1. Question :

Details of a category having maximum price for a DVD.

Answer

##Subqueries in a SELECT List

SELECT Category,

(SELECT MAX(DVDPrice) FROM Films WHERE Films.CategoryId =

Category.CategoryId),

CategoryId

FROM Category;

---- **IMPLEMENTATION USING GROUP CONCAT**

SELECT FilmName, PlotSummary, (SELECT MeetingDate FROM Attendance WHERE MemberId = 1) FROM Films;

##Subquery returning more than 1

SELECT group\_concat(meetingDate) FROM Attendance WHERE MemberId = 1 group by MemberId

1. QUESTION

Chose only the cheapest DVD from each category

ANSWER 2)

##You could use a GROUP BY clause, as shown in the following query:

SELECT Category, MIN(DVDPrice)

FROM Category INNER JOIN Films

ON Category.CategoryId = Films.CategoryId

GROUP BY Category;

1. QUESTION

Now add the film name to the list of columns I want

ANSWER 3)

##using joins

SELECT Category, FilmName, MIN(DVDPrice)

FROM Category INNER JOIN Films

ON Category.CategoryId = Films.CategoryId

GROUP BY Category, FilmName;

The results are wrong because they’re grouped by Category and FilmName, so the MIN(DVDPrice) value is not the minimum price for a particular category but rather the minimum price for a particular film in a particular category!

1. QUESTION

List of all the lowest prices for a DVD per category

ANSWER 4)

SELECT Category, FilmName, DVDPrice

FROM Category INNER JOIN Films

ON Category.CategoryId = Films.CategoryId

WHERE Films.DVDPrice =

(SELECT MIN(DVDPrice) FROM Films WHERE

Films.CategoryId = Category.CategoryId)

MULTIPLE VALUES FROM SUBQUERY

1. QUESTION

finds all the members born in 1967, 1992, or 1937:

ANSWER 5)

##Operators:

SELECT FirstName, LastName, YEAR(DateOfBirth)

FROM MemberDetails

WHERE YEAR(DateOfBirth) IN (1967, 1992, 1937)

1. QUESTION

if any members have the same birth year as the release date of a film in the Films table

##ANy and Some operators:

ANSWER 6)

SELECT FirstName, LastName, YEAR(DateOfBirth) FROM MemberDetails

WHERE YEAR(DateOfBirth) = ANY (SELECT YearReleased FROM Films);

**Use of <ALL statement**

1. QUESTION

Select MemberId

from MemberDetails

where MemberId < ALL (Select FilmId from Films where FilmId > 5);

ANSWER 7)

#

Select categories that have rating 4 or higher and selected favourite by 3 or more members)

Select Category

from Category

where exists (select \* from Films

where Category.CategoryId = Films.CategoryId

And Rating > 3

And (Select count(CategoryId)

from FavCategory

Where FavCategory.CategoryId = Category.CategoryId) >=3);

##------------------------------------------------------------------------------

1. QUESTION

Create a list of cities where the average year of birth is later than the average of the membership as a whole.

---------------------------------------------------------------

Using the Having Clause with Subqueries

ANSWER 8)

Select City

From MemberDetails

Group by City

Having Avg(Year(DateOfBirth))> (Select Avg(Year(DateOfBirth)) From MemberDetails);



