McMaster University
CS 3DB3 Fall 2017
Assignment 3
Due: Dec. 6, 2017 at 10:00pm

November 17, 2017

I. Database Design (60 marks)

Question 1 (35 marks)

The Department has a database containing information about all lectures during a term. The `Schedule` relation has the following schema:

`Schedule` (C, D, T, R, P, A)

where C represents a course, D for day (weekday), T for time, R for room, P for professor, and A for head TA. The set of functional dependencies F defined over `Schedule` are:
$F = \{RDT \rightarrow P, RDT \rightarrow C, C \rightarrow A, PDT \rightarrow R, PDT \rightarrow C, CDT \rightarrow P, CDT \rightarrow R\}$

Answer the following questions:

(a) (4 marks) What do these functional dependencies mean? Describe them in words.

(b) (4 marks) Is the FD $RDT \rightarrow AP$ entailed by F? Explain and show your work with references to Armstrong's axioms.

(c) (6 marks) Find all the key(s) of relation `Schedule`. Show your work (i.e., how each key is derived).

(d) (4 marks) Is `Schedule` in BCNF? If not, decompose it into smaller relations that are each in BCNF. Again, show your work.

(e) (6 marks) We can create a new relation `ProfsSchedule`(D, T, P) by projecting some attributes of `Schedule`. Are there new functional dependencies that hold over `ProfsSchedule`? If so, state these FDs. If not, state why not. In both cases, show your work to justify your answer.

(f) (11 marks) Find a minimal cover F_{min} for F. Show all the steps in your derivation of F_{min}.

Question 2 (15 marks)

Consider the relation R(A,B,C,D,E), and the decomposition of R into R1(ABC) and R2(ADE).

(a) (4 marks) Give a set of functional dependencies (FDs) such that the decomposition into R1 and R2 is lossless join and dependency preserving. Justify why your FDs satisfy the criteria.

(b) (4 marks) Give a set of functional dependencies such that the decomposition into R1 and R2 is lossless join, but not dependency preserving. Justify why your FDs satisfy the criteria.

(c) (4 marks) Give a set of functional dependencies such that the decomposition into R1 and R2 is not lossless join, but dependency preserving. Justify why your FDs satisfy the criteria.

(d) (3 marks) Is it true or false that every binary relation is in BCNF? If true, clearly justify and prove it. If false, give an example relation not in BCNF.

Question 3 (10 marks)

Prove the following inference rules for functional dependencies using Armstrong's axioms (using only the axioms presented in class). Show the steps of your proof, and indicate which of Armstrong's axioms is applied in each step.

(a) (5 marks) If $X \to Y$ and $W \to Z$ then $XW \to YZ$.

(b) (5 marks) Given the relational schema $R(A, B, C, D, E, F)$ and the FDs: $C \to D$, $BE \to A$, and $BEF \to C$. Show that BEF is a key.

II. Transactions and Concurrency (20 marks)

Question 4 (8 marks)

a) (3 marks) Give an example of a conflict-serializable schedule that is not a serial schedule.

b) (5 marks) Consider the following locking protocol: Before a transaction T writes a data object A, T has to obtain an exclusive lock on A. For a transaction T, we hold these exclusive locks until the end of the transaction. If a transaction T reads a data object A, no lock on A is obtained. State which of the following properties are ensured by this locking protocol: serializability, conflict-serializability, recoverability, avoids cascading aborts, avoids deadlock. Justify your answer for each property.

Question 5 (12 marks)

In this question, you will execute a few concurrent transactions against a banking table. The `accounts` table is shown in Table 1. Create the `accounts` table in DB2. You may assume `username` is of type char(10), `name` is char(20), and `balance` is of type decimal(10,2). Write the appropriate INSERT INTO statements to load the five records shown into your `accounts` table.

Table 1: `accounts`

username	**name**	**balance**
luke	Skywalker	8000.00
kylo	Ren	12500.00
rey	Rey	5000.00
finn	Finn	1500.00
leia	Skywalker	2000.00

Setup

You will simulate concurrent database queries, using two different database connections, against the `accounts` table. You will open two terminal sessions, and execute the following commands to prepare your environments. We will call the two sessions, A and B.

In Session A

1. Invoke db2 using `db2 +c`. This command turns off the AUTO COMMIT feature in DB2. You will now be at the DB2 command line.
2. Verify AUTO COMMIT is turned OFF by running the command `list command options`. (You should see ”Auto-Commit OFF” in the second entry of the table.)
3. Change the transaction isolation level to ’Read Stability’ by running the command `change isolation to rs`. Transaction isolation levels define the degree of access and interaction among a set of concurrent transactions which operate against the same data. Please see the DB2 reference for more information on isolation levels in DB2.
4. Connect to your database.

In Session B

1. Invoke DB2 by typing `db2` (do not turn AUTO COMMIT off).
2. Run `change isolation to rs`
3. Connect to your database.

Transactions

Run the following commands (in the given order) and provide your answers to the stated questions.

1. In Session A, insert the record `('snoke', 'First Order', 20000.00)`, followed by a `select * from accounts`.
2. In Session B, run `select * from accounts`. Is the output you get the same or different than in (1)? Why did this occur? What is a possible solution?
3. Return to Session A, and implement your solution from step #2.

4. In Session B, do the SELECT * query again (to list all records). Provide your output.
5. We will update accounts from two different transactions. In Session A, change the isolation level to cursor stability (CS). In Session B, change the isolation level to uncommitted read (UR). (You will have to disconnect and reconnect to the database.)
6. In Session A, transfer $5000 from Kylo's account into Rey's account.
7. Then, in Session B, list all accounts and their balances. Then issue a $10000 transfer from Snoke's account to Finn's account. What happens and why? Explain the interaction between the two sessions.
8. Commit the transaction in Session A. Report the latest balances.
9. Change the transaction isolation level in Session A to 'Uncommitted Read'.
10. Now, in Session A, transfer 50% of Kylo's account balance to Leia's account. Commit the transaction.
11. In Session A, transfer all of Snoke's funds to Luke's account. In session B, list all data records. What is Luke's balance? Does it reflect the latest transfer from Snoke? Based on your transaction in step (7) and this step, what can you say about the allowed actions in CS and UR isolation levels? How do these compare with the RS isolation level?
12. Abort the Snoke to Luke transfer transaction in Session A, by executing the command `rollback`.
13. In session A and B, list all data records. What are the final balances?

Grading

This assignment is worth 14% towards your final grade.

Submission

All files are to be submitted using the Avenue system. Please ensure your answers are typed and clearly readable. Include your name and student ID number in the file. Submit your solutions to all questions in a file called **`asg3.pdf`**.