Reading mySQL

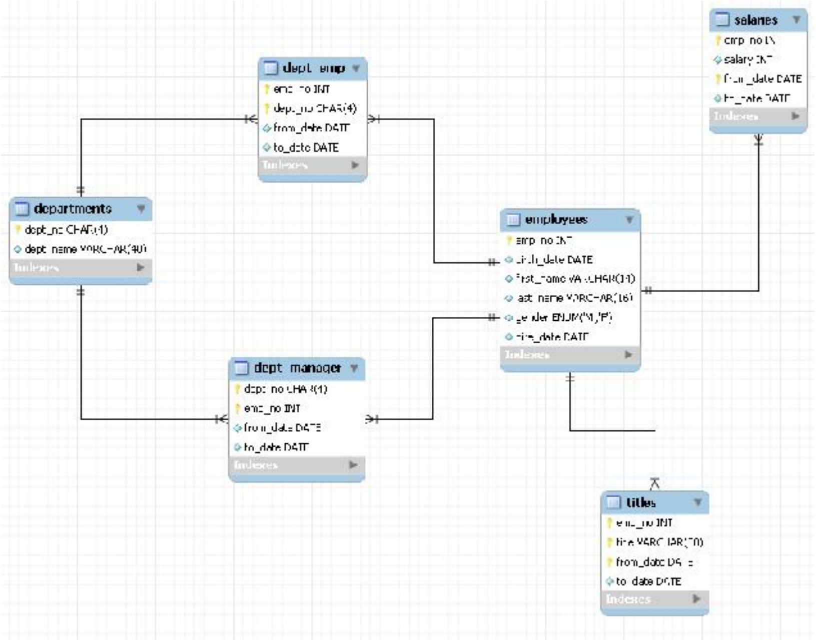
Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

mySQL

- Free and widely used open source database software
- Widely used in internet based applications
- Data are structured in
  - Databases
  - Tables within databases
  - Fields within tables
- Each row is called a record

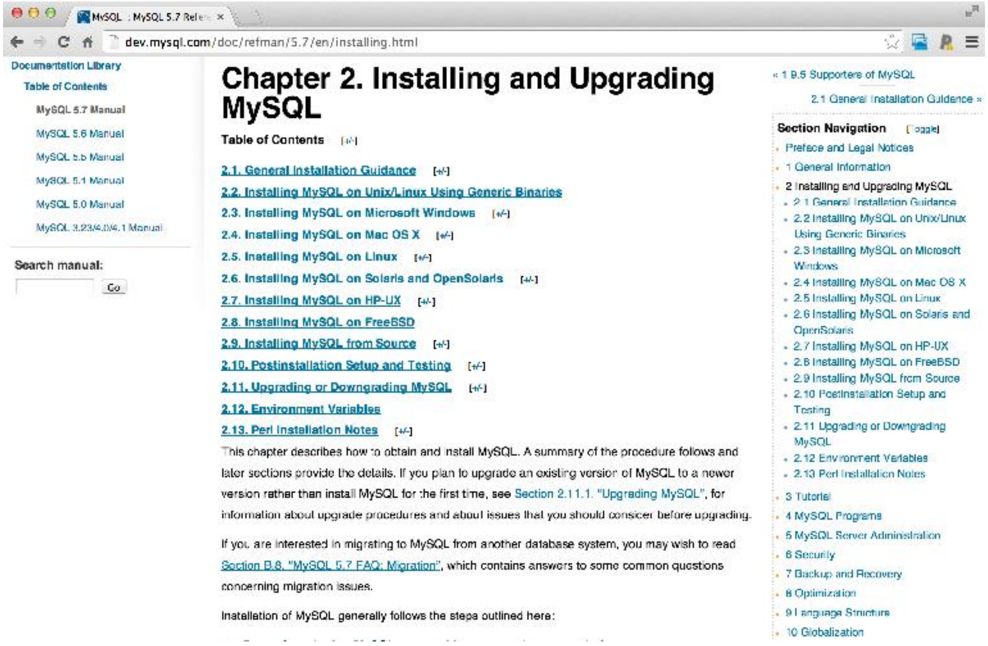
http://en.wikipedia.org/wiki/MySQL http://www.mysql.com/

Example structure



http://dev.mysql.com/doc/employee/en/sakila-structure.html

Step 1 - Install MySQL



Documentation Library

Table of Contents

MySQL 5.7 Manual

MySQL 5.6 Manual

MySQL 5.5 Manual

MySQL 5.1 Manual

MySQL 5.0 Manual

MySQL 3.23/4.0/4.1 Manual

Search manual:

Chapter 2. Installing and Upgrading MySQL

Table of Contents [+/-]

2.1. General Installation Guidance [+/-]
2.2. Installing MySQL on Unix/Linux Using Generic Binaries
2.3. Installing MySQL on Microsoft Windows [+/-]
2.4. Installing MySQL on Mac OS X [+/-]
2.5. Installing MySQL on Linux [+/-]
2.6. Installing MySQL on Solaris and OpenSolaris [+/-]
2.7. Installing MySQL on HP-UX [+/-]
2.8. Installing MySQL on FreeBSD
2.9. Installing MySQL from Source [+/-]
2.10. Postinstallation Setup and Testing [+/-]
2.11. Upgrading or Downgrading MySQL [+/-]
2.12. Environment Variables
2.13. Perl Installation Notes [+/-]

This chapter describes how to obtain and install MySQL. A summary of the procedure follows and later sections provide the details. If you plan to upgrade an existing version of MySQL to a newer version rather than install MySQL for the first time, see Section 2.11.1, "Upgrading MySQL", for information about upgrade procedures and about issues that you should consider before upgrading.

If you are interested in migrating to MySQL from another database system, you may wish to read Section B.8, "MySQL 5.7 FAQ: Migration", which contains answers to some common questions concerning migration issues.

Installation of MySQL generally follows the steps outlined here:

« 1.9.5 Supporters of MySQL

2.1 General Installation Guidance »

Section Navigation [Toggle]

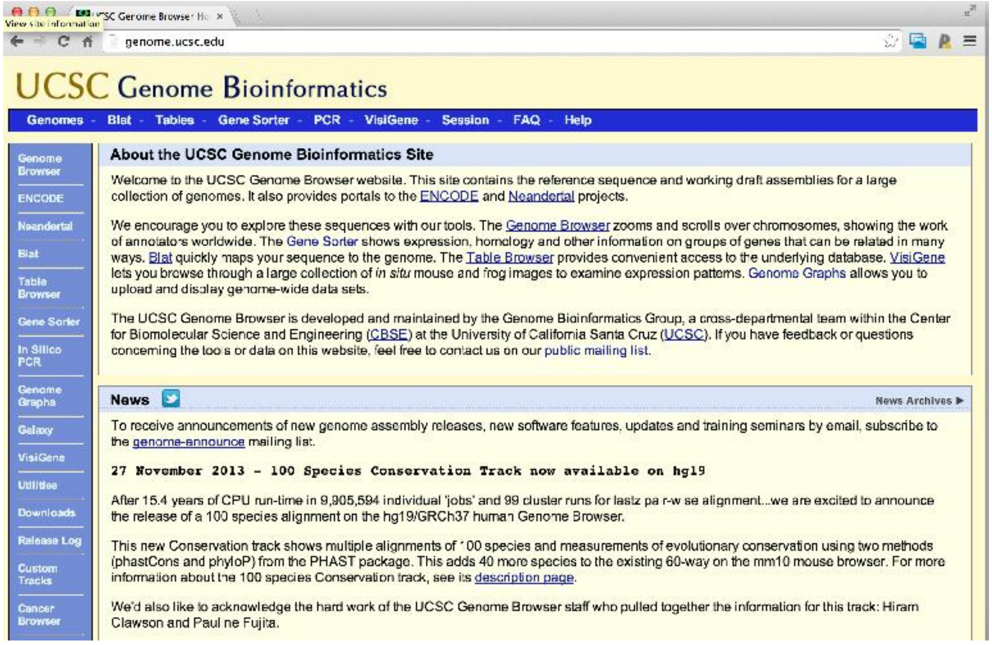
- Preface and Legal Notices
- 1 General Information
- 2 Installing and Upgrading MySQL
  - 2.1 General Installation Guidance
  - 2.2 Installing MySQL on Unix/Linux Using Generic Binaries
  - 2.3 Installing MySQL on Microsoft Windows
  - 2.4 Installing MySQL on Mac OS X
  - 2.5 Installing MySQL on Linux
  - 2.6 Installing MySQL on Solaris and OpenSolaris
  - 2.7 Installing MySQL on HP-UX
  - 2.8 Installing MySQL on FreeBSD
  - 2.9 Installing MySQL from Source
  - 2.10 Postinstallation Setup and Testing
  - 2.11 Upgrading or Downgrading MySQL
  - 2.12 Environment Variables
  - 2.13 Perl Installation Notes
- 3 Tutorial
- 4 MySQL Programs
- 5 MySQL Server Administration
- 6 Security
- 7 Backup and Recovery
- 8 Optimization
- 9 Language Structure
- 10 Globalization

http://dev.mysql.com/doc/refman/5.7/en/installing.html

Step 2 - Install RMySQL

- On a Mac: `install.packages("RMySQL")`
- On Windows:
  - Official instructions - http://biostat.mc.vanderbilt.edu/wiki/Main/RMySQL (may be useful for Mac/UNIX users as well)
  - Potentially useful guide - http://www.ahschulz.de/2013/07/23/installing-rmysql-under-windows/

Example - UCSC database



About the UCSC Genome Bioinformatics Site

Welcome to the UCSC Genome Browser website. This site contains the reference sequence and working draft assemblies for a large collection of genomes. It also provides portals to the ENCODE and Neandertal projects.

We encourage you to explore these sequences with our tools. The Genome Browser zooms and scrolls over chromosomes, showing the work of annotators worldwide. The Gene Sorter shows expression, homology and other information on groups of genes that can be related in many ways. Blat quickly maps your sequence to the genome. The Table Browser provides convenient access to the underlying database. VisiGene lets you browse through a large collection of *in situ* mouse and frog images to examine expression patterns. Genome Graphs allows you to upload and display genome-wide data sets.

The UCSC Genome Browser is developed and maintained by the Genome Bioinformatics Group, a cross-departmental team within the Center for Biomolecular Science and Engineering (CBSE) at the University of California Santa Cruz (UCSC). If you have feedback or questions concerning the tools or data on this website, feel free to contact us on our public mailing list.

News

To receive announcements of new genome assembly releases, new software features, updates and training seminars by email, subscribe to the genome-announce mailing list.

27 November 2013 - 100 Species Conservation Track now available on hg19

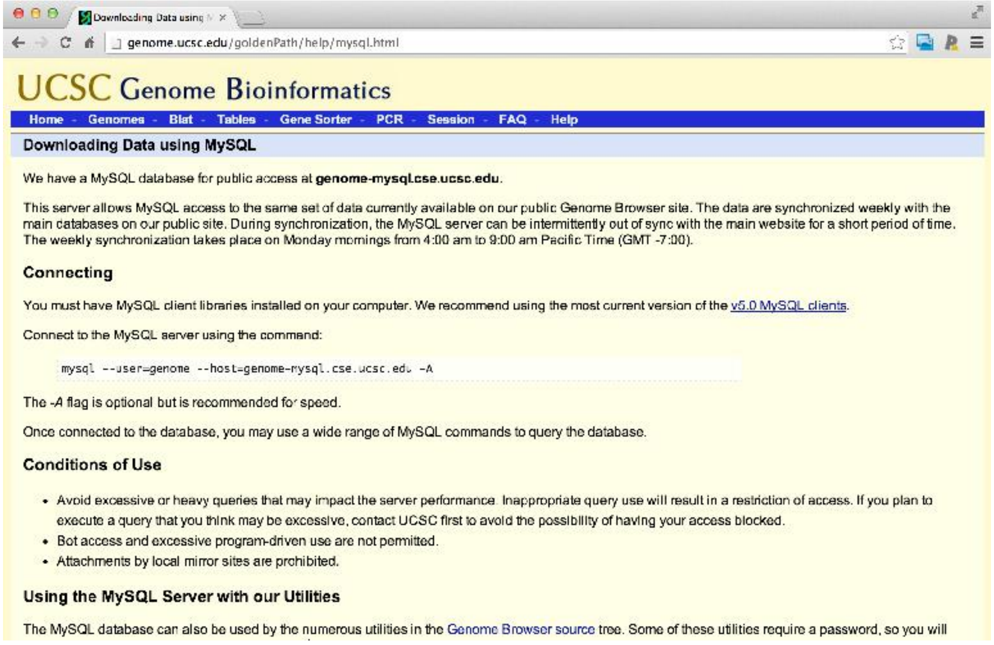
After 15.4 years of CPU run-time in 9,905,594 individual 'jobs' and 99 cluster runs for lastz pair-wise alignment...we are excited to announce the release of a 100 species alignment on the hg19/GRCh37 human Genome Browser.

This new Conservation track shows multiple alignments of 100 species and measurements of evolutionary conservation using two methods (phastCons and phyloP) from the PHAST package. This adds 40 more species to the existing 60-way on the mm10 mouse browser. For more information about the 100 species Conservation track, see its description page.

We'd also like to acknowledge the hard work of the UCSC Genome Browser staff who pulled together the information for this track: Hiram Clawson and Pauline Fujita.

http://genome.ucsc.edu/

UCSC MySQL


Downloading Data using M
genome.ucsc.edu/goldenPath/help/mysql.html
UCSC Genome Bioinformatics
Home - Genomes - Blat - Tables - Gene Sorter - PCR - Session - FAQ - Help

Downloading Data using MySQL

We have a MySQL database for public access at **genome-mysql.cse.ucsc.edu**.

This server allows MySQL access to the same set of data currently available on our public Genome Browser site. The data are synchronized weekly with the main databases on our public site. During synchronization, the MySQL server can be intermittently out of sync with the main website for a short period of time. The weekly synchronization takes place on Monday mornings from 4:00 am to 9:00 am Pacific Time (GMT -7:00).

Connecting

You must have MySQL client libraries installed on your computer. We recommend using the most current version of the v5.0 MySQL clients.

Connect to the MySQL server using the command:

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A
```

The *-A* flag is optional but is recommended for speed.

Once connected to the database, you may use a wide range of MySQL commands to query the database.

Conditions of Use

- Avoid excessive or heavy queries that may impact the server performance. Inappropriate query use will result in a restriction of access. If you plan to execute a query that you think may be excessive, contact UCSC first to avoid the possibility of having your access blocked.
- Bot access and excessive program-driven use are not permitted.
- Attachments by local mirror sites are prohibited.

Using the MySQL Server with our Utilities

The MySQL database can also be used by the numerous utilities in the Genome Browser source tree. Some of these utilities require a password, so you will

http://genome.ucsc.edu/goldenPath/help/mysql.html

Connecting and listing databases

```
ucscDb <- dbConnect(MySQL(),user="genome",
                    host="genome-mysql.cse.ucsc.edu")
result <- dbGetQuery(ucscDb,"show databases;"); dbDisconnect(ucscDb);
```

```
[1] TRUE
```

```
result
```

```
              Database
1   information_schema
2              ailMel1
3              allMis1
4              anoCar1
5              anoCar2
6              anoGam1
7              apiMel1
8              apiMel2
9              aplCal1
10             bosTau2
11             bosTau3
12             bosTau4
13             bosTau5
14             bosTau6
15             bosTau7
16           bosTauMd3
17             braFlo1
18             caeJap1
19              caePb1
20              caePb2
21             caeRem2
22             caeRem3
23             calJac1
24             calJac3
25             canFam1
26             canFam2
27             canFam3
28             cavPor3
29                 cb1
30                 cb3
31                ce10
32                 ce2
33                 ce4
34                 ce6
```

Connecting to hg19 and listing tables

```
hg19 <- dbConnect(MySQL(),user="genome", db="hg19",
                    host="genome-mysql.cse.ucsc.edu")
allTables <- dbListTables(hg19)
length(allTables)
```

```
[1] 10949
```

```
allTables[1:5]
```

```
[1] "HInv"         "HInvGeneMrna" "acembly"      "acemblyClass" "acemblyPep"
```

Get dimensions of a specific table

```
dbListFields(hg19,"affyU133Plus2")
```

```
 [1] "bin"         "matches"     "misMatches"  "repMatches"  "nCount"      "qNumInsert"
 [7] "qBaseInsert" "tNumInsert"  "tBaseInsert" "strand"      "qName"       "qSize"
[13] "qStart"      "qEnd"        "tName"       "tSize"       "tStart"      "tEnd"
[19] "blockCount"  "blockSizes"  "qStarts"     "tStarts"
```

```
dbGetQuery(hg19, "select count(*) from affyU133Plus2")
```

```
  count(*)
1    58463
```

Read from the table

```
affyData <- dbReadTable(hg19, "affyU133Plus2")
head(affyData)
```

```
  bin matches misMatches repMatches nCount qNumInsert qBaseInsert tNumInsert tBaseInsert strand
1 585     530          4          0     23          3          41          3         898      -
2 585    3355         17          0    109          9          67          9       11621      -
3 585    4156         14          0     83         16          18          2          93      -
4 585    4667          9          0     68         21          42          3        5743      -
5 585    5180         14          0    167         10          38          1          29      -
6 585     468          5          0     14          0           0          0           0      -
         qName qSize qStart qEnd tName     tSize tStart  tEnd blockCount
1  225995_x_at   637      5  603  chr1 249250621  14361 15816          5
2  225035_x_at  3635      0 3548  chr1 249250621  14381 29483         17
3  226340_x_at  4318      3 4274  chr1 249250621  14399 18745         18
4 1557034_s_at  4834     48 4834  chr1 249250621  14406 24893         23
5    231811_at  5399      0 5399  chr1 249250621  19688 25078         11
6    236841_at   487      0  487  chr1 249250621  27542 28029          1
                                                                  blockSizes
1                                                          93,144,229,70,21,
2              73,375,71,165,303,360,198,661,201,1,260,250,74,73,98,155,163,
3                 690,10,32,33,376,4,5,15,5,11,7,41,277,859,141,51,443,1253,
4 99,352,286,24,49,14,6,5,8,149,14,44,98,12,10,355,837,59,8,1500,133,624,58,
5                                       131,26,1300,6,4,11,4,7,358,3359,155,
6                                                                       487,
                                                                                                 qS
1                                                                                    34,132,278,541
2                        87,165,540,647,818,1123,1484,1682,2343,2545,2546,2808,3058,3133,3206,3317,
3                   44,735,746,779,813,1190,1195,1201,1217,1223,1235,1243,1285,1564,2423,2565,2617,
4 0,99,452,739,764,814,829,836,842,851,1001,1016,1061,1160,1173,1184,1540,2381,2441,2450,3951,4103,
5                                                     0,132,159,1460,1467,1472,1484,1489,1497,1856,
6

1
2                                     14381,14454,14969,15075,15240,15543,15903,16104,16853,17054,1
3                               14399,15089,15099,15131,15164,15540,15544,15549,15564,15569,15580,1
4 14406,20227,20579,20865,20889,20938,20952,20958,20963,20971,21120,21134,21178,21276,21288,21298,2
5                                                                          19688,19819,19845,21145,2
6
```

Select a specific subset

```
query <- dbSendQuery(hg19, "select * from affyU133Plus2 where misMatches between 1 and 3")
affyMis <- fetch(query); quantile(affyMis$misMatches)
```

```
  0%  25%  50%  75% 100% 
   1    1    2    2    3 
```

```
affyMisSmall <- fetch(query,n=10); dbClearResult(query);
```

```
[1] TRUE
```

```
dim(affyMisSmall)
```

```
[1] 10 22
```

Don't forget to close the connection!

```
dbDisconnect(hg19)
```

```
[1] TRUE
```

Further resources

- RMySQL vignette http://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf
- List of commands http://www.pantz.org/software/mysql/mysqlcommands.html
  - **Do not, do not, delete, add or join things from ensembl. Only select.**
  - In general be careful with mysql commands
- A nice blog post summarizing some other commands http://www.r-bloggers.com/mysql-and-r/