**Consulta**

**¿Qué relación tienen los datos maestros con las dimensiones en un modelo multidimensional?**

* Las tablas maestras representas las dimensiones en una base de datos multidimensional, en donde cada tabla está compuesta por datos maestros, los cuales son los datos más importantes de la organización.

**¿Que agregaciones se podrían hacer en la tabla countries resultado que se obtuvo en el ejercicio en lenguaje SQL?**

* Cuantos países hay por cada región.

SELECT COUNT(name), region

FROM sakila.country2

GROUP BY region;

* La subregión que más países tiene.

SELECT subregion, count(name) as max

FROM sakila.country2

GROUP BY subregion

ORDER BY max DESC

LIMIT 1;

* La región con más población.

SELECT region, MAX(population) max

FROM sakila.country2

GROUP BY region

ORDER BY max DESC

LIMIT 1;

* Cantidad de clientes por país.

SELECT count(customer.customer\_id) as cant, country2.name

FROM sakila.customer

INNER JOIN sakila.address ON customer.address\_id = address.address\_id

INNER JOIN sakila.city ON address.city\_id = city.city\_id

INNER JOIN sakila.country2 ON city.country\_id = country2.country\_id

GROUP BY country2.name

ORDER BY cant DESC;

* País con más alquileres.

SELECT count(rental.rental\_id) as c, country2.name

FROM sakila.rental

INNER JOIN sakila.customer ON rental.customer\_id = customer.customer\_id

INNER JOIN sakila.address ON customer.address\_id = address.address\_id

INNER JOIN sakila.city ON address.city\_id = city.city\_id

INNER JOIN sakila.country2 ON city.country\_id = country2.country\_id

GROUP BY country2.name

ORDER BY c DESC

LIMIT 1;

* Cantidad de veces que ha sido rentada una categoría por país.

SELECT category.name, COUNT(category.category\_id) as max , country2.name

FROM sakila.category

INNER JOIN sakila.film\_category ON category.category\_id = film\_category.category\_id

INNER JOIN sakila.film ON film\_category.film\_id = film.film\_id

INNER JOIN sakila.inventory ON film.film\_id = inventory.film\_id

INNER JOIN sakila.store ON inventory.store\_id = store.store\_id

INNER JOIN sakila.address ON store.address\_id = address.address\_id

INNER JOIN sakila.city ON address.city\_id = city.city\_id

INNER JOIN sakila.country2 ON city.country\_id = country2.country\_id

GROUP BY category.name, country2.name

ORDER BY country2.name DESC;

* Cantidad de películas rentadas por País

SELECT COUNT(rental.rental\_id), c2.name

FROM sakila.rental

INNER JOIN sakila.customer ON customer.customer\_id = rental.customer\_id

INNER JOIN sakila.address as ad ON ad.address\_id = customer.address\_id

INNEr JOIN sakila.city as c ON c.city\_id = ad.city\_id

INNER JOIN sakila.country2 as c2 ON c2.country\_id = c.country\_id

GROUP BY c2.name;

**¿Que vistas podría tener en Sakila con la nueva tabla de countries?**

* El top 5 de categorías que más alquilan.

CREATE VIEW count\_category AS

SELECT category.name

FROM sakila.category

INNER JOIN film\_category ON category.category\_id = film\_category.category\_id

INNER JOIN film ON film.film\_id = film\_category.film\_id

INNER JOIN inventory ON inventory.film\_id = film.film\_id

INNER JOIN rental ON rental.inventory\_id = inventory.inventory\_id

GROUP BY category.name

ORDER BY count(category.category\_id) DESC

LIMIT 5;

* Trabajadores por tienda.

CREATE OR REPLACE VIEW TrabajadoresPorTienda AS

SELECT CONCAT(staff.first\_name, " ", staff.last\_name), address.address

FROM sakila.staff

INNER JOIN sakila.store ON staff.store\_id = store.store\_id

INNER JOIN sakila.address ON store.address\_id = address.address\_id;

* Cuantas películas de cada categoría están en inventario.

CREATE OR REPLACE VIEW categoryInventory AS

SELECT category.name, COUNT(category.category\_id)

FROM sakila.category

INNER JOIN film\_category ON category.category\_id = film\_category.category\_id

INNER JOIN film ON film.film\_id = film\_category.film\_id

INNER JOIN inventory ON inventory.film\_id = film.film\_id

GROUP BY category.name;

* Clientes que aun no han retornado la película.

CREATE OR REPLACE VIEW deben AS

SELECT CONCAT(customer.first\_name, " ", customer.last\_name), film.title

FROM sakila.customer

INNER JOIN sakila.rental ON customer.customer\_id = rental.customer\_id

INNER JOIN sakila.inventory ON rental.inventory\_id = inventory.inventory\_id

INNER JOIN sakila.film ON inventory.film\_id = film.film\_id

WHERE rental.return\_date > '2005-05-30';

show create view deben;