



SQL for Data Analyst 101

☰ Tags	Foundation	SQL
↗ Class		
<input checked="" type="checkbox"/> Finished Yet?	<input checked="" type="checkbox"/>	
↗ Knowledge	 The Second Sprint: SQL	

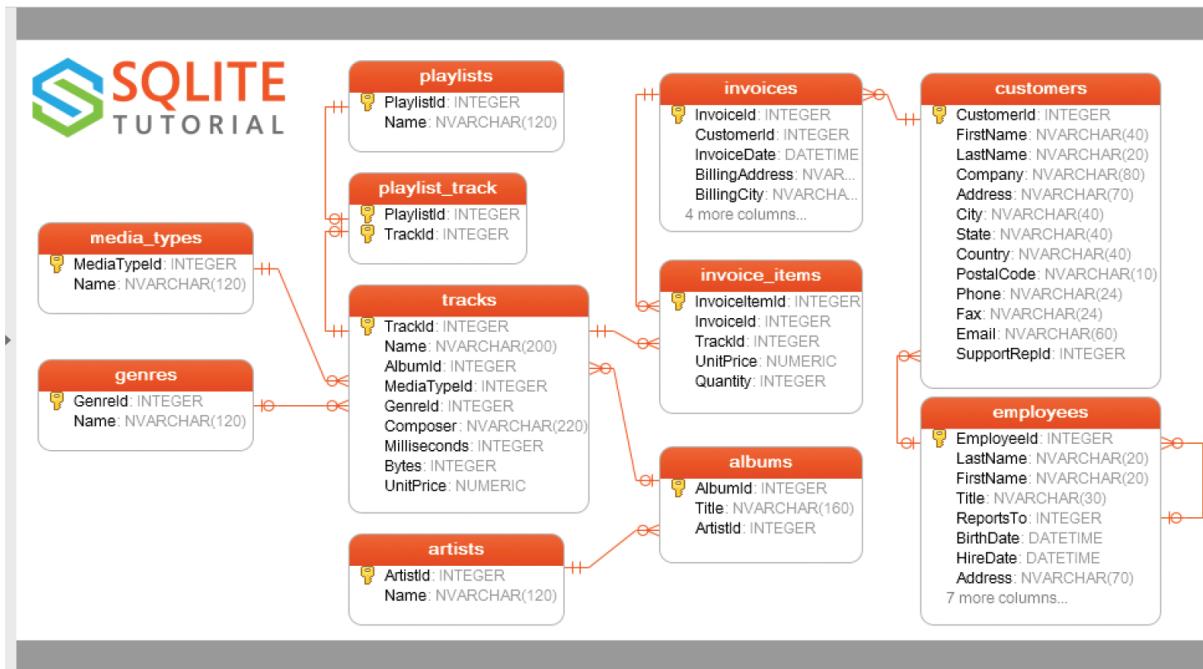
Lesson 1: Database and ER Diagram

-ER (Entity-Relationship) Diagram เปรียบเสมือนแผนผังที่บอกเราว่าข้อมูลในแต่ละ Database มีอะไรบ้าง และความสัมพันธ์ของแต่ละ Table นั้นเป็นอย่างไร

-รูปแบบความสัมพันธ์ที่ใช้ใน Database มี 3 แบบหลัก ๆ คือ

1. One-to-One
2. One-to-Many (ใช้เยอะสุด)
3. Many-to-Many

-ใน Bootcamp ของ Datarockie จะใช้ Database ชื่อ chinook.db เป็นข้อมูลของ Music Store



-ใน 1 Database นั้นประกอบไปด้วยหลาย Table ซึ่งเป็น Structured Data (ข้อมูลที่ถูกจัดเก็บไว้อย่างเป็นระเบียบและมีโครงสร้างชัดเจน ทำให้นำไปใช้ได้ง่ายกว่า Unstructured Data)

-ในแต่ละ column จะมีการบอกประเภทของข้อมูล (ตัวเลข ข้อความ) เรียกว่า Data Type เช่น Int ก็จะได้เฉพาะตัวเลขจำนวนเต็ม เป็นต้น

-แม่กุญแจสีเหลืองในแต่ละตาราง เรียกว่า Primary Key (Unique Identifier) ซึ่ง column ที่เป็น Primary Key (PK) ในตารางจะมีค่าไม่ซ้ำกัน แต่ถ้า column นี้ไปอยู่ในอีกตารางหนึ่ง ก็จะเรียกว่า Foreign Key (FK) ซึ่งสามารถซ้ำกันได้

-การ JOIN TABLE คือการดึงข้อมูลจากหลาย ๆ ตารางพร้อม ๆ กัน ด้วยการนำ Primary Key ไป JOIN กับ Foreign Key เมื่อมันเป็นตัวเลขที่สามารถนำมา Match กันได้

-พอแต่ละ Table มีความสัมพันธ์กัน ก็จะสามารถ JOIN เข้าด้วยกันได้ ตรงนี้คือ [Relationship] ของแต่ละ Table ดังนั้น ER Diagram ก็คือแผนผังที่แสดงถึง Relationship ของแต่ละ Table นั้นเอง

-ข้อดีของ ER Diagram คือ มองแล้วรู้ได้กันทีว่าควรจะ JOIN TABLE ไหนได้ด้วย column ใดบ้าง

Lesson 2: DB Browser Installation

-เราสามารถโหลด DB Browser ได้ที่ <https://sqlitebrowser.org/dl/>

-เราสามารถติดตั้งโปรแกรมตามปกติได้เลย

- Double Click (คลิกสองครั้ง) ที่ไอคอนของ DB Browser เพื่อเปิดตัวโปรแกรมขึ้นมา
- เราสามารถเปิดไฟล์ .db ที่ต้องการได้ง่าย ๆ ด้วยการเลือก File (มุมซ้ายบน) → Open Database → เลือกไฟล์ chinook.db ที่ download มาไว้ก่อนหน้านี้
- เราสามารถคลิกที่แท็บ Browse Data เพื่อตรวจสอบข้อมูลขั้นเบื้องต้นได้
- แท็บที่เราจะใช้เยօงคือ Execute SQL เพราะใช้ในการเขียน Query
- การรัน Query ให้กด CTRL + Enter แล้วผลลัพธ์จะปรากฏที่หน้าจอตรงกลาง ส่วนหน้าจอข้างล่างจะเป็น Log
- เราสามารถแก้ไขตัวอักษรที่พิมพ์ใน Execute SQL ให้กลับไปที่ Edit (มุมซ้ายบน) → Preferences → SQL → ปรับขนาดฟอนต์ให้กับ SQL editor font size → กดปุ่ม save เพื่อบันทึกการเปลี่ยนแปลง
- เราสามารถแก้ไขแบบข้างขวาแสดงแต่ DB Schema (ใน Database นี้มี Table อะไรบ้าง และในแต่ละ Table มี Attribute อะไรบ้าง) ได้ด้วยการตีก้อย่างอื่นออกให้เหลือแต่ DB Schema

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'customers' table with 9 rows of data. The columns are: CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, and Phone. The data includes entries for Luis Goncalves, Leonie Köhler, François Tremblay, Bjørn Hansen, František Wichterlová, Helena Holý, Astrid Gruber, Daan Peeters, and Kara Nielsen. The right side of the interface shows the 'DB Schema' pane, which lists tables (13) such as albums, artists, customers, employees, genres, invoice_items, invoices, media_types, playlist_track, playlists, sqlite_sequence, sqlite_stat1, and tracks; and indices (10) such as IFK_AlbumArtistId, IFK_CustomerSupportRepId, IFK_EmployeeReportsTo, IFK_InvoiceCustomerId, IFK_InvoiceLineInvoiceId, IFK_InvoiceLineTrackId, IFK_PlaylistTrackTrackId, IFK_TrackAlbumId, IFK_TrackGenreId, IFK_TrackMediaTypeId, Views (0), and Triggers (0).

CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	Phone
1	Luis	Goncalves	Embraer - Empresa Brasileira de ...	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	SP	Brazil	12227-000	+55 (12) 3333-1111
2	Leonie	Köhler	NULL	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	+49 0711 1234-5678
3	François	Tremblay	NULL	1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7	+1 (514) 555-1234
4	Bjørn	Hansen	NULL	Ullevålsveien 14	Oslo	NULL	Norway	0171	+47 22 1234-5678
5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague	NULL	Czech Republic	14700	+420 2 222-1111
6	Helena	Holý	NULL	Rilská 3174/6	Prague	NULL	Czech Republic	14300	+420 2 222-1111
7	Astrid	Gruber	NULL	Rotenturmstraße 4, 1010 Innere Stadt	Vienne	NULL	Austria	1010	+43 01 1234-5678
8	Daan	Peeters	NULL	Grétrystraat 63	Brussels	NULL	Belgium	1000	+32 02 1234-5678
9	Kara	Nielsen	NULL	Sønder Boulevard 51	Copenhagen	NULL	Denmark	1720	+453 3 1234-5678

Execution finished without errors.
Results: 99 rows returned in 56ms
At line 1 of 1
SELECT * FROM customers;

Lesson 3: Select Columns

- SELECT ใช้ในการดึงข้อมูลจาก Table ส่วน * หมายความว่าเราต้องการดึงทุก column ของ Table ส่วน FROM ชื่อตาราง หมายความว่า เราต้องการดึงข้อมูลจาก Table ชื่อนี้

SQL1

```
1 SELECT * FROM customers;
```

	CustomerId	FirstName	LastName	Company	Address	City	State	Country	PostalCode	^
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de ...	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	SP	Brazil	12227-000	+55 (1)
2	2	Leони	Кöhler	NULL	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	+49 07
3	3	François	Tremblay	NULL	1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7	+1 (51)
4	4	Bjørn	Hansen	NULL	Ullevålsveien 14	Oslo	NULL	Norway	0171	+47 22
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague	NULL	Czech Republic	14700	+420 2
6	6	Helena	Holý	NULL	Rilská 3174/6	Prague	NULL	Czech Republic	14300	+420 2
7	7	Astrid	Gruber	NULL	Rotenturmstraße 4, 1010 Innere Stadt	Vienne	NULL	Austria	1010	+43 01
8	8	Daan	Peeters	NULL	Grétrystraat 63	Brussels	NULL	Belgium	1000	+32 02
9	9	Kara	Nielsen	NULL	Sønder Boulevard 51	Copenhagen	NULL	Denmark	1720	+453 3

< ^ >

Execution finished without errors.
Result: 59 rows returned in 4ms
At line 1:
SELECT * FROM customers;

[ดึงข้อมูลทุก column จากตาราง customers]

- เราสามารถดึงข้อมูลเฉพาะ column ที่เราสนใจได้ เช่น:

SQL 1

```

1 SELECT FirstName, LastName, Country
2 FROM customers;
```

	FirstName	LastName	Country
1	Luís	Gonçalves	Brazil
2	Leonie	Köhler	Germany
3	François	Tremblay	Canada
4	Bjørn	Hansen	Norway
5	František	Wichterlová	Czech Republic
6	Helena	Holý	Czech Republic
7	Astrid	Gruber	Austria
8	Daan	Peeters	Belgium
9	Kara	Nielsen	Denmark
10	Eduardo	Martins	Brazil

```

Execution finished without errors.
Result: 59 rows returned in 3ms
At line 1:
SELECT FirstName, LastName, Country
FROM customers;
```

[ดึงข้อมูลเฉพาะ column FirstName, LastName, และ Country จากตาราง Customers]

- เราสามารถเปลี่ยนการเขียนโค้ดให้อ่านง่ายขึ้นได้ และผลลัพธ์ที่ออกมานะจะเหมือนกัน

```

1 SELECT
2   FirstName,
3   LastName,
4   Country
5 FROM customers;
```

- เราสามารถ Limit จำนวนแถวที่ต้องการดึงได้ดังนี้:

```
1   SELECT
2       FirstName,
3       LastName,
4       Country
5   FROM customers
6   LIMIT 5;
```

	FirstName	LastName	Country
1	Luís	Gonçalves	Brazil
2	Leonie	Köhler	Germany
3	François	Tremblay	Canada
4	Bjørn	Hansen	Norway
5	František	Wichterlová	Czech Republic

[ดึงข้อมูลเฉพาะ column FirstName, LastName, และ Country เฉพาะ 5 แถวแรก จากตาราง Customers]

Lesson 4: Create New Columns

- เราสามารถสร้าง column ใหม่ได้โดยอิงจาก column เดิม เช่น:

```

SQL 1
1 SELECT
2     FirstName,
3     LastName,
4     FirstName || ' ' || LastName AS FullName
5 FROM customers;

```

	FirstName	LastName	FullName
1	Luís	Gonçalves	Luís Gonçalves
2	Leonie	Köhler	Leonie Köhler
3	François	Tremblay	François Tremblay
4	Bjørn	Hansen	Bjørn Hansen
5	František	Wichterlová	František Wichterlová
6	Helena	Holý	Helena Holý
7	Astrid	Gruber	Astrid Gruber
8	Daan	Peeters	Daan Peeters
9	Kara	Nielsen	Kara Nielsen
10	Eduardo	Martins	Eduardo Martins

[ดึงข้อมูลเฉพาะ column FirstName, LastName, และ FullName ที่ได้จากการเชื่อมกันระหว่างค่าของ FirstName และ LastName โดยมีช่องว่างคั่นกลาง จากตาราง Customers]

-เราสามารถปรับตัวพิมพ์ใหญ่-ตัวพิมพ์เล็กได้เหมือน =QUERY() ใน Google Sheets เช่น:

```

1 SELECT
2     FirstName,
3     LastName,
4     FirstName || ' ' || LastName AS FullName,
5     LOWER(FirstName) || '@company.com' AS Email
6 FROM customers;

```

	FirstName	LastName	FullName	Email
1	Luís	Gonçalves	Luís Gonçalves	luís@company.com
2	Leonie	Köhler	Leonie Köhler	leonie@company.com
3	François	Tremblay	François Tremblay	francois@company.com
4	Bjørn	Hansen	Bjørn Hansen	bjørn@company.com
5	František	Wichterlová	František Wichterlová	františek@company.com
6	Helena	Holý	Helena Holý	helena@company.com
7	Astrid	Gruber	Astrid Gruber	astrid@company.com
8	Daan	Peeters	Daan Peeters	daan@company.com
9	Kara	Nielsen	Kara Nielsen	kara@company.com
10	Eduardo	Martins	Eduardo Martins	eduardo@company.com

[ดึงข้อมูลเฉพาะ column FirstName, LastName, FullName ที่ได้จากการเชื่อมกับระหว่างค่าของ FirstName และ LastName โดยมีช่องว่างคืนกลาง, และ Email ที่ได้จากการเชื่อมกับระหว่าง FirstName ที่ถูกปรับค่าให้เป็นตัวพิมพ์เล็กกึ่งหนด เชื่อมกับ '@company.com' จากตาราง Customers]

-ตัวอย่างการเปลี่ยน column ในเชิงตัวเลข:

```
1   SELECT
2       Name,
3      Milliseconds / 60000 AS Minute,
4       Bytes / (1024*1024) AS MB
5   FROM tracks;
```

	Name	Minute	MB
1	For Those About To Rock (We Salute...	5	10
2	Balls to the Wall	5	5
3	Fast As a Shark	3	3
4	Restless and Wild	4	4
5	Princess of the Dawn	6	5
6	Put The Finger On You	3	6
7	Let's Get It Up	3	7
8	Inject The Venom	3	6
9	Snowballed	3	6
10	Evil Walks	4	8

[ดึงข้อมูลเฉพาะ column Name, Minute โดยการหารค่าใน Milliseconds ด้วย 60000, และ MB โดยการหารค่าใน Bytes ด้วย 1024*1024 จากตาราง tracks]

-หากต้องการเลขที่ติดกเศษนัยน์ ให้พิมพ์ .0 ต่อท้ายเลขจำนวนเต็ม ผลลัพธ์จะได้ออกมาเป็นเลขติดกเศษนัยน์ เช่น:

```

1  SELECT
2      Name,
3          ROUND(Milliseconds/ 60000.0, 2) AS Minute,
4          ROUND(Bytes/ (1024*1024.0), 2) AS MB
5  FROM tracks;

```

	Name	Minute	MB
1	For Those About To Rock (We Salute...)	5.73	10.65
2	Balls to the Wall	5.71	5.26
3	Fast As a Shark	3.84	3.81
4	Restless and Wild	4.2	4.13
5	Princess of the Dawn	6.26	6.0
6	Put The Finger On You	3.43	6.4
7	Let's Get It Up	3.9	7.28
8	Inject The Venom	3.51	6.54
9	Snowballed	3.39	6.29
10	Evil Walks	4.39	8.21

Lesson 5: CASE WHEN

- CASE WHEN ใช้ในการเขียนเงื่อนไข สร้าง column ใหม่ เขียนใน SELECT clause
- เขียนได้มากกว่า 1 เงื่อนไข ได้พลัตฟอร์มอื่น =IFS() ของ Google Sheets

```

1  SELECT
2      Company,
3      CASE
4          WHEN Company is not NULL THEN 'CORPO'
5          ELSE 'End Customer'
6      END AS Segment
7  FROM customers

```

	Company	Segment
1	Embraer - Empresa Brasileira de ...	CORPO
2	NULL	End Customer
3	NULL	End Customer
4	NULL	End Customer
5	JetBrains s.r.o.	CORPO
6	NULL	End Customer
7	NULL	End Customer
8	NULL	End Customer
9	NULL	End Customer
10	Woodstock Discos	CORPO

[ดึงข้อมูลเฉพาะ column Company โดยกำหนดเงื่อนไขว่า ถ้าค่าใน Company ไม่ใช่ค่า Null และจะเป็น CORPO แต่ถ้าเป็นค่า Null และจะเป็น End Customer จากตาราง customers]

Lesson 6: Working with DateTime in SQLite

- Feature Engineering คือการสร้าง column ใหม่อิงจากข้อมูลของ column ก่อน เก่า
- เราสามารถแยกวันเดือนปีออกจากข้อมูลวันที่ได้ด้วย STRFTIME (ใช้ได้เฉพาะใน SQLite) เช่น:

```

1 SELECT
2     InvoiceDate,
3     STRFTIME ('%Y', InvoiceDate) AS Year,
4     STRFTIME ('%m', InvoiceDate) AS Month,
5     STRFTIME ('%d', InvoiceDate) AS Day
6 FROM invoices;

```

	InvoiceDate	Year	Month	Day
1	2009-01-01 00:00:00	2009	01	01
2	2009-01-02 00:00:00	2009	01	02
3	2009-01-03 00:00:00	2009	01	03
4	2009-01-06 00:00:00	2009	01	06
5	2009-01-11 00:00:00	2009	01	11
6	2009-01-19 00:00:00	2009	01	19
7	2009-02-01 00:00:00	2009	02	01
8	2009-02-01 00:00:00	2009	02	01
9	2009-02-02 00:00:00	2009	02	02
10	2009-02-03 00:00:00	2009	02	03

[ดึงข้อมูลจาก column InvoiceDate โดยแยกค่าปีไว้ใน column Year, เดือนไว้ใน column Month, และวันไว้ใน column Day จากตาราง invoices]

-เราสามารถดึงเป็นปีพร้อมเดือน หรือวันเดือนปีพร้อมกันได้เช่นกัน เช่น:

```

1 SELECT
2     InvoiceDate,
3     STRFTIME ('%Y', InvoiceDate) AS Year,
4     STRFTIME ('%m', InvoiceDate) AS Month,
5     STRFTIME ('%d', InvoiceDate) AS Day,
6     STRFTIME ('%Y-%m', InvoiceDate) AS MonthID
7 FROM invoices;

```

	InvoiceDate	Year	Month	Day	MonthID
1	2009-01-01 00:00:00	2009	01	01	2009-01
2	2009-01-02 00:00:00	2009	01	02	2009-01
3	2009-01-03 00:00:00	2009	01	03	2009-01
4	2009-01-06 00:00:00	2009	01	06	2009-01
5	2009-01-11 00:00:00	2009	01	11	2009-01
6	2009-01-19 00:00:00	2009	01	19	2009-01
7	2009-02-01 00:00:00	2009	02	01	2009-02
8	2009-02-01 00:00:00	2009	02	01	2009-02
9	2009-02-02 00:00:00	2009	02	02	2009-02
10	2009-02-03 00:00:00	2009	02	03	2009-02

[ดึงข้อมูลจาก column InvoiceDate โดยแยกค่าปีไว้ใน column Year, เดือนไว้ใน column Month, วันไว้ใน column Day, และปีพร้อมเดือนไว้ใน column MonthID จากตาราง invoices]

-เราสามารถใส่ WHERE clause ข้างหลังเพื่อใช้กรองข้อมูลได้ด้วย เช่น:

```
1 SELECT
2     InvoiceDate,
3     STRFTIME ('%Y', InvoiceDate) AS Year,
4     STRFTIME ('%m', InvoiceDate) AS Month,
5     STRFTIME ('%d', InvoiceDate) AS Day,
6     STRFTIME ('%Y-%m', InvoiceDate) AS MonthID
7 FROM invoices
8 WHERE Year = '2010';
```

	InvoiceDate	Year	Month	Day	MonthID
1	2010-01-08 00:00:00	2010	01	08	2010-01
2	2010-01-08 00:00:00	2010	01	08	2010-01
3	2010-01-09 00:00:00	2010	01	09	2010-01
4	2010-01-10 00:00:00	2010	01	10	2010-01
5	2010-01-13 00:00:00	2010	01	13	2010-01
6	2010-01-18 00:00:00	2010	01	18	2010-01
7	2010-01-26 00:00:00	2010	01	26	2010-01
8	2010-02-08 00:00:00	2010	02	08	2010-02
9	2010-02-08 00:00:00	2010	02	08	2010-02
10	2010-02-09 00:00:00	2010	02	09	2010-02

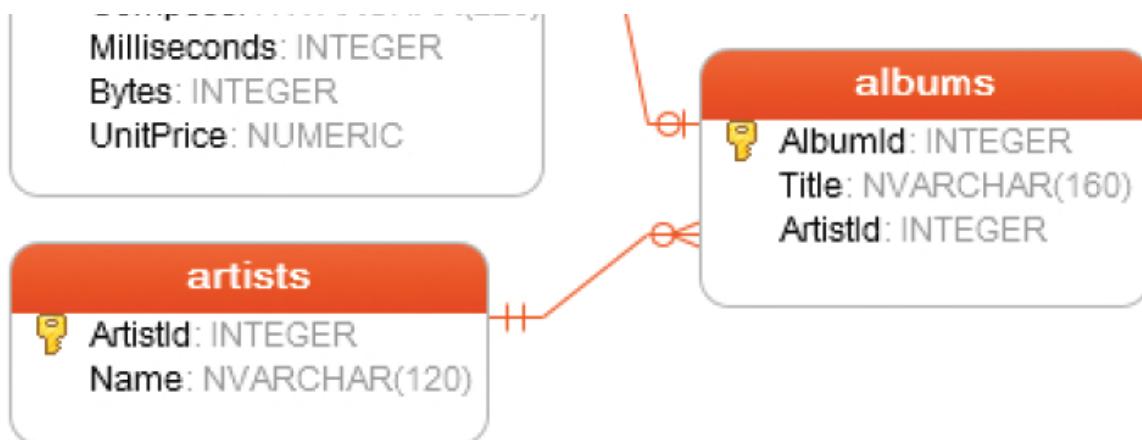
[ดึงข้อมูลจาก column InvoiceDate โดยแยกค่าปีไว้ใน column Year, เดือนไว้ใน column Month, วันไว้ใน column Day, และปีพร้อมเดือนไว้ใน column MonthID จากตาราง invoices โดยเลือกเฉพาะข้อมูลจากปี 2010 เท่านั้น]

-ถ้าเราไม่อยากพิมพ์ ‘‘ ครอบเลขปีที่เราต้องการกรอง เราสามารถครอบให้ข้อมูลของเรามาเป็น Type INT ได้ด้วยการใช้ CAST คุณ เมื่อ:

```
1 SELECT
2     invoicedate,
3     CAST(STRFTIME('%Y', invoicedate) AS INT) AS year,
4     STRFTIME('%m', invoicedate) AS month,
5     STRFTIME('%d', invoicedate) AS day,
6     STRFTIME('%Y-%m', invoicedate) AS monthid
7 FROM invoices
8 WHERE year = 2010;
```

Lesson 7: Select Data From Multiple Tables

-เช็คข้อมูลจาก ER Diagram ก่อนเพื่อดูว่า JOIN TABLE ไหนได้บ้าง เช่น เราสามารถ JOIN ตาราง Artists กับ Albums ได้ด้วย ArtistId:



-การ JOIN TABLE สามารถทำได้ดังนี้:

```
1 SELECT *
2 FROM artists
3 JOIN albums
4 ON artists.ArtistId = albums.ArtistId;
```

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

[ดึงข้อมูลจาก column กั้งหนดจากตาราง artists และ JOIN เข้ากับตาราง albums เนื่องจากใน การ JOIN คือ Primary Key (column ArtistId ของตาราง artists) = Foreign Key (column ArtistId ของตาราง albums)]

-เราสามารถกำหนดเงื่อนไขเพิ่มเติมในการดึงข้อมูลได้ด้วย WHERE เช่น:

```
1 SELECT *
2 FROM artists
3 JOIN albums
4 ON artists.ArtistId = albums.ArtistId
5 WHERE artists.ArtistId = 100;
```

	ArtistId	Name	AlbumId	Title	ArtistId
1	100	Lenny Kravitz	141	Greatest Hits	100

[ดึงข้อมูลจาก column กั้งหนดจากตาราง artists และ JOIN เข้ากับตาราง albums เนื่องจากใน การ JOIN คือ Primary Key (column ArtistId ของตาราง artists) = Foreign Key (column ArtistId ของตาราง albums) โดยมีเงื่อนไขว่า ดึงเฉพาะ row ที่ ArtistID ในตาราง artists = 100 เท่านั้น]

Lesson 8: INNER vs. LEFT JOIN

-JOIN SYNTAX:

```
SELECT
    A.artistid,
    A.name,
    B.title
FROM artists as A
JOIN albums as B
on A.artistid = B.artistid;
```

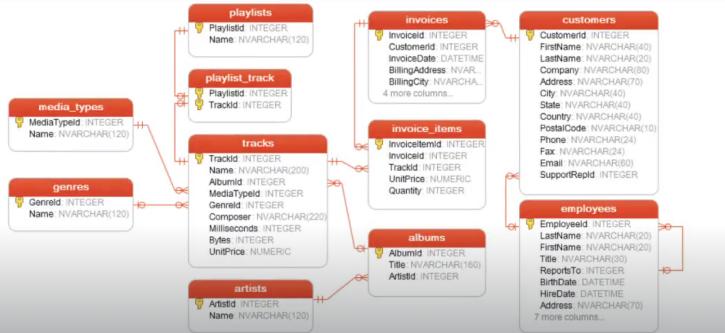
Join Syntax

```

SELECT
    A.artistid,
    A.name,
    B.title
FROM artists AS A
JOIN albums AS B
ON A.artistid = B.artistid;

```

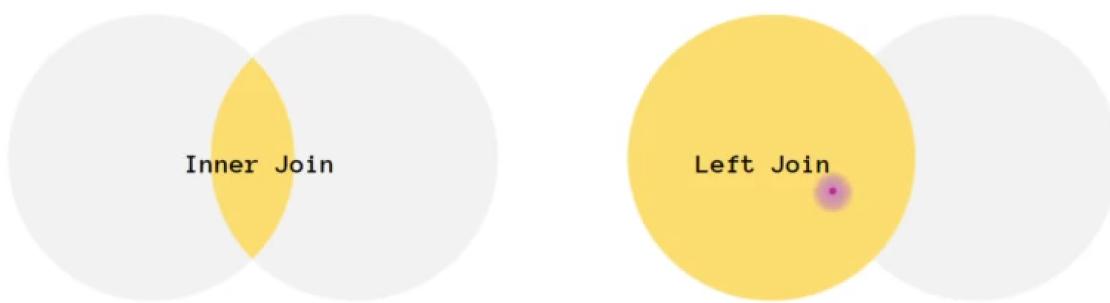
Primary Key = Foreign Key



-INNER JOIN จะ JOIN แบบ Intersect (ต้องอยู่ในส่วนที่ Overlap หรืออยู่ร่วมกันระหว่าง 2 Table)

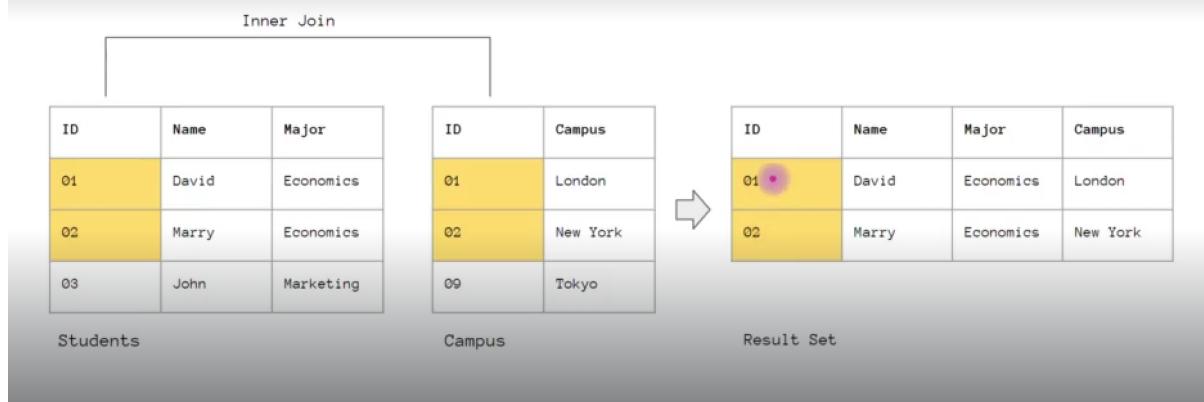
-LEFT JOIN จะเอา Table ซ้ายมือเป็นตัวตั้ง และคืนค่าผลลัพธ์ row ที่ match กันได้ออกมา

Inner vs. Left Join



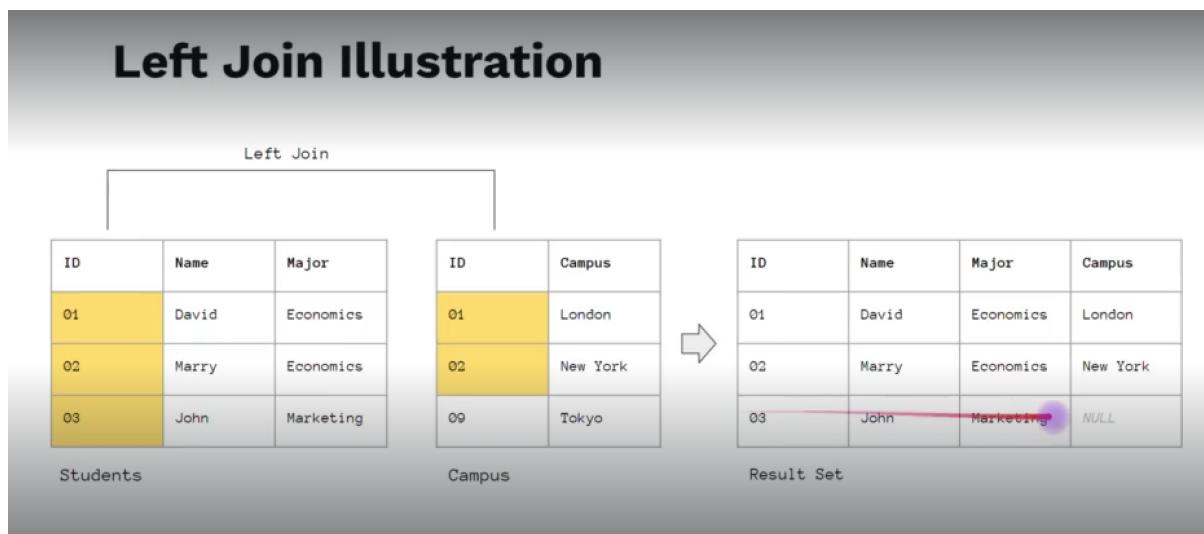
-ผลลัพธ์การ INNER JOIN จะได้เฉพาะส่วนที่ row สามารถ match กันได้เท่านั้น

Inner Join Illustration



-ผลลัพธ์การ LEFT JOIN คือการใช้ Table ตัวซ้ายเป็นตัวตั้ง แล้ว row ไหนที่ค่า match กับก็จะดึงค่ามาเพิ่ม

Left Join Illustration



-ในเบื้อง Syntax การเขียน แค่เพิ่ม INNER หรือ LEFT ตรงหน้า JOIN เพื่อใช้ในการระบุประเภทของ การ JOIN

-จำไว้เสมอว่า INNER JOIN จะคืนค่าเฉพาะ row ที่ match ค่ากันหรือ overlap กัน แต่ LEFT JOIN คือการยก Table ซ้ายมาตั้งรองไว้ก่อน row ไหน match ค่ากันได้ก็ดึงมา แต่ row ที่ไม่ match กันจะแสดงเป็น NULL (Missing Value)

-JOIN โดย default คือ INNER JOIN

```
#INNER JOIN: Returns only row with overlapping values
SELECT
    A.artistid,
    A.name,
    B.title
```

```

FROM artists as A
INNER JOIN albums as B
on A.artistid = B.artistid;

#LEFT JOIN: Returns all rows from the left table, shows NULL if the record doesn't match.
SELECT
    A.artistid,
    A.name,
    B.title
FROM artists as A
LEFT JOIN albums as B
on A.artistid = B.artistid;

```

Lesson 9: More JOIN Examples

- เราสามารถ JOIN TABLE ได้มากกว่า 2 ตาราง และเราสามารถเลือก column ที่เราต้องการจะให้แสดงผลได้ดังนี้:

```

1  SELECT
2      artists.ArtistId,
3      artists.Name,
4      albums.Title,
5      tracks.Name,
6      tracks.Composer
7  FROM artists
8  JOIN albums
9  ON artists.ArtistId = albums.ArtistId
10 JOIN tracks
11 ON albums.AlbumId = tracks.AlbumId;_

```

	ArtistId	Name	Title	Name	Composer
1	1	AC/DC	For Those About To Rock We Salute ...	For Those About To Rock (We Salute...)	Angus Young, Malcolm Young, Brian ...
2	1	AC/DC	For Those About To Rock We Salute ...	Put The Finger On You	Angus Young, Malcolm Young, Brian ...
3	1	AC/DC	For Those About To Rock We Salute ...	Let's Get It Up	Angus Young, Malcolm Young, Brian ...
4	1	AC/DC	For Those About To Rock We Salute ...	Inject The Venom	Angus Young, Malcolm Young, Brian ...
5	1	AC/DC	For Those About To Rock We Salute ...	Snowballed	Angus Young, Malcolm Young, Brian ...
6	1	AC/DC	For Those About To Rock We Salute ...	Evil Walks	Angus Young, Malcolm Young, Brian ...
7	1	AC/DC	For Those About To Rock We Salute ...	C.O.D.	Angus Young, Malcolm Young, Brian ...
8	1	AC/DC	For Those About To Rock We Salute ...	Breaking The Rules	Angus Young, Malcolm Young, Brian ...

[ดึงข้อมูลจาก 3 ตารางคือ artists, albums, และ tracks ให้แสดงผลเฉพาะ column ArtistId และ Name จากตาราง artists, column Title จากตาราง albums, และ column Name กับ Composer จากตาราง tracks โดยเชื่อมตาราง artist กับ albums ด้วย ArtistId และเชื่อมตาราง albums กับ tracks ด้วย AlbumID]

- เราสามารถเปลี่ยนชื่อของตารางเพื่อให้อ้างอิงถึงได้ง่ายขึ้นแต่ได้ผลลัพธ์เหมือนเดิม เช่น:

```

1   SELECT
2       art.artistid,
3       art.name,
4       alb.title,
5       tra.name,
6       tra.composer
7   FROM artists AS art
8   JOIN albums AS alb
9   ON art.artistid = alb.artistid
10  JOIN tracks AS tra
11  ON alb.albumid = tra.albumid
12
13 WHERE art.name = 'Aerosmith';
14

```

[ดึงข้อมูลจาก 3 ตารางคือ artists, albums, และ tracks (เปลี่ยนชื่อเป็นตัวอักษรสามตัวแรกเพื่อให้อ่านง่ายขึ้น) ให้แสดงผลเฉพาะ column ArtistId และ Name จากตาราง artists, column Title จากตาราง albums, และ column Name กับ Composer จากตาราง tracks โดยเชื่อมตาราง artist กับ albums ด้วย ArtistId และเชื่อมตาราง albums กับ tracks ด้วย AlbumID โดยมีเงื่อนไขเพิ่มเติมว่า Name จะต้องเป็น Aerosmith]

Lesson 10: Random Rows

-ทุกครั้งที่ใช้ RANDOM() ค่าก็จะเปลี่ยนไป เป็น function ที่ใช้ในการ random (สุ่ม) ซึ่งเราสามารถนำมาประยุกต์ใช้กับการสุ่มเรียกข้อมูลในฐานข้อมูลได้ เช่น:

```

1   SELECT name, RANDOM() FROM tracks
2   ORDER BY RANDOM();

```

[เลือก column name และ RANDOM (สุ่มเลข) จากตาราง tracks และให้เรียง row แบบค่าที่ถูก random จากน้อยไปมาก การสุ่มแต่ละครั้ง ลำดับของชื่อเพลงที่สุ่มได้ก็จะไม่เหมือนกัน]

-ASC = ASCENDING (น้อยไปมาก), DESC = DESCENDING (มากไปน้อย) ก้าไม่พิมพ์หลัง ORDER BY จะเป็น ASC โดย default

-เราไม่จำเป็นต้องใส่ RANDOM() หลัง SELECT ขอแค่ให้อัญหลัง ORDER BY ก็พอแล้ว เช่น:

```
1 SELECT name FROM tracks
2 ORDER BY RANDOM() DESC
3 LIMIT 1;
```

	Name
1	Kickstart My Heart

(เลือก column name จากตาราง TRACKS และให้เรียง row แบบสุ่ม+มากไปน้อย และจำกัดให้แสดงผลเฉพาะ row แรกสุดเท่านั้น)