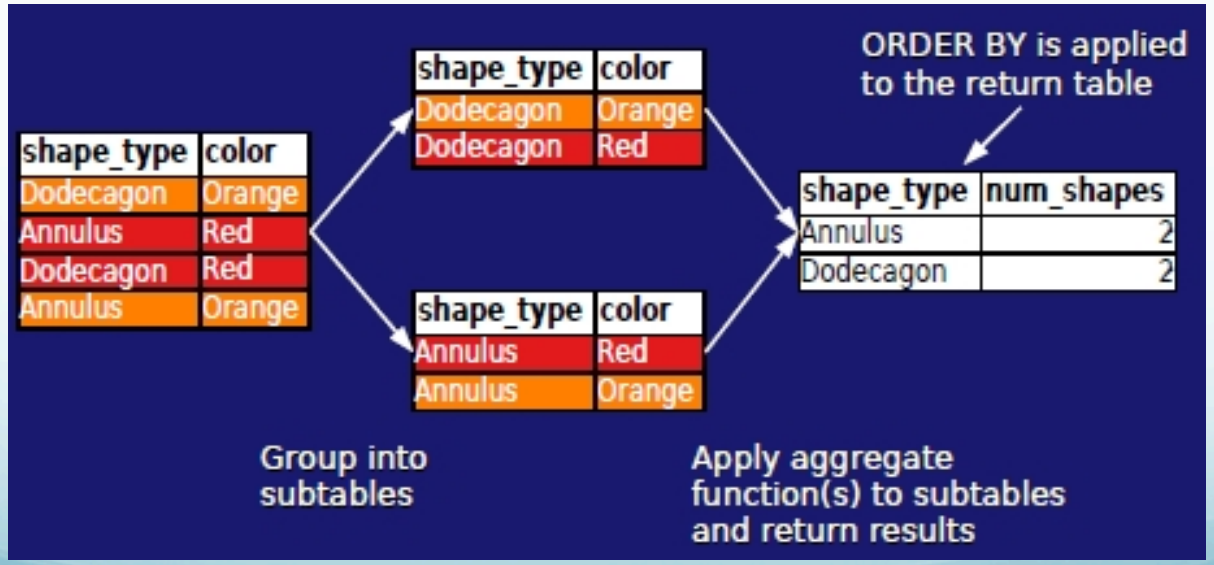
BINF 8211/6211
Design and Implementation of Bioinformatics Databases
Lecture #12

Dr. D. Andrew Carr
Dept. Bioinformatics and Genomics UNCC
Spring 2016

A simple single-column example of the GROUP BY statement is shown

```
SELECT     shape_type,
           COUNT(shape_type) as num_shapes
FROM       shape
GROUP BY   shape_type
ORDER BY   shape_type ASC
```

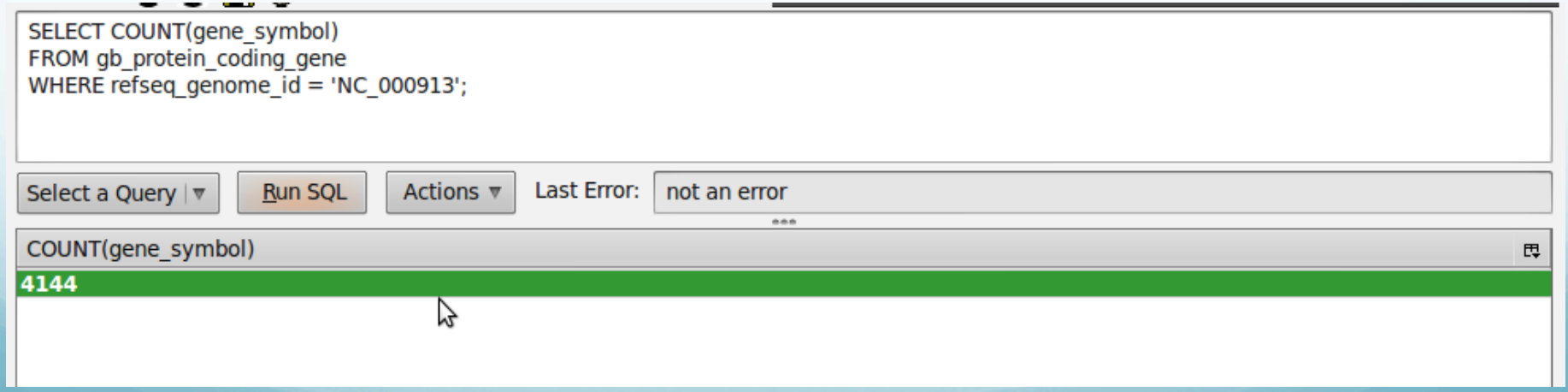


You can group data by the values in one or more columns and then use aggregate functions within those groups (GROUP BY)

Step 1: group your data; you can think of the groups as subtables.

Step 2: apply aggregate functions to the groups (subtables) and re-sort the results back to a table

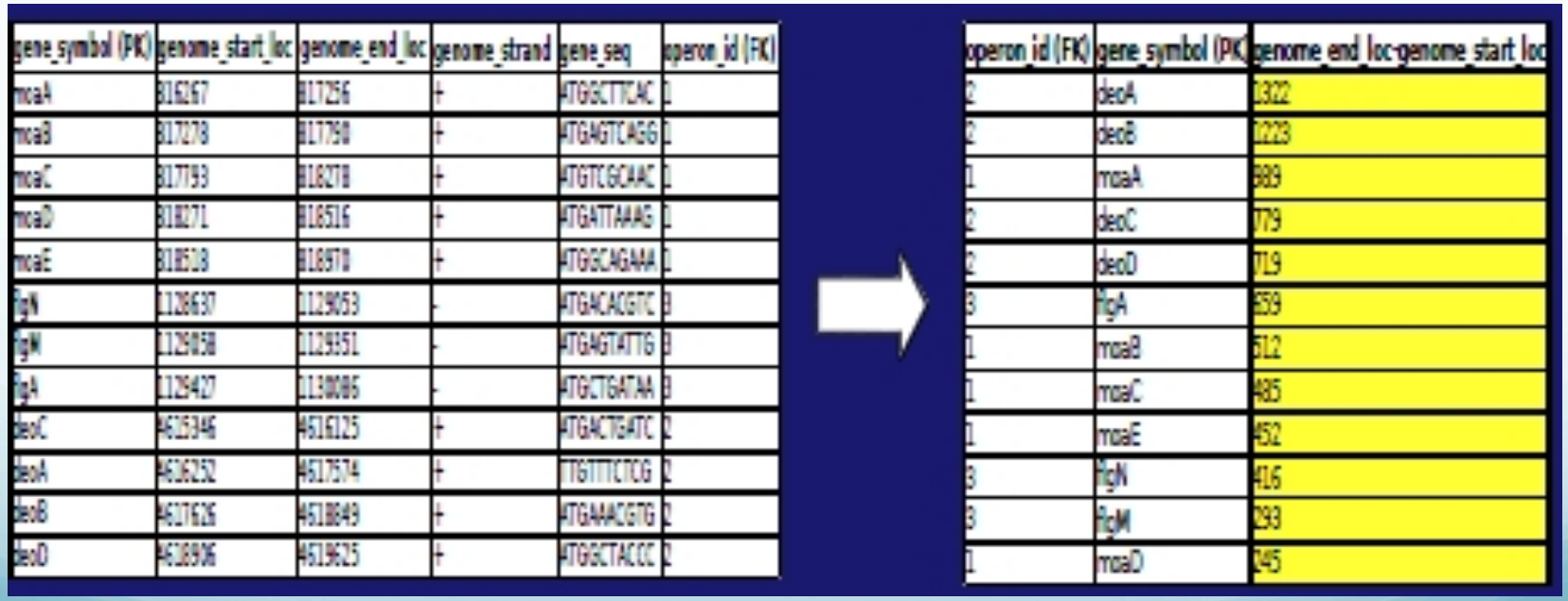
```
SELECT COUNT(gene_symbol)
FROM gb_protein_coding_gene
WHERE refseq_genome_id = 'NC_000913';
```



You can compute values in your SELECT statements

Query: what is the length of each gene; order the output from longest to shortest

```
SELECT     operon_id, gene_symbol,
           (genome_end_loc-genome_start_loc) + 1
FROM       gene
ORDER BY   genome_end_loc-genome_start_loc + 1 DESC
```



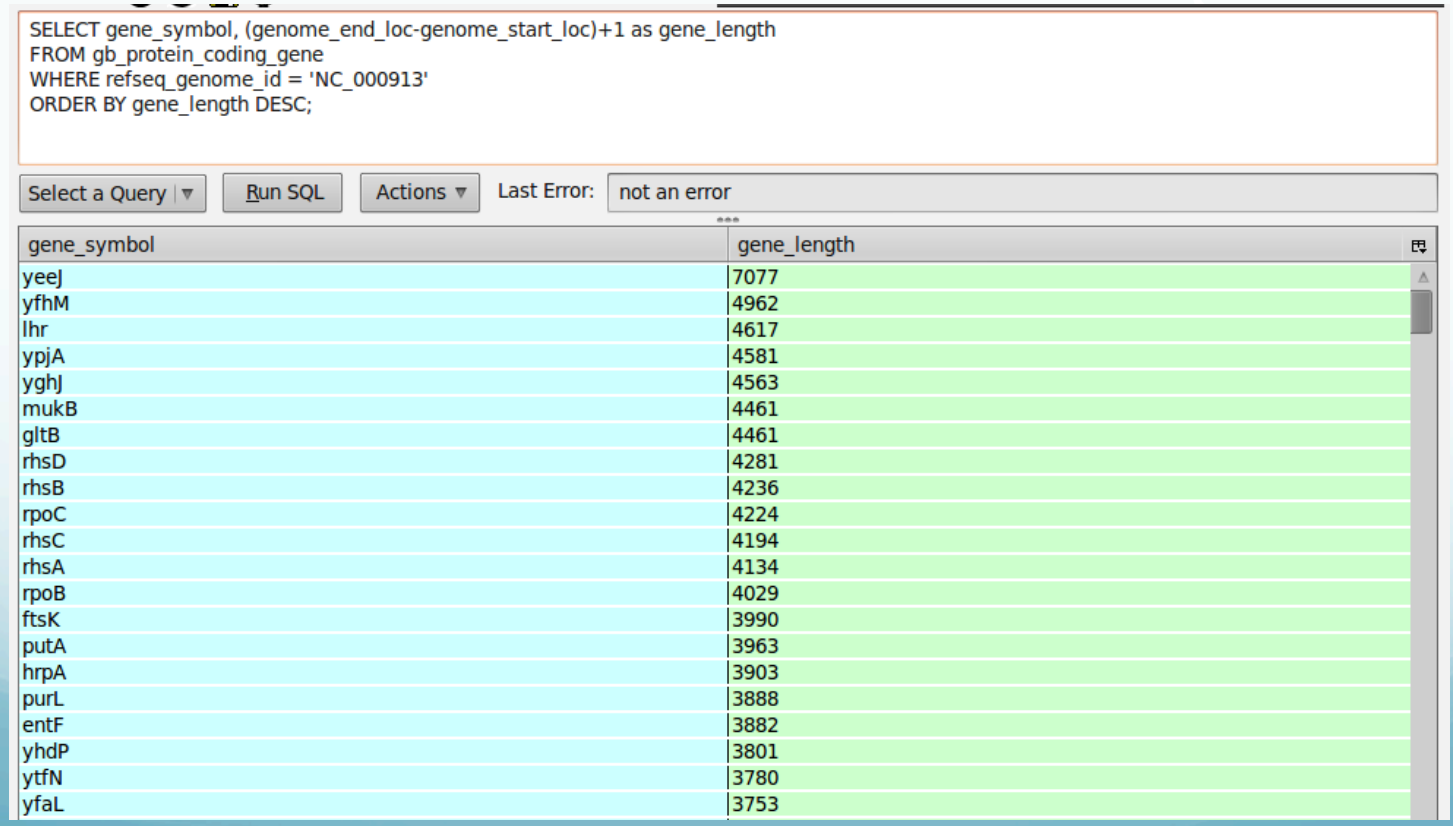
gene_symbol (PK)	genome_start_loc	genome_end_loc	genome_strand	gene_seq	operon_id (FK)
moaA	8162167	8117256	+	ATGGCTTCAC	1
moaB	8172278	8117798	+	ATGAGTCAGG	1
moaC	8177193	8182718	+	ATGTCGCAAC	1
moaD	8181871	8186516	+	ATGATTAAAG	1
moaE	8185128	8185978	+	ATGGCAGAAA	1
hgN	1126637	1129053	-	ATGACACGTC	3
hgM	1129038	1129351	-	ATGAGTATTG	3
hgA	1129427	1130886	-	ATGCTGATAA	3
deoC	4615346	4516125	+	ATGACTGATC	2
deoA	4616252	4517574	+	TTGTTTCTCG	2
deoB	4517626	4518849	+	ATGAAACGTG	2
deoD	4518906	4519625	+	ATGGCTACCC	2

operon_id (FK)	gene_symbol (PK)	genome_end_loc-genome_start_loc
2	deoA	1322
2	deoB	1223
1	moaA	889
2	deoC	779
2	deoD	719
3	hgA	559
1	moaB	512
1	moaC	485
1	moaE	452
3	hgN	416
3	hgM	293
1	moaD	245

Computed Columns

Query: Compute the gene length for each gene in the NC_000913 genome; present the results in order from longest to shortest.

```
SELECT gene_symbol, (genome_end_loc-genome_start_loc)+1 as gene_length
FROM gb_protein_coding_gene
WHERE refseq_genome_id = 'NC_000913'
ORDER BY gene_length DESC;
```



gene_symbol	gene_length
yeeJ	7077
yfhM	4962
lhr	4617
ypjA	4581
yghJ	4563
mukB	4461
gltB	4461
rhsD	4281
rhsB	4236
rpoC	4224
rhsC	4194
rhsA	4134
rpoB	4029
ftsK	3990
putA	3963
hrpA	3903
purL	3888
entF	3882
yhdP	3801
ytfN	3780
yfaL	3753

Database Design

- What is missing in most database systems?
- Project for two lectures:
  - Import data into database.
    - Sequence data from different sources
      - .fastq
      - .sff 454 data

Some notes from Demo

- Not all files will easliy fit into sqllite
  - http://www.pythoncentral.io/introduction-to-sqlite-in-python/
- Examples:
  - F3P5W4J04_CR_Pcoffeae.sff
  - F3P5W4J04_CR_Pcoffeae.fastq
    - ~156MB
- Illumina Data much bigger.
  - ~30GB
  - ~100GB

- Postgres
  - Psycopg2 for python
- MySQL
  - Mostpopular version
  - Owned by Oracle
  - “MySQL can be used directly with biopython.”
  - Postgresql too.

- Biopython.org
  - Open the file
  - SeqIO.parse

Sequence Data Example Fastq

@F3P5W4J04IX4F3
TTTGGACGTATACTAAACATGGAAACAAACATAGAACCAAACGTGGAATCGGGTACCAAATTCTC
TGGGCCGCAAATCGCAGCATTGTGACCCA
+
???FFFFFFFFIIHHFFGFF??
666:<<<DFGHFHIIIHHHIIIIIIIIIHHFFFFFFFFFFFFFFFFFFFFDDDFFFFFFFDDAAA<<4444
@F3P5W4J04IEQHF
TTTTTTTTATTTCCAAAAATTTCCAACCAAAATTTTGAGCACAAATCCTCGAAAACAATTTTTTTTA
CTGCCCCCAAATTTTCATAAAAATATTTTTTTTAATGAATTTTCCCCTTGGAAAAGTGCAAAAATTA
TTTCTTCCCCAATCTTCGATAAATTTTCAAATTTTTTTTGATTTTTTGATTTTTGCCAATTCATTT
ATTTCTTCCCCAAATATTATTTTTTTCTTTTTTTTTAA
+
--------857:CC/
0000///4499894669=@BBIIIIIIHHHIIIIII@@@@IAA4444444II6I777::FHH@@@@GII
866558/,,,,,,,--/<<?
4468244477EE<<==IIII;;===II>HGGGIIAACCIIIIIIIIIIHHC9974=222,,,,,,,,33------::,,,,,
32222227?///85::I335555B><F9>?:,,,,,,,3++++++++9=

Steps taken prior to examples

- Downloaded MySQL
  - Mysql
    - Set binary path
    - Set root user password
      - Mysql admin –u root –p password ‘xxxxxxxx’
    - Used Mysql Workbench to set up connection to DB
  - Checked setting to make sure the that the mysql connects on entry
  - Set a new user in MySQL Workbench
  - Created Lecture12 db

MYSQL Python connection

- Creating a connection
  - To attach to a database
  - use mysql.connector.connect
- To Query
  - Must create a cursor
  - Remember to close the cursor
- Changes will have to be committed
  - Cnx.commit()

```
import mysql.connector

config = {
  'user': 'DAC_icloud',
  'password': 'Temp1',
  'host': '127.0.0.1',
  'database': 'Lecture12',
  'raise_on_warnings': True,
}

cnx = mysql.connector.connect(**config)



cursor = cnx.cursor()

cnx.close()
```

Create a table

- Use standard syntax inside quotes
- Not ; needed
- Execute the command

```
cursor =cnx.cursor()

cmd2 = "CREATE TABLE Fastqdata3 (\
    idFastq_data3 int(11), \
    Header varchar(150) DEFAULT NULL,\
    Sequence varchar(150) DEFAULT NULL,\
    Score varchar(150) DEFAULT NULL,\
    PRIMARY KEY (idFastq_data3))
ENGINE=InnoDB"

cursor.execute(cmd2)
```

Insert data via python

- To insert data
  - Using open connection
  - Create the SQL for the insert
  - Create data entry
- Execute the command
- Commit

```
add_fastq = ("INSERT INTO Fastqdata3 "
             "(idFastq_data3, Header, Sequence, Score) "
             "VALUES (%s, %s, %s, %s)")


data_fastq = ('1', 'h.1', 'ATCAAGATGCATTGAC',
'lskdjfoalslls')

# Insert new employee
cursor.execute(add_fastq, data_fastq)


cnx.commit()

cursor.close()
cnx.close()
```

Moving data from Python into a DB

- Requires that the database and tables exist.
- Create the following tables: (Gene)
  - Seq_id
  - Seq_strand
  - Seq_start
  - Seq_stop
  - Exon_start
  - Ref_seq_gene_id
  - What is missing
    - Gene_id
- Access the db and then access the tables that you need.
- Protein
  - Protein_id
  - Gene_id
  - Protein_length
  - Sequence
  - Protein_description
  - COG_ID
  - Protien_gi_num

Homework 4

- Load your data into a database:
  - NON-TRIVIAL
  - Create tables to load the data that you collected
    - Nucleotide sequence data
    - Protein data
      - Sequence data (translation of the DNA data)
      - Structure data (Think about how you want to do this!!!)
    - Pathway data
    - Table to hold data lineage
      - 1 Meta data table
  - You will decide the attributes needed and their data types.
    - Use the file definitions as guidance
      - FASTQ, FASTA ....
  - Link these tables so that they can be queried
  - Ultimate GOAL: Pull the structure information for a given DNA sequence.
  - Due March 1st at 8:00 a.m.