Consider the Entity-Relationship diagram distributed with this lab. NOTE: the + sign with a crow’s foot provides information about the identifier of the table. For example, a person-composition can be uniquely identified by the combination of psnid, compid and psncomprole. IMPORTANT: if present, ignore the + sign on the crow’s foot by CD – the lbltitle is not part of the primary key of cd in my insert statements (should be whited out...).

You’re again going to turn in a text file for this lab (submit it as lab3). Be sure to include both the SQL statements and the results for each query. You may use either MySQL or SQLServer.

Note that there are some differences in the SQL that can be accepted here because of differences in the data types in the two databases. I’ve ironed out the differences in the INSERT statements, but you need to be careful to choose appropriate types for the database you’re working in. For example, the rcddate field should be of type “datetime” for SQLServer and just “date” for MySQL.

1. (30 pts) Write and execute SQL statements to create the tables described above. Remember that you will need foreign key constraints to capture the one- and many-to-many relationships. You may need to drop your existing track, label, and cd tables. You can refer to the insert statements for more information on data types. The URL for the insert statements is: http://www.cs.millersville.edu/~schwartz/466/Labs/Chap5CDinserts.txt. The provided insert statements will work if you’ve constructed the tables as shown (with columns created in the same order they appear).

**It’s going to be trickier to use the insert statements this time because the foreign keys aren’t shown as columns in the data model. Assume that the foreign keys are always created as the last columns. If there is an association table like person-composition, assume that the primary key from the person table is the first foreign key column and the primary key to composition is last. The exception to all of this is track! The order of columns assumed in the track table is cdid, trknum, rcdid, compid.**

2. (10 pts) Write a SQL query to list the track number and composition title of all of the tracks on Giant Steps.

3. (15 pts) Write a SQL query to list the first and last names and the instruments they played for the May 4, 1959 recording of ‘Giant Steps’. Note: This could take some investigation into how the date or datetime fields of your database work, especially for SQLServer. For example, in SQLServer, you will probably want to check out the datepart function.

4. (15 pts) Write a SQL query to list the composers who write music and play tenor sax. Each person should only be listed once in the results.

5. (15 pts) List the name, track number and cdtitle of any compositions where the same composition appears as multiple tracks on a CD. Sort the data by composition title and track number. Note that this should include two different recordings of the same composition (so Countdown should be in your results).

6. (15 pts) Write a SQL query to list the recording id and recording date of all recordings that appear on all of the cds in the cd table. Write the query twice, once using EXISTS and once without EXISTS.