```sql
-- ------------------------------------------------------------------------
-- File:    advancedSQLExamples.sql
-- Author:  Chadd Williams
-- Date:    11/17/2017
-- Class:   CS445
-- Purpose: Demonstrate View, Trigger
-- ------------------------------------------------------------------------


-- View
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;


-- Triggers
CREATE TABLE user (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    first_name CHAR(20),
    last_name CHAR(20),
    email CHAR(100)
) Engine = InnoDB;

-- run triggerExamples.sql

SHOW TRIGGERS;

INSERT INTO user (first_name, last_name, email) VALUES ('John', 'Doe',
'john_doe.example.net');

select * from user;
```

```sql
Drop table user;
```

```sql
-- ------------------------------------------------------------------------
-- File:    triggerExample.sql
-- Author:  Chadd Williams
-- Date:    11/17/2017
-- Class:   CS445
-- Purpose: Demonstrate Trigger
--          This file is necessary so you can run the entire script at once.
--          The Delimiter lines are necessary since ; is used in the trigger
--          The Delimiter lines are their own statements so this file contains
--          three statements that must be run together
-- ------------------------------------------------------------------------

-- 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 ;
```

```sql
-- -----------------------------------------------------------------------------
-- File:    boldLineTriggerExample.sql
-- Author:  Chadd Williams
-- Date:    11/17/2017
-- Class:   CS445
-- Purpose: Demonstrate Trigger
-- -----------------------------------------------------------------------------

drop table if exists Wrote;
drop table if exists Authors;
drop table if exists Books;

CREATE TABLE Authors (
    AuthorID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    FName VARCHAR(20),
    LName VARCHAR(20)
) Engine = InnoDB;

CREATE TABLE Books (
    BookID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Title CHAR(200)
) Engine = InnoDB;

CREATE TABLE Wrote (
    BookID INT NOT NULL,
    AuthorID INT NOT NULL,

    CONSTRAINT Wrote_BookIDAuthorID_PK PRIMARY KEY
        (BookID, AuthorID),

    CONSTRAINT Wrote_BookID_FK FOREIGN KEY (BookID)
        REFERENCES Books(BookID) ON DELETE CASCADE,

    CONSTRAINT Wrote_AuthorID_FK FOREIGN KEY (AuthorID)
        REFERENCES Authors(AuthorID) ON DELETE CASCADE

) Engine = InnoDB;

INSERT INTO Authors (FName, LName) VALUES ('Raghu', 'Ramakrishnan');

INSERT INTO Books (Title) VALUES ('Database Management Systems');

INSERT INTO Wrote (BookID, AuthorID) VALUES (1, 1);
```

```sql
DELIMITER //
Create Trigger BoldLine BEFORE DELETE ON  Wrote
For each row
Begin
    declare homeRowCount INTEGER;
    declare rowCount INTEGER;

    SELECT COUNT(*) FROM Books WHERE
        BookID=OLD.BookID into homeRowCount;

    SELECT Count(*) from Wrote Where
        Wrote.BookID=OLD.BookID into rowCount;

    If homeRowCount > 0 and rowCount = 1 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Bold Line';
    END IF;
END; // -- check with on delete cascade
DELIMITER ;

-- show triggers;

-- delete from Wrote where BookID=1;

-- delete from Books where BookID=1;

-- select * from Wrote;

-- select * from Books;
```

```
-- ------------------------------------------------------------------------
-- File:    StoredProcedureExample.sql
-- Author:  Chadd Williams
-- Date:    11/17/2017
-- Class:   CS445
-- Purpose: Demonstrate Stored Procedures
-- ------------------------------------------------------------------------

DROP PROCEDURE IF EXISTS AddTo250;
DELIMITER //
CREATE PROCEDURE
    AddTo250 (parameter_StudentID INTEGER)
    MODIFIES SQL DATA
    BEGIN
        DELETE FROM CurrentlyEnrolled where
            StudentID=parameter_StudentID and CourseID=1;
        INSERT INTO CurrentlyEnrolled (CourseID, StudentID)
            VALUES(2, parameter_StudentID);
    END ;
//
DELIMITER ;


-- INSERT INTO CurrentlyEnrolled (CourseID, StudentID) VALUES (1, 5);

-- select * from CurrentlyEnrolled Where StudentID=5;

-- CALL AddTo250(5);

-- select * from CurrentlyEnrolled Where StudentID=5;

-- SHOW CREATE PROCEDURE AddTo250;

-- SHOW PROCEDURE STATUS;
```