CS3563: DBMS II : Final Exam
(Closed Book, 2 hrs, 100 marks)

Name: **ID:**

Marks Obtained

Question	Marks	Question	Marks
1		6	
2		7	
3			
4			
5			
Total			

Instructions:

- You have to *return* the question paper along with the answers written in the empty spaces provided.
  Please use the extra sheets provided to do the rough work and *not* on the question paper.
- Make sure that the answers for the subjective type and the reasons for the objective type are *clearly* written.
  Note that we value *precision* and *conciseness* in the answers.
  If you need to make any additional assumptions, write them clearly.
- The marks for each question and each sub-question are given in square brackets in **bold** font.

1. **[2 + 2 + 2 + 2 + 2 = 10 marks]**

   Consider the following relational schema for a library:

```
member(memb_no, name, dob)
books(isbn, title, authors, publisher)
borrowed(memb_no, isbn, date)
```

   Write the following queries in **relational algebra** (Not SQL!).

   1.1. Find the names of members who have borrowed any book published by "McGraw-Hill".

   1.2. Find the name of members who have borrowed all books published by "McGraw-Hill".

   1.3. Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".

   1.4. For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.

   1.5. Find the average number of books borrowed per member. Take into account that if a member does not borrow any books, then that member does not appear in the borrowed relation at all.

2. **[10 marks]**

The following relations keep track of airline flight information:

```
Flights(flno: integer, from: string, to: string, distance: integer,
departs: time, arrives: time, price: real)
Aircraft(aid: integer, aname: string, cruisingrange: integer)
Certified(eid: integer, aid: integer)
Employees(eid: integer, ename: string, salary: integer)
```

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly.

Consider the SQL below:

```
SELECT F.departs
FROM   Flights F
WHERE  F.flno IN ( ( SELECT F0.flno
FROM   Flights F0
WHERE  F0.from = ’Madison’  AND F0.to = ’New York’
AND    F0.arrives < ’18:00’ )
UNION
( SELECT F0.flno
FROM     Flights F0, Flights F1
WHERE    F0.from = ’Madison’ AND F0.to <> ’New York’
AND      F0.to = F1.from AND F1.to = ’New York’
AND      F1.departs > F0.arrives
AND      F1.arrives < ’18:00’ )
UNION
( SELECT F0.flno
FROM     Flights F0, Flights F1, Flights F2
WHERE    F0.from = ’Madison’
AND      F0.to = F1.from    AND F1.to = F2.from
AND      F2.to = ’New York’ AND F0.to <> ’New York’
AND      F1.to <> ’New York’
AND      F1.departs > F0.arrives
AND      F2.departs > F1.arrives
AND      F2.arrives < ’18:00’ ))
```

Describe in plain English (1-3 sentences) what the above SQL statement does. Be precise and do not give long answers.

3. **[4 + 2 + 2 + 2 = 10 marks]**

3.1. Consider a relation R with five attributes ABCDE. You are given the following dependencies: $A \rightarrow B$, $BC \rightarrow E$, and $ED \rightarrow A$.

1. List all super-keys for R.

2. Is R in 3NF? (Provide a clear and concise explanation)

3.2. Consider the relation shown below:

X	**Y**	**Z**
x_1	y_1	z_1
x_1	y_1	z_2
x_2	y_1	z_1
x_2	y_1	z_3

1. List all the functional dependencies that this relation instance satisfies.

2. Assume that the value of attribute Z of the last record in the relation is changed from z_3 to z_2. Now list all the functional dependencies that this relation instance satisfies.

4. **[10 marks]**

Consider the following relations for a company database:

- **emp** (ename, dname, salary)
- **mgr** (ename, mname)

and the Java code given below, which uses the JDBC API.

```
import java.sql.*;
public class Mystery {
public static void main(String[] args) {
try {
Connection con=null;
Class.forName("oracle.jdbc.driver.OracleDriver");
con=DriverManager.getConnection(
"jdbc:oracle:thin:star/X@//edgar.cse.lehigh.edu:1521/XE");
Statement s=con.createStatement();

String q;
String empName = "dog";
boolean more;
ResultSet result;
do {
q = "select mname from mgr where ename = ?" + empName + "?";
result = s.executeQuery(q);
more = result.next();
if (more) {
empName = result.getString("mname");
System.out.println (empName);
}
} while (more);
s.close();
con.close();
} catch(Exception e){e.printStackTrace();}
}
}
```

Assume that the userid, password, machine name, etc. are all okay.

Describe in concise English what the Java program does. (That is, produce an English sentence like "It finds the ...," not a line-by-line description of what each Java statement does.)

5. **[4 + 4 + 4 + 4 + 4 = 20 marks]**

Consider a disk with a sector size of 512 bytes, 2000 tracks per surface, 50 sectors per track, five double-sided platters, and average seek time of 10 msec.

5.1. What is the capacity of a track in bytes? What is the capacity of each surface? What is the capacity of the disk?

5.2. How many cylinders does the disk have?

5.3. Give examples of valid block sizes. Is 256 bytes a valid block size? 2048? 51200?

5.4. If the disk platters rotate at 5400 rpm (revolutions per minute), what is the maximum rotational delay?

5.5. If one track of data can be transferred per revolution, what is the transfer rate?

6. **[10 + 10 = 20 marks]**

Assume that you have just built a dense B^+ tree index on a heap file containing 20,000 records. The key field for this B^+ tree index is a 40-byte string, and it is a candidate key. Pointers (i.e., record ids and page ids) are (at most) 10-byte values. The size of one disk page is 1000 bytes. The index was built in a bottom-up fashion using the bulk-loading algorithm, and the nodes at each level were filled up as much as possible.

6.1. How many levels does the resulting tree have?

6.2. For each level of the tree, how many nodes are at that level?

7. **[4 + 8 + 8 = 20 marks]**

Use Extendible hashing with the initial hash function $h(x) = x$ mod 4 and a bucket size of 3 to create an index on a relation with the following search keys: $2, 3, 5, 7, 11, 17, 19, 23, 29,$ and 31.

7.1. Show the final directory and data pages after inserting the given keys given to you.

7.2. Show the result of inserting 1 into the hash index that you derived in part 7.1.

7.3. Show the result of inserting 15 into the hash index that you derived in part 7.1.

.

Good Luck!