M&IS 44043 – SQL Practicum – 100 points total **Name_____**

1. (10 pts) Write a query that shows all attributes of P_WORK_OF_ART, sort by ART_CREATED_YEAR as the primary sort key and by ART_TITLE as the secondary sort key. Both ascending.

```
SELECT *
FROM P_WORK_OF_ART
ORDER BY ART_CREATED_YEAR, ART_TITLE;
```

2. (5 pts) Write a statement that changes the value for ARTIST_LNAME from 'Snow' to 'Rain' for the artist with ARTIST_ID = '30044'. Make sure no other artist that happens to have the last name of 'Snow' gets accidently changed.

```
UPDATE P_ARTIST
SET ARTIST_LNAME = 'Rain'
WHERE ARTIST_ID = '30044';
```

3. (5 pts) Write a statement that would insert the first record of the P_PERFORMANCE table if it was not there already.

```
Insert into P_PERFORMANCE (PERF_ART_ID,PERF_LENGTH_IN_MIN) values
('10022',120);
```

4. (10 pts) Write a query that returns all the attributes of the P_WORK_OF_ART table when *any* of the following conditions are true: 1. The ART_CREATED_YEAR is between 1962 and 1970 inclusive; 2. The ART_ID is 10021, 10022, 0r 10023; 3. The ART_CONDITION is 'UNKNOWN'.

```
SELECT *
FROM P_WORK_OF_ART
WHERE (ART_CREATED_YEAR BETWEEN 1962 AND 1970)
OR (ART_ID IN ('10021', '10022', '10023'))
OR (ART_CONDITION = 'UNKNOWN');
```

5. (10 pts) Show each LOC_DESC in P_LOCATION with a count of the number of works of art at the location, alias the count column appropriately:

```
SELECT LOC_DESC, COUNT(ART_ID) AS ART_COUNT
FROM P_LOCATION L, P_WORK_OF_ART A
WHERE L.LOCATION_ID = A. LOC_ID
GROUP BY LOC_DESC;
```

6. (10 pts) Use a correlated subquery to show the ART_ID and ART_TITLE for any works of art that also appear in the P_SEASONAL table.

```
SELECT ART_ID, ART_TITLE
FROM P_WORK_OF_ART A
WHERE EXISTS
     (SELECT *
     FROM P_SEASONAL S
     WHERE A.ART_ID = S.SEAS_ART_ID);
```

7. (10 pts) For each performance in the database show its associated values for: PERF_DESC, PERF_LENGTH_IN_MIN, ART_ID, ART_TITLE, ARTIST_FNAME, ARTIST_LNAME.

```
SELECT DISTINCT PERF_DESC, PERF_LENGTH_IN_MIN, W.ART_ID,
ART_TITLE, ARTIST_FNAME, ARTIST_LNAME
FROM P_ARTIST A, P_ARTIST_ART_INT I, P_WORK_OF_ART W,
P_PERFORMANCE P, P_PERFORMANCE_TIMINGS PT
WHERE A.ARTIST_ID = I.ARTIST_ID
AND I.ART_ID = W.ART_ID
AND P.PERF_ART_ID = W.ART_ID
AND P.PERF_ART_ID = PT.PERF_ART_ID;
```

8. (10 pts) Write a query that shows works of art that are associated with *every* type in the P_TYPE table. (show ART_ID, ART_TITLE, ART_CREATED_YEAR, ART_PICFILE):

```
SELECT ART_ID, ART_TITLE, ART_CREATED_YEAR, ART_PICFILE
FROM P_WORK_OF_ART A
WHERE NOT EXISTS
     (SELECT *
     FROM P_TYPE_ART_INT I
     WHERE NOT EXISTS
          (SELECT *
          FROM P_TYPE T
          WHERE A.ART_ID = I.ART_ID
          AND I.TYPE_ID = T.TYPE_ID));
```

9. (10 pts) In the database, values for P_ARTIST.ARTIST_ID always begin with a '3' and are always five characters long. Write a statement to add a constraint to the existing P_ARTIST table (assume it has been appropriately created previously – except for this constraint). The constraint should verify that all values entered into the ARTIST_ID begin with a '3' and have exactly 5 characters. You do not have to make sure each character is a number.

```
ALTER TABLE P_ARTIST ADD
CONSTRAINT P_ARTIST_FROM_EXAM3_CHK CHECK
(ARTIST_ID LIKE '3____');
```

10. (10 pts) Write a query that lists out each artist in the P_ARTIST table (whether or not they have any art works in the database) along with each work of art they are associated with. Write the query so that the artist's information is shown even if they have zero works of art in the database (the work of art attributes would be NULL for these records). Show ARTIST_ID, ARTIST_FNAME, ARTIST_LNAME, ART_ID, ART_TITLE.

```
SELECT A.ARTIST_ID, ARTIST_FNAME,
ARTIST_LNAME, W.ART_ID, ART_TITLE
FROM P_ARTIST A LEFT JOIN P_ARTIST_ART_INT I
ON A.ARTIST_ID = I.ARTIST_ID
LEFT JOIN P_WORK_OF_ART W ON I.ART_ID = W.ART_ID;
```

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

```
CREATE TABLE P_TYPE (
  TYPE_ID    CHAR(5) NOT NULL,
  TYPE_TITLE VARCHAR2(30) NOT NULL,
  TYPE_DESC  VARCHAR2(1000),
    CONSTRAINT P_TYPE_PK PRIMARY KEY (TYPE_ID),
    CONSTRAINT P_TYPE_UK1 UNIQUE (TYPE_TITLE));
```