School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSCI235
Database Systems
Wollongong Campus

Examination Paper
Spring Session 2018

Exam duration	3 hours
Weighting	60%
Items permitted by examiner	None
Aids supplied	None
Directions to students	6 questions to be answered.

This exam paper must not be removed from the exam venue

The questions 2, 3, and 4 refer to the relational tables created through processing of `CREATE TABLE` statements listed below.

```
CREATE TABLE EMPLOYEE(
        E#              NUMBER(12)      NOT NULL,
        NAME            VARCHAR(50)     NOT NULL,
        DOB             DATE                    ,
        HIREDATE        DATE            NOT NULL,
        CONSTRAINT EMPLOYEE_PKEY PRIMARY KEY(E#) );

CREATE TABLE DRIVER(
        E#              NUMBER(12)      NOT NULL,
        L#              NUMBER(8)       NOT NULL,
        STATUS          VARCHAR(10)     NOT NULL,
        CONSTRAINT DRIVER_PKEY PRIMARY KEY(E#),
        CONSTRAINT DRIVER_UNIQUE UNIQUE(L#),
        CONSTRAINT DRIVER_FKEY FOREIGN KEY(E#) REFERENCES EMPLOYEE(E#),
        CONSTRAINT DRIVER_STATUS CHECK ( STATUS IN
                        ('AVAILABLE', 'BUSY', 'ON LEAVE')) )

CREATE TABLE MECHANIC(
        E#              NUMBER(12)      NOT NULL,
        L#              NUMBER(8)       NOT NULL,
        STATUS          VARCHAR(10)     NOT NULL,
        CONSTRAINT MECHANIC_PKEY PRIMARY KEY(E#),
        CONSTRAINT MECHANIC_UNIQUE UNIQUE(L#),
        CONSTRAINT MECHANIC_FKEY FOREIGN KEY(E#) REFERENCES EMPLOYEE(E#),
        CONSTRAINT MECHANIC_STATUS CHECK ( STATUS IN
                        ('AVAILABLE', 'BUSY', 'ON_LEAVE')) );

CREATE TABLE TRUCK(
        REG#            VARCHAR(10)     NOT NULL,
        CAPACITY        NUMBER(7)       NOT NULL,
        WEIGHT          NUMBER(5)       NOT NULL,
        STATUS          VARCHAR(10)     NOT NULL,
        CONSTRAINT TRUCK_PKEY PRIMARY KEY(REG#),
        CONSTRAINT TRUCK_STATUS CHECK ( STATUS IN
                        ('AVAILABLE', 'USED', 'MAINTAINED')) );

CREATE TABLE TRIP(
        T#              NUMBER(10)      NOT NULL,
        L#              NUMBER(8)       NOT NULL,
        REG#            VARCHAR(10)     NOT NULL,
        TRIP_DATE       DATE            NOT NULL,
        CONSTRAINT TRIP_PKEY PRIMARY KEY (T#),
        CONSTRAINT TRIP_FKEY1 FOREIGN KEY (L#) REFERENCES DRIVER(L#),
        CONSTRAINT TRIP_FKEY2 FOREIGN KEY (REG#) REFERENCES TRUCK(REG#)

CREATE TABLE TRIPLEG(
        T#              NUMBER(10)      NOT NULL,
        LEG#            NUMBER(2)       NOT NULL,
        DEPARTURE       VARCHAR(30)     NOT NULL,
        DESTINATION     VARCHAR(30)     NOT NULL,
        CONSTRAINT TRIPLEG_PKEY PRIMARY KEY (T#, LEG#),
        CONSTRAINT TRIPLEG_UNIQUE UNIQUE(T#, DEPARTURE, DESTINATION),
        CONSTRAINT TRIPLEG_FKEY1 FOREIGN KEY (T#) REFERENCES TRIP(T#) );
```

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. **Justify your answer.**

A correct guess without the comprehensive justifications scores no marks!

(1) 3 marks

R = (A, B, C, D).
A → B
A → C
BC → D

(2) 3 marks

R = (A, B, C, D).
A → B
B → C
C → D

(3) 3 marks

R = (A, B, C, D).
AB → CD
CD → AB

(4) 1 mark

R = (A, B, C, D).
No functional dependencies are valid in a relational schema R.

QUESTION 2 (10 marks)

This question is related to a sample database created through processing of `CREATE TABLE` statements listed on a page 2 of the examination paper.

Write a stored PL/SQL procedure that inserts into the database information about a new trip completed by a driver. The procedure must verify the following consistency constraint before information about a new trip can be inserted into a relational table `TRIP`.

A truck cannot be used for more than 2 trips on the same day.

If verification of the consistency constraint fails then information about a new trip must not be inserted into the database. Otherwise information must be permanently saved in the database.

The procedure must have the following two input parameters:

(1) a licence number of a driver who performed a trip,
(2) registration number of a truck used for a trip.

A trip number must be automatically created by the procedure. A trip date is the same as moment in time when the procedure is processed.

You do not need to write SQL statement that stores the procedure in a data dictionary.

QUESTION 3 (10 marks)

This question is related to a sample database created through processing of `CREATE TABLE` statements listed on a page 2 of the examination paper.

(1) 3 marks
Write SQL statements that modify the structures and contents of the sample database such after a modification the database contains information about the total number of times each truck has been used for the trips. The new trucks that have never been used for any trip must have the total number of trips set to zero.

Note, that after a structural modification listed above all relational schemas of the sample database must still be in BCNF.

(2) 7 marks
Write a row database trigger that automatically modifies the total number of times a truck has been used for the trips whenever information about a new trip is inserted into the database or information about an old trip is deleted from the database. Beware of "mutating table" error.

QUESTION 4 (10 marks)

This question is related to a sample database created through processing of `CREATE TABLE` statements listed on a page 2 of the examination paper.

A relational table `AVGYEAR` has been created in the following way.

```
CREATE TABLE AVGYEAR(
  YEAR       NUMBER(4)  NOT NULL,
  AVGLEN     NUMBER(5,2) NOT NULL,
    CONSTRAINT AVGYEAR_PKEY PRIMARY KEY(YEAR) );
```

A relational table `AVGYEAR` has been filled with information about the average lengths of all trips in different years.

The following PL/SQL stored procedure finds an average length of all trips performed in a given year and updates a relational table `AVGYEAR`.

```
CREATE OR REPLACE PROCEDURE UPD_AVGYEAR(trip_year IN VARCHAR ) IS
tottrip, totleg NUMBER;
BEGIN
 SELECT COUNT(*)
 INTO tottrip
 FROM TRIP
 WHERE TO_CHAR(TRIP_DATE,'YYYY') = YEAR;

 SELECT COUNT(*)
 INTO totleg
 FROM TRIP JOIN TRIPLEG
           ON TRIP.T# = TRIPLEG.T#
 WHERE TO_CHAR(TRIP_DATE,'YYYY') = YEAR;

 UPDATE AVGYEAR
 SET AVGLEN = totleg/tottrip
 WHERE YEAR = trip_year;

 COMMIT;
END;
/
```

Assume that a procedure `UPD_AVGYEAR` is concurrently processed with the other database applications that change information about the trips performed by the drivers and the detailed descriptions of the trips.

Decide what isolation level the procedure `UPD_AVGYEAR` should be processed at and justify your decision.

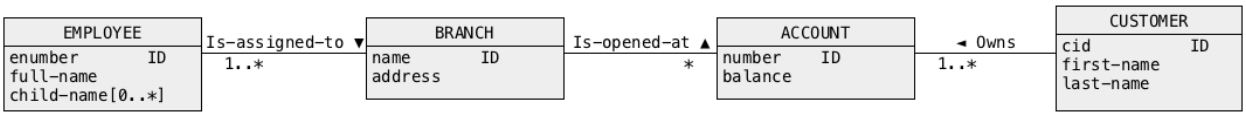
You have the following two options: `READ COMMITTED` or `SERIALIZABLE`.

If you decide that the procedure can be processed at `READ COMMITTED` level then as a justification of your decision provide a proof that any concurrent processing of the procedure at `READ COMMITTED` level does not corrupt a sample database.

If you decide that the procedure can be processed at `SERIALIZABLE` level then as a justification provide a sample concurrent processing of the procedure at `READ COMMITTED` level and the other transactions that corrupts a database. You can apply the two-dimensional visualisation of concurrent processing of database transactions used in the lecture slides and Assignment 2.

QUESTION 5 (10 marks)

Consider the following conceptual schema of a sample a database that contains information about the bank accounts opened at the bank branches, customers who own the accounts, and the employees assigned to the bank branches.



Write a sample BSON document whose structure is consistent with a conceptual schema given above. Your document must contain information about at least one branch, two employees assigned to the branch, two accounts opened at the branch and two customers.

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. Implementation of each query is worth 2 marks.

(1) Find the first and the last name of all drivers who at least once performed a trip on a truck with a registration `PKR856`.

(2) Find the numbers ("number" key) of all trips that originated in Sydney.

(3) Find the total number of trips performed by each driver. For each driver, list his/her licence number and the total number pf trips performed.

Use either a method `remove()` or a method `update()` to write the implementations of the following data manipulation operations. Implementation of each data manipulation operation is worth 2 marks.

(4) Delete from a collection driver the documents that contain information about the drivers whose first name is `James` and the last name is `Bond` or whose live in `Perth`.

(5) Change a departure city of the first leg in a trip number `5` to `Brisbane`.