ICS 311 - Database Management Systems
Assignment 6

My data:

```
INSERT INTO DEPT VALUES(1,'HR','NEW YORK', CURRENT_TIMESTAMP(), current_timestamp());
INSERT INTO DEPT VALUES(10,'ACCOUNTING', 'NEW YORK', CURRENT_TIMESTAMP(),
current_timestamp());
INSERT INTO DEPT VALUES(5,'IT', 'NEW YORK', CURRENT_TIMESTAMP(), current_timestamp());

INSERT INTO SALGRADE VALUES(1,20500.00,25600.00);
INSERT INTO SALGRADE VALUES(2,25600.00,30700.00);
INSERT INTO SALGRADE VALUES(3,30700.00,35800.00);

INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9618', 'Sanchez',
'Samantha', 'Clerk', '7782', '2005-06-19', '22400.00', '1', '10', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('6541', 'Harrison', 'Claire',
'Support', '5685', '2015-07-06', '27600.00', '2', '5', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('7890', 'Richard', 'James',
'Manager', '4845', '2001-11-11', '35800.00', '3', '1', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9878', 'Smith', 'Melanie',
'Support', '5685', '2006-05-15', '28600.00', '2', '5', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('6423', 'Jackson', 'Joey',
'Support', '5685', '2009-09-12', '25600.00', '1', '5', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('8456', 'Lucas', 'Angie',
'HR', '4845', '2003-05-05', '27500.00', '2', '1', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9564', 'Tiu', 'Angel', 'HR',
'4845', '2019-06-19', '23400.00', '1', '1', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('8488', 'Sy', 'Leslie',
'Assistant', '7782', '2010-03-05', '29600.00', '2', '1', '2021-08-03', '2021-08-03');
```

```
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('8215', 'Pincher', 'Sam',
'Clerk', '7782', '2012-06-18', '28400.00', '2', '10', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9401', 'Hendricks', 'Randy',
'Assistant', '7782', '2009-08-31', '24500.00', '1', '10', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9455', 'Cruz', 'Daniel',
'Clerk', '4845', '2007-10-10', '26500.00', '2', '1', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9989', 'Smith', 'John',
'Support', '5685', '2018-02-10', '28999.00', '2', '1', '2021-08-03', '2021-08-03');
INSERT INTO EMP (EMP_ID, E_LAST_NAME, E_FIRST_NAME, JOB, MGR_ID, HIREDATE, SAL,
GRADE_ID, DEPT_ID, DATE_INSERTED, DATE_LASTUPDATED) VALUES ('9783', 'Smith', 'John',
'Support', '5685', '2018-02-18', '28900.00', '2', '1', '2021-08-03', '2021-08-03');
```

1. Create a procedure that takes in a DNAME (Department Name) value as an incoming parameter and show the total salary for that department, include DEPT_ID, DNAME, and total salary as output.

```
DELIMITER $$;
CREATE PROCEDURE PRC_SAL_PER_DEPT(
      IN p_dept_id int(2)
)
BEGIN
      SELECT D.DEPT_ID,D.DNAME, SUM(E.SAL)
    FROM DEPT D INNER JOIN EMP E on D.DEPT_ID = E.DEPT_ID
    WHERE D.DEPT_ID = p_dept_id;
END $$;
```

1. Create a procedure that takes in an EMP_ID value as in incoming parameter and show Employee’s full name, current salary, and the salary range (losal – hisal) for that Employee’s salary grade.

```
DELIMITER $$;
CREATE PROCEDURE PRC_GET_EMP_SALARY(
      IN p_emp_id INT(4)
)
BEGIN
      SELECT CONCAT(E.E_FIRST_NAME, ' ' , E.E_FIRST_NAME) AS FULL_NAME,
E.SAL, CONCAT(S.LOSAL , ' - ' , S.HISAL) AS SALARY_RANGE
      FROM EMP E INNER JOIN SALGRADE S ON E.GRADE_ID = S.GRADE_ID
    WHERE E.EMP_ID = p_emp_id;
END $$;
```

2. Create the best index (show create command) that would be helpful to both these queries, and prove the index is being used with the Explain command:

   Select e_last_name, hiredate, sal from employee where e_last_name = ‘Smith’;
   Select e_last_name, e_fist_name, dept_id from employee where e_last_name = ‘Smith’ and e_first_name = ‘Jonn’;

   ```
   CREATE INDEX E_LAST_NAME_IDX ON EMP(E_LAST_NAME);
   EXPLAIN SELECT E_LAST_NAME, HIREDATE, SAL FROM EMP WHERE E_LAST_NAME =
   'Smith';
   EXPLAIN SELECT E_LAST_NAME, E_FIRST_NAME, DEPT_ID FROM EMP WHERE
   E_LAST_NAME = 'Smith' AND E_FIRST_NAME = 'John';
   ```

3. Create a procedure that has one input parameter a value for MGR_ID , and an out parameter to return the employee id with the highest salary for that manager. You will have to do a little research on the output parameter and how to use it.

```
DELIMITER // ;
CREATE PROCEDURE PRC_GET_EMP_ID(
      IN p_mgr_id int(4),
    OUT p_emp_id INT(4)
)
BEGIN
      SELECT EMP_ID INTO p_emp_id
    FROM EMP
    WHERE MGR_ID = p_mgr_id
    ORDER BY SAL desc LIMIT 1;
END// ;
```

4. Which one of these should have an index? A query that runs every hour with DNAME in the PREDICATE clause, or a query that runs once a month with LOC in the WHERE clause? Show the

create index command for the one you picked. Create a SELECT query that will utilize the index you created and prove it was used with the Explain command.

```
CREATE INDEX LOC_IDX ON DEPT(LOC);

EXPLAIN SELECT * FROM DEPT WHERE LOC = 'New York';
```