


SQL Workshop

☰ Tags	Foundation SQL
↗ Class	
☑ Finished Yet?	☑
↗ Knowledge	 <u>The Second Sprint: SQL</u>

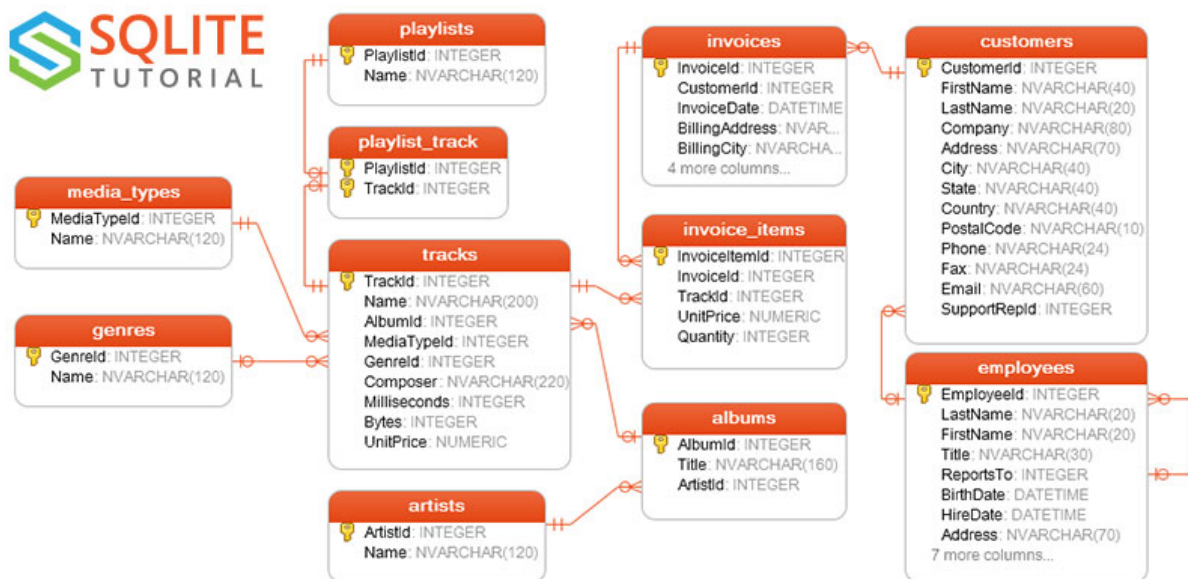
Part 1

- SQLite Online IDE: <https://sqliteonline.com/>
- SQLite Tutorial: <https://www.sqlitetutorial.net/>
- SQLite Sample Database: <https://www.sqlitetutorial.net/sqlite-sample-database/>
- Chinook Database:

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/84251582-f5db-4d7a-a27c-bf2e66685e1f/chinook.db>

- SQL มีหลาย Distribution เช่น MariaDB, MySQL, PostgreSQL เป็นต้น
- เราสามารถ Import File ที่เป็น .db (Database) เข้าไปใน SQLite ได้

[Entity-Relationship Diagram ของ chinook.db]



-Primary Key: ค่าที่ไม่ซ้ำกันในแต่ละ row ของ Table นั้น ๆ

-แต่ละ Column ใน SQL Database จะเป็นได้แค่ประเภทเดียวนั่น ข้อมูลสี่เทาใน ER Diagram คือการระบุประเภทของข้อมูลในแต่ละ Column

-Relationship:

1. One-to-One (ประเทศ 1 ประเทศมีเมืองหลวงได้แค่ 1 เมือง)
2. One-to-Many (ครู 1 คนมีนักเรียนได้หลายคน) *ใช้เยอะสุด
3. Many-to-Many (1 Playlist มีได้หลายเพลง และ 1 เพลงอยู่ได้ในหลาย Playlist) *ควรหลีกเลี่ยง

-SELECT ใช้ในการคำนวณได้ด้วย เช่น:

chinook.db

```

1 SELECT
2 100 * 5 columnA ,
3 5 / 6.0 columnB,
4 6 + 809 columnC,
5 900-167 columnD;

```

columnA	columnB	columnC	columnD
500	0.8333333333333334	815	733

-Single line comment ใช้ -- (double dash) เช่น --Calc

-SELECT * FROM customers: เลือกทุก column จาก table ชื่อ customers (ถ้าต้องการดึง column เฉพาะบางอัน ให้พิมพ์ชื่อ column ที่ต้องการดึงแทน *)

```

1 --Select all columns from customers table
2 SELECT * FROM customers

```

-ดึง firstname, lastname, company, country จาก table ชื่อ customers ดึงเฉพาะ 5 row แรก:

chinook.db

```
1 SELECT firstname, lastname, company, country FROM customers LIMIT 5;
```

FirstName	LastName	Company	Country
Luís	Gonçalves	Embraer - Empresa Brasileira d...	Brazil
Leonie	Köhler	NULL	Germany
François	Tremblay	NULL	Canada
Bjørn	Hansen	NULL	Norway
František	Wichterlová	JetBrains s.r.o.	Czech Republic

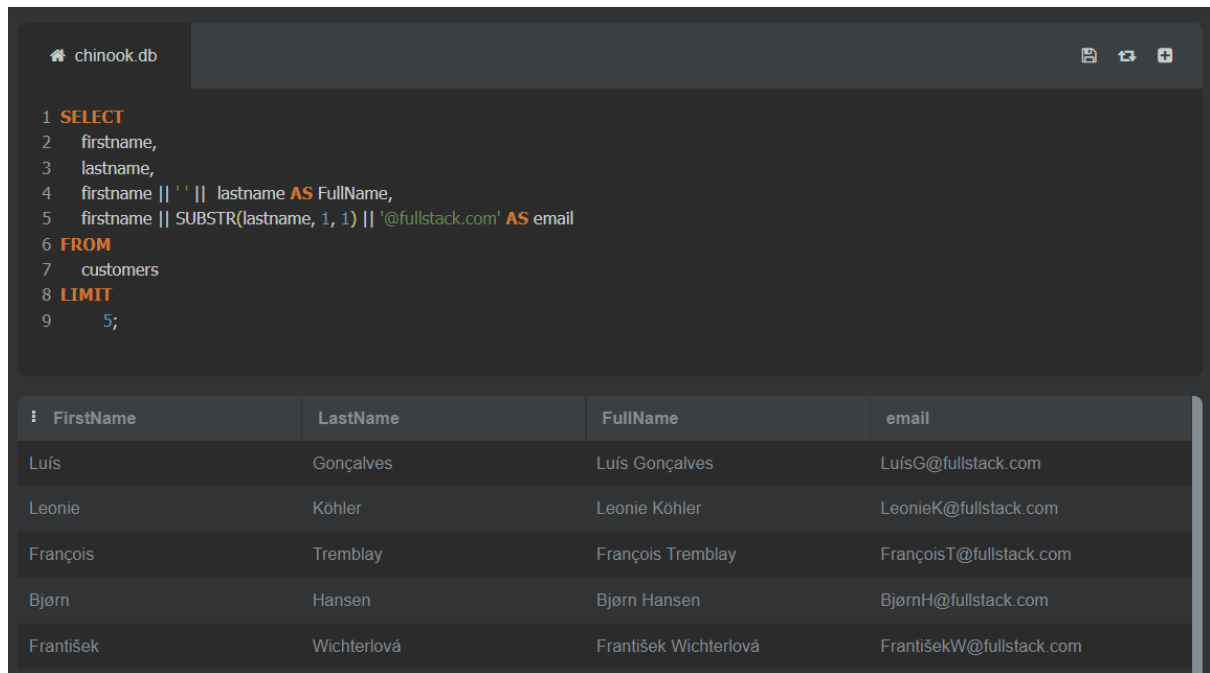
-เราสามารถเชื่อมค่าของ 2 column เข้าด้วยกันได้ เช่น:

chinook.db

```
1 SELECT
2   firstname,
3   lastname,
4   firstname || ' ' || lastname AS FullName,
5   firstname || '@fullstack.com' AS email
6 FROM
7   customers
8 LIMIT
9   5;
```

FirstName	LastName	FullName	email
Luís	Gonçalves	Luís Gonçalves	Luís@fullstack.com
Leonie	Köhler	Leonie Köhler	Leonie@fullstack.com
François	Tremblay	François Tremblay	François@fullstack.com
Bjørn	Hansen	Bjørn Hansen	Bjørn@fullstack.com
František	Wichterlová	František Wichterlová	František@fullstack.com

-เราสามารถใช้ SUBSTR() ในการดึงค่าจาก String ที่ต้องการได้ เช่น:



```

1 SELECT
2   firstname,
3   lastname,
4   firstname || ' ' || lastname AS FullName,
5   firstname || SUBSTR(lastname, 1, 1) || '@fullstack.com' AS email
6 FROM
7   customers
8 LIMIT
9   5;

```

FirstName	LastName	FullName	email
Luis	Gonçalves	Luis Gonçalves	LuisG@fullstack.com
Leonie	Köhler	Leonie Köhler	LeonieK@fullstack.com
François	Tremblay	François Tremblay	FrançoisT@fullstack.com
Bjørn	Hansen	Bjørn Hansen	BjørnH@fullstack.com
František	Wichterlová	František Wichterlová	FrantišekW@fullstack.com

-เราสามารถครอบค่าใน column ได้ด้วย function เช่น UPPER() หรือ LOWER() เพื่อเปลี่ยนให้เป็นตัวพิมพ์ใหญ่หรือพิมพ์เล็กตามลำดับ

-SQL เป็นภาษาที่ Case Insensitive แต่เราจะนิยมเขียน Keyword เช่น SELECT หรือ FROM เป็นตัวพิมพ์ใหญ่

-เราสามารถครอบค่าด้วย ROUND() เพื่อปัดค่าเป็นทศนิยมตำแหน่งที่เราต้องการ เช่น:

```
chinook.db

1 SELECT
2   name,
3   ROUND(milliseconds / 6000.0, 2) AS minutes,
4   ROUND(bytes / (1024*1024.0), 4) AS mb
5 FROM
6   tracks
```

Name	minutes	mb
For Those About To Rock (We Salute You)	57.29	10.6529
Balls to the Wall	57.09	5.2551
Fast As a Shark	38.44	3.8061
Restless and Wild	42.01	4.1311
Princess of the Dawn	62.57	5.9991
Put The Finger On You	34.28	6.4024
Let's Get It Up	38.99	7.2828
Inject The Venom	35.14	6.5354
Snowballed	33.85	6.2937
Evil Walks	43.92	8.2123

-เราสามารถใช `STRFTIME()` เพื่อ Extract วันเดือนปี ออกจาก Datetime ได้ แต่ค่าที่ได้จะออกมาเป็น String เช่น:

chinook.db

```

1 --strtime works with DATETIME
2 --Date format: YYYY-MM-DD
3 SELECT
4   invoicedate,
5   STRFTIME('%Y', invoicedate) AS Year,
6   STRFTIME('%m', invoicedate) AS Month,
7   STRFTIME('%d', invoicedate) AS Day
8 FROM
9   invoices;

```

InvoiceDate	Year	Month	Day
2009-01-01 00:00:00	2009	01	01
2009-01-02 00:00:00	2009	01	02
2009-01-03 00:00:00	2009	01	03
2009-01-06 00:00:00	2009	01	06
2009-01-11 00:00:00	2009	01	11
2009-01-19 00:00:00	2009	01	19
2009-02-01 00:00:00	2009	02	01
2009-02-01 00:00:00	2009	02	01
2009-02-02 00:00:00	2009	02	02
2009-02-03 00:00:00	2009	02	03

-เราสามารถใช้ WHERE ในการกรองข้อมูลตามเงื่อนไขได้ เช่น:

chinook.db

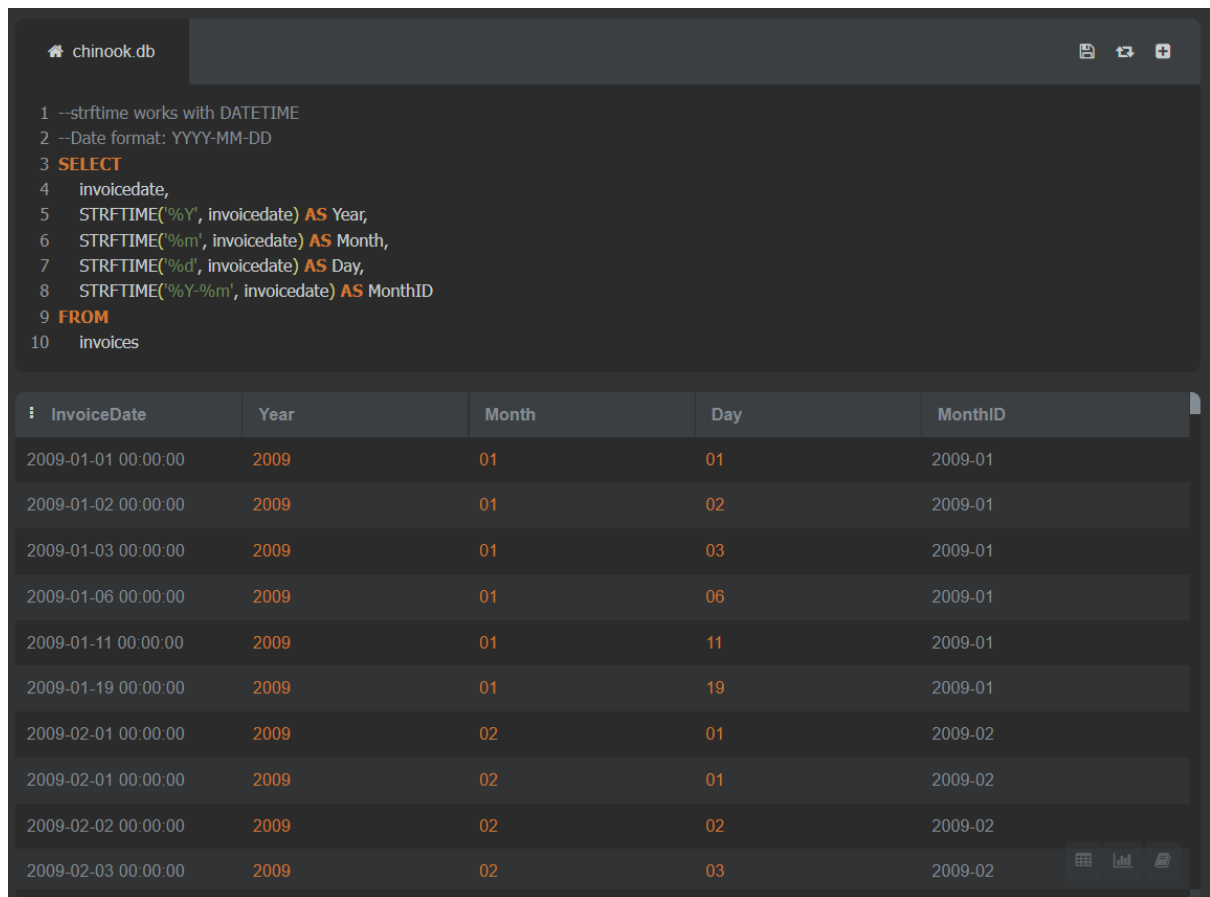
```

1 --strtime works with DATETIME
2 --Date format: YYYY-MM-DD
3 SELECT
4   invoicedate,
5   STRFTIME('%Y', invoicedate) AS Year,
6   STRFTIME('%m', invoicedate) AS Month,
7   STRFTIME('%d', invoicedate) AS Day
8 FROM
9   invoices
10 WHERE
11   Month = '04' AND Year = '2009';

```

InvoiceDate	Year	Month	Day
2009-04-04 00:00:00	2009	04	04
2009-04-04 00:00:00	2009	04	04
2009-04-05 00:00:00	2009	04	05
2009-04-06 00:00:00	2009	04	06
2009-04-09 00:00:00	2009	04	09
2009-04-14 00:00:00	2009	04	14
2009-04-22 00:00:00	2009	04	22

-เราสามารถรวม Year กับ Month เป็น MonthID รวมปีกับเดือนเข้าด้วยกันได้ เช่น:



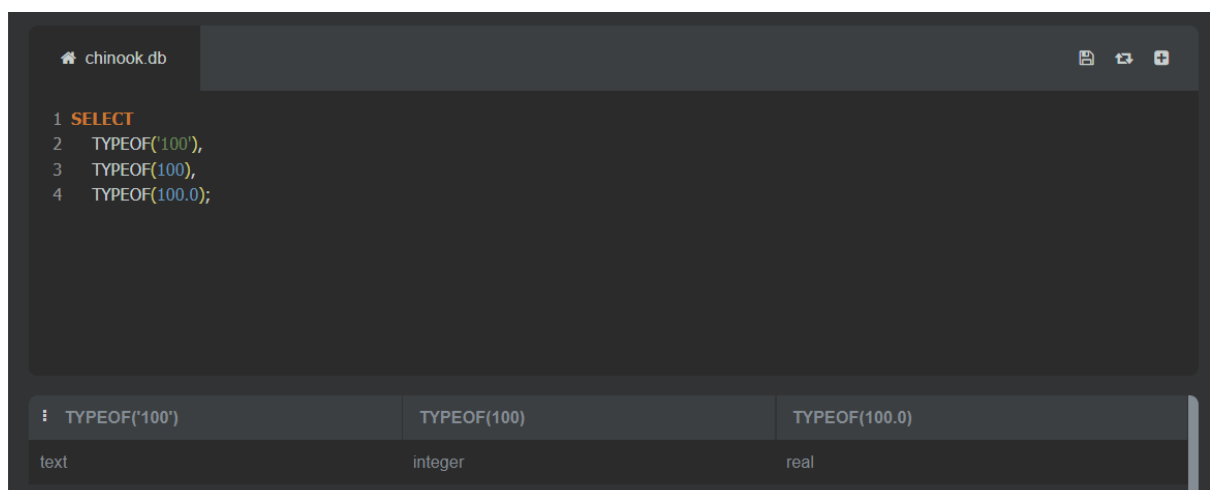
The screenshot shows a SQL query in a database client interface. The query is as follows:

```
1 --strftime works with DATETIME
2 --Date format: YYYY-MM-DD
3 SELECT
4   invoicedate,
5   STRFTIME('%Y', invoicedate) AS Year,
6   STRFTIME('%m', invoicedate) AS Month,
7   STRFTIME('%d', invoicedate) AS Day,
8   STRFTIME('%Y-%m', invoicedate) AS MonthID
9 FROM
10  invoices
```

The result is a table with the following columns: InvoiceDate, Year, Month, Day, and MonthID. The data shows invoices from 2009-01-01 to 2009-02-03, with the MonthID column combining the Year and Month.

InvoiceDate	Year	Month	Day	MonthID
2009-01-01 00:00:00	2009	01	01	2009-01
2009-01-02 00:00:00	2009	01	02	2009-01
2009-01-03 00:00:00	2009	01	03	2009-01
2009-01-06 00:00:00	2009	01	06	2009-01
2009-01-11 00:00:00	2009	01	11	2009-01
2009-01-19 00:00:00	2009	01	19	2009-01
2009-02-01 00:00:00	2009	02	01	2009-02
2009-02-01 00:00:00	2009	02	01	2009-02
2009-02-02 00:00:00	2009	02	02	2009-02
2009-02-03 00:00:00	2009	02	03	2009-02

-ประเภทต่าง ๆ ของข้อมูล:



The screenshot shows a SQL query in a database client interface. The query is as follows:

```
1 SELECT
2   TYPEOF('100'),
3   TYPEOF(100),
4   TYPEOF(100.0);
```

The result is a table with the following columns: TYPEOF('100'), TYPEOF(100), and TYPEOF(100.0). The data shows the types of the respective values: text, integer, and real.

TYPEOF('100')	TYPEOF(100)	TYPEOF(100.0)
text	integer	real

-เราสามารถ Cast (เปลี่ยนประเภทของข้อมูล text, int, real) ใน SQL ได้ เช่น:

The screenshot shows a database client window titled 'chinook.db'. The SQL editor contains the query: `1 SELECT TYPEOF(CAST('100.00' AS REAL));`. The results pane below shows the output: `1 TYPEOF(CAST('100.00' AS REAL))` with the value `real`.

*พยายามอย่า Cast อะไรแปลก ๆ เช่น Cast Hello World ให้กลายเป็น Real เป็นต้น

-ในกรณีที่เราไม่มั่นใจว่าค่าใน column เป็นตัวพิมพ์ใหญ่หรือพิมพ์เล็ก เราสามารถประยุกต์การใช้ UPPER() หรือ LOWER() เพื่อใช้ในการหาค่าจาก column ได้ เช่น:

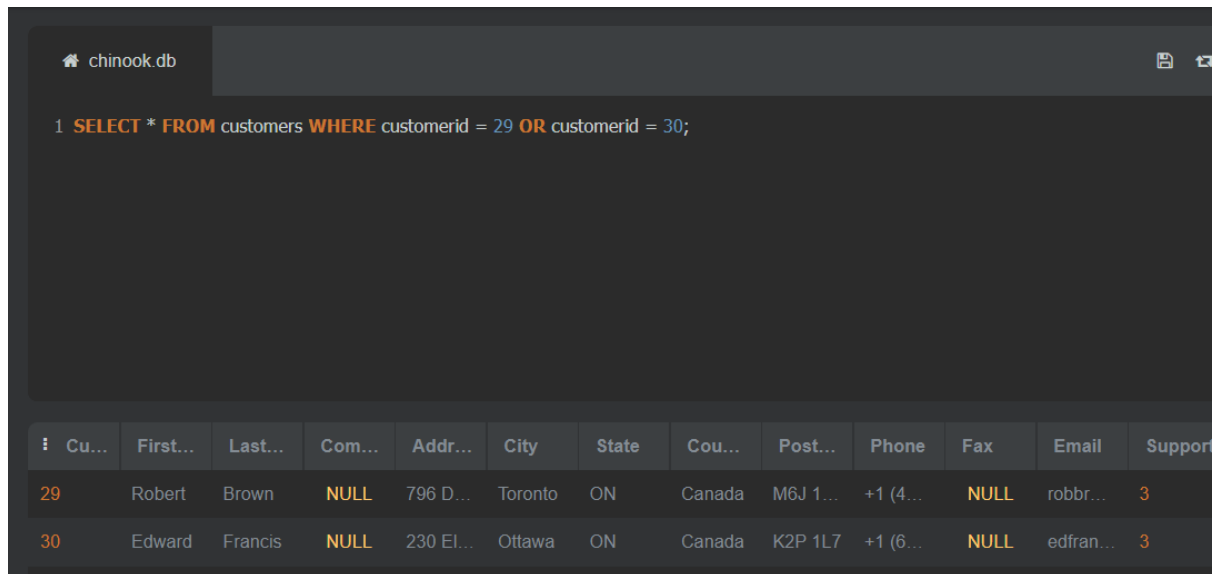
The screenshot shows a database client window titled 'chinook.db'. The SQL editor contains the query: `1 SELECT * FROM customers WHERE UPPER(country) IN ('UNITED KINGDOM', 'USA', 'CANADA');`. The results pane displays a table with 13 columns: Cu..., First..., Last..., Com..., Addr..., City, State, Cou..., Post..., Phone, Fax, Email, and SupportRe... The table contains 10 rows of data, with the 'Cou...' column showing 'Canada' or 'USA'.

	Cu...	First...	Last...	Com...	Addr...	City	State	Cou...	Post...	Phone	Fax	Email	SupportRe...
3		François	Trem...	NULL	1498 r...	Montr...	QC	Canada	H2G ...	+1 (5...	NULL	ftremb...	3
14		Mark	Philips	Telus	8210 ...	Edmo...	AB	Canada	T6G 2...	+1 (7...	+1 (7...	mphili...	5
15		Jennifer	Peter...	Roger...	700 ...	Vanco...	BC	Canada	V6C 1...	+1 (6...	+1 (6...	jennif...	3
16		Frank	Harris	Googl...	1600 ...	Mount...	CA	USA	94043...	+1 (6...	+1 (6...	fharris...	4
17		Jack	Smith	Micro...	1 Micr...	Redm...	WA	USA	98052...	+1 (4...	+1 (4...	jacks...	5
18		Michelle	Brooks	NULL	627 B...	New Y...	NY	USA	10012...	+1 (2...	+1 (2...	miche...	3
19		Tim	Goyer	Apple ...	1 Infin...	Cuper...	CA	USA	95014	+1 (4...	+1 (4...	tgoyer...	3
20		Dan	Miller	NULL	541 D...	Mount...	CA	USA	94040...	+1 (6...	NULL	dmille...	4
21		Kathy	Chase	NULL	801 ...	Reno	NV	USA	89503	+1 (7...	NULL	kacha...	5
22		Heather	Leacock	NULL	120 S...	Oriando	FL	USA	32801	+1 (4...	NULL	hleac...	4

-ในโลกความเป็นจริง WHERE clause สำคัญมาก ยิ่งดึงข้อมูลได้ Specific ตรงตามความต้องการเท่าไรก็ยิ่งดี

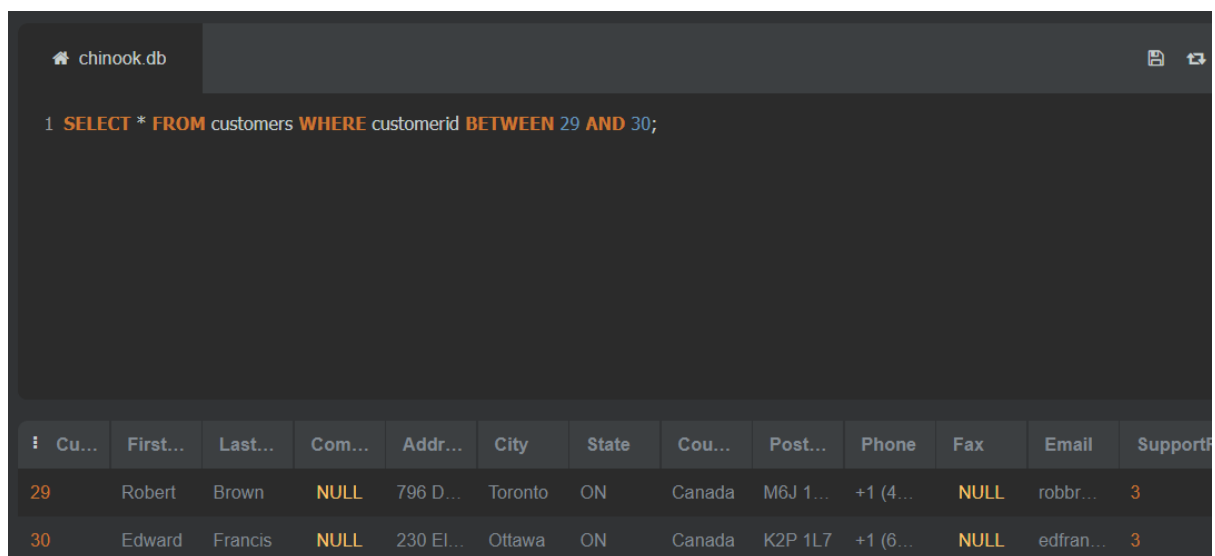
Part 2

-เราสามารถใช่ AND หรือ OR เชื่อมค่าความจริงเป็นเงื่อนไขใน WHERE ได้ เช่น:



The screenshot shows a database interface with a query editor and a results table. The query is: `1 SELECT * FROM customers WHERE customerid = 29 OR customerid = 30;`. The results table has 13 columns: Cu..., First..., Last..., Com..., Addr..., City, State, Cou..., Post..., Phone, Fax, Email, and SupportF. It contains two rows of data for customerid 29 and 30.

	Cu...	First...	Last...	Com...	Addr...	City	State	Cou...	Post...	Phone	Fax	Email	SupportF
29		Robert	Brown	NULL	796 D...	Toronto	ON	Canada	M6J 1...	+1 (4...	NULL	robbr...	3
30		Edward	Francis	NULL	230 El...	Ottawa	ON	Canada	K2P 1L7	+1 (6...	NULL	edfran...	3



The screenshot shows a database interface with a query editor and a results table. The query is: `1 SELECT * FROM customers WHERE customerid BETWEEN 29 AND 30;`. The results table is identical to the one in the previous screenshot, showing two rows of data for customerid 29 and 30.

	Cu...	First...	Last...	Com...	Addr...	City	State	Cou...	Post...	Phone	Fax	Email	SupportF
29		Robert	Brown	NULL	796 D...	Toronto	ON	Canada	M6J 1...	+1 (4...	NULL	robbr...	3
30		Edward	Francis	NULL	230 El...	Ottawa	ON	Canada	K2P 1L7	+1 (6...	NULL	edfran...	3

*ใช้ Between ได้ผลลัพธ์เท่ากัน

-เราสามารถดึงค่าจาก column ที่มีค่า NULL ได้ เช่น:

chinook.db

```
1 SELECT * FROM customers WHERE company IS NULL;
```

	Cu...	First...	Last...	Com...	Addr...	City	State	Cou...	Post...	Phone	Fax	Email	SupportRe...
2		Leonie	Köhler	NULL	Theod...	Stuttgart	NULL	Germ...	70174	+49 0...	NULL	leone...	5
3		François	Trem...	NULL	1498 r...	Montr...	QC	Canada	H2G ...	+1 (5...	NULL	ftremb...	3
4		Bjørn	Hansen	NULL	Ullevå...	Oslo	NULL	Norway	0171	+47 2...	NULL	bjorn...	4
6		Helena	Holý	NULL	Rilská...	Prague	NULL	Czech...	14300	+420 ...	NULL	hholy...	5
7		Astrid	Gruber	NULL	Roten...	Vienne	NULL	Austria	1010	+43 0...	NULL	astrid...	5
8		Daan	Peeters	NULL	Grétry...	Brussels	NULL	Belgium	1000	+32 0...	NULL	daan_...	4
9		Kara	Nielsen	NULL	Sønd...	Cope...	NULL	Denm...	1720	+453 ...	NULL	kara.n...	4
13		Ferna...	Ramos	NULL	Qe 7 ...	Brasília	DF	Brazil	71020...	+55 (...)	+55 (...)	fernad...	4
18		Michelle	Brooks	NULL	627 B...	New Y...	NY	USA	10012...	+1 (2...	+1 (2...	miche...	3
20		Dan	Miller	NULL	541 D...	Mount...	CA	USA	94040...	+1 (6...	NULL	dmille...	4

-ถ้า Primary Key ของ Table หนึ่ง ไปเป็น column ของอีก Table หนึ่ง เราจะเรียก column นั้นว่า Foreign Key

-การ INNER JOIN ด้วย WHERE:

chinook.db

```

1 SELECT * FROM artists, albums
2 WHERE artists.artistid = albums.artistid;

```

Artistid	Name	AlbumId	Title	ArtistId
1	AC/DC	1	For Those About To Roc...	1
2	Accept	2	Balls to the Wall	2
2	Accept	3	Restless and Wild	2
1	AC/DC	4	Let There Be Rock	1
3	Aerosmith	5	Big Ones	3
4	Alanis Morissette	6	Jagged Little Pill	4
5	Alice In Chains	7	Facelift	5
6	Antônio Carlos Jobim	8	Warner 25 Anos	6
7	Apocalyptica	9	Plays Metallica By Four ...	7
8	Audioslave	10	Audioslave	8

chinook.db

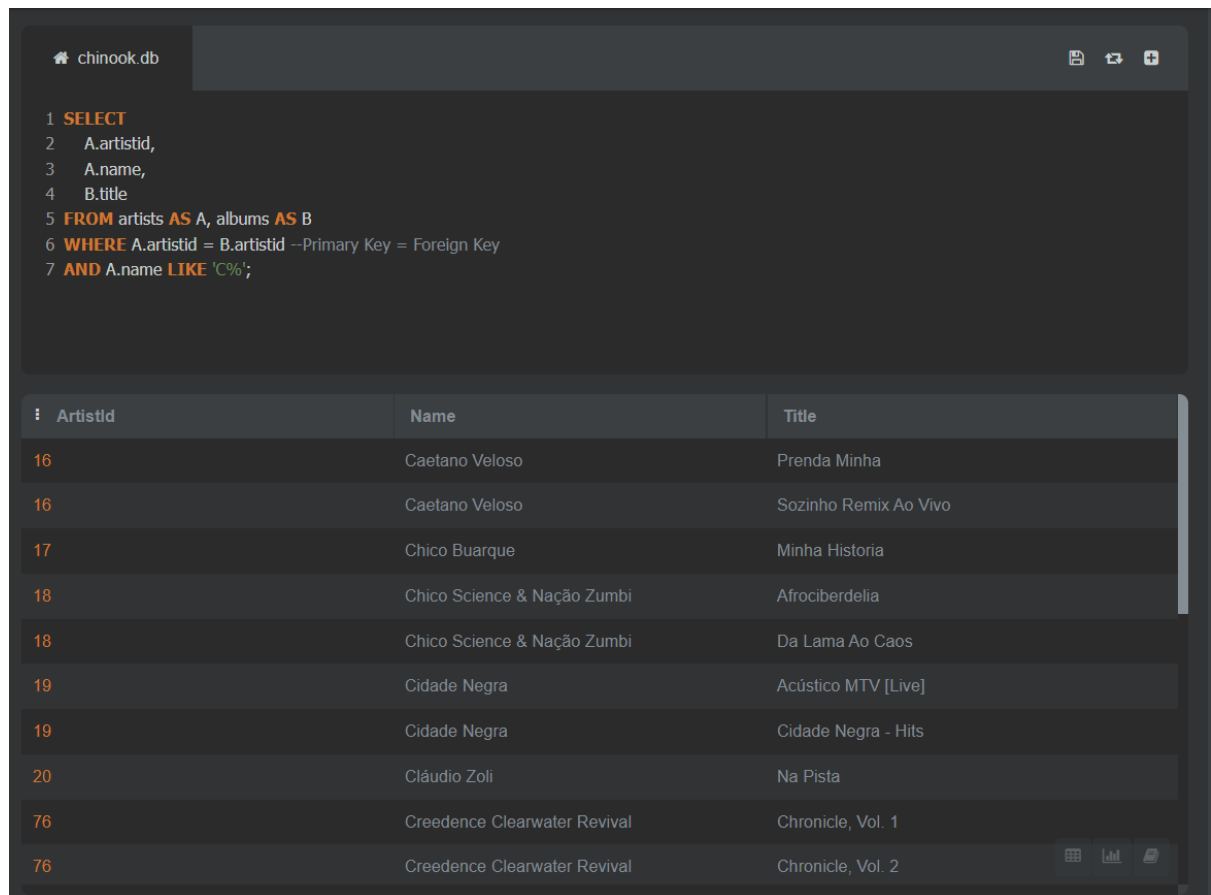
```

1 SELECT
2   A.artistid,
3   A.name,
4   B.title
5 FROM artists AS A, albums AS B
6 WHERE A.artistid = B.artistid;

```

Artistid	Name	Title
1	AC/DC	For Those About To Rock We Salute You
2	Accept	Balls to the Wall
2	Accept	Restless and Wild
1	AC/DC	Let There Be Rock
3	Aerosmith	Big Ones
4	Alanis Morissette	Jagged Little Pill
5	Alice In Chains	Facelift
6	Antônio Carlos Jobim	Warner 25 Anos
7	Apocalyptica	Plays Metallica By Four Cellos
8	Audioslave	Audioslave

-เราสามารถใช่ LIKE ทำ Pattern Matching ได้ เช่น:



The screenshot shows a SQL query in a database client window titled 'chinook.db'. The query is as follows:

```
1 SELECT
2   A.artistid,
3   A.name,
4   B.title
5 FROM artists AS A, albums AS B
6 WHERE A.artistid = B.artistid --Primary Key = Foreign Key
7 AND A.name LIKE 'C%';
```

Below the query, the results are displayed in a table with three columns: Artistid, Name, and Title. The results show 10 rows of data where the artist's name starts with 'C'.

Artistid	Name	Title
16	Caetano Veloso	Prenda Minha
16	Caetano Veloso	Sozinho Remix Ao Vivo
17	Chico Buarque	Minha História
18	Chico Science & Nação Zumbi	Afrociberdelia
18	Chico Science & Nação Zumbi	Da Lama Ao Caos
19	Cidade Negra	Acústico MTV [Live]
19	Cidade Negra	Cidade Negra - Hits
20	Cláudio Zoli	Na Pista
76	Creedence Clearwater Revival	Chronicle, Vol. 1
76	Creedence Clearwater Revival	Chronicle, Vol. 2

-INNER JOIN (ไม่ต้องเขียน WHERE clause):

chinook.db

1 SELECT

2 A.artistid,

3 A.name,

4 B.title

5 FROM artists AS A JOIN albums AS B

6 ON A.artistid = B.artistid --Primary Key = Foreign Key

7 AND A.name LIKE 'C%'

8 AND B.title LIKE 'C%';

Artistid	Name	Title
19	Cidade Negra	Cidade Negra - Hits
76	Creedence Clearwater Revival	Chronicle, Vol. 1
76	Creedence Clearwater Revival	Chronicle, Vol. 2
77	Cássia Eller	Cássia Eller - Coleção Sem Limite [Disc 2]
77	Cássia Eller	Cássia Eller - Sem Limite [Disc 1]
196	Cake	Cake: B-Sides and Rarities
205	Chris Cornell	Carry On
253	Calexico	Carried to Dust (Bonus Track Version)

-ดึงข้อมูลมากกว่า 2 Table:

chinook.db

chinook.db.1

1 SELECT *

2 FROM artists, albums, tracks




3 WHERE artists.ArtistId = albums.artistid

4 AND albums.albumid = tracks.albumid;

A...	Name	Alb...	Title	Arti...	Trac...	Name	Alb...	Med...	Gen...	Co...	Milli...	Bytes	UnitPrice
1	AC/DC	1	For T...	1	1	For T...	1	1	1	Angu...	343719	1117...	0.99
1	AC/DC	1	For T...	1	6	Put T...	1	1	1	Angu...	205662	6713...	0.99
1	AC/DC	1	For T...	1	7	Let's ...	1	1	1	Angu...	233926	7636...	0.99
1	AC/DC	1	For T...	1	8	Inject...	1	1	1	Angu...	210834	6852...	0.99
1	AC/DC	1	For T...	1	9	Snow...	1	1	1	Angu...	203102	6599...	0.99
1	AC/DC	1	For T...	1	10	Evil ...	1	1	1	Angu...	263497	8611...	0.99
1	AC/DC	1	For T...	1	11	C.O.D.	1	1	1	Angu...	199836	6566...	0.99
1	AC/DC	1	For T...	1	12	Brea...	1	1	1	Angu...	263288	8596...	0.99
1	AC/DC	1	For T...	1	13	Night...	1	1	1	Angu...	205688	6706...	0.99
1	AC/DC	1	For T...	1	14	Spell...	1	1	1	Angu...	270863	8817...	0.99




chinook.db

chinook.db.1



```
1 SELECT
2   artists.artistid,
3   artists.name Artist_Name,
4   albums.title Album_Name,
5   tracks.name Song,
6   genres.name Genre_Name
7 FROM artists, albums, tracks, genres
8 WHERE artists.ArtistId = albums.artistid
9 AND albums.albumid = tracks.albumid
10 AND tracks.genreid = genres.genreid;
```

ArtistId	Artist_Name	Album_Name	Song	Genre_Name
1	AC/DC	For Those About To Roc...	For Those About To Roc...	Rock
1	AC/DC	For Those About To Roc...	Put The Finger On You	Rock
1	AC/DC	For Those About To Roc...	Let's Get It Up	Rock
1	AC/DC	For Those About To Roc...	Inject The Venom	Rock
1	AC/DC	For Those About To Roc...	Snowballed	Rock
1	AC/DC	For Those About To Roc...	Evil Walks	Rock
1	AC/DC	For Those About To Roc...	C.O.D.	Rock
1	AC/DC	For Those About To Roc...	Breaking The Rules	Rock
1	AC/DC	For Those About To Roc...	Night Of The Long Knives	Rock
1	AC/DC	For Those About To Roc...	Spellbound	Rock



-เปลี่ยนจาก WHERE ให้เป็น JOIN:

chinook.db

chinook.db.1

```
1 SELECT
2   artists.artistid,
3   artists.name Artist_Name,
4   albums.title Album_Name,
5   tracks.name Song,
6   genres.name Genre_Name
7 FROM artists JOIN albums JOIN tracks JOIN genres
8 ON artists.ArtistId = albums.artistid
9    AND albums.albumid = tracks.albumid
10   AND tracks.genreid = genres.genreid;
```

ArtistId	Artist_Name	Album_Name	Song	Genre_Name
1	AC/DC	For Those About To Rock We Salute You	For Those About To Roc...	Rock
1	AC/DC	For Those About To Rock We Salute You	Put The Finger On You	Rock
1	AC/DC	For Those About To Rock We Salute You	Let's Get It Up	Rock
1	AC/DC	For Those About To Rock We Salute You	Inject The Venom	Rock
1	AC/DC	For Those About To Rock We Salute You	Snowballed	Rock
1	AC/DC	For Those About To Rock We Salute You	Evil Walks	Rock
1	AC/DC	For Those About To Rock We Salute You	C.O.D.	Rock
1	AC/DC	For Those About To Rock We Salute You	Breaking The Rules	Rock
1	AC/DC	For Those About To Rock We Salute You	Night Of The Long Knives	Rock
1	AC/DC	For Those About To Rock We Salute You	Spellbound	Rock

The screenshot shows a SQL query in SQL Workshop that joins the artists, albums, tracks, and genres tables. The query is as follows:

```

1 SELECT
2   artists.artistid,
3   artists.name Artist_Name,
4   albums.title Album_Name,
5   tracks.name Song,
6   genres.name Genre_Name
7 FROM artists
8 JOIN albums ON artists.ArtistId = albums.artistid
9 JOIN tracks ON albums.albumid = tracks.albumid
10 JOIN genres ON tracks.genreid = genres.genreid;

```

The results table shows 10 rows of data for the artist AC/DC, all from the album 'For Those About To Rock We Salute You'. The songs listed are: 'For Those About To Rock...', 'Put The Finger On You', 'Let's Get It Up', 'Inject The Venom', 'Snowballed', 'Evil Walks', 'C.O.D.', 'Breaking The Rules', 'Night Of The Long Knives', and 'Spellbound'. All songs are categorized as 'Rock'.

Artistid	Artist_Name	Album_Name	Song	Genre_Name
1	AC/DC	For Those About To Rock We Salute You	For Those About To Rock...	Rock
1	AC/DC	For Those About To Rock We Salute You	Put The Finger On You	Rock
1	AC/DC	For Those About To Rock We Salute You	Let's Get It Up	Rock
1	AC/DC	For Those About To Rock We Salute You	Inject The Venom	Rock
1	AC/DC	For Those About To Rock We Salute You	Snowballed	Rock
1	AC/DC	For Those About To Rock We Salute You	Evil Walks	Rock
1	AC/DC	For Those About To Rock We Salute You	C.O.D.	Rock
1	AC/DC	For Those About To Rock We Salute You	Breaking The Rules	Rock
1	AC/DC	For Those About To Rock We Salute You	Night Of The Long Knives	Rock
1	AC/DC	For Those About To Rock We Salute You	Spellbound	Rock

- การ JOIN TABLE ส่งผลต่อ Performance ในการดึงข้อมูล ยิ่ง JOIN มากยิ่งช้า
- ตัวอย่าง Aggregate functions ใน SQL:

The screenshot shows a SQL query in SQL Workshop that uses aggregate functions on the tracks table. The query is as follows:

```

1 --Aggregate functions
2 SELECT
3   AVG(milliseconds),
4   SUM(milliseconds),
5   MIN(milliseconds),
6   MAX(milliseconds),
7   COUNT(milliseconds)
8 FROM tracks;

```

The results table shows the following values:

AVG(milliseconds)	SUM(milliseconds)	MIN(milliseconds)	MAX(milliseconds)	COUNT(milliseconds)
393599.2121039109	1378778040	1071	5286953	3503

- *COUNT ใน SQL จะไม่นับค่า NULL
- เราสามารถเช็ค column ที่มี Missing Value ได้ด้วย COUNT เช่น:

chinook.db	chinook.db.1	chinook.db.2
------------	--------------	--------------

```
1 SELECT COUNT(*), COUNT(company), COUNT(email), COUNT(state) FROM customers;
```

COUNT(*)	COUNT(company)	COUNT(email)	COUNT(state)
59	10	59	30

-column ไหนอยู่ใน SELECT clause ที่ไม่ใช่ Aggregate Function จะต้องอยู่ใน GROUP BY clause เช่น:

chinook.db	chinook.db.1	chinook.db.2
------------	--------------	--------------

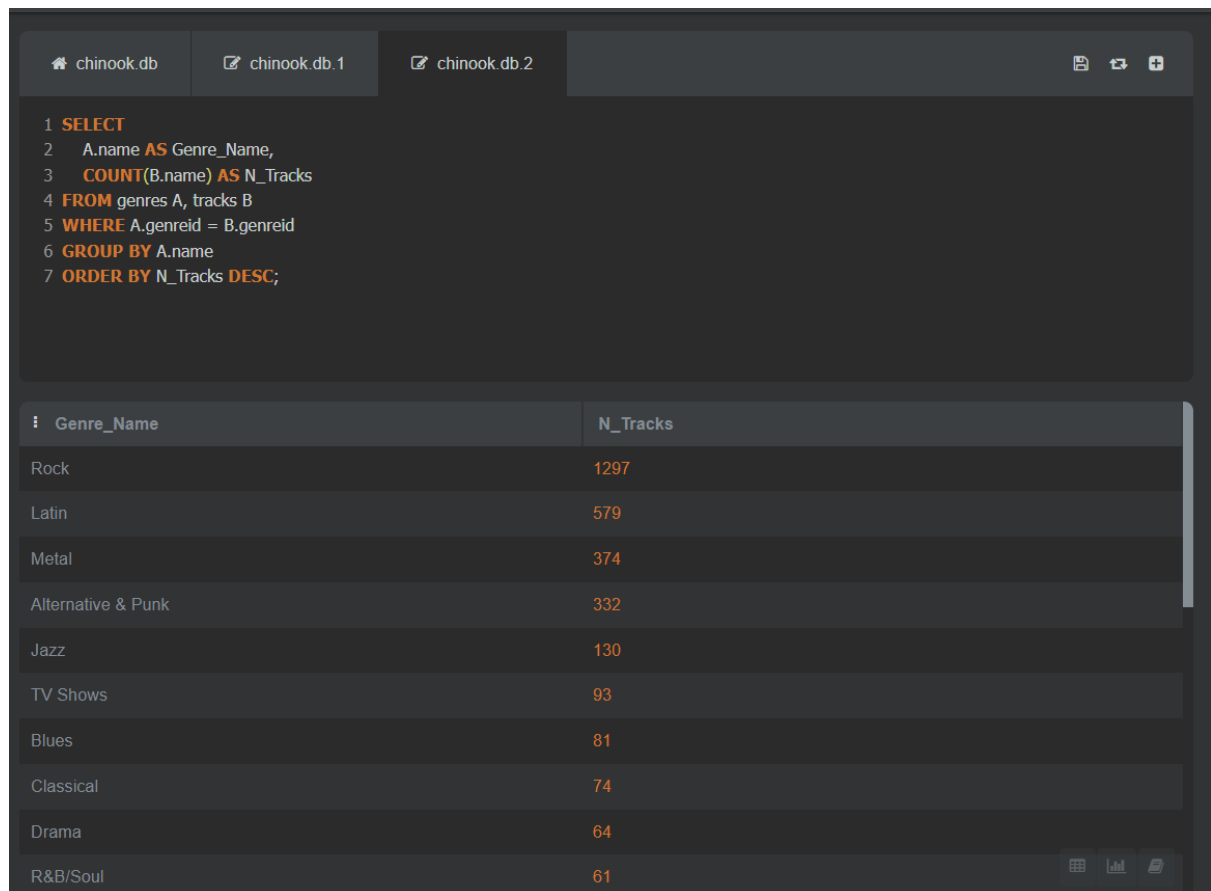
```
1 SELECT country, COUNT(*)
2 FROM customers
3 GROUP BY country
4 ORDER BY COUNT(*) DESC;
```

Country	COUNT(*)
USA	13
Canada	8
France	5
Brazil	5
Germany	4
United Kingdom	3
Portugal	2
India	2
Czech Republic	2
Sweden	1

*ถ้า ORDER BY ไม่ได้ ASC หรือ DESC จะเรียงจากน้อยไปมาก (ASC) เป็น Default

-ORDER BY อยู่หลัง GROUP BY

-เราสามารถใช่ GROUP BY และ ORDER BY ควบคู่กับการ JOIN TABLE ได้เช่นกัน เช่น:



The screenshot shows a SQL IDE interface with three tabs labeled 'chinook.db', 'chinook.db.1', and 'chinook.db.2'. The active tab 'chinook.db' contains a SQL query. Below the query editor, the results of the query are displayed in a table with two columns: 'Genre_Name' and 'N_Tracks'. The results are ordered by 'N_Tracks' in descending order, showing the top 10 genres.

```
1 SELECT
2   A.name AS Genre_Name,
3   COUNT(B.name) AS N_Tracks
4 FROM genres A, tracks B
5 WHERE A.genreid = B.genreid
6 GROUP BY A.name
7 ORDER BY N_Tracks DESC;
```

Genre_Name	N_Tracks
Rock	1297
Latin	579
Metal	374
Alternative & Punk	332
Jazz	130
TV Shows	93
Blues	81
Classical	74
Drama	64
R&B/Soul	61




-Default JOIN ใน SQL คือ INNER JOIN

-ตัวอย่างการหาศิลปินที่ปล่อยอัลบั้มออกมา Top 10 ใน Database:

chinook.db

chinook.db.1

chinook.db.2



```
1 SELECT
2   A.name Artist_Name,
3   COUNT(B.title) N_Albums
4 FROM artists A, albums B
5 WHERE A.artistid = B.artistid
6 GROUP BY 1
7 ORDER BY 2 DESC
8 LIMIT 10;
```

Artist_Name	N_Albums
Iron Maiden	21
Led Zeppelin	14
Deep Purple	11
U2	10
Metallica	10
Ozzy Osbourne	6
Pearl Jam	5
Various Artists	4
Van Halen	4
Lost	4

