## Part II: Attempt all questions [70 pts].

### Database Context:

Consider the following database schema (**EmployeeProject*DB***) to answer the questions that follow.

| FirstName | LastName | Dept | Salary |
| :--- | :--- | :--- | :--- |
| John | Smith | Sales | 50000 |
| Mary | Johnson | HR | 55000 |
| Robert | Brown | HR | 48000 |
| Susan | Davis | IT | 60000 |
| Marx | Prince | Sales | 64500 |
| David | Lee | HR | 52000 |
| Peter | Kelly | Sales | 70000 |
| Kaya | Lui | Sales | 45000 |
| Son | Xue | HR | 55000 |
| Tim | Terry | IT | 80000 |

**Employee table**

| ProjectName | Dept | Manager |
| :--- | :--- | :--- |
| Project A | Sales | John Smith |
| Project B | HR | Mary Johnson |
| Project C | HR | Robert Brown |
| Project D | Sales | Peter Kelly |
| Project E | Sales | Kayla Lui |

**Project table**

| JobID | Job | ProjectName |
| :--- | :--- | :--- |
| J001 | Survey | Project B |
| J002 | Inspection | Project B |
| J003 | Excavation | Project B |
| J004 | Development | Project A |

Estimate the result sizes (in # bytes) of the queries below: (**Assume:** FirstName = 8 bytes of string; LastName = 10 bytes of string; Dept = 6 bytes of string; Salary = 9 bytes of integer; ProjectName = 6 bytes of string; Manager = 10 bytes of string; JobID = 4 bytes; and Job = 8 bytes of string)
**Note: Points will be awarded for stating the correct formulas.**

1. *Query 1*
   *SELECT FirstName, LastName, Dept*
   *FROM Employee*
   *WHERE Dept != "IT";*

2. **Query 3**
   *SELECT \**
   *FROM Employee*
   *WHERE Dept ="Sales" OR Salary < 50000;*

3. **Query 3**
   *SELECT \**
   *FROM Project P*
   *INNER JOIN Job J*
   *ON P.ProjectName = J.ProjectName;*

4. *Query 4*
   *SELECT ***
   *FROM Employee, Job;*

5. *Query 5*
   *SELECT ***
   *FROM Employee*
   *NATURAL JOIN Project*
   *NATURAL JOIN Job;*

6.  Assume that each B+-tree node can hold a maximum of two (2) keys. Construct the B+-trees resulting from the insertion and deletion operations of the key sequence provided below

    (a)     Insert 11, 20, 40, 13, 45, 18, 1, 29, 38, 60, 4, 9, 5, 12, 70, 30 **[show individual tree at each insertion] [8 pts].**

**(b)**    **Deletion: Consider the tree after inserting 30 in part (a).**
Delete 4, 12, 9, 13, 45, 18, 40, 70, 38, 5, 1 **[show individual tree at each deletion]**
**[12 pts].**

[cont'd: Show individual trees at each deletion].

1. Consider the following schedules **(S1 and S2)** with two transactions **T1** and **T2** as shown below.

Schedule 1 (S1)

| Instructions | T1 | T2 |
|---|---|---|
| 1 | Read (A) | |
| 2 | | Read (A) |
| 3 | | A = A + 20 |
| 4 | | Write (A) |
| 5 | | Read (B) |
| 6 | | B = B - 20 |
| 7 | | Write (B) |
| 8 | | commit2 |
| 9 | A = A + 10 | |
| 10 | Write (A) | |
| 11 | Read (B) | |
| 12 | B = B - 10 | |
| 13 | Write (B) | |
| 14 | commit1 | |

Schedule 2 (S2)

| Instructions | T1 | T2 |
|---|---|---|
| 1 | Read (A) | |
| 2 | A = A + 10 | |
| 3 | Write (A) | |
| 4 | Read (B) | |
| 5 | B = B - 10 | |
| 6 | Write (B) | |
| 7 | commit1 | |
| 8 | | Read (A) |
| 9 | | A = A + 20 |
| 10 | | Write (A) |
| 11 | | Read (B) |
| 12 | | B = B - 20 |
| 13 | | Write (B) |
| 14 | | Commit2 |

a. Are S1 and S2 serializable schedules? Please explain along with any form of evidence or proof. ***Assume an initial value: A = B = 100 [10 pts]***

2. Check whether schedule S is conflict serializable or not. If S is conflict serializable, give all possible schedules equivalent to Schedule S **[15 pts]**

**Schedule S:** W1(A), W1(B), R2(B), W2(C), R3(C), W3(D), R4(D), R2(C), W4(A), R5(B), W5(E), R6(E), W6(D), R5(E).

**Good luck**