SQL - Exercise

Deal with Date

```
SELECT
  CAST(STRFTIME('%Y%m',invoicedate) AS INT) AS monthid

 FROM invoices

 WHERE monthid = 200909
```

```
SELECT
 firstname,
    country,
    email
FROM customers
WHERE UPPER(country) = 'FRANCE' AND
      email like '%@yahoo%';  --pattern matching
```

```
SELECT
  firstname,
    country,
    email
FROM customers
WHERE customerid BETWEEN 1 AND 5;
```

```
SELECT
 firstname,
    country,
    email
```

```
FROM customers
WHERE customerid BETWEEN 1 AND 5;
```

```
SELECT
  STRFTIME('%Y',invoicedate) AS year,
    STRFTIME('%m',invoicedate) AS month,
    STRFTIME('%d',invoicedate) AS day
 FROM invoices;

 --postgreSQL SELECT EXTRACT ('year' from invoicedate)
```

```
SELECT
  STRFTIME('%Y%m',invoicedate) AS monthid

 FROM invoices

 WHERE monthid = '202212'
```

```
SELECT
  CAST(STRFTIME('%Y%m',invoicedate) AS INT) AS monthid

 FROM invoices

 WHERE monthid = 200909

CASt คำสั่งเปลี่ยน datatypr
```

```
SELECT * FROM artists
JOIN albums
ON artists.artistid = albums.artistid
JOIN tracks
ON albums.albumid = tracks.albumid


WHERE t2.Title LIKE 'The%' AND t3.composer LIKE 'M%';
```

```
--aggregrate
SELECT COUNT(*) AS no_tracks , AVG(tr.bytes) AS avg_track, SUM(tr.bytes) as sum_bytes
FROM artists AS ar
JOIN albums AS al
ON ar.artistid = al.artistid
JOIN tracks AS tr
ON al.albumid = tr.albumid
JOIN genres AS ge ON tr.genreid = ge.genreid
```

```
--aggregrate column ที่ group by ต้อง select ด้วย
--aggregrate
SELECT
  ge.name,
    COUNT(*) AS no_tracks ,
  AVG(tr.bytes) AS avg_track,
    SUM(tr.bytes) as sum_bytes

FROM artists AS ar
JOIN albums AS al
ON ar.artistid = al.artistid
JOIN tracks AS tr
ON al.albumid = tr.albumid
JOIN genres AS ge ON tr.genreid = ge.genreid
WHERE al.title NOT LIKE 'The %'
GROUP BY ge.name
HAVING ge.name LIKE 'S%' --HAVing filter group / where fit table--
ORDER BY no_tracks DESC
```

```
--sub query
-- common table ex
WITH sub AS (
SELECT
  ar.name AS artist,
    al.title,
    tr.name AS track,
    tr.bytes,
    ge.Name AS genre
FROM artists AS ar
JOIN albums AS al
ON ar.artistid = al.artistid
JOIN tracks AS tr
ON al.albumid = tr.albumid
JOIN genres AS ge ON tr.genreid = ge.genreid
)
```

```
SELECT track, bytes, genre FROM sub
WHERE genre = "Rock" AND artist LIKE 'D%'
```

With Clause

```
WITH usa_customer AS (
  SELECT * FROM customers
  WHERE country = 'USA'
), invoice_2010 AS (
  SELECT * FROM invoices
  WHERE STRFTIME("%Y", invoicedate) ="2010"
)
SELECT firstname,SUM(i.total)
FROM usa_customer AS c
JOIN invoice_2010 AS i ON c.customerid = i.customerid
GROUP BY 1;
```

Window Function

```
--window founctionalbums
SELECT * FROM (
SELECT
  ROW_NUMBER() OVER(PARTITION BY country ORDER BY firstname) AS rowNum,
  firstname, country FROM customers
)
WHERE rowNum =1;
```

```
-- window function NTIlE()
SELECT *,
  CASE WHEN segment IN (1,2) THEN 'Low'
       WHEN segment IN (4,5) THEN 'High'
```

```
        ELSE 'Medium'
        END label_segment



FROM(SELECT segment, COUNT(*) AS n FROM (
SELECT
  name AS sing_name,
    milliseconds,
    NTILE(5) OVER(ORDER by milliseconds) AS segment
FROM tracks
)

GROUP BY segment
)
```
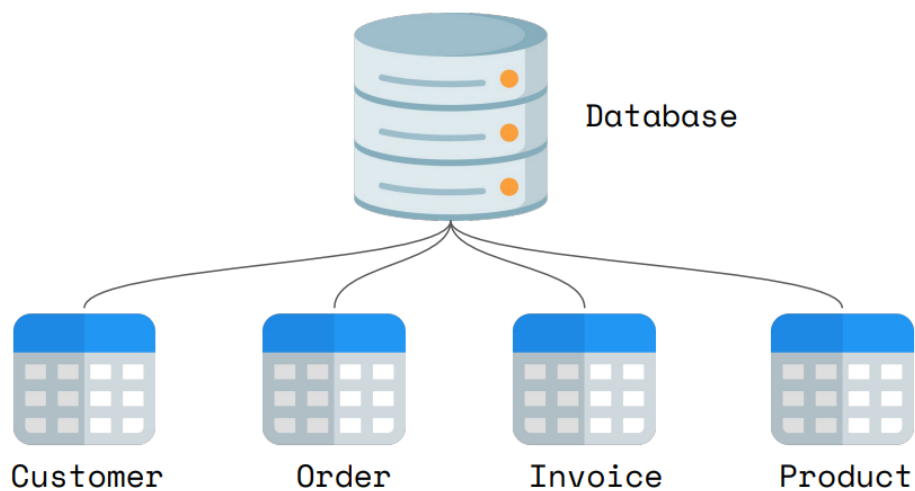
```
SELECT
    firstname, country,
      CASE WHEN country IN ('USA', 'Canada') THEN 'America'
      WHEN country in ('belgium', 'Italy', 'France') THEN 'Europe'
      ELSE 'Other Regions'
      END As region
 FROM customers
```
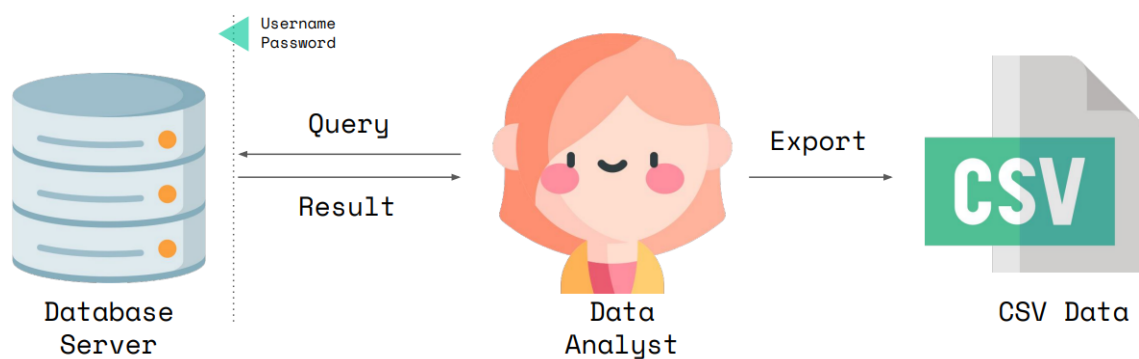
Content

- What's Database?

You should understand "Business" first. Then you can work so far.

Data Analyst Workflow

How can I access Database?
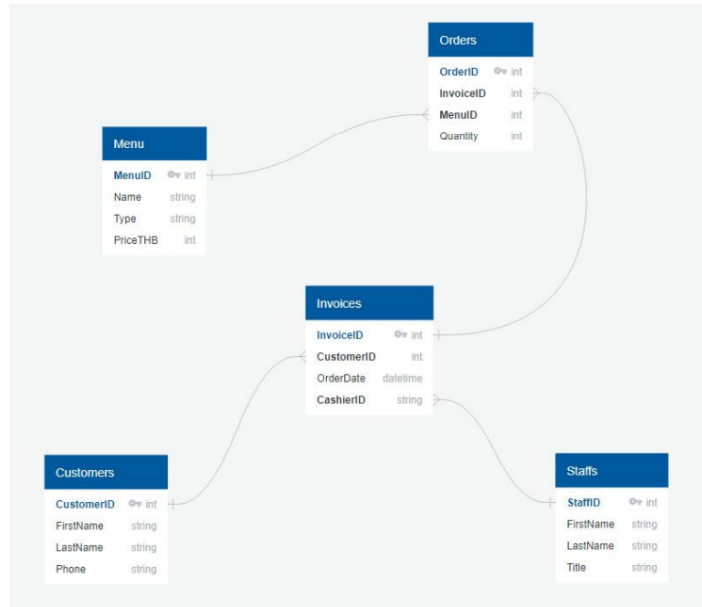
What's Database that company use?

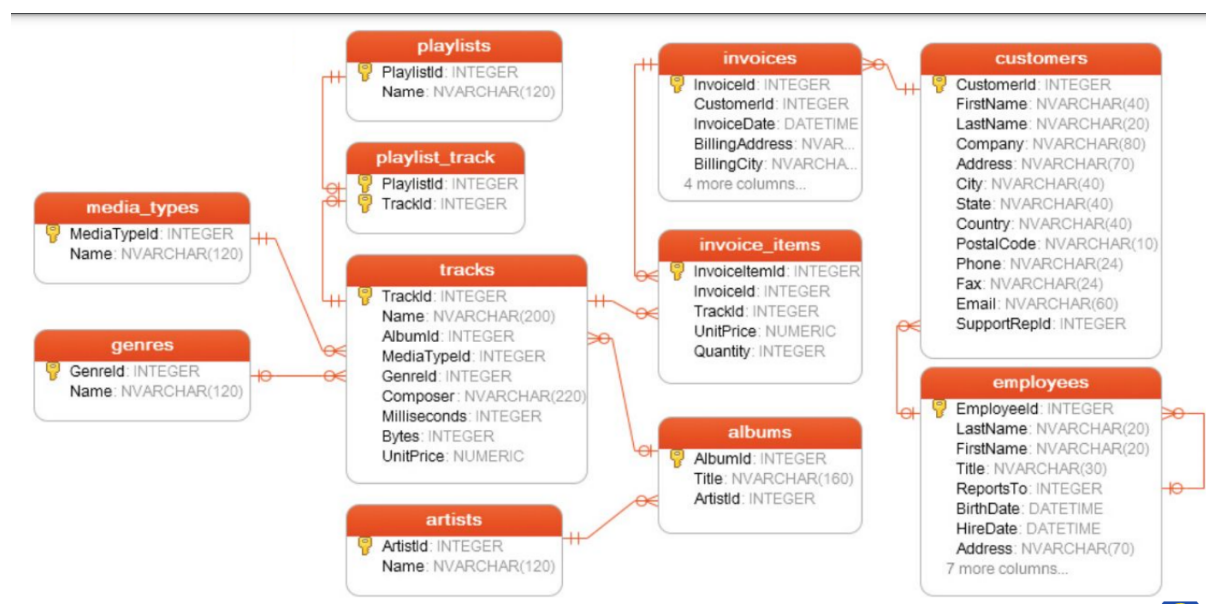Do they have ER diagram?

- **ER Diagram**

Mapping table

ER Diagram

Entity Relationship Diagram

Each table has "Primary Key" (It's unique) to join other table that call "Foreign Key" in other table.

many to many relationship

must have "bridge table"

- CRUD (Create, Retrieve, Update, Delete)

CRUD (Create, Retrieve, Update, Delete)

Concat by ||

```
SELECT
  firstname || " " ||lastname AS fullname,
    'Data Rockie' AS tag
FROM customers
```

fullname	tag
Luís Gonçalves	Data Rockie
Leonie Köhler	Data Rockie
François Tremblay	Data Rockie
Bjørn Hansen	Data Rockie
František Wichterlová	Data Rockie

Divide

If divide by int result comes int, divide by decimal (6000.0) result come decimal

```
SELECT
  name,
```

```
    ROUND(milliseconds/60000.0,2) AS minute,
    ROUND(bytes/(1024*1024.0),4) AS mb
FROM tracks
LIMIT 10;
```

Name	minute	mb
For Those About To Rock (We Salute You)	5.73	10.6529
Balls to the Wall	5.71	5.2551
Fast As a Shark	3.84	3.8061
Restless and Wild	4.2	4.1311
Princess of the Dawn	6.26	5.9991

```
SELECT
  firstname,
    country,
    email
FROM customers
WHERE UPPER(country) = 'FRANCE'
  AND firstname = 'Marc';
```

FirstName	Country	Email
Marc	France	marc.dubois@hotmail.com

```
-- commemt pattern matching
SELECT
  firstname,
    country,
    email
FROM customers
WHERE UPPER(country) = 'FRANCE'
  AND email LIKE '%@yahoo%';
```

FirstName	Country	Email
Camille	France	camille.bernard@yahoo.fr
Wyatt	France	wyatt.girard@yahoo.fr

```
-- commemt
SELECT
  firstname,
    country,
    email
FROM customers
WHERE country in ( 'France','USA','United Kingdom','Italy');
```

FirstName	Country	Email
Frank	USA	fharris@google.com
Jack	USA	jacksmith@microsoft.com
Michelle	USA	michelleb@aol.com
Tim	USA	tgoyer@apple.com
Dan	USA	dmiller@comcast.com

รันออกเพราะรัน from ก่อน where แล้ว Select

```
-- commemt
SELECT
  firstname,
    country,
    email
FROM customers
WHERE customerid >=1 AND customerid <= 5;
      customerid Between 1 AND 5
```

-บน sql ไม่สามารถ define datatype เป็น date ได้ วิธีการอ่านค่า text ใช้คำสั่ง
"STRFTIME('%Y%m, column_date)" as example below

```
SELECT
  STRFTIME('%Y',invoicedate) AS year,
    STRFTIME('%m',invoicedate) AS month,
    STRFTIME('%d',invoicedate) AS day
FROM invoices --PosgreSQL SELECT EXTRACT
```

⋮ year	month	day
2009	01	01
2009	01	02
2009	01	03
2009	01	06
2009	01	11

-บน sql ไม่สามารถ define datatype เป็น date ได้ วิธีการอ่านค่า text ใช้คำสั่ง "STRFTIME('%Y%m, column_date)" as example below

ใช้คำสั่ง "typeof" to check data type

CAST(value AS INT) ใช้คำสั่ง CAST เปลี่ยน data type ka

```
--typeof check data type
SELECT
  CAST(STRFTIME('%Y%m',invoicedate) AS INT) AS monthid

FROM invoices
WHERE monthid = 200909
```
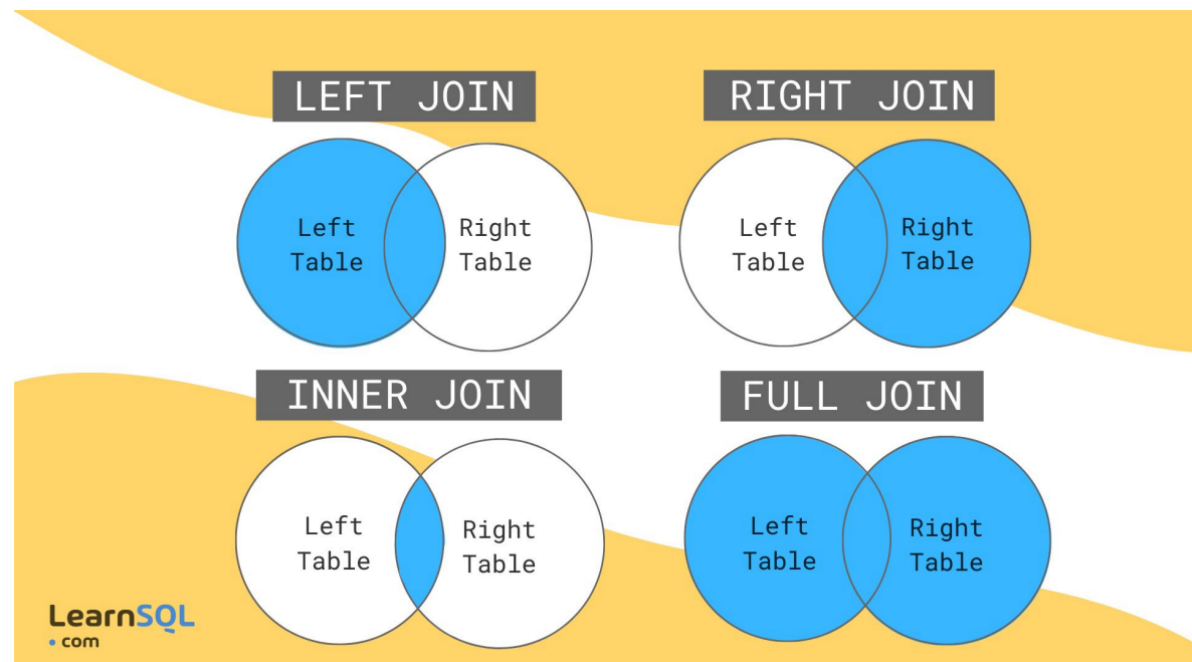
⋮ monthid
200909
200909
200909
200909
200909

Subquery & JOIN table

ดึง table ใส่ ( )

(default JOIN = INNER JOIN)

Left Join & Inner Join ต่างกันยังไง


```
--
SELECT
  *
FROM artists
JOIN albums ON artists.artistid = albums.artistid
JOIN tracks ON albums.albumid = tracks.albumid;
```

```
--
SELECT
  *
FROM artists AS t1
JOIN albums AS t2 ON t1.artistid = t2.artistid
JOIN tracks AS t3 ON t2.albumid = t3.albumid
WHERE t2.title LIKE 'The %' AND t3.composer LIKE 'm%';
```

Arti...	Name	AlbumId	Title	ArtistId	TrackId	Name	AlbumId	Media...	GenreId	Comp...	Millis...	Bytes	UnitPr..
68	Miles D…	48	The Es…	68	597	Now's …	48	1	2	Miles D…	197459	6358868	0.99
68	Miles D…	48	The Es…	68	598	Jeru	48	1	2	Miles D…	193410	6222536	0.99
68	Miles D…	48	The Es…	68	599	Compu…	48	1	2	Miles D…	345025	11254474	0.99
68	Miles D…	48	The Es…	68	600	Tempu…	48	1	2	Miles D…	231784	7548434	0.99
68	Miles D…	48	The Es…	68	601	Walkin'	48	1	2	Miles D…	807392	26411634	0.99

column in GROUP BY, should be in SELECT

```
--aggregrate function
SELECT
  ge.name,
  COUNT(*) AS no_tracks,
    AVG(tr.bytes) AS avg_bytes,
    SUM(tr.bytes) AS sum_bytes,
    MIN(tr.bytes) AS min_bytes,
    MAX(tr.bytes) AS max_bytes
FROM artists AS ar
JOIN albums AS al ON ar.artistid = al.artistid
JOIN tracks AS tr ON al.albumid = tr.albumid
JOIN genres AS ge ON tr.genreid = ge.genreid
GROUP BY ge.Name
ORDER BY no_tracks DESC;
```

Name	no_tracks	avg_bytes	sum_bytes	min_bytes	max_bytes
Rock	1297	9007374.267540477	11682564425	38747	52490554
Latin	579	7710588.575129534	4464430785	1095012	18092739
Metal	374	9234573.256684491	3453730398	1351993	25966720
Alternative & Punk	332	7691002.942771085	2553412977	161266	18139840
Jazz	130	9488136.546153845	1233457751	4011615	29416781

"Having" ใช้ในการ filter กลุ่ม

หลัง "GROUP BY" ก่อน "ORDER BY"

Where clause ใช้ในการ filter table ก่อน Group By

```
--aggregrate function
SELECT
  ge.name,
  COUNT(*) AS no_tracks,
    AVG(tr.bytes) AS avg_bytes,
    SUM(tr.bytes) AS sum_bytes,
    MIN(tr.bytes) AS min_bytes,
    MAX(tr.bytes) AS max_bytes
FROM artists AS ar
JOIN albums AS al ON ar.artistid = al.artistid
JOIN tracks AS tr ON al.albumid = tr.albumid
JOIN genres AS ge ON tr.genreid = ge.genreid
WHERE al.title NOT LIKE 'The %'
GROUP BY ge.Name
HAVING ge.name LIKE 'S%'
ORDER BY ge.name;
```

Name	no_tracks	avg_bytes	sum_bytes	min_bytes	max_bytes
Sci Fi & Fantasy	26	532930426.15384614	13856191080	486233524	587051735
Science Fiction	13	507078984.15384614	6592026794	462818231	552893447
Soundtrack	43	8090771.837209302	347903189	1039615	12727928

```
SELECT * FROM (
  SELECT
      ar.name AS artist_name,
      al.title,
      tr.name AS track,
      tr.bytes,
      ge.Name AS genre
  FROM artists AS ar
  JOIN albums AS al ON ar.artistid = al.artistid
  JOIN tracks AS tr ON al.albumid = tr.albumid
  JOIN genres AS ge ON tr.genreid = ge.genreid
)
WHERE genre = "Rock" AND artist_name LIKE 'd%'
```

artist_name	title	track	bytes	genre
David Coverdale	Into The Light	Into The Light	2452653	Rock
David Coverdale	Into The Light	River Song	14359478	Rock
David Coverdale	Into The Light	She Give Me ...	8385478	Rock
David Coverdale	Into The Light	Don't You Cry	11269612	Rock
David Coverdale	Into The Light	Love Is Blind	11409720	Rock

Common table expression (WITH)

WITH Clause เหมือนประกาศตัวแปร

```
-- sub = select *

WITH sub AS (
  SELECT
      ar.name AS artist_name,
      al.title,
      tr.name AS track,
      tr.bytes,
      ge.Name AS genre
  FROM artists AS ar
  JOIN albums AS al ON ar.artistid = al.artistid
  JOIN tracks AS tr ON al.albumid = tr.albumid
  JOIN genres AS ge ON tr.genreid = ge.genreid


)

SELECT artist_name, track, bytes, genre FROM sub
WHERE genre = "Rock" AND artist_name LIKE 'M%'
```

artist_name	track	bytes	genre
Men At Work	No Sign of Yesterday	11829011	Rock

GROUP BY 1 คอลัมน์ 1

```
WITH usa_customers AS (
  SELECT * FROM customers
  WHERE country ='USA'
), invoice_2010 AS (
  SELECT * FROM invoices
  WHERE STRFTIME("%Y",invoicedate) = "2010"
)

SELECT c.firstname, SUM(i.total) total FROM usa_customers c
JOIN invoice_2010 i ON c.customerid = i.customerid
GROUP BY 1 ;
```

FirstName	total
Dan	15.84
Frank	42.59
Heather	11.879999999999999
Jack	0.99
Julia	8.91

Window Function

```
-- window functionalbums

SELECT
  ROW_NUMBER() OVER() AS rowNum,
  firstname,
    country
FROM customers
ORDER BY 1;
```

rowNum	FirstName	Country
1	Luís	Brazil
2	Leonie	Germany
3	François	Canada
4	Bjørn	Norway
5	František	Czech Republic

```
-- window functionalbums

SELECT
  ROW_NUMBER() OVER(PARTITION BY country ORDER BY firstname) AS rowNum,
  firstname,
    country
FROM customers;
```

rowNum	FirstName	Country
1	Aaron	Canada
2	Edward	Canada
3	Ellie	Canada
4	François	Canada
5	Jennifer	Canada

Where ใช้ดึง ใน window ตรงๆไม่ได้ๆ ต้องใส่ subquery

```
-- window functionalbums

SELECT * FROM (

SELECT
  ROW_NUMBER() OVER(PARTITION BY country ORDER BY firstname) AS rowNum,
  firstname,
    country
FROM customers
  )
WHERE rowNum = 1;
```

rowNum	firstname	country
1	Diego	Argentina
1	Mark	Australia
1	Astrid	Austria
1	Daan	Belgium
1	Alexandre	Brazil

```
-- window function NTILE()
SELECT segment, COUNT(*) FROM (
SELECT
  name AS songname,
    milliseconds,
    NTILE(5) OVER(ORDER BY milliseconds) AS segment
FROM tracks
)
GROUP BY segment
```

segment	COUNT(*)
1	701
2	701
3	701
4	700
5	700

CASE WHEN เงื่อนไข THEN value

ELSE ค่าอื่น จบที่ END

```
-- window function NTILE()
SELECT *,
  CASE WHEN segment IN (1,2) THEN 'LOW'
       WHEN segment in (4,5) THEN 'High'
     ELSE 'Medium'
     END label_segment
FROM(

SELECT segment, COUNT(*) FROM (
SELECT
  name AS songname,
    milliseconds,
    NTILE(5) OVER(ORDER BY milliseconds) AS segment
FROM tracks
)
GROUP BY segment)
```

segment	COUNT(*)	label_segment
1	701	LOW
2	701	LOW
3	701	Medium
4	700	High
5	700	High

```
SELECT
  firstname,
    country,
    CASE WHEN country IN ( 'USA','Canada') THEN  'America'
      WHen country IN ('Belgium','Italy','France') THEN 'Europe'
        ELSE 'Other Region'
     END AS Region
FROM customers
```

FirstName	Country	Region
Luís	Brazil	Other Region
Leonie	Germany	Other Region
François	Canada	America
Bjørn	Norway	Other Region
František	Czech Republic	Other Region

```
UPDATE film
SET rental_rate = rental_rate - 1
WHERE film_id IN
  (SELECT film_id from actor AS a
   INNER JOIN film_actor AS f
     ON a.actor_id = f.actor_id
   WHERE last_name IN ('WILLIS', 'CHASE', 'WINSLET', 'GUINESS', 'HUDSON'));
```