MCS-274 Final Exam

Serial #:

This exam is closed-book and mostly closed-notes. You may, however, use a single 8 1/2 by 11 sheet of paper with *hand-written* notes for reference. (Both sides of the sheet are OK.)

Please write your name only on this page. Be sure to look at all problems before deciding which one to do first. Some problems are easier than others, so plan your time accordingly. You have 120 minutes to work.

Write the answer to each problem on the page on which that problem appears. You may also attach additional paper, which should be labeled with your test number and the problem number.

You must sign the honor pledge below and abide by it.

Printed name: ____

On my honor, I pledge that I have not given, received, nor tolerated others' use of unauthorized aid in completing this work.

Signature for above honor pledge: ____

Problem	Page	Possible	Score
1	2	10	
2	3	10	
3	4	10	
4	5	10	
5	6	10	
6	7	10	
7	8	10	
8	10	10	
9	12	10	
10	14	10	
Total		100	

1. [**10 Points**] The following tables are used to describe a simplified bank's customers and their transactions:

```
create table customers (
  name varchar(30) not null,
  id int primary key
  );

create table transactions (
  customer int not null references customers(id),
  amount int not null,
  id int primary key
  );
```

Write each of the following SQL queries. The word "deposit" means a transaction with an amount that is greater than zero. None of your queries should produce duplicate rows.

(a) List the following information for each transaction: the transaction ID number, the customer ID number, the customer name, and the amount.

(b) List the following information for each customer: the customer ID number, the customer name, and the number of transactions the customer has on record. This last piece of information should be in a column labeled NUM and should be present even if it is zero.

(c) List the ID number and name of each customer who has made at least one deposit.

(d) List the ID number and name of each customer who has made no deposits.

(e) List the ID number and total of deposit amounts for each customer whose total deposit amounts is greater than 1000. The list should be arranged so that the ID numbers are increasing. (Remember: this problem only refers to the amounts for deposit transactions.)

2. [**10 Points**] Consider the following relation:

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_1
a_2	b_1	c_1	d_2

(a) Cross out those FDs that don't hold in the relation:

$A \to B$
$A \to C$
$A \to D$
$B \to A$
$B \to C$
$B \to D$
$C \to A$
$C \to B$
$C \to D$
$D \to A$
$D \to B$
$D \to C$
$A\,B \to C$
$A\,B \to D$
$A\,C \to B$
$A\,C \to D$
$A\,D \to B$
$A\,D \to C$
$B\,C \to A$
$B\,C \to D$
$B\,D \to A$
$B\,D \to C$
$C\,D \to A$
$C\,D \to B$
$A\,B\,C \to D$
$A\,B\,D \to C$
$A\,C\,D \to B$
$B\,C\,D \to A$

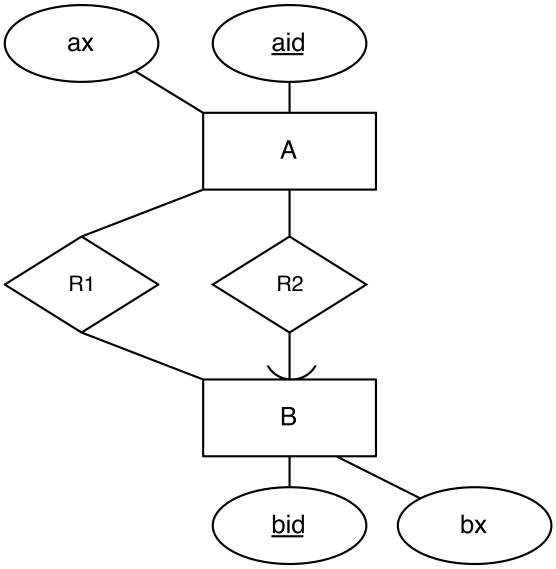
(b) Put a + sign next to some of the non-crossed-out FDs, such that those you have marked can serve as a minimal basis for all those you didn't cross off.

(c) List all the keys for the relation.

(d) Put a − sign next to each non-crossed-out FD that violates BCNF.

(e) Put a × sign next to each non-crossed-out FD that violates 3NF.

3. [**10 Points**] Consider the following E/R diagram:



(a) Translate this design into an appropriate collection of CREATE TABLE statements, including all relevant constraints. You may assume that all attributes are of type INT.

(b) Make the necessary modifications to the diagram to indicate that A is a weak entity set.

(c) State how you would change your CREATE TABLE statements to reflect A being a weak entity set.

4. [**10 Points**] Next to each of the CREATE VIEW statements, indicate whether the view is updatable, and if not, why not.

```
create table A (
  a1 int,
  a2 int
  );

create table B (
  b1 int not null,
  b2 int not null
  );

create view V1 as
select a1, a2
from A
where a1 > 10;

create view V2 as
select a1
from A
where a1 > 10;

create view V3 as
select b1, b2
from B
where b1 > 10;

create view V4 as
select b1
from B
where b1 > 10;

create view V5 as
select a1, a2, b1, b2
from A, B
where a1 > 10
and a2 = b1;
```

5. [**10 Points**] The following problems concern programming in PL/SQL:

(a) PL/SQL distinguishes functions from procedures. A function returns a result value, whereas a procedure does not. Nonetheless, when some PL/SQL code calls a procedure, it frequently wants to receive some information back from the called procedure. What language mechanism is typically used for the procedure to provide information to its caller?

(b) Suppose some PL/SQL code has declared a numeric variable, BEST, and that this variable should contain the maximum value found in the RATING column of the BEERS table. Write a reasonably simple PL/SQL statement to achieve this.

(c) How would the BEST variable be declared to have the same type as the BEERS table's RATING column, without needing to know what that type is?

(d) What would a suitable declaration be for BEERS_CURSOR to enable the following loop?

```
for b in beers_cursor loop
  -- The body of the loop goes here.
  -- This body is executed for each row of the BEERS table,
  -- from highest RATING to lowest RATING, and can make use
  -- of each row's NAME, QUANTITY, and RATING.
end loop;
```

(e) Within the body of the preceding loop, what notation would be used for the value of the NAME column in the current row?

(f) Suppose that instead of using the preceding FOR loop, the cursor is accessed using an OPEN statement and a FETCH statement. What notation would be used to determine whether the FETCH statement succeeded in retrieving any data from the table?

6. [**10 Points**] Considering the following excerpt from my beer inventory:

Quantity	*Name*	*Rating*
4	Ommegang Hennepin	93
5	Two Hearted Ale	100
5	Founders Porter	99

(a) Write an XML file corresponding to this data that uses empty elements with attributes. You can omit the initial XML declaration line, which is

```
<?xml version="1.0" encoding="UTF-8" ?>
```

(b) Write a second XML file corresponding to the same data but this time using only nonempty elements and no attributes. You can again omit the initial XML declaration.

7. [**10 Points**]

```
import java.sql.*;

public class Beers{
    public static void main(String[] args){
        try{
            int a = Integer.parseInt(args[0]);
            int b = Integer.parseInt(args[1]);
            Class.forName("oracle.jdbc.OracleDriver");
            Connection myCon = DriverManager.getConnection
                ("jdbc:oracle:oci:@//thebe.gac.edu:1521/xe",
                 "anon", "anon");
            PreparedStatement s = myCon.prepareStatement
                ("select rating, quantity, name from max.beers" +
                 " where quantity >= ?" +
                 " and rating >= ?");
            s.setInt(2, a); //<----------------------------------LINE 1
            s.setInt(1, b);
            ResultSet rs = s.executeQuery();
            while(rs.next()){ //<--------------------------------LINE 2
                System.out.print(rs.getInt(1)); //<----------------LINE 3
                System.out.print(" ");
                System.out.print(rs.getInt(2));
                System.out.print(" ");
                System.out.print(rs.getString(3));
                System.out.println();
            }
        } catch(Exception e){
            System.err.println(e);
            System.exit(1);
        }
    }
}
```

Answer the following questions on the next page:

(a) What does the JDBC method invocation on LINE 1 do?

(b) What does the JDBC method invocation on LINE 2 do? Be sure to consider both the first time it is invoked and any subsequent times.

(c) What does the JDBC method invocation on LINE 3 do?

(d) The integer variables a and b are initialized from information passed to the program. By looking further down in the program at how these variables are used, write a simple description of the meaning of each of these two quantities. That is, you should explain the meaning of a and b in terms that make sense for a user of the program who doesn't care about the code.

(e) Write a description of the output the user should expect to see when running this program. Your description should make reference both to the format of the output and its meaning for a reader who doesn't care about the code. Assume no Exception occurs.

8. [**10 Points**] Suppose this XML document:

```
<?xml version = ’1.0’ encoding="UTF-8"?>
<hotels
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="bogotels.xsd">
  <hotelinfo NAME="ACME Luxury Acapulco">
    <STARS>4</STARS>
    <BEACH>Y</BEACH>
    <DESCRIPTION>The ACME Luxury Acapulco is a luxurious beautiful
      beach front with all the amenities that a 4 star hotel
      is expected to have.</DESCRIPTION>
  </hotelinfo>
  <hotelinfo NAME="ACME Luxury San Francisco">
    <STARS>5</STARS>
    <BEACH>Nearby</BEACH>
  </hotelinfo>
  <hotelinfo>
    <DESCRIPTION>Also contains the CICE</DESCRIPTION>
    <STARS>3</STARS>
    <BEACH>N</BEACH>
  </hotelinfo>
</hotels>
```

is validated relative to the Schema on the next page:

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:simpleType name="starsType">
  <xs:restriction base ="xs:integer">
   <xs:minInclusive value="1"/>
   <xs:maxExclusive value="5"/>
  </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="ynType">
  <xs:restriction base ="xs:string">
   <xs:enumeration value="Y"/>
   <xs:enumeration value="N"/>
  </xs:restriction>
 </xs:simpleType>
 <xs:element name="hotels">
  <xs:complexType>
    <xs:sequence>
     <xs:element name="hotelinfo" maxOccurs="unbounded">
       <xs:complexType>
        <xs:all>
         <xs:element name="DESCRIPTION" type="xs:string"/>
         <xs:element name="STARS" type="starsType"/>
         <xs:element name="BEACH" type="ynType"/>
        </xs:all>
        <xs:attribute name="NAME" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

(a) Circle of the four places in the XML document where it violates the Schema and write next to each a brief explanation.

(b) Show how to modify the Schema so as to make the DESCRIPTION optional.

9. [**10 Points**] Consider this XML document:

```
<Candidates>
  <County id="88" name="MULTI-COUNTY">
    <Office id="0101" name="US PRESIDENT AND VICE PRESIDENT">
      <Candidate id="01010301" party="R">JOHN MCCAIN AND SARAH PALIN</Candidate>
      <Candidate id="01010401" party="DFL">BARACK OBAMA AND JOE BIDEN</Candidate>
      <Candidate id="01010501" party="GP">CYNTHIA MCKINNEY AND ROSA CLEMENTE</Candidate>
      <Candidate id="01010601" party="SWP">ROGER CALERO AND ALYSON KENNEDY</Candidate>
      <Candidate id="01010801" party="IND">RALPH NADER AND MATT GONZALEZ</Candidate>
      <Candidate id="01010901" party="LIB">BOB BARR AND WAYNE A. ROOT</Candidate>
      <Candidate id="01011001" party="CP">CHUCK BALDWIN AND DARRELL CASTLE</Candidate>
      <Candidate id="01019901" party="WI">WRITE-IN**</Candidate>
    </Office>
    <Office id="0103" name="US SENATOR">
      <Candidate id="01030201" party="IP">DEAN BARKLEY</Candidate>
      <Candidate id="01030301" party="R">NORM COLEMAN</Candidate>
      <Candidate id="01030401" party="DFL">AL FRANKEN</Candidate>
      <Candidate id="01030901" party="LIB">CHARLES ALDRICH</Candidate>
      <Candidate id="01031001" party="CP">JAMES NIEMACKL</Candidate>
      <Candidate id="01039901" party="WI">WRITE-IN**</Candidate>
    </Office>
  </County>
  <County id="52" name="NICOLLET">
    <Office id="0392" name="COUNTY COMMISSIONER DISTRICT 2">
      <Candidate id="03929001" party="NP">JAMES STENSON</Candidate>
      <Candidate id="03929901" party="WI">WRITE-IN**</Candidate>
    </Office>
  </County>
</Candidates>
```

Write XQuery expressions for each of the following:

(a) Find the `Candidate` elements for the `LIB` party and produce the sequence of these elements surrounded by the tags `<Libertarians>` and `</Libertarians>`.

(b) Find the names (as strings, not name attributes) for the offices that have candidates from both the R and DFL parties. List each only once.

(c) Find the `Office` elements that have more than two candidates.

(d) Find the `Office` elements where every candidate for that office has either NP or WI as its party attribute.

(e) Transform the given data into this form:

```
<Candidates>
  <Candidate>
    <Name>JOHN MCCAIN AND SARAH PALIN</Name>
    <Office>US PRESIDENT AND VICE PRESIDENT</Office>
    <Party>R</Party>
  </Candidate>
  ...
  <Candidate>
    <Name>WRITE-IN**</Name>
    <Office>COUNTY COMMISSIONER DISTRICT 2</Office>
    <Party>WI</Party>
  </Candidate>
</Candidates>
```

10. [**10 Points**] An XML document begins as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE A [
  <!ELEMENT I EMPTY>
  <!ELEMENT II EMPTY>
  <!ELEMENT III EMPTY>
  <!ELEMENT B EMPTY>
  <!ELEMENT C (I | II | III)>
  <!ATTLIST C
    x (alpha | beta) #REQUIRED
    y CDATA #IMPLIED>
  <!ELEMENT A (B*, C+)>
  ]>
```

Write the shortest possible conclusion to this XML document so that it will be valid with regard to the included DTD. That is, you should use as few characters as possible while keeping the XML valid (not just well formed).