Database Management Systems (COP 5725)

Spring 2014

Instructor:
Dr. Markus Schneider

TAs:
Soham Das
Yan Qiao
Yi Wang

Exam 3 Solutions

Name:	
UFID:	
Email Address:	

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
Question 1	25	
Question 2	25	
Question 3	25	
Question 4	25	
Total	100	

Question 1 (Functional Dependencies) [25 points]

(a) [5 points] Define 3NF and BCNF. Give an example of a relation along with the FDs on it, which is in 3NF, but not in BCNF.

Soln. A relation schema R with associated FDs F is in **third normal form** (**3NF**), if, and only if, it is in 2NF and for each FD $A \rightarrow B \in F$, at least one of the following conditions holds:

− $B \subseteq A$, i.e., the FD $A \rightarrow B$ is trivial.

− A is superkey of R.

− B is (part of) some candidate key of R.

A relation schema R with FDs F is in **Boyce-Codd normal form** (**BCNF**), if, and only if, it is in 3NF and for each FD $A \rightarrow B \in F$, at least one of the following conditions holds:

− $B \subseteq A$, i.e., the FD $A \rightarrow B$ is trivial.

− A is superkey of R.

Example:

− CarIndex(manufacturer, manufacturer-id, model-id)

− consider FDs:

+ FD1: {model-id, manufacturer} → {manufacturer-id}

+ FD2: {manufacturer-id} → {manufacturer}

(b) [6 points] Consider the relational schema R=ABCD and the set F={AC → B, A → C, D → A} of FDs. Find all candidate keys of R and show how you systematically do this.

Soln.

L	B	R
D	A	B
	C	

D^+={A,B,C,D}. No other candidate key is possible, because D should be a part of any candidate key and D itself is a candidate key.

(c) [6 points] Given the database schema R(A,B,C), and a relation r on the schema R, write an SQL command (integrity constraint) to test if B→C holds on r.

Soln. create assertion CheckFD check (NOT EXISTS (select * from r as r1, r as r2 where r1.B=r2.B and r1.C<>r2.C));

(d) [8 points] Consider the relational schema R=ABCD and the two sets F={AC → B, A → C, D → A} and G={A → B, A → C, D → A, D → B} of FDs. Examine if these two sets are equivalent.

Soln. We start with F and check if we can derive all FDs in G.
$A \rightarrow C$ and $D \rightarrow A$ are already present.
We have $A \rightarrow C$ in F. Augmenting A, we have $A \rightarrow AC$. We have $AC \rightarrow B$. Using transitivity, we have $A \rightarrow B$.
We have $D \rightarrow A$ and $A \rightarrow B$ now. So applying transitivity again, we can get $D \rightarrow B$.
Thus we can derive all the FDs in G from F.

Similarly we start at G.
$A \rightarrow C$ and $D \rightarrow A$ are already present.
$AC \rightarrow B$ can be derived from $A \rightarrow B$ by augmentation.
Thus we can derive all the FDs in F from G.

So F and G are equivalent.

Question 2 (Normalization) [25 points]

(a) [2 points] Every relation with 2 attributes is in BCNF. Why?

Soln. Check BCNF condition: for each FD $A \rightarrow B$, A should be a superkey.

(b) [8 points] Let R(A,B,C,D,E,G,H) be a relation schema with the set $F=\{H \rightarrow GD, E \rightarrow D, HD \rightarrow CE, BD \rightarrow A\}$ of FDs. Find a lossless join and dependency preserving 3NF decomposition of R.

Soln. We apply the 3NF Synthesis Algorithm.
We compute F_c in the first step. We perform left reduction on $HD \rightarrow CE$. Since we have $H \rightarrow D$ i.e. $H \rightarrow HD$, we can remove D from the FD. So the reduced FD becomes $H \rightarrow CE$.
Next we perform right reduction. In the FD, $H \rightarrow GD$, D is extraneous, because $H \rightarrow D$ can be derived from $H \rightarrow E$ and $E \rightarrow D$.
So $F_c=\{H \rightarrow CEG, E \rightarrow D, BD \rightarrow A\}$.

Now we look at each FD in F_C and decompose the relation into: {H, G, C, E} ($F_1=\{H \rightarrow CEG\}$), {E, D} ($F_2=\{E \rightarrow D\}$), {B, D, A} ($F_3=\{BD \rightarrow A\}$) is the decomposition. There is no candidate key in any of the relations- so we add the relation {H,B} ($F4=\phi$) in it.

(c) [8 points] Let R(A,B,C,D,E) be a relation schema with the set $F=\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ of FDs. We decompose R into 2 schemas R1(A,B,C) and R2(A,D,E). Show whether this decomposition is lossless join decomposition. Show whether it is dependency preserving.

Soln. Lossless: $(R1 \cap R2) \rightarrow R1 \in F^+$ is present in F: $A \rightarrow BC$

Dependency Preservation: Let us compute F_1 and F_2 that holds on the 2 decomposed relations respectively. $F_1=\{A\rightarrow BC\}$. $F_2=\{E\rightarrow A, A\rightarrow D, A\rightarrow E\}$, where $A\rightarrow D$ is derived from $A\rightarrow B$ and $B\rightarrow D$. $A\rightarrow E$ is derived from $A\rightarrow BC$, $BC\rightarrow CD$ and $CD\rightarrow E$.
We compute F_1^+ and F_2^+. We check if we can derive all the FDs in F from $F_1^+ \cup F_2^+$.
We notice that $B\rightarrow D$ cannot be derived.
Hence the decomposition is not dependency preserving.

(d) [7 points] Suppose we have a relation R(A,B,C,D,E) with the FDs $AB\rightarrow DE$, $C\rightarrow E$, $D\rightarrow C$ and $E\rightarrow A$ and the relation is decomposed into a set of relations. Let S(A,B,C) be one of the decomposed relations. Determine the FDs that hold on S. Describe all concepts that you employ.

Soln. We compute the closure of the set of FDs.
We add $AB\rightarrow C$ by transitivity on $AB\rightarrow D$ and $D\rightarrow C$.
We add $D\rightarrow E$ by transitivity on $D\rightarrow C$ and $C\rightarrow E$.
We add $C\rightarrow A$ by transitivity on $C\rightarrow E$ and $E\rightarrow A$.
So $F^+=\{ AB\rightarrow DE, C\rightarrow E, D\rightarrow C, E\rightarrow A, AB\rightarrow C, D\rightarrow E, C\rightarrow A\}$.

From the closure, we take the FDs that involve only A, B and C.
So we have $AB\rightarrow C$ and $C\rightarrow A$.

Question 3 (Data Integrity) [25 points]

Consider the following database concerning World War II ships:

Classes (class, type, country, numGuns, weight)
Ships (name, class, launched)
Battles (name, date)
Outcomes (ship, battle, result)

Ships are built in "classes" of the same design, and a class is usually named after the first ship of that class. The relation Classes records the name of the class, the type ('bb' for battleship, 'bc' for battlecruiser), the country that built the ship, the number of main guns, and the weight (in tons). Relation Ships records the name of the ship, the name of its class, and the year in which the ship was launched. Relation Battles gives the name and date of battles involving these ships, and relation Outcomes gives the result (sunk, damaged, or ok) for each ship in each battle.

1. Write **assertions** for each of the following conditions.
(a) No class may have more than 5 ships. [6 points]
(b) No ship may be launched before the ship that bears the name of this class. [6 points]

Solution:
```
a) CREATE ASSERTION CHECK
   ( 5 >= ALL
```

```
        (SELECT COUNT(*) FROM Ships GROUP BY class)
   );
```

```
b) CREATE ASSERTION CHECK
   ( NOT EXISTS
     (SELECT s1.name FROM Ships s1
      WHERE s1.launched < (SELECT s2.launched FROM Ships s2
                             WHERE s2.name = s1.class
                          )
        )
    );
```

2. Write the following triggers. In each case, disallow or undo the modification if it does not satisfy the stated constraints.
(a) When a new class is inserted into “Classes”, also insert a ship with the name of that class and a NULL launch data. [6 points]
(b) If a tuple is inserted into “Outcomes”, check that the battle is listed in “Battles”, and if not, insert tuples into one or both of these relations, with NULL components where necessary. [7 points]

Solution:

```
a)
CREATE TRIGGER TriggerA
AFTER INSERT ON Classes
REFERENCING
        NEW ROW AS NewRow
FOR EACH ROW
BEGIN
        INSERT INTO Ships (name, class, lunched)
                VALUES (NewRow.class, NewRow.class, NULL);
END;
```

```
b)
CREATE TRIGGER TriggerB
AFTER INSERT ON Outcomes
REFERENCING
        NEW ROW AS NewRow
FOR EACH ROW
WHEN (NewRow.battle NOT IN (SELECT name FROM Battles))
      INSERT INTO Battles (name, date)
        VALUES (NewRow.battle, NULL);
```

Question 4 (Database Integrity)   [25 points]

Consider the same database schema as in Question 3.
(a) Write a PL/SQL function to return and print out the total number of ships built before 1930. [8 points]

Solution:
```
CREATE OR REPLACE FUNCTION ShipNumber
RETURN number IS
   total number(5) := 0;
BEGIN
   SELECT count(*) into total
   FROM Ship
   Where launch < 1930;

   dbms_output.put_line('Number of Ship before 1930: ' || total);
   RETURN total;
END;
/
```

(b) Write a PL/SQL block to put a class of ship into battlecruiser category if it weighs more than 5000 ton. Then print out the max weight of all ships. [8 points]

Solution:
```
DECLEAR
   max_weight number;
BEGIN
   select max(price) into max_weight from Classes;
   update Classes
   set type := “battlecruiser”
   where weight > 5000;
   commit;
   dbms_output.put_line(“Max weight is “ || max_weight);
EXCEPTION
   when others then dbms_output.put_line(“Update error!”);
   rollback;
END;
/
```

(c) Write a PL/SQL procedure to print out the name and country of the classes that has at least one ship sunk in a battle. Each class should occupy one line in the output. [9 points]

Solution:
```
CREATE OR REPLACE PROCEDURE sunk_ship
IS
   CURSOR c1 IS
   SELECT name, country FROM Classes
```

```
   WHERE class IN ( SELECT class FROM Ships, Outcomes
                    WHERE name = Outcomes.ship AND result = “sunk” );
BEGIN
   FOR emp_rec in c1 LOOP
      dbms_output.put_line(c1.name || c1.country);
   END LOOP;
EXCEPTION
   when others then dbms_output.put_line(“Query error!”);
END;
/
```