Reading mySQL

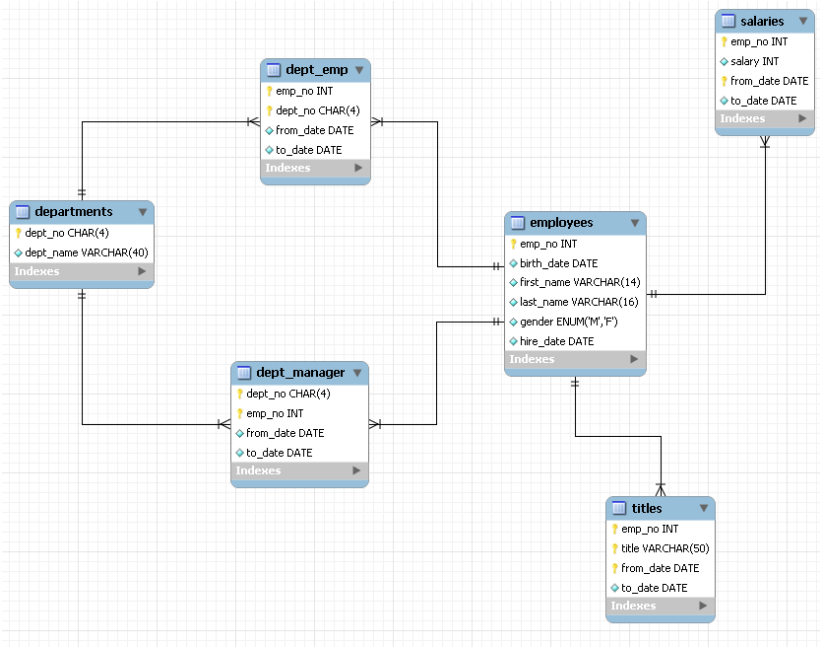
Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

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

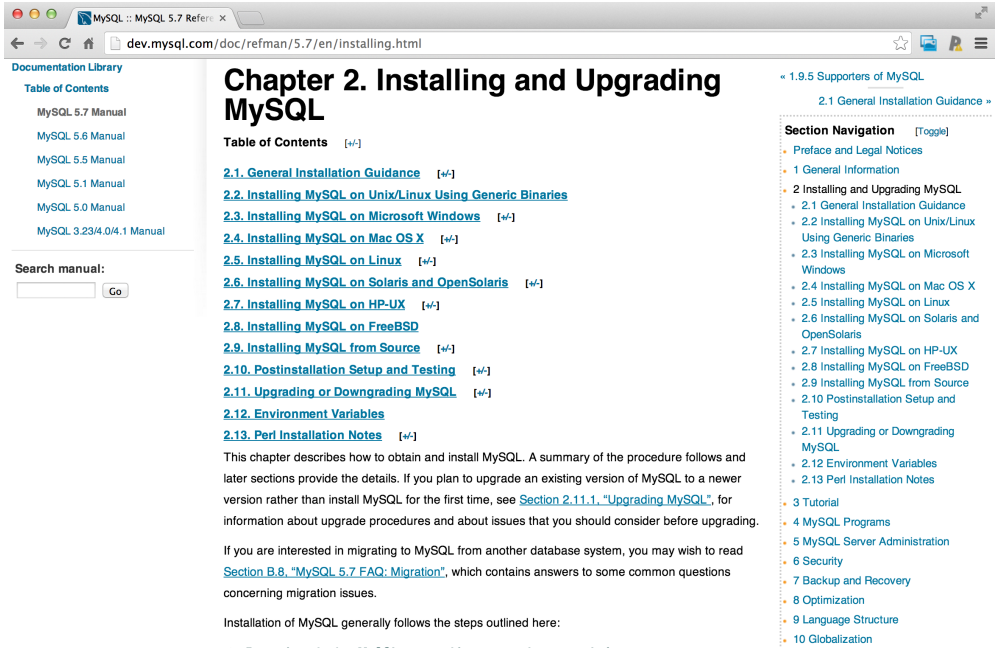
http://en.wikipedia.org/wiki/MySQL http://www.mysql.com/

Example structure



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

Step 1 - Install MySQL

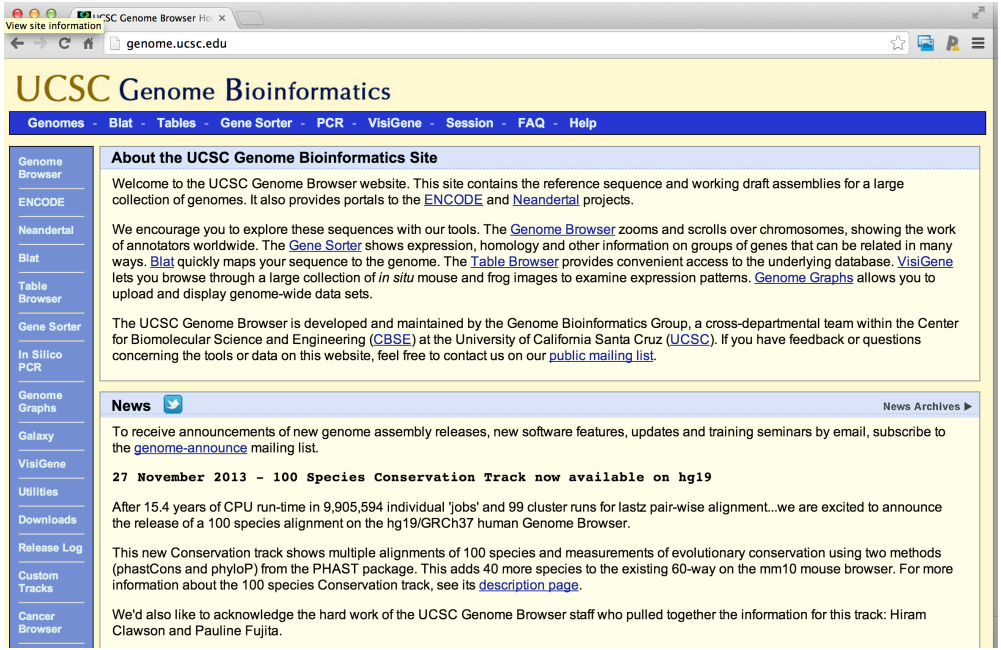


http://dev.mysql.com/doc/refman/5.7/en/installing.html

Step 2 - Install RMySQL

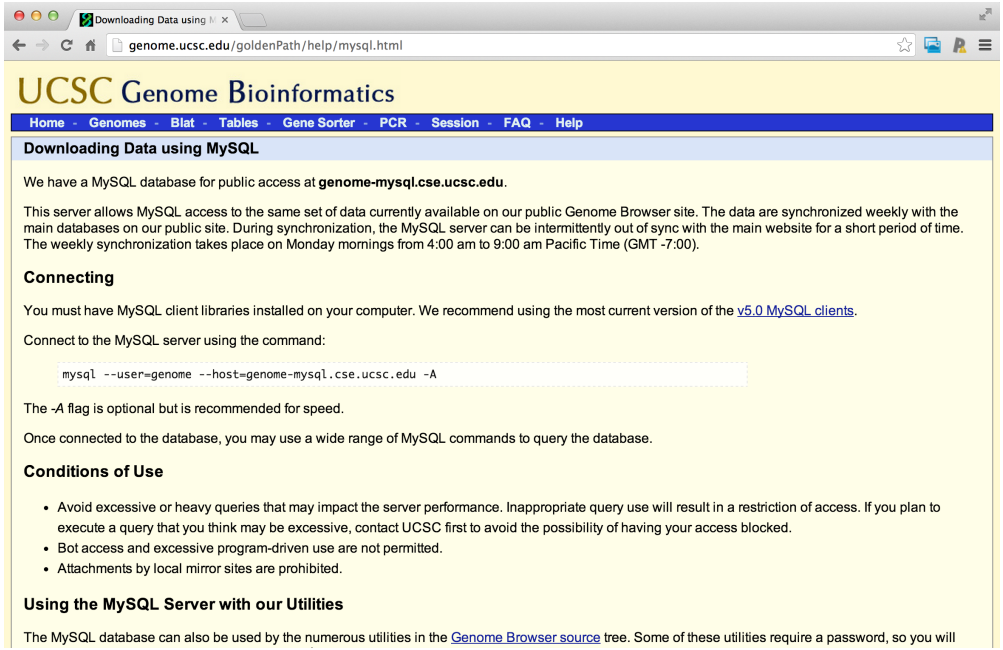
- On a Mac: `install.packages("RMySQL")`
- On Windows:
  - Official instructions - http://biostat.mc.vanderbilt.edu/wiki/Main/RMySQL (may be useful for Mac/UNIX users as well)
  - Potentially useful guide - http://www.ahschulz.de/2013/07/23/installing-rmysql-under-windows/

Example - UCSC database



http://genome.ucsc.edu/

UCSC MySQL



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
```

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,
```

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/