# Databases

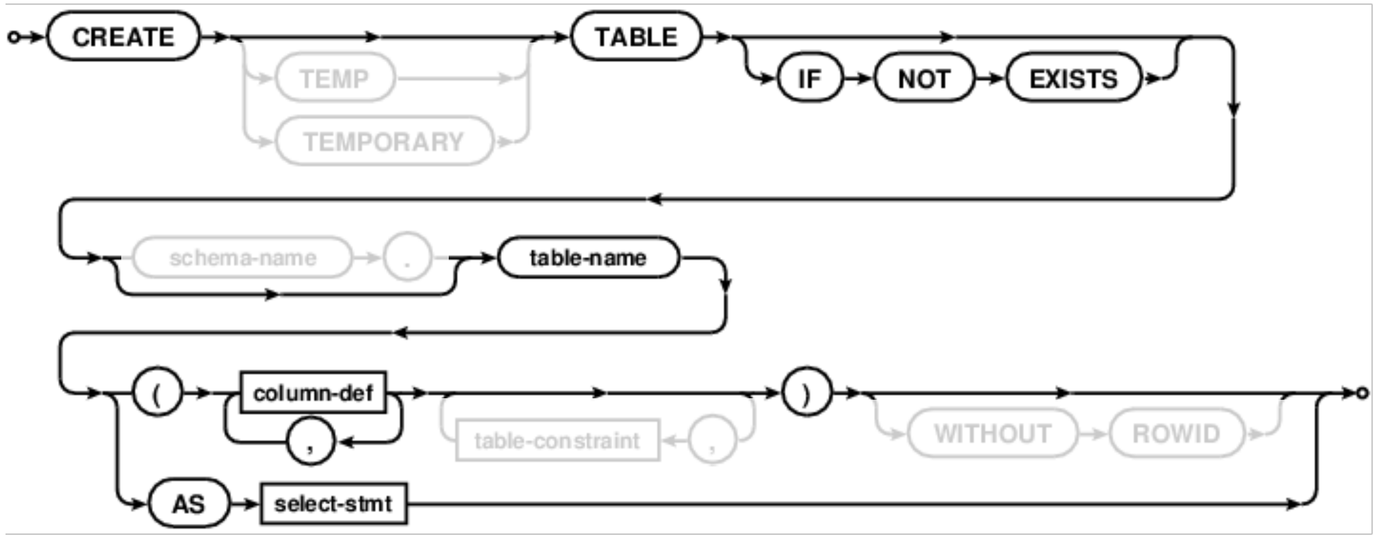
# Announcements

# Aggregation Review

# Create Table and Drop Table

# Create Table

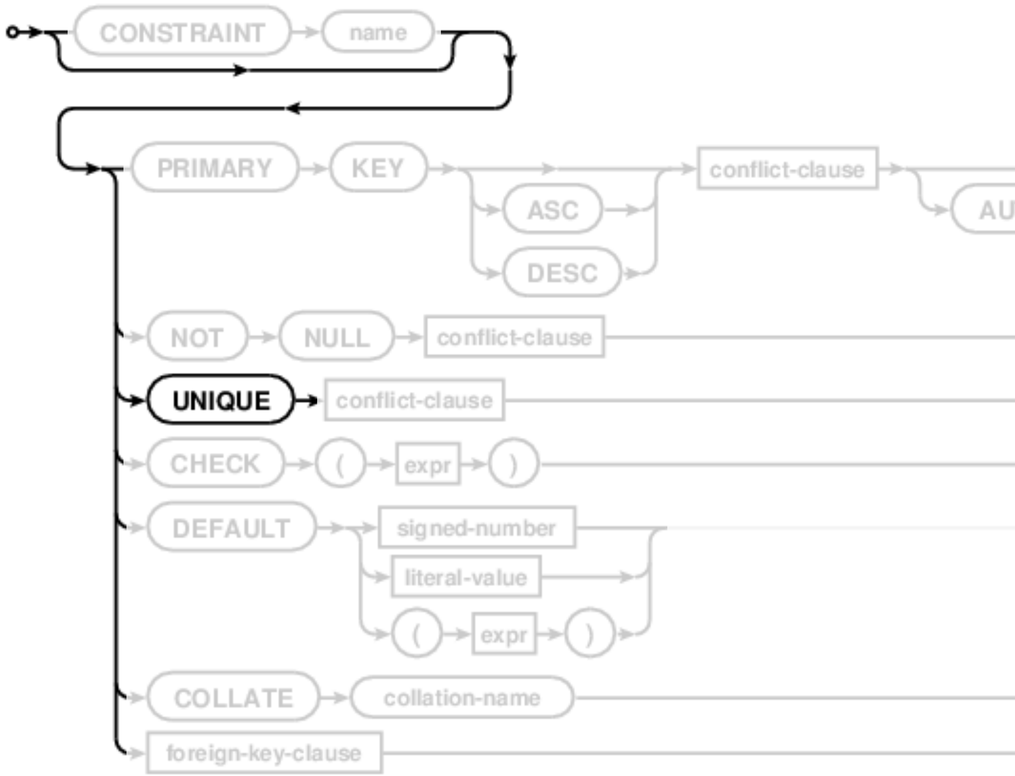
CREATE TABLE expression syntax:




column-def:




column-constraint:



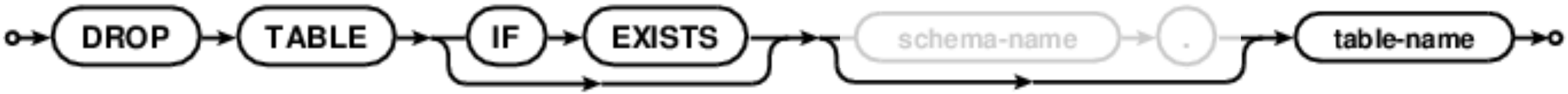

Examples:

**CREATE TABLE users (id, first, last, email, role);**

**CREATE TABLE users (id UNIQUE, first, last, email, role);**

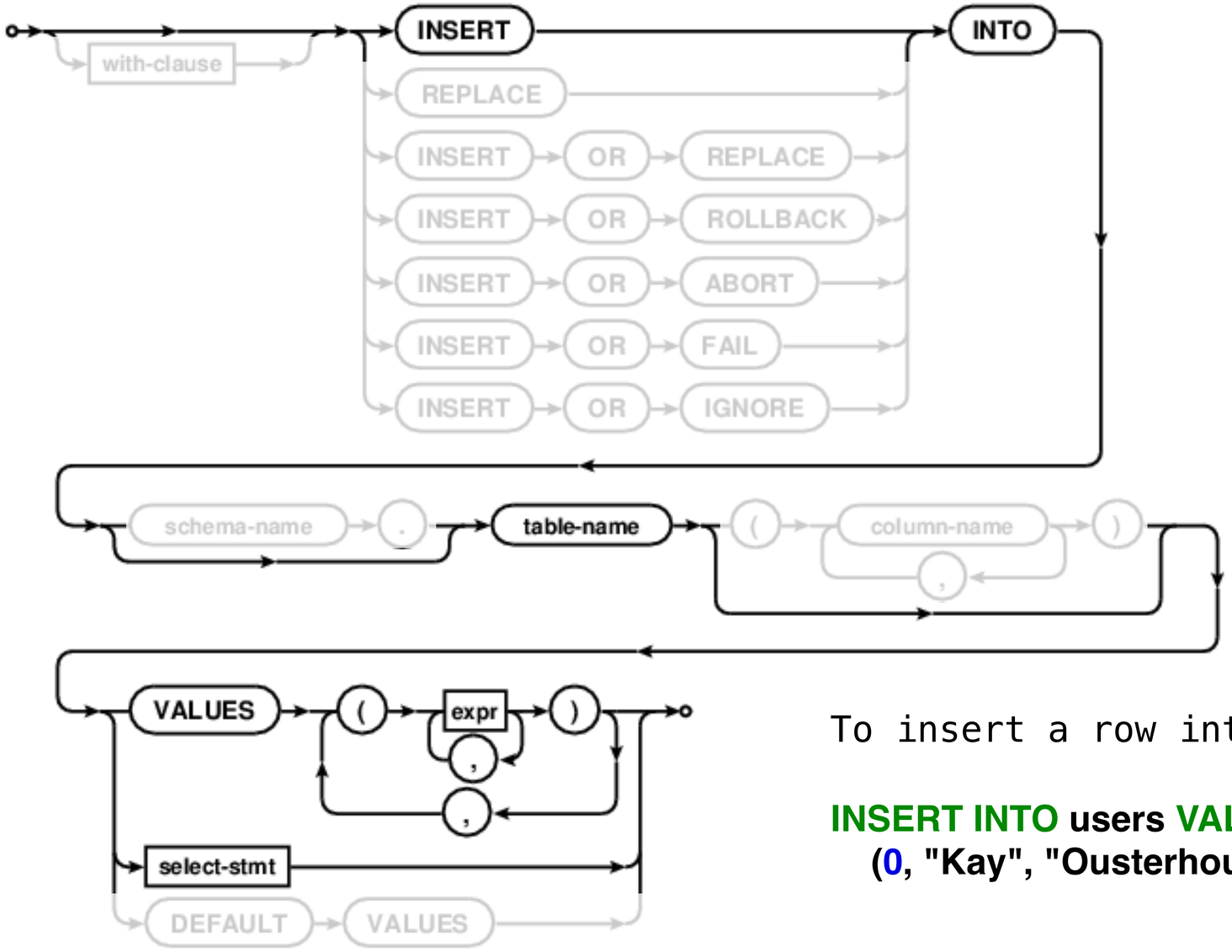
(Demo)

# Drop Table



# Modifying Tables

# Insert

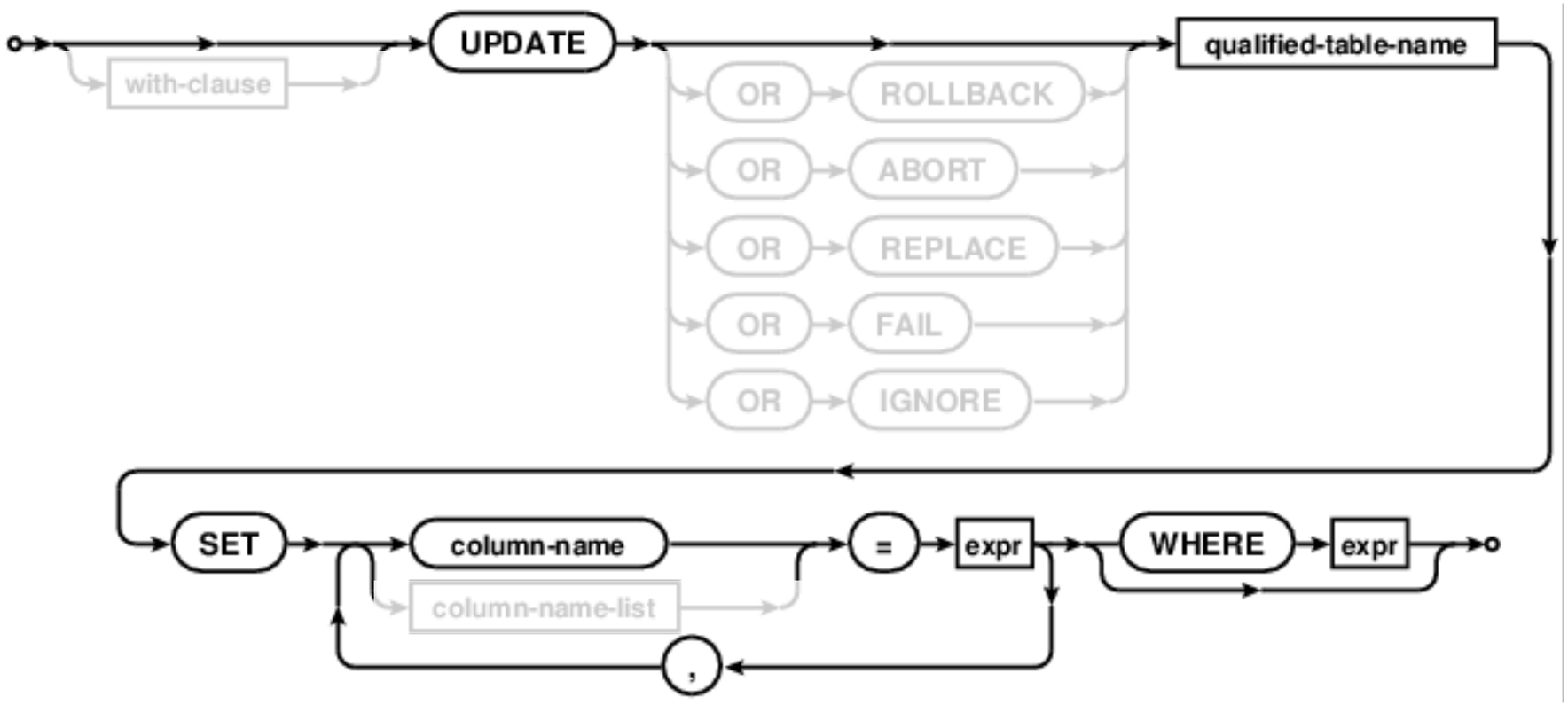



To insert a row into a users table with 5 columns:

**INSERT INTO users VALUES**
**(0, "Kay", "Ousterhout", "kayo@berkeley.edu", "Instructor");**

(Demo)

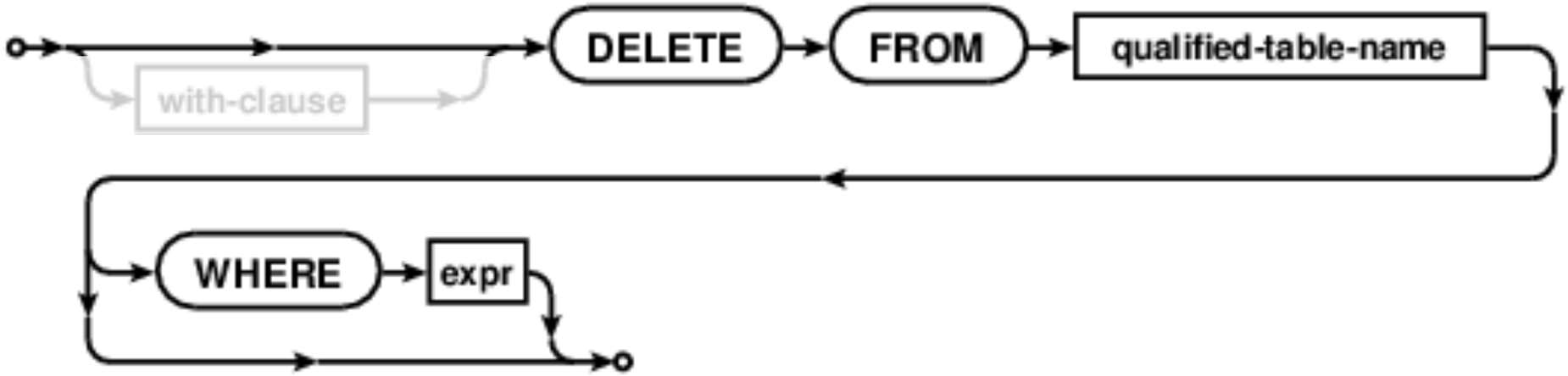
# Update




Update sets all entries in certain columns to new values, just for some subset of rows.

(Demo)

# Delete




Delete removes some or all rows from a table.

(Demo)

# Python and SQL

# Python Can Access Sqlite Databases

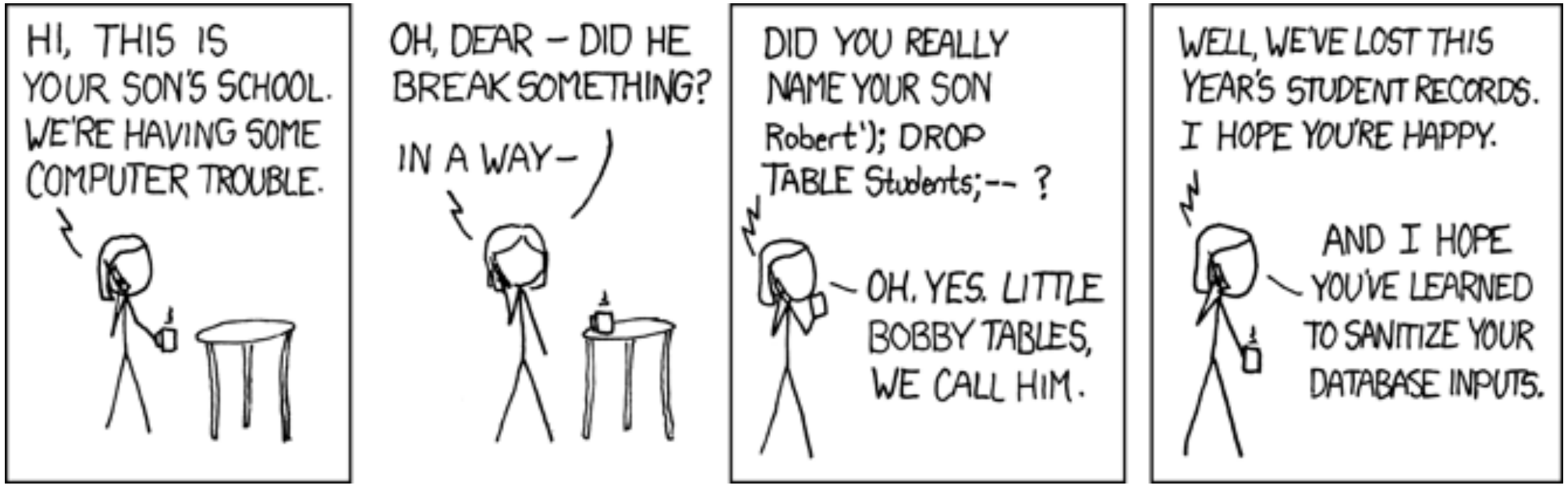
```
import sqlite3

db = sqlite3.Connection('users.db')

db.execute('CREATE TABLE IF NOT EXISTS users (id UNIQUE, first, last, email, role);')

cmd = f"INSERT INTO users VALUES (0, 'Kay', 'Ousterhout', 'kayo@berkeley.edu',
'Instructor');"
db.executescript(cmd);

users = db.execute("SELECT * from users").fetchall()
for user in users:
    print(user)
```

(Demo)

# SQL Injection Attack

(Demo)

# A Program Vulnerable to a SQL Injection Attack




```
name = "Robert'); DROP TABLE Students; --"
cmd = "INSERT INTO Students VALUES ('" + name + "');"
db.executescript(cmd)
```

```
db.execute("INSERT INTO Students VALUES (?)", [name])
```

SQLite makes a query plan before substitution happens

SQLite gets parameters separately

SQLite gets a string:

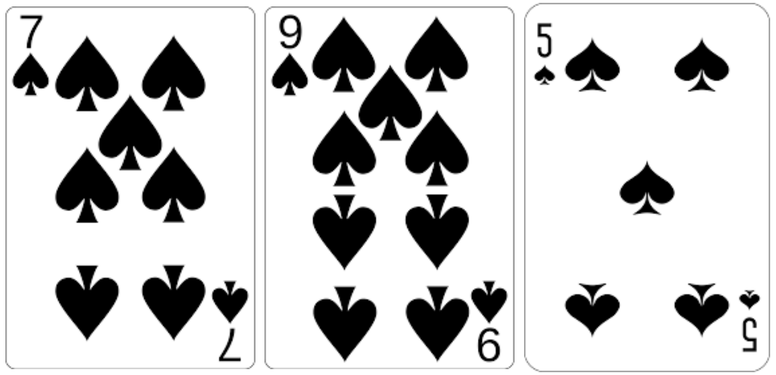
```
INSERT INTO Students VALUES ('Robert'); DROP TABLE Students; --');
```

https://xkcd.com/327/

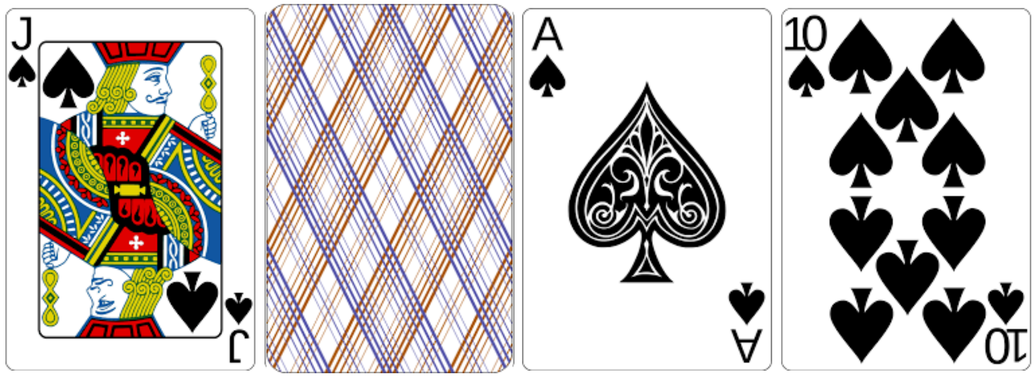
# Blackjack

# Casino Blackjack

**Player:**



**Dealer:**



(Demo)