

Reading. Read the WEEK #7 and #8 chapter in ZyBooks and complete all of the corresponding reading “*activities*”. Note that only readings that begin with WEEK #*n* are ready for reading.

Technical Work. Use SQL to write the following queries against the tables you created and populated for your Music database in HW 4. You may need to modify the design of your tables and will likely need to add more rows to each table so that your queries below return results. When multiple rows can be returned for a query, you should have enough data so that your answers contain at least 3 rows (some queries will only ever return at most one row).

For each of the queries below, pick specific values to use from your tables in place of a “specific year”, a “specific music group”, and so on. *Your queries must be written to be generic, i.e., they should work with any possible instance and not just for the particular example instances you define* (assuming the specific values mentioned above are present). When writing your queries, be careful in terms of attributes that can have NULL values. Finally, you *may not use subqueries* in any of the queries below, and you *may not use any techniques we haven’t covered in class*.

Note that you must create a `hw5.sql` script file that contains your queries. You must also include comments in your script file, including a header file comment (with your name, the course, assignment number, and brief description) as well as comments for each query as needed. As part of your query comments include the problem number as well. You must also include the contents of each table in your database and the results of running your queries against your database as a separate file. If you are unsure what is being asked for below, please ask for clarification on Piazza. Submit your `hw5.sql` file to the GitHub classroom for HW-5 along with a print-out showing your database tables and the results of running your queries (a link will be provided in Piazza).

1. Find all album titles recorded in a specific year.
2. Find the names of all record labels that were founded in a specific year with a specific label type.

3. Find the names of all record labels that have one of two specific types and that was founded in the 1980's.
4. Find the names of all artists that have been members of a specific music group.
5. Find the names of all artists that have been members of at least two different music groups.
6. Find the name and year formed of each music group that plays at least three different genres of music.
7. Find the name of each music group that was influenced by a specific music group.
8. Find all of the album titles recorded within a specific range of years and whose corresponding group contained a specific artist that was a member of the group during the same range of years.
9. Find all of the artists that were members of groups that played a specific genre of music and that were members of the group within a specific range of years (e.g., in the 2000s). Return the artist name and group name.
10. Find all of the tracks that were on at least two albums, but where two of the albums were associated with different record labels and different music groups.
11. Find each music group that is influenced by another group such that both groups have albums that share a song title. Return the name of each group and the song title they share.
12. Find the artists that are either in at least two groups or wrote at least two songs, but not necessarily both. Note that you should have at least one artist in your database that didn't do either of these. Return the unique set of artist names.
13. Find the first and last year a specific group created albums. Although not part of the query itself, select a group that is associated with more than two albums.
14. Find the total number of albums that were recorded by a group for a specific record label.
15. Develop an "interesting" query involving joins and multiple WHERE conditions. Explain in plain English what your query returns and show the results of the query.