In [3]:

```
import sqlite3
conn = sqlite3.connect("Chinook_Sqlite.sqlite")
cursor = conn.cursor()
```

1 Basic SQL Queries

(a)

In [24]:

```
invoices = cursor.execute('SELECT InvoiceID, CustomerID FROM Invoice ORDER BY Total DESC LIMIT 1
5')
for row in invoices:
    print(row)
```

```
(404, 6)
(299, 26)
(96, 45)
(194, 46)
(89, 7)
(201, 25)
(88, 57)
(306, 5)
(313, 43)
(103, 24)
(208, 4)
(193, 37)
(5, 23)
(12, 2)
(19, 40)
```

(b)

In [23]:

```
first_names = cursor.execute('SELECT Employee.FirstName,Employee.LastName FROM Employee,Customer
WHERE Employee.FirstName == Customer.FirstName')
for row in first_names:
    print(row)
```

```
('Steve', 'Johnson')
('Robert', 'King')
```

(c)

In [25]:

```
last_names = cursor.execute('SELECT Employee.FirstName,Employee.LastName FROM Employee,Customer
 WHERE Employee.LastName == Customer.LastName')
for row in last_names:
    print(row)
```

('Michael', 'Mitchell')

(d)

In [23]:

```
cursor.execute('SELECT g.Name,COUNT(*) FROM Track t, Genre g ON g.GenreID == t.GenreID GROUP BY
 g.GenreID ORDER BY COUNT(*) DESC')
output = cursor.fetchall();
print(output[0][0])
# for row in output:
#     print(row)
```

Rock

(e)

In [29]:

```
cursor.execute('SELECT FirstName, LastName,COUNT(TrackId)\
               FROM \
               (SELECT c.FirstName,c.LastName,c.CustomerId, t.TrackId FROM Customer c\
               INNER JOIN Invoice i ON c.CustomerId == i.InvoiceId\
               INNER JOIN InvoiceLine il ON i.InvoiceId == il.InvoiceId\
               INNER JOIN Track t ON il.TrackId == t.TrackId)\
               GROUP BY CustomerId\
               ORDER BY COUNT(TrackId) DESC')
output = cursor.fetchall()
num = output[0][2]
for row in output:
    if(row[2] == num):
        print(row[:2])
```

('František', 'Wichterlová')
('Roberto', 'Almeida')
('Tim', 'Goyer')
('Richard', 'Cunningham')
('Ellie', 'Sullivan')
('Dominique', 'Lefebvre')
('Lucas', 'Mancini')
('Steve', 'Murray')

(f)

In [46]:

```
cursor.execute("""
                SELECT Name, Cnt
                FROM
                    (
                    SELECT Name,COUNT(AlbumId) AS Cnt
                        FROM
                        (
                        SELECT Artist.Name, Album.AlbumId FROM Artist
                        INNER JOIN Album USING (ArtistId)
                        )
                    GROUP BY Name
                    )
               WHERE Cnt > 5
               """)
Qf = cursor.fetchall()
for row in Qf:
    print(row)
```

```
('Deep Purple', 11)
('Iron Maiden', 21)
('Led Zeppelin', 14)
('Metallica', 10)
('Ozzy Osbourne', 6)
('U2', 10)
```

(g)

In [52]:

```
cursor.execute("""
                SELECT DISTINCT Artist, Genre
                FROM
                    (
                    SELECT ar.Name AS Artist, g.Name AS Genre
                    FROM Artist ar
                    INNER JOIN Album al ON ar.ArtistId == al.ArtistId
                    INNER JOIN Track t ON al.AlbumId == t.AlbumId
                    INNER JOIN Genre g ON t.GenreId == g.GenreId
                    )
                WHERE Artist == 'Iron Maiden'
               """)
Qg = cursor.fetchall()
for row in Qg:
    print(row)
```

```
('Iron Maiden', 'Rock')
('Iron Maiden', 'Metal')
('Iron Maiden', 'Heavy Metal')
('Iron Maiden', 'Blues')
```

(h)

In [48]:

```
cursor.execute("""
                SELECT Title,Cnt
                FROM
                (
                    SELECT Title,COUNT(DISTINCT PlaylistId) AS Cnt
                    FROM
                       (
                       SELECT a.Title, p.PlaylistId
                       FROM Album a
                       INNER JOIN Track t ON a.AlbumId == t.AlbumId
                       INNER JOIN PlaylistTrack pl ON t.TrackId == pl.TrackId
                       INNER JOIN Playlist p ON p.PlaylistId == pl.PlaylistId
                       )
                    GROUP BY Title
                )
                WHERE Cnt > 3
                """)
Qh = cursor.fetchall()
for row in Qh:
    print(row)
```

```
('A Copland Celebration, Vol. I', 4)
('A Soprano Inspired', 5)
('A-Sides', 4)
('Adams, John: The Chairman Dances', 4)
('Adorate Deum: Gregorian Chant from the Proper of the Mass', 5)
('Allegri: Miserere', 5)
('Armada: Music from the Courts of England and Spain', 4)
('Bach: Goldberg Variations', 5)
('Bach: Orchestral Suites Nos. 1 - 4', 5)
('Bach: The Brandenburg Concertos', 4)
('Bach: The Cello Suites', 5)
('Bach: Toccata & Fugue in D Minor', 5)
('Bach: Violin Concertos', 4)
('Bartok: Violin & Viola Concertos', 4)
('Beethoven Piano Sonatas: Moonlight & Pastorale', 5)
('Beethoven: Symhonies Nos. 5 & 6', 5)
("Beethoven: Symphony No. 6 'Pastoral' Etc.", 4)
('Berlioz: Symphonie Fantastique', 5)
('Bizet: Carmen Highlights', 5)
('Black Album', 4)
('Carmina Burana', 5)
('Charpentier: Divertissements, Airs & Concerts', 4)
('Chopin: Piano Concertos Nos. 1 & 2', 5)
('Core', 4)
('Djavan Ao Vivo - Vol. 02', 4)
('Djavan Ao Vivo - Vol. 1', 4)
('Elgar: Cello Concerto & Vaughan Williams: Fantasias', 4)
('English Renaissance', 6)
('Facelift', 4)
('Fauré: Requiem, Ravel: Pavane & Others', 4)
('Great Opera Choruses', 5)
("Great Performances - Barber's Adagio and Other Romantic Favorites for Strings",
4)
('Great Recordings of the Century - Mahler: Das Lied von der Erde', 5)
('Great Recordings of the Century - Shubert: Schwanengesang, 4 Lieder', 4)
("Great Recordings of the Century: Paganini's 24 Caprices", 4)
('Grieg: Peer Gynt Suites & Sibelius: Pelléas et Mélisande', 4)
('Górecki: Symphony No. 3', 5)
('Handel: Music for the Royal Fireworks (Original Version 1749)', 5)
('Handel: The Messiah (Highlights)', 5)
('Haydn: Symphonies 99 - 104', 4)
('Holst: The Planets, Op. 32 & Vaughan Williams: Fantasies', 5)
('J.S. Bach: Chaconne, Suite in E Minor, Partita in E Major & Prelude, Fugue and A
llegro', 5)
('Koyaanisqatsi (Soundtrack from the Motion Picture)', 5)
("Liszt - 12 Études D'Execution Transcendante", 4)
('Locatelli: Concertos for Violin, Strings and Continuo, Vol. 3', 5)
('Mascagni: Cavalleria Rusticana', 5)
("Mendelssohn: A Midsummer Night's Dream", 5)
('Meus Momentos', 4)
('Minha Historia', 4)
("Monteverdi: L'Orfeo", 4)
('Motley Crue Greatest Hits', 4)
('Mozart Gala: Famous Arias', 5)
('Mozart: Chamber Music', 4)
('Mozart: Symphonies Nos. 40 & 41', 5)
('Mozart: Wind Concertos', 4)
('Nevermind', 4)
('Nielsen: The Six Symphonies', 4)
('Pachelbel: Canon & Gigue', 4)
('Palestrina: Missa Papae Marcelli & Allegri: Miserere', 4)
```

```
("Pavarotti's Opera Made Easy", 5)
('Prenda Minha', 4)
('Prokofiev: Romeo & Juliet', 5)
('Prokofiev: Symphony No.1', 4)
('Prokofiev: Symphony No.5 & Stravinksy: Le Sacre Du Printemps', 4)
('Puccini: Madama Butterfly - Highlights', 4)
('Purcell: Music for the Queen Mary', 4)
('Purcell: The Fairy Queen', 4)
('Quanta Gente Veio Ver (Live)', 4)
('Respighi:Pines of Rome', 5)
('Restless and Wild', 4)
('SCRIABIN: Vers la flamme', 4)
('Scheherazade', 5)
("Schubert: The Late String Quartets & String Quintet (3 CD's)", 4)
('Sibelius: Finlandia', 5)
('Sir Neville Marriner: A Celebration', 4)
('South American Getaway', 4)
('Strauss: Waltzes', 5)
('Szymanowski: Piano Works, Vol. 1', 5)
("Tchaikovsky: 1812 Festival Overture, Op.49, Capriccio Italien & Beethoven: Welli
ngton's Victory", 5)
('Tchaikovsky: The Nutcracker', 4)
('Temple of the Dog', 4)
('Ten', 4)
('The Last Night of the Proms', 5)
('The Ultimate Relexation Album', 4)
('The World of Classical Favourites', 6)
('Unplugged', 4)
('Vivaldi: The Four Seasons', 4)
('Vs.', 4)
('Wagner: Favourite Overtures', 5)
('Weill: The Seven Deadly Sins', 5)
```

2 Clustering Artists

(a)

In [2]:

```
import pandas as pd
```

In [31]:

```
df = pd.read_sql_query("""
                SELECT ar.Name AS Artist, al.Title AS Album, g.Name AS Genre, p.Name AS Playlis
t, t.Name AS Track
                FROM Artist ar
                INNER JOIN Album al ON ar.ArtistId == al.ArtistId
                INNER JOIN Track t ON t.AlbumId == al.AlbumId
                INNER JOIN Genre g ON g.GenreId == t.GenreId
                INNER JOIN PlaylistTrack pt ON t.TrackId == pt.TrackId
                INNER JOIN Playlist p ON p.PlaylistId == PT.PlaylistId
               """,conn)
df
```

Out[31]:

	Artist	Album	Genre	Playlist	Track
0	Audioslave	Revelations	Alternative	Music	Band Members Discuss Tracks from "Revelations"
1	Audioslave	Revelations	Alternative	Music	Revelations
2	Audioslave	Revelations	Alternative	Music	One and the Same
3	Audioslave	Revelations	Alternative	Music	Sound of a Gun
4	Audioslave	Revelations	Alternative	Music	Until We Fall
5	Audioslave	Revelations	Alternative	Music	Original Fire
6	Audioslave	Revelations	Alternative	Music	Broken City
7	Audioslave	Revelations	Alternative	Music	Somedays
8	Audioslave	Revelations	Alternative	Music	Shape of Things to Come
9	Audioslave	Revelations	Alternative	Music	Jewel of the Summertime
10	Audioslave	Revelations	Alternative	Music	Wide Awake
11	Audioslave	Revelations	Alternative	Music	Nothing Left to Say But Goodbye
12	Audioslave	Revelations	Alternative	Music	Moth
13	Audioslave	Revelations	Alternative	Music	Show Me How to Live (Live at the Quart Festival)
14	Cake	Cake: B-Sides and Rarities	Alternative	Music	War Pigs
15	Calexico	Carried to Dust (Bonus Track Version)	Alternative	Music	Slowness
16	Chris Cornell	Carry On	Alternative	Music	No Such Thing
17	Chris Cornell	Carry On	Alternative	Music	Poison Eye
18	Chris Cornell	Carry On	Alternative	Music	Arms Around Your Love
19	Chris Cornell	Carry On	Alternative	Music	Safe and Sound
20	Chris Cornell	Carry On	Alternative	Music	She'll Never Be Your Man
21	Chris Cornell	Carry On	Alternative	Music	Ghosts
22	Chris Cornell	Carry On	Alternative	Music	Killing Birds
23	Chris Cornell	Carry On	Alternative	Music	Billie Jean
24	Chris Cornell	Carry On	Alternative	Music	Scar On the Sky
25	Chris Cornell	Carry On	Alternative	Music	Your Soul Today
26	Chris Cornell	Carry On	Alternative	Music	Finally Forever
27	Chris Cornell	Carry On	Alternative	Music	Silence the Voices

	Artist	Album	Genre	Playlist	Track
28	Chris Cornell	Carry On	Alternative	Music	Disappearing Act
29	Chris Cornell	Carry On	Alternative	Music	You Know My Name
...	...	...	...	...	...
8685	Nirvana	Nevermind	Rock	Grunge	Lithium
8686	Nirvana	Nevermind	Rock	Grunge	Drain You
8687	Nirvana	Nevermind	Rock	Grunge	On A Plain
8688	AC/DC	For Those About To Rock We Salute You	Rock	Heavy Metal Classic	For Those About To Rock (We Salute You)
8689	Accept	Balls to the Wall	Rock	Heavy Metal Classic	Balls to the Wall
8690	Accept	Restless and Wild	Rock	Heavy Metal Classic	Fast As a Shark
8691	Accept	Restless and Wild	Rock	Heavy Metal Classic	Restless and Wild
8692	Accept	Restless and Wild	Rock	Heavy Metal Classic	Princess of the Dawn
8693	Black Sabbath	Black Sabbath	Metal	Heavy Metal Classic	N.I.B.
8694	Black Sabbath	Black Sabbath Vol. 4 (Remaster)	Metal	Heavy Metal Classic	Supernaut
8695	Iron Maiden	Killers	Heavy Metal	Heavy Metal Classic	Wrathchild
8696	Iron Maiden	Killers	Heavy Metal	Heavy Metal Classic	Killers
8697	Iron Maiden	The Number of The Beast	Metal	Heavy Metal Classic	Run to the Hills
8698	Iron Maiden	Piece Of Mind	Metal	Heavy Metal Classic	Where Eagles Dare
8699	Iron Maiden	Powerslave	Metal	Heavy Metal Classic	2 Minutes To Midnight
8700	Iron Maiden	Somewhere in Time	Metal	Heavy Metal Classic	Wasted Years
8701	Metallica	Black Album	Metal	Heavy Metal Classic	Enter Sandman
8702	Metallica	Kill 'Em All	Metal	Heavy Metal Classic	The Four Horsemen

	Artist	Album	Genre	Playlist	Track
8703	Metallica	Kill 'Em All	Metal	Heavy Metal Classic	Seek & Destroy
8704	Metallica	Master Of Puppets	Metal	Heavy Metal Classic	Master Of Puppets
8705	Metallica	Ride The Lightning	Metal	Heavy Metal Classic	For Whom The Bell Tolls
8706	Metallica	Ride The Lightning	Metal	Heavy Metal Classic	Creeping Death
8707	Mötley Crüe	Motley Crue Greatest Hits	Metal	Heavy Metal Classic	Looks That Kill
8708	Motörhead	Ace Of Spades	Metal	Heavy Metal Classic	Ace Of Spades
8709	Motörhead	Ace Of Spades	Metal	Heavy Metal Classic	Live To Win
8710	Ozzy Osbourne	Blizzard of Ozz	Rock	Heavy Metal Classic	I Don't Know
8711	Ozzy Osbourne	Blizzard of Ozz	Rock	Heavy Metal Classic	Crazy Train
8712	Ozzy Osbourne	Diary of a Madman (Remastered)	Rock	Heavy Metal Classic	Flying High Again
8713	Scorpions	20th Century Masters - The Millennium Collecti...	Rock	Heavy Metal Classic	The Zoo
8714	Miles Davis	The Essential Miles Davis [Disc 1]	Jazz	On-The-Go 1	Now's The Time

8715 rows × 5 columns

(b)

In [47]:

```
# Use sql query to find artists who have more than one album
artist = pd.read_sql_query("""
                SELECT Name, Cnt
                FROM
                    (
                    SELECT Name,COUNT(AlbumId) AS Cnt
                        FROM
                        (
                        SELECT Artist.Name, Artist.ArtistId,Album.AlbumId FROM Artist
                        INNER JOIN Album USING (ArtistId)
                        )
                    GROUP BY Name
                    )
                WHERE Cnt > 1
                """,conn)
artist = list(artist['Name'])
print("Artists who have more than one ablum:")
print(artist)
df = df[df['Artist'].isin(artist)]
df
```

Artists who have more than one ablum:
['AC/DC', 'Accept', 'Amy Winehouse', 'Antônio Carlos Jobim', 'Audioslave', 'Battle star Galactica', 'Berliner Philharmoniker & Herbert Von Karajan', 'Black Label Society', 'Black Sabbath', 'Caetano Veloso', 'Chico Science & Nação Zumbi', 'Cidade Negra', 'Creedence Clearwater Revival', 'Cássia Eller', 'Deep Purple', 'Djavan', 'English Concert & Trevor Pinnock', 'Eric Clapton', 'Eugene Ormandy', 'Faith No More', 'Foo Fighters', 'Gilberto Gil', 'Green Day', "Guns N' Roses", 'Iron Maiden', 'Jamiroquai', 'Kiss', 'Led Zeppelin', 'Legião Urbana', 'Lost', 'Lulu Santos', 'Metallica', 'Michael Tilson Thomas & San Francisco Symphony', 'Miles Davis', 'Milton Nascimento', 'Nirvana', 'Os Paralamas Do Sucesso', 'Ozzy Osbourne', 'Pearl Jam', 'Queen', 'R.E.M.', 'Red Hot Chili Peppers', 'Santana', 'Skank', 'Smashing Pumpkins', 'Spyro Gyra', 'The Black Crowes', 'The Cult', 'The Office', 'The Rolling Stones', 'The Tea Party', 'Tim Maia', 'Titãs', 'U2', 'Van Halen', 'Various Artists']

Out[47]:

	Artist	Album	Genre	Playlist	Track
0	Audioslave	Revelations	Alternative	Music	Band Members Discuss Tracks from "Revelations"
1	Audioslave	Revelations	Alternative	Music	Revelations
2	Audioslave	Revelations	Alternative	Music	One and the Same
3	Audioslave	Revelations	Alternative	Music	Sound of a Gun
4	Audioslave	Revelations	Alternative	Music	Until We Fall
5	Audioslave	Revelations	Alternative	Music	Original Fire
6	Audioslave	Revelations	Alternative	Music	Broken City
7	Audioslave	Revelations	Alternative	Music	Somedays
8	Audioslave	Revelations	Alternative	Music	Shape of Things to Come
9	Audioslave	Revelations	Alternative	Music	Jewel of the Summertime
10	Audioslave	Revelations	Alternative	Music	Wide Awake
11	Audioslave	Revelations	Alternative	Music	Nothing Left to Say But Goodbye
12	Audioslave	Revelations	Alternative	Music	Moth
13	Audioslave	Revelations	Alternative	Music	Show Me How to Live (Live at the Quart Festival)
40	Audioslave	Out Of Exile	Alternative & Punk	Music	Your Time Has Come
41	Audioslave	Out Of Exile	Alternative & Punk	Music	Out Of Exile
42	Audioslave	Out Of Exile	Alternative & Punk	Music	Be Yourself
43	Audioslave	Out Of Exile	Alternative & Punk	Music	Doesn't Remind Me
44	Audioslave	Out Of Exile	Alternative & Punk	Music	Drown Me Slowly
45	Audioslave	Out Of Exile	Alternative & Punk	Music	Heaven's Dead
46	Audioslave	Out Of Exile	Alternative & Punk	Music	The Worm
47	Audioslave	Out Of Exile	Alternative & Punk	Music	Man Or Animal
48	Audioslave	Out Of Exile	Alternative & Punk	Music	Yesterday To Tomorrow
49	Audioslave	Out Of Exile	Alternative & Punk	Music	Dandelion
50	Audioslave	Out Of Exile	Alternative & Punk	Music	#1 Zero
51	Audioslave	Out Of Exile	Alternative & Punk	Music	The Curse
87	Faith No More	Album Of The Year	Alternative & Punk	Music	Collision
88	Faith No More	Album Of The Year	Alternative & Punk	Music	Stripsearch

	Artist	Album	Genre	Playlist	Track
89	Faith No More	Album Of The Year	Alternative & Punk	Music	Last Cup Of Sorrow
90	Faith No More	Album Of The Year	Alternative & Punk	Music	Naked In Front Of The Computer
...	...	...	...	...	...
8678	Pearl Jam	Vs.	Rock	Grunge	Daughter
8682	Nirvana	Nevermind	Rock	Grunge	Smells Like Teen Spirit
8683	Nirvana	Nevermind	Rock	Grunge	In Bloom
8684	Nirvana	Nevermind	Rock	Grunge	Come As You Are
8685	Nirvana	Nevermind	Rock	Grunge	Lithium
8686	Nirvana	Nevermind	Rock	Grunge	Drain You
8687	Nirvana	Nevermind	Rock	Grunge	On A Plain
8688	AC/DC	For Those About To Rock We Salute You	Rock	Heavy Metal Classic	For Those About To Rock (We Salute You)
8689	Accept	Balls to the Wall	Rock	Heavy Metal Classic	Balls to the Wall
8690	Accept	Restless and Wild	Rock	Heavy Metal Classic	Fast As a Shark
8691	Accept	Restless and Wild	Rock	Heavy Metal Classic	Restless and Wild
8692	Accept	Restless and Wild	Rock	Heavy Metal Classic	Princess of the Dawn
8693	Black Sabbath	Black Sabbath	Metal	Heavy Metal Classic	N.I.B.
8694	Black Sabbath	Black Sabbath Vol. 4 (Remaster)	Metal	Heavy Metal Classic	Supernaut
8695	Iron Maiden	Killers	Heavy Metal	Heavy Metal Classic	Wrathchild
8696	Iron Maiden	Killers	Heavy Metal	Heavy Metal Classic	Killers
8697	Iron Maiden	The Number of The Beast	Metal	Heavy Metal Classic	Run to the Hills
8698	Iron Maiden	Piece Of Mind	Metal	Heavy Metal Classic	Where Eagles Dare
8699	Iron Maiden	Powerslave	Metal	Heavy Metal Classic	2 Minutes To Midnight
8700	Iron Maiden	Somewhere in Time	Metal	Heavy Metal Classic	Wasted Years

	Artist	Album	Genre	Playlist	Track
8701	Metallica	Black Album	Metal	Heavy Metal Classic	Enter Sandman
8702	Metallica	Kill 'Em All	Metal	Heavy Metal Classic	The Four Horsemen
8703	Metallica	Kill 'Em All	Metal	Heavy Metal Classic	Seek & Destroy
8704	Metallica	Master Of Puppets	Metal	Heavy Metal Classic	Master Of Puppets
8705	Metallica	Ride The Lightning	Metal	Heavy Metal Classic	For Whom The Bell Tolls
8706	Metallica	Ride The Lightning	Metal	Heavy Metal Classic	Creeping Death
8710	Ozzy Osbourne	Blizzard of Ozz	Rock	Heavy Metal Classic	I Don't Know
8711	Ozzy Osbourne	Blizzard of Ozz	Rock	Heavy Metal Classic	Crazy Train
8712	Ozzy Osbourne	Diary of a Madman (Remastered)	Rock	Heavy Metal Classic	Flying High Again
8714	Miles Davis	The Essential Miles Davis [Disc 1]	Jazz	On-The-Go 1	Now's The Time

5697 rows × 5 columns

(c)

In [33]:

```
genre = pd.read_sql_query("""SELECT g.Name,COUNT(*)
                             FROM Track t, Genre g ON g.GenreID == t.GenreID
                             GROUP BY g.GenreID ORDER BY COUNT(*) DESC""",conn)
genre = list(genre['Name'])[:7]
genre
```

Out[33]:

```
['Rock', 'Latin', 'Metal', 'Alternative & Punk', 'Jazz', 'TV Shows', 'Blues']
```

In [34]:

```
artists = pd.unique(df['Artist'])
genre = pd.read_sql_query("""SELECT g.Name,COUNT(*)
                             FROM Track t, Genre g
                             ON g.GenreID == t.GenreID
                             GROUP BY g.GenreID
                             ORDER BY COUNT(*) DESC
                             """,conn)
genre = list(genre['Name'])[:7]
col = genre
col.insert(0,'Artist')
col.extend(['AlbumsNumber','TracksNumber','PlaylistsNumber'])
print(col)
matrix = pd.DataFrame(columns = col)
for artist in artists:
    row = []
    row.append(artist)
    temp = df[df['Artist'] == artist]
    albumNum = temp['Album'].nunique()
    trackNum = temp['Track'].nunique()
    playlistNum = temp['Playlist'].nunique()
    temp = temp.drop_duplicates(subset = ['Track'])
    for g in genre[1:8]:
        if g not in list(temp['Genre']):
            row.append(0)
        else:
            cnt = temp[temp['Genre'] == g].shape[0]
            row.append(cnt)
    row.append(albumNum)
    row.append(trackNum)
    row.append(playlistNum)
    matrix = matrix.append(pd.Series(row, index = col), ignore_index=True)

matrix
```

```
['Artist', 'Rock', 'Latin', 'Metal', 'Alternative & Punk', 'Jazz', 'TV Shows', 'Bl
ues', 'AlbumsNumber', 'TracksNumber', 'PlaylistsNumber']
```

Out[34]:

	Artist	Rock	Latin	Metal	Alternative & Punk	Jazz	TV Shows	Blues	AlbumsNumber	Tra
0	Audioslave	14	0	0	12	0	0	0	3	
1	Faith No More	15	0	0	37	0	0	0	4	
2	Foo Fighters	33	0	0	11	0	0	0	4	
3	Green Day	0	0	0	34	0	0	0	2	
4	Pearl Jam	50	0	0	13	0	0	0	5	
5	R.E.M.	14	0	0	27	0	0	0	3	
6	Red Hot Chili Peppers	31	0	0	17	0	0	0	3	
7	Smashing Pumpkins	0	0	0	34	0	0	0	2	
8	The Tea Party	0	0	0	22	0	0	0	2	
9	Titãs	0	0	0	38	0	0	0	2	
10	The Black Crowes	0	0	0	0	0	0	19	2	
11	Eric Clapton	0	16	0	0	0	0	31	2	
12	Iron Maiden	42	0	72	0	0	0	9	21	
13	Berliner Philharmoniker & Herbert Von Karajan	0	0	0	0	0	0	0	3	
14	English Concert & Trevor Pinnock	0	0	0	0	0	0	0	2	
15	Eugene Ormandy	0	0	0	0	0	0	0	3	
16	Michael Tilson Thomas & San Francisco Symphony	0	0	0	0	0	0	0	2	
17	Jamiroquai	10	0	0	0	0	0	0	3	
18	Antônio Carlos Jobim	0	16	0	0	14	0	0	2	
19	Gilberto Gil	0	15	0	0	3	0	0	3	
20	Miles Davis	0	0	0	0	36	0	0	3	
21	Spyro Gyra	0	0	0	0	21	0	0	2	
22	Caetano Veloso	0	21	0	0	0	0	0	2	
23	Cássia Eller	0	30	0	0	0	0	0	2	
24	Chico Science & Nação Zumbi	0	35	0	0	0	0	0	2	
25	Djavan	0	24	0	0	0	0	0	2	
26	Legião Urbana	0	30	0	0	0	0	0	2	
27	Lulu Santos	0	28	0	0	0	0	0	2	

	Artist	Rock	Latin	Metal	Alternative & Punk	Jazz	TV Shows	Blues	AlbumsNumber	Tra
28	Milton Nascimento	0	26	0	0	0	0	0	2	
29	Os Paralamas Do Sucesso	0	45	0	0	0	0	0	3	
30	Tim Maia	0	30	0	0	0	0	0	2	
31	Various Artists	0	28	0	0	0	0	0	4	
32	Black Label Society	0	0	18	0	0	0	0	2	
33	Black Sabbath	0	0	17	0	0	0	0	2	
34	Guns N' Roses	28	0	14	0	0	0	0	3	
35	Metallica	0	0	112	0	0	0	0	10	
36	Ozzy Osbourne	14	0	14	0	0	0	0	6	
37	U2	103	0	0	0	0	0	0	10	
38	Amy Winehouse	0	0	0	0	0	0	0	2	
39	Cidade Negra	0	0	0	0	0	0	0	2	
40	AC/DC	18	0	0	0	0	0	0	2	
41	Accept	4	0	0	0	0	0	0	2	
42	Creedence Clearwater Revival	40	0	0	0	0	0	0	2	
43	The Cult	30	0	0	0	0	0	0	2	
44	Deep Purple	87	0	0	0	0	0	0	11	
45	Kiss	29	0	0	0	0	0	0	2	
46	Led Zeppelin	91	0	0	0	0	0	0	14	
47	Nirvana	24	0	0	0	0	0	0	2	
48	Queen	43	0	0	0	0	0	0	3	
49	The Rolling Stones	40	0	0	0	0	0	0	3	
50	Santana	27	0	0	0	0	0	0	3	
51	Skank	23	0	0	0	0	0	0	2	
52	Van Halen	50	0	0	0	0	0	0	4	
53	Battlestar Galactica	0	0	0	0	0	5	0	2	
54	Lost	0	0	0	0	0	48	0	4	
55	The Office	0	0	0	0	0	36	0	3	

(d)

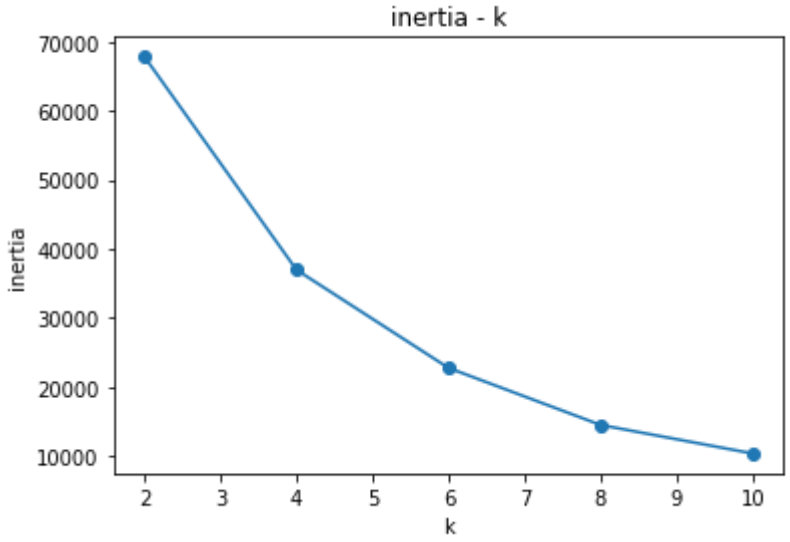
In [38]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
import matplotlib

ks=[2,4,6,8,10]
inerias = []
matrix_n = matrix.drop(['Artist'], axis = 1)
for k in ks:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(matrix_n)
    y_kmeans = kmeans.predict(matrix_n)
    ineria = kmeans.inertia_
    inerias.append([k, ineria])
    print(ineria)

inerias = np.array(inerias)
plt.scatter(inerias[:, 0], inerias[:, 1])
plt.plot(inerias[:, 0], inerias[:, 1])
plt.xlabel("k")
plt.ylabel("inertia")
plt.title("inertia - k")
```

```
67803.79607843136
36988.67647058823
22804.80586080586
14555.621565934067
10429.565476190477
```

Out[38]:

```
Text(0.5, 1.0, 'inertia - k')
```



Choose k = 4.
From the plot, we can see that k = 4 is at the “elbow” of the plot. Therefore, it provides a good tradeoff between accuracy (low value of inertia) and complexity (small value of k), which makes it the ideal number of k.

(e)

In [45]:

```
from sklearn.metrics import pairwise_distances_argmin_min

kmeans = KMeans(n_clusters=4)
km = kmeans.fit(matrix_n)
closest, _ = pairwise_distances_argmin_min(km.cluster_centers_, matrix_n)
print("ID:" + str(closest))
print("Artist:\n")
ind = 1
for i in closest:
    print("Centroid %d\n"%ind)
    print(matrix['Artist'][i])
    print(matrix.iloc[i,:])
    print("\n")
    ind += 1
```

```
ID:[ 0 19 46 12]
Artist:

Centroid 1

Audioslave
Artist                Audioslave
Rock                          14
Latin                          0
Metal                          0
Alternative & Punk            12
Jazz                           0
TV Shows                       0
Blues                          0
AlbumsNumber                   3
TracksNumber                  40
PlaylistsNumber                2
Name: 0, dtype: object


Centroid 2

Gilberto Gil
Artist                Gilberto Gil
Rock                             0
Latin                           15
Metal                            0
Alternative & Punk               0
Jazz                             3
TV Shows                         0
Blues                            0
AlbumsNumber                     3
TracksNumber                    32
PlaylistsNumber                  3
Name: 19, dtype: object


Centroid 3

Led Zeppelin
Artist                Led Zeppelin
Rock                            91
Latin                            0
Metal                            0
Alternative & Punk               0
Jazz                             0
TV Shows                         0
Blues                            0
AlbumsNumber                    14
TracksNumber                    91
PlaylistsNumber                  2
Name: 46, dtype: object


Centroid 4

Iron Maiden
Artist                Iron Maiden
Rock                           42
Latin                           0
Metal                          72
```

```
Alternative & Punk            0
Jazz                          0
TV Shows                      0
Blues                         9
AlbumsNumber                 21
TracksNumber                150
PlaylistsNumber               3
Name: 12, dtype: object
```

I think the data is clustered highly because of genre. From above we can see that each cluster centroid has must of their songs in different genre. For example, Led Zeppelin's songs are all rock songs, while Gilberto Gil's are about Latin. Iron Maiden and Foo Fighters's songs are more comprehensive. But we can still clear see that their focusing genre are quite different. Thus, I think they are mainly clstered by genre.