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;
```