SQL Practicum – 100 points total **Name**___________________________

1. (10 pts) Write a query that shows all attributes of CERTIFICATION. Sort by descending STATE_CERT_CODE as the primary sort key and by ascending CERT_DESC as the secondary sort key.

```
SELECT *
FROM CERTIFICATION
ORDER BY STATE_CERT_CODE DESC, CERT_DESC ASC;
```

2. (5 pts) Write a statement that changes the value of EMPLOYEE.DIRECT_ADMIN_ID to '12560' for any employee who has '11599' for their DIRECT_ADMIN_ID.

```
UPDATE EMPLOYEE
SET DIRECT_ADMIN_ID = '12560'
WHERE DIRECT_ADMIN_ID = '11599';
```

3. (5 pts) Write a statement that would insert the first record shown of the OT_PAY table if it was not in the database already. The first record is shown below.

```
PAB_ITEM_ID HOLIDAY_MULTIPLIER
----------- ------------------
   40011            1
```

```
Insert into OT_PAY (PAB_ITEM_ID,HOLIDAY_MULTIPLIER) values ('40011',1);
```

4. (10 pts) Write a query that returns the last name and the first name of every employee in the EMPLOYEE table if *both* of the following conditions are true: 1. the employee is either an administrator OR a teacher; 2. the employee was hired before June 18th, 2005.

```
SELECT EMP_LNAME, EMP_FNAME
FROM EMPLOYEE
WHERE (IS_ADMIN = 'Y' OR IS_TEACHER = 'Y')
AND HIREDATE < '18-JUN-2005';
```

5. (10 pts) Show each DISTRICT_NAME in the DISTRICT table with a count of the number of UNIQUE DIRECT_ADMIN_IDs associated with each district's employees. Alias the count appropriately.

```
SELECT DISTRICT_NAME, COUNT(DISTINCT DIRECT_ADMIN_ID) AS DA_COUNT
FROM DISTRICT, EMPLOYEE
WHERE DISTRICT.DISTRICT_ID = EMPLOYEE.DISTRICT_ID
GROUP BY DISTRICT.DISTRICT_ID, DISTRICT_NAME;
```

6. (10 pts) Use a subquery (either correlated or non-correlated) in a query to list the EMP_ID and EMP_LNAME for all employees who have a record in the TOTAL_PAB table for the 2009 TAX_YEAR;

```
SELECT EMP_ID, EMP_LNAME
FROM EMPLOYEE
WHERE EMP_ID IN
(SELECT EMP_ID
FROM TOTAL_PAB
WHERE TAX_YEAR = 2009);
```

7. (10 pts) For each teacher that has ***any*** certifications in the database, show the associated DISTRICT_NAME, EMP_ID, CERT_DESC, DATE_EFFECTIVE, and GRADE_OR_SPECIAL for each certification.

```
SELECT DISTRICT_NAME, EMP_ID, CERT_DESC, DATE_EFFECTIVE, GRADE_OR_SPECIAL
FROM DISTRICT D, EMPLOYEE E, TEACHER T,
TEACHER_CERT_INT I, CERTIFICATION C
WHERE D.DISTRICT_ID = E.DISTRICT_ID
AND E.EMP_ID = T.T_EMP_ID
AND I.T_EMP_ID = T.T_EMP_ID
AND C.CERT_ID = I.CERT_ID;
```

8. (10 pts) Write a query that shows any TOTAL_PAB record that is associated with *every* PAB_ITEM record. This would mean that the records in the result should have a relationship with each and every record in the PAB_ITEM table regardless of how many there are or if the records should change. Show EMP_ID, TAX_YEAR, and PAB_ID in your result.

```
SELECT EMP_ID, TAX_YEAR, PAB_ID
FROM TOTAL_PAB T
WHERE NOT EXISTS
(SELECT *
FROM PAB_ITEM I
WHERE NOT EXISTS
(SELECT *
FROM PAB_LINEITEM LI
WHERE T.PAB_ID = LI.PAB_ID
AND I.PAB_ITEM_ID = LI.PAB_ITEM_ID));
```

9. (10 pts) In the database, values for PAB_LINEITEM.AMOUNT_POSTED should always be greater than zero (0) – so it cannot be zero or negative. Write a statement to *add a constraint* to the existing PAB_LINEITEM table (assume that the table has been appropriately created previously – except for this constraint). The constraint should verify that all values entered into the AMOUNT_POSTED field are *greater than* zero (0).

```
ALTER TABLE PAB_LINEITEM ADD
CONSTRAINT LINEITEM_AMOUNT_POSTED_CHK CHECK (AMOUNT_POSTED > 0);
```

10. (10 pts) Write a query that lists out each administrator in the ADMIN table (whether or not they have serve as direct administrator for any employee) along with each employee they are associated with as direct administrator (so, EMPLOYEE.DIRECT_ADMIN_ID would have the administrator's A_EMP_ID in it for that employee). Show A_EMP_ID, ADMIN_TITLE, EMP_LNAME, EMP_FNAME, and EDU_EMAIL (name and email information is for the employee with that direct administrator, this is not the information for the administrator). *Write the query so that the administrators' information is shown (A_EMP_ID, ADMIN_TITLE), even if they are not the direct administrator for anyone* (the EMP_LNAME. EMP_FNAME, and EDU_EMAIL attributes would be NULL for these records).

```
SELECT A_EMP_ID, ADMIN_TITLE, EMP_LNAME, EMP_FNAME, EDU_EMAIL
FROM ADMIN LEFT JOIN EMPLOYEE ON
A_EMP_ID = DIRECT_ADMIN_ID;
```

11. (10 pts) Write the SQL statement needed to create the table OTHER_EMP. Include the primary key constraint and any alternate key and foreign key constraints necessary.

```
  CREATE TABLE OTHER_EMP    (
  O_EMP_ID CHAR(5) NOT NULL,
     TYPE CHAR(30) NOT NULL,
     TITLE CHAR(50),
      CONSTRAINT OTHER_EMP_PK PRIMARY KEY (O_EMP_ID),
      CONSTRAINT OTHER_EMP_FK1 FOREIGN KEY (O_EMP_ID)
       REFERENCES EMPLOYEE (EMP_ID);
```