

## **Lab Members: Lucy Shichman, John Le, Shannon Tran, Umar Abushaban, Graham Twente**

### **WORKSHEET**

Names: Avneet Chhabra, John Le, Lucy Shichman, Graham Twente, Shannon Tran, Umar Abushaban

#### **1. What types of information does Firefox manage using SQLite? Briefly explain the contents of two or three databases.**

The places database contains 22 tables, such as "moz\_bookmarks" and "moz\_historyvisists." These tables contain variables that keep track of browser history and some of the bookmarks that we have created within our Firefox browsers.

The permissions database contains a "moz\_hosts" and "moz\_perms" table, which contains a couple variables in each of them such as id, permission, expire time, and modification time.

The storage database contains a "cache" and a "database" table, which tracks some cache information.

The cookies database contains a single "moz\_cookies" table, containing information about the name, value, host, path, etc of cookies.

#### **2. In what database and tables are your bookmarks stored? Why do you think Firefox stores the title and url separately?**

Bookmarks are stored in the places.sqlite database, in the "moz\_bookmarks" table. The URLs are stored in the "moz\_places" table, where each URL visited is given a unique ID. That id operates as a foreign key in the moz\_bookmarks table when it wants to reference a bookmark to its corresponding URL. This is probably so that if you remove or change a particular bookmark, it can update/delete the foreign key or row from moz\_bookmarks without deleting the browsing history from moz\_places.

#### **3. Write an SQL statement that selects your bookmarks. For each one, display only the title, url, date added (as an integer), and visit count.**

```
SELECT moz_bookmarks.title, moz_bookmarks.dateAdded, moz_places.url,  
moz_places.visit_count  
FROM moz_bookmarks, moz_places  
WHERE moz_bookmarks.id=moz_places.id;
```

#### **4. Write an SQL statement that lists the base domain, name, and value for each of your cookies in order of expiration date.**

```
SELECT host, name, value
FROM moz_cookies
ORDER BY expiry DESC
```

**5. Write an SQL statement to show how many cookies you have for each domain. Note that you will need to use the "count" function instead of sum.**

```
SELECT COUNT (*)
FROM moz_cookies
GROUP BY host;
```

**6. What are the top five movies since the year 2000, in terms of adjusted gross income?**

```
select * FROM Movie WHERE (year > 2000) ORDER BY adjusted DESC LIMIT 5
```

10	Avatar	Fox	2009	760.51
24	Marvel's The Avengers	Disney	2012	623.36
25	The Dark Knight	Warner Bros.	2008	534.86
28	Shrek 2	DreamWorks	2004	441.23
32	Spider-Man	Sony / Columbia	2002	403.71

7. What are the title, author, and year of books from Russia in the top 100?  
(For convenience, you can right-click the results and "Copy Rows as CSV.")

```
SELECT title, author, year
FROM book
WHERE country = 'Russia';
```

title	author	year
Stories	Anton Chekhov	1886
Crime and Punishment	Fyodor Dostoevsky	1866
The Idiot	Fyodor Dostoevsky	1869
The Possessed	Fyodor Dostoevsky	1872
The Brothers Karamazov	Fyodor Dostoevsky	1880
Dead Souls	Nikolai Gogol	1842
War and Peace	Leo Tolstoy	1865?1869
Anna Karenina	Leo Tolstoy	1877
The Death of Ivan Ilyich	Leo Tolstoy	1886

**8. What are the names and symbols of the periodic elements with an atomic mass of more than 280? Display the results from lightest to heaviest.**

```
SELECT name, symbol
FROM periodic
WHERE mass > 280
ORDER BY mass;
```

name	symbol
Ununtrium	Uut
Ununbium	Uub
Ununpentium	Uup
Ununquadium	Uuq
Ununhexium	Uuh
Ununoctium	Uuo

**9. Which of the top 15 songs were featured in the top 100 movies? (Hint: Figure out how to join the two tables, and then use AND clauses to filter by rank.)**

```
SELECT song.film, song.title
from song
Inner Join movie on song.film = movie.title where song.rank <16 and
movie.rank<101
```

The Graduate	Mrs. Robinson
Pinocchio	When You Wish upon a Star
The Sound of Music	The Sound of Music
Titanic	My Heart Will Go On

**10. Do your Python Lab**

```

1  # create data base connection
2  import sqlite3
3  conn = sqlite3.connect('orders.db')
4  # create cursor object
5  cur = conn.cursor() # execute with cur.execute()
6  # creating tables in SQLite for Python
7  cur.execute("""CREATE TABLE IF NOT EXISTS users(
8      userid INT PRIMARY KEY,
9      fname TEXT,
10     lname TEXT,
11     gender TEXT);
12 """)
13 conn.commit()
14 cur.execute("""CREATE TABLE IF NOT EXISTS orders(
15     orderid INT PRIMARY KEY,
16     date TEXT,
17     userid TEXT,
18     total TEXT);
19 """)
20 conn.commit()
21 # adding data
22 cur.execute("""INSERT INTO users(userid, fname, lname, gender)
23     VALUES('00001', 'Nik', 'Piepenbreier', 'male');""")
24 conn.commit()
25 user = ('00002', 'Lois', 'Lane', 'Female')
26 cur.execute("INSERT INTO users VALUES(?, ?, ?, ?);", user)
27 conn.commit()
28 more_users = [('00003', 'Peter', 'Parker', 'Male'), ('00004', 'Bruce', 'Wayne', 'male')]
29 cur.executemany("INSERT INTO users VALUES(?, ?, ?, ?);", more_users)
30 conn.commit()
31 # load more meaningful data
32 customers = [('00005', 'Stephanie', 'Stewart', 'female'), ('00006', 'Sincere', 'Sherman', 'female'), ('00007', 'Sidney', 'Horn'),
33 orders = [('00001', '2020-01-01', '00025', '178'), ('00002', '2020-01-03', '00025', '39'), ('00003', '2020-01-07', '00016', '178'), ('00004', '2020-01-07', '00016', '178')]
34 cur.executemany("INSERT INTO users VALUES(?, ?, ?, ?);", customers)
35 cur.executemany("INSERT INTO orders VALUES(?, ?, ?, ?);", orders)
36 conn.commit()
37 # selecting data
38 cur.execute("SELECT * FROM users;")
39 one_result = cur.fetchone()
40 print(one_result)
41 cur.execute("SELECT * FROM users;")
42 three_results = cur.fetchmany(3)
43 print(three_results)
44 cur.execute("SELECT * FROM users;")
45 all_results = cur.fetchall()
46 print(all_results) # returns error
47 # deleting data
48 cur.execute("DELETE FROM users WHERE lname='Parker';")
49 conn.commit()
50 cur.execute("select * from users where lname='Parker'")
51 print(cur.fetchall())
52 # joining tables
53 cur.execute("""SELECT *, users.fname, users.lname FROM orders
54     LEFT JOIN users ON users.userid=orders.userid;""")
55 print(cur.fetchall())

```

\*\*Used under the open commons from JMU CS101 Labs