

## Evaluación SQL

Candidato: Axel Andrei Zetina Cuevas

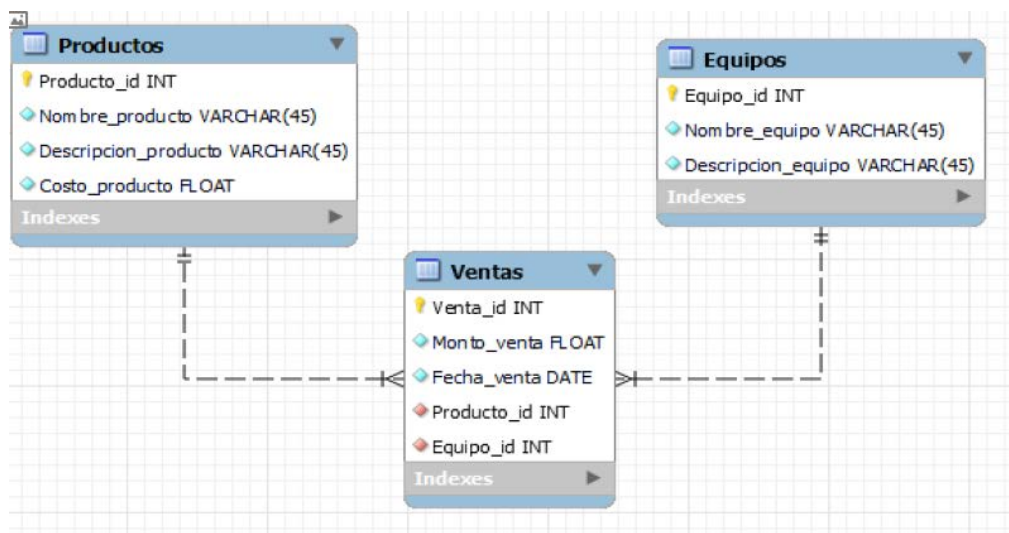
Date: 25/02/2024

- I. Para la siguiente necesidad de información, diseñe un diagrama ilustrando las tablas que mejor responden a las preguntas del negocio.

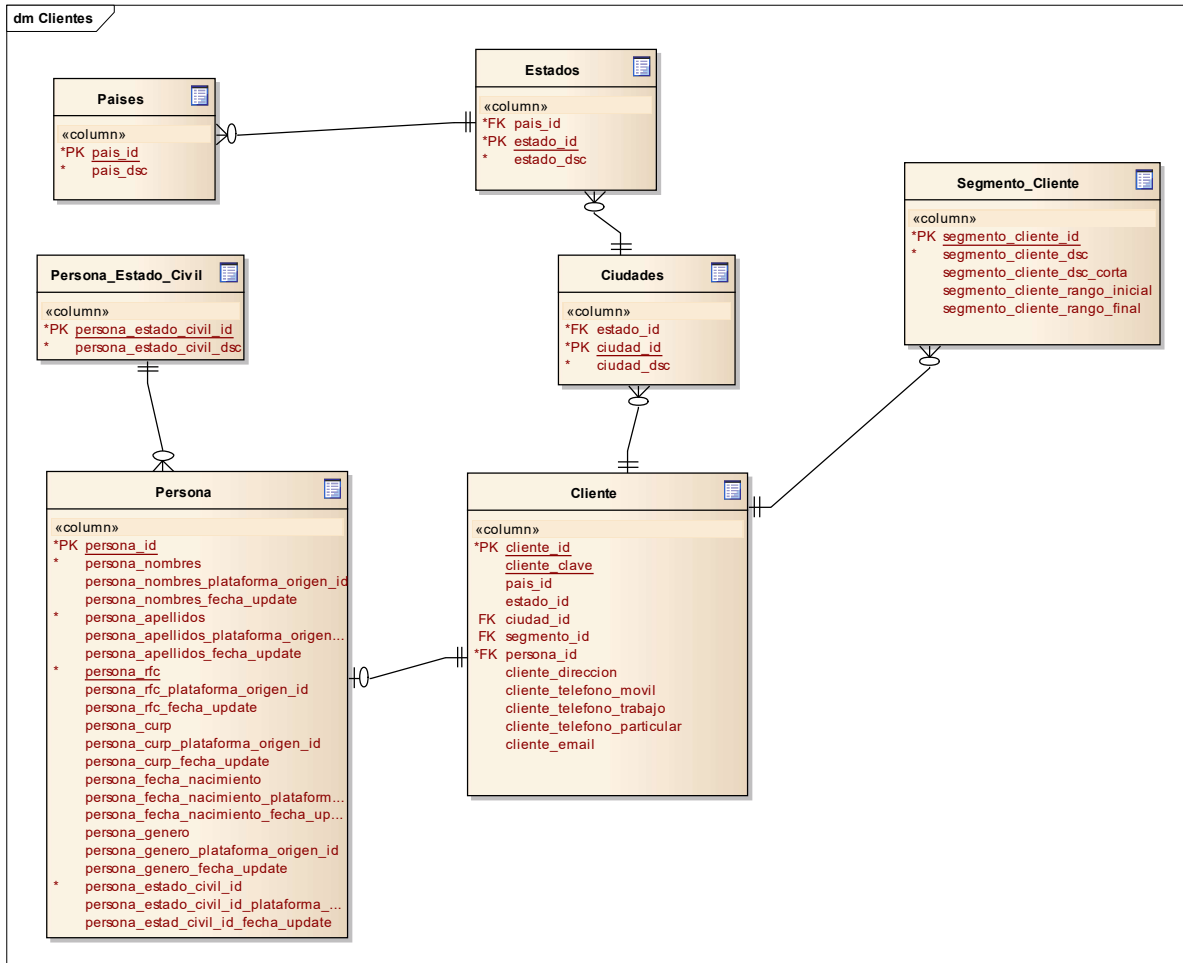
Skandia, dentro de su portafolio de productos, ofrece varios productos de inversión y los productos de inversión son puestos a disposición del público a través de diversos equipos de venta (integrados por varios vendedores).

La compañía requerirá monitorear en el futuro el total de ventas por producto, para identificar entre los mejores y que deben ser proyectados como los productos estrella de la compañía y aquellos con niveles de venta bajos, que son candidatos a desaparecer.

Además, la compañía también requiere evaluar a sus equipos de venta (individualmente por cada vendedor o a nivel total por cada equipo de ventas), para monitorear su productividad.



II. Considere el diagrama siguiente.



Indique cuál es el resultado del siguiente query.

```
SELECT persona_id, persona_nombres, 'Skip Columns', persona_apellidos, 'Rollback persona_apellidos'
```

```
FROM Persona P
```

```
WHERE Persona_Id NOT IN (
```

```
SELECT Persona_Id
FROM Persona P
INNER JOIN Persona_Estado_Civil PEC
    ON P.Persona_Estado_Civil_Id = PEC.Persona_Estado_Civil_Id
WHERE PEC.Persona_Estado_Civil_dsc = 'casado'
INTERSECT
SELECT Persona_Id
FROM Persona P, Segmento_Cliente
INNER JOIN Cliente C
    ON P.Persona_Id = C.Persona_Id
WHERE C.segmento_Id = Segmento_Cliente.Segmento_Id
AND Segmento_Cliente_Dsc NOT LIKE '%Norte%'
```

)

ORDER BY 4

Busca devolver un conjunto de resultados basado en la tabla persona y establece ciertas condiciones. El resultado de esta consulta será un conjunto de filas de la tabla persona que están ordenadas por el apellido de la persona. Las columnas 'Skip Columns' y 'Rollback persona\_apellidos' son literales de cadena y se agregarán a la salida sin ningún valor real de la tabla.

III. Escriba el query necesario para responder a la siguiente necesidad.

Como parte de la estrategia de servicio de Skandia, se necesita identificar los estados con más clientes con el objetivo de abrir nuevas oficinas de atención directa y resolución de dudas.

La función de Servicio requiere un listado con dos columnas donde se listen todos los estados y el total de clientes en cada una de ellas. Como la función de marketing se enfocará en los cinco estados con más clientes requiere ver el resultado en orden descendente de acuerdo al número de clientes.

En la última fila del mismo reporte necesitan el total de clientes (de todos los estados).

Además de abrir nuevas oficinas de servicio se abrirán oficinas de venta en los estados que aún no tienen clientes, por lo que los estados sin clientes también deben aparecer en el reporte.

```
SELECT Estado, Total_Clientes -- Mostrar las dos columnas mencionadas
FROM (
SELECT E.estado_dsc AS Estado, COUNT(C.cliente_id) AS Total_Clientes
FROM Estados E
LEFT JOIN Cliente C ON E.estado_id = C.estado_id
GROUP BY E.estado_dsc -- Combina la tabla estados con cliente y permite
contar el número de clientes en cada estado incluyendo a los que no tienen clientes.
UNION
SELECT 'Total' AS Estado, COUNT(cliente_id) AS Total_Clientes
FROM Cliente
) AS CombinedData -- Calcula el total de clientes en todos los estados.
ORDER BY Total_Clientes DESC, Estado; -- Ordeno el conjunto de
resultados en orden descendente por el número de clientes y, en caso de empate,
en orden ascendente por el nombre del estado.
```

- IV. Escriba el query necesario para reasignar de segmento a los clientes que actualmente están ubicados en segmentos de la zona sur. Anteriormente se tenían centros de venta en diversas zonas sur (sur oriente, sur poniente, sur 1, sur 5, etc.) y todos han sido concentrados en la zona “Sur”.

```
UPDATE Cliente
SET segmento_id = (SELECT segmento_cliente_id FROM Segmento_Cliente
WHERE segmento_cliente_dsc = 'Sur')
WHERE segmento_id IN (
    SELECT segmento_cliente_id
    FROM Segmento_Cliente
    WHERE segmento_cliente_dsc LIKE 'Sur%'
); -- Se utiliza una subconsulta para obtener el ID del nuevo segmento
llamado "Sur" desde la tabla Segmento_Cliente.
```

- V. Se desea completar el catálogo de ciudades de tal forma que los clientes (futuros) que pertenezcan a una que actualmente no esté registrada sean agrupados en una ciudad “default”. La solución decidida es ingresar la ciudad “Otra” en cada uno de los estados. Escriba el query necesario para insertar la ciudad “Otra” en todos los estados registrados. Este query será entregado al equipo de Producción de Tecnología de tal manera que puedan copiar y pegar y ejecutar.

```
-- Iniciar una transacción
START TRANSACTION;
-- Insertar la ciudad "Otra" en todos los estados registrados
INSERT INTO Ciudades (estado_id, ciudad_dsc)
SELECT E.estado_id, 'Otra'
FROM Estados E
WHERE E.estado_id NOT IN (SELECT DISTINCT estado_id
FROM Ciudades);
-- Confirmar la transacción si todo es exitoso
COMMIT;
-- Revertir la transacción si ocurren errores
ROLLBACK;
```

VI. De acuerdo al esquema presentado, ¿puede ejecutarse la siguiente instrucción? Explique su respuesta.

`DELETE FROM Ciudades`

Si la tabla ciudades no tiene registros, al ejecutar la instrucción eliminará cualquier fila de la tabla (independientemente si tiene datos o no). No habrá problemas en ese caso.

Por otra parte, si la tabla tiene registros y existe una relación de clave externa con otras, entonces ejecutar la instrucción podría provocar errores si estas restricciones impiden la eliminación de filas relacionadas.