**Database Management System – cs422 DE**

**Assignment 2 – Week 2**

-------------------------------------------------------------------------------------------------------------------------------------

**This assignment is based on lecture 2 (chapters 4 & 5).**

* Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
* Note that the completed assignment should be submitted in .doc, .docx, .rtf or .pdf format only.
* If you think that your answer needs explanation to get credit then please write it down.
* You are encouraged to discuss these questions in the Sakai forum.

---------------------------------------------------------------------------------------------------------------------------------------

1. **A relational database consists of a collection of**
   1. Tables
   2. Fields
   3. Records
   4. Keys  
      ANS: A
2. **A \_\_\_\_\_\_\_\_ in a table represents a relationship among a set of values.**
   1. Column
   2. Key
   3. Row
   4. Entry  
      ANS: C
3. **For each attribute of a relation, there is a set of permitted values, called the \_\_\_\_\_\_\_\_ of that attribute.**
4. Domain
5. Relation
6. Set
7. Schema

ANS: A

1. **Course(course\_id, sec\_id, semester)  
   Here the course\_id, sec\_id and semester are \_\_\_\_\_\_\_\_\_\_ and course is a \_\_\_\_\_\_\_\_\_ .**
   * 1. Relations, Attribute
     2. Attributes, Relation
     3. Tuple, Relation
     4. Tuple, Attributes

ANS: B

1. **Department (dept\_name, building, budget) and   
   Employee (emp\_id , name, dept\_name, salary)   
   Here the dept\_name attribute appears in both the relations.   
   Using the common attributes in relation schema is one way of relating \_\_\_\_\_\_\_\_\_\_\_ relations.** 
   * 1. Attributes of common
     2. Tuple of common
     3. Tuple of distinct
     4. Attributes of distinct

ANS: C

1. **Student (ID, name, dept\_name, tot\_pts)  
   In this query which attribute form the primary key?**
2. name
3. dept\_name
4. tot\_pts
5. ID  
   ANS: D
6. **The\_\_\_\_\_ operation allows the combining of two relations by merging pairs of tuples, one from each relation, into a single tuple.**
7. Select
8. Join
9. Union
10. Intersection

ANS: B

1. **Discuss the differences between the five Join operations: Theta join, Equijoin, Natural join,**

**Outer join (left), and Semijoin. Example of each is appreciated.**

ANS:

* **Theta join** – This join type combines tuples from different relations provided they satisfy the theta condition. It is also referred as a join that links tables based on a relationship other than equality between two columns.
* **Equijoin** – Equijoin is a Theta join that uses only equality comparison operators.
* **Natural join** - Natural join does not use any comparison operator. We can perform a Natural Join only if there is at least one common attribute that exists between two relations.
* **Outer join (left)** - All the tuples from the left relation are included in the resulting relation. If there are tuples in the left relation without any matching tuple in the right relation, then the right relation attributes of the resulting relation will be set to NULL.
* **Semijoin** – This join type returns a row from one join input if there is at least one matching row on the other join input.

1. **A relational database contains details about journeys from Chicago to a variety of destinations and contains the following relations:**

*Operator* (**opCode**, opName)

*Journey* (**opCode, destCode**, price)

*Destination* (**destCode**, destName, distance)

**Each operator is assigned a unique code (opCode) and the relation *Operator* records the association between this code and the Operator’s name (opName).   
  
Each destination has a unique code (destCode) and the relation *Destination* records the association between this code and the destination name (destName), and the distance of the destination from Chicago.**

**The relation *Journey* records the price of an adult fare from Chicago to the given destination by a specified operator; several operators may operate over the same route.**

**Formulate the following queries using relational algebra.**

1. List the details of journeys less than $100.
2. List the names of all destinations.
3. Find the names of all destinations within 20 miles.

)

1. List the names of all operators with at least one journey priced at under $5.
2. List the names of all operators and prices of journeys to ‘Boston’.
3. **Solve Q 5.8 (a-d) on page no. 130 from the course text book (5th edition).**
   * 1. ΠhotelNo (σprice > 50 (Room))  
        ANS: The result of this query is a relation with one attribute (i.e. hotelNo) that contains hotel rooms whose price is above $50.
     2. σHotel.hotelNo = Room.hotelNo(Hotel × Room)  
        ANS: This query joins the Hotel and Room relations using hotelNo (as a common attribute) and selects all rooms in every hotel.
     3. ΠhotelName (Hotel ⋈ Hotel.hotelNo = Room.hotelNo (σprice > 50 (Room)))  
        ANS: This query produces a relation that has all hotelNames that have a room price above $50.
     4. Guest ⋊ (σdateTo ≥ ‘1-Jan-2007’ (Booking))  
        ANS: This query will select all guests whose Booking dateTo is greater than or equal to 1-Jan-2007. Since it uses the left outer join, the query will also include those guests that do not meet this selection criterion.