Database Task:

Given the following movie database schema, Write out SQL statements for the following 10 queries

# Schema:

**Customers**

|  |  |  |
| --- | --- | --- |
| CustID | LastName | FirstName |

**Inventory**

|  |  |
| --- | --- |
| TapeID | MovieID |

**Movies**

|  |  |
| --- | --- |
| MovieID | MovieName |

**MovieSupplier**

|  |  |  |
| --- | --- | --- |
| SupplierID | MovieID | Price |

**Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| OrderID | SupplierID | MovieID | Copies |

**Rentals**

|  |  |  |  |
| --- | --- | --- | --- |
| CustomerID | TapeID | CkoutDate | Duration |

**Suppliers**

|  |  |
| --- | --- |
| SupplierID | SupplierName |

# Queries:

1. Which movies are supplied by "Joe's House of Video" or "Video Warehouse"?

SELECT MovieName

FROM Movies INNER JOIN MovieSupplier

ON Movies.MovieID = MovieSupplier.MovieID

INNER JOIN Suppliers

ON MovieSupplier.SupplierID = Suppliers.SupplierID

WHERE SupplierName = ‘Joe’s House of Video ’ or

SupplierName= ‘Video Warehouse’

1. Which movie was rented for the longest duration (by any customer)?

SELECT MovieName, Duration

FRON Movies INNER JOIN Inventory

ON Movies.MovieID = Inventory.MovieID

INNER JOIN Rentals

ON Rentals.TapeID = Inventory.TapeID

ORDER BY Duration DESC

LIMIT 1

1. Which suppliers supply all the movies in the inventory? (Hint: first get a list of the movie suppliers and all the movies in the inventory using the cross product. Then find out which of these tuples are invalid.)

SELECT SupplierName, SupplierID

FROM Suppliers

WHERE SupplierID NOT IN (SELECT SupplierID

FROM MovieSupplier

INNER JOIN Inventory

ON MovieSupplier.MovieID

= Inventory.MovieID

WHERE NOT EXISTS( SELECT \*

FROM MovieSupplier

CROSS JOIN Inventory

ON MovieSupplier.MovieID =

Inventory.MovieID ))

1. How many movies in the inventory does each movie supplier supply? That is, for each movie supplier, calculate the number of movies it supplies that also happen to be movies in the inventory.

SELECT SupplierID, COUNT(DISTINCT(MovieID))

FROM Inventory INNER JOIN MovieSupplier

ON Inventory.MovieID = MovieSupplier.MovieID

GROUP BY SupplierID

1. For which movies have more than 4 copies been ordered?

SELECT MovieName

FROM Movies

INNER JOIN Orders

ON Movies.MovieID = Orders.MovieID

GROUP BY MovieName

HAVING COUNT(Copies)>4

1. Which customers rented "Fatal Attraction 1987" or rented a movie supplied by "VWS Video"?

SELECT LastName, FirstName

FROM Customers INNER JOIN Rentals

ON Customers.CustID = Rentals.CustomerID

INNER JOIN Inventory

ON Rentals.TapeID = Inventory.TapeID

INNER JOIN Movies

ON Inventory.MovieID = Movies.MovieID

INNER JOIN MovieSupplier

ON Movies.MovieID = MovieSupplier.MovieID

INNER JOIN Suppliers

ON MovieSupplier.SupplierID = Suppliers.SupplierID

WHERE MovieName = ‘Fetal Attraction 1987’

OR SupplierName = ‘VWS Video’

1. For which movies are there more than 1 copy in our inventory? (Note that the TapeID in inventory is different for different copies of the same MovieID)

SELECT MovieName

FROM Movies

INNER JOIN Orders

ON Movies.MovieID = Orders.MovieID

GROUP BY MovieID

HAVING COUNT (Copies)>1

OR:

SELECT MovieName

FROM Movies

INNER JOIN Inventory AS A

ON Movies.MovieID = A.MovieID

INNER JOIN Inventory AS B

ON Movies.MovieID = B.MovieID

WHERE A.MovieID = B.MovieID AND A.TapeID != B.TapeID

1. Which customers rented movies for 5 days or more?

SELECT FirstName, LastName, CustID

FROM Customers

INNER JOIN Rentals

ON Customers.CustID = Rentals.CustomerID

WHERE Duration>5

1. Which supplier has the cheapest price for the movie "Almost Angels 1962"?

SELECT SupplierName

FROM Suppliers

INNER JOIN MovieSupplier

ON Suppliers.SupplierID = MovieSupplier.SupplierID

INNER JOIN Movies

ON Movies.MovieID = MovieSupplier.MovieID

WHERE MovieName = ‘Almost Angels 1962’

ORDER BY Price ASC

LIMIT 1

1. Which movies aren't in the inventory?

SELECT MovieName

FROM Movies

INNER JOIN Inventory

ON Movies.MovieID != Inventory.MovieID