CS 600.105, M & Ms: Data Management Homework
Due date: Tuesday, 11/15/2011.

1 Question 1: Database Design

In this question, we will use a simple single-table database listing well-known paintings, and the gallery where the painting is on view, with a schema: *(Painting, Artist, Year, Gallery)*

An example of this database is:

Painting	Artist	Year	Gallery
Madonna Litta	Da Vinci	1490	Hermitage Museum, St. Petersburg
The Last Supper	Da Vinci	1495	Santa Maria delle Grazie, Milan
La Gioconda (Mona Lisa)	Da Vinci	1503	Louvre, Paris
St. George and the Dragon	Raphael	1504	National Gallery of Art, DC
The Fighting Temeraire	Turner	1838	National Gallery, London
Starry Night Over the Rhone	Van Gogh	1888	Musee d'Orsay, Paris
The Scream	Munch	1893	National Gallery, Oslo
The Scream	Munch	1910	The Munch Museum, Oslo
Painting with Troika	Kandinsky	1911	The Art Institute, Chicago
Three Musicians	Picasso	1921	Museum of Modern Art, New York
Composition VIII	Kandinsky	1923	Guggenheim Museum, New York
The Persistence of Memory	Dali	1931	Museum of Modern Art, New York
Guernica	Picasso	1937	Museo Reina Sofia, Madrid
Nighthawks	Hopper	1942	The Art Institute, Chicago
The Snail	Matisse	1953	Tate Gallery, London

Consider the following questions:

1. for the above database, what are the candidate keys?

   (Year) and any combination of Year and other attributes, (Painting, Gallery), (Artist, Gallery). Of these, in a more complex database, (Year) and (Artist, Gallery) are unlikely to be candidate keys – there are certainly many paintings created every year, and an artist may have multiple works displayed at a gallery.

2. what are the implications if the entire tuple is the only candidate key?

   A painting with a certain name, artist and year of creation is on display at two different galleries. This implies duplicates of the painting.

3. how and why would you use multiple tables to represent the above data?

   I would separate the above table into two: Painting(id, name, artist, year) and OnDisplay(painting id, gallery). While paintings may not move around very frequently, maintaining the two tables above would be easier when updates occur. Also, extending the schema with additional attributes relating to the painting or to the showing, for example as in the question below, would be more straightforward.

4. if you were a museum curator, what additional data would you collect to make a decision on the pieces to include in your gallery's collection?

This is an open-ended question, you might want to collect statistics on the showing and the painting's popularity, as well as more detailed artistic attributes to maintain a diverse collection of works.

2 Question 2: Object-Relational Mappings

Database query languages are often used through general-purpose programming languages, for example to build Web services. This question explores the relationship between data represented in a database, and in a programming environment that you may be more familiar with, namely an object-oriented language such as Java. Consider the following simplified Java code:

```
public class MovingObject {
  public double t;
  public double latitude;
  public double longitude;
  public double speed;
  public Moveable o;

  // Remainder of class, e.g. constructors, methods, etc.
}

public class World {
  MovingObject[] objects;

  // Constructor goes here...

  MovingObject[] getFastObjects() {
    LinkedList<MovingObject> r = new LinkedList<MovingObject>();
    for (MovingObject mo : objects) {
      if ( mo.speed > 60 ) { r.add(mo); }
    }
    return r.toArray(new MovingObject[0]);
  }
}
```

With the above program:

1. Considering that the subclasses of `Moveable` are `Car`, `Bike`, `Person`, what tables and attributes would you use in a database to represent a `World`?

   Tables:

   MovingObjects(o_id bigint, t double, latitude double, longitude double, speed double),

   Moveables(o_id bigint, type int, other attributes...), where type = 1 for Car, 2 for Bike, 3 for Person

2. With your database, how would you implement the `getFastObjects()` method in SQL?

   select Moveables.* from MovingObjects, Moveables
   where MovingObjects.o_id = Moveables.o_id and MovingObjects.speed > 60;

3 Question 3: Join Algorithms

In the third lecture of this section, we saw the nested loops join algorithm. A sketch of this algorithm is listed below, and as described in class, its I/O cost in terms of the number of pages read is: $cost = p_R + t_R * p_S$, where p_R and p_S are the number of pages used by two tables R, S respectively, and t_R, t_S the number of records they contain. How would you improve the algorithm to achieve better I/O performance? Assume the total memory in the system is M. (Hint: think about how you could use the available memory)

```
1. for each page r in table R:
2.   read r into memory
3.   for each tuple i in page r:
4.     for each page s in table S:
5.       read s into memory
6.       for each tuple j in page s:
7.         if match(i,j) then output(i,j)
```

In the above algorithm, pages of S are read multiple times, once for each tuple i from R. We can do better by reading pages of S only once, for as many pages of R as we can fit in memory, except for one page worth of space to read S. This algorithm is an aggressive variant of the block nested loops join:

```
1. while pages remain in table R:
2.   read M-1 pages or whatever remains from R into memory
3.   for each tuple i from R in memory:
4.     for each page s in table S:
5.       read s into memory
6.       for each tuple j in page s:
7.         if match(i,j) then output(i,j)
```