

# SQL: Database Access

Luís Pedro Coelho

Programming for Scientists

April 2, 2009



University of Pittsburgh

**Carnegie Mellon**

## Problem

- Many different “experiments”.
- Each experiment has input parameters and output measurements.

Experiments with input parameters and output values.

Experiments with input parameters and output values.

Tables

- 1 Experiments
- 2 Experiment parameters
- 3 Experiment results

# SQL For Table Creation

```
CREATE TABLE experiments (  
    eid INT PRIMARY KEY,  
    name VARCHAR(255),  
    date DATE NOT NULL  
);  
  
CREATE TABLE parameters (  
    pid INT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE experiment_parameters (  
    eid INT NOT NULL,  
    pid INT NOT NULL,  
    paramvalue VARCHAR(255)  
);  
  
CREATE TABLE experiment_results (  
    eid INT NOT NULL,  
    measure VARCHAR(255).
```

## Example Query

```
SELECT name, date
FROM experiments, parameters, experiment_parameters
WHERE experiments.eid = experiment_parameters.eid AND
      experiment_parameters.pid = parameters.pid AND
      parameters.name = 'Generation/model' AND
      experiment_parameters.paramvalue = 'brownian';
```

# A Traditional Database Management System

- 1 Install it.
- 2 Set it up to run as daemon.
- 3 Create database(s).
- 4 Create user(s).
- 5 Create tables.
- 6 Use it.

# SQLite: Database in a File

SQLite implements SQL-like databases in a single file.

# Using SQLite

```
import sqlite3
from os.path import exists
filename = 'experiment.db'
create = not exists(filename)
connection = sqlite3.connect(filename)
cursor = connection.cursor()
if create:
    cursor.execute('''
        CREATE TABLE ...''')
```

## Using SQLite (II)

```
cursor.execute(""" SELECT eid,name,date FROM experiments WHERE  
date < DATE('2008-12-01') """)  
for eid,name,date in cursor.fetchall(): print eid, name, date
```

A different type of database,  
based on **HDF5**.