Databases for Biologists

Session 3
Building And Modifying A Database With SQL

Robert Latek, Ph.D.
Sr. Bioinformatics Scientist
Whitehead Institute for Biomedical Research
Session 3 Outline

• SQL Query Review
• Creating Databases
• Creating Tables
• Altering Table Structure
• Inserting Data
• Deleting Data
• Updating/Modifying Data
• Automating Repetitive Tasks
Connecting To MySQL

• If No Local MySQL, In Terminal Window
  – % ssh hebrides.wi.mit.edu -l username

• Connect to MySQL Database Server
  – % mysql -u username -p -D db4bio
  – mysql>

• SQL Commands Are Case-Insensitive
• Tables And Attributes Are Case-Sensitive
### SELECT

```sql
> SELECT *
FROM Data
LIMIT5;
```

### SELECT DISTINCT species

```sql
> SELECT DISTINCT species
FROM LocusDescr;
```

<table>
<thead>
<tr>
<th>affyId</th>
<th>exptId</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFFX-MurIL2_at</td>
<td>hs-cer-1</td>
<td>20</td>
</tr>
<tr>
<td>AFFX-MurIL10_at</td>
<td>hs-cer-1</td>
<td>8</td>
</tr>
<tr>
<td>AFFX-MurIL4_at</td>
<td>hs-cer-1</td>
<td>77</td>
</tr>
<tr>
<td>AFFX-MurFAS_at</td>
<td>hs-cer-1</td>
<td>30</td>
</tr>
<tr>
<td>AFFX-BioB-5_at</td>
<td>hs-cer-1</td>
<td>258</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>species</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hs</td>
</tr>
<tr>
<td>Mm</td>
</tr>
</tbody>
</table>
WHERE And ORDER BY

> SELECT *
   FROM RefSeqs
   WHERE linkId BETWEEN 50 AND 100
   LIMIT 5;

> SELECT *
   FROM RefSeqs
   WHERE linkId BETWEEN 50 AND 100
   ORDER BY ntRefSeq DESC
   LIMIT 5;

<table>
<thead>
<tr>
<th>linkId</th>
<th>ntRefSeq</th>
<th>aaRefSeq</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>NM_001098</td>
<td>NP_001089</td>
</tr>
<tr>
<td>51</td>
<td>NM_004035</td>
<td>NP_004026</td>
</tr>
<tr>
<td>52</td>
<td>NM_004300</td>
<td>NP_004291</td>
</tr>
<tr>
<td>53</td>
<td>NM_001610</td>
<td>NP_001601</td>
</tr>
<tr>
<td>54</td>
<td>NM_001611</td>
<td>NP_001602</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>linkId</th>
<th>ntRefSeq</th>
<th>aaRefSeq</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>NM_005159</td>
<td>NP_005150</td>
</tr>
<tr>
<td>81</td>
<td>NM_004924</td>
<td>NP_004915</td>
</tr>
<tr>
<td>91</td>
<td>NM_004302</td>
<td>NP_004293</td>
</tr>
<tr>
<td>86</td>
<td>NM_004301</td>
<td>NP_004292</td>
</tr>
<tr>
<td>52</td>
<td>NM_004300</td>
<td>NP_004291</td>
</tr>
</tbody>
</table>
GROUP BY And HAVING

> SELECT *
  FROM Data
  GROUP BY affyId
  HAVING level < AVG(level)
  LIMIT 5;

> SELECT uId
  FROM UniSeqs
  GROUP BY uId
  HAVING count(gbId) > 1
  LIMIT 5;
Table Joining

```sql
> SELECT DISTINCT Unigenes.uId, GO_Descr.description AS GO_description
  FROM Unigenes, LocusLinks, Ontologies, GO_Descr
  WHERE Unigenes.linkId=LocusLinks.linkId
  AND LocusLinks.linkId=Ontologies.linkId
  AND Ontologies.goAcc=GO_Descr.goAcc
  LIMIT 5;
```

<table>
<thead>
<tr>
<th>uId</th>
<th>GO_description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hs.373554</td>
<td>calcium ion binding</td>
</tr>
<tr>
<td>Hs.74561</td>
<td>protein carrier</td>
</tr>
<tr>
<td>Hs.155956</td>
<td>arylamine N-acetyltransferase</td>
</tr>
<tr>
<td>Hs.2</td>
<td>arylamine N-acetyltransferase</td>
</tr>
<tr>
<td>Hs.234726</td>
<td>serine protease inhibitor</td>
</tr>
</tbody>
</table>
CREATE DATABASE

• Allows You To Create A New Database On The Database Server

> SHOW DATABASES;
> CREATE DATABASE mfdb;
> SHOW DATABASES;
> USE mfdb;
Access Privileges

• Restrict Access And Prevent Accidental Alteration Of Important Information
• Can Limit What Individual Users Can See And Do On Particular Databases And Specific Tables
• Access Privileges Are Stored In The mysql Database

> GRANT ALL PRIVILEGES ON db4bio.* TO superuser@"%" IDENTIFIED BY "password";
> GRANT SELECT,INSERT ON db4bio.Data TO admin@"18.157.*.*" IDENTIFIED BY "pass2";
CREATE TABLE

• Translate An E-R Diagram (Schema) Into a Functioning Database

> CREATE TABLE Descriptions (  
  gbId VARCHAR(20) NOT NULL,  
  description VARCHAR(100),  
  PRIMARY KEY (gbId)  
);
> CREATE TABLE Targets ( 
  affyId VARCHAR(20) NOT NULL, 
  gbId VARCHAR(20) NOT NULL, 
  species VARCHAR(20), 
  PRIMARY KEY (affyId, gbId) 
); 

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>affyId</td>
<td>varchar(20)</td>
<td></td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gbId</td>
<td>varchar(20)</td>
<td></td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>species</td>
<td>varchar(20)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>
ALTER TABLE

• Modify A Table’s Attributes
  – Attribute Names, Type, Null, Key, Default
  – Add Or Drop Attributes

> ALTER TABLE Data
  CHANGE level level DOUBLE;

> ALTER TABLE Data
  RENAME level expression;

> ALTER TABLE Data
  ADD PRIMARY KEY (exptId);

> ALTER TABLE Data
  DROP COLUMN affyId;

> ALTER TABLE Data
  ADD date TIMESTAMP;

> DROP TABLE Data;
• Finally, Add Data Into Tables

> INSERT INTO Data (level, exptId, affyId) VALUES (215, “hs-hrt-1”, “100008_at”);

> INSERT INTO Data VALUES ("100008_at", "hs-hrt-1", 215);

> INSERT INTO Data2 (affyId2,level2) SELECT Data.affyId, Data.level FROM Data WHERE Data.level < 250;
DELETE FROM

• Delete Data From Tables
• Similar Syntax As SELECT

> DELETE FROM Data
   WHERE exptId="hs-hrt-1";

> DELETE FROM Sources
   WHERE exptId= "hs-hrt-1";

BE CONSISTENT
UPDATE

• Modify Data Already Stored In A Table
• Again, Similar Syntax As SELECT

> UPDATE Data
  SET exptId="hs-hrt-2"
  WHERE exptId="hs-hrt-1";

> UPDATE Source
  SET exptId= "ms-hrt-1", source="Mm"
  WHERE exptId="hs-hrt-1";

> UPDATE Data
  SET level=level*2
  WHERE exptId="hs-hrt-1";
LOAD DATA And Export

• Read Rows From A Text File Into A Table And Vice Versa

> LOAD DATA LOCAL INFILE "data.txt"
  INTO TABLE db4bio.Data
  FIELDS TERMINATED BY "\t" ENCLOSED BY "" ESCAPED BY"\"\" (ASSUMED)
  LINES TERMINATED BY "\n" (ASSUMED)

> LOAD DATA LOCAL INFILE "data.txt"
  INTO TABLE db4bio.Data
  FIELDS TERMINATED BY ",'";

> SELECT * INTO OUTFILE "data.txt"
  FIELDS TERMINATED BY ",'"
  FROM Data;
Automating Repetitive Tasks

• Use .SQL Files To Perform SQL Commands Automatically

% mysql -h hebrides.wi.mit.edu -u guest -p -D databasename < create.sql

• Automatically Create A Series Of Tables

• Feed A Complicated Query To The Database And Receive The Results In A Text File

% mysql -h hebrides.wi.mit.edu -u web -p -D db4bio < query1.sql > query1.out
Summary

- Design Databases With E-R Diagrams
- Data Mine Using Combinations Of SELECT/FROM With WHERE, GROUP BY, HAVING, ORDER BY, And Aggregates
- Create And Implement Databases
- Input and Output Data From Databases
- Modify Existing Data Within Databases
Where To Go From Here?

• Consult SQL And MySQL Resources
  – http://www.mysql.com
  – http://neo.bu.edu/be768/2003Class/

• Database Tools
  – VisualCase2 (draw E-R diagrams)
  – Data Architect (draw E-R diagrams)
  – Visio (PC - draw E-R diagrams & DB Administration)
  – SQL4XManagerJ (DB Administration)
Exercises

• Create Tables
• Input Data
• Modify/Delete Particular Data

• Accessing Your Database:
  – mysql -u username -p -D username