

```

1  # ENSF 608 F23 Assignment 3 Suggested Solutions
2  USE COMPETITION;
3
4  # Question 1 (1 mark)
5  SELECT C.FName, C.LName, T.StudioName
6  FROM (COMPETITOR AS C JOIN TEACHER AS T ON C.TeacherID = T.TeacherID);
7
8  # Question 2 (1 mark)
9  SELECT T.StudioName, COUNT(*)
10 FROM (COMPETITOR AS C JOIN TEACHER AS T ON C.TeacherID = T.TeacherID)
11 GROUP BY T.StudioName;
12
13 # Question 3 (1 mark)
14 SELECT StudioName, COUNT(*)
15 FROM TEACHER
16 GROUP BY StudioName;
17
18 # Question 4 (1 mark)
19 SELECT T.LName
20 FROM (COMPETITOR AS C JOIN TEACHER AS T ON C.TeacherID = T.TeacherID)
21 GROUP BY C.TeacherID
22 HAVING COUNT(*) > 1;
23
24 # Question 5 (2 marks)
25 SELECT LName, FName, Title
26 FROM ((COMPETITOR AS C JOIN PERFORMANCE AS P ON C.CompetitorID = P.CompetitorID) JOIN
27 COMPOSITION ON P.MusicID = COMPOSITION.MusicID)
28 WHERE COMPOSITION.Genre = "Romantic";
29
30 # Question 6 (2 marks)
31 SELECT C.Title, P.CategoryID FROM (COMPOSITION AS C LEFT OUTER JOIN PERFORMANCE AS P ON
32 C.MusicID = P.MusicID);
33
34 # Question 7 (1 mark)
35 DROP VIEW IF EXISTS SCORE_ANALYSIS;
36 CREATE VIEW SCORE_ANALYSIS(CompAge, Score)
37 AS SELECT C.Age, P.Score
38 FROM PERFORMANCE AS P JOIN COMPETITOR AS C ON P.CompetitorID = C.CompetitorID;
39
40 # Question 8 (1 mark)
41 SELECT * FROM SCORE_ANALYSIS
42 ORDER BY Score DESC;
43
44 # Question 9 (1 mark)
45 SELECT MAX(Score), MIN(Score), AVG(Score)
46 FROM SCORE_ANALYSIS;
47
48 # Question 10 (2 marks)
49 ALTER TABLE COMPOSITION ADD COLUMN Copyright varchar(15) DEFAULT 'SOCAN';
50 SELECT * FROM COMPOSITION;
51
52 # Question 11 (2 marks)
53 SELECT Lname, Fname, Age FROM COMPETITOR AS COMP WHERE NOT EXISTS(
54 SELECT C.Age FROM (COMPETITOR AS C JOIN PERFORMANCE AS P ON C.CompetitorID =
55 P.CompetitorID) JOIN CATEGORY as G ON P.CategoryID = G.CategoryID
56 WHERE (C.Age <= G.AgeMax AND C.Age >= AgeMin AND COMP.CompetitorID = C.CompetitorID)
57 );
58
59 # Question 12 (1 mark)
60 ALTER TABLE COMPETITOR ADD CHECK (Age >= 5 AND Age <= 18);
61
62 # Question 13 (2 marks)
63 # Updated throughout the database due to the ON UPDATE CASCADE when creating the table.
64 UPDATE STUDIO SET Name = 'Harmony Studio' WHERE Name = 'Harmony Inc.';
65 SELECT * FROM STUDIO;
66

```

```
67 # Question 14 (1 mark)
68 # Deletion creates an error because the selected composer data has referential
69 # integrity constraints with COMPOSITION and PERFORMANCE.
70 DELETE FROM COMPOSITION WHERE Composer = 'Beethoven';
71
72 # Question 15 (1 mark)
73 # The following code will prevent any teacher information from being updated. It will
74 # send an error message asking for documentation to be provided to the office.
75 #DROP TRIGGER IF EXISTS Certification;
76 #CREATE TRIGGER Certification
77 #BEFORE UPDATE ON TEACHER FOR EACH ROW
78 #SIGNAL SQLSTATE '45000'
79 #SET MESSAGE_TEXT = 'Proof of certification must be provided to the main office.';
80
81 # Example: UPDATE TEACHER SET LName = 'Stone' WHERE LName = 'Steele';
82
```