Information and Database Management Systems I

(CIS 4301 UF Online)

Fall 2019 Instructor:

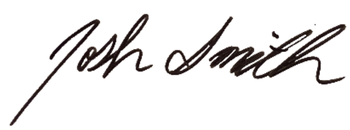
Dr. Markus Schneider

Homework 5

Name:	Joshua Main-Smith
UFID:	7160-7655
Email Address:	joshuamainsmith@ufl.edu

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.



____

Signature

For scoring use only:

	Maximum	Received
Exercise 1	20	
Exercise 2	25	
Exercise 3	25	
Exercise 4	15	
Exercise 5	15	
Total	100	

Exercise 1 - Normalization [20 points]

Consider the following table which is used to store students and courses records.

UFID	Course_ID	Grade	Student_Name	Department	Tuition Fee	Instructor
4114123	COP01, COP02, COP03	A, A, B	John Smith	CISE	250	James, Andrew, Peter
3124234	BU01, BU02	B, B	Roger Hicks	Business	300	Alan, Alan

Please note that *Tuition Fee* depends on the department.

1. Normalize the table to the 1st Normal Form and explain your answer. [5 points]
2. Explain the criteria for 2nd Normal Form and normalize the table you obtained from the previous part to meet them. Then explain which anomalies can occur with your answer. [5 points]
3. Explain the criteria for 3rd Normal Form and normalize the table you obtained for the previous question to meet them. [5 points]
4. Explain if the tables you obtained for the previous question is in BCNF and, if not, normalize it to BCNF. [5 points]

1.

UFID	Course ID	Grade	Student Name	Department	Tuition Fee	Instructor
4114123	COP01	A	John Smith	CISE	250	James
4114123	COP02	A	John Smith	CISE	250	Andrew
4114123	COP03	B	John Smith	CISE	250	Peter
3124234	BU01	B	Roger Hicks	Business	300	Alan
3124234	BU02	B	Roger Hicks	Business	300	Alan

I added rows to eliminate multiple values in any given cell (unique rows for each of John Smith's classes and unique rows for Roger Hicks' classes) and each of the columns are already of the same type. I used the UFID as the primary key to uniquely identify the rows.

2.

To qualify for 2NF, all attributes that are not a key must be dependent on a key. I decided to split the tables into a student table (containing an UFID, student name, department and tuition fee) and a course table (containing the course ID, grade received from the course, and the instructor that teaches the course). These two tables are connected by the junction table, containing the students UFID and the ID of the course they're taking. One anomaly to note is the tuition fee appeared to be uniquely associated with the department, which is why I included it in the student table. If it turns out that the tuition fee is instead uniquely associated with a course, then I would move the tuition fee field to the course table.

Student Table

UFID	Student Name	Department	Tuition Fee
4114123	John Smith	CISE	250
3124234	Roger Hicks	Business	300

Course Table

Course ID	Grade	Instructor
COP01	A	James
COP02	A	Andrew
COP03	B	Peter
BU01	B	Alan
BU02	B	Alan

Junction

UFID	Course ID
4114123	**COP01**
4114123	**COP02**
4114123	**COP03**
3124234	**BU01**
3124234	**BU02**

3.

In order to qualify for 3NF, all fields must be determined only by the key. In the previous example, tuition fee could be determined if the department is known. To solve for 3NF, I added a Department Cost table which contains the department as the primary key and its corresponding tuition fee cost. I then removed the Tuition Fee field from the Student Table.

Student Table

UFID	Student Name	Department
4114123	John Smith	CISE
3124234	Roger Hicks	Business

Department Cost

Department	Tuition Fee
CISE	250
Business	300

Course Table

Course ID	Grade	Instructor
COP01	A	James
COP02	A	Andrew
COP03	B	Peter
BU01	B	Alan
BU02	B	Alan

Junction

UFID	Course ID
4114123	**COP01**
4114123	**COP02**
4114123	**COP03**
3124234	**BU01**
3124234	**BU02**

4.

To qualify for BCNF, a table needs to be in 3NF and for any A -> B, A should be a super key. The previous tables are not I BCNF since the instructors (a non-prime attribute) can identify their corresponding course ID (a prime attribute). To convert it to BCNF, I added a professor table with PID as a primary key and the instructors' name.

Student Table

UFID	Student Name	Department
4114123	John Smith	CISE
3124234	Roger Hicks	Business

Department Cost

Department	Tuition Fee
CISE	250
Business	300

Course Table

Course ID	Grade	Instructor
COP01	A	1
COP02	A	2
COP03	B	3
BU01	B	4
BU02	B	4

Professor Table

PID	Instructor
1	James
2	Andrew
3	Peter
4	Alan

Junction

UFID	Course ID
4114123	**COP01**
4114123	**COP02**
4114123	**COP03**
3124234	**BU01**
3124234	**BU02**

Exercise 2 – Normal Forms [25 points]

Consider the relation schema $R = (A, B, C, D, E)$ for the following questions.

1. Assume we have the following functional dependencies:
   - $AB \rightarrow C$
   - $C \rightarrow D$
   - $B \rightarrow E$

   Briefly explain if the relation R is in 2NF. If not, what modifications can be made to normalize it into 2NF? [5 points]
2. Is R in 2NF with the following functional dependencies? If not, normalize it. [5 points]
   - $A \rightarrow BC$
   - $AD \rightarrow E$
   - $B \rightarrow C$
3. Are the relations from the answer of question 2 in 3NF? If not, normalize it. [5 points]
4. Briefly explain if the relation R is in 2NF. [2 points].
   - $A \rightarrow BCDE$
   - $BC \rightarrow ADE$
   - $D \rightarrow E$

   Further, is R in 3NF? If not, what modifications can be made to normalize it into 3NF? [3 points]
5. Assume we have the following functional dependencies:
   - $AB \rightarrow D$
   - $C \rightarrow E$
   - $E \rightarrow C$
   - $C \rightarrow A$
   - $A \rightarrow C$

   We decompose R into schemas R1(ABC) and R2(ABDE). Show whether it is dependency preserving by using one of the algorithms that covered in the lecture. [5 points]

1.

Yes, with the primary key being AB since $AB^+ = R$. C -> D violates 3NF since C is not a super key and D isn't part of any key in R.

2.

Yes, with AD being a key with $AD^+ = R$. B -> C violates 3NF since B isn't a super key and C isn't part of a key in R.

3.

No, because of B -> C. This can be normalized by removing B -> C, which still gives $AD^+ = ABCDE = R$.

4.

Yes, with A being a primary key. It is not in 3NF since D -> E and BC -> ADE both violate 3NF, with D or BC not being a super key and E not being part of any key. This can be normalized to 3NF by removing D -> E and BC -> ADE, which would give A^+ = ABCDE = R.

5.

R_1: C -> A and A -> C

R_2: AB -> D

It's not dependency preserving since there's no way to obtain C -> E and E -> C.

Exercise 3 – Lossless Join Decomposition [25 points]

1. For the relation schema R = (ABCDEF) and functional dependencies F = $\{AB \to C, AC \to B, AD \to E, B \to D, BC \to A, E \to F\}$, determine whether the following decomposition is lossless. Also, determine if it is dependency preserving.

   P = {R1(AB), R2(BC), R3(ABDE), R4(EF)} [10 points]

2. Consider the relation schema R = (ABCDE).

   a. For the functional dependencies F = $\{AB \to C, C \to E, B \to D, E \to A\}$, is P = {R1 (BCD), R2 (ACE)} a lossless decomposition? Show all the steps. [5 points]

   b. For the functional dependencies F = $\{A \to CD, B \to CE, E \to B\}$, give a lossless-join decomposition of R into BCNF. [5 points]

   c. For the functional dependencies F = $\{A \to CD, B \to CE, E \to B\}$, give a lossless-join decomposition of R into 3NF preserving functional dependencies. [5 points]

1.

Lossless:

Not lossless. To be lossless, R_1 and R_2 = R_1 or R_1 and R_2 = R_2.

So, R_1(AB) and R_2(BC) = B with B -> R_1 and B -> R_2 not being true (B^+ = BD). So, the decomposition is not lossless.

Dependency Preserving:

R_1:

R_2:

R_3: AD -> E, B -> D

R_4: E -> F

It is not dependency preserving since AB -> C, AC -> B and BC -> A can't be reached.

2.

a. It is lossless because: BCD and ACE = C, with C^+ = ACE = R_2.

b. R_1(ACD) and R_2(ABE) = A. A^+ = ACD = R_1.

c.

Exercise 4 - Normalization [15 points]

Suppose we have a relation schema R(A, B, C, D, E, F, G) and a set of functional dependencies $F = \{BCD \rightarrow A, BC \rightarrow E, A \rightarrow F, F \rightarrow G, C \rightarrow D, A \rightarrow G, A \rightarrow B\}$. Decompose R into 3NF by using the 3NF synthesis algorithm. Show all steps and argue precisely. Is this decomposition also in BCNF? If so, why? If not, why not? [15 points]

Exercise 4

Reducing the left and right side of the FDs, we get:

F= {BC->A, BC->E, A->F, F->G, C->D, A->B}

= {BC->AE, A->BF, F->G, C->D}

R can then be decomposed to:

R_1(ABF): A->BF

R_2(C,D): C->D

R_3(ABCE): BC->AE

R_4(FG): F->G

A key that's also in one of the above relations is BC, where

BC^+ = ABCDEFG = R

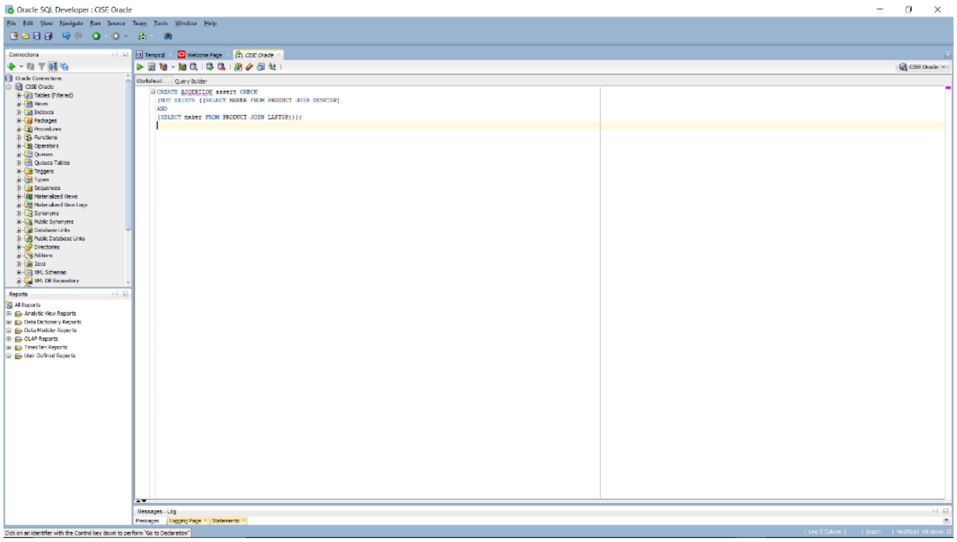
Exercise 5 – Integrity Constraints [15 points]

Consider the following tables:

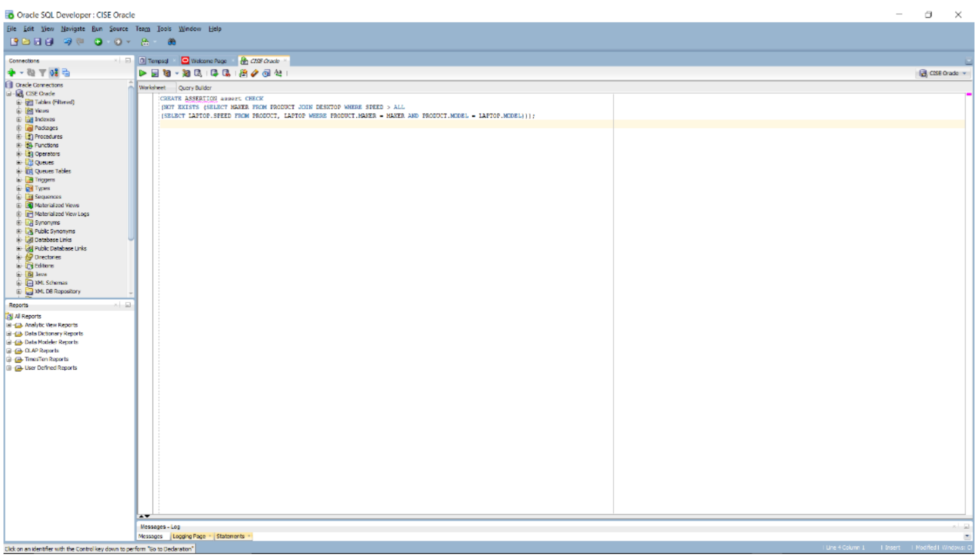
```
CREATE TABLE PRODUCT
(MAKER VARCHAR2(50),
MODEL VARCHAR2(50),
TYPE VARCHAR2(30));

CREATE TABLE DESKTOP
(MODEL VARCHAR2(50) NOT NULL,
SPEED NUMBER(8),
RAM VARCHAR2(30),
HD VARCHAR2(30),
PRICE NUMBER(8));

CREATE TABLE LAPTOP
(MODEL VARCHAR2(50) NOT NULL,
SPEED NUMBER(8),
RAM VARCHAR2(30),
HD VARCHAR2(30),
SCREEN VARCHAR2(30),
PRICE NUMBER(8));

CREATE PRINTER
(MODEL VARCHAR2(50) NOT NULL,
COLOR VARCHAR2(30),
TYPE VARCHAR2(30),
PRICE NUMBER(8));
```

1. Write a check condition to ensure that no manufacturer of desktops also makes laptops. [3 points]
2. Write a check condition to ensure that a manufacturer of a desktop also makes a laptop with at least the same processor speed. [4 points]
3. Create a trigger that checks that there is no lower priced desktop with the same speed when the price of a desktop is updated. [4 points]
4. Create a trigger that checks if the model number exists in the *Product* table when a new printer is inserted. [4 points]

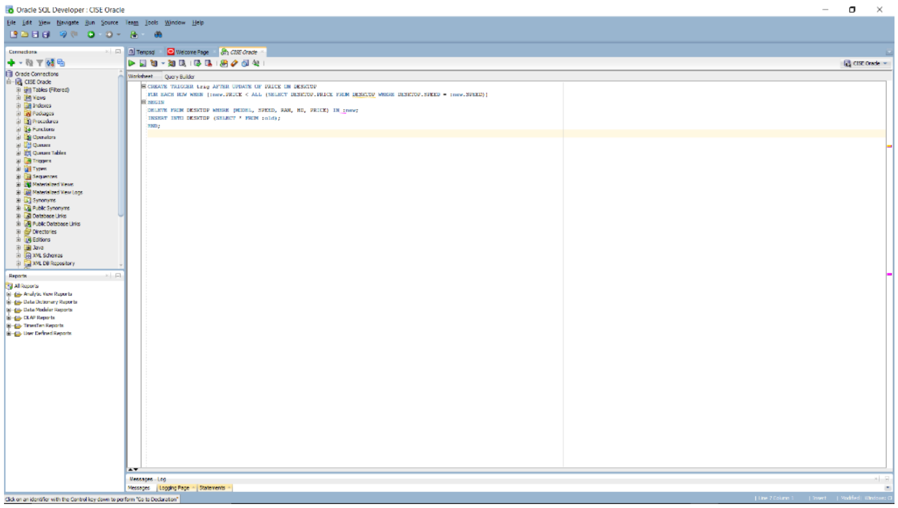
1.


Oracle SQL Developer : CISE Oracle
CREATE ASSERTION assert CHECK
(NOT EXISTS ((SELECT MAKER FROM PRODUCT JOIN DESKTOP)
AND
(SELECT maker FROM PRODUCT JOIN LAPTOP)));

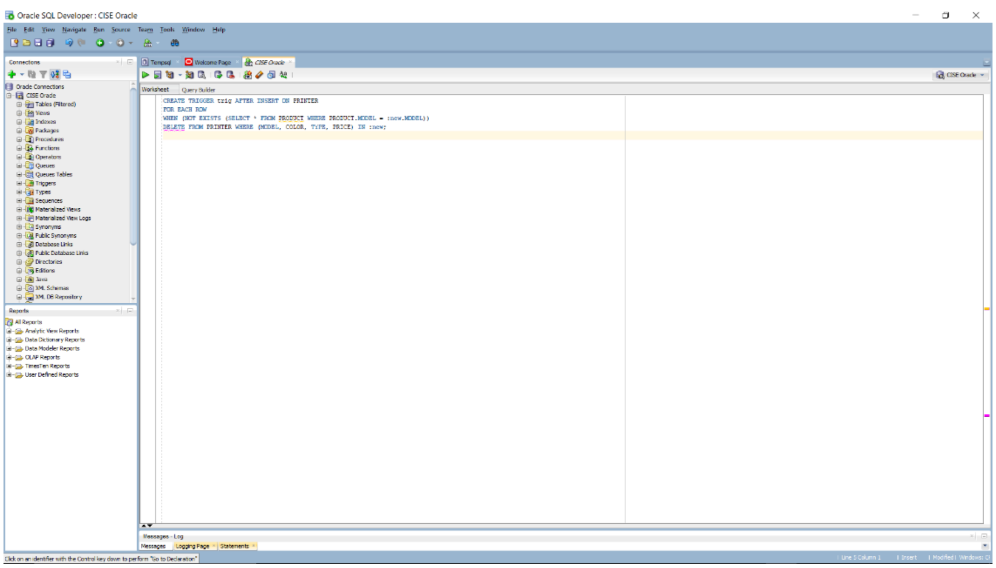
2.


Oracle SQL Developer : CISE Oracle
CREATE ASSERTION assert CHECK
(NOT EXISTS (SELECT MAKER FROM PRODUCT JOIN DESKTOP WHERE SPEED > ALL
(SELECT LAPTOP.SPEED FROM PRODUCT, LAPTOP WHERE PRODUCT.MAKER = MAKER AND PRODUCT.MODEL = LAPTOP.MODEL)));

3.


Oracle SQL Developer : CISE Oracle
CREATE TRIGGER trig AFTER UPDATE OF PRICE ON DESKTOP
FOR EACH ROW WHEN (:new.PRICE < ALL (SELECT DESKTOP.PRICE FROM DESKTOP WHERE DESKTOP.SPEED = :new.SPEED))
BEGIN
DELETE FROM DESKTOP WHERE (MODEL, SPEED, RAM, HD, PRICE) IN :new;
INSERT INTO DESKTOP (SELECT * FROM :old);
END;

4.


Oracle SQL Developer : CISE Oracle
CREATE TRIGGER trig AFTER INSERT ON PRINTER
FOR EACH ROW
WHEN (NOT EXISTS (SELECT * FROM PRODUCT WHERE PRODUCT.MODEL = :new.MODEL))
DELETE FROM PRINTER WHERE (MODEL, COLOR, TYPE, PRICE) IN :new;