Laboratory 6: intermediate MySQL

This set of exercises is designed to build on laboratory 5 by using the same data set (Nakamura et al. 2000) stored in a more conventionally designed relational database. The monolithic table has been replaced by four tables: (1) ‘Sequences’ for storage of sequence metadata; (2) ‘Taxa’ for storage of taxonomic information; (3) ‘Codons’ for storage of codon frequencies; and (4) ‘AminoAcids’ for translation of codons to/from amino acids. The design of the database will require variations on the JOIN syntax to be used for most queries. As a result of their GenBank origin, some entries are still idiosyncratic, but an effort has been made to standardize the data (a small number of records were deleted in the process).

Tasks

1. Download and install the database on your computer.
   
   (a) To get the multipart archive (6 files, a result of a 10 MB limit imposed on NYBG’s web server), use the following terminal commands to download the files one at a time:
   
   ```
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.00
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.01
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.02
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.03
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.04
wget http://www.nybg.org/files/scientists/dlittle/codon.tar.xz.05
```
   
   (b) Verify that the data have not been corrupted by typing `cat codon.tar.xz.* | tar xJof - | sha512sum` in the terminal. The resulting hash should be: 0199068622c6f8162b6e9804ee848a36ccde9c57. Answer question [1].
   
   (c) To recreate the database on your computer without using disk space unnecessarily, several operations will be required. First extract and build the database structure one table at a time:
   
   i. Type `cat codon.tar.xz.* | tar xJof - AminoAcids.sql | mysql -u working -p codon` in the terminal to add the AminoAcids table. Type your password when requested.
   
   ii. Type `cat codon.tar.xz.* | tar xJof - Taxa.sql | mysql -u working -p codon` in the terminal to add the Taxa table. Type your password when requested.
   
   iii. Type `cat codon.tar.xz.* | tar xJof - Sequences.sql | mysql -u working -p codon` in the terminal to add the Sequences table. Type your password when requested.
   
   iv. Type `cat codon.tar.xz.* | tar xJof - Codons.sql | mysql -u working -p codon` in the terminal to add the Codons table. Type your password when requested.
   
   (d) Likewise extract and insert the database data from the archive one table at a time:
   
   i. Type `cat codon.tar.xz.* | tar xJof - AminoAcids.txt | mysql --local-infile -u working -p codon` in the terminal to add the AminoAcids table. Type your password when requested.
   
   ii. Type `cat codon.tar.xz.* | tar xJof - Taxa.txt | mysql --local-infile -u working -p codon` in the terminal to add the Taxa table. Type your password when requested.
iii. Type `cat codon.tar.xz.* | tar xJOf - Sequences.txt | mysql --local-infile -u working -p -e "LOAD DATA LOCAL INFILE '/dev/stdin' INTO TABLE Sequences` codon in the terminal to add the Sequences table. Type your password when requested.

iv. Type `cat codon.tar.xz.* | tar xJOf - Codons.txt | mysql --local-infile -u working -p -e "LOAD DATA LOCAL INFILE '/dev/stdin' INTO TABLE Codons` codon in the terminal to add the Codons table. Type your password when requested.

2. Explore the new database.

   (a) Login to mysql using the ‘working’ user.
   
   (b) Change to the codon database.

   (c) Type `SHOW CREATE TABLE Sequences;` in the terminal to review the new sequence metadata table structure. Answer question 2.

   (d) Type `SHOW CREATE TABLE Taxa;` in the terminal to review the new taxon table structure. Answer question 3.

   (e) Type `SHOW CREATE TABLE Codons;` in the terminal to review the new codon frequencies table structure. Answer question 4.

   (f) Type `SHOW CREATE TABLE AminoAcids;` in the terminal to review the new amino acid/codon translation table structure. Answer question 5.

3. Recreate the laboratory 5 queries using the new table structure.

   (a) To determine the number of ‘species’ represented in the database type `SELECT DISTINCT Genus, SpecificEpithet FROM Taxa;` in the terminal. Answer question 6.

   (b) Determine the number of ‘species’ that are represented by nuclear sequences by typing `SELECT DISTINCT Genus, SpecificEpithet, Genome FROM Taxa, Sequences WHERE Taxa.TaxonID = Sequences.Taxon AND Genome = 'nuclear';` in the terminal. Answer question 7.

   (c) Determine the contribution by each ‘species’ by typing `SELECT Genus, SpecificEpithet, COUNT(SequenceID) AS Sequences FROM Taxa, Sequences WHERE Taxa.TaxonID = Sequences.Taxon GROUP BY Genus, SpecificEpithet ORDER BY Sequences ASC;` in the terminal. Answer question 8.

   (d) Find the total and average number of reports for each of the 64 codons in the database. Type `SELECT AminoAcids.Codon, SUM(Frequency) AS Sum, AVG(Frequency) AS Mean, STD(Frequency) AS STD FROM Codons, AminoAcids WHERE AminoAcids.AminoAcidID = Codons.Codon GROUP BY Codon;` in the terminal to get the sum, mean, and standard deviation. Answer question 9.

   (e) Find the total number of reports for each amino acid in the database. Type `SELECT AminoAcids.AminoAcid, AminoAcids.Codon, SUM(Frequency) AS Sum, AVG(Frequency) AS Mean, STD(Frequency) AS STD FROM Codons, AminoAcids WHERE Codons.Codon = AminoAcids.AminoAcidID GROUP BY AminoAcid, AminoAcids.Codon WITH ROLLUP;` in the terminal to get the sum, mean, and standard deviation of occurrences by codon and amino acid. Answer question 10.
Questions

1. For task 1b, explain what ‘shasum’ does. Would it be more or less secure to use a faster algorithm such as ‘md5sum’? What about an algorithm that outputs a longer hash such as ‘sha512sum’?

2. For task 2c, why is the field ‘Genome’ stored as an ENUM? Why not a SET or CHAR? Does the ‘Description’ field hold all descriptions in their entirety? Is this table fully normalized? Why or why not? Can a VARCHAR field in the table ‘Sequences’ store the characters ‘a’, ‘á’, ‘ā’ and ‘à’? How can you tell?

3. For task 2d, how are taxon names linked (related) to sequences? What happens if a row from the ‘Taxa’ table is deleted? Updated? Is the ‘Taxa’ table fully normalized? What prevents duplicate rows from being inserted?

4. For task 2e, if a given sequence had 35 different codons how many entries (rows) would it have in the ‘Codons’ table? What if it had 20 different codons? Instead of using a separate table for codon type, could one use an ENUM field? Would an ENUM field make queries easier or harder?

5. For task 2f, there are no keys other than the primary key for the ‘AminoAcids’ table. It is likely that queries will join the ‘Codon’ field and/or the ‘AminoAcid’ field from this table to the ‘Codons’ table. What kind of keys are used in such queries? Could you change the field types for greater speed?

6. For task 3a, how many records are retrieved? Does this number reflect the actual number of species? How does this number compare to the answer from laboratory 5? What could explain the difference?

7. For task 3b, what kind of join was used to make the query? How many records are retrieved? Does this number reflect the actual number of species with nuclear sequences? Do any of these species also have plastid sequences? How would you modify the query to find only the species represented by at least one plastid sequence and at least one nuclear sequence?

8. For task 3c, why is the GROUP BY statement needed? What does the ORDER BY statement do? What are the top 5 contributing species? How many species are represented by more than ten sequences? What query did you use to determine that value?

9. For task 3d, why is ‘AminoAcids.Codon’ specified in the query rather than just ‘Codon’? What is the most frequently used codon? The least frequent?

10. For task 3e, what kind of join is used to make the query? What is the most widely used amino acid? Are all of its codons used equally?

Literature cited


Due at the start of class February 13.