

Bases de datos a grande escala [P4181101] [2021/2022]

[Inicio](#) / [Os meus cursos](#) / [Curso 2021/2022](#) / [Posgrao](#) / [Bases de datos a grande escala \[P4181101\].\[2021/2022\]](#) / [Prácticas](#)
/ [Práctica 1: Consulta de bases de datos relacionales...](#)

Práctica 1: Consulta de bases de datos relacionales con SQL

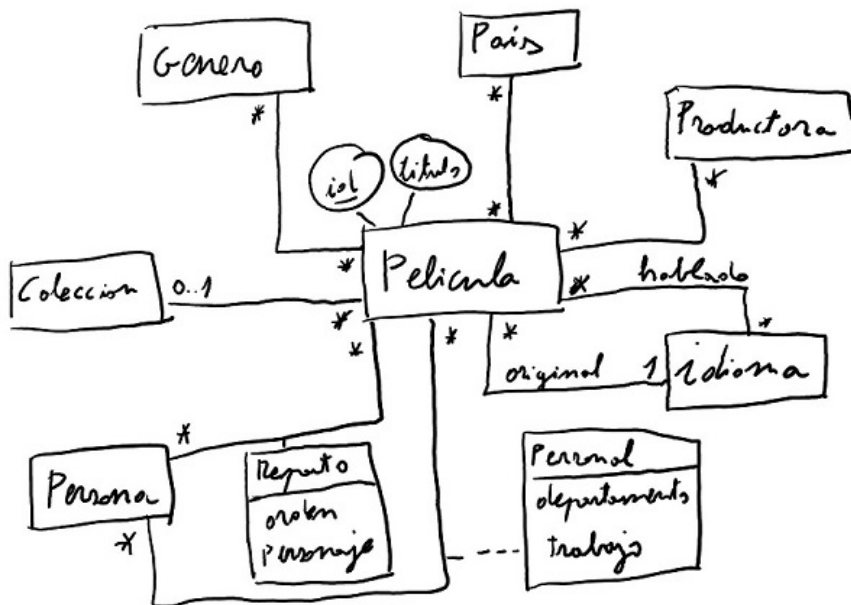
Marcar como feito

1.-Consulta de bases de datos relacionales con SQL

El objetivo de esta práctica es trabajar el uso del lenguaje SQL para resolver consultas sobre bases de datos relacionales. Usaremos la versión 13 del SGBDs PostgreSQL. La documentación puede consultarse aquí: <https://www.postgresql.org/docs/13/index.html>

1.1 Creación del esquema e inserción de datos

El esquema de la base de datos de ejemplo se encuentra en ./BDGE/datos/PeliculasSchema.sql. Si abrimos el archivo con un editor de texto podemos ver el código DDL de creación de cada tabla. El siguiente gráfico muestra el modelo conceptual de la base de datos.



Modelo conceptual de la base de datos

Después de crear una nueva base de datos en PostgreSQL llamada "bdge" ejecutar el siguiente comando para crear todas las tablas.

```
psql -U alumnogreibd -d bdge -f /home/alumnogreibd/BdGE/datos/PeliculasSchema.sql
```

Las sentencias DML de inserción de datos de cada una de las tablas se encuentra en archivos .sql en la misma carpeta ./BDGE/datos. Ejecutar los siguientes comandos para importar todos los datos.

```
psql -U alumnogreibd -d bdge -c "\copy colecciones from /home/alumnogreibd/BdGE/datos/colecciones.csv csv"
psql -U alumnogreibd -d bdge -c "\copy generos from /home/alumnogreibd/BdGE/datos/generos.csv csv"
psql -U alumnogreibd -d bdge -c "\copy idiomas from /home/alumnogreibd/BdGE/datos/idiomas.csv csv"
psql -U alumnogreibd -d bdge -c "\copy paises from /home/alumnogreibd/BdGE/datos/paises.csv csv"
psql -U alumnogreibd -d bdge -c "\copy personas from /home/alumnogreibd/BdGE/datos/personas.csv csv"
psql -U alumnogreibd -d bdge -c "\copy productoras from /home/alumnogreibd/BdGE/datos/productoras.csv csv"
psql -U alumnogreibd -d bdge -c "\copy peliculas from /home/alumnogreibd/BdGE/datos/peliculas.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_genero from /home/alumnogreibd/BdGE/datos/pelicula_genero.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_idioma_hablado from /home/alumnogreibd/BdGE/datos/pelicula_idioma_hablado.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_pais from /home/alumnogreibd/BdGE/datos/pelicula_pais.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_personal from /home/alumnogreibd/BdGE/datos/pelicula_personal.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_productora from /home/alumnogreibd/BdGE/datos/pelicula_productora.csv csv"
psql -U alumnogreibd -d bdge -c "\copy pelicula_reparto from /home/alumnogreibd/BdGE/datos/pelicula_reparto.csv csv"
```

1.2 Consultas simples sobre una tabla (SELECT FROM T WHERE)

1.- Obtener los datos de las películas con idioma original Español y con unos ingresos superiores a 15 millones.

```
select *
from peliculas
where idioma_original = 'es' and ingresos > 15000000
```

2.- Obtener el título, sinopsis y lema de las películas que tengan la palabra "Pirate" en su título. Ordena el resultado de forma descendente por presupuesto

```
select titulo, sinopsis, lema
from peliculas
where titulo like '%Pirate%'
order by presupuesto desc
```

3.- Muestra los títulos de las 10 películas con mayor presupuesto con una duración inferior a 60.

```
select titulo
from peliculas
where duracion < 60
order by presupuesto desc
limit 10
```

1.3 Consultas simples sobre varias tablas (SELECT FROM T1, T2, ... WHERE)

4.- Para las películas en las que actuó "Penélope Cruz", muestra el título original de la película y el personaje que interpretó. Ordena las películas por popularidad de forma descendente.

```
select pel.titulo_original, pr.personaje
from peliculas pel, pelicula_reparto pr, personas per
where pel.id = pr.pelicula
and pr.persona = per.id
and per.nombre = 'Penélope Cruz'
order by pel.popularidad desc
```

5.- Obtén un listado de las películas dirigidas por "Steven Spielberg" (el trabajo es "Director"). Para cada película muestra su título, presupuesto e ingresos. Ordena el listado por popularidad de forma descendente.

1.4 Agregados (AVG, MIN, MAX, SUM, COUNT, etc.)

6.- Obtén el máximo, mínimo y media del presupuesto e ingresos de las películas producidas en España. Muestra la media del presupuesto sin decimales.

```
select max(presupuesto) as maximo_presupuesto,
min(presupuesto) as minimo_presupuesto,
cast(avg(presupuesto) as integer) as media_presupuesto,
max(ingresos) as maximo_ingresos,
min(ingresos) as minimo_ingresos,
cast(avg(ingresos) as integer) as media_ingresos
from peliculas pel, pelicula_pais pp
where pel.id = pp.pelicula
and pp.pais = 'ES'
```

7.- Para las películas de la colección "Star Wars Collection", obtener el mínimo, máximo y media del beneficio obtenido (ingresos - presupuesto) y los coeficientes de correlación entre presupuesto e ingresos, entre la popularidad y el presupuesto e ingresos, y entre la fecha de emisión y el presupuesto, ingresos y popularidad. Nota: Ver como poder aplicar la correlación a una fecha, transformándola a un número.

1.5 Agregados sobre grupos (GROUP BY, HAVING)

8.- Obtén un informe en el que se muestre para cada película de la colección "Star Wars Collection" el número de personas que han trabajado en cada departamento.

```
select p.titulo, pp.departamento, count(distinct pp.persona) as personas
from peliculas p, colecciones c, pelicula_personal pp
where p.coleccion = c.id
and p.id = pp.pelicula
and c.nombre = 'Star Wars Collection'
group by p.titulo, pp.departamento
order by p.titulo, pp.departamento
```

9.- Para las diez productoras que más películas tengan, muestra el nombre de la productora, el número de películas, el número distinto de idiomas originales de las películas, a suma de presupuesto, la suma de ingresos, el beneficio (resta de ingresos - presupuesto) y la primera y última fecha de emisión.

10.- Obtener la lista de directores cuyas películas hayan generado más de 1500 millones de ingresos. Para cada director, obtener el número de películas que ha dirigido, el total de ingresos y presupuesto, y el beneficio total. Mostrar también el número de idiomas originales y la lista separada por comas de los códigos de los idiomas (usar string_agg). Ordena el resultado en orden descendente por beneficio.

```
select per.nombre as director,
       count(pel.id) as peliculas,
       sum(ingresos) as ingresos,
       sum(presupuesto) as presupuesto,
       sum(ingresos-presupuesto) as beneficio,
       count(distinct idioma_original) as num_idiomas,
       string_agg(distinct idioma_original, ',' ) as idiomas
from peliculas pel, pelicula_personal pp, personas per
where pel.id = pp.pelicula
  and per.id = pp.persona
  and pp.trabajo = 'Director'
group by per.id, per.nombre
having sum(pel.ingresos) > 1500000000
order by beneficio desc
```

11.- Para las películas producidas en España (ES), obtener la lista de actores/actrices que han participado en más de 15 películas distintas. Muestra el número de películas, la media de popularidad de las películas (con dos decimales) y ordena el resultado por este campo.

Ver otro tipo de expresiones más complejas con funciones de agregado incluyendo (<https://www.postgresql.org/docs/12/sql-expressions.html>):

- Ordenación y filtrado durante el cálculo de agregados.
- Funciones de ventana (window functions).

1.6 Subconsultas en cláusula FROM

12.- Para las 10 películas con más ingresos, muestra su título y el número de personas distintas del reparto, y los nombres separados por comas (ordenados por orden), el número de personas distintas del personal, y el número de departamentos y trabajos distintos de su personal. Muestra también la lista de sus directores (ordenados alfabéticamente).

```
select reparto.titulo, reparto.num_reparto, reparto.reparto,
       personal.num_personal, personal.departamentos, personal.trabajos, personal.directores
from
  (select pel.id as id, pel.titulo as titulo, pel.ingresos as ingresos,
         count(distinct pr.persona) as num_reparto,
         string_agg(per.nombre, ' ' order by pr.orden) as reparto
   from peliculas pel, pelicula_reparto pr, personas per
  where pel.id = pr.pelicula and pr.persona = per.id
  group by pel.id, pel.titulo) as reparto,
  (select pel.id as id,
         count(distinct pp.persona) as num_personal,
         count(distinct pp.departamento) as departamentos,
         count(distinct pp.trabajo) as trabajos,
         string_agg(per.nombre, ' ' order by per.nombre) filter (where pp.trabajo = 'Director') as directores
   from peliculas pel, pelicula_personal pp, personas per
  where pel.id = pp.pelicula and pp.persona = per.id
  group by pel.id) as personal
where reparto.id = personal.id
order by reparto.ingresos desc
limit 10
```

1.7 Subconsultas en cláusulas SELECT y WHERE

13.- Obtener el título, fecha de emisión y sinopsis de la película más reciente con un beneficio superior a 1000 millones.

```
select titulo, fecha_emision, sinopsis
from peliculas pel
where (ingresos - presupuesto) > 1000000000
  and fecha_emision = (select max(fecha_emision)
                      from peliculas pel1
                      where (ingresos - presupuesto) > 1000000000)
```

14.- Obtener el título original, presupuesto e ingresos de las 10 películas producidas solo en España con más ingresos. Obtener también los directores y el reparto.

1.8 Operaciones de conjuntos compatibles (UNION, INTERSECCIÓN y DIFERENCIA)

15.- Para las películas en las que participó "Quentin Tarantino", como actor o como miembro del personal, muestra el título, fecha de emisión, y trabajo (actor en el caso de que fuese miembro del reparto). Ordena el resultado por fecha de emisión.

```
select pel.titulo, pel.fecha_emision, 'Actor: ' || pr.personaje as trabajo
from peliculas pel, pelicula_reparto pr, personas per
where pel.id = pr.pelicula and pr.persona = per.id
and per.nombre = 'Quentin Tarantino'
union all
select pel.titulo, pel.fecha_emision, pp.trabajo as trabajo
from peliculas pel, pelicula_personal pp, personas per
where pel.id = pp.pelicula and pp.persona = per.id
and per.nombre = 'Quentin Tarantino'
order by fecha_emision, titulo
```

1.9 Operaciones de Join

16.- Para las 20 películas más populares del 2017 muestra el título, fecha de emisión, popularidad y colección.

```
select pels.titulo, pels.fecha_emision, pels.popularidad, c.nombre as coleccion
from peliculas pels, colecciones c
where pels.coleccion = c.id
and extract('year' from fecha_emision) = 2017
order by popularidad desc
limit 20
```

Las películas sin colección no salen en el resultado. Cambia la consulta de arriba para que aparezcan también las películas sin colección.

Ejercicios

Realiza las siguientes consultas sobre la base de datos de películas y entrega el código SQL generado para cada consulta en un único archivo de texto a través del campus virtual.

E1.- Obtén un listado ordenado de forma descendente por beneficios (ingresos - presupuesto) de las películas en idioma italiano (it) del año 1960. Para cada película muestra su título, su título original, la fecha de emisión y su sinopsis.

E2.- Obtener el reparto de la película "Titanic" de 1997. Para cada miembro del reparto mostrar el orden, el nombre del actor/actriz y el personaje. Mostrar el resultado ordenado por orden.

E3.- Obtener un listado de países, en el que se muestre para cada país: i) El número de películas producidas, ii) La cantidad de beneficios producida, iii) La media de popularidad de las películas (con dos cifras decimales). Ordena el listado de forma descendente por beneficios producidos.

E4.- Obtener un listado de personas que hayan dirigido y actuado en alguna película. Mostrar el número de películas dirigidas y las fechas de la primera y última. Mostrar también el número de películas en las que actuó, y la fecha de la primera y última. Ordena el resultado por el producto del número de películas dirigidas por el número de películas en la que actuó.

E5.- Para cada género, muestra la popularidad media de sus películas, la suma de beneficios, el beneficio por película medio, el título de la película que generó más beneficios y el actor/actriz principal (entre los 5 primeros en orden) y el director que han generado más beneficios. Ordena el resultado por popularidad media de forma descendente.

Última modificación: Luns, 6 de Setembro de 2021, 14:29

◀ Ejemplo HBase Java

Ir a...

Práctica 2: Datos Agregados en SQL ▶

Vostede accedeu como Andrés Campos Cuiña (Sair)
Bases de datos a grande escala [P4181101] [2021/2022]

Galego (gl)

Català (ca)

Dansk (da)

Deutsch (de)

English (en)

English (ja)

English (United States) (en_us)

Español - Internacional (es)

Euskara (eu)

Français (fr)
Galego (gl)
Italiano (it)
Nederlands (nl)
Norsk (no)
Polski (pl)
Português - Portugal (pt)
Română (ro)
Slovenščina (sl)
Svenska (sv)
Türkçe (tr)
Тоҷикӣ (tg)
Українська (uk)
עברית (he)
日本語 (ja_old)
正體中文 (zh_tw)
简体中文 (zh_cn)