions applied */
anumber      NUMBER(6)      NOT NULL, /* Applicant number */
pnumber      NUMBER(8)      NOT NULL, /* Position number */
appdate      DATE          NOT NULL, /* Application date */
CONSTRAINT APPLIES_pkey PRIMARY KEY ( anumber, pnumber, appdate
),
CONSTRAINT APPLIES_fkey1 FOREIGN KEY ( anumber ) REFERENCES
APPLICANT ( anumber ),
CONSTRAINT APPLIES_fkey2 FOREIGN KEY ( pnumber ) REFERENCES
POSITION ( pnumber ) );

```

Assume, that SQL script with **CREATE TABLE** statements listed above has been used to create the relational tables on a database server located at a computer system **data-pc01.adeis.uow.edu.au**.

Assume, that sample data has been loaded to the relational tables located on a database server at **data-pc01.adeis.uow.edu.au**.

Assume, that another database server is available on a computer system **data-pc02.adeis.uow.edu.au**. Assume, that all information about port numbers, system identifiers, database users, passwords on both database servers at **data-pc01.adeis.uow.edu.au** and at **data-pc02.adeis.uow.edu.au** is up to you. To make it nice and easy you can re-apply information valid for your laboratory classes.

(1) 2 marks

Explain how SQL script with **CREATE TABLE** statements can be used to create the relational tables on a computer system **data-pc02.adeis.uow.edu.au**.

(2) 8 marks

Write SQL statements to copy over a database link the contents of the relational tables located on a computer system **data-pc01.adeis.uow.edu.au** to the relational tables located on a computer system **data-pc02.adeis.uow.edu.au**.

Question 3 (10 marks)

This question is related to the relational tables created through processing of **CREATE TABLE** statements listed below.

```

CREATE TABLE APPLICANT(                                     /* Applicants */
anumber      NUMBER(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 */
city         VARCHAR(30)    NOT NULL, /* City */
CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(                                     /* Skills possessed */
anumber      NUMBER(6)      NOT NULL, /* Applicant number */
sname        VARCHAR(30)    NOT NULL, /* Skill name */
slevel       NUMBER(1)      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 ) ) );

CREATE TABLE POSITION(                                       /* Advertised positions */
pnumber      NUMBER(8)      NOT NULL, /* Position number */
title        VARCHAR(30)    NOT NULL, /* Position title */
salary       NUMBER(9,2)    NOT NULL, /* Salary */
CONSTRAINT POSITION_pkey PRIMARY KEY ( pnumber ) );

CREATE TABLE APPLIES(                                         /* Positions applied */
anumber      NUMBER(6)      NOT NULL, /* Applicant number */
pnumber      NUMBER(8)      NOT NULL, /* Position number */
appdate      DATE           NOT NULL, /* Application date */
CONSTRAINT APPLIES_pkey PRIMARY KEY ( anumber, pnumber, appdate ),
CONSTRAINT APPLIES_fkey1 FOREIGN KEY ( anumber ) REFERENCES
APPLICANT ( anumber ),
CONSTRAINT APPLIES_fkey2 FOREIGN KEY ( pnumber ) REFERENCES
POSITION ( pnumber ) );

```

Write an implementation of a **statement database** trigger, that automatically verifies a consistency constraint listed below.

"An applicant cannot apply twice for the same position on two different days".

Note, that implementation of a row trigger scores no marks because you will certainly get mutating table error.

Write SQL statements to comprehensively test the trigger.

Question 4 (10 marks)

This question is related to the relational tables created through processing of **CREATE TABLE** statements listed below.

```

CREATE TABLE APPLICANT(                                /* Applicants */
anumber          NUMBER(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 */
city             VARCHAR(30)         NOT NULL, /* City */
CONSTRAINT APPLICANT_pkey PRIMARY KEY ( anumber ) );

CREATE TABLE SPOSSESSED(                                /* Skills possessed */
anumber          NUMBER(6)          NOT NULL, /* Applicant number */
sname            VARCHAR(30)         NOT NULL, /* Skill name */
slevel           NUMBER(1)          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) ) );

CREATE TABLE POSITION(                                /* Advertised positions */
pnumber          NUMBER(8)          NOT NULL, /* Position number */
title            VARCHAR(30)         NOT NULL, /* Position title */
salary           NUMBER(9,2)        NOT NULL, /* Salary */
CONSTRAINT POSITION_pkey PRIMARY KEY ( pnumber ) );

CREATE TABLE APPLIES(                                /* Positions applied */
anumber          NUMBER(6)          NOT NULL, /* Applicant number */
pnumber          NUMBER(8)          NOT NULL, /* Position number */
appdate          DATE                NOT NULL, /* Application date */
CONSTRAINT APPLIES_pkey PRIMARY KEY ( anumber, pnumber, appdate ),
CONSTRAINT APPLIES_fkey1 FOREIGN KEY ( anumber ) REFERENCES
APPLICANT ( anumber ),
CONSTRAINT APPLIES_fkey2 FOREIGN KEY ( pnumber ) REFERENCES
POSITION ( pnumber ) );

```

Assume that the structures and the contents of a sample database have been changed in the following way.

```
ALTER TABLE APPLICANT ADD (total_apps NUMBER(7));
```

```

UPDATE APPLICANT
SET total_apps = ( SELECT count(*)
                  FROM APPLIES
                  WHERE anumber = APPLICANT.anumber);

```

A database developer implemented the following stored procedure.

```
CREATE OR REPLACE PROCEDURE INSERT_APP(app_number IN NUMBER,  
                                         pos_number IN NUMBER  
) IS  
total NUMBER(3);  
  
BEGIN  
  SELECT COUNT(*)  
  INTO total  
  FROM APPLIES  
  WHERE anumber = app_number;  
  
  INSERT INTO APPLIES VALUES(app_number, pos_number, SYSDATE);  
  
  total := total + 1;  
  
  UPDATE APPLICANT  
  SET total_apps = total  
  WHERE anumber = app_number;  
  
  COMMIT;  
END;
```

(1) 1 mark

Explain why a procedure **INSERT_APP** cannot be processed at **READ COMMITTED** isolation level together with another concurrently running procedure **INSERT_APP**.

(2) 2 marks

Explain what phenomenon may occur during a concurrent processing of two procedures **INSERT_APP** at **READ COMMITTED** isolation level.

(3) 7 marks

Show a sample concurrent processing of two procedures **INSERT_APP** that prohibits a concurrent processing of two procedures **INSERT_APP** at **READ COMMITTED** isolation level. Apply two-dimensional visualisation of concurrent processing of database transactions explained to you during the lecture classes.

Question 5 (10 marks)

Consider a conceptual schema given below. The schema represents a sample database domain where vehicle repair facilities employ employees and repair vehicles.

Question 5

(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 23 BSON Design, slide 53. Draw a logical schema 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 the logical schema. Your documents must contain information about at least one vehicle repair facility, two employees, and two cars repaired at the facility. 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 **driver** have the same structure as the document listed below.

```
db.driver.insert(
  { "first_name":"James",
    "last_name":"Bond",
    "licence":007,
    "address":{"street":"Northfields Ave",
               "bldg":3,
               "city":"Wollongong",
               "country":"Australia"},
    "trips":[ { "number":5,
                 "truck_rego":"PKR856",
                 "date":"12-DEC-2017",
                 "legs": [ { "number":1,
                             "departure":"Sydney",
                             "destination":"Melbourne" },
                           { "number":2,
                             "departure":"Melbourne",
                             "destination":"Sydney" } ] },
               { "number":25,
                 "truck_rego":"AL08UK",
                 "date":"03-JUN-2018",
                 "legs": [ { "number":1,
                             "departure":"Sydney",
                             "destination":"Melbourne" } ] }
            ]
  }
);
```

Use either a method find() or a method aggregate() available in MongoDB to write the implementations of the following queries.

(1) 2 marks

Find a registration number ("truck_rego" key) of a truck that at least one time has

been used on **12-DEC-2017**.

(2) 2 marks

Find the last names ("**last_name**" key) of all drivers who performed no trips so far.

(3) 2 marks

For each trip find the total number of legs. List a trip number ("**number**" key) and the total number of legs.

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 **driver** the documents that contain information about the drivers who live in a country different from Australia.

(5) 2 marks

Remove a description of leg number 2 from a trip number 5.

Submit