```
-- Queries
-- Jaden Miguel CSCI330 Summer 2022
-- ------------------------------------------------------------
-- a) Find all distinct track names that start with the letter Z and sort them alphabetically.
SELECT DISTINCT Name FROM Track WHERE Name LIKE 'Z%' ORDER BY Name;

-- b) Find the first name of employees who are older than their supervisors.
-- Compare the BirthDate of the employee with the BirthDate of the Employee's supervisor. (ReportsTo attribute)
-- Sort alphabetically by first name of the employee.
SELECT e.FirstName, e.LastName, e.BirthDate
FROM Employee e
JOIN Employee s ON e.ReportsTo = s.EmployeeId
WHERE e.BirthDate > s.BirthDate
ORDER BY e.FirstName;

-- c) Find the name of the highest-priced track. If more than one track has the highest price, return the names of all such tracks. Sort the output alphabetically based on the track name.
SELECT Name FROM Track
WHERE UnitPrice = (SELECT MAX(UnitPrice) FROM Track)
ORDER BY Name;

-- d) List all customers by ID and last name as well as total amount spent per customer. Sort the output by total spent descending.
--  Include any customers who have never purchased anything.
SELECT c.CustomerId, c.LastName, SUM(il.UnitPrice * il.Quantity) AS TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId, c.LastName
HAVING SUM(il.UnitPrice * il.Quantity) >= 0
ORDER BY TotalSpent DESC;

-- e) Find the highest priced album by adding up the prices of all tracks on the album. Find the highest-priced album.
SELECT a.Title, SUM(t.UnitPrice) AS TotalPrice
FROM Album a
JOIN Track t ON a.AlbumId = t.AlbumId
GROUP BY a.Title
```

```
HAVING SUM(t.UnitPrice) = (SELECT MAX(TotalPrice) FROM (SELECT SUM(t.UnitPrice)
AS TotalPrice
FROM Album a
JOIN Track t ON a.AlbumId = t.AlbumId
GROUP BY a.Title) AS MaxPrice);
```

-- f) Find all albums that have ALL tracks that do not have an invoice line and sort them by album title.

```
SELECT DISTINCT a.Title, a.AlbumId
FROM Album a
JOIN Track t ON a.AlbumId = t.AlbumId
LEFT JOIN InvoiceLine il ON t.TrackId = il.TrackId
WHERE il.InvoiceLineId IS NULL
ORDER BY a.Title;
```

-- g) Create a view that returns customers' first and last names along with corresponding sums of all their invoice totals. Name the view as "CustomerInvoices."

```
CREATE VIEW CustomerInvoices AS
SELECT c.CustomerId, c.FirstName, c.LastName, SUM(il.UnitPrice * il.Quantity) AS
TotalSpent
FROM Customer c
JOIN Invoice i ON c.CustomerId = i.CustomerId
JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
GROUP BY c.CustomerId, c.FirstName, c.LastName;
```

Output screenshots:

a)

Name
Zambação
Zé Trindade
Zeca Violeiro
Zero
ZeroVinteUm
Zither
Zombie Eaters
Zoo Station
Zooropa

b)

FirstName	LastName	BirthDate
Laura	Callahan	1968-01-09 00:00:00
Margaret	Park	1947-09-19 00:00:00
Nancy	Edwards	1958-12-08 00:00:00
Robert	King	1970-05-29 00:00:00

c)

Name
"?"
...And Found
...In Translation
.07%
A Benihana Christmas, Pts. 1 & 2
A Day In the Life

d)

CustomerId	LastName	TotalSpent
6	Holý	49.62
26	Cunningham	47.62
57	Rojas	46.62
45	Kovács	45.62
46	O'Reilly	45.62
28	Barnett	43.62

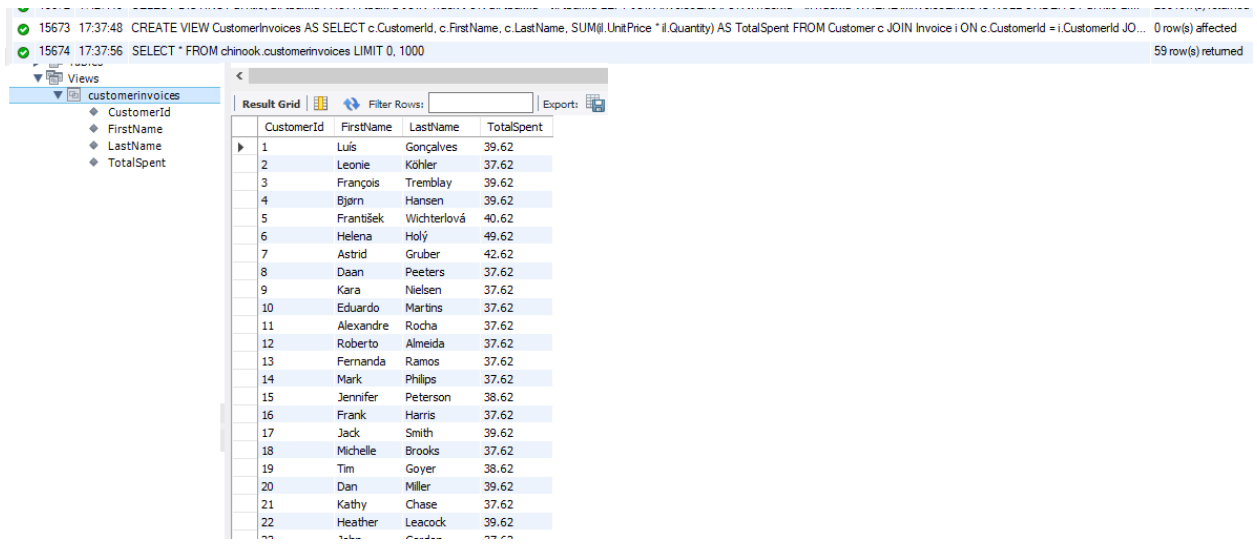
e)

Title	TotalPrice
Greatest Hits	56.43

f)

Title	AlbumId
...And Justice For All	156
20th Century Masters - The Millennium Collectio...	257
A Copland Celebration, Vol. I	296
A Matter of Life and Death	94
A Real Dead One	95
A Real Live One	96

g)



15673 17:37:48 CREATE VIEW CustomerInvoices AS SELECT c.CustomerId, c.FirstName, c.LastName, SUM(il.UnitPrice * il.Quantity) AS TotalSpent FROM Customer c JOIN Invoice i ON c.CustomerId = i.CustomerId JO... 0 row(s) affected

15674 17:37:56 SELECT * FROM chinook.customerinvoices LIMIT 0, 1000 59 row(s) returned

Views
customerinvoices
CustomerId
FirstName
LastName
TotalSpent

Result Grid | Filter Rows: | Export:

CustomerId	FirstName	LastName	TotalSpent
1	Luís	Gonçalves	39.62
2	Leonie	Köhler	37.62
3	François	Tremblay	39.62
4	Bjørn	Hansen	39.62
5	František	Wichterlová	40.62
6	Helena	Holý	49.62
7	Astrid	Gruber	42.62
8	Daan	Peeters	37.62
9	Kara	Nielsen	37.62
10	Eduardo	Martins	37.62
11	Alexandre	Rocha	37.62
12	Roberto	Almeida	37.62
13	Fernanda	Ramos	37.62
14	Mark	Philips	37.62
15	Jennifer	Peterson	38.62
16	Frank	Harris	37.62
17	Jack	Smith	39.62
18	Michelle	Brooks	37.62
19	Tim	Goyer	38.62
20	Dan	Miller	39.62
21	Kathy	Chase	37.62
22	Heather	Leacock	39.62