

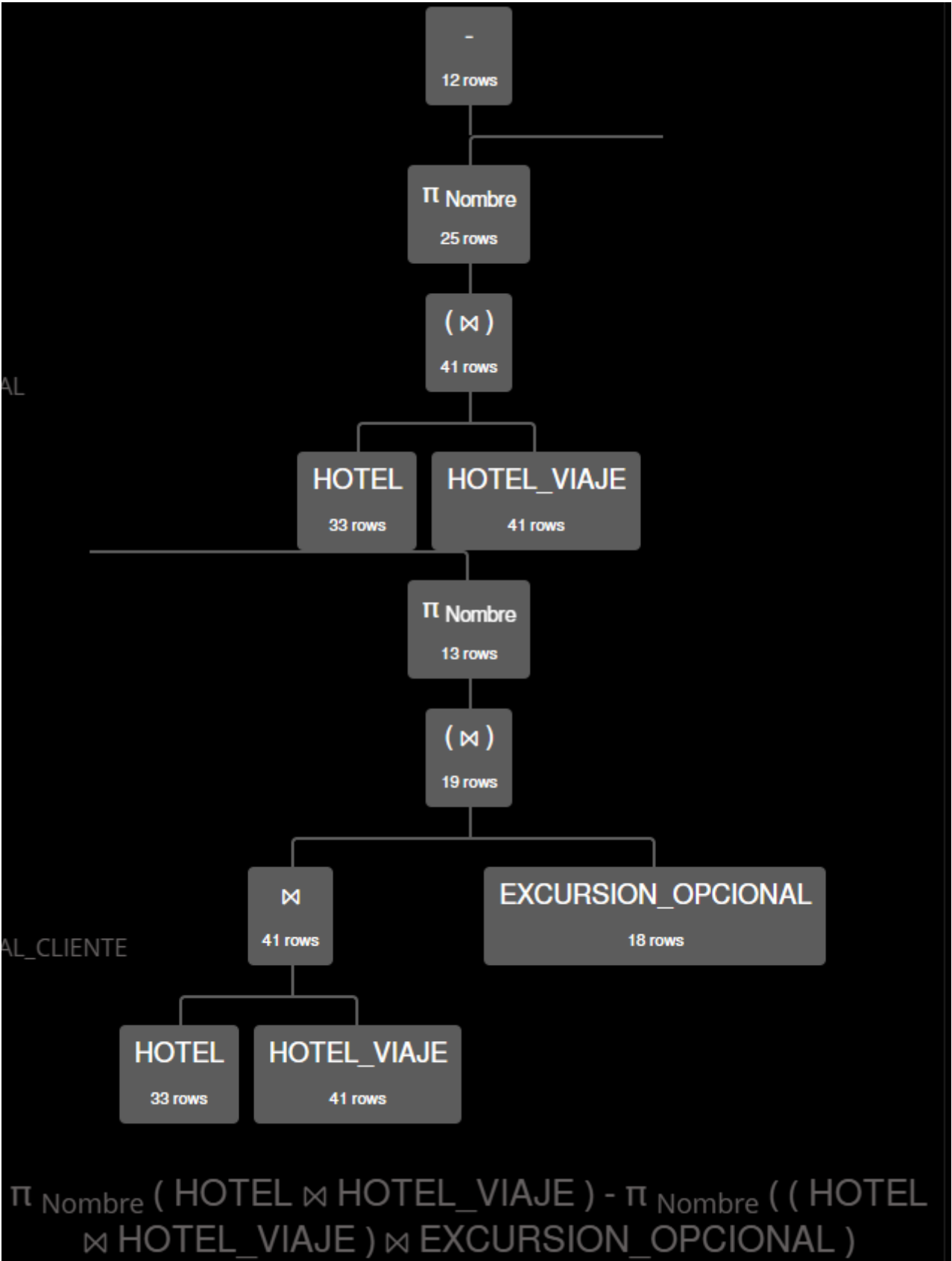
# Proyecto BD (22 de mayo, sabado)

NUEVO ENUNCIADO EN EGELA CON EL 2B BIEN

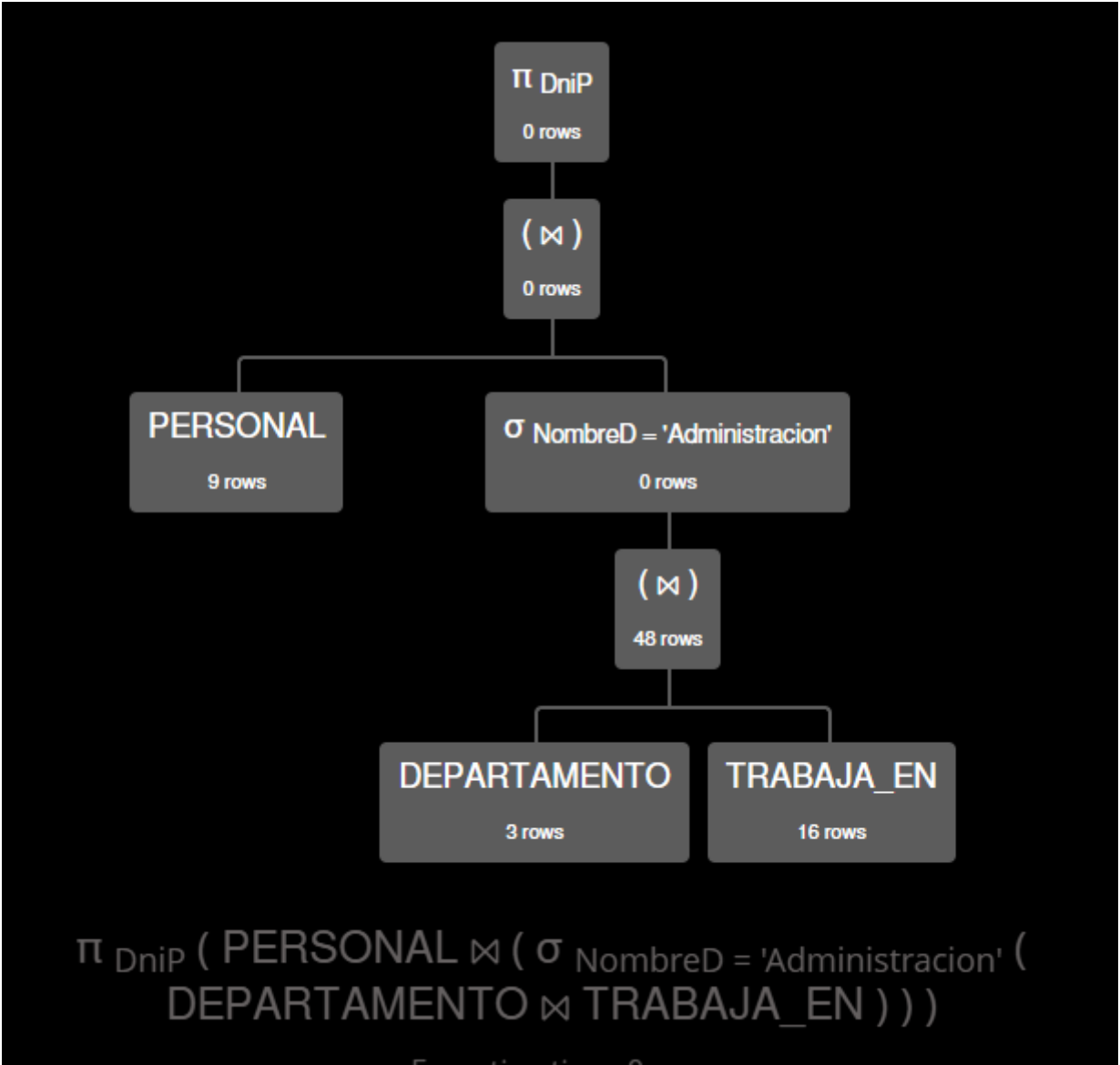
## Algebra relacional

1.  $G = \pi_{\text{DESTINOEXCURSION}, \text{COUNT}(\text{DNI})} (\sigma_{\text{DESTINO} = \text{DESTINO AND FECHASALIDA} = \text{FECHASALIDA AND NUMERO} = \text{NUMERO}} (\text{EXCURSION\_OPCIONAL\_CLIENTE} \bowtie \text{EXCURSION\_OPCIONAL}))$  RESULTADO =  $\pi_{\text{MAX}(\text{count})} (G)$

2.  $\pi_{\text{Nombre}} (\text{HOTEL} \bowtie \text{HOTEL\_VIAJE}) - \pi_{\text{Nombre}} (\text{HOTEL} \bowtie \text{HOTEL\_VIAJE} \bowtie \text{EXCURSION\_OPCIONAL})$



3.  $\pi_{\text{DniP}} (\text{PERSONAL} \bowtie (\sigma_{\text{NombreD} = \text{'Administracion'}} (\text{DEPARTAMENTO} \bowtie \text{TRABAJA\_EN})))$



# SQL

1.

```
SELECT c.NTelefono
FROM CLIENTE c
WHERE c.DNI IN (
    SELECT hvc.DNI
    FROM HOTEL_VIAJE_CLIENTE hvc
    WHERE hvc.Destino = 'Toledo'
)
AND c.DNI NOT IN (
    SELECT hvc.DNI
    FROM HOTEL_VIAJE_CLIENTE hvc
    WHERE hvc.Destino != 'Toledo'
)
```

2.

```
SELECT h.idhotel, h.Nombre, COUNT(h.IdHotel)
FROM hotel_viaje_cliente hvc RIGHT JOIN hotel h
ON hvc.IdHotel=h.IdHotel
WHERE h.IdHotel LIKE 'h%'
GROUP BY h.IdHotel
HAVING COUNT(h.IdHotel)<=2
```

3.

```
SELECT h.IdHotel, h.Nombre, h.Capacidad
FROM HOTEL h
WHERE h.Capacidad > (
    SELECT MAX(h2.Capacidad)
    FROM HOTEL h2
    WHERE h2.Localidad = 'Lima')
ORDER BY h.Capacidad ASC;
```

4.

```
SELECT g.dni, COUNT(i.DNI), COUNT(g.DNI)
FROM (guia g inner JOIN idioma i ON g.dni=i.DNI) INNER JOIN viaje v ON i.DNI=v.DNI
GROUP BY g.dni
```

5.

```
SELECT v.Destino, MAX(eoc.DNI), COUNT(eoc.DNI)
FROM VIAJE v JOIN EXCURSION_OPCIONAL_CLIENTE eoc ON v.Destino = eoc.Destino AND v.FechaSalida = eoc.FechaSalida
WHERE EXISTS (
    SELECT 1
    FROM EXCURSION_OPCIONAL_CLIENTE eoc2
    WHERE eoc2.Destino = v.Destino
    GROUP BY eoc2.Destino AND eoc2.FechaSalida
    HAVING COUNT(DISTINCT eoc2.DNI) > 1)
GROUP BY v.Destino;
```

6.

```
SELECT c.DNI, c.Nombre, v.Destino, v.FechaSalida
FROM CLIENTE c
JOIN VIAJE v ON c.DNI = v.DNI
LEFT JOIN EXCURSION_OPCIONAL_CLIENTE eoc ON v.Destino = eoc.Destino
AND v.FechaSalida = eoc.FechaSalida
WHERE v.FechaSalida like '2019%'
AND eoc.Destino IS NULL
```

## Implementacion en java

```
package proyectoBaseDatos;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class DatabaseExample {
    public static void main(String[] args) {
        // Establecer conexión con la base de datos MySQL
        try (Connection mysqlConnection = DriverManager.getConnection("jdbc:mysql://dif-mysql.ehu.es:3306/DBC49?&useSSL=false", "DBC49",
"DBC49")) {
            // Obtener los datos del cliente con DNI 10000001
            String query = "SELECT * FROM cliente WHERE DNI = ?";
            try (PreparedStatement statement = mysqlConnection.prepareStatement(query)) {
                statement.setInt(1, 10000001);
                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
```

```
String name = resultSet.getString("Nombre");
int address = resultSet.getInt("NTelefono");
// ...

// Establecer conexión con la base de datos Oracle
try (Connection oracleConnection = DriverManager.getConnection("jdbc:oracle:thin:@vsids11.si.ehu.es:1521:gipuzkoa",
"BDC49", "BDC49")) {
    // Insertar al cliente como guía en la base de datos Oracle
    String insert = "INSERT INTO GUIA(DNI, NOMBRE, NTELEFONO) VALUES (?, ?, ?)";
    try (PreparedStatement insertStatement = oracleConnection.prepareStatement(insert)) {
        insertStatement.setInt(1, 10000001);
        insertStatement.setString(2, name);
        insertStatement.setInt(3, 123456789);
        // ...
        insertStatement.executeUpdate();
    }
}
}
}
}
}
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
}
```