Ejercicios Debezium

Instrucciones para la entrega de ejercicios:

Los ejercicios deben ser entregados en un documento de Word que debe incluir, para cada una de las acciones realizadas en cada ejercicio:

• Una explicación acompañada de imágenes de las modificaciones realizadas en la base de datos.

• Una explicación e imágenes de los mensajes generados, describiendo lo ocurrido.

Importante: La fecha límite para la entrega de los ejercicios es el 12 de noviembre, durante las horas lectvas del módulo.

Schema de Base de Datos Inventory

Ejercicio 1

En la tabla Customers debes realizar las siguientes acciones:

- Añade un nuevo registro a la tabla

```
-- Añade un nuevo registro a la tabla
INSERT INTO customers(id, first_name, last_name, email) VALUES
(1005, 'Samuel', 'Arteaga', 'samu@gmail.com')
```

Consulta ejecutada, 1 registro afectado. (0.026 s) Modificar

```
SELECT *
FROM customers
```

id	first_name	last_name	email
1001	Sally 4444	Thomas	sally.thomas@acme.com
1002	George	Bailey	gbailey@foobar.com
1003	Edward	Walker	ed@walker.com
1004	Anne	Kretchmar	annek@noanswer.org
1005	Samuel	Arteaga	samu@gmail.com

- Modificar un registro de la tabla

```
-- Añade un nuevo registro a la tabla
-- INSERT INTO customers(id, first_name, last_name, email) VALUES
-- (1005, 'Samuel', 'Arteaga', 'samu@gmail.com');
```

```
-- Modificar un registro de la tabla
UPDATE customers
SET  last_name = 'Lopez'
WHERE last_name = 'Arteaga'
```

Consulta ejecutada, 1 registro afectado. (0.028 s) Modificar

```
SELECT *
FROM customers
```

id	first_name	last_name	email
1001	Sally 4444	Thomas	sally.thomas@acme.com
1002	George	Bailey	gbailey@foobar.com
1003	Edward	Walker	ed@walker.com
1004	Anne	Kretchmar	annek@noanswer.org
1005	Samuel	Lopez	samu@gmail.com

- Borra un campo de la tabla

```
-- Borra un campo de la tabla
ALTER TABLE customers
DROP COLUMN email
```

Consulta ejecutada, 0 registros

```
SELECT *
FROM customers
```

id	first_name	last_name
1001	Sally 4444	Thomas
1002	George	Bailey
1003	Edward	Walker
1004	Anne	Kretchmar
1005	Samuel	Lopez

Ejercicio 2

Crea una nueva tabla debes realizar las siguientes acciones:

```
CREATE TABLE customers2(
id INT PRIMARY KEY,
nombre VARCHAR(255) NOT NULL,
apellido1 VARCHAR(255) NOT NULL,
apellido2 VARCHAR(255)
)
```

Consulta ejecutada, 0 registros afectados. (0.129 s) Modificar

- Añade registros a la nueva tabla.

```
-- Añade registros a la nueva tabla.
INSERT INTO customers2(id, nombre, apellido1, apellido2) VALUES
(1,'Samuel','Arteaga','Lopez'),
(2,'Lucas','Tarazona','Comendador'),
(3,'Roberto','Ripoll','Cones')
```

Consulta ejecutada, 3 registros afectados. (0.026 s) Modif

```
SELECT *
FROM customers2
```

id	nombre	apellido1	apellido2
1	Samuel	Arteaga	Lopez
2	Lucas	Tarazona	Comendador
3	Roberto	Ripoll	Cones

- Modifica registros de la nueva tabla.

```
-- Modifica registros de la nueva tabla.
UPDATE customers2
SET apellido2 = 'Perez'
WHERE id = 1
```

Consulta ejecutada, 1 registro afectado.

```
SELECT *
FROM customers2
```

id	nombre	apellido1	apellido2
1	Samuel	Arteaga	Perez
2	Lucas	Tarazona	Comendador
3	Roberto	Ripoll	Cones

- Borra registros de la nueva tabla.

```
-- Borra registros de la nueva tabla.
DELETE FROM customers2
WHERE id = 3
```

Consulta ejecutada, 1 registro afect

```
SELECT *
FROM customers2
```

id	nombre	apellido1	apellido2
1	Samuel	Arteaga	Perez
2	Lucas	Tarazona	Comendador

- Elimina campos de la nueva tabla

```
SELECT *
FROM customers2
```

id	nombre
1	Samuel
2	Lucas

2 registros (0.002 s) Mo

```
-- Agrega un nuevo campo
```

Consulta ejecutada,

```
-- Elimina campos de la

ALTER TABLE customers2
DROP COLUMN apellido1;

ALTER TABLE customers2
DROP COLUMN apellido2;

SELECT *
FROM customers2;
```

- Agrega un nuevo campo a la nueva tabla.

```
ALTER TABLE customers2 ADD COLUMN edad INT
```

Consulta ejecutada, 0 registros afectados.

```
SELECT *
FROM customers2
```

id	nombre	edad
1	Samuel	*NULL*
2	Lucas	*NULL*

Ejercicio 3

Crea una nueva tabla debes realizar las siguientes acciones:

```
USE inventory;


CREATE TABLE ejercicio3(
id INT PRIMARY KEY,
nombre VARCHAR(255) NOT NULL,
edad INT NOT NULL
);
```

- Modifica el conector (register-mysql.json) para que sólo recoja mensajes de esta tabla.

```
    "table.include.list": "inventory.ejercicio3",
```

esta linea del json se ha usado para que única y exclusivamente consuma de esa tabla

```
"snapshot.mode": "when_needed",
    "snapshot.locking.mode": "minimal",
    "snapshot.fetch.size": "1000"
```

estas tres lineas se han añadido al json porque daba un error de conexión con kafka

- Añade registros a la nueva tabla.

```
{ 2 items
  "schema" : { 5 items
    "type" : "struct"
    "fields" : [ 6 items
      0 : { 5 items
        "type" : "struct"
        "fields" : [ 3 items
          0 : { 3 items
            "type" : "int32"
            "optional" : false
            "field" : "id"
          }
          1 : { 3 items
            "type" : "string"
            "optional" : false
            "field" : "nombre"
          }
          2 : { 3 items
            "type" : "int32"
            "optional" : false
            "field" : "edad"
          }
        ]
```


- Modifica registros de la nueva tabla.

```
{ 2 items
  "schema" : { 5 items
    "type" : "struct"
    "fields" : [...] 6 items
    "optional" : false
    "name" : "dbserver1.inventory.ejercicio3.Envelope"
    "version" : 1
  }
  "payload" : { 6 items
    "before" : { 3 items
      "id" : 3
      "nombre" : "Roberto"
      "edad" : 21
    }
    "after" : { 3 items
      "id" : 3
      "nombre" : "Roberto"
      "edad" : 23
    }
```

- Borra registros de la nueva tabla.


```
{ 2 items
  "schema" : { 4 items
      "type" : "struct"
    "fields" : [ 1 item
        0 : { 3 items
            "type" : "int32"
            "optional" : false
            "field" : "id"
          }
      ]
      "optional" : false
      "name" : "dbserver1.inventory.ejercicio3.Key"
  }
  "payload" : { 1 item
      "id" : 3
```


- Elimina campos de la nueva tabla


```
{ 2 items
  "schema" : { 4 items
      "type" : "struct"
    "fields" : [ 1 item
        0 : { 3 items
            "type" : "int32"
            "optional" : false
            "field" : "id"
          }
      ]
      "optional" : false
      "name" : "dbserver1.inventory.ejercicio3.Key"
  }
  "payload" : { 1 item
      "id" : 3
  }
```


- Agrega un nuevo campo a la nueva tabla

Consulta ejecutada, 0 registros afectados. (0.000 s) Modificar

```
-- ALTER TABLE ejercicio3
-- ADD COLUMN altura INT;

SELECT *
FROM ejercicio3
```

id	nombre	edad	altura
1	Samuel	19	*NULL*
2	Lucas	22	*NULL*

Ejercicio 4
Modifica de nuevo el conector (register-mysql.json) para que recoja los mensajes de nuevo todos los cambios realizados en el esquema Inventory.

```
{
  "name": "inventory-connector",
  "config": {
    "connector.class": "io.debezium.connector.mysql.MySqlConnector",
    "tasks.max": "1",
    "database.hostname": "mysql",
    "database.port": "3306",
    "database.user": "debezium",
    "database.password": "dbz",
    "database.server.id": "184054",
    "topic.prefix": "dbserver1",
    "database.include.list": "inventory",
    "schema.history.internal.kafka.bootstrap.servers": "kafka:9092",
    "schema.history.internal.kafka.topic": "schema-changes.inventory",
    "column.mask.with.length.chars": "inventory.Customers.first_name:3"

  }
}
```

- En la tabla Customers enmascara la columna first_name (“column.mask.with.length.chars)

```
"column.mask.with.length.chars": "inventory.Customers.first_name:3"
```

Esta propiedad permite enmascarar la columna especificada (first_name) con el número de caracteres que determines (en este caso, 3).

- Por ejemplo, si el nombre es "John", se verá como "Joh***".

- Inserta nuevos registros en la tabla Customers


```
"payload" : { 6 items
    "before" : NULL
  "after" : { 3 items
      "id" : 8
      "first_name" : "Sam***"
      "last_name" : "sl"
    }
  "source" : {...} 15 items
    "op" : "c"
    "ts_ms" : 1731415600160
    "transaction" : NULL
}
```

```
"payload" : { 6 items
    "before" : NULL
  "after" : { 3 items
      "id" : 6
      "first_name" : "Tom"
      "last_name" : "Mater"
    }
```

- Modifica registros de la tabla Customers.

Ejercicio 5

En este ejercicio utliza de la documentación https://debezium.io/documentation/reference/2.5/connectors/mysql.html cualquier otra sentencia que te parezca interesante realizando los cambios en register-mysql.json. Captura los cambios de Inventory (si los hay) y los mensajes explicando qué ocurre.

column.truncate.to.length

permite truncar el contenido de una columna al número de caracteres especificado.

```
"column.truncate.to.length": "inventory.Customers.email:10"
```

NOTAS: Documentación: https://debezium.io/documentaϴon/reference/2.5/connectors/mysql.html

Recuerda para matar el conector si no lo haces desde UI puedes hacerlo así:

Para ello puedes bien desde Windows Command Line o directamente desde Kafka UI.

curl -X DELETE http://localhost:8083/connectors/inventory-connector

Si utlizas PowerShell:

Invoke-RestMethod -Uri http://localhost:8083/connectors/inventory-connector -Method DELETE

Para levantar de nuevo el conector ejecuta de nuevo:

curl -i -X POST -H "Accept:applicattion/json" -H "Content-Type:applicatti
on/json" http://localhost:8083/connectors/ -d @register-mysql.json