Name: ____________________________________ Std # ________

National University of Computer & Emerging Sciences
(Karachi Campus)

Final Examination – Fall 2016 sol

Course: Databases Systems (CS204) Time Allowed: 180 Min.
Date: December 26, 2016 Max. Points: 100

Note: Attempt all questions. *Start each question on a new page of the answer book; answer all queries of the question in consecutive order. Answer to the point. Return this exam paper along with the answer book.*

Fundamental of Databases (Concept & Architecture)

Question No. 1 [Time:20 min] [Points: 10]

1. Why would you use a database management system instead of simply storing data in files? Justify your answer with technical arguments and the knowledge that you have taken from this course. [5]

   Most of the time it is convenient to stores data in DBMS (preferred RDBMS), if we later want a quick access and update on the data. The data in DBMS offers several advantages, some are as below:

   - It can provide persistent storage with backup and recovery
   - It can control redundancy and multiple users access
   - Authorized access
   - Access through SQL (declarative sense)
   - Enforcing integrity and consistency rules.
   - Rapid application development and modification

2. What is the difference between logical data independence and physical data independence? Which one is harder to achieve? Why? [5]

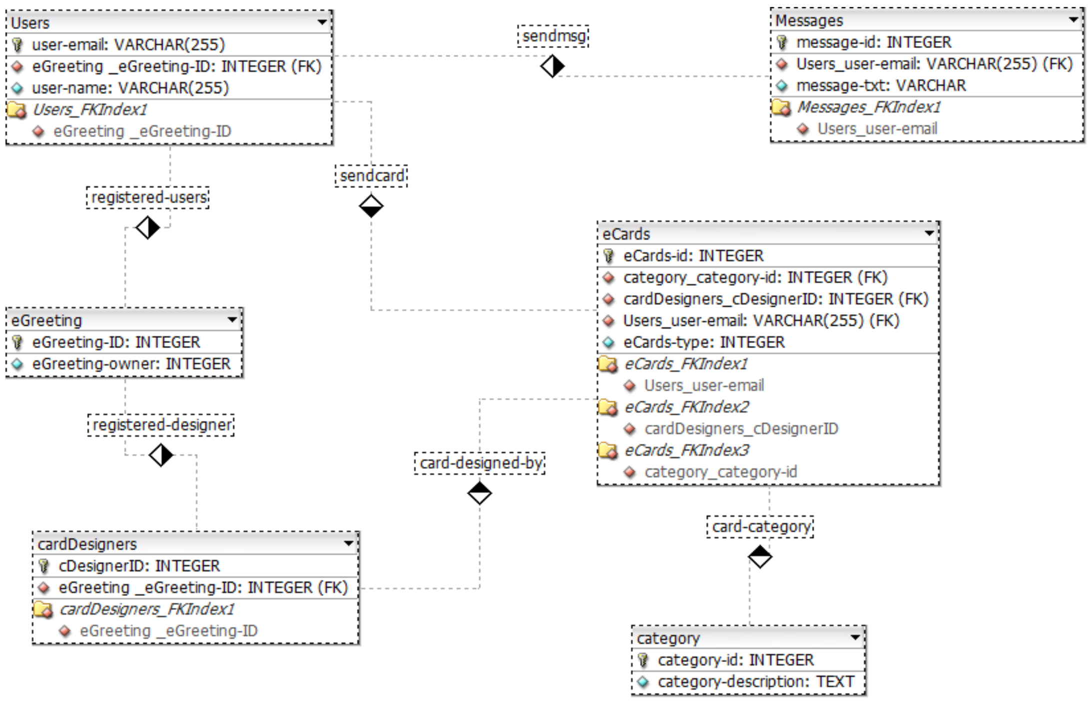
   - Physical data independence
     - The ability to modify the physical scheme without causing application programs to be rewritten
     - Modifications at this level are usually to improve performance
   - Logical data independence
     - The ability to modify the conceptual scheme without causing application programs to be rewritten
     - Usually done when logical structure of database is altered
     - Logical data independence is harder to achieve as the application programs are usually heavily dependent on the logical structure of the data.

Database Design (ER Modeling)

Question No. 2 [Time:40 min] [Points: 20]

eGreetings.com is an online business for sending greeting e-card and messages to online users. In order to send an e-card one has to be a register user at the website. The register users can send any number of e-card and messages. The website keeps track of all the messages sent by a user and respective recipients for each sender. The e-card are classify into different categories like: birthday, friendship, get well, miss you and many others. They can create new categories as and when the need arises. The company purchase e-card from different designers all over the world. A designer (who create e-cards) has to register before the submission of any e-card to eGreetings.com database. The company also tracks designers and their work and duly acknowledges them. Develop an ER model for the above scenario. Identify the entities, attributes and their relationships among the entities.

For each entity set, mark clearly the primary key (If the primary key is not specified by requirement, use your best knowledge to add your own key or use existing attributes). For each relationship you identified, state the cardinalities (1:1, 1:m or m:n) on the entities participating in this relationship. Simply follow the stepwise approach you have being taught.



SQL

Question No. 3 [Time:30 min][Points: 5X4=20]

Consider the following database schema for a Football-League Tournament.

TEAM (TEAM–ID, COUNTRY)
PLAYER (PLAYER–ID, PLAYER–NAME, TEAM–ID, POSITION)
VENUE (VENUE–ID, VENUE–NAME)
MATCH (MATCH–ID, TEAM–ID1, TEAM–ID2, DATE, VENUE–ID)
GOAL (MATCH–ID, PLAYER–ID, TIME, GOAL–TYPE)

Answer the following queries in Standard SQL.

a. List the name of country for which more than two teams are participating in the tournament. [Output: COUNTRY]

```
SELECT T1.COUNTRY
FROM TEAM T1,T2, T3
WHERE T1.COUNTRY=T2.COUNTRY AND T2.COUNTRY=T3.COUNTRY
AND T2.TEAM-ID != T1.TEAM-ID AND T2.TEAM-ID != T3.TEAM-ID AND
T3.TEAM-ID != T1.TEAM-ID

OR

SELECT T.COUNTRY
FROM TEAM T
GROUP BY T.COUNTRY
HAVING COUNT (T.COUNTRY) > 2
```

b. List those players who scored at least one goal in each match which he played. [Output: PLAYER–NAME]

```
SELECT P.PLAYER-NAME
FROM PLAYER P
WHERE
NOT EXIST   (SELECT M.MATCH-ID
             FROM MATCH M, TEAM T, PLAYER P1
             WHERE (M.TEAM-ID= T.TEAM-ID1 OR M.TEAM-ID= T.TEAM-ID2)
          AND (P1.PLAYER-ID= P.PLAYER-ID) AND (T.TEAM-ID = P1.TEAM-
          ID))   // ALL MATCHES PLAYER P PLAYED
WHERE
NOT EXIST (SELECT DISTINCT M1.MATCH-ID
                    FROM MATCH M1, PLAYER P2, GOAL G
                    WHERE M.MATCH-ID= M1.MATCH-ID=G.MATCH-ID
AND (P1.PLAYER-ID= P2.PLAYER-ID=G.PLAYER-ID=P.PLAYER-ID))
// ALL MATCHED PLAYER P HAS SCORED A GOAL
```

c. List the name of all goal keepers of the tournament, who has not let the ball passed them in any match of the tournament (no goal). [Output: PLAYER–NAME, Hint: POSITION="goal keeper".]

```
SELECT P.PLAYER-NAME
FROM PLAYER P, MATCH M1, TEAM T
WHERE M1.TEAM-ID1 = T.TEAM-ID OR M1.TEAM-ID2 = T.TEAM-ID AND
P.TEAM-ID=T.TEAM-ID AND P.POSITION = "goal keeper"
// goal keeper from all teams who played any match
MINUS
SELECT P1.PLAYER-NAME
FROM PLAYER P1, MATCH M, GOAL G, TEAM T
WHERE M.TEAM-ID1 = T.TEAM-ID AND P.TEAM-ID=T.TEAM-ID AND
P1.PLAYER-ID=P.PLAYER-ID AND P1.PLAYER-ID= T.PLAYER-ID AND
G.MATCH-ID=M1.MATCH-ID AND M1.MATCH-ID= M.MATCH-ID
// goal keepers who let the ball passed them in any match they played
```

d. List all matches in which result is x : x (There are goals scored and match is a draw x>0)[Output: MATCH-ID]

```
SELECT DISTINCT MATCH-ID
FROM MATCH
WHERE NOT EXIST
((SELECT COUNT (G.*)
    FROM GOAL G, MATCH M, PLAYER P
         WHERE G.MATCH-ID=M.MATCH-ID
         AND P.PLAYER-ID= G.PLAYER-ID
         AND M.TEAM-ID1= P.TEAM-ID
        GROUP BY (M.TEAM-ID1))
        HAVING COUNT(G.*) > 0
    // number goals in a match m from team1
  MINUS
   (SELECT COUNT (G.*)
    FROM GOAL G, MATCH M, PLAYER P
         WHERE G.MATCH-ID=M.MATCH-ID
         AND P.PLAYER-ID= G.PLAYER-ID
         AND M.TEAM-ID2= P.TEAM-ID
        GROUP BY (M.TEAM-ID2) )
        HAVING COUNT (G.*) > 0
         // goals in a match m from team2

    )
```

Normalization/Relational Schema

Question No. 4 [Time:20min] [Points: 10]

a. Consider an initial relational design given below:

PROJECT = (PROJECT_NUMBER, PROJECT_NAME, START_DATE, PROJECT_STATUS, {EMPLOYEE_NUMBER, EMPLOYEE_NAME, JOB_TITLE, DEPT_NUMBER, DEPT_NAME, PROJECT_HOURS})

You need to perform normalization in stepwise fashion, starting from initial design to un-normalized (UNF) to 1NF then to 2NF and finally into 3NF. Justify your normalization process by definition of each form. [5]

UNF:
PROJECT = (PROJECT_NUMBER, PROJECT_NAME, START_DATE, PROJECT_STATUS, EMPLOYEE_NUMBER, EMPLOYEE_NAME, JOB_TITLE, DEPT_NUMBER, DEPT_NAME, PROJECT_HOURS)

INF: Every attribute is atomic, no repeating groups, all relations have a PK.

PROJECT = (***PROJECT_NUMBER***, PROJECT_NAME, START_DATE, PROJECT_STATUS)
PROJECT-EMPLOYEE=(***EMPLOYEE_NUMBER***,EMPLOYEE_NAME, JOB_TITLE, DEPT_NUMBER, DEPT_NAME, PROJECT_HOURS)

2NF: Every Relation in 1NF and PK can determine all attributes, PK can be composite.

PROJECT = (***PROJECT_NUMBER***, PROJECT_NAME, START_DATE, PROJECT_STATUS)
PROJECT-EMP=(*EMPLOYEE_NUMBER*,*PROJECT_NUMBER* ,PROJECT_HOURS)
EMPLOYEE=(***EMPLOYEE_NUMBER***,EMPLOYEE_NAME,JOB_TITLE, DEPT_NUMBER, DEPT_NAME)

3NF: Every Relation in 2NF and No NON-key attribute can determine any attribute.

PROJECT = (***PROJECT_NUMBER***, PROJECT_NAME, START_DATE, PROJECT_STATUS)
PROJECT-EMP=(*EMPLOYEE_NUMBER*,*PROJECT_NUMBER* ,PROJECT_HOURS)
EMPLOYEE=(***EMPLOYEE_NUMBER***,EMPLOYEE_NAME,JOB_TITLE, DEPT_NUMBER)
DEPT=(***DEPT_NUMBER***, DEPT_NAME)

b. Consider a relation R(A,B,C,D,E) along with a set of FDs for R as { A -> BC, CD->E, B->D, and E->A}, answer the following queries

   1. Find a candidate key for R. [1]

      Candidate Key = A

   2. Is R in 3NF? Justify your answer. [2]

      No R is not in 3NF. If A is the key than B->D violates rules for 3NF

   3. Is the decomposition R1 (A,B,C) and R2(A,D,E) of R lossless or lossy? Justify your answer. State whether this decomposition is dependency preserving? If it is not, which dependency is not preserved? [2]

      R1 ∩R2 = A which is a key hence it will be a lossless decomposition.
      R1 has FDs={ A->BC} and R2 has FDs={E->A}, The FD B->D cannot be infer from these two FDs sets. Hence it is not dependency preserving.

Transaction Processing

Question No. 5 [Time:20min][15 Points]

a. Define Transaction. What are the desirable properties of transaction? [5]

   A transaction is the smallest unit of work that can be performed against a database system. There are four desirable properties of a transaction processing systems (DBMS) these are: 1. Atomicity 2. Consistency 3. Isolation and 4. Durability

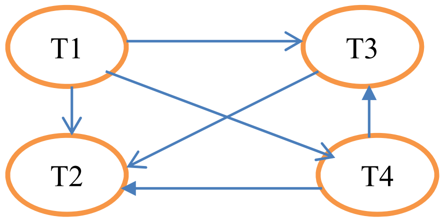
b. What is the difference between conflict equivalence and view equivalence? [5]

   Consider two schedules S1 and S2, they are said to be view equivalent if following conditions are true:
   Initial read must be same.
   There are two transactions say Ti and Tj, The schedule S1 and S2 are view equivalent if in schedule S1, Ti reads A that has been updated by Tj, and in schedule S2, Ti must read A from Tj. i.e. write-read(WR) sequence must be same between S1 and S2.
   Final write operations should be same between S1 and S2.

   Conflict equivalence- Two schedules are conflict equivalent if the order of any two conflicting operations is same in both the two schedules.

   All conflict serializable schedules are view serializable. But all view serializable schedules are not conflict serializable.

c. Consider the following schedule of transactions:
S: r1(x), r4(x), w4(x), r1(y), r4(z), w4(z), w3(y), w3(z), w1(t), w2(z), w2(t)
Check whether the schedule is conflict serializable or not? Also give all possible order of execution of transactions, if possible. [5]



Hence the precedence graph has no cycle, the serial equivalent schedule will be T1,T4, T3 and T2.

Concurrency Control

Question No. 6 [Time:30 min] [15 Points]

a. Why Concurrency Control protocols are necessary for multi-users, high-transaction environment of an RDBMS? How locking protocol ensures serializable schedule? How time-stamp ordering ensures serializable schedule. [5]

The concurrency control protocol is a necessary module for supporting multiple users, concurrent access to shared data from a database. It ensures that the data is being accessed by concurrent users without the critical problems of concurrent access like: dirty read, lost update and incorrect summary. There are two broad types of protocols (i) Locking based and (ii) Time Stamp ordering based.

In Locking based protocol the conflicting access that can cause a problem is controlled through locking, the lock grant access to one at a time and hence enforced serial order in execution.

In Time Stamped based protocol, the access to shared resource is granted by monitoring a time-stamp order associated to each resource. Hence a time-order based permission to access ensures a serial execution of transactions.

b. Consider the following schedule of transactions

```
T1: r1(X);w1(X);r1(Y);w1(Y);C1 ;
T2: r2(X);w2(X);C2 ;
T3: r3(Y),w3(Y),C3
```

Assuming a time stamp based protocol control the concurrent execution of given schedule. Let's transaction ID like T1 represent a time stamp of 1 and hence T1 is very first transaction of the system. Now, consider the following non-serial schedule

Sa - r1(X); w1(X); r3(Y); w3(Y); r1(Y); w1(Y); r2(X); w2(X); C1; C3; C2;

if this schedule is run by the time-stamp based protocol and T1,T3 and T2 is the serial equivalent schedule. Provide a time-stamp based ordering of fundamental operations and also illustrate the number of restart of transactions, if any. [5]

T1	**T2**	**T3**	**X**	**Y**
			RTS=0 WTS=0	RTS=0 WTS=0
r1(X)				
w1(X)			RTS=1 WTS=1	
		r3(Y)		
		w3(Y)		RTS=3 WTS=3
r1(Y)				
Based on TS Protocol either T1 or T3 will restart , let the younger transaction get started, hence T3 is started with new time stamp say T4.				
w1(Y)				RTS=1 WTS=1
	r2(X)		RTS= 2 as it was 1 earlier.	
	w2(X)		RTS=2 WTS=2	
C1				
		r4(Y)		
		w4(Y)		RTS=4 WTS=4
		C3		
	C2			

c. Consider the following group of transactions:

T_1: r1(X),w1(X),r1(Y),w1(Y)
T_2: r2(Z),r2(Y),w2(Y),r2(X),w2(X)
T_3: r3(Y),r3(Z),w3(Y),w3(Z)

Assuming a locking based protocol is used for concurrency control, let's give a non-serial schedule with locking preference in which commit order of transactions should be equivalent to serial schedule T_1 , T_2 and T_3. [5]

T1	**T2**	**T3**
RL(X)	RL(Z)	
r1(X)	WL(Y)	
w1(X)	r2(Y)	
UL(X)	w2(Y)	
WL(Y)	UL (Y)	
r1(Y)	UL(Z)	
w1(Y)	WR(X)	
UL(Y)	r2(X)	WL(Y)
C1	w2(X)	r3(Y)
	UL(X)	WL(Z)
	C2	r3(Z)
		w3(Y)
		w3(Z)
		UL(Y)
		UL(Z)
		C3

Recovery

Question No. 7 [Time:20][2.5x4=10 Points]

Consider the following transaction log, generated by RDBMS at some permanent storage.

LSN	TxID	Operation	Page#	DB-item	BFIM	AFIM
1000	-	**CKPT 121**	-	-	-	-
1001	T1	Read	103	X	-	-
1002	T1	Read	105	Y	-	-
1003	T1	write	103	X	20	30
1004	T2	Read	103	X	-	-
1005	T1	write	105	Y	40	80
1006	T1	Commit	105	Y	40	80
1007	T3	Read	103	X	-	-
1008	T2	Read	107	Z	-	-
1009	T2	write	103	X	30	50
1010	T2	write	107	Z	60	80
1011	T3	write	103	X	50	70
1012	T4	Read	107	Z	-	-
1013	T5	Read	100	A	-	-
1014	-	**CKPT 122**	-	-	-	-
1015	T3	Commit	103	X	50	70
1016	T5	Write	100	A	1020	100
1017	T2	Commit	107	Z	60	80

a. Assume that system crash occurred after the log sequence number 1013. The database is using immediate update, and the transactional log is also maintained on a separate disk. Give the details of active transactions, dirty pages, redo and undo log-action details.

LSN=1013; immediate update
Active Transactions: T2,T3,T4,T5
Dirty Pages: 103,107
Undo: T2 and T3
Redo: nil

b. Assume that system crash occurred after the log sequence number 1013. The database is using deferred update with policy that it will update each commit transaction as soon as it is committed. The transactional log is also maintained on a separate disk. Give the details of active transactions, dirty pages, redo and undo log-action details.

LSN=1013; deferred update (C=1)
Active Transactions: T2,T3,T4,T5
Dirty Pages: 103,107
Undo: nil
Redo: T2,T3,T4 and T5

c. Assume that system crash occurred after the log sequence number 1015. The database is using deferred update with policy that it will update each commit transaction as soon as it is committed. The transactional log is also maintained on a separate disk. Illustrate how recovery is implemented in this case.

   LSN=1015; deferred update
   Active Transactions: T2, T4,T5
   Dirty Pages: 107
   Redo: T2,T3,T4
   Undo: nil

d. Assume that system crash occurred after the log sequence number 1017. The database is using immediate update. The transactional log is also maintained on a separate disk. Illustrate how recovery is implemented in this case.

   LSN=1017; immediate update
   Active Transactions: T4, T5
   Dirty Pages: 100
   Undo: T5
   Redo: nil

The End & Good Luck ☺