



# Realización de Consultas

Bases de Datos



# Ordenación

## ➤ Clausula ORDER BY

Select name, job, date\_hired from employees.employees order by date\_hired;

Select name, job, date\_hired from employees.employees order by date\_hired asc;

Select name, job, date\_hired from employees.employees order by date\_hired desc;

Select name, sku, unit, unit\_price from sales.products order by unit\_price desc;

Select name, countrycode, population from world.city order by countrycode,  
population desc ;



# Ejemplos

- En la base de datos movielens, utilizando las tablas movie y movierating. Ordenar por rating y después por año. Descendente en los dos casos.
- En la base de datos movielens, utilizando las tablas movie, moviegenre y genre, ordenar por genero ascendente y por año descendente.
- En la base de datos world, utilizando las tablas country y city. Mostrar el nombre del país, la ciudad y la población ordenadas por país ascendente y población descendente
- En la base de datos sakila, utilizando las tablas film, film\_actor y actor, ordenar las películas y los actores de cada película alfabéticamente por película, nombre y apellido.



# Ejemplos



- Nombre de las ciudades con una población superior a 1.000.000, de los países donde la esperanza de vida es superior a 70 años ordenados de mayor a menor esperanza de vida.
- Lenguaje oficial de los países donde el producto interior bruto es mayor de 1.000.000 millones de USD ordenados de menor a mayor PIB
- Mostrar la lista de pagos del 19/08/2005 entre las 12 y las 13 horas, mostrando la cantidad, el nombre del cliente y su dirección. Ordenados de mayor a menor cantidad
- Mostrar los pagos de la tabla payment del 19 de Agosto de 2005 entre las 10:00 y las 10:30 mostrando el nombre del cliente, su dirección y el nombre del empleado que le ha atendido. Ordenado por pago ascendente y nombre del cliente alfabéticamente.

# Funciones y Agrupaciones

- Función COUNT() – Cuenta el número de registros en cada agrupación

Select count(\*) from city;

Select countrycode, count(\*) from city group by countrycode;

Select continent, count(\*) from country group by continent;

- Función MAX() – Valor máximo de cada agrupación

Select max(population) from city;

Select countrycode, max(population) from city group by countrycode;

Select continent, max(population) from country group by continent;

- Función MIN() – Valor mínimo de cada agrupación



# Funciones y Agrupaciones

- Función SUM() – Suma los valores en cada agrupación

Select sum(population) from city;

Select countrycode, sum(population) from city group by countrycode;

Select continent, sum(population) from country group by continent;

- Combinación de funciones

Select sum(population)/count(population) from city;

Select countrycode, sum(population)/count(population) from city group by countrycode;

Select continent, sum(population)/count(population) from country group by continent;



# Ejercicios Agrupaciones

- ▶ En la base de datos **employees**, utilizando la vista **emp\_v**, mostrar el número de empleados por departamento y el número de empleados por localización.
- ▶ En la base de datos **movielens**, utilizando las tablas **movie** y **movierating**, mostrar el rating medio de cada una de las películas ordenado de menor a mayor rating. Mostrar también el número de valoraciones o ratings por película de menor a mayor.
- ▶ En la base de datos **sales**, utilizando la tabla **customers**, mostrar el número de clientes agrupados por género.
- ▶ En la base de datos **sakila**, utilizando la tabla **film\_actor**, mostrar el número de actores por película y el número de películas que ha realizado cada actor. Incorporar la información de las tablas **actor** y **film**.



# Clausula HAVING

- Se utiliza de una manera similar a la clausula WHERE pero aplica a campos en los que se ha aplicado una función.

```
Select countrycode,count(name) from city group by countrycode having  
count(name)>10;
```

- La consulta cuenta el número de ciudades que hay por país pero solo muestra los países que tienen más de 100 ciudades. La siguiente consulta suma la población de las ciudades por país mostrando los países cuya suma da más de 30 millones.

```
Select countrycode,sum(population) from city group by countrycode having  
sum(population)>30000000;
```



# Ejercicios

- En la base de datos employees mostrar la media de salario por departamento mostrando solo los que tengan una media superior a 60000,  

```
select departments.name, avg(employees.salary) from employees, departments where  
departments.id = employees.department_id group by department_id having avg(salary)  
> 60000;
```
- En la base de datos world, calcular la población por continentes sin mostrar los continentes con una población por debajo de 400 millones.  

```
select continent, sum(population) from country group by continent having  
sum(population) > 400000000;
```
- En la base de datos movielens mostrar el número de películas que hay por cada género mostrando los géneros que tienen más de 200 películas.  

```
select genre.name, count(moviegenre.movieid) from genre, moviegenre where genre.id  
= moviegenre.genreid group by genre.name having count(moviegenre.movieid) > 200;
```



# Ejercicios

- En la BBDD movielens, utilizando las tablas movie y movierating, mostrar la valoración media de las películas que cuentan con más de 2.000 valoraciones y que, su valoración media, es mayor que 4.
- En la BBDD logistics, utilizando las tablas drivers y timesheet, mostrar la suma de horas y millas por conductor, solo los conductores que suman más de 3000 horas.
- En la BBDD sakila, utilizando las tablas customer\_list y payment, sumar los pagos de cada cliente (amount) mostrando solo los clientes cuya suma de pagos sea superior a 160
- En la BBDD movielens, usando las tablas user y occupation, mostrar cuantos usuarios hay por ocupación. Solo las ocupaciones con más de 400 usuarios.



# Subconsultas

- Utiliza, en la clausula WHERE, el resultado de otra consulta  
`Select name, population from city where countrycode in (select countrycode from city where population > 8000000);`
- Busca todas las ciudades de los países con ciudades de más de 800000 de habitantes
- Aunque no es obligatorio, se suelen utilizar cuando, en la condición, se utiliza una consulta a la misma tabla.



# Ejercicios Subconsultas

- En la base de datos sales, tabla products, mostrar los productos que tienen un precio unitario superior al precio medio de los productos.
- En la base de datos sakila, utilizando las tablas payment y customer, averiguar quienes son los clientes que realizan los mayores pagos. Considerar los mayores pagos un 80% del pago máximo.
- En la base de datos world, utilizando la tabla city, mostrar las ciudades con una población un 20% alrededor de la población media de las ciudades.
- En la base de datos employees averiguar los empleados que ganan un 70% de la media y en qué departamentos trabajan.

# Subconsultas con dos campos

- Lista de las ciudades que tienen la mayor población en cada país. Con una población de más de 8 millones de habitantes

```
select name, population from city where (countrycode,population) in (select  
countrycode, max(population) from city group by countrycode having  
max(population)>8000000);
```

- En la base de datos de empleados mostrar, de cada departamento, el empleado que menos gana

```
select departments.name, employees.name, employees.salary from departments,  
employees where departments.id = employees.department_id and  
(employees.department_id,employees.salary) in (select department_id, min(salary)  
from employees group by department_id);
```



# Subconsultas de dos campos

- En la base de datos world averiguar, dentro de cada continente, el país más poblado. Descontar los casos en los que el máximo de 0

```
select continent, name, population from country where (continent,population) in  
(select continent, max(population) from country group by continent having  
max(population) > 0);
```

- En la base de datos sakila, en la tabla film, mostrar, de cada clasificación (rating) la película de mayor duración.

```
select title, rating, length from film where (rating,length) in (select rating,  
max(length) from film group by rating);
```





GRACIAS