National University of Computer & Emerging Sciences (NUCES – FAST)
Department of Computer Science
CS203 – Database Systems
Final Examination – Fall 2015

Date: December 17, 2015 **Time Allowed: 3 Hours**
Max. Points: 100

Instructions: (Read carefully before the exam begins)
1. Answer ALL question and their parts in consecutive order.
2. Start each question on the new page of the answer book.
3. Questions will not be interpreted. Instructors will only confirm or deny errors in the questions.
4. You may not trivialize the problem in your assumptions.
5. Return this paper along with the answer book.

Fundamentals of Database Systems

Question No.1 **[10 Points]** **[Time: 15 Minutes]**

a) A driver's license bureau maintains a database of licensed drivers. State whether each of the following represents data or metadata. If it represents data, state whether it is structured or unstructured data.

S.No.	Description	Data/Meta-Data	Structured/Unstructured
1.	Driver's name, address, and birthdate.	**Data**	**Structured**
2.	The fact that the driver's name is a 30-character field.	**Meta-Data**	**-**
3.	A photo image of the driver.	**Data**	**Unstructured**
4.	An image of the driver's fingerprint.	**Data**	**Unstructured**
5.	The make and serial number of the scanning device that was used to scan the fingerprint.	**Data**	**Structured**
6.	The resolution (in megapixels) of the camera that was used to photograph the driver.	**Meta-Data**	**-**
7.	The fact that the driver's birthdate must precede today's date by at least 1 6 years.	**Meta-Data**	**-**

b) Mention two situations where it is not desirable to use DBMS.

1) **If the database and applications are simple, well defined, and not expected to change.**
2) **If there are stringent real-time requirements that may not be met because of DBMS overhead.**
3) **If access to data by multiple users is not required.**
4) **If the database system is not able to handle the complexity of data because of modeling limitations.**
5) **If the database users need special operations not supported by the DBMS.**

ER-Modeling

Question No.2	**[20 Points]**	**[Time: 30 Minutes]**

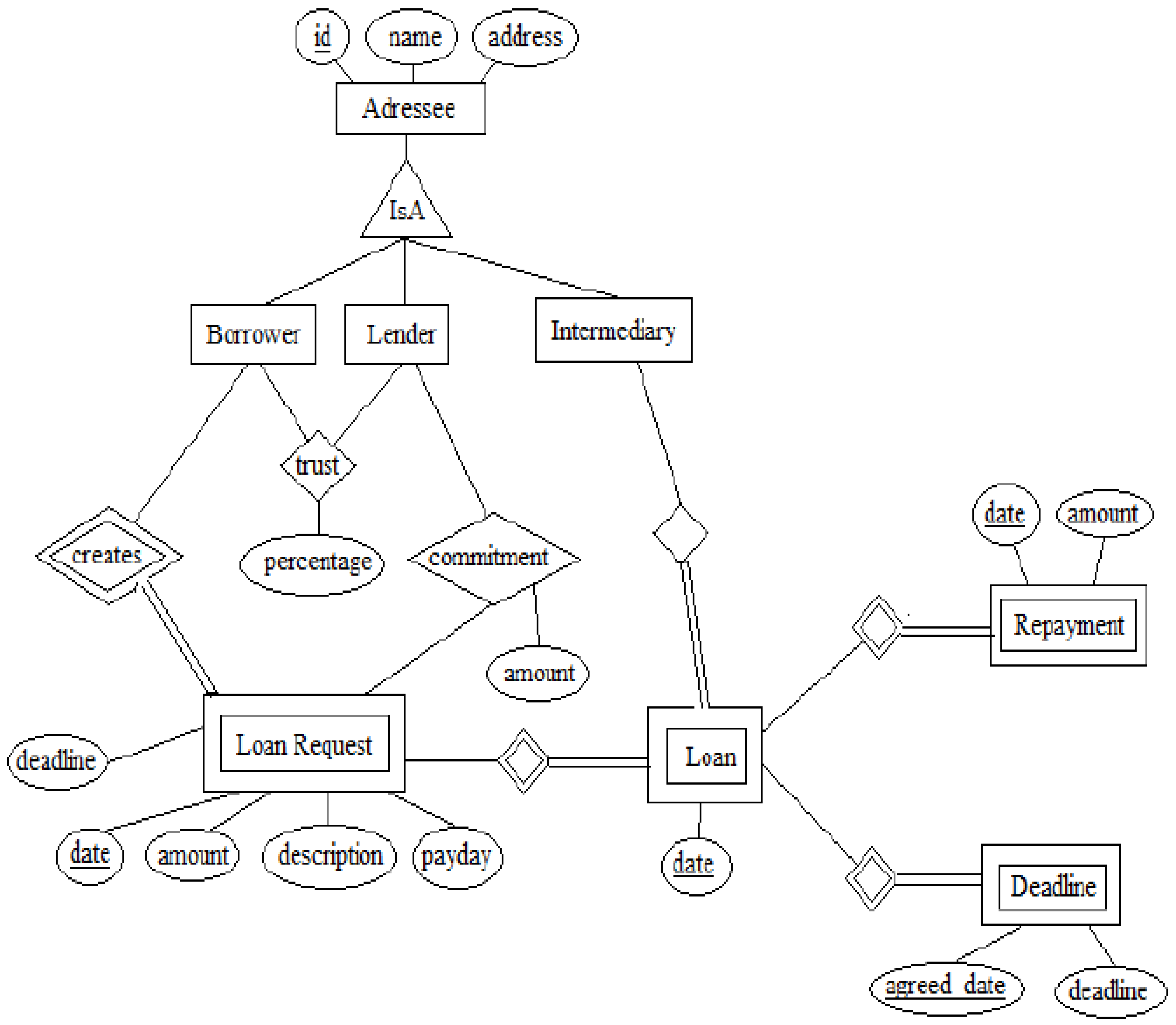
Micro loans are small loans, which is beginning to gain popularity especially among borrowers in developing countries like Pakistan. To practically implement this idea, Tameer Micro Finance Bank Limited launched its operations in 2005 as the first scheduled Micro Finance Bank in Pakistan to provide micro-finance and related financial services to the less privileged and unbanked segment of the society.

The idea is to bring venture lenders together using information technology. Typically, the loans will be used to finance startup or development of the borrower's company, so that there is a realistic chance for repayment. The money in a loan can, unlike traditional loans, come from many lenders. In this problem, you are asked to create an ER model that describes the information necessary to manage micro loans at Tameer Bank.

The following information forms the basis for creating the model:

- Each borrower and lender must be registered with information about name and address.
- A loan starts with a loan request, which contains information about when the loan should at latest be granted, the total amount being discussed (Pak-Rupees), and how long the payback period is. Also, a description is included of how the money will be used. The rent on the payment is calculated in the loan amount, which is to say, the full amount is not paid.
- Lenders can commit to an optional portion of the total amount of a loan request.
- When the commitments for the loan request covers the requested amount, the request is converted to a loan. If not enough commitments can be reached, the loan request is cancelled. A borrower can have more than one request and more than one loan at a time, but can at most make one request per day.
- The loan is paid through an "intermediary", typically a local department of a charity, who has a name and an address.
- The borrower chooses when he or she will make a payment. Every payment must be registered in the database with an amount and a date (at most one payment per loan per day). The lenders share the repayment based on how large a part of the loan they are responsible for.
- If the loan is not repaid before the agreed upon deadline, a new date is agreed. The database must not delete the old deadline, but save the history (the deadline can be overridden multiple times).
- Each lender can for each borrower save a "trust", which is a number between 0 and 100 that determines the lender's evaluation of the risk of lending money to that person. The number must only be saved for the borrowers, for whom there has been made such an evaluation.

Make an ER model for the data described above. If you make any assumptions about data that doesn't show from the problem, they must be described. Use the Peter Chen notation. Put an emphasis on having the model express as many properties about the data as possible, for instance participation constraints.


id
name
address
Adressee
IsA
Borrower
Lender
Intermediary
trust
percentage
creates
commitment
amount
Loan Request
deadline
date
amount
description
payday
Loan
date
Repayment
date
amount
Deadline
agreed date
deadline

ER-to- Relational Mapping		
Question No.3	**[15 Points]**	**[Time: 20 Minutes]**

Make a relational data model for Tameer Micro Finance Bank Limited discussed in Question No.2 by describing the relations using SQL DDL statements. Make reasonable assumptions about data types.
The emphasis is if there is a correlation between the relational model and the ER diagram from Question No.2, along with primary key and foreign key constraints being stated for all relation.

[Note: It is assumed, that borrowers, lenders, and intermediaries are disjoint entities. Below MySQLs ENUM type is used, which is not part of the syllabus.]

```
CREATE TABLE Adressee (
id INT PRIMARY KEY,
type ENUM('borrower', 'lender', 'intermediary'),
name VARCHAR(50),
address VARCHAR(50)
);
CREATE TABLE Trust (
borrower INT REFERENCES Adressee(id),
lender INT REFERENCES Adressee(id),
percentage INT,
PRIMARY KEY (borrower,lender)
);
CREATE TABLE LoanRequest (
id INT REFERENCES Adressee(id),
date DATE,
amount INT,
description VARCHAR(1000),
payday DATE,
deadline DATE,
PRIMARY KEY (id,date)
);
CREATE TABLE Commitment (
lender INT REFERENCES Adressee(id),
borrower INT,
loanrequestDate DATE,
FOREIGN KEY (borrower,loanrequestDate) REFERENCES LoanRequest(id,dato),
amount INT,
PRIMARY KEY (lender, borrower, loanrequestDate,amount)
);
CREATE TABLE Loan (
id INT,
RequestDate DATE,
date DATE,
intermediary REFERENCES Adressee(id),
FOREIGN KEY (id,RequestDate) REFERENCES LoanRequest(id,date),
PRIMARY KEY(date,id,RequestDate)
);
CREATE TABLE Repayment (
id INT,
date DATE,
RequestDate DATE,
amount INT,
FOREIGN KEY (id,RequestDate) REFERENCES LoanRequest(id,date),
PRIMARY KEY (date,id,RequestDate)
);
CREATE TABLE Deadline (
id INT,
agreedDate DATE,
RequestDate DATE,
deadline DATE,
FOREIGN KEY (id,RequestDate) REFERENCES LoanRequest(id,date),
PRIMARY KEY (agreedDate,id,RequestDate)
);
```

Normalization

Question No.4	**[15 Points]**	**[Time: 20 Minutes]**

The following relation schema can be used to register information on the repayments on micro loans (see the text in Question No.2 for the explanation on micro loans).

`Repayment(borrower_id,name,address,loanamount,requestdate,repayment_date,repayment_amount)`

A borrower is identified with an unique **`borrower_id`**, and has only one **`address`**. Borrowers can have multiple simultaneous loans, but they always have different request dates. The borrower can make multiple repayments on the same day, but not more than one repayment per loan per day.

a) State candidate keys for **`Repayment`**.

{borrower_id, requestdate, repayment_date}

b) Make the normalization to 3NF. State for every step in the normalization, which functional dependency that causes it.

Functional Dependencies:
borrower_id → name, address
borrower_id, requestdate → loanamount
3NF:
Repayment1(borrower_id, name, address)
Repayment2(borrower_id, requestdate, loanamount)
Repayment3(borrower_id, requestdate, repayment_date, repayment_amount)

SQL

Question No.5	[2.5X4=10 Points]	[Time: 20 Minutes]

This problem is about writing SQL for the relation `Repayment` from Question No.4:
`Repayment(borrower_id,name,address,loanamount,requestdate,repayment_date,repayment_amount)`
To solve the problem, the information from the description in Question No.4 must be used.

a) Write an SQL request that returns all the tuples with information on repayments from the borrower with id equal to 42, and where the lent amount exceeds 1000 PKR.

```
SELECT *
FROM Repayment
WHERE borrower_id=42 AND loanamount>1000;
```

b) Write an SQL request that for each address finds the total repaid amount for the address.

```
SELECT address, SUM(repayment_amount)
FROM Repayment
GROUP BY address;
```

c) Write an SQL request that finds all names which has a unique address, which to say is where there does not exist a tuple with a different name and same address.

```
SELECT name
FROM Repayment A
WHERE 1=
            (SELECT COUNT(DISTINCT name)
             FROM Repayment B
             WHERE A.address=B.address);
```

d) Write an SQL command, which deletes all information on ended loans, which is to say loans where the total repaid amount equals the lend amount.

```
DELETE FROM Repayment A
WHERE loanamount=
                  (SELECT SUM(repayment_amount)
                  FROM Repayment B
                  WHERE B.borrower_id=A.borrower_id AND B.requestdate=A.requestdate);
```

Transaction Processing Concepts

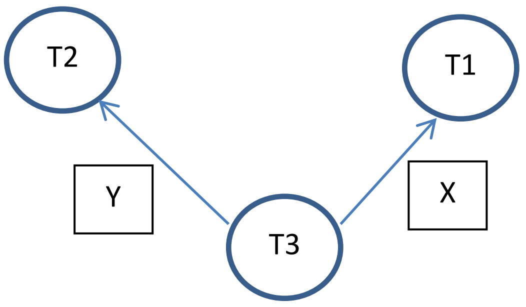
Question No.6	[15 Points]	[Time: 30 Minutes]

a) Consider the three transactions T1, T2, and T3, and the schedules S1 and S2 given below: [5]

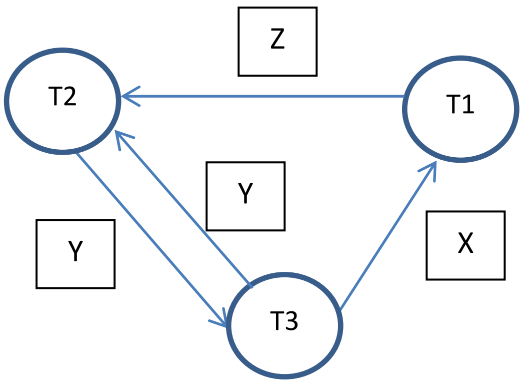
T1: r1 (X); r1 (Z); w1 (X);
T2: r2 (Z); r2 (Y); w2 (Z); w2 (Y);
T3: r3 (X); r3 (Y); w3 (Y);
S1: r1 (X); r2 (Z); r1 (Z); r3 (X); r3 (Y); w1 (X); w3 (Y); r2 (Y); w2 (Z); w2 (Y);
S2: r1 (X); r2 (Z); r3 (X); r1 (Z); r2 (Y); r3 (Y); w1 (X); w2 (Z); w3 (Y); w2 (Y);

Draw the serializability (precedence) graphs for *S*1 and *S*2, and state whether each schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s).

In S1, there is no cycle. Hence, S1 is conflict serializable.
The serial order for S1 is: T3 → T1 → T2 <u>or</u> T3 → T2 → T1



In S2, there is a cycle. Hence, S2 is NOT conflict serializable.



b) Consider the following two transactions and some of their schedules:

T1: r1(A),r1(B),w1(A),w1(B)
T2: r2(C),r2(B),w2(B),w2(C)
S1: r1(A),r2(C),r1(B),r2(B),w1(A),w2(B),w1(B),w2(C),c1,c2
S2: r1(A),r1(B),w1(A),r2(C),r2(B),w2(B),w1(B),w2(C),c1,c2
S3:r1(A),r1(B),w1(A),w1(B),c1,r2(C),r2(B),w2(B),w2(C),c2
S4: r2(C),r2(B),w2(B),r1(A),r1(B),w1(A),w2(B),w2(C),c2,c1
S5: r1(A),r1(B), r2(C), w1(A),w1(B), r2(B),c1,w2(B),w2(C),c2
S6: r1(A),r1(B), r2(C), w1(A),w1(B), r2(B),w2(B),w2(C),c2,c1

Answer the following questions using transactions schedules given above:

i) Which schedule(s) is/are strict, serial schedule? [1]
S3

ii) Is Schedule S3 recoverable? [1]
Yes

iii) Is schedule S4 cascadeless? [1]
Yes

iv) Which pair of schedules are conflict equivalent? Just give one such schedule.[2]
S1 and S2

v) Is schedule S5 a recoverable schedule? [1]
Yes

vi) Are schedule S1 and S5 result equivalent? [1]
Yes

vii) Is schedule S6 cascadeless? [1]
No

viii) Which pair of schedule are view equivalent? Just give one such schedule.[2]
S5 and S6

Concurrency Control Techniques

Question No.7	[5 Points]	[Time: 15 Minutes]

Consider the following schedule of three transactions:

T1	T2	T3
R1(X)		
W1(X)		
R1(Y)		
W1(Y)		
	R2(X)	
	W2(X)	
		R3(P)
		R3(Y)
		W3(Y)
		W3(P)

Use basic two-phase locking (2PL) to show that the schedule can run concurrent transactions and serializable. Assume each operation of transaction takes unit time, provide locking/unlocking of resources for the three transactions.

T1	T2	T3
Lock(X)		
Lock(Y)		
R1(X)		
W1(X)		
R1(Y)		
W1(Y)		
Unlock(Y)		
Unlock(X)		
	Lock(X)	
	R2(X)	
	W2(X)	
	Unlock(X)	
		Lock(P)
		Lock(Y)
		R3(P)
		R3(Y)
		W3(Y)
		W3(P)
		Unlock(Y)
		Unlock(P)

Recovery Techniques

Question No.8	[2.5X4=10 Points]	[Time: 20 Minutes]

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.

Active Transactions: T2, T3, T4, T5 **Dirty Pages:** 103, 107	**Redo:** NIL **Undo:** T2, T3, T4, T5

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.

Active Transactions: T2, T3, T4, T5 **Dirty Pages:** 103, 107	**Redo:** T2, T3, T4, T5 **Undo:** NIL

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.

In this case, CKPT 122 shall be utilized.
Hence only T2, T4, and T5 shall be restarted.

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.

In this case T4, T5 will be undone.
The recovery shall only go till CKPT 122 and shall use check pointing data to undone T4 and T5.

Good Luck ☺