

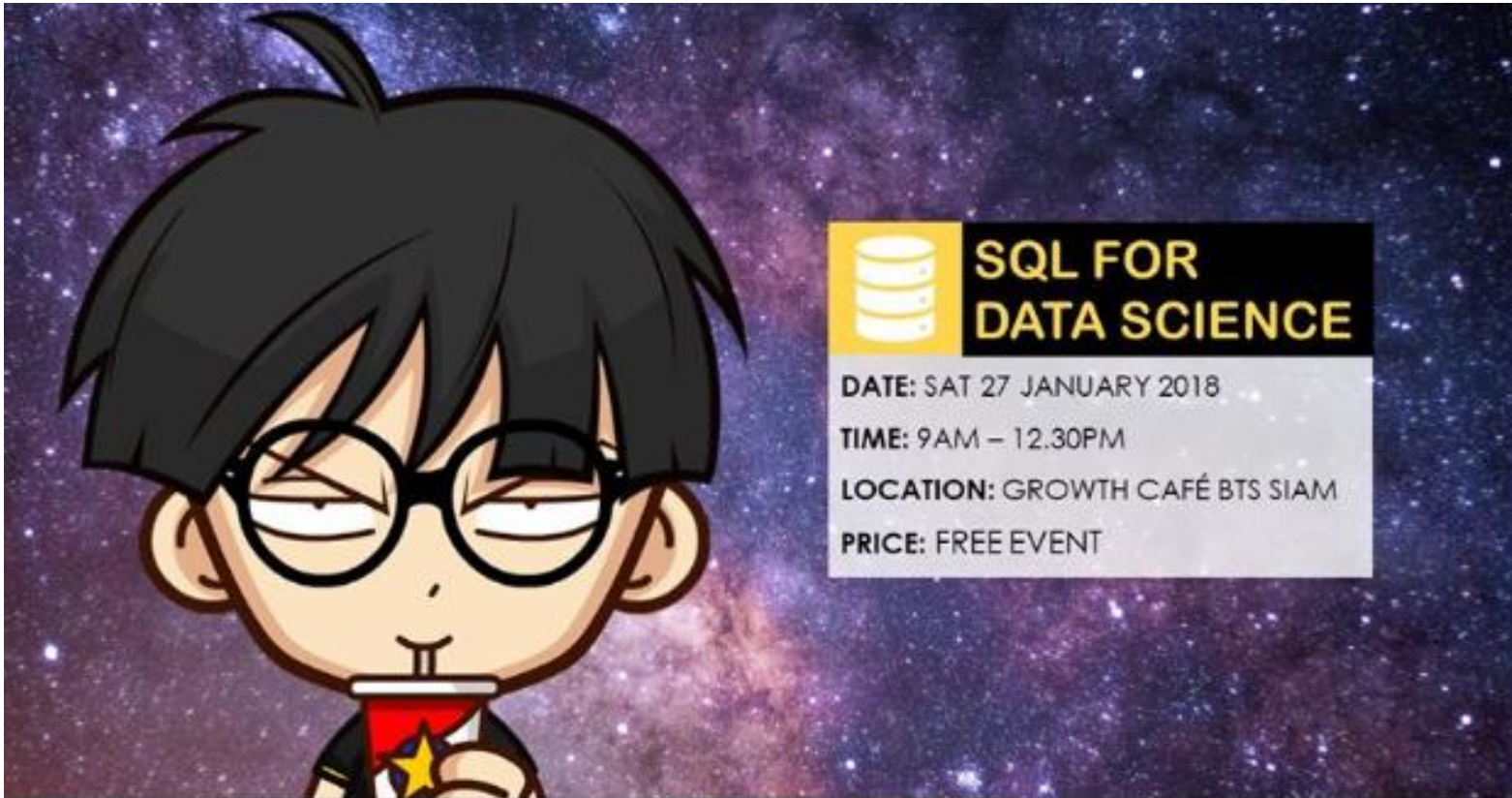


# Intro to SQL (data analyst)

สไลด์โดย DataRockie ธันวาคม 2562

ดาวน์โหลด slide, example database  
และ ER diagram ได้ที่

<https://bit.ly/DATASQL2020>



JAN  
27

## SQL for Data Science สายบู๊บบบบ

Public · Hosted by DataRockie

✓ Going ▼

544 Went · 727 Interested

[See All](#)



Kittipong, Chaiyasit and 18 other friends went

motivations

Who should use SQL?

Everybody should use it.

ทุกคนที่อยู่ใน business ควรเขียน SQL ให้เป็น



Press **F11** to exit full screen

# แอดเริ่มเรียน SQL ตอนสัมภาษณ์ งาน Facebook ตอนปี 2016

SNG - FORTTRAN Canning



SHARE SCREEN...

facebook

[ABOUT](#) • [TERMS](#) • [POLICY](#) • [THIRD PARTY COPYRIGHT](#)

Blue Jeans Network, Inc. © 2016

[HELP CENTER](#)  
+1 (408) 791 2830



Leave



Participants



Share Screen



Settings



Switch Connection



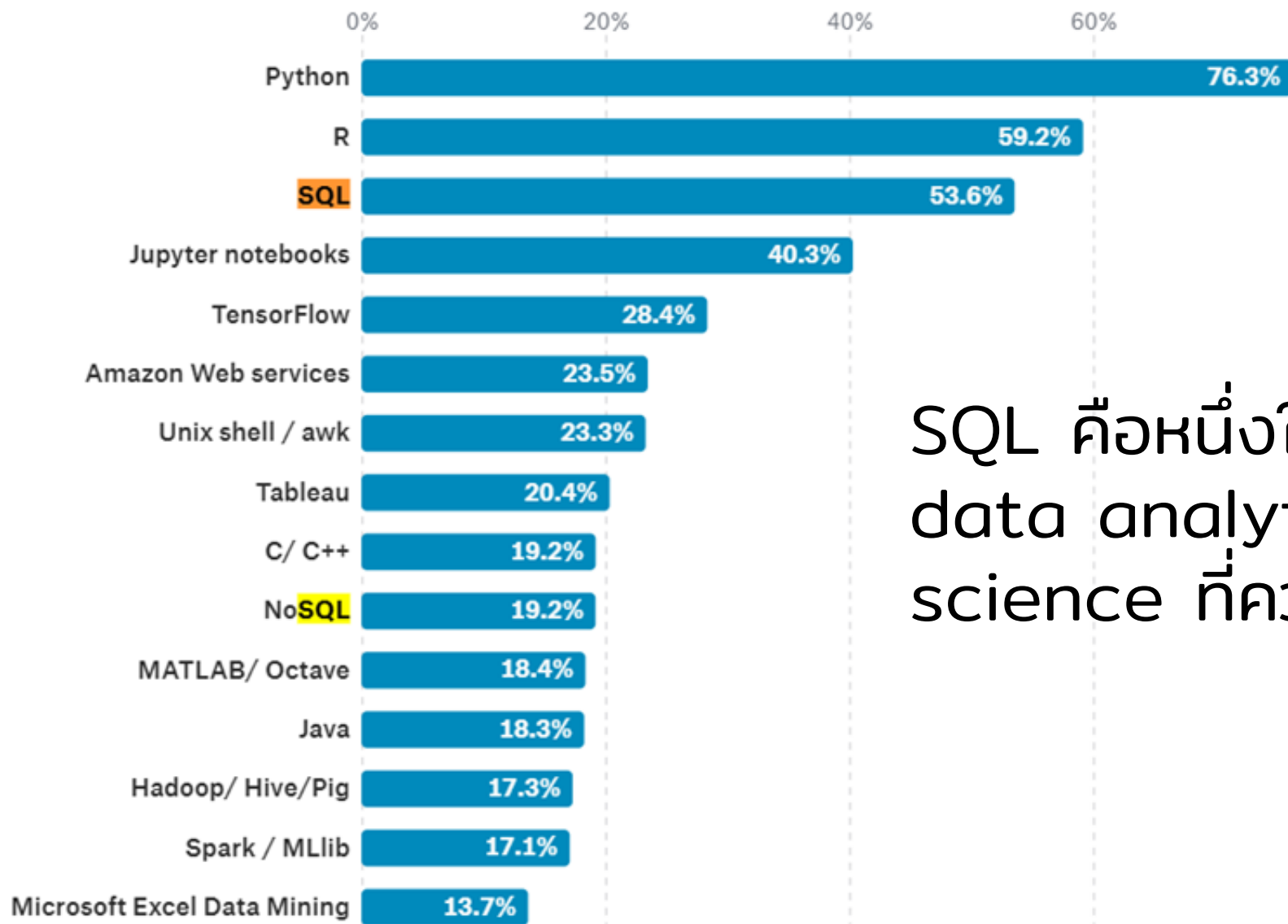
Audio ON



Video ON



Full Screen



SQL คือหนึ่งในสามภาษา  
data analytics/ data  
science ที่ควรรู้จัก

7,955 responses

Only displaying the top 15 answers. There are 38 answers not shown.

# What is SQL?

## Structured Query Language


ใช้กันมาจะ 50 ปีแล้ว



กิป – เวลาคำนวณเปลี่ยนฐานเงินเดือน USA เป็น Thai ใช้  
สูตร  $USD * 30 / 2.5$  โดยประมาณ

## SQL Data Analyst Salaries

18,600 Salaries Updated Oct 24, 2019

 Very High  
Confidence

Industries



Company Sizes



Years of Experience



Average Base Pay

**\$69,291** /yr

Not enough reports to show salary distribution



Additional Cash Compensation ?

Average \$4,772

Range \$1,119 - \$13,837

How much does a SQL Data Analyst make?

The national average salary for a SQL Data Analyst is \$69,291 in United States. Filter by location to see... [More](#)

Salaries for Related Job Titles

It-pl SQL Developer/Data Anal... \$83K

SQL Analyst \$64K

Business Data Analyst \$31

Data Mining Analyst \$77K

SQL Server Developer \$91K

[https://www.glassdoor.com/Salaries/sql-data-analyst-salary-SRCH\\_KO0,16.htm](https://www.glassdoor.com/Salaries/sql-data-analyst-salary-SRCH_KO0,16.htm)

# que·ry

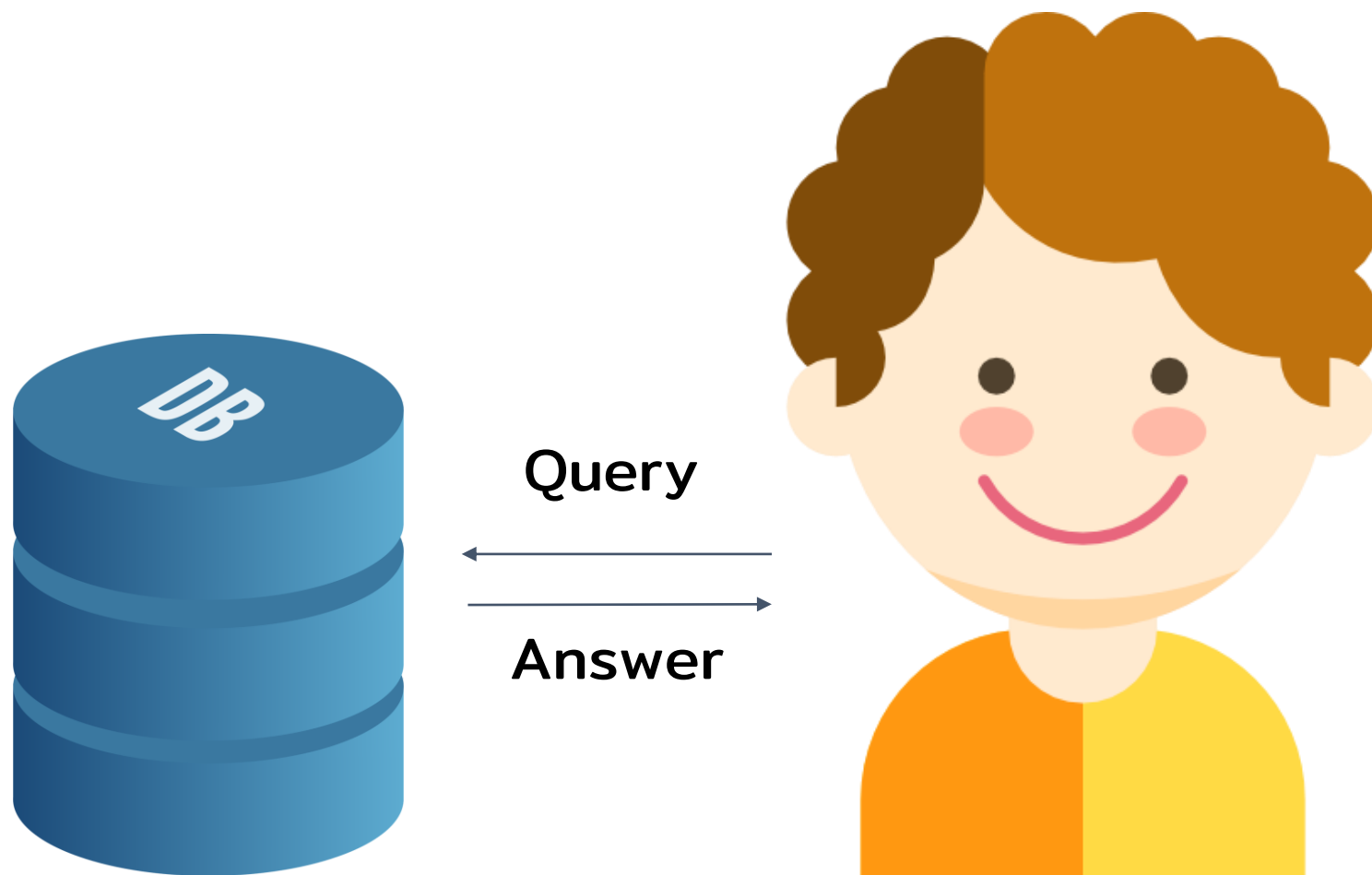
/ˈkwɪrē/ 

*noun*

1. a question, especially one addressed to an official or organization.  
"a spokeswoman said queries could not be answered until Monday"  
*synonyms:* [question](#), [inquiry](#) [More](#)

*verb*

1. ask a question about something, especially in order to express one's doubts about it or to check its validity or accuracy.  
"many people queried whether any harm had been done"  
*synonyms:* [ask](#), [inquire](#), [question](#) [More](#)

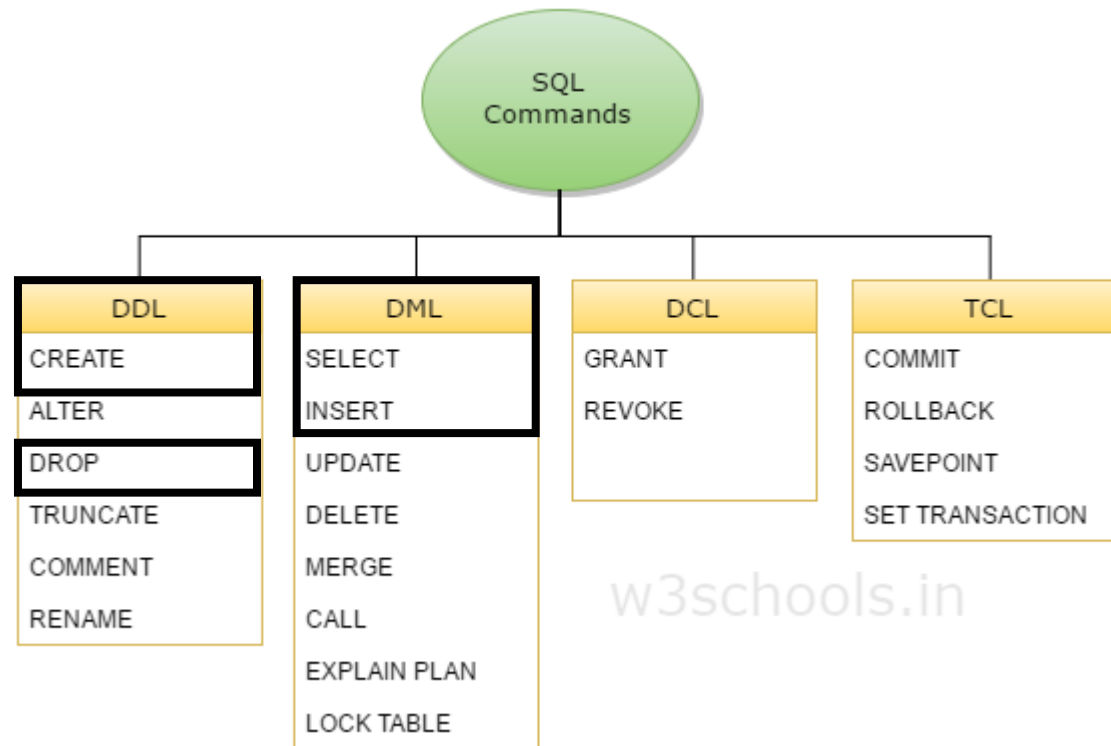


การเขียน SQL คือการส่งคำถามไปที่ database





# Types of SQL

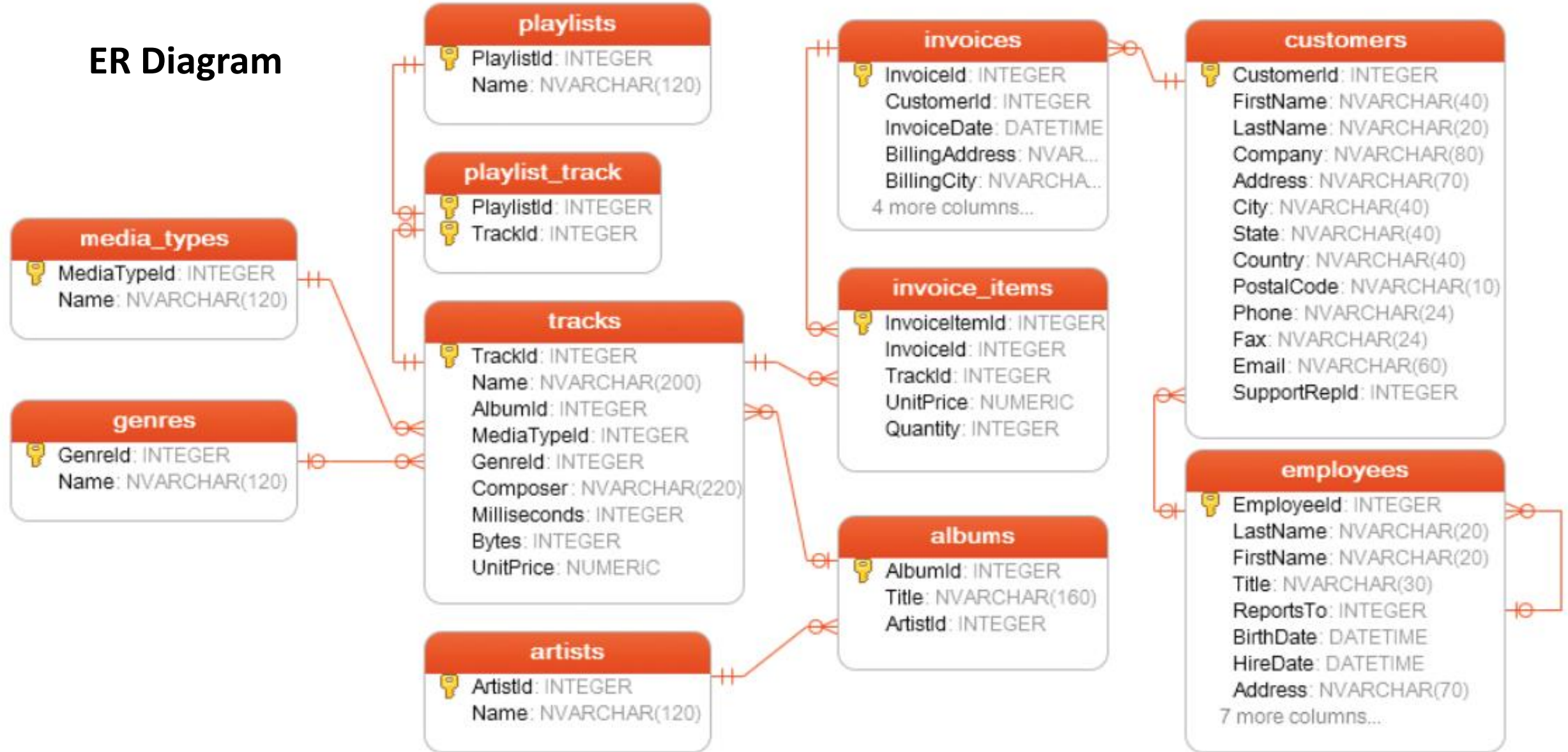


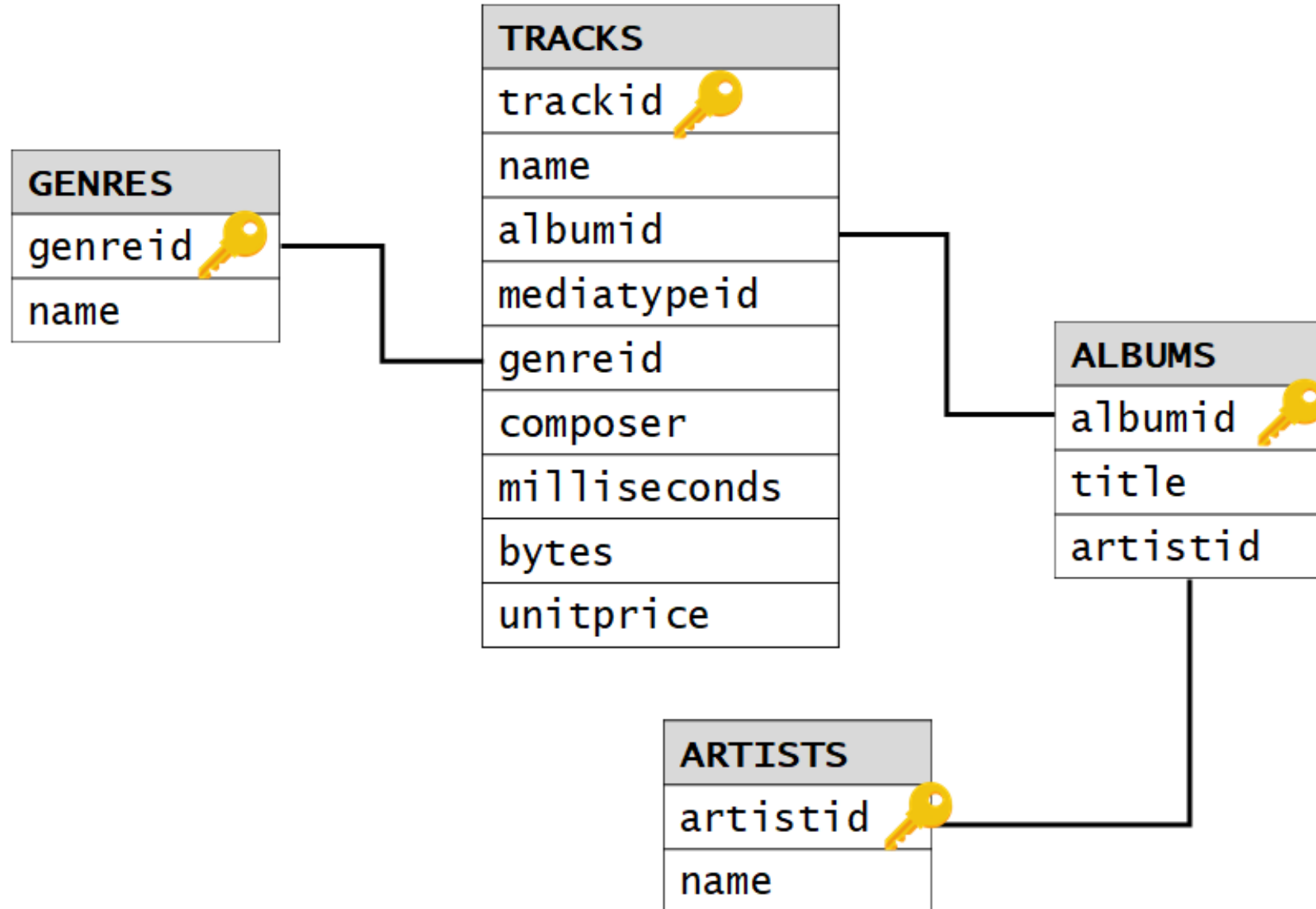
SQL commands for data analyst

- SELECT (90-95%)
  - CREATE
  - INSERT
  - DROP
- } 5-10%



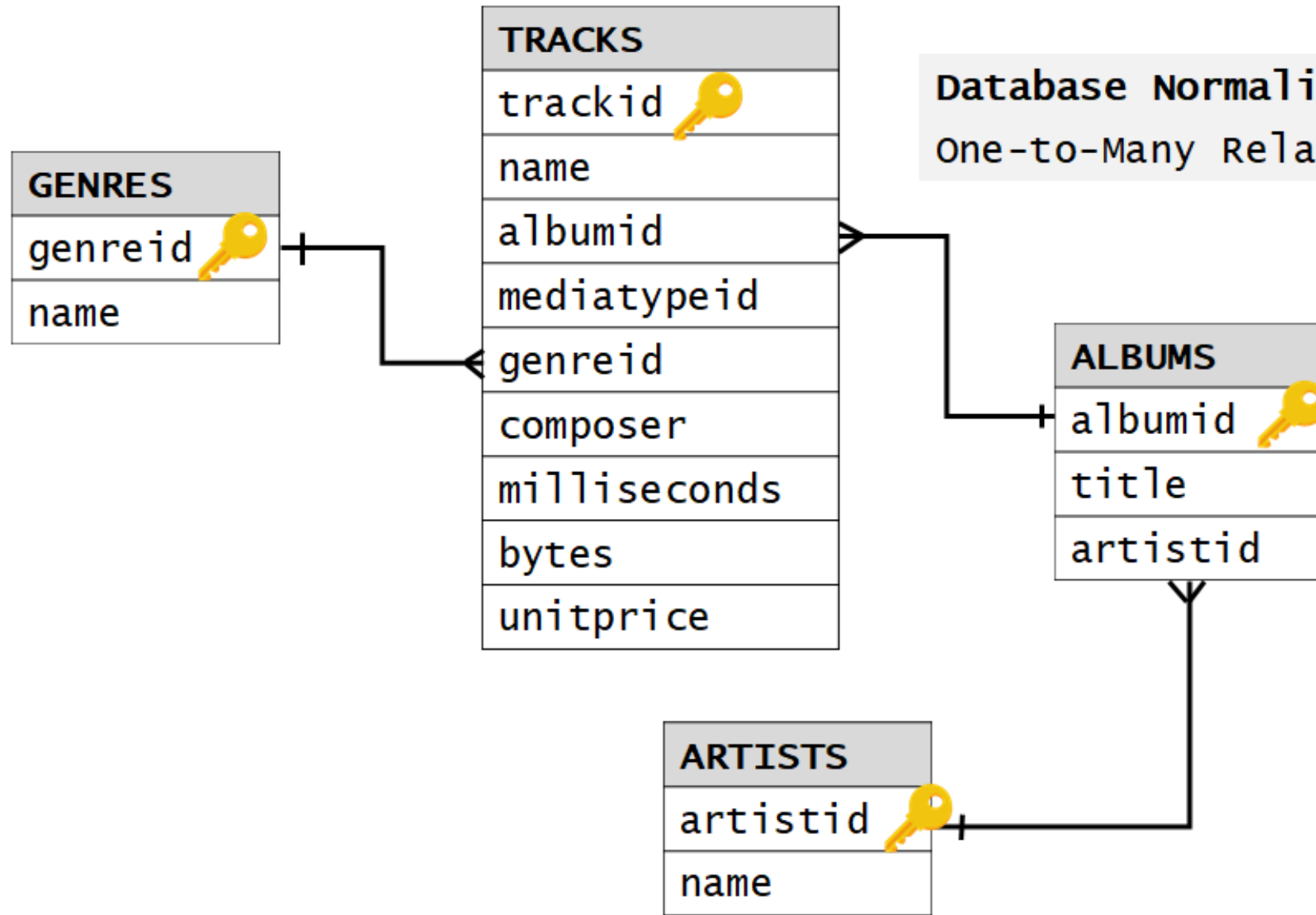
# ER Diagram





 Primary Key (Unique)

Database Normalization  
One-to-Many Relationship



 Primary Key (Unique)



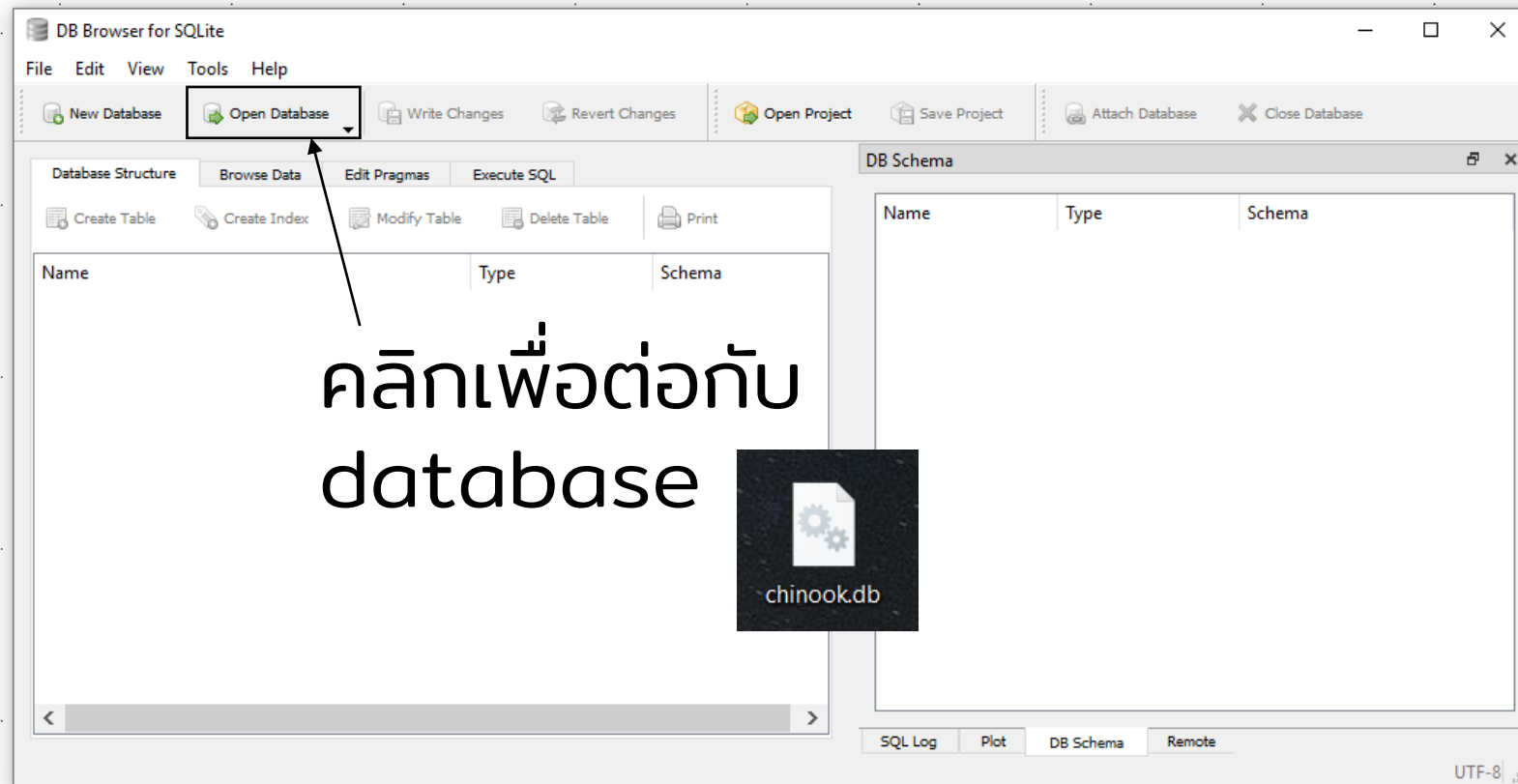
# Connect to Database

สร้างการเชื่อมต่อไปที่ฐานข้อมูล

# DB Browser

<https://sqlitebrowser.org/>

ดาวน์โหลดโปรแกรมฟรี





Name	Type	Schema
Tables (13)		
albums	CREATE TABLE "albums" ( [AlbumId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Title] NVARCHAR(160) NOT NULL )	
artists	CREATE TABLE "artists" ( [ArtistId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Name] NVARCHAR(120) )	
customers	CREATE TABLE "customers" ( [CustomerId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [FirstName] NVARCHAR(40) NOT NULL, [LastName] NVARCHAR(40) NOT NULL, [Address] NVARCHAR(70) NOT NULL, [City] NVARCHAR(40) NOT NULL, [State] NVARCHAR(40) NOT NULL, [Country] NVARCHAR(40) NOT NULL, [SupportRepId] INTEGER NOT NULL )	
employees	CREATE TABLE "employees" ( [EmployeeId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [LastName] NVARCHAR(40) NOT NULL, [FirstName] NVARCHAR(40) NOT NULL, [Title] NVARCHAR(80) NOT NULL, [ReportsTo] INTEGER NOT NULL, [BirthDate] DATETIME NOT NULL, [Address] NVARCHAR(70) NOT NULL, [City] NVARCHAR(40) NOT NULL, [State] NVARCHAR(40) NOT NULL, [Country] NVARCHAR(40) NOT NULL, [SupportRepId] INTEGER NOT NULL )	
genres	CREATE TABLE "genres" ( [GenreId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Name] NVARCHAR(120) )	
invoice_items	CREATE TABLE "invoice_items" ( [InvoiceLineId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [InvoiceId] INTEGER NOT NULL, [TrackId] INTEGER NOT NULL, [UnitPrice] NUMERIC(10,2) NOT NULL, [Quantity] INTEGER NOT NULL )	
invoices	CREATE TABLE "invoices" ( [InvoiceId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [CustomerId] INTEGER NOT NULL, [InvoiceDate] DATETIME NOT NULL, [BillingAddress] NVARCHAR(70) NOT NULL, [BillingCity] NVARCHAR(40) NOT NULL, [BillingState] NVARCHAR(40) NOT NULL, [BillingCountry] NVARCHAR(40) NOT NULL, [SupportRepId] INTEGER NOT NULL )	
media_types	CREATE TABLE "media_types" ( [MediaTypeId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Name] NVARCHAR(80) NOT NULL )	
playlist_track	CREATE TABLE "playlist_track" ( [PlaylistId] INTEGER NOT NULL, [TrackId] INTEGER NOT NULL, CONSTRAINT [PK_PlaylistTrack] PRIMARY KEY ([PlaylistId], [TrackId]) )	
playlists	CREATE TABLE "playlists" ( [PlaylistId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Name] NVARCHAR(120) )	
sqlite_sequence	CREATE TABLE sqlite_sequence(name,seq)	
sqlite_stat1	CREATE TABLE sqlite_stat1(tbl,idx,stat)	
tracks	CREATE TABLE "tracks" ( [TrackId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [Name] NVARCHAR(200) NOT NULL, [AlbumId] INTEGER NOT NULL, [GenreId] INTEGER NOT NULL, [Composer] NVARCHAR(100) NOT NULL, [Lyricist] NVARCHAR(100) NOT NULL, [ComposerId] INTEGER NOT NULL, [LyricistId] INTEGER NOT NULL, [Millisecond] INTEGER NOT NULL, [Bytes] INTEGER NOT NULL )	

## Indices (10)

IFK_AlbumArtistId	CREATE INDEX [IFK_AlbumArtistId] ON "albums" ([AlbumId])
IFK_CustomerSupportRepId	CREATE INDEX [IFK_CustomerSupportRepId] ON "customers" ([SupportRepId])
IFK_EmployeeReportsTo	CREATE INDEX [IFK_EmployeeReportsTo] ON "employees" ([ReportsTo])
IFK_InvoiceCustomerId	CREATE INDEX [IFK_InvoiceCustomerId] ON "invoices" ([CustomerId])
IFK_InvoiceLineInvoiceId	CREATE INDEX [IFK_InvoiceLineInvoiceId] ON "invoice_items" ([InvoiceId])
IFK_InvoiceLineTrackId	CREATE INDEX [IFK_InvoiceLineTrackId] ON "invoice_items" ([TrackId])
IFK_PlaylistTrackTrackId	CREATE INDEX [IFK_PlaylistTrackTrackId] ON "playlist_track" ([TrackId])
IFK_TrackAlbumId	CREATE INDEX [IFK_TrackAlbumId] ON "tracks" ([AlbumId])
IFK_TrackGenreId	CREATE INDEX [IFK_TrackGenreId] ON "tracks" ([GenreId])
IFK_TrackMediaTypeId	CREATE INDEX [IFK_TrackMediaTypeId] ON "tracks" ([MediaTypeId])

## DB Schema

Name	Type	Schema
Tables (13)		
albums	CREATE TAB	
artists	CREATE TAB	
customers	CREATE TAB	
employees	CREATE TAB	
genres	CREATE TAB	
invoice_items	CREATE TAB	
invoices	CREATE TAB	
media_types	CREATE TAB	
playlist_track	CREATE TAB	
playlists	CREATE TAB	
sqlite_sequence	CREATE TAB	
sqlite_stat1	CREATE TAB	
tracks	CREATE TAB	
Indices (10)		
IFK_AlbumArtistId	CREATE INDI	
IFK_CustomerSupportRepId	CREATE INDI	
IFK_EmployeeReportsTo	CREATE INDI	
IFK_InvoiceCustomerId	CREATE INDI	
IFK_InvoiceLineInvoiceId	CREATE INDI	
IFK_InvoiceLineTrackId	CREATE INDI	
IFK_PlaylistTrackTrackId	CREATE INDI	
IFK_TrackAlbumId	CREATE INDI	
IFK_TrackGenreId	CREATE INDI	
IFK_TrackMediaTypeId	CREATE INDI	
Views (0)		



- SQL เป็นภาษาแบบ case insensitive (ตัวพิมพ์เล็กใหญ่ ไม่มีผล)
- ปกติเราพิมพ์ SQL clause เป็นตัวพิมพ์ใหญ่ และชื่อคอลัมน์และตารางเป็นตัวพิมพ์เล็ก



# Foundation 1

เรียนวิธีการเขียน query เบื้องต้น

Command	What it does
SELECT	เลือกคอลัมน์จาก table
WHERE	ฟิเตอร์ rows
ORDER BY	เรียงลำดับข้อมูล

```
SELECT * FROM table;
```

จงเลือกทุกคอลัมน์จาก table นี้

```
SELECT  
    firstname,  
    lastname  
FROM customers;
```

จงเลือกคอลัมน์ firstname และ lastname จาก table customers นี้

# QUIZ

1. จงเลือกทุกคอลัมน์ของ **table invoices**
2. **table invoices** มีกี่ columns มีกี่ rows?
3. จงเลือกคอลัมน์ albumid, name, milliseconds, bytes จาก **table tracks**







ทำไมมันง่ายยังจ๊ะครับ 55+

เราใช้ AS (Alias)  
เพื่อเปลี่ยนชื่อคอลัมน์

```
SELECT  
  firstname AS fn,  
  lastname AS ln  
FROM customers;
```

เปลี่ยนชื่อคอลัมน์ firstname เป็น fn และ lastname เป็น ln

**SELECT**

firstname **AS** fn,

lastname **AS** ln

**FROM** customers

**LIMIT** 10;

ดึงข้อมูลออกมาแค่ 10 แถวแรกเท่านั้น

LIMIT ใช้เพื่อกำหนด  
จำนวนแถวที่ต้องการ



**SELECT** 10 + 2;

**SELECT** 10 - 4;

**SELECT** 10 \* 5;

**SELECT** 10 / 3;

เปลี่ยน milliseconds เป็น minutes  
และเปลี่ยน bytes เป็น megabytes

**SELECT**

milliseconds,

milliseconds/60000 **AS** mins,

bytes,

bytes/ (1024\*1024) **AS** mb

**FROM** tracks;

สร้างคอลัมน์ใหม่ชื่อว่า mins และ mb

Command	What it does
SELECT	เลือกคอลัมน์จาก table
WHERE	ฟิเตอร์ rows
ORDER BY	เรียงลำดับข้อมูล



**SELECT**

firstname,

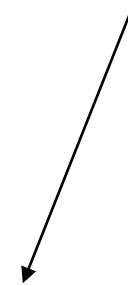
lastname,

country

**FROM** customers

**WHERE** country = 'USA';

ใน SQL เราใช้ single quote เวลา  
พิมพ์คอลลัมน์ที่เป็น text/ character



```
SELECT * FROM customers  
WHERE country = 'USA'  
      AND state = 'CA';
```

```
SELECT * FROM customers  
WHERE country = 'USA'  
      OR country = 'France';
```

# QUIZ

จงเลือกลูกค้าทุกคนที่อาศัยในประเทศ **USA, Brazil,**  
**France** จาก table customers



```
SELECT * FROM customers  
WHERE country IN ('USA', 'Brazil', 'France');
```

```
SELECT * FROM invoices  
WHERE invoiceDate BETWEEN '2009-01-01' AND '2009-01-31';
```

ท๊อป - เราใช้ BETWEEN AND เพื่อฟ้ลเตอร์วันที่ได้ด้วย

**LIKE + WILDCARD**

```
SELECT * FROM customers  
WHERE country LIKE 'U%';
```

เลือกชื่อประเทศที่ชื่อขึ้นต้นด้วยตัว U

# QUIZ

1. จงเลือกลูกค้าทุกคนที่ firstname เริ่มต้นด้วยตัว L
2. จงเลือกลูกค้าทุกคนที่ใช้อีเมล @gmail.com



Command	What it does
SELECT	เลือกคอลัมน์จาก table
WHERE	ฟิเตอร์ rows
ORDER BY	เรียงลำดับข้อมูล

```
SELECT customerid, invoicedate, total  
FROM invoices  
ORDER BY total;
```

```
SELECT customerid, invoicedate, total  
FROM invoices  
ORDER BY total DESC;
```

เรียงข้อมูลจากมากไปน้อย  
descending order ด้วย DESC







## Foundation 2

เรียนฟังชั้นพื้นฐานของ SQL

There are **2 types** of  
function in SQL

1. Aggregate Functions
2. Value Functions

# Aggregate Functions

ใช้สำหรับคำนวณค่าสถิติ  
เบื้องต้นของคอลัมน์ที่เราเลือก

**SELECT**

**AVG**(total),  
**SUM**(total),  
**MAX**(total),  
**MIN**(total),  
**COUNT**(total)

มีอยู่ 5 aggregate  
functions ที่เราใช้บ่อยๆ :D

**FROM**

invoices;

**SELECT**

**AVG**(total) **AS** mean\_invoice,

**SUM**(total) **AS** sum\_invoice,

**MAX**(total) **AS** max\_invoice,

**MIN**(total) **AS** min\_invoice,

**COUNT**(total) **AS** n\_invoice

**FROM**

invoices;



# Value Functions

ใช้ปรับค่า value  
ในคอลัมน์นั้นๆ

**SELECT**

**ROUND**(**AVG**(total), 2)

**FROM**

invoices;

ROUND(value, 2)  
ปรับค่าทศนิยม 2  
ตำแหน่ง





**NULL** เรียกว่าค่า missing value ก็ได้

```
SELECT * FROM customers  
WHERE state IS NULL;
```

# COALESCE ( )

**SELECT**

state,

**COALESCE**(state, 'BKK')

**FROM** customers;

# QUIZ

1. เพลงที่มีขนาดใหญ่ที่สุดใน **table tracks** มีขนาดกี่ MB?
2. เพลงที่มีขนาดเล็กที่สุดใน **table tracks** มีขนาดกี่ MB?
3. จงเปลี่ยนชื่อลูกค้า `firstname` ใน **table customers** ให้เป็นตัวพิมพ์ใหญ่ทั้งหมด





# Intermediate SQL

เรียนการจับและสรุปผลตามกลุ่มข้อมูล

Verbs	Meaning
GROUP BY	จับกลุ่มข้อมูล เช่น prepaid / postpaid
HAVING	เลือก group (กลุ่มที่เราต้องการ)

**SELECT**

country,

**COUNT**(\*)

**FROM**

customers

**GROUP BY**

country;

ทิป – คอลัมน์ที่อยู่ใน GROUP BY  
ต้องอยู่ใน SELECT clause เสมอ

**SELECT**

country,

**COUNT**(\*) **AS** n

**FROM**

customers

**GROUP BY**

Country

**HAVING**

n > 10;



ใส่เงื่อนไขเพื่อ filter กลุ่มที่เราต้องการ

# QUIZ

จงหาประเทศ TOP 5 ที่มีลูกค้าเราอยู่มากที่สุดจาก **table**  
**customers**





# CASE

เขียนเงื่อนไขใน SQL เช่น high, med, low ARPU

**SELECT**

**CASE**

**WHEN** ..... **THEN** .....

**WHEN** ..... **THEN** .....

**ELSE** .....

**END**

**FROM**

table;

**SELECT**

**CASE**

**WHEN** total >= 20 **THEN** 'High'

**WHEN** total >= 10 **THEN** 'Medium'

**ELSE** 'Low'

**END**

**FROM**

table;

# QUIZ

จงเลือกคอลัมน์ country และสร้างคอลัมน์ใหม่ด้วย CASE ..

WHEN โดยที่ประเทศ USA ให้แทนค่าเป็น 1 และประเทศอื่นๆ ให้แทนค่าเป็น 0 ดึงข้อมูลจาก **table customers**

USA	1
Brazil	0
USA	1
France	0





# Relational Databases

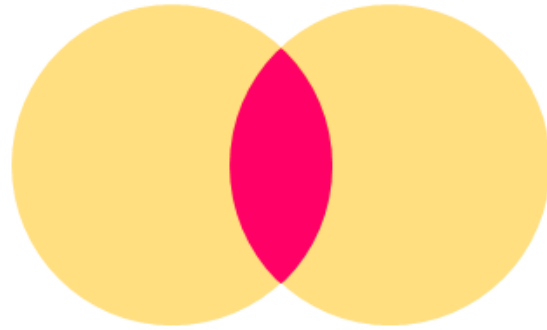
เรียนการเขียน query เพื่อดึงข้อมูลจากหลายๆตาราง

A man with a beard and short dark hair is shown from the chest up, screaming with his mouth wide open. He is wearing a white t-shirt under a dark red button-down shirt. He is standing in the rain, with water droplets visible in the air around him. The background is a dark, textured wall.

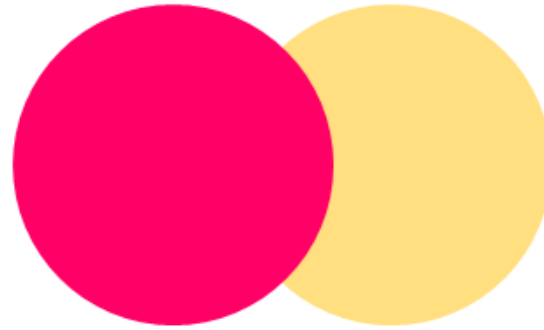
NOW FEEL THE POWER

# Four Types of JOINS

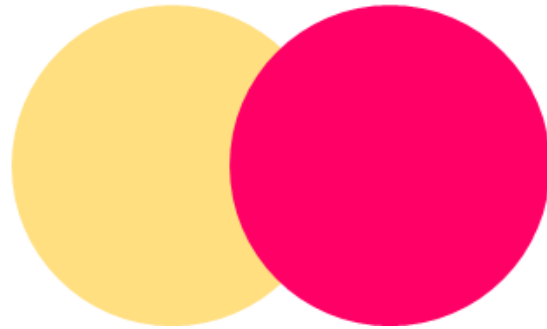
INNER JOIN



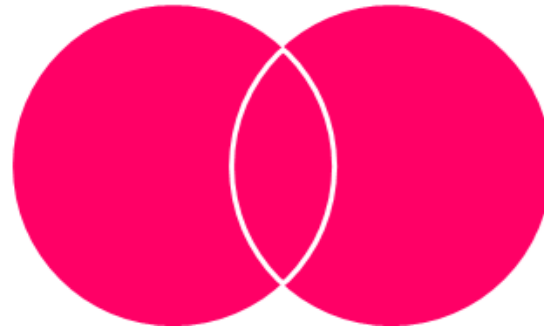
LEFT JOIN

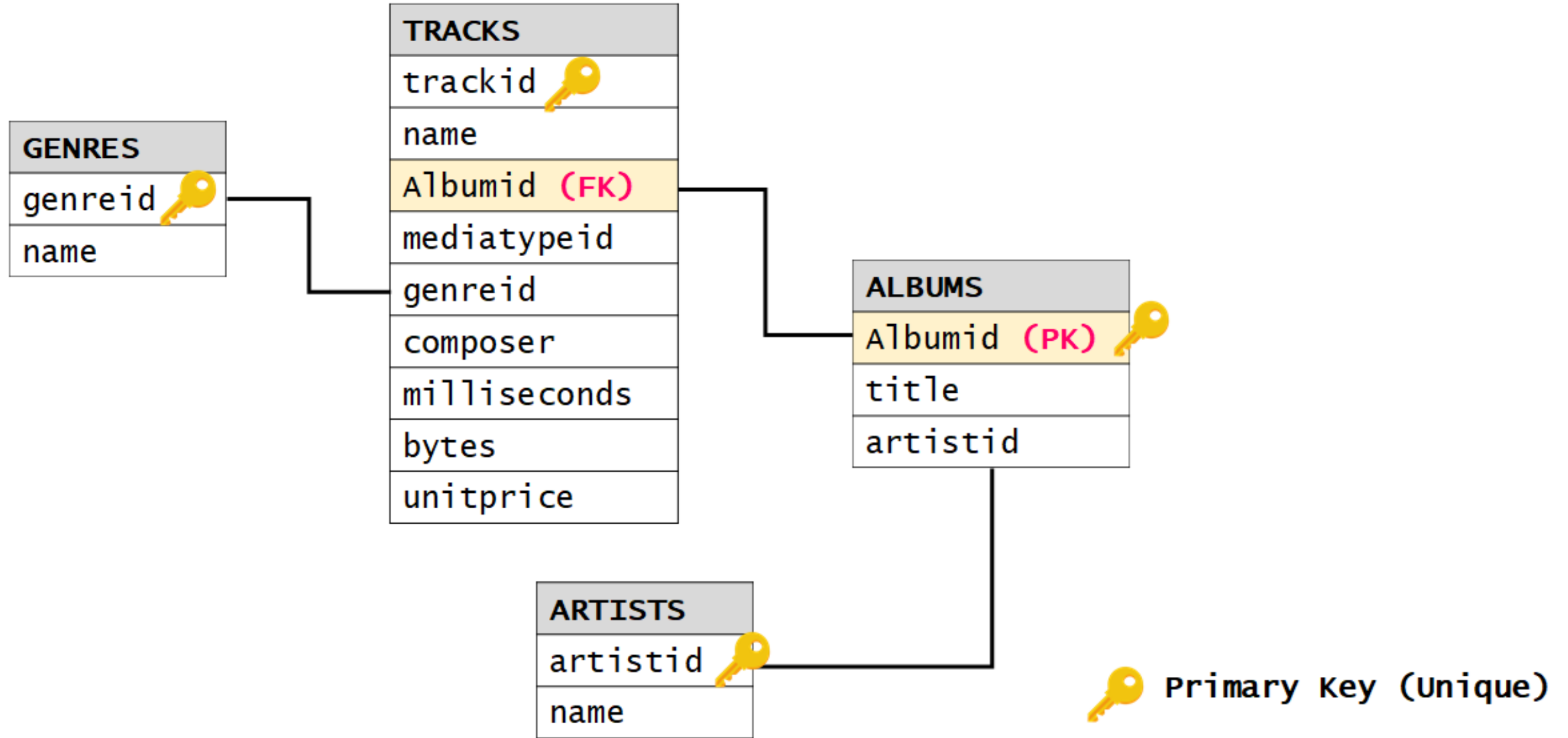


RIGHT JOIN

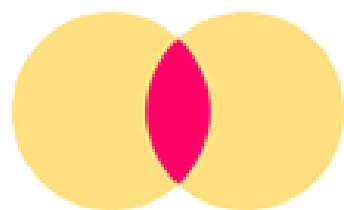


FULL OUTER JOIN









# INNER JOIN

Customer

ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

Age

ID	Age
1	29
2	30
4	18
6	25
7	26



Result

ID	Name	Age
1	Toy	29
2	Hello	30
4	SQL	18

ผลลัพธ์ออกมาเฉพาะ ROW ที่ matched กันได้ของสองตารางเท่านั้น

# SELECT

A.column,

B.column

เราใช้ syntax นี้เวลาเขียน join หลายๆ ตาราง table.column

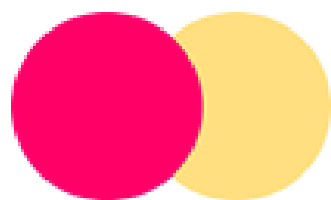
FROM tableA AS A

INNER JOIN tableB AS B

เราใช้ AS เพื่อตั้งชื่อตารางสั้นๆ

ON A.id = B.id

ไล่นี้นี้คือเงื่อนไขการ join: Primary key = Foreign key



# LEFT JOIN

ตารางซ้ายมือยังอยู่เหมือนเดิม แต่จะ  
เชื่อมตารางขวาใน row ที่ matched

**Customer**

ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

**Age**

ID	Age
1	29
2	30
4	18
6	25
7	26



**Result**

ID	Name	Age
1	Toy	29
2	Hello	30
3	World	<i>NULL</i>
4	SQL	18
5	Awesome	<i>NULL</i>

**SELECT**

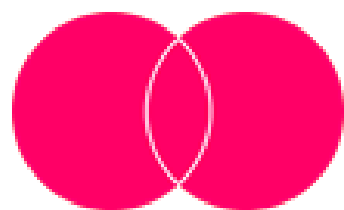
**A**.column,

**B**.column

**FROM** tableA **AS** **A**

**LEFT JOIN** tableB **AS** **B**

**ON** **A**.id = **B**.id



# FULL OUTER JOIN

Customer

ID	Name
1	Toy
2	Hello
3	World
4	SQL
5	Awesome

Age

ID	Age
1	29
2	30
4	18
6	25
7	26



Result

ID	Name	Age
1	Toy	29
2	Hello	30
3	World	<i>NULL</i>
4	SQL	18
5	Awesome	<i>NULL</i>
6	<i>NULL</i>	25
7	<i>NULL</i>	26

**SELECT**

**A**.column,

**B**.column

**FROM** tableA **AS** **A**

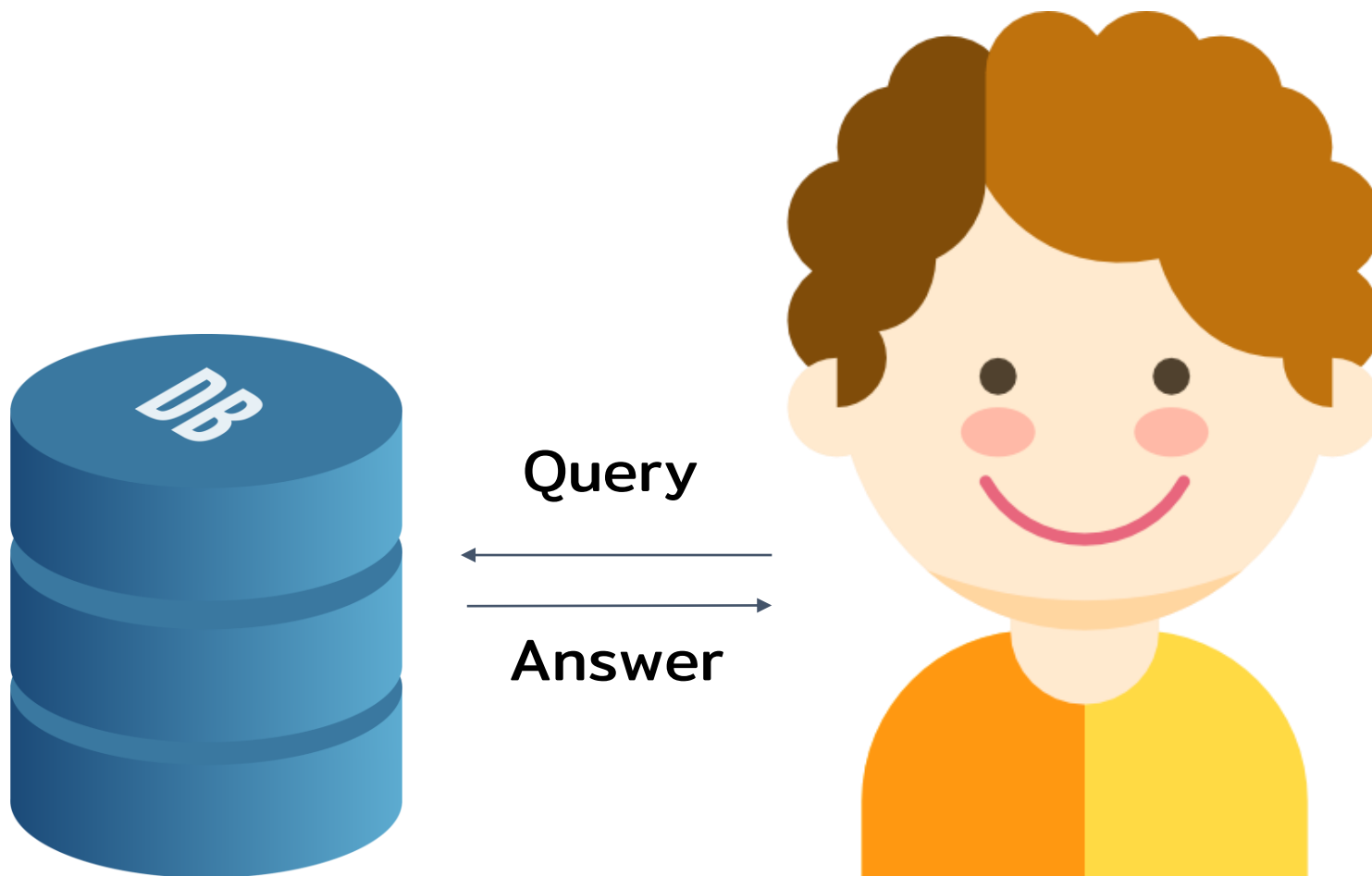
**FULL JOIN** tableB **AS** **B**

**ON** **A**.id = **B**.id



## Course Summary

สรุปเนื้อหาที่เราเรียนกันวันนี้





**SELECT** columns  
**FROM** table  
**(JOIN)**  
**WHERE** conditions  
**GROUP BY** columns  
**HAVING** conditions  
**ORDER BY** columns  
**LIMIT** n\_rows;

# Structured

SQL clause ต้องเขียน  
เรียงตามลำดับนี้เสมอ



Facebook: DataRockie

Website: <https://datarockie.com>