

# DATA MANAGEMENT

## SQL Segmento 2

# Índice

## 1. AGREGACIONES

- a. Funciones de agregación
- b. GROUP BY statement
- c. HAVING clause

## 2. TRABAJANDO CON FECHAS

- a. EXTRACT function
- b. DATE\_TRUNC function

## 3. AGREGACIONES CONDICIONALES

## 4. PRÁCTICA



# Agregaciones



# Agregaciones

## Introducción a agregaciones en SQL

En el contexto del análisis de datos, agregar data significa realizar cálculos que permitan resumir información, abstrayendo conocimiento de data que se encuentra en estado bruto. Cuando realizamos agregaciones, podemos elegir agrupar observaciones en una o más variables categóricas.

Para realizar una agregación, debemos utilizar funciones específicas (funciones de agregación). En caso de querer agrupar observaciones, debemos operar utilizando el **GROUP BY** statement.

### *Funciones de agregación básicas*

|                              |  |
|------------------------------|--|
| <b>COUNT(*)</b>              | Counts observations in data table, takes NA in consideration |
| <b>COUNT(field)</b>          | Counts non-NA observations in field                          |
| <b>COUNT(distinct field)</b> | Counts non-NA unique values in a field                       |
| <b>SUM(field)</b>            | Returns the sum of all value (only numerical fields)         |
| <b>MAX(field)</b>            | Returns the largest value in a column                        |
| <b>MIN(field)</b>            | Returns the smallest value in a column                       |
| <b>AVG(field)</b>            | Returns the average value of a field (only numerical fields) |

# Agregaciones

## GROUP BY Statement

La GROUP BY statement agrupa filas que tienen los mismos valores en filas de resumen, como "encontrar la cantidad de clientes en cada país".

La GROUP BY statement se usa a menudo con funciones agregadas COUNT(), MAX(), MIN(), SUM(), AVG() para agrupar el conjunto de resultados por una o más columnas.

# Agregaciones

## Group By statements

*SQL Query*

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

*Output*

| COUNT(movie_title) | country |
|--------------------|---------|
| 315                | USA     |
| 43                 | UK      |
| 33                 | France  |
| 20                 | India   |
| 13                 | Cada    |
| 9                  | Chi     |

# Agregaciones

## HAVING Clause

La HAVING clause se agregó a SQL porque la WHERE clause no se puede usar con funciones agregadas.

SQLite.3

```
SELECT COUNT(movie_title), country
FROM imdb_movies
WHERE COUNT(movie_title) > 20
GROUP BY country
-
...
```

Help: misuse of aggregate: COUNT()

18:15:28

# Agregaciones

## Having Clause

*SQL Query*

```
1 SELECT COUNT(movie_title), country
2 FROM imdb_movies
3 GROUP BY country
4 HAVING COUNT(movie_title) > 20
5 ORDER BY COUNT(movie_title) DESC;
```

*Output*

| COUNT(movie_title) | country |
|--------------------|---------|
| 315                | USA     |
| 43                 | UK      |
| 33                 | France  |
|                    |         |
|                    |         |
|                    |         |



# Trabajando con Fechas en SQL

# Trabajando con Fechas en SQL

## Funciones Extract() & Date\_trunc()

Funciones fundamentales que utilizaremos para operar variables de formato fecha:

- *EXTRACT()*: esta función permite extraer un *part* (elemento) concreto de la fecha, dando como resultado un número  
*Sintaxis: EXTRACT(date\_part FROM date\_field)*  
*Ejemplo: EXTRACT(month FROM date\_field); esto devolvería el mes de la fecha*
- *DATE\_TRUNC()*: esta función devuelve la fecha al inicio del *part* y genera un resultado en formato fecha  
*Sintaxis: DATE\_TRUNC(date\_field, date\_part)*  
*Ejemplo: DATE\_TRUNC(month, date\_part); esto devuelve la fecha al inicio del mes*

# Trabajando con Fechas en SQL

## Funciones Extract() & Date\_trunc()

```
SELECT
    Date
    ,EXTRACT(DAY FROM Date) as day_number
    ,EXTRACT(WEEK FROM Date) as week_number
    ,EXTRACT(MONTH FROM Date) as month_number
    ,EXTRACT(YEAR FROM Date) as year_number
    ,DATE_TRUNC(Date,WEEK) as week_starting_date
    ,DATE_TRUNC(Date,MONTH) as month_starting_date

FROM 'amazon_table'

ORDER BY Date
```

### ***Date Parts***

- year = year
- quarter = quarter
- month = month
- dayofyear = day of the year
- dayofweek = day of the week
- day = day of the month
- week = week
- hour = hour
- time = time
- minute = minute
- second = second
- millisecond = millisecond

# Trabajando con Fechas en SQL

## Funciones Extract() & Date\_trunc()

| Row | Date       | day_number | week_number | month_number | year_number | week_starting_date | week_starting_date_1 |
|-----|------------|------------|-------------|--------------|-------------|--------------------|----------------------|
| 1   | 2010-01-04 | 4          | 1           | 1            | 2010        | 2010-01-03         | 2010-01-01           |
| 2   | 2010-01-05 | 5          | 1           | 1            | 2010        | 2010-01-03         | 2010-01-01           |
| 3   | 2010-01-06 | 6          | 1           | 1            | 2010        | 2010-01-03         | 2010-01-01           |
| 4   | 2010-01-07 | 7          | 1           | 1            | 2010        | 2010-01-03         | 2010-01-01           |
| 5   | 2010-01-08 | 8          | 1           | 1            | 2010        | 2010-01-03         | 2010-01-01           |
| 6   | 2010-01-11 | 11         | 2           | 1            | 2010        | 2010-01-10         | 2010-01-01           |
| 7   | 2010-01-12 | 12         | 2           | 1            | 2010        | 2010-01-10         | 2010-01-01           |
| 8   | 2010-01-13 | 13         | 2           | 1            | 2010        | 2010-01-10         | 2010-01-01           |
| 9   | 2010-01-14 | 14         | 2           | 1            | 2010        | 2010-01-10         | 2010-01-01           |
| 10  | 2010-01-15 | 15         | 2           | 1            | 2010        | 2010-01-10         | 2010-01-01           |

# Agregaciones<sup>ⓧ</sup>

# Condicionales<sup>ⓧ</sup>

# Agregaciones Condicionales

## Introducción a agregaciones condicionales en SQL

En una agregación condicional restringimos la variable sobre la cual buscamos operar al cumplimiento de una o más condiciones. En otras palabras, realizamos una operación sobre un subconjunto de observaciones (filas) que cumplan con una o más condiciones específicas.

Para implementar una agregación condicional, simplemente necesitamos incluir un **CASE WHEN** statement dentro de una función de agregación.

```
SELECT
```

```
...
```

```
FUNCIÓN DE AGREGACIÓN (CASE WHEN statement)
```

# Agregaciones Condicionales

## Ejemplo

SQL Query

```
1 SELECT
2 SUM(CASE WHEN Sector='Technology' THEN Revenue END) AS revenue_technology,
3 SUM(CASE WHEN Sector='Energy' THEN Revenue END) AS revenue_energy,
4 SUM(CASE WHEN Sector='Media' THEN Revenue END) AS revenue_media
5 FROM fortune
```

Output

| revenue_technology | revenue_energy | revenue_media |
|--------------------|----------------|---------------|
| 988182             | 1517809        | 220764        |

Data Management

# Práctica





# Práctica

## Ex.0 (Recap): Descripción

Extraer top 10 personajes de Star Wars con mayor masa (*mass*)

- *Tabla*: star\_wars\_characters
- *Descripción de tabla*: esta tabla incluye datos sobre personajes de la saga Star Wars

*Resultado esperado*

| Row | name                  | mass   |
|-----|-----------------------|--------|
| 1   | Jabba Desilijic Tiure | 1358.0 |
| 2   | Grievous              | 159.0  |
| 3   | IG-88                 | 140.0  |
| 4   | Tarfful               | 136.0  |
| 5   | Darth Vader           | 136.0  |
| 6   | Owen Lars             | 120.0  |
| 7   | Bossk                 | 113.0  |
| 8   | Chewbacca             | 112.0  |
| 9   | Jek Tono Porkins      | 110.0  |
| 10  | Dexter Jettster       | 102.0  |

## Práctica

### Ex.1: Descripción

Agregar las siguientes métricas para todos los países africanos,

- average gross income per capita
- total population
- number of countries

*Tabla:* world\_health\_org (`Basics.world\_health\_org`)

*Descripción de la tabla:* esta tabla contiene data de países del mundo publicados por la Organización Mundial de la Salud

*Resultado esperado*

| Row | africa_avg_GDP | total_population | count_countries |
|-----|----------------|------------------|-----------------|
| 1   | 3127.95        | 759147           | 46              |

# Práctica

## Ex.2: Descripción

Calcular número de personajes según planeta (*homeworld*). Evitar personajes sin información sobre planeta de origen.

*Tabla:* star\_wars\_characters (`star\_wars\_characters`)

*Descripción de la tabla:* esta tabla incluye datos sobre personajes de Star Wars

*Resultado esperado*

| Row | homeworld | count_characters |
|-----|-----------|------------------|
| 1   | Naboo     | 11               |
| 2   | Tatooine  | 10               |
| 3   | Coruscant | 3                |
| 4   | Kamino    | 3                |
| 5   | Alderaan  | 3                |

## Práctica

### Ex.3: Descripción

Calcular el total de salario percibido por cada actor en todas las películas. Omitir películas sin data sobre salario.

*Tabla:* james\_bond (`sepe-sql-256409.SQL\_Basics.james\_bond`)

*Descripción de la tabla:* esta tabla reporta data sobre películas de James Bond

*Resultado esperado*

| Row | Actor          | bond_salary |
|-----|----------------|-------------|
| 1   | Pierce Brosnan | 46.5        |
| 2   | Daniel Craig   | 25.9        |
| 3   | Sean Connery   | 20.3        |
| 4   | Roger Moore    | 16.9        |
| 5   | Timothy Dalton | 13.1        |
| 6   | George Lazenby | 0.6         |

# Práctica

## Ex.4: Descripción

¿Podemos asegurar que las películas de acción tienen de media mejor valoración que el resto de películas? Extraer total de películas y media de IMDB score para películas de acción vs. el resto (de forma conjunta).

*Tabla: imdb\_movies*

*Resultado esperado*

| gender         | total_movies | AVG_imdb_score |
|----------------|--------------|----------------|
| Action Movies  | 84           | 6.52           |
| Rest of Movies | 416          | 6.52           |

## Práctica

### Ex.5: Descripción

Calcular la facturación (*box office*) según director. Filtrar por aquellos directores que hayan generado más de 1500 en el total de facturación (todas las películas).

*Tabla:* james\_bond

*Descripción de la tabla:* esta tabla reporta data sobre películas de James Bond

*Resultado esperado*

| Row | Director      | total_box_office |
|-----|---------------|------------------|
| 1   | Guy Hamilton  | 2057.0           |
| 2   | Terence Young | 1841.0           |
| 3   | Sam Mendes    | 1670.0           |
| 4   | John Glen     | 1663.0           |
| 5   | Lewis Gilbert | 1582.0           |

## Práctica

### Ex.6: Descripción

Calcular número total de álbumes según *sub metal genre*, filtrar por aquellos subgéneros con al menos 10 álbumes.

*Tabla:* rolling\_top\_albums

*Descripción de la tabla:* esta tabla incluye datos sobre álbumes de música heavy metal

*Resultado esperado*

| Row | Sub_Metal_Genre   | count_albums |
|-----|-------------------|--------------|
| 1   | Thrash Metal      | 15           |
| 2   | Heavy Metal       | 14           |
| 3   | Alternative Metal | 10           |

## Práctica

### Ex.7: Descripción

¿Cuántos artistas hay incluidos en el dataset cuyo nombre incluye las palabras 'god', 'death' or 'black'?

*Tabla:* rolling\_top\_albums

*Descripción de la tabla:* esta tabla incluye datos sobre álbumes de música heavy metal

*Resultado esperado*

| Row | artist_keyword | count_artist |
|-----|----------------|--------------|
| 1   | GOD            | 3            |
| 2   | DEATH          | 2            |
| 3   | BLACK          | 1            |



# Práctica

## Ex.8: Descripción

Extraer media mensual de la cotización (*open rate*) y volumen de operación (volume) del bitcoin desde el año 2016.

*Tabla: bitcoin\_daily\_rates*

*Descripción de la tabla: esta tabla incluye cotizaciones diarias de bitcoin*

*Resultado esperado*

| Row | month | year | AVG_open_rate | AVG_volume_USD |
|-----|-------|------|---------------|----------------|
| 1   | 1     | 2016 | 413.79        | 384478.0       |
| 2   | 2     | 2016 | 400.82        | 398438.0       |
| 3   | 3     | 2016 | 415.19        | 547732.0       |
| 4   | 4     | 2016 | 433.56        | 541883.0       |
| 5   | 5     | 2016 | 459.65        | 483617.0       |
| 6   | 6     | 2016 | 638.62        | 1304165.0      |
| 7   | 7     | 2016 | 663.95        | 647291.0       |

## Práctica

### Ex.9: Descripción

¿Cuál fue la semana con el valor mayor de cotización? Utilizar cotización *high*.

*Tabla: amazon\_stocks*

*Descripción de la tabla: esta tabla incluye cotizaciones diarias de la acción de Amazon*

*Resultado esperado*

| Row | week_starting_date | weeks_highest_rate |
|-----|--------------------|--------------------|
| 1   | 2017-06-18         | 1017.0             |

# Práctica

## Ex.10: Descripción

Calcular el total de sesiones según canal para octubre de 2019. Crear métricas específicas agregadas para cada dispositivo.

*Tabla: google\_analytics*

*Descripción de la tabla: contiene data extraída de Google Analytics para una web dummy*

*Resultado esperado*

| Row | channelGrouping | mobile_sessions | desktop_sessions | tablet_sessions | total_sessions |
|-----|-----------------|-----------------|------------------|-----------------|----------------|
| 1   | Paid Search     | 2052            | 61               | 52              | 2165           |
| 2   | Social          | 1634            | 83               | 37              | 1754           |
| 3   | Direct          | 641             | 88               | 38              | 767            |
| 4   | Organic Search  | 440             | 149              | 33              | 622            |
| 5   | Referral        | 215             | 23               | 10              | 248            |

## Práctica (individual)

### Ex.11.1: Descripción

¿Cuántas películas duran menos de 60 minutos?; ¿Cuántas entre 60 y 100? Y  
¿Cuántas más de 100?

*Tabla: imdb\_movies*

## Práctica (individual)

### Ex.11.2: Descripción

¿Cuántas películas de acción hay que duren menos de 60 minutos? Haz un listado de las mismas.

*Tabla: imdb\_movies*

## Práctica (individual)

### Ex.11.3: Descripción

¿Cuál sería el día de menos cotización en una tendencia alcista en el año 2018?; ¿Y la media ese mismo día?

*Tabla: bitcoin\_daily\_rates\_formatdate*

## Práctica (individual)

### Ex.11.4: Descripción

Mostrar el conteo de las películas relacionadas con los géneros (Action, Crime, Comedy, Drama, Romance), indicando la película con mayor número de votos en cada caso (num\_voted\_users).

\*\* Utilizar el orden de (Action, Crime, Comedy, Drama, Romance) al relizar la tabla.

*Tabla: imdb\_movies*

## Práctica (individual)

### Ex.11.5: Descripción

Mostrar el número de personajes que tienen el mismo color de ojos (eye\_color) y el planeta de origen (homeworld). No mostrar color de ojos desconocidos (unknown) ni planetas sin datos/nombre (NA).

*Tabla: star\_wars\_characters*



## Práctica (individual)

### Ex.11.6: Descripción

Identificar y calcular el presupuesto de aquellas películas de James Bond que fueron dirigidas por John Glen y protagonizadas por Timothy Dalton.

*Tabla: jamesbond*

## Práctica (individual)

### Ex.11.7: Descripción

¿Cuál es el monto de los créditos otorgados y no otorgados según el Status personal?

*Tabla: loan-data*

## Práctica (individual)

### Ex.11.8: Descripción

Obtén un listado de las películas de acción con actor protagonista con más de 10000 likes en Facebook y cuyas películas hayan sido valoradas con al menos un 8 en imdb. Todo ello con fechas anteriores a 2012.

*Tabla: imdb\_movies*

## Práctica (individual)

### Ex.11.9: Descripción

Queremos saber cuáles son las 20 películas y género al que pertenecen,

- 1) con mayor presupuesto
- 2) con mayor beneficio

*Tabla: imdb\_movies*

## Práctica (individual)

### Ex.11.10: Descripción

Agregar las siguientes métricas para 10 primeras localizaciones:

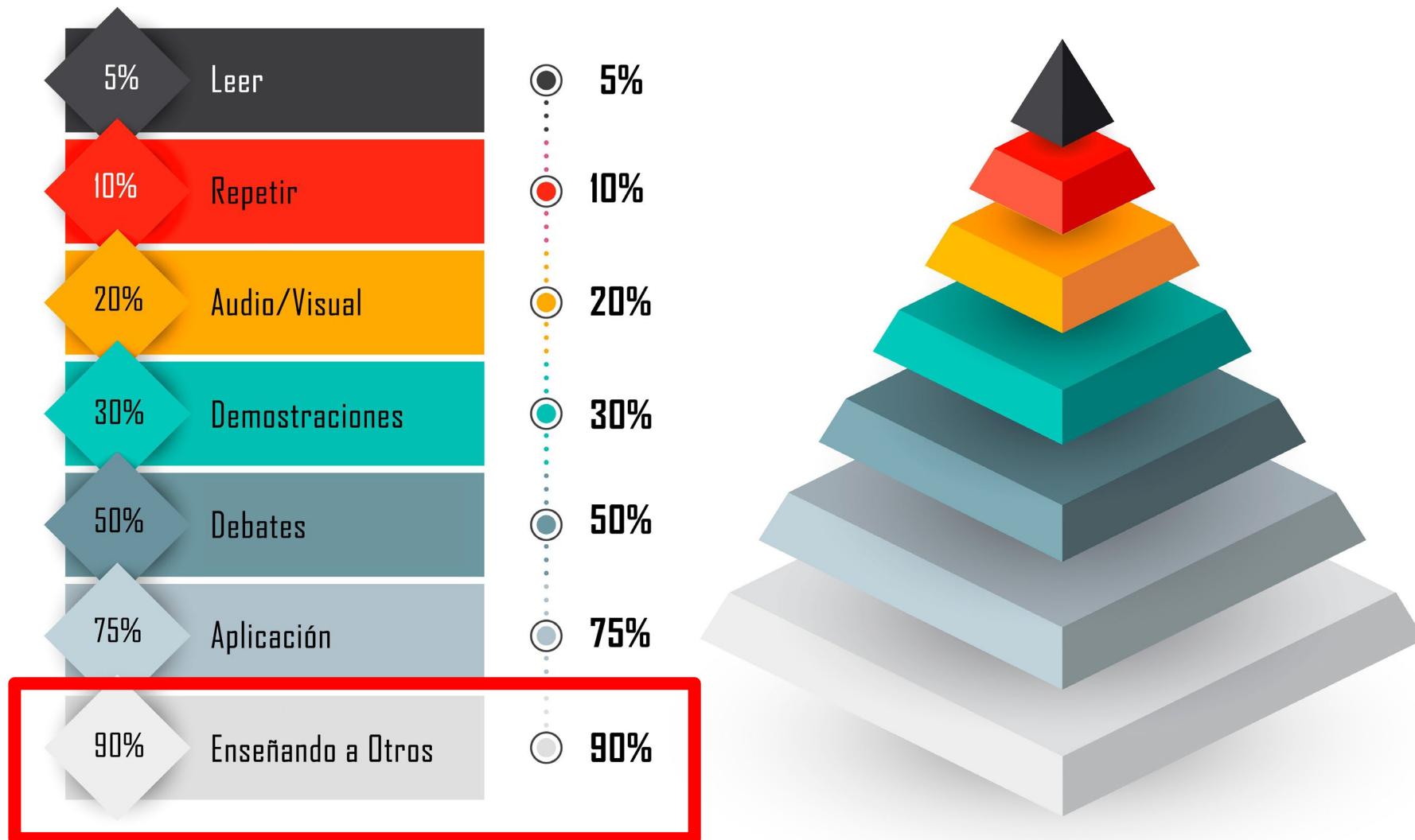
- Profits
- Rank
- Employees

*Tabla: fortune*

Práctica (individual)

**REGALARLE UN EJERCICIO A TUS  
COMPAÑEROS**

## La Pirámide del Aprendizaje



## Práctica (individual)

### Ex.12: Descripción

Formula un ejercicio y su sintaxis con los siguientes elementos:

- Enunciado
- La cosa curiosa de tu ejercicio que destacarías
- Tu solución

*Tabla: #túdecides*





red.es

Centro de  
Referencia Nacional  
en Comercio Electrónico  
y Marketing

CRN  
Digital



UNIÓN EUROPEA

*"El FSE invierte en tu futuro"*

**Fondo Social Europeo**

