UOW AUSTRALIA

School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSCI235
Database Systems
Wollongong Campus

Examination Paper
Autumn Session 2018

Exam duration	3 hours
Weighting	60 %
Items permitted by examiner	*None*
Aids supplied	None
Directions to students	6 questions to be answered. Answer each question in a separate booklet.

This exam paper must not be removed from the exam venue

Question 1 **(9 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) R = (A, B, C, D).

A, B $\rightarrow$ C

C $\rightarrow$ D

(3 marks)

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

A $\rightarrow$ B

B $\rightarrow$ C

(3 marks)

(3) Sale (date, customer#, product#, vendor, salesrep)

date, customer#, product# $\rightarrow$ salesrep

date, customer#, product# $\rightarrow$ vendor

customer# $\rightarrow$ salesrep

(3 marks)

THE QUESTIONS 2, 3 and 4 REFER TO THE RELATIONAL TABLES LISTED BELOW

The schemas of relational tables, specifications of primary, candidate, foreign keys and check constraints are given below.

```
CREATE TABLE SKILL(
    sname          VARCHAR(30)     NOT NULL, /* Skill name  */
    CONSTRAINT SKILL_pkey PRIMARY KEY (sname) );
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                 */
    state          VARCHAR(20) NOT NULL,   /* State                */
    phone          NUMBER(10)  NOT NULL,   /* Phone number         */
    email          VARCHAR(50),            /* E-mail address       */
    CONSTRAINT APPLICANT_pkey PRIMARY KEY (anumber) );
CREATE TABLE EMPLOYER(                     /* Employers            */
    ename          VARCHAR(100)NOT NULL,   /* Employer name        */
    city           VARCHAR(30) NOT NULL,   /* City                 */
    state          VARCHAR(20) NOT NULL,   /* State                */
    phone          NUMBER(10)  NOT NULL,   /* Phone number         */
    fax            NUMBER(10),             /* Fax number           */
    email          VARCHAR(50)             , /* E-mail address     */
   CONSTRAINT EMPLOYER_pkey PRIMARY KEY (ename) );
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               */
    bonus          NUMBER(9,2),            /* End of year bonus    */
    ename          VARCHAR(100)NOT NULL,   /* Employer name        */
    CONSTRAINT POSITION_pkey PRIMARY KEY (pnumber),
    CONSTRAINT POSITION_fkey FOREIGN KEY (ename) REFERENCES EMPLOYER(ename) );
CREATE TABLE SPOSSESSED(
    anumber        NUMBER(6)   NOT NULL,   /* Applicant number      */
    sname          VARCHAR(30) NOT NULL,   /* Skill name            */
    slevel         NUMBER(2)   NOT NULL,   /* Skill level           */
    CONSTRAINT SPOSSESSED_pkey PRIMARY KEY (anumber, sname),
    CONSTRAINT SPOSSESSED_fkey1 FOREIGN KEY (anumber) REFERENCES APPLICANT
(anumber),
    CONSTRAINT SPOSSESSED_fkey2 FOREIGN KEY (sname) REFERENCES SKILL (sname),
    CONSTRAINT SPOSSESSED_check1 CHECK ( slevel between 1 and 10) );
CREATE TABLE SNEEDED(
    pnumber        NUMBER(8)   NOT NULL,   /* Position number        */
    sname          VARCHAR(30) NOT NULL,   /* Skill name             */
    slevel         NUMBER(2)   NOT NULL,   /* Skill level            */
    CONSTRAINT SNEEDED_pkey PRIMARY KEY (pnumber, sname),
    CONSTRAINT SNEEDED_fkey1 FOREIGN KEY (pnumber)REFERENCES POSITION (pnumber),
    CONSTRAINT SNEEDED_fkey2 FOREIGN KEY (sname) REFERENCES SKILL (sname),
    CONSTRAINT SNEEDED_check1 CHECK ( slevel between 1 and 10) );
CREATE TABLE APPLIES(
    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),
    CONSTRAINT APPLIES_fkey1 FOREIGN KEY (anumber) REFERENCES APPLICANT
(anumber),
    CONSTRAINT APPLIES_fkey2 FOREIGN KEY (pnumber) REFERENCES POSITION (pnumber)
);
```

Question 2 **(11 marks)**

Implement the following PL/SQL stored procedure and function and SQL statements and write comprehensive testing statement to test your implementations. Assume that a stored procedure and stored function must be applied to the sample database listed on page 3 of the examination paper.

(1) Implement a stored PL/SQL procedure `ALLPOSITIONS` to list all the position numbers, titles and the salaries offered by each employer. The procedure takes no parameter.

The names of employer must be listed in the ascending order. The positions offered by each employer must be listed in descending order of their titles. Execute the stored PL/SQL procedure `ALLPOSITIONS`. A fragment of expected sample printout is given below.

```
University of Adelaide:
University of Melbourne:
University of New South Wales:
        6 professor 100000
        1 lecturer 45000
University of Queensland:
        5 professor 200000
…
```

(5 marks)

(2) Implement a stored PL/SQL function `FINDAPPLICANTS` that takes a position number (pnumber) as a parameter, and finds all the applicants who have applied the position and fully satisfied the position needs. The function must return a string of characters that contains a position name, employer and applicant numbers and full names (first name, last name) that applied the position and satisfy the needed skills for a given position.

Hint: An applicant has all the skills that needed means the applicant possesses all (maybe more) skills and those skill levels are equal or higher than the needed skills for that position.

Execute the stored PL/SQL function `FINDAPPLICANTS` for all positions. A fragment of sample printout is given below:

```
1 lecturer, University of New South Wales:
        1 Harry Potter

2 lecturer, University of Wollongong:

3 senior lecturer, University of Technology, Sydney:

4 associate professor, University of Wollongong:
        7 James Bond
        8 Stephen Staunton

5 professor, University of Queensland:

…
```

(6 marks)

Question 3 **(10 marks)**

Write the implementations of the following database triggers. Assume that the implementations of the triggers must be applied to the sample database listed on page 3 of the examination paper.

(1) Implement a `statement trigger` that enforces the following consistency constraint.

An applicant cannot apply more than two positions in one day.

Write SQL statements that comprehensively test the trigger. A comprehensive test must consist of at least two cases for each operation: one accepted and one rejected.

(5 marks)

(2) Implement a `row trigger` that verifies the following consistency constraint.

The value of position number must be continuous.

"Contiguous" in a relationship to the position number means, that the first position must have a number 1, the second position must have a number 2, and each next position must have a number greater by one than a number of the previous position.

Do not need to consider the UPDATING event.

Hint:
To avoid "mutating table" error, insert a line

```
PRAGMA AUTONOMOUS_TRANSACTION
```

At the beginning of `DECLARE` block.

Write SQL statements that comprehensively test the trigger. A comprehensive test must consist of at least two cases for each operation: one accepted and one rejected.

(5 marks)

Question 4 **(10 marks)**

Consider the sample database listed on page 3 of the examination paper.

(1) Show a sample concurrent execution of two database transactions such that the execution reveals a **phantom phenomenon**. Thoroughly explain why the execution reveals a phantom phenomenon. Remember to set an appropriate isolation level for the transactions. **Note, that a sample execution without the detailed explanations scores no marks**.

Use a technique of presentation of concurrent execution of two database transactions that explained to you during the lecture classes and such that the statements of the first transaction are listed on the left-hand side of a page and the statements of the second transaction are listed on the right-hand side of a page. Make sure that each statement starts in a different line to represent a different moment in time when its execution starts.

(5 marks)

(2) Read the database transactions T_1 and T_2 included in SQL scripts `below`.

T_1:

```
UPDATE POSITION
SET bonus = salary / 10
WHERE salary > 100000;

COMMIT;
```

T_2:

```
UPDATE POSITION
SET bonus = NVL(bonus, 0) + &AdditionBonus
WHERE salary < 400000;

SELECT pnumber, title, salary, bonus
FROM POSITION;

COMMIT;
```

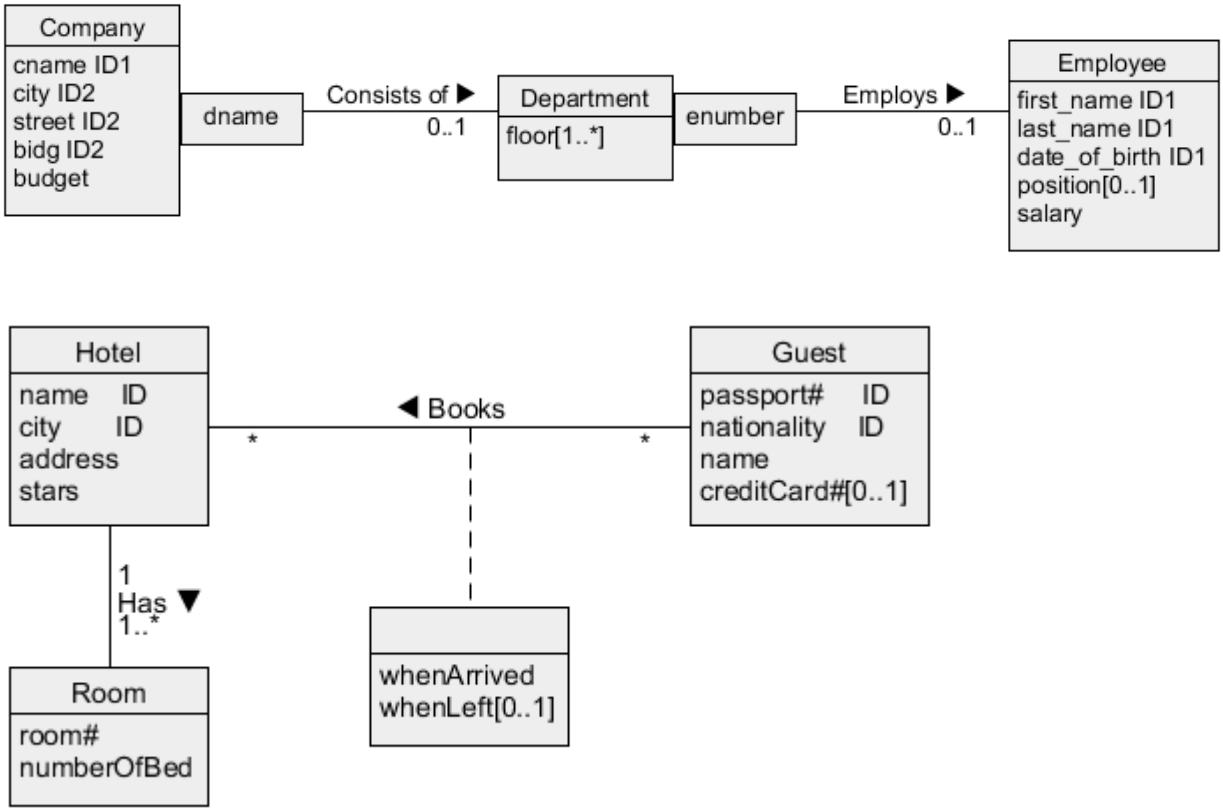
Assume that a database user would like to concurrently run both transactions and assume that in the same moment no other transactions will be processed. What the best isolation levels would you execute each transaction at to avoid the corruption of the database and to achieve the best performance for each transaction? You have 3 options: `READ ONLY`, `READ COMMITTED`, and `SERIALIZABLE`.

Use a technique of presentation of concurrent execution of two database transactions that explained to you during the lecture classes and such that the statements of the first transaction are listed on the left-hand side of a page and the statements of the second transaction are listed on the right-hand side of a page. Make sure that each statement starts in a different line to represent a different moment in time when its execution starts.

(5 marks)

Question 5 **(10 marks)**

Consider the following conceptual schemas representing the sample database domains.



For each one of the conceptual schemas given above creates a sample BSON document (one document for one schema) whose contents are consistent the respective conceptual schema. Your documents must contain information about at least one instance of an object for each one of the classes included in the schemas.

Question 6 **(10 marks)**

A JSON document listed below belongs to a collection of documents `Person` that contains information about the persons work in companies. All documents in a collection `Person` have the same structure as the document listed below.

```
{
  "_id": "57f6f57a51fb4f930ced06f3",
  "index": 1,
  "guid": "be2b8d4f-da16-435d-bf27-5bd66991244a",
  "isActive": true,
  "balance": 1667.16,
  "picture": "http://placehold.it/32x32",
  "age": 29,
  "eyeColor": "brown",
  "name": "Lois Valentine",
  "gender": "female",
  "company": "SPRINGBEE",
  "email": "loisvalentine@spreengbee.com",
  "phone": "+1 (811) 533-2305",
  "address": "948 Calder Place, Dennard, Texas, 8054",
  "about": "Occaecat tempor deserunt id ipsum. Do dolore adipisicing anim in
 occaecat enim et. Mollit labore ad ad do proident aliquip ut cupidatat do
 voluptate culpa do labore sit.\r\n",
  "registered": "2014-08-26T05:21:44 -10:00",
  "latitude": -36.258817,
  "longitude": -47.342096,
  "tags": [
    "enim",
    "quis",
    "enim",
    "non",
    "commodo",
    "ea",
    "exercitation"
  ],
  "friends": [
    {
      "id": 0,
      "name": "Mcintosh Rivera"
    },
    {
      "id": 1,
      "name": "Burt Leach"
    },
    {
      "id": 2,
      "name": "Nichols Alvarado"
    }
  ],
  "greeting": "Hello, Lois Valentine! You have 7 unread messages.",
  "favoriteFruit": "apple"
}
```

Write in a query language of MongoDB database system the implementations of the following queries.

(1) Find the names of people whose friend is `Sonia Bonner`.

(2 marks)

(2) Find the names of people whose tags include both `quis` and `laborum`.

(2 marks)

(3) Find the names of people who have no friends.

(2 marks)

(4) Find the names of people who live in `Tennessee`.

(2 marks)

(5) Find the name and gender of people who work in a company `MOBILDATA`.

(2 marks)

End of examination paper