



Prueba de Conocimiento

Instrucciones y entregables

Resuelva y desarrolle lo que se pide en cada parte de esta prueba. Debe incluir no sólo el resultado sino también el código utilizado para crearlo. Este código debe estar claramente comentado para documentar lo que cada parte hace.

Debe **entregar un documento** donde explique los pasos y comentarios relevantes de cada parte. La idea es que explique qué decisiones tomó y cuáles son sus argumentos para haberlas tomado.

Así mismo debe incluir todos los scripts que realice, comandos de consola y APIs debe entregarlos documentados y en los formatos adecuados para ello (por ejemplo los archivos Python, en archivos .py). Así mismo la ruta de la API pública construida.

Parte 1:

Debe cargar las tablas incluídas en esta prueba en una base de datos **local** de su elección y a continuación desarrollar lo que se indica en los siguientes incisos. Las soluciones deben de evitar usar cursores, procedimientos almacenados o similares; se pretende que las soluciones se hagan en SQL estándar. **Para cada inciso, debe incluir la tabla resultante en archivo csv o excel.**

Inciso 1:

Utilizando las tablas `SalesOrderHeader` y `SalesTerritory` escriba una consulta para calcular la cantidad de transacciones y monto total mensual por territorio para cada uno de los estados según el campo `Status`. Los valores de `Status` son: 1 = In process ; 2 = Approved ; 3 = Backordered ; 4 = Rejected ; 5 = Shipped ; 6 = Cancelled

La tabla de resultado deberá contener las siguientes campos:

1. Mes : mes en formato YYYY-MM
2. NombreTerritorio : Nombre del Territorio

3. TrProceso : Transacciones en proceso (Cantidad)
4. TrAprobadas : Transacciones Aprobadas (Cantidad)
5. TrAtrasadas : Transacciones Atrasadas (Cantidad)
6. TrRechazadas : Transacciones Rechazadas (Cantidad)
7. TrEnviadas : Transacciones Enviadas (Cantidad)
8. TrCanceladas : Transacciones Canceladas (Cantidad)
9. MntProceso : Monto total en proceso
10. MntAprobadas : Monto total Aprobadas
11. MntAtrasadas : Monto total Atrasadas
12. MntRechazadas : Monto total Rechazadas
13. MntEnviadas : Monto total Enviadas
14. MntCanceladas : Monto total Canceladas

Inciso 2:

Utilizando las tablas `SalesOrderDetail` , `SalesOrderHeader` , `Product` , `Customer` y `Person` escriba una consulta que liste los *nombres* de los top 3 productos por cantidad de unidades despachadas y los *nombres* de los top 3 clientes por monto total facturado. Utilice `SalesOrderHeader.OrderDate` como fecha de referencia y debe considerar solo las órdenes despachadas según `Status` (ver inciso anterior).

1. Mes : mes en formato `YYYY-MM`
2. Producto_N1 : Producto con mayor número unidades despachadas en el mes
3. Producto_N2 : Producto con 2do. mayor número unidades despachadas en el mes
4. Producto_N3 : Producto con 3er. mayor número unidades despachadas en el mes
5. Cliente_N1 : Cliente con mayor facturación despachada en el mes
6. Cliente_N2 : Cliente con 2da. mayor facturación despachada en el mes
7. Cliente_N3 : Cliente con 3era. mayor facturación despachada en el mes

Inciso 3:

Utilizando las tablas `SalesOrderHeader` y `SalesTerritory` , escriba una consulta que muestre la comparación entre el promedio mensual (según fecha `SalesOrderHeader.OrderDate`) del subtotal de ventas (`SalesOrderHeader.SubTotal`) de toda la organización vs el promedio mensual de los *grupos territoriales* (`SalesTerritory.Group`) para los ultimos 6 meses contenidos en la tabla. El resultado debe contener las siguientes columnas:

1. OrderMonth : mes, incluyendo año, de las ordenes en formato `YYYY-MM`
2. TerritoryGroup : Grupo territorial
3. MonthAvg : Promedio mensual del subtotal de ventas de toda la organización

4. `Diff` : Diferencia del promedio grupal vs el general (`GroupMonthAvg` - `MonthAvg`)
5. `Indicator` : "Above" si el promedio grupal esta por encima del promedio total, "Below" si esta por debajo y "Equal" si es igual.

Inciso 4:

Escribe una consulta SQL para encontrar todos los valores del campo `codigo` que aparecen *por lo menos* tres veces consecutivas según el ordenamiento del campo `secuencial` de la tabla `Secuencial.csv` . La consulta debe generar una lista con los códigos que cumplen con la condición indicada.

Inciso 5:

Utilizando las tablas `SalesOrderHeader` , `Customer` y `Person` escriba una consulta que extraiga los nombres de los clientes que realizaron una compra en el mes y que además incluya el cálculo de los días desde la penúltima compra y desde la antepenúltima compra.

La tabla entonces debe contener los siguientes campos:

1. `Mes` : mes en formato `YYYY-MM`
2. `NombreCliente` : Nombre del Cliente
3. `FechaUltCompra` : Fecha de la última compra
4. `DiasUC_PC` : días entre la última compra y la penúltima compra
5. `DiasPC_APC` : días entre la penúltima compra y la antepenúltima compra

Parte 2:

Debe realizar un script de `python` programado de forma tal que pueda recibir los siguientes parámetros:

1. Referencia a la base de datos origen
2. un archivo de texto con una consulta SQL (`.sql`) a ejecutar
3. Referencia a la base de datos/tabla destino para almacenar el resultado de la misma
4. Nombre y ubicación del archivo para registrar una bitácora de ejecución `.log`

Debe probar la funcionalidad de este script al ejecutarlo con cualquiera de las consultas de la **Parte 1** en un archivo de texto `.sql` y validar que el resultado quede guardado en una tabla resultado en la misma base de datos local que usó para el desarrollo de la parte anterior, configurandola dentro de los parámetros, y también verificar que se generó el log de ejecución en la ruta especificada.

Parte 3:

En el archivo `datos_base_clientes.csv` se encuentra una exportación de datos de información de consumos de clientes. En el archivo `metadatos_clientes.xlsx` se encuentra la explicación de los datos que estamos adjuntando. Estos datos normalmente se encontrarían en arquitecturas de BigData (Hadoop/Impala) dado su tamaño real, sin embargo para motivos de la prueba trabajará con un set de datos reducido. A continuación los pasos de lo que usted debería realizar y entregar.

1. Debe configurar una cuenta gratis cualquier nube (mientras sea AWS, Azure o GoogleCloud).
Todas las nubes ofrecen capacidades gratis y no tendrá que incurrir en costos para este ejercicio.
2. Debe realizar un script que envíe los datos desde su almacenamiento local a un almacenamiento NoSQL proporcionado por la nube. Este script debe ser lo más genérico posible para que en el caso que un futuro se necesitase subir otros datos reutilizar el script sea relativamente fácil.

Extra Puntos: Debe programar un API a manera de exponer estos datos. Básicamente lo que se desea es consultar la información completa de un cliente a través de un API en donde le enviemos la llave del cliente como parámetro y nos devuelva los datos correspondientes a ese cliente.

Extra Extra Puntos: Haga un “FrontEnd” sencillo que consuma la api y muestre los datos con **visualizaciones gráficas**.