

# COP 4710 Database Systems

## Assignment 3

**Instructor:** Dr. David A. Gaitros

### **Problem1 (25 Points)**

Consider the following enrollment database used at FSU:

- Student (FSUID, Name, Major, Level, Age)
- Course (CourseName, Department, Time, Venue, FacultyID)
- Faculty (FacultyID, Name, Department)
- Enrollment (FSUID, CourseName, score)

Write the following queries in SQL. (Note: No duplicate tuples should be returned in any of the answers.)

1. (5 Points) Find the name of the youngest student (If more than one student is the youngest, return them all);
2. (5 Points) Find the age of the youngest student enrolled in the course "COP4710";
3. (5 Points) For every course taught by Professor Indiana Jones, Find the course name and the average score of that course;
4. (10 Points) Find the names of faculty members for whom the combined enrollment in the courses they teach is less than 50 (Hint: you are encouraged to define and use views to break down the task).

### **Problem2 (20 Points)**

Consider a database about the customers who purchase laptops:

- Laptop(model, manufacturer, type, price)
- Customer(custID, zipcode)
- Purchase(model, custID, date)

1. (5 Points) For the Black Friday promotion, the manufacturer DELL decides to lower prices on expensive laptops. Write a SQL statement that takes a 20% discount from the price of all DELL laptops currently priced over \$1,750;
2. (10 Points) Create a view called LoyalCustomers which shows every custID and manufacturer pair if that customer has bought more than 2 laptops from that manufacturer (thus, showing loyalty to a particular manufacturer);
3. (5 Points) Not all manufacturers have loyal customers. Using the view LoyalCustomers you have just created, find all manufacturers which have no loyal customers. Your query should return a distinct list of manufacturers.

**Problem3 (15 Points)**

Consider the same schema from problem 2,

1. (10 Points) Suppose you want to enforce the following constraint: when inserting a tuple into the Purchase relation, custID must also exist in the Customer relation. If custID does not exist in the Customer table, automatically insert it into the Customer table (you can set the zipcode value to be NULL). Write a Trigger statement to do this;
2. (5 Points) Can we specify Check constraints to do the same thing? Why or why not?

**Problem4 (20 Points)**

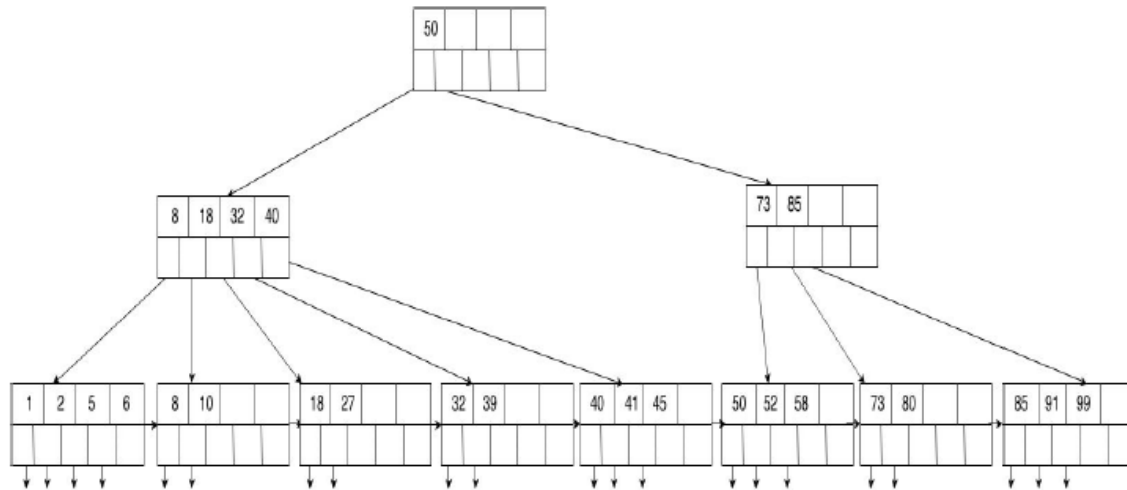
Suppose we have a relation of 3,000 tuples. Each memory block can hold 4 tuples, or 25 key-pointer pairs for indexes.

1. (7 Points) How many blocks do we need for a dense index of this relation?
2. (8 Points) How many blocks do we need for a sparse index of this relation?
3. (5 Points) The above questions are concerned with clustered indices, which can be either dense or sparse. In contrast, an unclustered index is always dense. Why is this true?

**Problem5 (20 Points)**

Consider the following B+ Tree, shown below, of order  $n = 2$  (i.e., each index node can hold at most 4 keys and 5 pointers):

1. (5 points) Show the steps necessary (i.e., how you would traverse the tree) to look up all the records in the range 8 to 30 in this tree;
2. (10 Points) Show the tree that would result from inserting a data entry with key 3 into the original tree. You only need to show the final tree, no intermediate trees;



3. (5 Points) Show the tree that would result from deleting the key 8 from the original tree. You only need to show the finished tree.