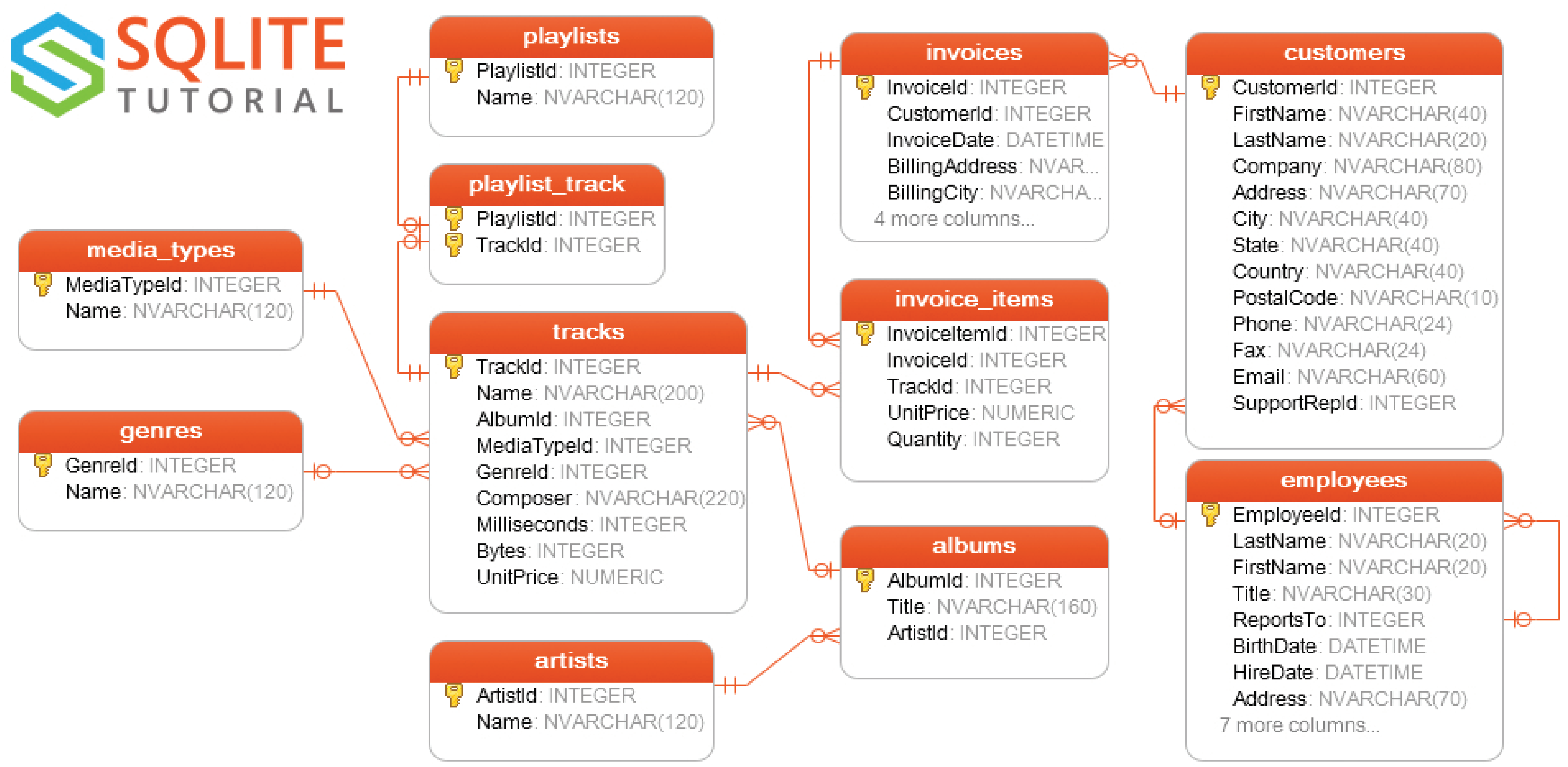
**SQL Lab 2**

We use the chinook.db database from textbook1 to get to know some simple SQL syntaxes.

This is the UML of this database:



**Example 1:**

List the first name, last name, City and Country of customers who has ever purchased Metallica

|  |
| --- |
| --E1  SELECT distinct FirstName, LastName, City, Country  FROM Customers inner join invoices  on Customers.CustomerId = invoices.CustomerId  inner join invoice\_items  on invoices.invoiceId = invoice\_items.invoiceId  inner join tracks  on invoice\_items.TrackId = tracks.TrackId  inner join albums  on albums.AlbumId = tracks.AlbumId  inner join artists  on artists.ArtistId = albums.ArtistId  where artists.Name = 'Metallica'; |

Like

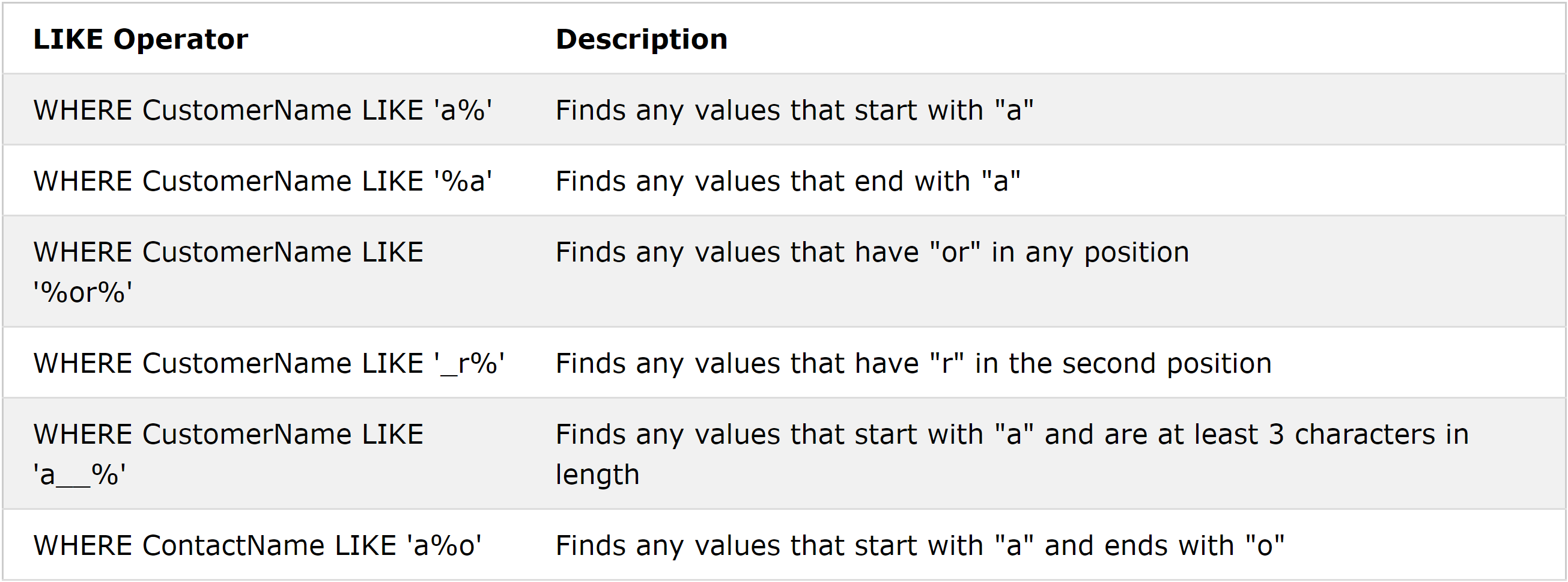
|  |
| --- |
|  |

Like Operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

%: The percent sign represents zero, one, or multiple characters

\_: The underscore represents a single character



Find all the customers whose country’s name starts with I

|  |
| --- |
| --Q1  SELECT \* FROM customers  Where Country like 'I%' |

Find all the customers whose city’s name ends with n

|  |
| --- |
| --Q2  SELECT \* FROM customers  Where City like '%n' |

Find all the tracks whose name include the word love

|  |
| --- |
| --Q3  SELECT \* FROM tracks  Where Name like '%love%'; |

Find all the employees whose birthdate is in month 10

|  |
| --- |
| --Q4  SELECT \* FROM employees  where BirthDate like '\_\_\_\_-09-%'; |

**Example 2:**

Write a query that count the number of customers who live in the 97XXX USA zip code.

|  |
| --- |
| --E2  SELECT Count(CustomerId)  FROM customers  where Country is 'USA' AND PostalCode like '95%'; |

Group By:

Group by groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| --Q5  select Country, Count(CustomerId) as CountryCount  From Customers  Group by Country  Order by CountryCount desc; |

**Example 3:**

List customers name and their total numbers of tracks they have purchased that the tracks have the word love in their title

|  |
| --- |
| --E3  SELECT distinct Customers.CustomerId, FirstName, LastName, Count(tracks.TrackId) as trackCount  FROM Customers inner join invoices  on Customers.CustomerId = invoices.CustomerId  inner join invoice\_items  on invoices.invoiceId = invoice\_items.invoiceId  inner join tracks  on invoice\_items.TrackId = tracks.TrackId  where tracks.Name like '%love%'  Group by Customers.CustomerId  Order by trackCount desc; |

**Example 4:**

Come up with a list that shows all the genres in the database and the number of times a track with a track with each genre has been purchased.

|  |
| --- |
| --E4  select genres.GenreID, genres.Name, count(invoice\_items.InvoiceLineId) as purchaseCount  from genres  inner join tracks  on genres.GenreId = tracks.GenreId  inner join invoice\_items  on invoice\_items.TrackId = tracks.TrackId  Group by genres.GenreID  Order by purchaseCount desc; |

Having Clause:

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

|  |
| --- |
|  |

The countries that have more than 3 customers in the database:

|  |
| --- |
| Select Country, count(CustomerID) as CountryCount  From customers  Group by Country  Having CountryCount > 3  Order by CountryCount Desc; |

**Example 5:**

List all the countries that have more than $50 total amount of sales coming from them.

|  |
| --- |
| --E5  Select Country, sum(UnitPrice\*Quantity) as Sales  From customers join invoices  on customers.CustomerId = invoices.CustomerId  join invoice\_items  on invoices.InvoiceId = invoice\_items.InvoiceId  group by Country  having Sales > 50  Order by Sales Desc |

**Example 6:**

List all customers from countries that has more than 50 total sales.

|  |
| --- |
| --E6  Select FirstName, LastName from customers  Where Country in (  Select Country  From customers join invoices  on customers.CustomerId = invoices.CustomerId  join invoice\_items  on invoices.InvoiceId = invoice\_items.InvoiceId  group by Country  having sum(UnitPrice\*Quantity) > 200  ); |