



---

CARRERA DE ESPECIALIZACIÓN EN INTELIGENCIA ARTIFICIAL

---

## **Bases de Datos para Inteligencia Artificial**

16Co2024

### **Trabajo Práctico N°1**

#### ***PostgreSQL***

DEGANO, MYRNA LORENA

N° SIU a1618

[myrna.l.degano@gmail.com](mailto:myrna.l.degano@gmail.com)

marzo de 2025

## Índice

SetUp.....	2
Consultas.....	4
1).....	4
2).....	5
3).....	6
4).....	7
5).....	8
6).....	9
7).....	10
8).....	11
9).....	12
10).....	14
11).....	17
12).....	18
13).....	19
14).....	21
15).....	22
16).....	24
17).....	26
18).....	28
19).....	30
20).....	32
Anexos.....	33
Referencias .....	33

## SetUp

Para crear el entorno de trabajo con Docker, ejecutar:

```
docker-compose up -d
```

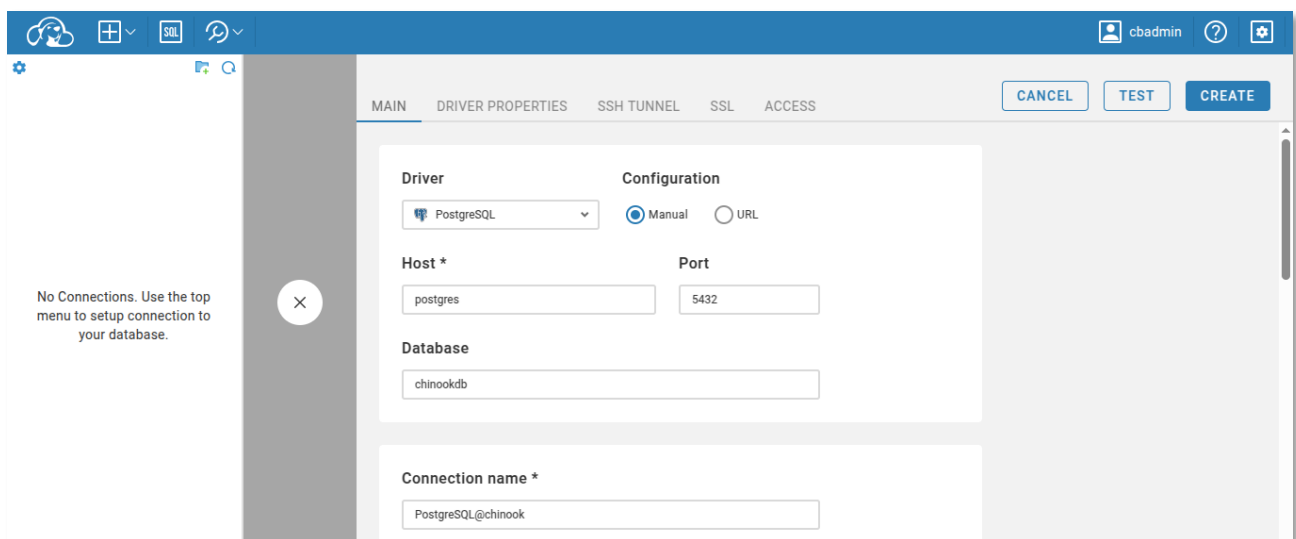
Esto crea los contenedores con PostgreSQL y CloudBeaver:

```
myrna@myrna-Latitude-E5470:~/Projects/BDIA/BDIA-TP1$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
273d47d36dba	dbeaver/cloudbeaver:latest	"/launch-product.sh"	About a minute ago	Up About a minute (healthy)	0.0.0.0:8978->8978/tcp, [::]:8
978->8978/tcp	cloudbeaver				
27299b501542	postgres:latest	"docker-entrypoint.s..."	About a minute ago	Up About a minute (healthy)	0.0.0.0:5432->5432/tcp, [::]:5
432->5432/tcp	postgres				

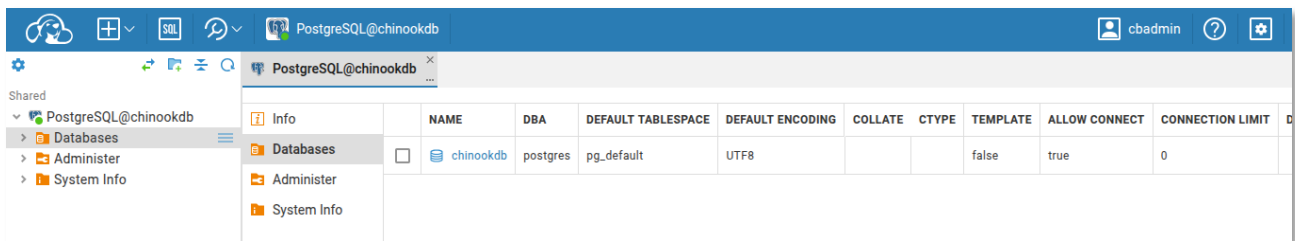
El contenedor de **PostgreSQL** se inicializa con la base de datos *Chinook* propuesta para el trabajo práctico.

Se utiliza **CloudBeaver** para conectarse, gestionar y consultar la base de datos a través de una interfaz gráfica accesible desde el navegador web.



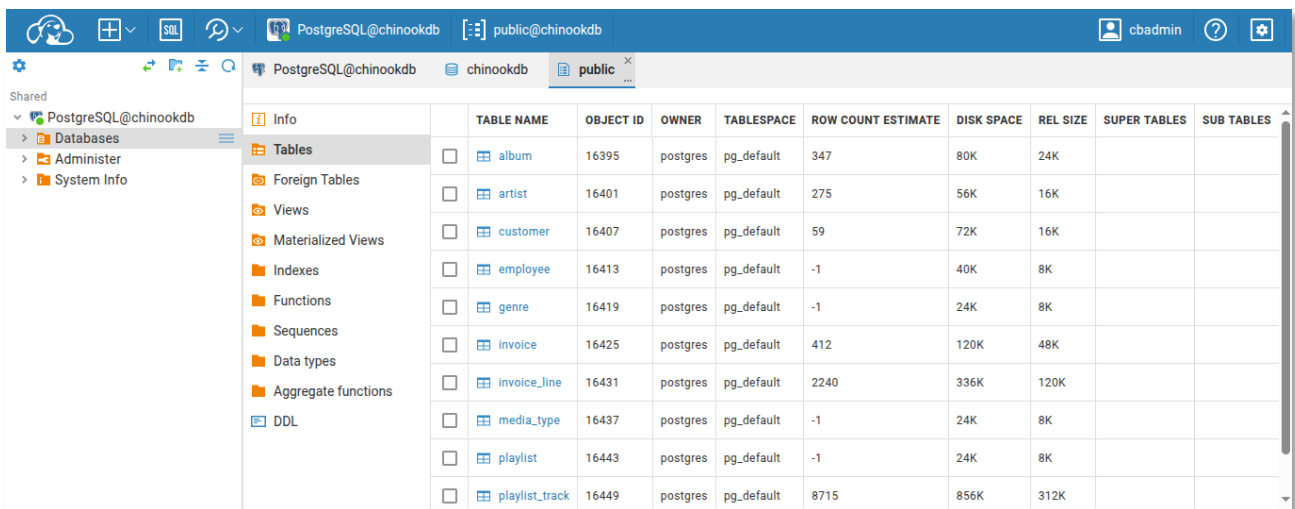
Una vez establecida la contraseña de administrador, se establecer la conexión a la DB.

- Driver: PostgreSQL
- Host: postgres
- Database: chinookdb
- Port: 5432
- User/Pass: postgres/postgres



The screenshot shows the PostgreSQL Enterprise Console interface. The left sidebar shows a tree view with 'PostgreSQL@chinookdb' expanded, showing 'Databases', 'Administer', and 'System Info'. The main pane displays the 'Databases' tab for 'chinookdb', showing a table with columns: NAME, DBA, DEFAULT TABLESPACE, DEFAULT ENCODING, COLLATE, CTYPE, TEMPLATE, ALLOW CONNECT, and CONNECTION LIMIT. The table contains one row for 'chinookdb'.

NAME	DBA	DEFAULT TABLESPACE	DEFAULT ENCODING	COLLATE	CTYPE	TEMPLATE	ALLOW CONNECT	CONNECTION LIMIT
chinookdb	postgres	pg_default	UTF8			false	true	0



The screenshot shows the PostgreSQL Enterprise Console interface. The left sidebar shows a tree view with 'PostgreSQL@chinookdb' expanded, showing 'Databases', 'Administer', and 'System Info'. The main pane displays the 'Tables' tab for the 'public' schema in 'chinookdb', showing a table with columns: TABLE NAME, OBJECT ID, OWNER, TABLESPACE, ROW COUNT ESTIMATE, DISK SPACE, REL SIZE, SUPER TABLES, and SUB TABLES. The table lists various tables including album, artist, customer, employee, genre, invoice, invoice\_line, media\_type, playlist, and playlist\_track.

TABLE NAME	OBJECT ID	OWNER	TABLESPACE	ROW COUNT ESTIMATE	DISK SPACE	REL SIZE	SUPER TABLES	SUB TABLES
album	16395	postgres	pg_default	347	80K	24K		
artist	16401	postgres	pg_default	275	56K	16K		
customer	16407	postgres	pg_default	59	72K	16K		
employee	16413	postgres	pg_default	-1	40K	8K		
genre	16419	postgres	pg_default	-1	24K	8K		
invoice	16425	postgres	pg_default	412	120K	48K		
invoice_line	16431	postgres	pg_default	2240	336K	120K		
media_type	16437	postgres	pg_default	-1	24K	8K		
playlist	16443	postgres	pg_default	-1	24K	8K		
playlist_track	16449	postgres	pg_default	8715	856K	312K		

Consultas

1)

```
-- Selecciona todos los registros de la tabla Albums.

SELECT

    *

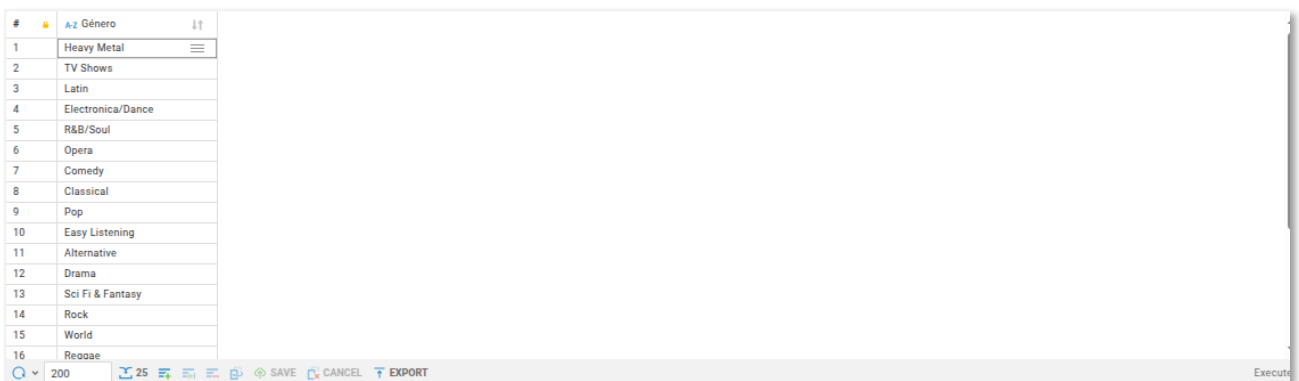
FROM

    album
```

#	123 album_id	A-2 title	123 artist_id
1	1	For Those About To Rock We Salute You	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cellos	7
10	10	Audioslave	8
11	11	Out Of Exile	8
12	12	BackBeat Soundtrack	9
13	13	The Best Of Billy Cobham	10
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11
16	16	Black Sabbath	12

2)

```
-- Selecciona todos los géneros únicos de la tabla Genres.  
  
SELECT  
  
    DISTINCT g.name AS "Género"  
  
FROM  
  
    genre AS g
```



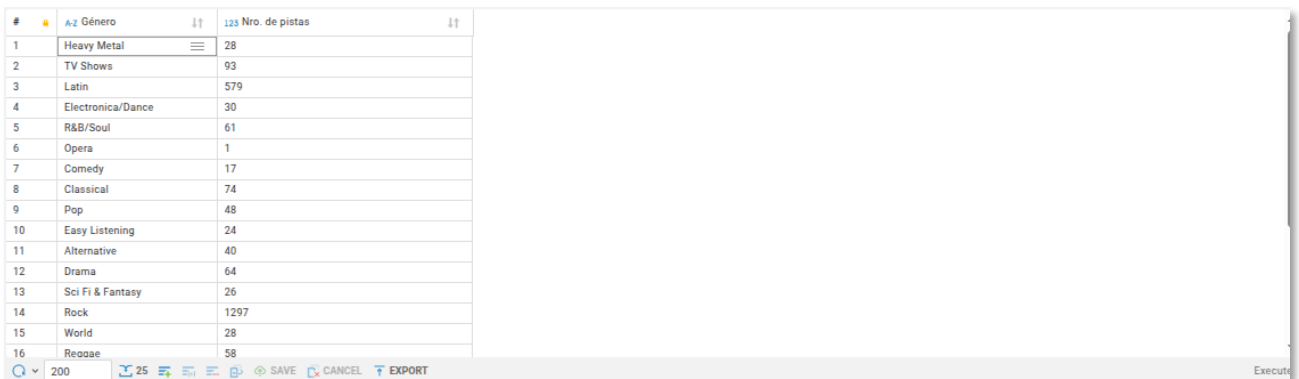
The screenshot shows a PostgreSQL query result window titled "Az Género". The window displays a table with 16 rows of genre names. The genres listed are: Heavy Metal, TV Shows, Latin, Electronica/Dance, R&B/Soul, Opera, Comedy, Classical, Pop, Easy Listening, Alternative, Drama, Sci Fi & Fantasy, Rock, World, and Reggae. The window also shows a search bar with "200" and buttons for "SAVE", "CANCEL", and "EXPORT".

#	Az Género
1	Heavy Metal
2	TV Shows
3	Latin
4	Electronica/Dance
5	R&B/Soul
6	Opera
7	Comedy
8	Classical
9	Pop
10	Easy Listening
11	Alternative
12	Drama
13	Sci Fi & Fantasy
14	Rock
15	World
16	Reggae

3)

```
-- Cuenta el número de pistas por género.

SELECT
    g.name AS "Género", COUNT(t.*) AS "Nro. de pistas"
FROM
    track AS t
    LEFT JOIN genre AS g ON t.genre_id = g.genre_id
GROUP BY
    g.name
```



The screenshot shows a PostgreSQL query result window. The query executed is: `SELECT g.name AS "Género", COUNT(t.*) AS "Nro. de pistas" FROM track AS t LEFT JOIN genre AS g ON t.genre_id = g.genre_id GROUP BY g.name`. The result is a table with two columns: "Género" and "Nro. de pistas". The data is as follows:

#	Género	Nro. de pistas
1	Heavy Metal	28
2	TV Shows	93
3	Latin	579
4	Electronica/Dance	30
5	R&B/Soul	61
6	Opera	1
7	Comedy	17
8	Classical	74
9	Pop	48
10	Easy Listening	24
11	Alternative	40
12	Drama	64
13	Sci Fi & Fantasy	26
14	Rock	1297
15	World	28
16	Reggae	58

The interface includes a search bar with "200" entered, and buttons for "SAVE", "CANCEL", and "EXPORT". The status bar at the bottom right indicates "Executing".

4)

```
-- Encuentra la longitud total (en milisegundos) de todas las pistas para cada
álbum.

SELECT
    a.title AS "Álbum", SUM(t.milliseconds) AS "Longitud Total (ms)"
FROM
    track AS t
    INNER JOIN album AS a ON t.album_id = a.album_id
GROUP BY
    a.title
```

#	Álbum	Longitud Total (ms)
1	Heart of the Night	3497581
2	The Cream Of Clapton	4373202
3	Van Halen III	3923402
4	Minha Historia	7875643
5	Chill: Brazil (Disc 2)	4221459
6	Górecki: Symphony No. 3	567494
7	IV	2557462
8	Beethoven Piano Sonatas: ...	391000
9	Miles Ahead	3134243
10	Faceless	2846138
11	Locatelli: Concertos for Viol...	493573
12	Pavarotti's Opera Made Easy	176911
13	American Idiot	3437890
14	Alcohol Fueled Brevtality Li...	4059919
15	Pachelbel: Canon & Gigue	271788
16	Great Recordings of the Cen...	265541



5)

```
-- Lista los 10 álbumes con más pistas.

SELECT
    a.title AS "Álbum",
    COUNT(t.track_id) AS "Nro. de pistas"
FROM
    album AS a
INNER JOIN
    track AS t ON a.album_id = t.album_id
GROUP BY
    a.album_id
ORDER BY
    "Nro. de pistas" DESC
LIMIT 10
```

#	A-z Álbum	123 Nro. de pistas
1	Greatest Hits	57
2	Minha Historia	34
3	Unplugged	30
4	Lost, Season 3	26
5	Lost, Season 1	25
6	The Office, Season 3	25
7	My Way: The Best Of Frank ...	24
8	Lost, Season 2	24
9	Battlestar Galactica (Classi...	24
10	Heroes, Season 1	23

200 10 SAVE CANCEL EXPORT Execute

6)

```
-- Encuentra la longitud promedio de la pista para cada género.

SELECT

    g.name AS "Género",

    AVG(t.milliseconds) "Longitud promedio del track (ms)"

FROM

    track AS t

INNER JOIN

    genre AS g ON t.genre_id = g.genre_id

GROUP BY

    g.name
```

#	Az Género	128 Longitud promedio del track (ms)
1	Heavy Metal	297452.928571428571
2	TV Shows	2145041.021505376344
3	Latin	232859.262521588946
4	Electronica/Dance	302985.8
5	R&B/Soul	220066.852459016393
6	Opera	174813
7	Comedy	1585263.705882352941
8	Classical	293867.567567567568
9	Pop	229034.104166666667
10	Easy Listening	189164.208333333333
11	Alternative	264058.525
12	Drama	2575283.78125
13	Sci Fi & Fantasy	2911783.038461538462
14	Rock	283910.043176561295
15	World	224923.821428571429
16	Reggae	247177.758620689655

7)

```
-- Para cada cliente, encuentra la cantidad total que han gastado.
```

```
SELECT
```

```
    c.first_name || ' ' || c.last_name AS "Cliente",
```

```
    SUM(i.total) AS "Total gastado"
```

```
FROM
```

```
    customer AS c
```

```
INNER JOIN
```

```
    invoice AS i ON c.customer_id = i.customer_id
```

```
GROUP BY
```

```
    c.customer_id
```

#	Az Cliente	128 Total gastado
1	Robert Brown	37.62
2	Steve Murray	37.62
3	Bjørn Hansen	39.62
4	João Fernandes	39.62
5	Joakim Johansson	38.62
6	Emma Jones	37.62
7	Eduardo Martins	37.62
8	Madalena Sampaio	37.62
9	Ladislav Kovács	45.62
10	Helena Holý	49.62
11	Camille Bernard	38.62
12	Hannah Schneider	37.62
13	Martha Silk	37.62
14	Enrique Muñoz	37.62
15	Mark Philips	37.62
16	Heather Leacock	39.62

8)

```
-- Para cada país, encuentra la cantidad total gastada por los clientes.

SELECT
    c.country AS "País",
    SUM(i.total) AS "Total gastado"
FROM
    customer AS c
INNER JOIN
    invoice AS i ON c.customer_id = i.customer_id
GROUP BY
    c.country
```

#	Az País	128 Total gastado
1	Argentina	37.62
2	Spain	37.62
3	Italy	37.62
4	Hungary	45.62
5	India	75.26
6	Czech Republic	90.24
7	Belgium	37.62
8	Sweden	38.62
9	Chile	46.62
10	Norway	39.62
11	France	195.1
12	USA	523.06
13	United Kingdom	112.86
14	Netherlands	40.62
15	Brazil	190.1
16	Austria	42.62

9)

```
-- Clasifica a los clientes en cada país por la cantidad total que han
gastado.

-- ** Clasificación propuesta: dividir el consumo total en cuartiles para
segmentar en:
-- Cuartil 1 → Consumo Bajo
-- Cuartil 2 → Consumo Medio
-- Cuartil 3 → Consumo Alto
-- Cuartil 4 → Cliente Premium
WITH TotalBilled AS (
    SELECT
        c.country AS "País del cliente",
        c.first_name || ' ' || c.last_name AS "Cliente",
        SUM(i.total) AS "Total gastado"
    FROM
        customer AS c
    JOIN
        invoice AS i ON c.customer_id = i.customer_id
    GROUP BY
        c.country, c.customer_id
)
SELECT
    "País del cliente",
    "Cliente",
    "Total gastado",
    CASE
        WHEN NTILE(4) OVER (ORDER BY "Total gastado" DESC) = 1 THEN 'Consumo Bajo'
        WHEN NTILE(4) OVER (ORDER BY "Total gastado" DESC) = 2 THEN 'Consumo
Medio'
        WHEN NTILE(4) OVER (ORDER BY "Total gastado" DESC) = 3 THEN 'Consumo Alto'
```

```
ELSE 'Cliente Premium'

END AS "Segmento de Consumo"

FROM

    TotalBilled

ORDER BY

    "País del cliente", "Total gastado" DESC
```

#	Az País del cliente	Az Cliente	123 Total gastado	Az Segmento de Consumo
1	Argentina	Diego Gutiérrez	37.62	Consumo Alto
2	Australia	Mark Taylor	37.62	Cliente Premium
3	Austria	Astrid Gruber	42.62	Consumo Bajo
4	Belgium	Daan Peeters	37.62	Cliente Premium
5	Brazil	Luis Gonçalves	39.62	Consumo Medio
6	Brazil	Eduardo Martins	37.62	Consumo Alto
7	Brazil	Alexandre Rocha	37.62	Cliente Premium
8	Brazil	Fernanda Ramos	37.62	Consumo Alto
9	Brazil	Roberto Almeida	37.62	Consumo Alto
10	Canada	François Tremblay	39.62	Consumo Bajo
11	Canada	Jennifer Peterson	38.62	Consumo Medio
12	Canada	Mark Phillips	37.62	Consumo Alto
13	Canada	Ellie Sullivan	37.62	Cliente Premium
14	Canada	Aaron Mitchell	37.62	Cliente Premium
15	Canada	Robert Brown	37.62	Consumo Alto
16	Canada	Edward Francis	37.62	Consumo Alto

10)

```
-- Para cada artista, encuentra el álbum con más pistas y clasifica a los
artistas por este número.

-- ** Clasificación propuesta: Álbum corto, mediano o largo en función de la
longitud promedio de todos los álbumes

WITH AlbumTrackCount AS (

    SELECT

        a.artist_id,

        a.title AS "Álbum",

        COUNT(t.track_id) AS "Nro. de Pistas"

    FROM

        album AS a

    JOIN

        track AS t ON a.album_id = t.album_id

    GROUP BY

        a.artist_id, a.album_id

),

MaxAlbumTrack AS (

    SELECT

        artist_id,

        MAX("Nro. de Pistas") AS "Máx Pistas en un Álbum"

    FROM

        AlbumTrackCount

    GROUP BY

        artist_id

),

AlbumStats AS (

    SELECT

        MIN("Nro. de Pistas") AS "MinPistas",

        MAX("Nro. de Pistas") AS "MaxPistas",

        AVG("Nro. de Pistas") AS "AvgPistas"
```

```
FROM
    AlbumTrackCount
)
SELECT
    ar.name AS "Artista",
    atc."Álbum",
    atc."Nro. de Pistas",
    CASE
        WHEN atc."Nro. de Pistas" <= (asv."AvgPistas" - (asv."MaxPistas" -
asv."MinPistas") / 3) THEN 'Álbum Corto'
        WHEN atc."Nro. de Pistas" <= (asv."AvgPistas" + (asv."MaxPistas" -
asv."MinPistas") / 3) THEN 'Álbum Mediano'
        ELSE 'Álbum Largo'
    END AS "Clasificación por nro. de pistas"
FROM
    AlbumTrackCount AS atc
JOIN
    artist AS ar ON atc.artist_id = ar.artist_id
JOIN
    MaxAlbumTrack AS mat ON atc.artist_id = mat.artist_id
JOIN
    AlbumStats AS asv ON 1=1
WHERE
    atc."Nro. de Pistas" = mat."Máx Pistas en un Álbum"
```



#	A-Z Artista	A-Z Álbum	123 Nro. de Pistas	A-Z Clasificación por nro. de pistas
14	Audiosiave	reveiations	14	Album Mediano
15	Habib Koité and Bamada	Muso Ko	2	Album Mediano
16	Battlestar Galactica	Battlestar Galactica, Season 3	19	Album Mediano
17	Berliner Philharmoniker, Cla...	Mozart: Wind Concertos	1	Album Mediano
18	Metallica	Garage Inc. (Disc 2)	16	Album Mediano
19	Eugene Ormandy	Strauss: Waltzes	1	Album Mediano
20	Temple of the Dog	Temple of the Dog	10	Album Mediano
21	Eric Clapton	Unplugged	30	Album Largo
22	Academy of St. Martin in th...	The World of Classical Favourites	2	Album Mediano
23	Os Mutantes	Minha História	14	Album Mediano
24	Mônica Marianno	Demorou...	12	Album Mediano
25	Joe Satriani	Surfing with the Alien (Remastered)	10	Album Mediano
26	Jamiroquai	Synkronized	11	Album Mediano
27	Godsmack	Faceless	12	Album Mediano
28	Scorpions	20th Century Masters - The Millennium Collection: The Bes...	12	Album Mediano
29	Judas Priest	Living After Midnight	16	Album Mediano

200

200+

SAVE

CANCEL

EXPORT

Executed

11)

```
-- Selecciona todas las pistas que tienen la palabra "love" en su título.

SELECT

    t.*

FROM

    track AS t

WHERE

    t.name LIKE '%love%'
```

#	123 track_id	A-Z name	123 album_id	123 media_type_id	123 genre_id
1	1134	Jesus Of Suburbia / City Of The Damned / I Don't Care / Dearly Beloved / Tales Of Another Broken Home	89	1	4
2	1468	Rollover D.J.	119	1	4
3	2401	This Velvet Glove	195	1	1

```
-- ** Considerando estrictamente la palabra "love" y no otras que la contengan
(por ejemplo, Glove, Beloved, Rollover)

SELECT

    t.*

FROM

    track AS t

WHERE

    t.name ~ '\\m love \\M';
```

12)

```
-- Selecciona a todos los clientes cuyo primer nombre comienza con 'A'.  
  
SELECT  
  
    c.*  
  
FROM  
  
    customer AS c  
  
WHERE  
  
    c.first_name LIKE 'A%'
```

#	123 customer_id	A-Z first_name	A-Z last_name	A-Z company	A-Z address	A-Z city	A-Z state	A-Z country	A-Z postal_code	A-Z phone	A-Z fax
1	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere Stadt	Vienne	[NULL]	Austria	1010	+43 01 5134505	[NULL]
2	11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo	SP	Brazil	01310-200	+55 (11) 3055-3278	+55 (11) 3055-813
3	32	Aaron	Mitchell	[NULL]	696 Osborne Street	Winnipeg	MB	Canada	R3L 2B9	+1 (204) 452-6452	[NULL]

200

3

SAVE

CANCEL

SCRIPT

EXPORT

Execute

13)

```
-- Calcula el porcentaje del total de la factura que representa cada factura.
WITH TotalBilling AS (
    SELECT
        SUM(i.total) AS "Total Facturado"
    FROM
        invoice AS i
),
TotalCustomer AS (
    SELECT
        inv.customer_id, SUM(inv.total) "Total Cliente"
    FROM
        invoice AS inv
    GROUP BY
        inv.customer_id
)
SELECT
    customer_id, "Total Cliente", "Total Cliente" * 100/"Total Facturado" AS
    "Porcentaje del total facturado"
FROM
    TotalCustomer
    LEFT JOIN TotalBilling ON 1=1
```

#	customer_id	Total Cliente	Porcentaje del total facturado
1	29	37.62	1.6155629992270033
2	54	37.62	1.6155629992270033
3	4	39.62	1.7014515159323198
4	34	39.62	1.7014515159323198
5	51	38.62	1.6585072575796616
6	52	37.62	1.6155629992270033
7	10	37.62	1.6155629992270033
8	35	37.62	1.6155629992270033
9	45	45.62	1.9591170660482693
10	6	49.62	2.1308940994589023
11	39	38.62	1.6585072575796616
12	36	37.62	1.6155629992270033
13	31	37.62	1.6155629992270033
14	50	37.62	1.6155629992270033
15	14	37.62	1.6155629992270033
16	22	39.62	1.7014515159323198

14)

```
-- Calcula el porcentaje de pistas que representa cada género

WITH TotalTracks AS (

    SELECT

        COUNT(*) AS "Tracks Totales"

    FROM

        track

)

SELECT

    g.name "Género", 1.0*COUNT(t.*)/"Tracks Totales" AS "Porcentaje del total de pistas"

FROM

    genre AS g

    LEFT JOIN track AS t ON (g.genre_id = t.genre_id)

    LEFT JOIN TotalTracks ON 1=1

GROUP BY

    g.name, "Tracks Totales"
```

#	A-Z Género	123 Porcentaje del total de pistas
1	Science Fiction	0.00371110476734227805
2	Alternative	0.01141878389951470168
3	Heavy Metal	0.00799314872966029118
4	Bossa Nova	0.00428204396231801313
5	Drama	0.01827005423922352269
6	R&B/Soul	0.01741364544675992007
7	Alternative & Punk	0.09477590636597202398
8	Rock	0.37025406794176420211
9	Electronica/Dance	0.00856408792463602626
10	Soundtrack	0.01227519269197830431
11	Sci Fi & Fantasy	0.00742220953468455609
12	Blues	0.02312303739651727091
13	World	0.00799314872966029118
14	Opera	0.00028546959748786754
15	Reggae	0.01655723665429631744
16	Metal	0.10676562946046246075

15)

```
-- Para cada cliente, compara su gasto total con el del cliente que gastó más.
WITH CustomerBill AS (
    SELECT
        c.customer_id "ID Cliente",
        SUM(i.total) AS "Gasto Total"
    FROM
        customer AS c
        LEFT JOIN invoice AS i ON c.customer_id = i.customer_id
    GROUP BY
        c.customer_id
),
MaxBill AS (
    SELECT
        MAX("Gasto Total") AS "Max Gasto"
    FROM
        CustomerBill
)
SELECT
    cb."ID Cliente",
    cb."Gasto Total",
    cmb."Max Gasto",
    (cb."Gasto Total" / cmb."Max Gasto") * 100 AS "Porcentaje del Gasto Máximo"
FROM
    CustomerBill cb
JOIN
    MaxBill cmb ON 1=1
ORDER BY
    cb."Gasto Total" DESC
```

#	123 ID Cliente	123 Gasto Total	123 Max Gasto	123 Porcentaje del Gasto Máximo
1	6	49.62	49.62	100
2	26	47.62	49.62	95.969367190648931882
3	57	46.62	49.62	93.954050785973397823
4	45	45.62	49.62	91.938734381297863765
5	46	45.62	49.62	91.938734381297863765
6	28	43.62	49.62	87.908101571946795647
7	37	43.62	49.62	87.908101571946795647
8	24	43.62	49.62	87.908101571946795647
9	7	42.62	49.62	85.892785167271261588
10	25	42.62	49.62	85.892785167271261588
11	44	41.62	49.62	83.877468762595727529
12	48	40.62	49.62	81.86215235792019347
13	5	40.62	49.62	81.86215235792019347
14	43	40.62	49.62	81.86215235792019347
15	3	39.62	49.62	79.846835953244659412
16	4	39.62	49.62	79.846835953244659412

200

59

SAVE

CANCEL

EXPORT

Execute



16)

```
-- Para cada factura, calcula la diferencia en el gasto total entre ella y la
factura anterior.

WITH Billing AS (

    SELECT

        i.invoice_id,

        i.customer_id,

        i.total AS "Gasto Total",

        LAG(i.total) OVER (ORDER BY i.invoice_date) AS "Gasto Anterior"

    FROM

        invoice AS i

)

SELECT

    b.invoice_id,

    b.customer_id,

    b."Gasto Total",

    b."Gasto Anterior",

    b."Gasto Total" - COALESCE(b."Gasto Anterior", 0) AS "Diferencia con Factura
Anterior"

FROM

    Billing b

ORDER BY

    b.invoice_id
```

#	123 invoice_id	123 customer_id	123 Gasto Total	123 Gasto Anterior	123 Diferencia con Factura Anterior
1	1	2	1.98	[NULL]	1.98
2	2	4	3.96	1.98	1.98
3	3	8	5.94	3.96	1.98
4	4	14	8.91	5.94	2.97
5	5	23	13.86	8.91	4.95
6	6	37	0.99	13.86	-12.87
7	7	38	1.98	0.99	0.99
8	8	40	1.98	1.98	0
9	9	42	3.96	1.98	1.98
10	10	46	5.94	3.96	1.98
11	11	52	8.91	5.94	2.97
12	12	2	13.86	8.91	4.95
13	13	16	0.99	13.86	-12.87
14	14	17	1.98	0.99	0.99
15	15	19	1.98	1.98	0
16	16	21	3.96	1.98	1.98

17)

```
-- Para cada factura, calcula la diferencia en el gasto total entre ella y la
próxima factura.

WITH Billing AS (
    SELECT
        i.invoice_id,
        i.customer_id,
        i.total AS "Gasto Total",
        LEAD(i.total) OVER (ORDER BY i.invoice_date) AS "Gasto Próximo"
    FROM
        invoice AS i
)
SELECT
    b.invoice_id,
    b.customer_id,
    b."Gasto Total",
    b."Gasto Próximo",
    b."Gasto Próximo" - b."Gasto Total" AS "Diferencia con Próxima Factura"
FROM
    Billing b
ORDER BY
    b.invoice_id
```

#	123 invoice_id	123 customer_id	123 Gasto Total	123 Gasto Próximo	123 Diferencia con Próxima Factura
1	1	2	1.98	3.96	1.98
2	2	4	3.96	5.94	1.98
3	3	8	5.94	8.91	2.97
4	4	14	8.91	13.86	4.95
5	5	23	13.86	0.99	-12.87
6	6	37	0.99	1.98	0.99
7	7	38	1.98	1.98	0
8	8	40	1.98	3.96	1.98
9	9	42	3.96	5.94	1.98
10	10	46	5.94	8.91	2.97
11	11	52	8.91	13.86	4.95
12	12	2	13.86	0.99	-12.87
13	13	16	0.99	1.98	0.99
14	14	17	1.98	1.98	0
15	15	19	1.98	3.96	1.98
16	16	21	3.96	5.94	1.98

18)

```
-- Encuentra al artista con el mayor número de pistas para cada género.

WITH ArtistGenreTrack AS (

    SELECT

        g.name AS "Género",
        a.name AS "Artista",
        COUNT(t.track_id) AS "Nro. de Pistas",
        RANK() OVER (PARTITION BY g.name ORDER BY COUNT(t.track_id) DESC) AS
"Ranking"

    FROM

        track AS t

    LEFT JOIN genre AS g ON t.genre_id = g.genre_id
    LEFT JOIN album AS al ON t.album_id = al.album_id
    LEFT JOIN artist AS a ON al.artist_id = a.artist_id

    GROUP BY

        g.name, a.name

)

SELECT

    agt."Género",
    agt."Artista",
    agt."Nro. de Pistas"

FROM

    ArtistGenreTrack agt

WHERE

    "Ranking" = 1

ORDER BY

    agt."Género"
```

#	A-Z Género	A-Z Artista	123 Nro. de Pistas
1	Alternative	Audioslave	14
2	Alternative	Chris Cornell	14
3	Alternative & Punk	Titãs	38
4	Blues	Eric Clapton	32
5	Bossa Nova	Toquinho & Vinícius	15
6	Classical	Eugene Ormandy	3
7	Classical	Berliner Philharmoniker & Herbert V...	3
8	Comedy	The Office	17
9	Drama	Lost	44
10	Easy Listening	Frank Sinatra	24
11	Electronica/Dance	O Rappa	17
12	Heavy Metal	Iron Maiden	28
13	Hip Hop/Rap	House Of Pain	19
14	Jazz	Miles Davis	37
15	Latin	Os Paralamas Do Sucesso	49
16	Metal	Metallica	112

19)

```
-- Compara el total de la última factura de cada cliente con el total de su
factura anterior.

WITH Billing AS (

    SELECT

        i.invoice_id,

        i.customer_id,

        i.total AS "Total Factura",

        LAG(i.total) OVER (PARTITION BY i.customer_id ORDER BY i.invoice_date) AS
        "Total Factura Anterior",

        ROW_NUMBER() OVER (PARTITION BY i.customer_id ORDER BY i.invoice_date
        DESC) AS "RowNum"

    FROM

        invoice AS i

)

SELECT

    b.customer_id,

    b.invoice_id AS "Última Factura",

    b."Total Factura" AS "Total Última Factura",

    b."Total Factura Anterior",

    b."Total Factura" - COALESCE(b."Total Factura Anterior", 0) AS "Diferencia
con Factura Anterior"

FROM

    Billing AS b

WHERE

    b."RowNum" = 1

ORDER BY

    b.customer_id
```

#	customer_id	Última Factura	Total Última Factura	Total Factura Anterior	Diferencia con Factura Anterior
1	1	382	8.91	13.86	-4.95
2	2	293	0.99	5.94	-4.95
3	3	391	0.99	5.94	-4.95
4	4	392	1.98	8.91	-6.93
5	5	361	8.91	16.86	-7.95
6	6	404	25.86	1.98	23.88
7	7	370	0.99	5.94	-4.95
8	8	394	3.96	1.98	1.98
9	9	340	8.91	13.86	-4.95
10	10	383	13.86	1.98	11.88
11	11	349	0.99	5.94	-4.95
12	12	395	5.94	3.96	1.98
13	13	319	8.91	13.86	-4.95
14	14	362	13.86	1.98	11.88
15	15	328	0.99	5.94	-4.95
16	16	374	5.94	3.96	1.98

200

59

SAVE

CANCEL

SCRIPT

EXPORT

Executed





20)

```
-- Encuentra cuántas pistas de más de 3 minutos tiene cada álbum.

SELECT
    al.title AS "Álbum",
    COUNT(t.track_id) AS "# Pistas de más de 3 mins"
FROM
    track AS t
JOIN
    album AS al ON t.album_id = al.album_id
WHERE
    t.milliseconds > (3*60*1000)
GROUP BY
    al.title
ORDER BY
    "# Pistas de más de 3 mins" DESC
```

#	Álbum	# Pistas de más de 3 mins
1	Greatest Hits	57
2	Unplugged	30
3	Lost, Season 3	26
4	Lost, Season 1	25
5	The Office, Season 3	25
6	Battlestar Galactica (Classic), Season 1	24
7	Lost, Season 2	24
8	Heroes, Season 1	23
9	Minha Historia	22
10	The Office, Season 2	22
11	Instant Karma: The Amnesty International Campaign to Save Darfur	20
12	Battlestar Galactica, Season 3	19
13	Greatest Kiss	18
14	Acústico MTV	18
15	Up An' Atom	18
16	Rotten Apples: Greatest Hits	17

## Anexos

	<ul style="list-style-type: none"><li>• <a href="#">Archivo SQL</a> con las consultas</li></ul>
	<ul style="list-style-type: none"><li>• <a href="#">Repositorio GitHub</a> con el archivo de configuración del entorno y los inputs.</li></ul>

## Referencias

- [https://github.com/FIUBA-Posgrado-Inteligencia-Artificial/BDIA/tree/main/clase\\_3](https://github.com/FIUBA-Posgrado-Inteligencia-Artificial/BDIA/tree/main/clase_3)
- <https://dbeaver.com/docs/cloudbeaver/>
- <https://www.postgresql.org/docs/17/index.html>