

Advanced SQL

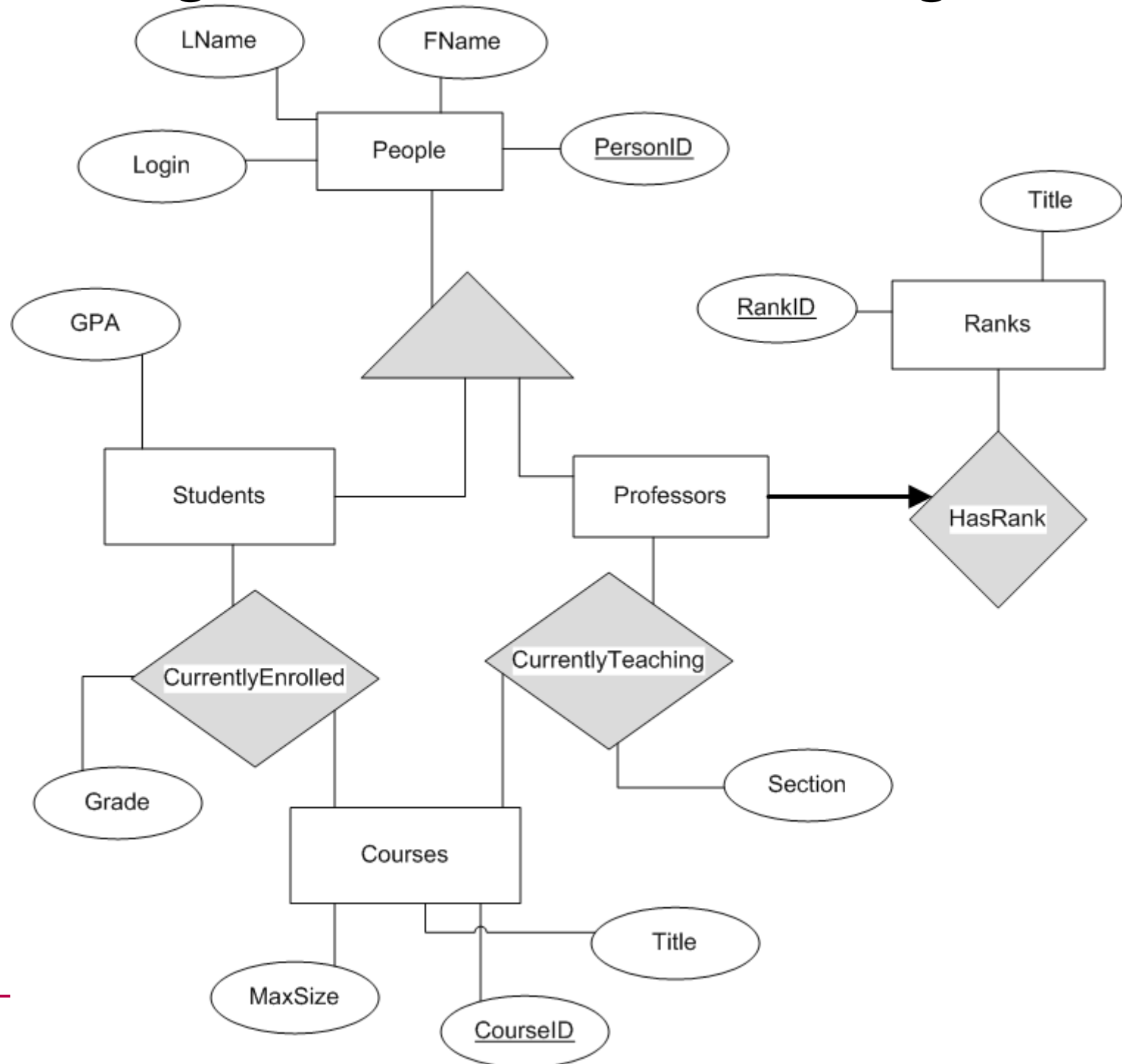
Nov 21, 2017

<http://zeus.cs.pacificu.edu/chadd/cs445f17/advancedSQL.tar.gz>

Topics

- Views
- Triggers
- Stored Procedures
- Control Flow
 - if / case
- Binary Data

College Database E-R Diagram



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

```
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%";
```

```
SELECT * FROM CS150_VW;
```

```
DELETE FROM People WHERE PersonID=5;
```

```
SELECT * FROM CS150_VW Order by Grade;
```

```
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 = 150 Grade is half the GPA, 1.5 150 Grade is 50% better than GPA)
- GPA goes from 0.0 to 4.0, Grade goes from 0.0 to 100.0
- You can do this without a View

Triggers

```
CREATE TRIGGER name BEFORE INSERT ON table
  FOR EACH ROW BEGIN
    -- SQL Statements or control flow (IF)
    INSERT INTO test2 SET a2 = NEW.a1;
  END
;
```

The row being inserted

BEFORE | AFTER

INSERT | DELETE | UPDATE

<https://mariadb.com/kb/en/library/trigger-overview/>

Example (triggerExample.sql)

```
CREATE TABLE user (  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  first_name CHAR(20),  
  last_name CHAR(20),  
  email CHAR(100)  
) ENGINE = MyISAM; https://mariadb.com/kb/en/library/trigger-overview/  
  
DELIMITER //  
CREATE TRIGGER validateEmail  
  BEFORE INSERT ON user  
  FOR EACH ROW  
BEGIN  
  IF NEW.email NOT LIKE '%@%.%' THEN  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Email field is not valid';  
  END IF;  
END; //  
DELIMITER ;  
  
INSERT INTO user (first_name, last_name, email) VALUES ('John', 'Doe',  
'john_doe.example.net');  
ERROR 1644 (45000): Email field is not valid  
  
SELECT * FROM user;  
Empty set (0.00 sec)
```


- `// delete trigger to enforce bold line`
`see boldLineTriggerExample.sql`

Stored Procedures

- see `StoredProcedureExample.sql`

<https://mariadb.com/kb/en/library/create-procedure/>
<https://mariadb.com/kb/en/library/stored-routines/>

Control Flow

IF(condition, trueValue, falseValue)

```
SELECT ActorID, IF( salary>1000, 1, 0)  
FROM WasIn;
```

IFNULL(Value, returnIfValueIsNull)

```
SELECT ActorID, IFNULL(StageName, "NONAME")  
FROM WasIn;
```

There is also a case (switch) statement

Blob: Binary Data

- BLOB

- Binary data can be images, sounds, video, OpenOffice documents, executable files, etc.

- TEXT

- Long chunk of text

Data Type	Max Size (Bytes)
TINYBLOB, TINYTEXT	2^8
BLOB, TEXT	2^{16}
MEDIUMBLOB, MEDIUMTEXT	2^{24}
LOB, LONGTEXT	2^{32}

Binary Data

```
CREATE TABLE pictures (  
  PicID int(11) NOT NULL auto_increment,  
  image mediumblob NOT NULL,  
  type varchar(255) NOT NULL,  
  PRIMARY KEY (`PicID`)) ENGINE=InnoDB;
```

For binary data, we need to track the type of data we have stored.

For data to display on the web, store the MIME type

Multipurpose **I**nternet **M**ail **E**xtensions

image/gif

image/png

INSERT Binary Data

```
INSERT INTO pictures (image, type) VALUES  
(LOAD_FILE('/tmp/g0/image.png'),  
"mime/png");
```

- From the command line you must specify the entire path to the file
- The file must be world readable

```
chadd@db:~> mkdir /tmp/g0
```

```
chadd@db:~> chmod o+r /tmp/g0/image.png
```

```
chadd@db:~> ls -al /tmp/g0/image.png
```

```
-rw-r--r-- /tmp/g0/image.png
```



This INSERT can be run via dbeaver IF the file is already in place on the db machine!

INTO OUTFILE

- Save a query to a text file

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

```
-- writes data on the server
```

```
gray> 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 test DISABLE KEYS;
```

```
mysql> SET FOREIGN_KEY_CHECKS=0;
```

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

```
mysql> SET FOREIGN_KEY_CHECKS=1;
```

```
mysql> ALTER TABLE test ENABLE KEYS;
```

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

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