SQL Basics

- Sqlite3 command line tool
- SQL Syntax
- Creating Tables, Columns, and Rows
- Primary and Foreign Keys
- Constraints

Advanced Python Programming

See Also

https://www.youtube.com/watch?v=U7nfe4adDw8

https://www.sqlite.org/about.html

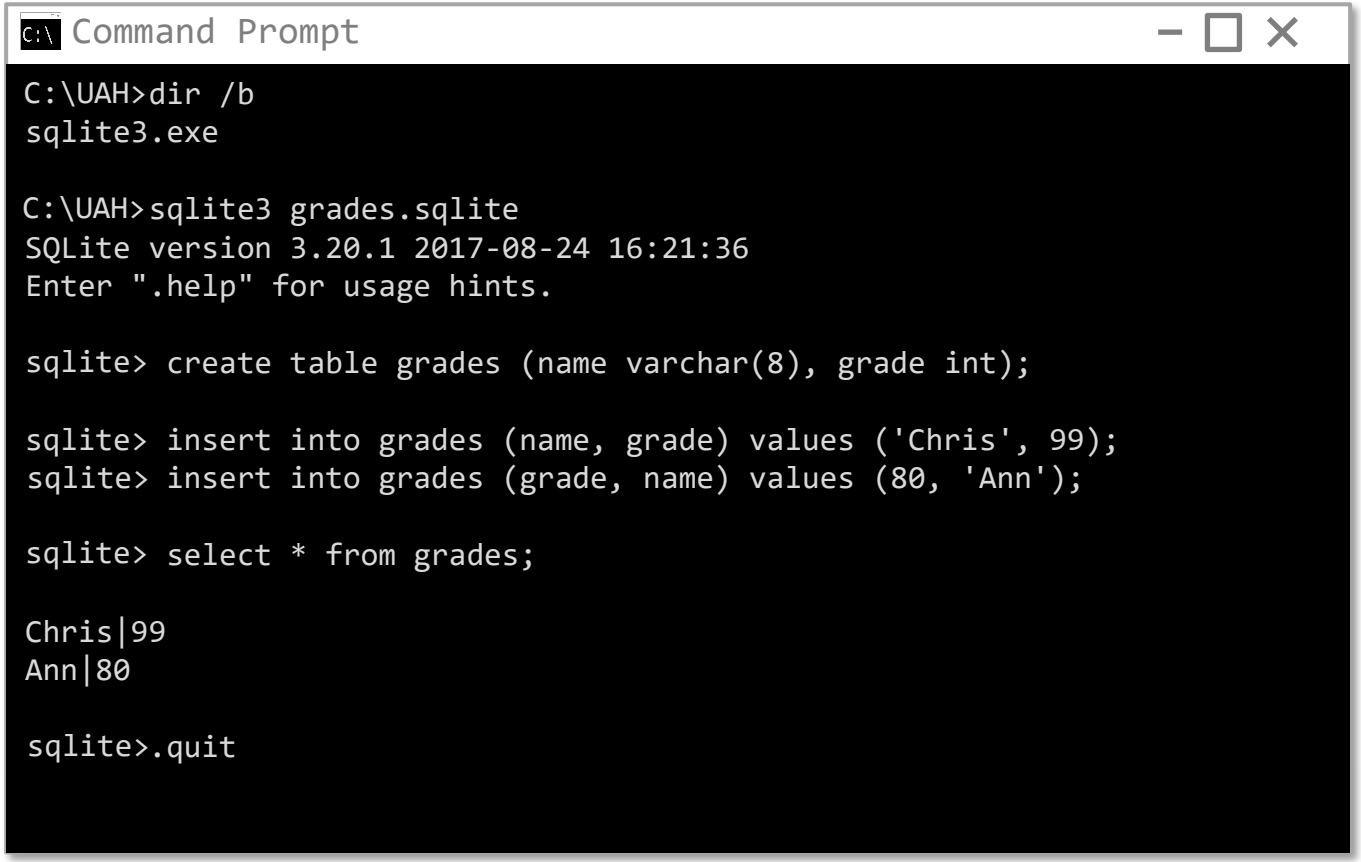
http://www.w3schools.com/sql/

Embedded Database

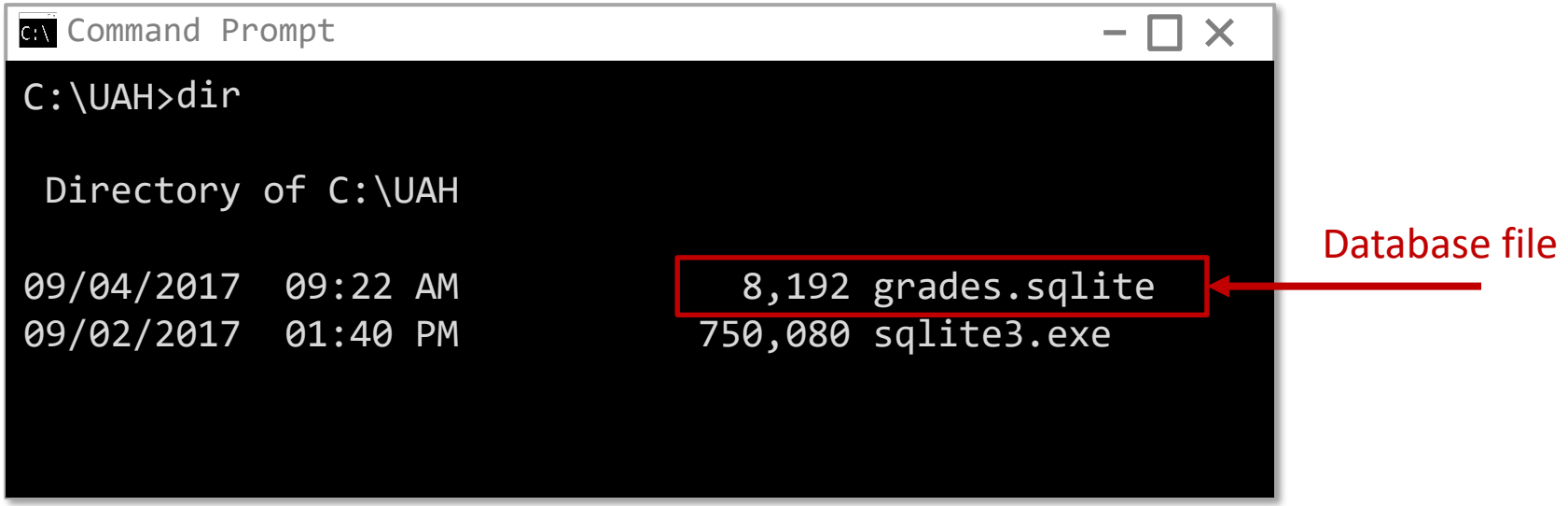
- SQLite runs embedded within your process
- Popular for “hidden” use in a program
  - Windows 10
  - Skype
  - Firefox
  - TurboTax and QuickBooks

- Other non-embedded (stand-alone server) databases:
  - Oracle
  - PostgreSQL
  - MySQL

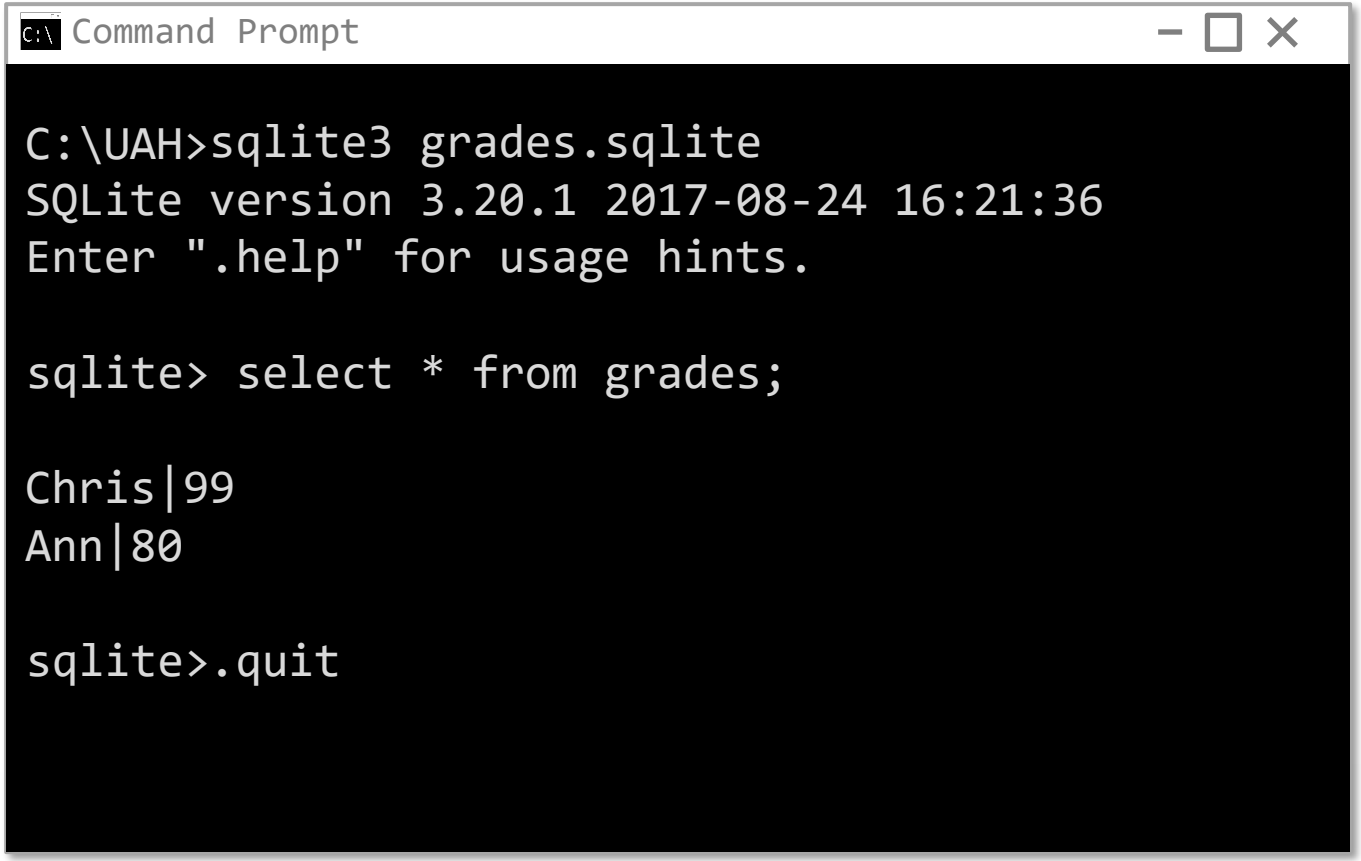
SQL Command Line



SQL Command Line



SQL Command Line



SQL Basics

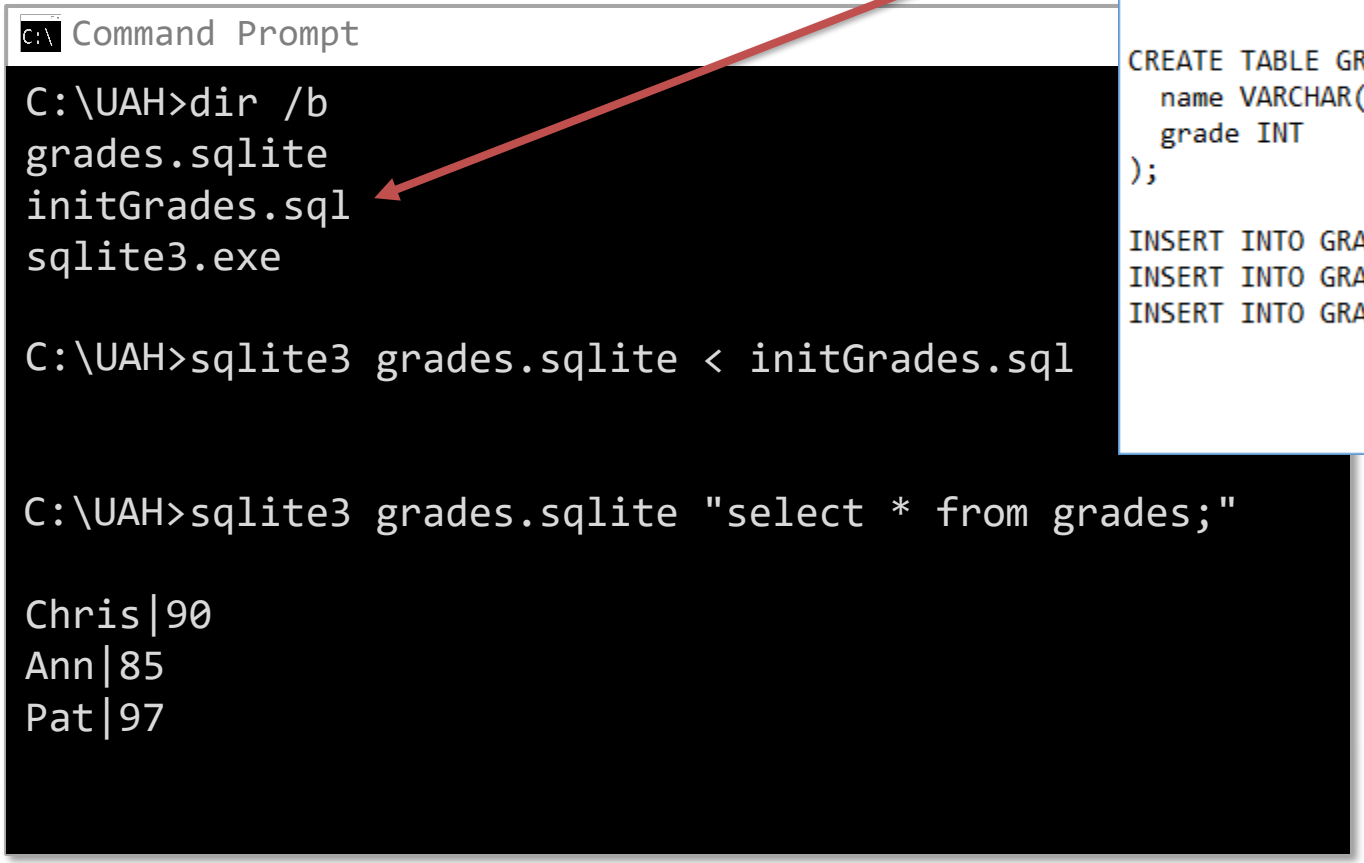
```
DROP TABLE grades;

CREATE TABLE grades (
  name VARCHAR(20), -- up to 20 characters
  grade INT         -- column name and type
);

INSERT INTO grades
  (grade,name)           -- columns to set
  VALUES (90,'Chris');   -- values in same order

INSERT INTO grades VALUES ('Ann',85);

SELECT * FROM grades; -- all columns, all rows

SELECT name FROM grades; -- name column from all rows
```

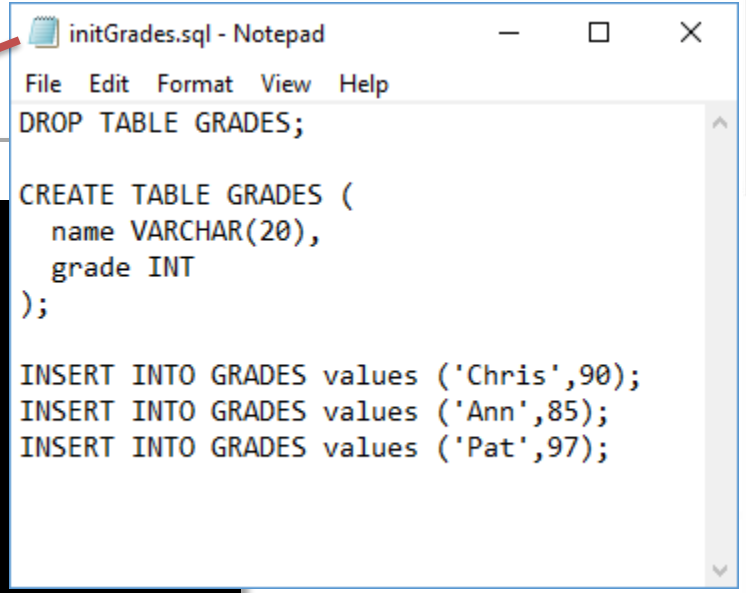
- -- comments
- Case INsensitive
- Use multi-lines as needed
- End commands with ;

Status | Result1

	NAME	GRADE
1	Chris	90
2	Ann	85

SQL Scripts


Command Prompt
C:\UAH>dir /b
grades.sqlite
initGrades.sql
sqlite3.exe
C:\UAH>sqlite3 grades.sqlite < initGrades.sql
C:\UAH>sqlite3 grades.sqlite "select * from grades;"
Chris|90
Ann|85
Pat|97


initGrades.sql - Notepad
File Edit Format View Help
DROP TABLE GRADES;
CREATE TABLE GRADES (
name VARCHAR(20),
grade INT
);
INSERT INTO GRADES values ('Chris',90);
INSERT INTO GRADES values ('Ann',85);
INSERT INTO GRADES values ('Pat',97);

SQL Basics

```
SELECT * FROM GRADES;

SELECT name, grade FROM GRADES
  WHERE name='Chris';

-- only rows that match 'Chris'
UPDATE GRADES
  SET grade=95
  WHERE name='Chris';

SELECT * FROM GRADES;

UPDATE GRADES
  SET grade=100;

SELECT * FROM GRADES;
```

name	grade
Chris	90
Ann	85
Pat	97

name	grade
Chris	90

name	grade
Chris	95
Ann	85
Pat	97

name	grade
Chris	100
Ann	100
Pat	100

STUDENTS Table

```
CREATE TABLE STUDENTS (
  name VARCHAR(20),
  phone VARCHAR(8)
);

INSERT INTO STUDENTS VALUES ('Chris','1234');
INSERT INTO STUDENTS VALUES ('Ann','9999');
INSERT INTO STUDENTS VALUES ('Pat','0110')

SELECT * FROM STUDENTS;
```

name	phone
Chris	1234
Ann	9999
Pat	0110

Joins

```
SELECT * FROM GRADES, STUDENTS;

SELECT * FROM GRADES, STUDENTS
 WHERE GRADES.name = STUDENTS.name;

SELECT GRADES.grade, GRADES.name, STUDENTS.phone
  FROM GRADES, STUDENTS
  WHERE GRADES.name = STUDENTS.name;

SELECT GRADES.grade, GRADES.name, STUDENTS.phone
  FROM GRADES, STUDENTS
  WHERE GRADES.name = STUDENTS.name
  ORDER BY GRADES.grade DESC;
```

name	grade	name	phone
Chris	90	Chris	1234
Chris	90	Ann	9999
Chris	90	Pat	0110
Ann	85	Chris	1234
Ann	85	Ann	9999
Ann	85	Pat	0110
Pat	97	Chris	1234
Pat	97	Ann	9999
Pat	97	Pat	0110

name	grade	name	phone
Chris	90	Chris	1234
Ann	85	Ann	9999
Pat	97	Pat	0110

grade	name	phone
97	Pat	0110
90	Chris	1234
85	Ann	9999

grade	name	phone
90	Chris	1234
85	Ann	9999
97	Pat	0110

Generated Primary Keys

```
DROP TABLE STUDENTS;

CREATE TABLE STUDENTS (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name VARCHAR(20) NOT NULL,
  phone VARCHAR(8)
);

INSERT INTO STUDENTS (name,phone) VALUES ('Chris','1234');
INSERT INTO STUDENTS (name,phone) VALUES ('Ann','9999');
INSERT INTO STUDENTS (name,phone) VALUES ('Pat','0110');
```

name	phone
Chris	1234
Ann	9999
Pat	0110

Foreign Keys

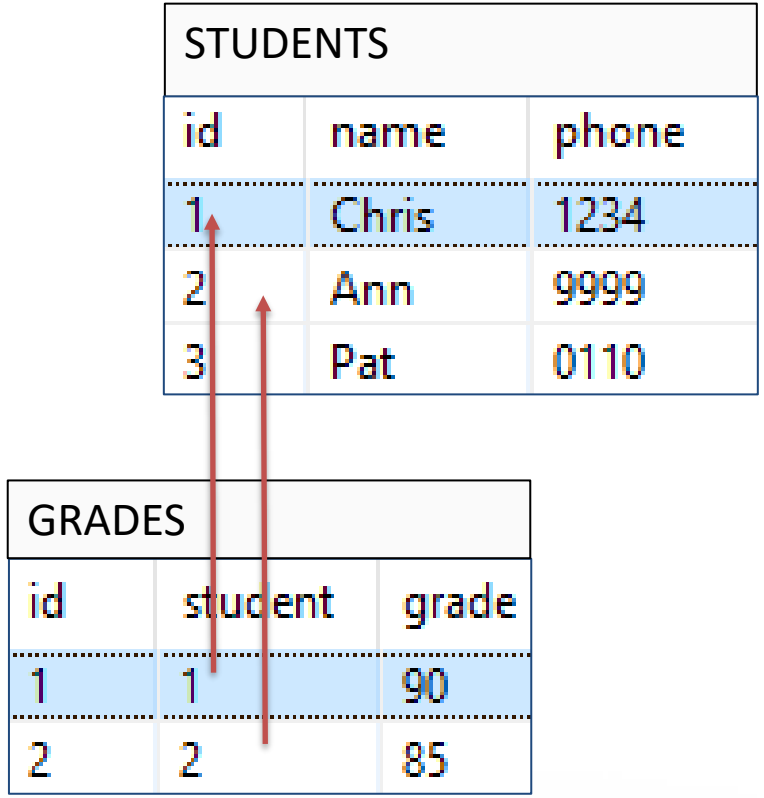
```
DROP TABLE GRADES;

PRAGMA foreign_keys = ON;

CREATE TABLE GRADES (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  student INT NOT NULL,
  grade INT,
  FOREIGN KEY(student) REFERENCES STUDENTS(id)
);

INSERT INTO GRADES (student,grade) values (1,90);
INSERT INTO GRADES (student,grade) values (2,85);

INSERT INTO GRADES (student,grade) values (null,75);
```

```
Error: NOT NULL constraint failed: GRADES.student
```

```
INSERT INTO GRADES (student,grade) values (123,75);
```

```
Error: FOREIGN KEY constraint failed
```

SQL Lite Specific



Other Constraints

```
DROP TABLE GRADES;

CREATE TABLE GRADES (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  student INT NOT NULL,
  grade INT NOT NULL CHECK (grade>=0),
  FOREIGN KEY(student) REFERENCES STUDENTS(id)
);


INSERT INTO GRADES (student,grade) values (1,90);
INSERT INTO GRADES (student,grade) values (2,85);

INSERT INTO GRADES (student,grade) values (1,-15);
```

```
Error: CHECK constraint failed: GRADES
```

Tinkering

- Use SQLITE3 to create the GRADES/STUDENTS database.
- Add column “SUBJECT” to GRADES and insert lots of grades for several students.
- Run a query to find everyone who failed a “Science” test.