More SQL

Views

- A View is a logical table backed up by a query
  - Changes automatically when the results of the query change

```
mysql> CREATE VIEW CS150_VW AS
  SELECT LName, FName, Grade, StudentID
  FROM Courses, CurrentlyEnrolled, People
  WHERE Courses.CourseID=
  CurrentlyEnrolled.CourseID and
  People.PersonID=StudentID and
  Title like “CS150%”;
```

Views

```
mysql> SELECT * FROM CS150_VW;
mysql> DELETE FROM People WHERE
  PersonID=5;
mysql> SELECT * FROM CS150_VW Order by Grade;
mysql> DROP VIEW CS150_VW;
```

Exercise

- Rebuild CS150_VW

- Determine how closely a student's grade in CS150 matches their GPA. (1.0 = perfect match, 0.5 = Grade is half the GPA, 1.5 Grade is 50% better than GPA)

- GPA goes from 0.0 to 4.0, Grade goes from 0.0 to 100.0

GROUP BY and HAVING

- Allows SQL to filter on calculated/aggregate values
- Similar to **WHERE**
- must be last

```
SELECT StudentID, avg(Grade) as AvgGrade,
count(*) as NumberRows

FROM CurrentlyEnrolled

WHERE Grade > 20

GROUP BY StudentID

HAVING AvgGrade > 80 and NumberRows > 1;
```

Limit – only show some results

```
SELECT StudentID, count(*) as Total
FROM WasIn
GROUP BY StudentID
HAVING Total > 1
LIMIT 2; -- show only first two rows

LIMIT 2,4; -- skip the first two rows, then show the next 4
```

```
LIMIT 3; is equivalent to LIMIT 0, 3
```

Backup Your Database!

```
@db:~> mysqldump PUNetID_test -u PUNetID -p  > backup_test.sql
```

Database Name

Output file

To see what this file looks like:

```
@db:~> cat  backup_test.sql  | less
```

Copy to Zeus for safe keeping!

```
@db:~> scp backup_test.sql PUNetID@zeus:
```

Don't forget the colon!

Explain

```
mysql> SHOW CREATE TABLE People;

mysql> SHOW CREATE TABLE CurrentlyTeaching;
```

```
mysql> EXPLAIN SELECT * FROM People,
  CurrentlyTeaching WHERE (PersonID=ProfID);
```

```
mysql> EXPLAIN SELECT * FROM People, CurrentlyTeaching
  WHERE (PersonID=ProfID)
  AND FName like '%a%';
```

EXPLAIN

TYPE: system, const, eq_ref, ref, index, all

ROWS: number of rows scanned

Indexes

```
mysql> USE enron;

mysql> SHOW TABLE STATUS LIKE  'EnronVocab';

mysql> SHOW TABLE STATUS LIKE  'EnronWordCount';

mysql> SHOW CREATE TABLE EnronVocab;

mysql> SHOW CREATE TABLE EnronWordCount;

mysql> SHOW PROCESSLIST;
```

```
mysql> EXPLAIN SELECT WordCount FROM  EnronWordCount WHERE DocID = ??;

mysql> EXPLAIN SELECT WordCount FROM  EnronWordCount WHERE WordID = ??;
```

- 1 to 39861 DocID
- 1 to 28102 WordID

http://archive.ics.uci.edu/ml/datasets/Bag+of+Words

INTO OUTFILE

- Save a query to a text file

```
SELECT StudenID, count(*) as Total
FROM CurrentlyEnrolled
GROUP BY StudentID
HAVING Total > 1
INTO OUTFILE '/tmp/PUNETID.txt';

-- writes data on the server

db$ scp /tmp/PUNETID.txt c@zeus:
mysql -u user -p -D database -e
“select ... “ > outfile
```

LOAD DATA INFILE

```
mysql> source /tmp/createTest.sql;

mysql> ALTER TABLE ReadTest DISABLE KEYS;
mysql> SET FOREIGN_KEY_CHECKS=0;

mysql> LOAD DATA INFILE '/tmp/test.txt' INTO TABLE ReadTest
COLUMNS TERMINATED BY ',';

mysql> SET FOREIGN_KEY_CHECKS=1;
mysql> ALTER TABLE ReadTest ENABLE KEYS;

Query OK, 69679427 rows affected (21 min 34.26 sec)
```

- with a well tuned MySQL (innodb_buffer_pool_size, innodb_log_*)

Practice

- List all Course titles and CourseID. For each Course, display the CourseID if Chadd teaches it and “Not A Chadd Course” otherwise.
- Find all courses whose maximum and minimum grade is at least 50 points different.
- Display each student name, course title, and student's grade in that course and the string “passing” or “not passing” if the student is not passing the course.
- Find each course that does not contain a Simpson.
- Display all students whose grade for a class is above the average grade for that class.

Practice

- How many students are in each class?

- For each class, what was the min, max, average grade ?
  - do this with and without using the AVG() function.

- Who took a class with Bart Simpson and received a higher grade than Bart? Lower Grade?

Update

- UPDATE [IGNORE] table SET col=value,... WHERE ...

Delete

- DELETE [IGNORE] FROM table WHERE ....

mysql client

- On the VM:

  start the tunnel

  ```
  mysql -h localhost -P 3306 -u punetid -p
  ```