-- Aman Bhatt

-- CSC 355 602T

-- Assignment 3

-- 19 Apr, 2023

-- 1. Give the title, price, and year of all CDs released by Joe Van after 2019, ordered from the newest CD to the oldest CD.

SELECT TITLE, PRICE, YEAR

FROM CD

WHERE ARTIST = 'Joe Van' AND YEAR > 2019

ORDER BY YEAR DESC;

-- 2. Give an alphabetical list of the names of all stores whose name contains ‘CDs’ anywhere in the name.

SELECT NAME

FROM CDSTORE

WHERE NAME LIKE '%CD%'

ORDER BY NAME ASC;

-- 3. For each artist, give the artist’s name and the average price of the CDs released by that artist, ordered from the lowest average price to the highest average price.

SELECT ARTIST, AVG(PRICE)

FROM CD

GROUP BY ARTIST

ORDER BY AVG(PRICE) ASC;

-- 4. Display the number of CDs released in each year, ordered from the latest year to the earliest. (Include the value of the year itself in each line of your output as well.)

SELECT YEAR, COUNT(CID)

FROM CD

GROUP BY YEAR

ORDER BY YEAR DESC;

-- 5. For each CD that has been ordered, give the ID of the CD and the total quantity of that CD that has been ordered. Order the output by the ID of the CD.

SELECT CDID, SUM(QUANTITY)

FROM CDORDER

GROUP BY CDID

ORDER BY CDID ASC;

-- 6. Give the names and cities of all CD stores that placed an order between March 1st, 2023 and March 15th, 2023 (inclusive).

SELECT DISTINCT NAME, CITY

FROM CDSTORE

JOIN CDORDER ON CDSTORE.SID = CDORDER.STOREID

WHERE CDORDER.ORDERDATE BETWEEN DATE '2023-03-01' AND DATE '2023-03-15'

ORDER BY CDSTORE.NAME ASC;

-- 7. List, in alphabetical order, the names of all CD stores that have ordered any CD by ‘Lee Robinson’.

--(Hint: You will need to join all three tables together to solve this problem.)

SELECT DISTINCT CDSTORE.NAME

FROM CDSTORE

JOIN CDORDER ON CDSTORE.SID = CDORDER.STOREID

JOIN CD ON CDORDER.CDID = CD.CID AND CD.ARTIST = 'Lee Robinson'

ORDER BY CDSTORE.NAME ASC;

-- 8. For each CD store, give the name of the store and how many orders it has placed – not the total quantity of the orders,

--just how many separate orders it has placed for any quantity of any CD. Order the output from the store placing the fewest orders to the store placing the most.

--(If a store has not placed any orders yet, you do not have to include it in the output.)

SELECT CDSTORE.NAME, COUNT(CDORDER.ORDERID)

FROM CDSTORE

LEFT JOIN CDORDER ON CDSTORE.SID = CDORDER.STOREID

GROUP BY NAME

HAVING COUNT(CDORDER.ORDERID) > 0

ORDER BY COUNT(CDORDER.ORDERID) ASC;