# SQL Assessment

For the SQL assessment, you will be given a series of prompts to test your knowledge and comprehension. The purpose of this assessment is to apply the SQL skills and fundamentals you have learned. Based on the prompts you will create queries that will return the correct results. Good luck!

To successfully complete this assessment, you will create SQL code based on the following prompts:

1. Create a table called Orders, with the following information:
   1. Primary Key called OrderID
   2. Company Name (include a NOT NULL constraint)
   3. Address
   4. City
   5. Phone
   6. Order Date in DATE format

**CREATE** **TABLE** Orders

Order\_id **Varchar** (255) **PRIMARY** **KEY**,

company\_name **varchar**(255) **NOT** **NULL**,

address **text**,

city **text**,

phone **text**,

order\_date **date**

);

**SELECT**\***FROM** orders

1. Write SQL statements to insert the following into your table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| order\_id | company\_name | address | city | phone | order date |
| 1 | Acme | 14 Hollywood Blvd | Los Angeles | 616-555-1234 | 1/14/15 |
| 2 | Amazon | 2801 S Western Ave | Chicago | 234-345-5151 | 5/2/15 |
| 3 | Netflix | 888 Broadway | New York | 642-612-6123 | 6/7/15 |

**INSERT** **INTO** orders **Values**( '1','Acme', '14 Hollywood Blvd', 'Los Angeles', '616-555-1234', '1/14/15')

**INSERT** **INTO** orders **Values**( '2','Amazon', '28010 S Western Ave', 'Chicago', '234-345-5151', '5/2/15')

**INSERT** **INTO** orders **Values**( '3','Netflix', '888 Broadway', 'New York', '642-612-6123', '6/7/15')

1. Create a query that will select company name, address, and city from the Orders table for companies located in Chicago.

**SELECT** company\_name, address ||', '||city **AS** full\_address

**FROM** orders

**WHERE** city = 'Chicago'

1. Create a query that will select all the records from the Orders table where the company name starts with an "A".

**SELECT** \*

**FROM** orders

**WHERE** company\_name **LIKE** 'A%'

The remaining questions will use the dBeaver Sample Database (music store).

1. Write a query that will list all of the genre names and a count of the tracks for each genre. Sort the list by largest track count to smallest.

**SELECT** g.Name, **count**(t.Name) **AS** tracks\_per\_genre

**FROM** Genre g

**JOIN** Track t **ON** g.GenreId = t.GenreId

**GROUP** **BY** g.Name

**ORDER** **BY** tracks\_per\_genre **DESC**

1. Write a query that will list all of the track names and the album names from the artist named ‘Jamiroquai’.

**SELECT** a2.Name **AS** Artist, t.Name **AS** track\_title, a.Title **AS** album\_title

**FROM** Track t

**JOIN** Album a **ON** t.AlbumId =a.AlbumId

**JOIN** Artist a2 **ON** a2.ArtistId = a.ArtistId

**WHERE** Artist ='Jamiroquai'

1. Write a query that will determine the top 5 countries measured by total revenue (dollars) sold by billing country. Include country and total revenue.

**SELECT** BillingCountry, **sum**(total) **AS** country\_total

**FROM** Invoice i

**GROUP** **BY** BillingCountry

**ORDER** **BY** country\_total **DESC**

**LIMIT** 5

1. Write a query that determines the total sales by global region. Use ‘Asia Pacific’ for India and Australia, ‘North America’ for Canada and the USA, ‘South America’ for Chile, Brazil and Argentina, and ‘Europe’ for the rest.

**SELECT** **sum**(total) **AS** regional\_sales,

**CASE**

**WHEN** BillingCountry **IN** ('India', 'Australia')

**THEN** 'Asia Pacific'

**WHEN** BillingCountry **IN** ('Canada', 'USA')

**THEN** 'North America'

**WHEN** BillingCountry **IN** ('Chile','Brazil','Argentina')

**THEN** 'South America'

**ELSE** 'Europe'

**END** **AS** Regions

**FROM** Invoice i

**GROUP** **BY** regions

**ORDER** **BY** regional\_sales **DESC**

1. Write a query that lists the artists that don’t have albums.

**SELECT** a.Name **AS** artist, **count**(a2.AlbumId) **AS** album\_count

**FROM** Artist a

**JOIN** Album a2 **ON** a.ArtistId =a2.ArtistId

**GROUP** **BY** artist

**HAVING** album\_count < 1

1. Write a query that lists all the invoice amounts, their billing country, and total amount of revenue from that country. Sort largest to smallest.

**WITH** total\_country\_revenue **AS**

(

**SELECT** InvoiceId, BillingCountry, total **AS** invoice\_total, **sum**(Total) **AS** country\_total

**FROM** Invoice i

**GROUP** **BY** InvoiceId

**ORDER** **BY** BillingCountry

)

**SELECT** InvoiceId, BillingCountry, invoice\_total, **sum**(country\_total) **AS** country\_revenue

**FROM** total\_country\_revenue

**GROUP** **BY** BillingCountry

**ORDER** **BY** country\_revenue **DESC**