Database Management System – cs422 DE

Assignment 2 – Week 2

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

---

(1) A relational database consists of a collection of

A. Tables
B. Fields
C. Records
D. Keys

ANS: A

(2) A _____ in a table represents a relationship among a set of values.

A. Column
B. Key
C. Row
D. Entry

ANS: C

(3) For each attribute of a relation, there is a set of permitted values, called the _____ of that attribute.

A. Domain
B. Relation
C. Set
D. Schema

ANS: A

(4) Course(course_id, sec_id, semester)
Here the course_id, sec_id and semester are _____ and course is a _____ .

A. Relations, Attribute
B. Attributes, Relation
C. Tuple, Relation
D. Tuple, Attributes

ANS: B

(5) 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.

A. Attributes of common
B. Tuple of common
C. Tuple of distinct
D. Attributes of distinct

ANS: C

(6) Student (ID, name, dept_name, tot_pts)
In this query which attribute form the primary key?

A. name
B. dept_name
C. tot_pts
D. ID
ANS: D

(7) The____ operation allows the combining of two relations by merging pairs of tuples, one from each relation, into a single tuple.

A. Select
B. Join
C. Union
D. Intersection
ANS: B

(8) 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: It allows to merge two tables based on the condition represented by Theta. Theta join works for all comparison operators. The general case of JOIN operation is called a Theta Join.

Example:

```
SELECT p1.FirstName, p2.FirstName
FROM Person p1
INNER JOIN Person p2
ON LEN(p1.FirstName)>LEN(p2.FirstName)
```

Equi Join: When Theta join uses only equality comparison operator, it is said to be equijoin. It is the most common form of SQL inner join. It contains an equality operator.

```
Select DISTINCT A.StateID, S.Name
From Person.Address A
Inner Join Person.Sate S
On A.StateID=S.StateID
```

Natural join: Natural join exists only if there is at least one common attribute that exists between two relations. In addition, the attributes must have the same name and domain. We can't use comparison operator in Natural join.

```
SELECT *
FROM Student NATURAL JOIN Marks;
```

Outer Join (left): Outer join is used to return the results by combining rows from two or more tables. Outer join will return every row from one specified table, even if the join condition fails. Ex:

1. Students Table with student_id, name
2. Books Table with title, student_id, author

Here we are selecting name form Student Table and title from the Books table based on the common student_id on both tables.

```
SELECT students.name, books.title
FROM students
FULL OUTER JOIN books ON students.student_id=books.student_id;
```

Semi Join: In this join operation where the EXISTS clause is used with a subquery. Even if duplicated rows are returned in the subquery, only one set of matching values in the outer query is returned.

```
SELECT P.ProductID
FROM Production.Product AS P
WHERE EXISTS
(
  SELECT *
  FROM Production.Inventory AS INV
  WHERE INV.ProductID = P.ProductID
);
```


(9) 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.
4) List the names of all operators with at least one journey priced at under $5.
5) List the names of all operators and prices of journeys to 'Boston'.

ANS:

1) $\sigma_{price < 100}(Journey)$
2) $\Pi_{destName}(Destination)$
3) $\Pi_{destName}(\sigma_{distance < 20}(Destination))$
4) $\Pi_{opName}(\sigma_{price<5}(Journey) \bowtie opcode\ Operator)$
5) $\Pi_{opName,\ price}((\sigma_{destName='Boston'}(Destination) \bowtie destCode(Journey \bowtie opcode\ Operator)$

(10) Solve Q 5.8 (a-d) on page no. 130 from the course text book (5th edition).

a) $\Pi_{hotelNo}(\sigma_{price > 50}(Room))$
ANS: This hetelNo attribute will give the number of hotels with a room of price more than $50.

b) $\sigma_{Hotel.hotelNo = Room.hotelNo}(Hotel \times Room)$
ANS: This is an equijoin relation of the Hoel and Room which gives the result containing all the attributes of both room and hotel and the output will contain all the rooms in a hotel.

c) $\Pi_{hotelName}(Hotel \bowtie_{Hotel.hotelNo = Room.hotelNo} (\sigma_{price > 50}(Room)))$
ANS: This gives the list of hotel names with hotel room prices higher than $50.

d) $Guest \bowtie (\sigma_{dateTo \geq '1\text{-}Jan\text{-}2007'}(Booking))$
ANS: Left outer join of Guest and tuples of booking with end date greater or equal to 1st January 2007. However, guests without this dated constraint will still appear.