Full Stack Development/Software Workshop 2

Formative Exercises 1

Based on lectures: 3,4,5

Exercise 1

Write a query for each table in the Music database that returns all columns and all rows.

Exercise 2

Write a query that returns the title, label and price of all the albums in the 'Album' table.

Exercise 3

Write a query that lists all of the *artistid* values in the 'Album' table but without duplicates.

Exercise 4

Write a query that lists the albums in the 'Album' table (all attributes) that were released prior to 1980 and cost more than £8.

Exercise 5

Write a query that lists the albums in the 'Album' table (all attributes) that have the *label* 'RCA' and were released between 1974 and 1990 (inclusive).

Exercise 6

Write a query that lists the albums in the 'Album' table (all attributes) that were released before 1989 but were not in the 'rock' or 'art rock' genres and do not cost more than £7.

Exercise 7

Write a query that lists the *custid* of all customers in the 'Review' table who have rated an album but exclude the customers with *custid* 3 and 8.

Exercise 8

Write a query that uses the NOT BETWEEN operator to list all of the album ratings in the 'Review' table (all attributes) that are either 1 or 5.

Exercise 9

Rewrite the previous query to give the same result but using the IN operator.

Exercise 10

Using the LIKE operator, write a query that lists all of the albums (all attributes) that have the word ‘rock’ somewhere in their genre.

Exercise 11

Write a query that lists only the *custid* numbers of the customers in the ‘Review’ table in ascending order of the *rating* they have given.

Exercise 12

Rewrite the previous query to list the *custid* numbers are given in descending order of the *rating* they have given.

Exercise 13

Write a query that lists the *albumid* numbers and *price* of albums in the ‘Album’ table that cost less than £8 in descending order of price.

Exercise 14

Write a query that lists the *albumid* numbers and *price* of albums in the ‘Album’, ordered by firstly *year* (ascending) and then by *price* (descending). This result should be in one table only.

Exercise 15

Write a query that lists the top three *rating*s given in in reviews in the ‘Review’ table. You don’t need to deal with the case where more than three queries have the same top values. For example, if the top five *rating*s were 5,5,5,5,5 you would still only need to list three of the those *rating*s.

Exercise 16

Write a query that counts the number of distinct genres in the ‘Album’ table.

The keyword DISTINCT is always placed immediately before the attribute name so think about how you will form the COUNT() operation.

Exercise 17

Write a query that counts the number of distinct genres in the ‘Album’ table that have the word ‘rock’ somewhere in the genre.

Exercise 18

Write a query that finds the maximum price of an album on the label ‘RCA’.

Exercise 19

Write a query that lists the year and the average price of albums by year.

You will need a GROUP BY clause.

As a secondary task, order the year groups by year.

Exercise 20

Adapt the above query so that it rounds the averages to 2dp.

Exercise 21

Rewrite the query from Exercise 19 so that only albums that cost more than £8 are included.

Exercise 22

Rewrite the query from Exercise 21 so that only groups having an average price less than £10 are included.