# SQLite3 J. Strait

April 20, 2020

## 1 SQLite Exercise

SQLite is a simple way to implement SQL in Python. It can work off of a database file (.db) that already has tables in it or an empty one. You can also insert tables into a .db file. We are going to to both!

## 2 Libraries

Two very important libraries we will need are pandas, and extremely popular data analysis library, and of course SQLite3.

[1]:
```
import sqlite3
import pandas as pd
```

## 3 Connecting to a Database File

- Need to establish a connection: conn = sqlite3.connect('insert database file name here')
- Need to establish a cursor object to execute queries: c = conn.cursor()
- To execute queries: c.execute('insert SQL statement here')

### 3.1 Let's Explore Chinook!

To see the database diagram follow this link: http://www.sqlitetutorial.net/wp-content/uploads/2018/03/sqlite-sample-database-diagram-color.pdf

[2]:
```
conn = sqlite3.connect('chinook.db') #Note: this file should be located in the␣
 ↪same folder you are running this notebook from!
c = conn.cursor()
```

To display query results, we can use a simple for loop. Run the next cell to see how to display the information in the playlists table.

[3]:
```
for row in c.execute('SELECT * FROM playlists'):
    print(row)
```

```
(1, 'Music')
(2, 'Movies')
(3, 'TV Shows')
```

```
(4, 'Audiobooks')
(5, '90s Music')
(6, 'Audiobooks')
(7, 'Movies')
(8, 'Music')
(9, 'Music Videos')
(10, 'TV Shows')
(11, 'Brazilian Music')
(12, 'Classical')
(13, 'Classical 101 - Deep Cuts')
(14, 'Classical 101 - Next Steps')
(15, 'Classical 101 - The Basics')
(16, 'Grunge')
(17, 'Heavy Metal Classic')
(18, 'On-The-Go 1')
```

Now, use the next cell to write a query that displays all of the customers from the customers table

[4]:
```
for row in c.execute('SELECT * FROM customers'):
    print(row)
```

```
(1, 'Luís', 'Gonçalves', 'Embraer - Empresa Brasileira de Aeronáutica S.A.',
'Av. Brigadeiro Faria Lima, 2170', 'São José dos Campos', 'SP', 'Brazil',
'12227-000', '+55 (12) 3923-5555', '+55 (12) 3923-5566', 'luisg@embraer.com.br',
3)
(2, 'Leonie', 'Köhler', None, 'Theodor-Heuss-Straße 34', 'Stuttgart', None,
'Germany', '70174', '+49 0711 2842222', None, 'leonekohler@surfeu.de', 5)
(3, 'François', 'Tremblay', None, '1498 rue Bélanger', 'Montréal', 'QC',
'Canada', 'H2G 1A7', '+1 (514) 721-4711', None, 'ftremblay@gmail.com', 3)
(4, 'Bjørn', 'Hansen', None, 'Ullevålsveien 14', 'Oslo', None, 'Norway', '0171',
'+47 22 44 22 22', None, 'bjorn.hansen@yahoo.no', 4)
(5, 'Frantiek', 'Wichterlová', 'JetBrains s.r.o.', 'Klanova 9/506', 'Prague',
None, 'Czech Republic', '14700', '+420 2 4172 5555', '+420 2 4172 5555',
'frantisekw@jetbrains.com', 4)
(6, 'Helena', 'Holý', None, 'Rilská 3174/6', 'Prague', None, 'Czech Republic',
'14300', '+420 2 4177 0449', None, 'hholy@gmail.com', 5)
(7, 'Astrid', 'Gruber', None, 'Rotenturmstraße 4, 1010 Innere Stadt', 'Vienne',
None, 'Austria', '1010', '+43 01 5134505', None, 'astrid.gruber@apple.at', 5)
(8, 'Daan', 'Peeters', None, 'Grétrystraat 63', 'Brussels', None, 'Belgium',
'1000', '+32 02 219 03 03', None, 'daan_peeters@apple.be', 4)
(9, 'Kara', 'Nielsen', None, 'Sønder Boulevard 51', 'Copenhagen', None,
'Denmark', '1720', '+453 3331 9991', None, 'kara.nielsen@jubii.dk', 4)
(10, 'Eduardo', 'Martins', 'Woodstock Discos', 'Rua Dr. Falcão Filho, 155', 'São
Paulo', 'SP', 'Brazil', '01007-010', '+55 (11) 3033-5446', '+55 (11) 3033-4564',
'eduardo@woodstock.com.br', 4)
(11, 'Alexandre', 'Rocha', 'Banco do Brasil S.A.', 'Av. Paulista, 2022', 'São
Paulo', 'SP', 'Brazil', '01310-200', '+55 (11) 3055-3278', '+55 (11) 3055-8131',
'alero@uol.com.br', 5)
```

```
(12, 'Roberto', 'Almeida', 'Riotur', 'Praça Pio X, 119', 'Rio de Janeiro', 'RJ',
'Brazil', '20040-020', '+55 (21) 2271-7000', '+55 (21) 2271-7070',
'roberto.almeida@riotur.gov.br', 3)
(13, 'Fernanda', 'Ramos', None, 'Qe 7 Bloco G', 'Brasília', 'DF', 'Brazil',
'71020-677', '+55 (61) 3363-5547', '+55 (61) 3363-7855',
'fernadaramos4@uol.com.br', 4)
(14, 'Mark', 'Philips', 'Telus', '8210 111 ST NW', 'Edmonton', 'AB', 'Canada',
'T6G 2C7', '+1 (780) 434-4554', '+1 (780) 434-5565', 'mphilips12@shaw.ca', 5)
(15, 'Jennifer', 'Peterson', 'Rogers Canada', '700 W Pender Street',
'Vancouver', 'BC', 'Canada', 'V6C 1G8', '+1 (604) 688-2255', '+1 (604)
688-8756', 'jenniferp@rogers.ca', 3)
(16, 'Frank', 'Harris', 'Google Inc.', '1600 Amphitheatre Parkway', 'Mountain
View', 'CA', 'USA', '94043-1351', '+1 (650) 253-0000', '+1 (650) 253-0000',
'fharris@google.com', 4)
(17, 'Jack', 'Smith', 'Microsoft Corporation', '1 Microsoft Way', 'Redmond',
'WA', 'USA', '98052-8300', '+1 (425) 882-8080', '+1 (425) 882-8081',
'jacksmith@microsoft.com', 5)
(18, 'Michelle', 'Brooks', None, '627 Broadway', 'New York', 'NY', 'USA',
'10012-2612', '+1 (212) 221-3546', '+1 (212) 221-4679', 'michelleb@aol.com', 3)
(19, 'Tim', 'Goyer', 'Apple Inc.', '1 Infinite Loop', 'Cupertino', 'CA', 'USA',
'95014', '+1 (408) 996-1010', '+1 (408) 996-1011', 'tgoyer@apple.com', 3)
(20, 'Dan', 'Miller', None, '541 Del Medio Avenue', 'Mountain View', 'CA',
'USA', '94040-111', '+1 (650) 644-3358', None, 'dmiller@comcast.com', 4)
(21, 'Kathy', 'Chase', None, '801 W 4th Street', 'Reno', 'NV', 'USA', '89503',
'+1 (775) 223-7665', None, 'kachase@hotmail.com', 5)
(22, 'Heather', 'Leacock', None, '120 S Orange Ave', 'Orlando', 'FL', 'USA',
'32801', '+1 (407) 999-7788', None, 'hleacock@gmail.com', 4)
(23, 'John', 'Gordon', None, '69 Salem Street', 'Boston', 'MA', 'USA', '2113',
'+1 (617) 522-1333', None, 'johngordon22@yahoo.com', 4)
(24, 'Frank', 'Ralston', None, '162 E Superior Street', 'Chicago', 'IL', 'USA',
'60611', '+1 (312) 332-3232', None, 'fralston@gmail.com', 3)
(25, 'Victor', 'Stevens', None, '319 N. Frances Street', 'Madison', 'WI', 'USA',
'53703', '+1 (608) 257-0597', None, 'vstevens@yahoo.com', 5)
(26, 'Richard', 'Cunningham', None, '2211 W Berry Street', 'Fort Worth', 'TX',
'USA', '76110', '+1 (817) 924-7272', None, 'ricunningham@hotmail.com', 4)
(27, 'Patrick', 'Gray', None, '1033 N Park Ave', 'Tucson', 'AZ', 'USA', '85719',
'+1 (520) 622-4200', None, 'patrick.gray@aol.com', 4)
(28, 'Julia', 'Barnett', None, '302 S 700 E', 'Salt Lake City', 'UT', 'USA',
'84102', '+1 (801) 531-7272', None, 'jubarnett@gmail.com', 5)
(29, 'Robert', 'Brown', None, '796 Dundas Street West', 'Toronto', 'ON',
'Canada', 'M6J 1V1', '+1 (416) 363-8888', None, 'robbrown@shaw.ca', 3)
(30, 'Edward', 'Francis', None, '230 Elgin Street', 'Ottawa', 'ON', 'Canada',
'K2P 1L7', '+1 (613) 234-3322', None, 'edfrancis@yachoo.ca', 3)
(31, 'Martha', 'Silk', None, '194A Chain Lake Drive', 'Halifax', 'NS', 'Canada',
'B3S 1C5', '+1 (902) 450-0450', None, 'marthasilk@gmail.com', 5)
(32, 'Aaron', 'Mitchell', None, '696 Osborne Street', 'Winnipeg', 'MB',
'Canada', 'R3L 2B9', '+1 (204) 452-6452', None, 'aaronmitchell@yahoo.ca', 4)
(33, 'Ellie', 'Sullivan', None, '5112 48 Street', 'Yellowknife', 'NT', 'Canada',
'X1A 1N6', '+1 (867) 920-2233', None, 'ellie.sullivan@shaw.ca', 3)
(34, 'João', 'Fernandes', None, 'Rua da Assunção 53', 'Lisbon', None,
'Portugal', None, '+351 (213) 466-111', None, 'jfernandes@yahoo.pt', 4)
(35, 'Madalena', 'Sampaio', None, 'Rua dos Campeões Europeus de Viena, 4350',
'Porto', None, 'Portugal', None, '+351 (225) 022-448', None,
'masampaio@sapo.pt', 4)
(36, 'Hannah', 'Schneider', None, 'Tauentzienstraße 8', 'Berlin', None,
'Germany', '10789', '+49 030 26550280', None, 'hannah.schneider@yahoo.de', 5)
(37, 'Fynn', 'Zimmermann', None, 'Berger Straße 10', 'Frankfurt', None,
'Germany', '60316', '+49 069 40598889', None, 'fzimmermann@yahoo.de', 3)
(38, 'Niklas', 'Schröder', None, 'Barbarossastraße 19', 'Berlin', None,
'Germany', '10779', '+49 030 2141444', None, 'nschroder@surfeu.de', 3)
(39, 'Camille', 'Bernard', None, '4, Rue Milton', 'Paris', None, 'France',
'75009', '+33 01 49 70 65 65', None, 'camille.bernard@yahoo.fr', 4)
(40, 'Dominique', 'Lefebvre', None, '8, Rue Hanovre', 'Paris', None, 'France',
'75002', '+33 01 47 42 71 71', None, 'dominiquelefebvre@gmail.com', 4)
(41, 'Marc', 'Dubois', None, '11, Place Bellecour', 'Lyon', None, 'France',
'69002', '+33 04 78 30 30 30', None, 'marc.dubois@hotmail.com', 5)
(42, 'Wyatt', 'Girard', None, '9, Place Louis Barthou', 'Bordeaux', None,
'France', '33000', '+33 05 56 96 96 96', None, 'wyatt.girard@yahoo.fr', 3)
(43, 'Isabelle', 'Mercier', None, '68, Rue Jouvence', 'Dijon', None, 'France',
'21000', '+33 03 80 73 66 99', None, 'isabelle_mercier@apple.fr', 3)
(44, 'Terhi', 'Hämäläinen', None, 'Porthaninkatu 9', 'Helsinki', None,
'Finland', '00530', '+358 09 870 2000', None, 'terhi.hamalainen@apple.fi', 3)
(45, 'Ladislav', 'Kovács', None, 'Erzsébet krt. 58.', 'Budapest', None,
'Hungary', 'H-1073', None, None, 'ladislav_kovacs@apple.hu', 3)
(46, 'Hugh', "O'Reilly", None, '3 Chatham Street', 'Dublin', 'Dublin',
'Ireland', None, '+353 01 6792424', None, 'hughoreilly@apple.ie', 3)
(47, 'Lucas', 'Mancini', None, 'Via Degli Scipioni, 43', 'Rome', 'RM', 'Italy',
'00192', '+39 06 39733434', None, 'lucas.mancini@yahoo.it', 5)
(48, 'Johannes', 'Van der Berg', None, 'Lijnbaansgracht 120bg', 'Amsterdam',
'VV', 'Netherlands', '1016', '+31 020 6223130', None, 'johavanderberg@yahoo.nl',
5)
(49, 'Stanisaw', 'Wójcik', None, 'Ordynacka 10', 'Warsaw', None, 'Poland',
'00-358', '+48 22 828 37 39', None, 'stanisaw.wójcik@wp.pl', 4)
(50, 'Enrique', 'Muñoz', None, 'C/ San Bernardo 85', 'Madrid', None, 'Spain',
'28015', '+34 914 454 454', None, 'enrique_munoz@yahoo.es', 5)
(51, 'Joakim', 'Johansson', None, 'Celsiusg. 9', 'Stockholm', None, 'Sweden',
'11230', '+46 08-651 52 52', None, 'joakim.johansson@yahoo.se', 5)
(52, 'Emma', 'Jones', None, '202 Hoxton Street', 'London', None, 'United
Kingdom', 'N1 5LH', '+44 020 7707 0707', None, 'emma_jones@hotmail.com', 3)
(53, 'Phil', 'Hughes', None, '113 Lupus St', 'London', None, 'United Kingdom',
'SW1V 3EN', '+44 020 7976 5722', None, 'phil.hughes@gmail.com', 3)
(54, 'Steve', 'Murray', None, '110 Raeburn Pl', 'Edinburgh ', None, 'United
Kingdom', 'EH4 1HH', '+44 0131 315 3300', None, 'steve.murray@yahoo.uk', 5)
(55, 'Mark', 'Taylor', None, '421 Bourke Street', 'Sidney', 'NSW', 'Australia',
'2010', '+61 (02) 9332 3633', None, 'mark.taylor@yahoo.au', 4)
(56, 'Diego', 'Gutiérrez', None, '307 Macacha Güemes', 'Buenos Aires', None,
'Argentina', '1106', '+54 (0)11 4311 4333', None, 'diego.gutierrez@yahoo.ar', 4)
(57, 'Luis', 'Rojas', None, 'Calle Lira, 198', 'Santiago', None, 'Chile', None,
'+56 (0)2 635 4444', None, 'luisrojas@yahoo.cl', 5)
(58, 'Manoj', 'Pareek', None, '12,Community Centre', 'Delhi', None, 'India',
'110017', '+91 0124 39883988', None, 'manoj.pareek@rediff.com', 3)
(59, 'Puja', 'Srivastava', None, '3,Raj Bhavan Road', 'Bangalore', None,
'India', '560001', '+91 080 22289999', None, 'puja_srivastava@yahoo.in', 3)
```

As we can see, the output isn't that pretty. We can use the pandas library to create a dataframe from our query results! - pd.read_sql_query(query, conn): where query is the query you want to run and connection is the connection to the database you have established

In the cell below, create a dataframe that holds all of Chinook's Canadian customers, by completing the query that is started:

[15]:
```
canadians = pd.read_sql_query("SELECT * FROM customers WHERE Country=='Canada'",␣
↪conn)
canadians
```

[15]:
```
   CustomerId FirstName  LastName        Company                Address  \
0           3  François  Tremblay           None       1498 rue Bélanger
1          14      Mark   Philips          Telus          8210 111 ST NW
2          15  Jennifer  Peterson  Rogers Canada      700 W Pender Street
3          29    Robert     Brown           None   796 Dundas Street West
4          30    Edward   Francis           None        230 Elgin Street
5          31    Martha      Silk           None     194A Chain Lake Drive
6          32     Aaron  Mitchell           None       696 Osborne Street
7          33     Ellie  Sullivan           None          5112 48 Street

         City State Country PostalCode              Phone                Fax  \
0    Montréal    QC  Canada    H2G 1A7  +1 (514) 721-4711               None
1    Edmonton    AB  Canada    T6G 2C7  +1 (780) 434-4554  +1 (780) 434-5565
2   Vancouver    BC  Canada    V6C 1G8  +1 (604) 688-2255  +1 (604) 688-8756
3     Toronto    ON  Canada    M6J 1V1  +1 (416) 363-8888               None
4      Ottawa    ON  Canada    K2P 1L7  +1 (613) 234-3322               None
5     Halifax    NS  Canada    B3S 1C5  +1 (902) 450-0450               None
6    Winnipeg    MB  Canada    R3L 2B9  +1 (204) 452-6452               None
7  Yellowknife   NT  Canada    X1A 1N6  +1 (867) 920-2233               None

                    Email  SupportRepId
0     ftremblay@gmail.com             3
1      mphilips12@shaw.ca             5
2     jenniferp@rogers.ca             3
3        robbrown@shaw.ca             3
4     edfrancis@yachoo.ca             3
5    marthasilk@gmail.com             5
6  aaronmitchell@yahoo.ca             4
7  ellie.sullivan@shaw.ca             3
```

Create a dataframe that holds the number of invoices per country in descending order

[25]:
```
invoicePerCountry = pd.read_sql_query("SELECT COUNT(InvoiceId) as COUNT,␣
↪BillingCountry FROM invoices GROUP BY BillingCountry ORDER BY COUNT DESC",␣
↪conn)
invoicePerCountry
```

[25]:
```
    COUNT   BillingCountry
0      91              USA
1      56           Canada
2      35           Brazil
3      35           France
4      28          Germany
5      21   United Kingdom
6      14   Czech Republic
7      14         Portugal
8      13            India
9       7        Argentina
10      7        Australia
11      7          Austria
12      7          Belgium
13      7            Chile
14      7          Denmark
15      7          Finland
16      7          Hungary
17      7          Ireland
18      7            Italy
19      7      Netherlands
20      7           Norway
21      7           Poland
22      7            Spain
23      7           Sweden
```

For each record in the Album table, we want the Title along with the Name of the Artist. This will require an inner join!

[33]:
```
records = pd.read_sql_query("SELECT albums.Title, artists.Name FROM albums INNER␣
↪JOIN artists ON albums.ArtistId = artists.ArtistId", conn)
records
```

[33]:
```
                                                 Title  \
0                    For Those About To Rock We Salute You
1                                    Balls to the Wall
2                                    Restless and Wild
3                                    Let There Be Rock
4                                             Big Ones
5                                   Jagged Little Pill
6                                             Facelift
7                                      Warner 25 Anos
8                         Plays Metallica By Four Cellos
9                                           Audioslave
```

```
10                                        Out Of Exile
11                                  BackBeat Soundtrack
12                              The Best Of Billy Cobham
13               Alcohol Fueled Brewtality Live! [Disc 1]
14               Alcohol Fueled Brewtality Live! [Disc 2]
15                                       Black Sabbath
16                    Black Sabbath Vol. 4 (Remaster)
17                                          Body Count
18                                     Chemical Wedding
19    The Best Of Buddy Guy - The Millenium Collection
20                                        Prenda Minha
21                               Sozinho Remix Ao Vivo
22                                      Minha Historia
23                                      Afrociberdelia
24                                     Da Lama Ao Caos
25                                   Acústico MTV [Live]
26                                  Cidade Negra - Hits
27                                            Na Pista
28                                      Axé Bahia 2001
29                         BBC Sessions [Disc 1] [Live]
..                                                 ...
317                             SCRIABIN: Vers la flamme
318  Armada: Music from the Courts of England and S...
319                     Mozart: Symphonies Nos. 40 & 41
320                                       Back to Black
321                                               Frank
322               Carried to Dust (Bonus Track Version)
323          Beethoven: Symphony No. 6 'Pastoral' Etc.
324                      Bartok: Violin & Viola Concertos
325              Mendelssohn: A Midsummer Night's Dream
326                       Bach: Orchestral Suites Nos. 1 - 4
327        Charpentier: Divertissements, Airs & Concerts
328                               South American Getaway
329                               Górecki: Symphony No. 3
330                              Purcell: The Fairy Queen
331                        The Ultimate Relexation Album
332                     Purcell: Music for the Queen Mary
333                         Weill: The Seven Deadly Sins
334  J.S. Bach: Chaconne, Suite in E Minor, Partita...
335  Prokofiev: Symphony No.5 & Stravinksy: Le Sacr...
336                     Szymanowski: Piano Works, Vol. 1
337                          Nielsen: The Six Symphonies
338  Great Recordings of the Century: Paganini's 24...
339          Liszt - 12 Études D'Execution Transcendante
340  Great Recordings of the Century - Shubert: Sch...
341  Locatelli: Concertos for Violin, Strings and C...
342                             Respighi:Pines of Rome
```

```
343  Schubert: The Late String Quartets & String Qu...
344                            Monteverdi: L'Orfeo
345                           Mozart: Chamber Music
346  Koyaanisqatsi (Soundtrack from the Motion Pict...

                                                Name
0                                              AC/DC
1                                             Accept
2                                             Accept
3                                              AC/DC
4                                          Aerosmith
5                                  Alanis Morissette
6                                    Alice In Chains
7                               Antônio Carlos Jobim
8                                       Apocalyptica
9                                         Audioslave
10                                        Audioslave
11                                          BackBeat
12                                      Billy Cobham
13                                Black Label Society
14                                Black Label Society
15                                     Black Sabbath
16                                     Black Sabbath
17                                        Body Count
18                                    Bruce Dickinson
19                                         Buddy Guy
20                                    Caetano Veloso
21                                    Caetano Veloso
22                                      Chico Buarque
23                       Chico Science & Nação Zumbi
24                       Chico Science & Nação Zumbi
25                                      Cidade Negra
26                                      Cidade Negra
27                                      Cláudio Zoli
28                                   Various Artists
29                                      Led Zeppelin
..                                               ...
317                             Christopher O'Riley
318                                         Fretwork
319     Berliner Philharmoniker & Herbert Von Karajan
320                                    Amy Winehouse
321                                    Amy Winehouse
322                                         Calexico
323           Otto Klemperer & Philharmonia Orchestra
324                                   Yehudi Menuhin
325     Philharmonia Orchestra & Sir Neville Marriner
326  Academy of St. Martin in the Fields, Sir Nevil...
```

```
327           Les Arts Florissants & William Christie
328         The 12 Cellists of The Berlin Philharmonic
329                     Adrian Leaper & Doreen de Feis
330         Roger Norrington, London Classical Players
331  Charles Dutoit & L'Orchestre Symphonique de Mo...
332  Equale Brass Ensemble, John Eliot Gardiner & M...
333      Kent Nagano and Orchestre de l'Opéra de Lyon
334                                       Julian Bream
335      Berliner Philharmoniker & Herbert Von Karajan
336                                      Martin Roscoe
337                   Göteborgs Symfoniker & Neeme Järvi
338                                     Itzhak Perlman
339                                 Michele Campanella
340                                       Gerald Moore
341   Mela Tenenbaum, Pro Musica Prague & Richard Kapp
342                                     Eugene Ormandy
343                              Emerson String Quartet
344  C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345                                      Nash Ensemble
346                              Philip Glass Ensemble

[347 rows x 2 columns]
```

## 3.2 Create Our Own Table

Within the chinook database we want to create a table that holds each Employee ID and their total sales amount from the invoices they are associated with. - Step 1: Write a query to sum the total sales per employee ID - Step 2: Create a table called sales in the chinook database that holds two columns: the employee ID and the sale amount - Step 3: To see if your insert was successful, join the employee ID columns of the employee table and your table to see their names

Complete the queries below to run this exercise. In many cases, if a query is partially formed, you must complete it where you see the three dots (...)

[34]:
```
#Step 1 - run this, but make sure you understand how it works
query = "SELECT c.SupportRepID, sum(total) dollars_spent FROM invoices i INNER␣
 ↪JOIN Customers c ON c.customerID = i.CustomerID GROUP BY SupportRepID"
df = pd.read_sql_query(query, conn)
df
```

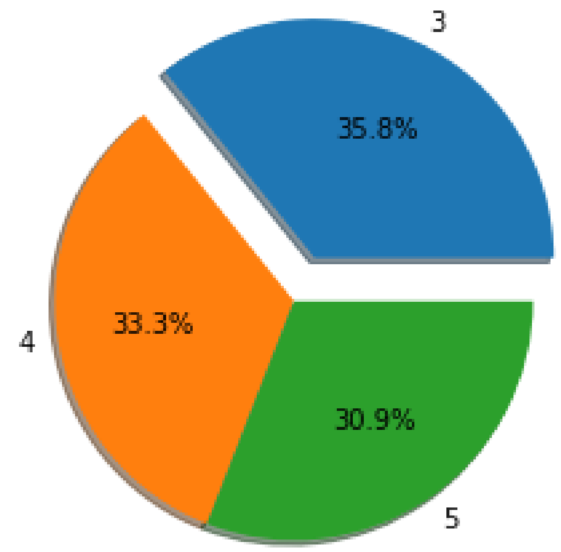
[34]:
```
   SupportRepId  dollars_spent
0             3         833.04
1             4         775.40
2             5         720.16
```

[36]:
```
#Quick Visualization -- Run Me!!
import matplotlib.pyplot as plt
import numpy as np
```

```
labels = df.SupportRepId.values
sizes = df.dollars_spent.values
explode = (.2, 0, 0)

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, explode = explode, autopct='%1.1f%%', shadow =␣
 ↪True)
ax1.axis('equal')

plt.show()
```



[38]:
```
#Step 2

#Create Table
# c.execute("CREATE TABLE sales ('employeeID' INT NOT NULL, 'total' DECIMAL NOT␣
 ↪NULL)")

#Insert into Table - from your df above choose several values
c.execute("INSERT INTO sales VALUES (3, 833.04)")
c.execute("INSERT INTO sales VALUES (4, 775.40)")
c.execute("INSERT INTO sales VALUES (5, 720.16)")
```

[38]:
```
<sqlite3.Cursor at 0x7f4dd3f5a570>
```

[50]:
```
#Step 3
for row in c.execute('SELECT sales.empID, employees.FirstName, employees.
 ↪LastName FROM sales INNER JOIN employees ON sales.empID = employees.
 ↪EmployeeId'):
    print(row)
```

```
(3, 'Jane', 'Peacock')
(4, 'Margaret', 'Park')
(5, 'Steve', 'Johnson')
```

After we are done we have to close the database to make sure it saves everything in our file:

[51]:
```
c.close()
```

*This assignment was completed independently by Jessica Strait.*

[ ]:
```

```