School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSCI235
Database Systems
Wollongong Campus
Liverpool Campus

Final Examination Paper
Spring Session 2023

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

CHEAT SHEET

Armstrong Axioms

Let **X**, **Y**, **Z** be the nonempty subsets of **$\{A_1, ..., A_n\}$**
(i) If **Y ⊆ X** then **X → Y** (**reflexivity axiom**)
(ii) If **X → Y** then **X, Z → Y, Z** (**augmentation axiom**)
(iii) If **X → Y** and **Y → Z** then **X → Z** (**transitivity axiom**)

Other inference rules

If **X → Y** and **X → Z** then **X → Y, Z** (**union rule**)
If **X → Y** and **W, Y → Z** then **W, X → Z** (**pseudotransitivity rule**)
If **X → Y** and **Z ⊆ Y** then **X → Z** (**decomposition rule** or **reduce right hand side rule**)
If **X → Y** then **X, Z → Y** (**extend left hand side rule**)

Statement database triggers

```
CREATE OR REPLACE TRIGGER <trigger name>
AFTER/BEFORE <event> ON<table>
  BEGIN       -- Statement triggers have no FOR EACH ROW clause!
  ...
  END;
```

Row database triggers

```
CREATE OR REPLACE TRIGGER <trigger name>
AFTER/BEFORE <event> ON<table>
  BEGIN
  FOR EACH ROW -- Row triggers must have FOR EACH ROW clause!
  WHEN <condition>
  ...
  END;
```

Database link

```
CREATE DATABASE LINK "DB.DATA-PCxx"
CONNECT TO <user-name>
IDENTIFIED BY <password>
USING 'data-pc07.adeis.uow.edu.au:1521/db';
```

View serializability

Concurrent execution of database transactions is **view serializable** if there **exists a possible serial execution of the same set of transactions** such that in both executions each **transaction reads the same values and the final states of the database are the same**

Conflict serializability

Concurrent execution of database transactions is **conflict serializable** if there exists **a possible serial execution of the same set of transactions** such that in both executions **the order of conflicting operations is the same**

Aggregation framework

```
db.orders.aggregate([ {$project:{"Company":"$CUSTOMER.company name","_id":0}}])
db.orders.aggregate([{$match:{"SUPPLIER.city":"Sandvika"}}])
db.orders.aggregate([{$unwind:"$SUPPLIER.supplies"},
                     {$project:{"supplier":"$SUPPLIER.company name",
                           "product":"$SUPPLIER.supplies.PRODUCT.product name",
                           "_id":0}}])
db.orders.aggregate([{$match:{"SUPPLIER":{$exists:true}}},
                     {$group:{"_id": null,"total":{$sum:1}}}])}
```

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 relational schemas* such 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)
AB → C
C → D
A → D
```

(2) 4 marks

```
R = (A, B, C, D)
A → BC
C → D
C → A
```

(3) 3 marks

```
R = (A, B, C, D)
B → C
B → D
```

QUESTION 2 (10 marks)

Assume that SQL script dbcreate.sql contains the following CREATE TABLE statements.

```
CREATE TABLE CUSTOMER(
C_CUSTKEY         NUMBER(12)  NOT NULL,
C_NAME            VARCHAR(25) NOT NULL,
C_ADDRESS         VARCHAR(40) NOT NULL,
C_PHONE           CHAR(15)    NOT NULL,
C_ACCTBAL         NUMBER(12,2)NOT NULL,
     CONSTRAINT CUSTOMER_PKEY PRIMARY KEY(C_CUSTKEY),
     CONSTRAINT CUSTOMER_CHECK1 CHECK(C_CUSTKEY >= 0) );

CREATE TABLE ORDERS(
O_ORDERKEY       NUMBER(12)   NOT NULL,
O_CUSTKEY        NUMBER(12)   NOT NULL,
O_ORDERSTATUS    CHAR(1)      NOT NULL,
O_TOTALPRICE     NUMBER(12,2) NOT NULL,
O_ORDERDATE      DATE         NOT NULL,
     CONSTRAINT ORDERS_PKEY PRIMARY KEY (O_ORDERKEY),
     CONSTRAINT ORDERS_FKEY1 FOREIGN KEY (O_CUSTKEY)
          REFERENCES CUSTOMER(C_CUSTKEY),
     CONSTRAINT ORDER_CHECK1 CHECK( O_TOTALPRICE >= 0) );
```

The SQL script dbcreate.sql was used to create the empty relational tables on a database server located at data-pc01.adeis.uow.edu.au,port 1521, SID=db within an account of user scott with a password tiger. Later on, a number of database applications processed over a longer period of time filled the relational tables with data.

(1) 1 mark
Explain how you would re-create the empty relational tables CUSTOMER and ORDERS on a database server located at data-pc02.adeis.uow.edu.au, port=1521, SID=db within an account of a user scott with a password tiger.

(2) 1 mark
Write SQL statement to create a database link from a database server at data-pc01.adeis.uow.edu.au to a database server data-pc02.adeis.uow.edu.au.

(3) 1 mark
Write SQL statements to create the synonyms of relational tables created on a database server at data-pc02.adeis.uow.edu.au. Assume that you are connected to a database server located at data-pc01.adeis.uow.edu.au.

(4) 4 marks
Assume that a frequent customer is defined as a customer who submitted more than 100 orders.

Write SQL statements to move information about the frequent customers from a database server located at data-pc01.adeis.uow.edu.au to a database server located at data-pc02.adeis.uow.edu.au. "Move" means that after a data transfer all information about the frequent customers must be removed from a database server located at data-pc01.adeis.uow.edu.au.

Your solution must use a database link and synonyms created in the steps (1) and (2).

(5) 3 marks
Write SELECT statement that finds the total number of orders submitted by each customer. For each customer list customer key and customer name and the total number submitted by a customer. Assume that some customers submitted no orders and in all such cases the total number of orders must listed as 0 (zero).

QUESTION 3 (10 marks)

This question is related to a sample database created through processing of `CREATE TABLE` statements listed below.

```
CREATE TABLE CUSTOMER(
C_CUSTKEY         NUMBER(12)  NOT NULL,
C_NAME            VARCHAR(25) NOT NULL,
C_ADDRESS         VARCHAR(40) NOT NULL,
C_PHONE           CHAR(15)    NOT NULL,
C_ACCTBAL         NUMBER(12,2)NOT NULL,
     CONSTRAINT CUSTOMER_PKEY PRIMARY KEY(C_CUSTKEY),
     CONSTRAINT CUSTOMER_CHECK1 CHECK(C_CUSTKEY >= 0) );

CREATE TABLE ORDERS(
O_ORDERKEY       NUMBER(12)   NOT NULL,
O_CUSTKEY        NUMBER(12)   NOT NULL,
O_ORDERSTATUS    CHAR(1)      NOT NULL,
O_TOTALPRICE     NUMBER(12,2) NOT NULL,
O_ORDERDATE      DATE         NOT NULL,
     CONSTRAINT ORDERS_PKEY PRIMARY KEY (O_ORDERKEY),
     CONSTRAINT ORDERS_FKEY1 FOREIGN KEY (O_CUSTKEY)
          REFERENCES CUSTOMER(C_CUSTKEY),
     CONSTRAINT ORDER_CHECK1 CHECK( O_TOTALPRICE >= 0) );
```

(1) 2 marks
Write SQL statements that change the structures of a sample database such that it is possible to store information about the statuses of customers.

We would like to distinguish the following categories of customers: `rare`, `typical` and `frequent`.

A customer is `rare` if he/she submitted less than 10 orders. A customer is `frequent` if he/she submitted more than 100 orders. A customer is `typical` if he/she is not `rare` and he/she is not `frequent`.

Note, that after the structural modifications all relational tables must be in BCNF and we try to minimize the total number of relational tables in a sample database.

(2) 2 marks
Write SQL statement that fills the new structures of a sample database created in a step (1) with data consistent with the present state of a database.

(3) 3 marks
Write an implementation of a row trigger that updates a status of a customer when a new order is submitted by a customer.

(4) 3 marks
Write an implementation of a statement trigger that updates a status of a customer when a new order is submitted by a customer.

QUESTION 4 (10 marks)

This question is related to a sample database created through processing of `CREATE TABLE` statements listed below.

```
CREATE TABLE CUSTOMER(
C_CUSTKEY         NUMBER(12)  NOT NULL,
C_NAME            VARCHAR(25) NOT NULL,
C_ADDRESS         VARCHAR(40) NOT NULL,
C_PHONE           CHAR(15)    NOT NULL,
C_ACCTBAL         NUMBER(12,2)NOT NULL,
     CONSTRAINT CUSTOMER_PKEY PRIMARY KEY(C_CUSTKEY),
     CONSTRAINT CUSTOMER_CHECK1 CHECK(C_CUSTKEY >= 0) );

CREATE TABLE ORDERS(
O_ORDERKEY       NUMBER(12)   NOT NULL,
O_CUSTKEY        NUMBER(12)   NOT NULL,
O_ORDERSTATUS    CHAR(1)      NOT NULL,
O_TOTALPRICE     NUMBER(12,2) NOT NULL,
O_ORDERDATE      DATE         NOT NULL,
     CONSTRAINT ORDERS_PKEY PRIMARY KEY (O_ORDERKEY),
     CONSTRAINT ORDERS_FKEY1 FOREIGN KEY (O_CUSTKEY)
          REFERENCES CUSTOMER(C_CUSTKEY),
     CONSTRAINT ORDER_CHECK1 CHECK( O_TOTALPRICE >= 0) );
```

Consider the following SQL script.

```
/* Transaction T */
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

UPDATE CUSTOMER
SET C_C_ACCTBAL = C_ACCTBAL + 100
WHERE C_CUSTKEY IN (SELECT O_CUSTKEY
                    FROM ORDERS
                    WHERE O_TOTALPRICE > 200);

UPDATE CUSTOMER
SET C_C_ACCTBAL = C_ACCTBAL + 100
WHERE C_CUSTKEY IN (SELECT O_CUSTKEY
                    FROM ORDERS
                    WHERE O_TOTALPRICE <= 100);
COMMIT;
```

Assume that the script listed above is processed as a database transaction `T` at `READ COMMITTED` isolation level.

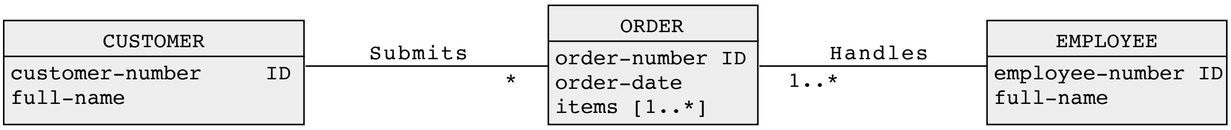
(1) 4 marks
Show a sample concurrent execution of a database transaction `T` together with another database transaction also running at `READ COMMITTED` isolation level, such that one of the transactions (`T` or another database transaction) is aborted and rolled back. When visualizing a concurrent execution use a technique of two-dimensional diagrams presented to you during the lecture classes.

(2) 3 marks
Explain all details on why a concurrent execution of a transaction `T` or another database transaction also running at `READ COMMITTED` isolation level ends with an automatic abort and rollback of one of the transactions.

(3) 3 marks
Show how a transaction `T` must be modified to avoid abort and rollback. A modified transaction still must be processed `at READ COMMITTED` level. Write a new version of the script with a transaction `T`.

QUESTION 5 (10 marks)

Consider a conceptual schema given below. The schema represents a sample database domain where customers submit orders handled by employees.



(1) 7 marks
Transform a conceptual schema given above into a logical schema of BSON documents.

Draw a logical schema of BSON documents obtained from a transformation of a conceptual schema above. To draw a logical schema, use a notation explained to you during the lecture classes.

(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 customer, two orders submitted by a customer and two employees two customers who handled the orders.

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":"11-NOV-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 `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 `11-NOV-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`.