

Lab 02: CD (basic)

1 Objective

After completing this assignment you should feel a bit more comfortable using the command-line PostgreSQL client and the Linux shell, converting an Entity-Relationship diagram into a set of tables, and writing more complicated SQL statements.

2 Setup

You will need the same two pieces of software as in the previous lab: a text editor and an SSH client.

You will need two files to start:

`Ch4CDinserts.sql` contains SQL instructions to insert rows into tables about record labels, albums, and tracks. You will use this in Section 3.

`handin.sql` is where you will record all of your answers, and is the file that you will submit when you are finished.

You can find both files at `/home/faculty/chogg/Public/366/a02cdbasic` on the server, but will also probably want to download an archive file containing both of them directly from AutoLab so that you work with them on your local computer.

It is very important that you **NOT** change the format of the `handin.sql` file and put your answers in between the appropriate “Begin” and “End” sections, as otherwise the autograder will not be able to find your answers. Please add your name to the comment at the beginning.

3 Assignment

For each question below, you are going to write your SQL code into your text file in the appropriate section. You will then use your SSH client to run that code in a database. For this assignment, your database will be named `a02cdbasic_username`, with `username` replaced by your actual username.

We are going to be creating a database about CDs, the tracks that are found on them, and the record labels that release them. Figure 1 shows an Entity-Relationship diagram for the desired database. It was created in Visio and may look somewhat different from other E-R diagrams you have seen. If you are confused about the meaning of any part of it, please ask for clarification.

1. (15 points) Write and execute statements that will create all tables needed for this database. Name the tables exactly the same as the entity sets, and the columns exactly the same as the attributes except that all names should be purely lowercase. List the columns in the same order as the attributes are listed in the diagram, and follow the convention that foreign key columns are added at the end. You can derive appropriate types for the columns by looking what data will be inserted by the statements in `Ch4CDinserts.sql`. After you have created your tables, run the code from that file to populate them. To do so, type `\include`

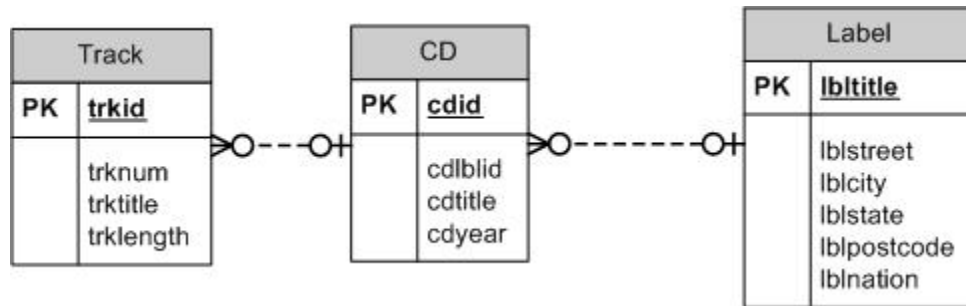


Figure 1: Entity-Relationship Diagram

/home/faculty/chogg/Public/366/a02cdbasic/Ch4CDinsert.sql. If any insertions fail, you will need to drop your tables, fix your table creation code, and then re-run it and re-run the insertion code.

2. (10 points) Write a query to retrieve the track title, CD title, and track length of all tracks, ordered from earliest (alphabetically) CD title to latest CD title, with ties broken shortest length to longest length, then further ties broken earliest (alphabetically) track title to latest track title.
3. (10 points) Write a query to retrieve the track title and track length of all tracks from the CD titled "Swing". Use a subquery instead of a join (just to demonstrate that you can).
4. (15 points) Write a query to retrieve the CD title, track title, and track length of (only) the longest track on each CD. If multiple tracks are tied for longest on a CD (which does not occur in the sample data) each of them would be retrieved.
5. (10 points) Write a query to retrieve the CD title, number of tracks on the CD, and total length of the CD for each CD. Show CDs with more tracks before CDs with fewer tracks, and do not attempt to break ties.
6. (10 points) Write a query to retrieve the label title, label nation, CD title, and total CD length of all CDs that are more than 40 minutes long. Name the total CD length column as **cdlength**.
7. (10 points) Write a query to retrieve the CD title, track title, and track length of the three shortest tracks, with shorter tracks before longer tracks. To accomplish this, you may want to look up the **LIMIT** SQL feature, which we did not discuss in class.
8. (10 points) Write an SQL statement that creates a view named **CDView** that contains columns **cdid**, **cdlblid**, **cdtitle**, **cdyear**, and **cdlength**. (The first four are columns that exist in various tables, while the fifth must be calculated.)
9. (10 points) Write a query to retrieve the track title, track length, and CD title of all tracks whose titles start with 'C'.

4 Submission

Please submit your completed `handin.sql` file to autolab. You can submit as many times as you like.