



Comandos SQL

Bases de Datos Relacionales

Relaciones con dos Tablas Base

```
mysql> select city.name, city.population, country.name, country.population
-> from city, country
-> where city.countrycode = country.code
-> and city.population > 8000000;
```

```
mysql> select city.name, city.population, country.name, country.population
-> from city, country
-> where city.countrycode = country.code
-> and city.population > 8000000;
```

name	population	name	population
SÃo Paulo	9968485	Brazil	170115000
Jakarta	9604900	Indonesia	212107000
Mumbai (Bombay)	10500000	India	1013662000
Shanghai	9696300	China	1277558000
Seoul	9981619	South Korea	46844000
Ciudad de MÃxico	8591309	Mexico	98881000
Karachi	9269265	Pakistan	156483000
Istanbul	8787958	Turkey	66591000
Moscow	8389200	Russian Federation	146934000
New York	8008278	United States	278357000

10 rows in set (0.01 sec)



Field
Code
Name
Continent
Region
SurfaceArea
IndepYear
Population
LifeExpectancy
GNP
GNPold
LocalName
GovernmentForm
HeadOfState
Capital
Code2

Field
ID
Name
CountryCode
District
Population



Opciones

```
mysql>select city.name, city.population, country.name, country.population
```

```
->from city, country
```

```
->where city.countrycode = country.code
```

```
->and city.population > 8000000;
```

```
mysql>select a.name, a.population, b.name, b.population
```

```
->from city a, country b
```

```
->where a.countrycode = b.code
```

```
->and a.population > 8000000;
```



Función JOIN

- Permite opciones adicionales

```
mysql>select city.name, city.population, country.name, country.population
```

```
->from city join country
```

```
->on (city.countrycode = country.code)
```

```
->where city.population > 8000000;
```

```
mysql>select a.name, a.population, b.name, b.population
```

```
->from city a join country b
```

```
->on (a.countrycode = b.code)
```

```
->where a.population > 8000000;
```

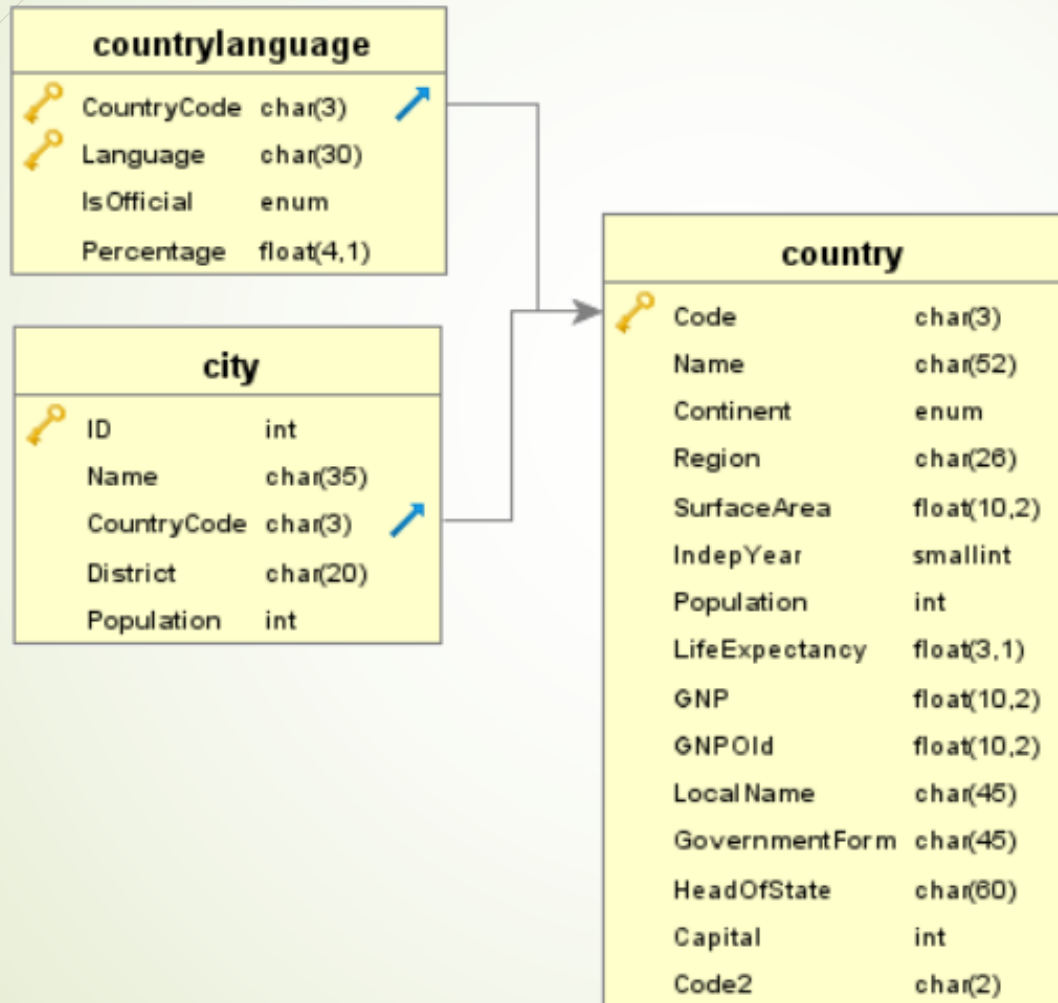


Opciones JOIN

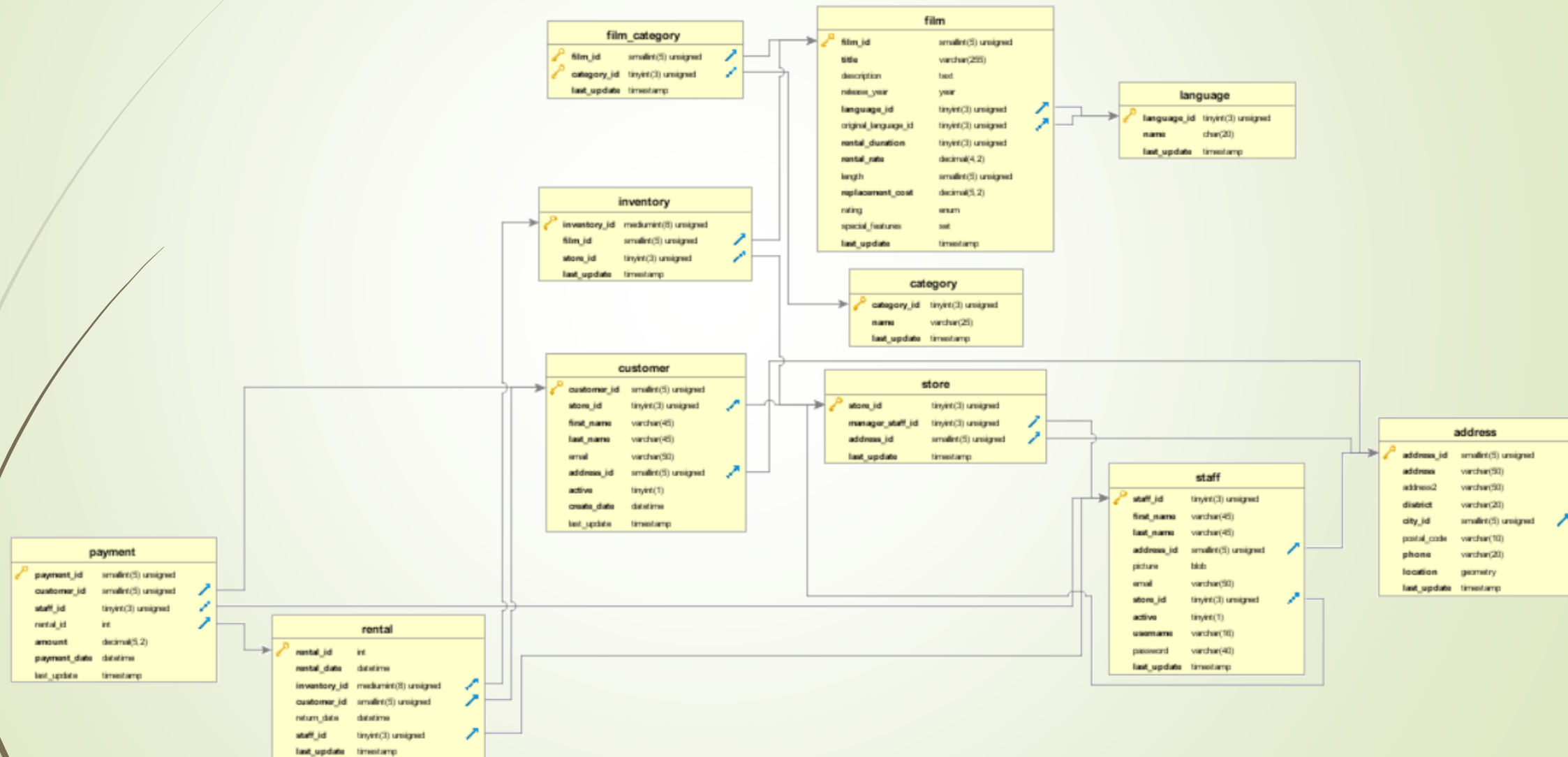


- JOIN, CROSS JOIN, INNER JOIN
 - Es el habitual. Solo aparecen los registros de una tabla que tienen su correspondiente en la otra tabla
 - Es similar a los comandos SELECT con dos tablas
- LEFT JOIN
 - Se muestran también los registros de la primera tabla que NO tienen su correspondiente en la segunda tabla.
 - RIGHT JOIN análogo a LEFT JOIN. Se muestran los registros de la segunda tabla.
- OUTER JOIN
 - Se muestran todos los registros aunque no haya correspondencia.

Base de Datos World



Base de Datos Sakila





Ejercicios Consultas con Dos Tablas

- Nombre de las ciudades con una población superior a 1.000.000, de los países donde la esperanza de vida es superior a 80 años.
- Lenguaje oficial de los países donde el producto interior bruto es mayor de 8.000.000 millones de USD
- Ciudades con una población mayor de 1.000.000, de los países donde se habla Inglés.
- Comparar la tabla **customer** y **customer_list** mostrando el nombre del cliente, su email y el país. Mostrar solo los que no están activos.
- 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.



Ejercicios Consultas de Dos Tablas

- Nombre de las capitales de los países con un GNP superior a 500.000 millones de euros.
- Utilizando las tablas **film** y **film_list** mostrar el título, las características especiales y la duración de las películas con una duración de más de 180 minutos.
- Utilizando las tablas **film** e **inventory** mostrar cuantas películas con el título 'ZORRO ARK' hay en cada tienda y el número de inventario.
- Utilizando las tablas **film** y **language** averiguar el idioma de las películas con una duración inferior a 50 minutos.
- Utilizando las tablas **rental** y **customer_list** mostrar los nombres de los clientes de España y las fechas de sus alquileres.

Consultas de tres tablas

- En la base de datos sakila, utilizando las tablas **customer**, **address** y **city**, mostrar el nombre, la dirección y la ciudad de los clientes que viven en una avenida

```
mysql> select customer.first_name, customer.last_name, address.address, city.city  
from customer, address, city where customer.address_id = address.address_id and  
address.city_id= city.city_id and address.address like '%avenue%';
```

- Utilizando la función JOIN

```
mysql> select customer.first_name, customer.last_name, address.address, city.city  
from customer join address on customer.address_id = address.address_id join city on  
address.city_id= city.city_id where address.address like '%avenue%';
```

Ejercicios Consultas con Tres Tablas

➤ Base de datos **World**

- Mostrar el nombre de las ciudades cuya población sea superior a 7 millón de habitantes mostrando también el nombre del país y el idioma oficial del país.

➤ Base de datos **Sakila**

- Mostrar el título de las películas que contengan la palabra **sea** (mar en inglés) de la tabla **film** junto con el nombre de la categoría a la que pertenecen de la tabla **category**. Utilizar la tabla **film_category** como nexo de unión
- 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.



Ejercicios Relaciones

- Base de datos **world**
 - Tipo de relación **city-country**
 - Tipo de relación **city-countrylanguage**
 - Tipo de relación **country-countrylanguage**
- Base de datos **sakila**
 - Tipo de relación **actor-actor_info**
 - Tipo de relación **customer-customer_list**
 - Tipo de relación **rental-customer**
 - Tipo de relación **customer-address**
 - Tipo de relación **film-language**



Ejercicios Vistas y Tablas Temporales

- Base de datos **world**
 - Crear una tabla temporal denominada **cityplus** con todos los campos de las tablas **city** y **country**
 - Crear una vista denominada cityview con todos los campos de las tablas **city** y **countrylanguage**
- Base de datos **sakila**
 - Crear una tabla temporal



GRACIAS