Retrieving data with SQL

# Introduction

For this activity you will need:

* DB Browser for SQLite
* A **copy** of the dbMusic.db file which you used last lesson.

# Operators A list of SQL operators

|  |  |
| --- | --- |
| **Comparison operator** | **Function** |
| = | Equal to |
| > | Greater than |
| < | Less than |
| => | Greater than or equal to |
| <= | Less than or equal to |
| <> | Not equal to |

|  |  |  |
| --- | --- | --- |
| **Logical operator** | **Function** | **Example** |
| AND | TRUE if both conditions are TRUE | SELECT \*  FROM tblTracks  Genre = "Pop" AND Artist = "The Springs"; |
| BETWEEN | TRUE if the range is within the comparisons | SELECT \*  FROM tblDownloads  WHERE Time BETWEEN "06:00" AND "07:00"; |
| OR | TRUE if any of the conditions are TRUE | SELECT \*  FROM tblTracks  WHERE Genre = "Pop" OR Genre = "Rock"; |

# Code snippets .

**Select all of the columns and records from the** tblTracks **table**

|  |  |
| --- | --- |
| 1  2 | SELECT \*  FROM tblTracks; |

**Select the** Title **and** Artist **columns from the** tblTracks **table where the** Genre **is not equal to** Pop**.**

|  |  |
| --- | --- |
| 1  2  3 | SELECT Title, Artist  FROM tblTracks  WHERE Genre <> "Pop"; |

# Task Retrieve data from the tblTracks table

For each question, design an SQL search that will reveal the answer.

**Question 1:** How many tracks in the database have the genre Rock?

|  |  |
| --- | --- |
| **Answer** | 14 |
| **Code** | SELECT Title, Artist  FROM tblTracks  WHERE Genre = "Rock"; |

**Question 2:** How many tracks in the database are created by the artist A Box of Spoons?

|  |  |
| --- | --- |
| **Answer** | 3 |
| **Code** | SELECT Title, Artist  FROM tblTracks  WHERE Artist = "A Box of Spoons"; |

**Question 3:** How many tracks in the database have a TrackID that is greater than 30 and are of the “Soul” genre?

|  |  |
| --- | --- |
| **Answer** | 4 |
| **Code** | SELECT Title, Artist  FROM tblTracks  WHERE Genre = "Soul" AND TrackID > 30; |

# Task Retrieve data from the tblDownloads table

For each question, design an SQL search that will reveal the answer.

**Question 1:** How many downloads were made in 2011?

|  |  |
| --- | --- |
| **Answer** | 23 |
| **Code** | SELECT Date  FROM tblDownloads  WHERE Date BETWEEN "2011-01-01" AND "2011-12-31"; |

**Question 2:** How many tracks were either downloaded before 6AM or after 10PM?

|  |  |
| --- | --- |
| **Answer** | 0 |
| **Code** | SELECT Time  FROM tblDownloads  WHERE Time BETWEEN "22:00" AND "06:00"; |

# Task Using ORDER BY

You can change the order of the retrieved records by sorting the data into ascending or descending order. The following steps will show you how this works.

**Step 1**

Take a look at the code snippet below. What do you think will happen when the data is retrieved?

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM tblMembers  ORDER BY Surname ASC; |

**Write your prediction below:**

|  |
| --- |
| TU RESPUESTA  TU RESPUESTA |

**Step 2**

Copy the code into DB Browser and execute it. Was your prediction correct?

|  |  |
| --- | --- |
| **Answer** | TU RESPUESTA  TU RESPUESTA |

**Step 3**

Change the final three letters from ASC to DESC.

|  |  |
| --- | --- |
| 1  2  3 | SELECT \*  FROM tblMembers  ORDER BY Surname DESC; |

What changed compared to when you used ASC?

|  |  |
| --- | --- |
| **Answer** | TU RESPUESTA  TU RESPUESTA |

# Task Sorting retrieved data

For each question, design an SQL search that will reveal the answer.

**Question 1:** What is the TrackID of the first ever download recorded in the database?

|  |  |
| --- | --- |
| **Answer** | 17 |
| **Code** | SELECT \*  FROM tblDownloads  ORDER BY date ASC; |

**Question 2:** What is the earliest time of day that a track has been downloaded?

|  |  |
| --- | --- |
| **Answer** | 00:03 |
| **Code** | SELECT \*  FROM tblDownloads  ORDER BY time ASC; |

# Explorer task .

Design and execute your own SQL searches. They should use at least 1 comparison operator and 1 logical operator. Copy and paste your code in the space below