Recurso complementario

Ejemplo: despliegue final y puesta en operación del almacén de datos

Finalmente llegó el momento de revisar el concepto de análisis *OLAP* para esta semana. En el siguiente ejercicio se utilizará un esquema estrella muy sencillo para generar un pequeño *data mart* (del área de remuneraciones de la empresa), con los datos de la base de datos cargada en el ejercicio anterior. Para esto, se utilizará la modalidad *ROLAP* (*relational OLAP*), aprovechando el hecho de que se dispone de MariaDB para todo el ejercicio. Por lo tanto, las tablas a considerar serán sólo las siguientes:

a) **Tablas de dimensión**: donde se agregarán los datos de identificación para la tabla central de datos (tabla de hechos - *fact table*). Estas tablas para este ejercicio serán la tabla **Dim_Employee** (que contiene los empleados de la empresa) y la tabla **Dim_Salary** (que contiene los sueldos que ganan cada uno de los empleados).

b) **Tabla de hechos**: donde se almacenará el total de lo ganado por cada empleado a lo largo de un año completo (mientras el empleado esté contratado en la empresa); su nombre será **Fact_Salary**. En este caso es bueno entender que la tabla **Salary** original solo contenía los valores mensuales ganados por cada empleado, mientras que en la tabla de hechos **Fact_Salary** se calcularán y almacenarán las sumas de todos los sueldos de cada empleado durante un año, por año contratado en la empresa.

En este sentido, se decidió solo almacenar los datos del empleado y de lo que ganan dentro del *data mart*, con lo cual a nivel de Pentaho se deberá crear una serie de transformaciones que soporten este esquema:

a) Una primera transformación que copie los datos desde la tabla relacional **Employee** hacia la tabla de dimensión **Dim_Employee** (que se supone, estará ubicada en el sistema *ROLAP* del *data mart*, en otra base de datos).

b) Una segunda transformación que copie los datos desde la tabla relacional **Salary** hacia la tabla de dimensión **Dim_Salary** (que se supone, también estará ubicada en el sistema *ROLAP* del *data mart*, en otra base de datos).

c) Una tercera transformación que sume los sueldos por número de mes, para cada empleado, de la tabla relacional **Salary**, genere los totales por empleado y envíe dicha información hacia la tabla de dimensión **Fact_Salary** (que igual que en los dos casos anteriores, estará ubicada en el sistema *ROLAP* del *data mart*, en otra base de datos).

Viendo otro ejemplo: si se considera la forma para sumar sueldos, considerando que solo se entregará la suma de sueldos de cada empleado por año, se puede ejecutar:

```
SELECT
salary.pay_date,
employee.employee_id, salary.salary_paid, SUM(salary.salary_paid)
FROM salary,employee
WHERE salary.employee_id = employee.employee_id
GROUP BY employee.employee_id, YEAR(salary.pay_date)
```

Mientras que otra forma es la siguiente, que además de la suma anual, suma todos los sueldos anuales en un único gran total, por cada empleado:

```
SELECT
salary.pay_date,
employee.employee_id, salary.salary_paid, SUM(salary.salary_paid)
FROM salary,employee
WHERE salary.employee_id = employee.employee_id
GROUP BY employee.employee_id,YEAR(salary.pay_date) WITH ROLLUP
```

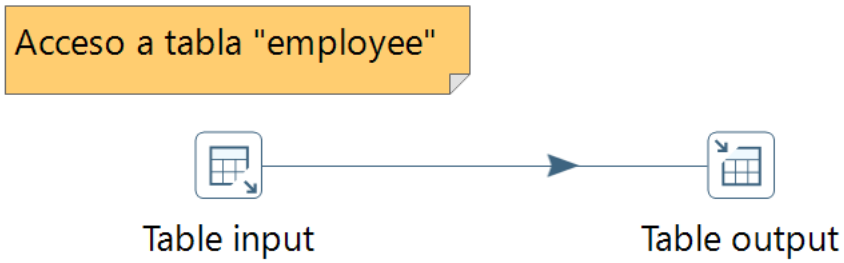
Para lograr automatizar este proceso analítico simple, se crearán las tres transformaciones en *Spoon*, un trabajo que llame a las tres tareas y las tablas *ROLAP* dentro de MariaDB (para aprovechar dicha infraestructura relacional). Los pasos son:

1. Aprovecha las dos transformaciones creadas anteriormente, llamadas **transformación_06d** (que crea y copia la tabla de **Employee**) y **transformación_06f** (que crea y copia la tabla de **Salary**) para revisar los detalles de transferencia de datos. La idea ahora no es copiar datos desde Microsoft Access hacia MariaDB, sino que desde tablas ya creadas en MariaDB (es decir, las tablas **Employee** y **Salary**) hacia las tablas *ROLAP* a crear (que se llamarán **Dim_Employee** y **Dim_Salary** y que también estarán en este ejercicio dentro de MariaDB). Para eso puede ocupar el modelo de la figura 1.

Figura 1.

Transformación tipo para copiar datos de una tabla a otra en MariaDB

Fuente: elaboración propia.



Por lo tanto, utiliza dos pasos **Table input** para leer las tablas **Employee** y **Salary** ya existentes, y dos pasos **Table output** para crear las tablas **Dim_Employee** y **Dim_Salary** y escribir los datos en ellas; luego una cada paso **Table input** con su paso **Table output** respectivo. También recuerda que, según requerimientos de datos, espacio o rendimiento, puede que no siempre sea necesario escribir todas las columnas de datos de las tablas originales en las tablas dimensión; ten esto en cuenta, dependiendo del problema a analizar y resolver.

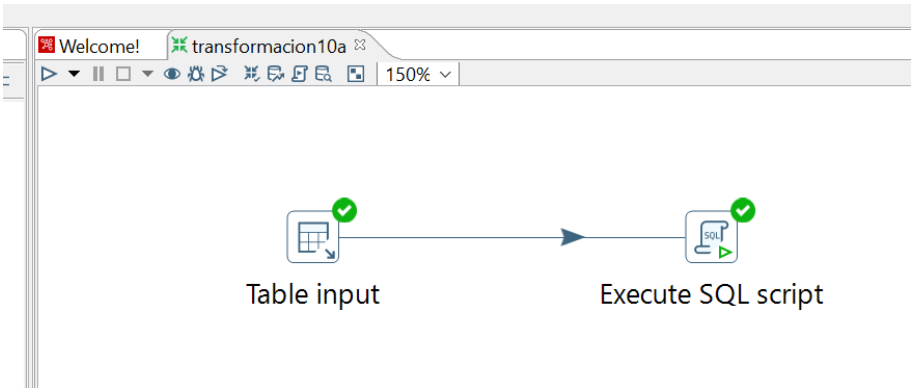
También ten en cuenta que deberás crear manualmente las tablas **Dim_Employee, Dim_Salary** y **Fact_salary** dentro de MariaDB, usando HeidiSQL para ello.

b) Crea la tercera transformación que calculará la suma de salarios de los empleados. Para eso, básate en el siguiente diagrama de transformación de *Kettle* (figura 2).

Figura 2.

Esquema de transformación para calcular la suma de salarios

Fuente:

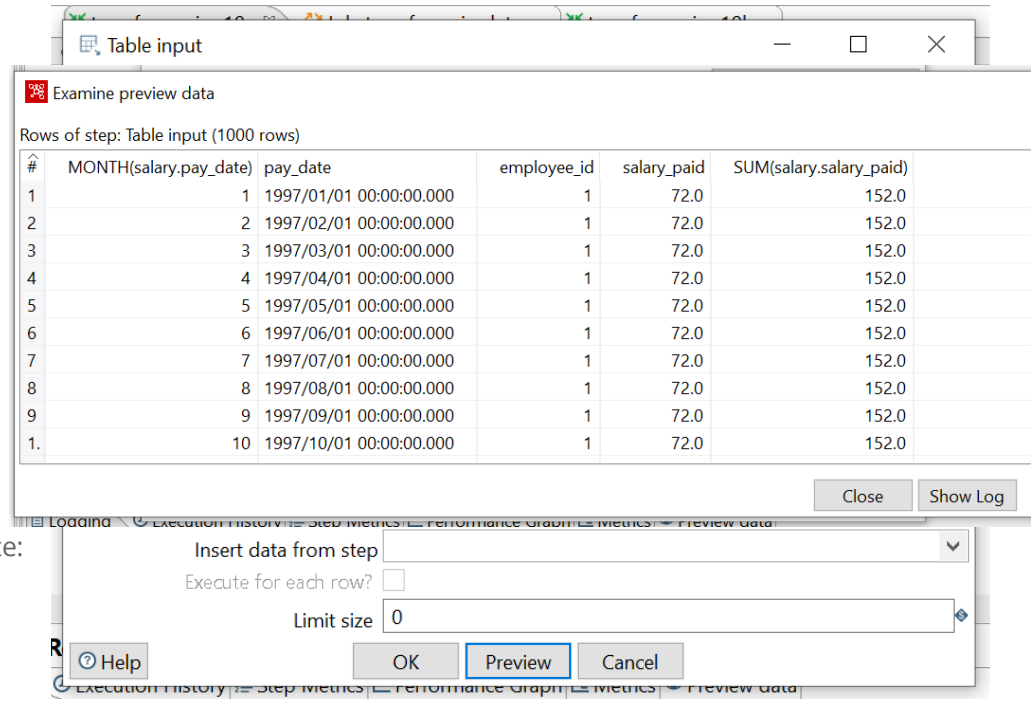


elaboración propia.

d) Configura el paso **Table input** de esta tercera transformación para leer los datos crudos de la tabla relacional y aprovechar de calcular la suma de los sueldos del mismo mes por empleado (figura 3).

Figura 3.

*Configuración del paso **Table input** (arriba) y datos leídos desde la tabla con la suma de salarios del mismo mes en diferentes años (abajo)*

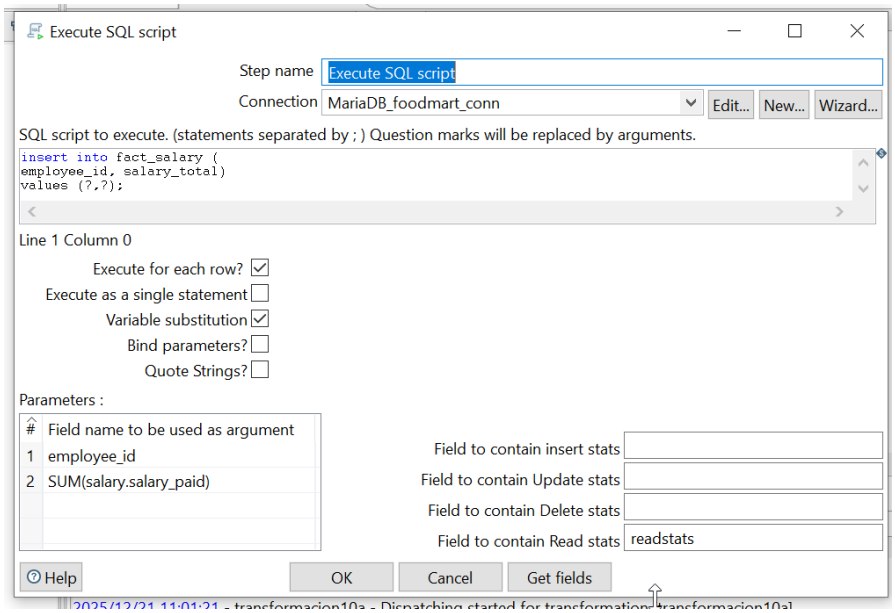


Fuente: elaboración propia.

e) Configura el paso **Execute SQL** *script* (que se puede utilizar para ejecutar sentencias de SQL *insert, delete, update, etc.*) de forma tal que tome todos los datos del paso anterior (mediante el uso de los símbolos de parámetros **'?'**), y envía solo el ID de empleado y la suma total de sueldos hacia la tabla final **Fact_Salary** (figura 4).

Figura 4.

Script de SQL para insertar los datos de las sumas de salarios en la tabla de hechos

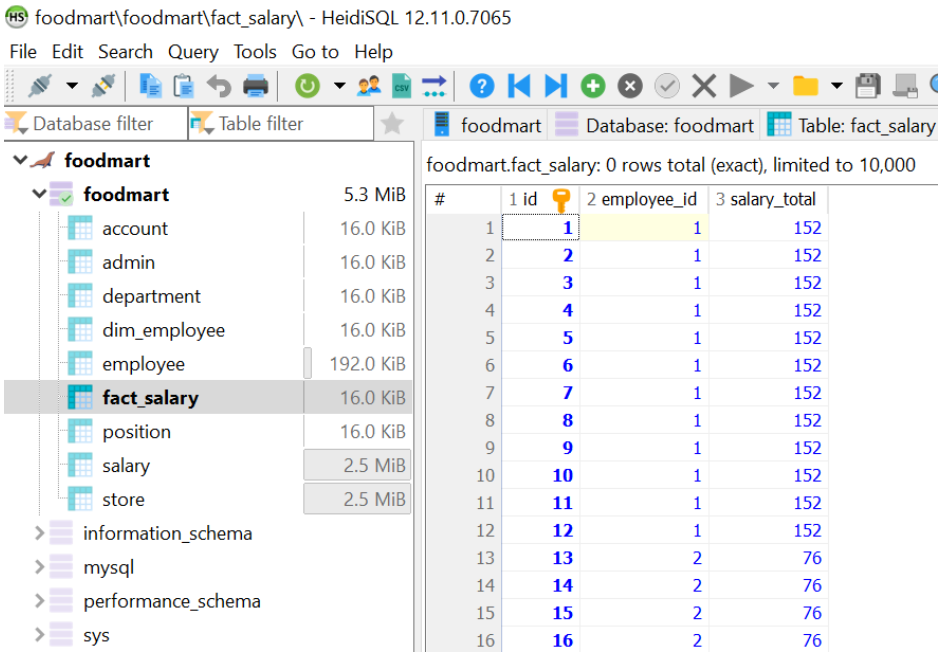


Fuente: elaboración propia.

e) Crea un nuevo trabajo (*job*) en *Spoon* y agrega en él las tres transformaciones anteriores. Ejecútalo y revisa desde HeidiSQL que los datos enviados y calculados estén presentes (figura 5).

Figura 5.

Revisión de datos en la tabla de hechos final, dentro de MariaDB

Fuente: 

elaboración propia.

Con estos pasos ya tendrás una idea de cómo partir la creación de un pequeño sistema similar a un *data mart* que te permitirá ir guardando datos analíticos de acuerdo con diferentes tipos de mediciones o dimensiones. Observa que en el ejemplo dado, **los sueldos se calculan por mismo mes, no importando el año para ese grupo de meses iguales** (por ejemplo, se suman todos los sueldos de los meses de enero de todos los años, todos los sueldos de los meses de febrero de todos los años, etc.), pero tal vez requieras sumar los sueldos por trimestres de un mismo año, por semestres de un mismo año, para los 12 meses de un mismo año, para los años de dos en dos, de tres en tres, etc.

Se sugiere que trates de mejorar el sistema propuesto, ya que probablemente encuentres que hay datos que no se estén calculando como se requiere o tengas ideas de cómo mejorarlo. Esto se deja como ejercicio para que busques formas de resolverlo de otra forma o mejorarlo con transacciones de SQL propias. También, nota la cantidad de tiempo que requiere esta "pequeña transacción/cálculo" de datos y extrapola luego esa cantidad de tiempo al cálculo de tablas históricas con millones de datos. ¿De qué otras formas piensas que el cálculo podría realizarse más rápido o más eficientemente?