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

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: B

(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: A

(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: Allows arbitrary join conditions using comparison operators.
- Equijoin: Matches rows based on equality of values in specified columns.
- Natural Join: Automatically matches columns with the same name in both tables.
- Outer Join (Left): Includes all rows from the left table and matching rows from the right table, with NULL values for non-matching rows.
- Semijoin: Returns rows from the first table where there is a match in the second table, without duplicating rows from the first table.

(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. σ(price < 100)(Journey)
2. π(destName)(Destination)
3. π(destName)(σ(distance < 20)(Destination))
4. π(opName)(σ(price < 5)(Journey ⋈ Operator))
5. π(opName, price)(σ(destName = 'Boston')(Journey ⋈ Operator ⋈ Destination))

(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: Selects hotelNo of Room with a price greater than $50.

b) $\sigma_{Hotel.hotelNo = Room.hotelNo}(Hotel \times Room)$
ANS: Joins Hotel and Room on matching hotelNo.

c) $\Pi_{hotelName}(Hotel \bowtie_{Hotel.hotelNo = Room.hotelNo} (\sigma_{price > 50}(Room)))$
ANS: Selects hotelName of Hotel with Room priced greater than $50.

d) $Guest \rtimes (\sigma_{dateTo \geq '1\text{-}Jan\text{-}2007'}(Booking))$
ANS: Left Outer Join all Guest and their Booking after '1-Jan-2007'.