

Reporte de Experimentación: Construcción del Proceso ETL

Junio 14, 2024

Luisa Fernanda Giraldo
Juan Sebastian Guzman

Introducción

La gestión de bases de datos relacionales y no relacionales, junto con los procesos de ETL (Extract, Transform, Load), son fundamentales para el ejercicio profesional de un científico de datos. Estas tecnologías permiten manejar, transformar e integrar grandes volúmenes de datos, lo que es crucial para el análisis y la toma de decisiones informadas.

Este trabajo consistió en un proyecto aplicado donde se trabajó con datos de propiedades raíces, los cuales se encontraban en diferentes fuentes relacionales y no relacionales. Utilizando herramientas de ETL, se extrajeron, transformaron y cargaron estos datos, integrándolos en un formato unificado para su análisis posterior.

Conceptos Claves: Bases de datos relacionales, bases de datos no relacionales, procesos ETL, integración de datos, transformación de datos.

Metodología y herramientas

1. Entendimiento de Fuentes Relacionales:

Se analizaron las fuentes de datos relacionales, comprendiendo su estructura y relación con el archivo de salida esperado. Esto incluyó el estudio de la base de datos **amesdbtemp** y las tablas **MSSubClass**, **MSZoning**, **TypeQuality**, **SaleProperties** y **FloorDetail**, alojadas en la instancia de **Elephant SQL** llamada **Proyecto_ETL_E7**. Se utilizó **PostgreSQL** como herramienta para la revisión de los datos, la estructura de la base y las consultas de los datos.

2. Entendimiento de Fuentes No Relacionales:

Se exploraron las colecciones de **MongoDB**, como **garage**, **pool**, **bsmt** y **misc**, para entender su estructura y su relación con el archivo de salida. Se utilizó **MongoDB** para la revisión de los datos, la estructura de la base y las consultas de los datos.

3. Creación del Pipeline ETL:

Se utilizó **Pentaho Data Integration (PDI)** para crear el pipeline y realizar los procesos de extracción, transformación e importación de los datos. Pentaho permitió la integración y transformación de datos de fuentes relacionales y no relacionales en un archivo CSV unificado.

Procesamiento de la información

Revisión y Comprensión de las Fuentes de Datos

1. Análisis archivo de salida:

- Se realizó una revisión del archivo de salida esperado, cruzando las variables requeridas, y las variables de encontradas en las diferentes tablas de las bases relacionales. Este acercamiento permitió definir los Inputs requeridos en la construcción del pipeline.

A continuación se relaciona matriz de cruce de información variables esperadas y variables en las bases relacionales.

Salida	AmesProperty	floordetail	amesdbtemp	mssubclass	mszoning	saleproperty	typequality
PID	1	1	1	0	0	1	0
MS SubClass	0	0	1	0	0	0	0
MS Zoning	0	0	1	0	0	0	0
Lot Frontage	1	0	0	0	0	0	0
Lot Area	1	0	0	0	0	0	0
Street	1	0	0	0	0	0	0
Alley	1	0	0	0	0	0	0
Lot Shape	1	0	0	0	0	0	0
Land Contour	1	0	0	0	0	0	0
Utilities	1	0	0	0	0	0	0
Lot Config	1	0	0	0	0	0	0
Land Slope	1	0	0	0	0	0	0
Neighborhood	1	0	0	0	0	0	0
Condition 1	1	0	0	0	0	0	0
Condition 2	1	0	0	0	0	0	0
Bldg Type	1	0	0	0	0	0	0
House Style	1	0	0	0	0	0	0
Overall Qual	1	0	0	0	0	0	0
Overall Cond	1	0	0	0	0	0	0
Year Built	1	0	0	0	0	0	0
Year Remod/Add	1	0	0	0	0	0	0
Roof Style	0	0	1	0	0	0	0
Roof Matl	0	0	1	0	0	0	0
Exterior 1st	0	0	1	0	0	0	0
Exterior 2nd	0	0	1	0	0	0	0
Mas Vnr Type	0	0	1	0	0	0	0
Mas Vnr Area	0	0	1	0	0	0	0
Exter Qual	0	0	1	0	0	0	0
Exter Cond	0	0	1	0	0	0	0
Foundation	0	0	1	0	0	0	0
Bsmt Qual	0	0	0	0	0	0	0
Bsmt Cond	0	0	0	0	0	0	0
Bsmt Exposure	0	0	0	0	0	0	0
BsmtFin Type 1	0	0	0	0	0	0	0
BsmtFin SF 1	0	0	0	0	0	0	0
BsmtFin Type 2	0	0	0	0	0	0	0

BsmtFin SF 2	0	0	0	0	0	0	0
Bsmt Unf SF	0	0	0	0	0	0	0
Total Bsmt SF	0	0	0	0	0	0	0
Heating	0	0	1	0	0	0	0
Heating QC	0	0	1	0	0	0	0
Central Air	0	0	1	0	0	0	0
Electrical	0	0	1	0	0	0	0
1st Flr SF	0	0	1	0	0	0	0
2nd Flr SF	0	0	1	0	0	0	0
Low Qual Fin SF	0	0	1	0	0	0	0
Gr Liv Area	0	0	0	0	0	0	0
Bsmt Full Bath	0	0	0	0	0	0	0
Bsmt Half Bath	0	0	0	0	0	0	0
Full Bath	0	1	0	0	0	0	0
Half Bath	0	1	0	0	0	0	0
Bedroom AbvGr	0	0	0	0	0	0	0
Kitchen AbvGr	0	0	1	0	0	0	0
Kitchen Qual	0	0	1	0	0	0	0
TotRms AbvGrd	0	0	1	0	0	0	0
Functional	0	0	1	0	0	0	0
Fireplaces	0	0	1	0	0	0	0
Fireplace Qu	0	0	1	0	0	0	0
Garage Type	0	0	0	0	0	0	0
Garage Yr Blt	0	0	0	0	0	0	0
Garage Finish	0	0	0	0	0	0	0
Garage Cars	0	0	0	0	0	0	0
Garage Area	0	0	0	0	0	0	0
Garage Qual	0	0	0	0	0	0	0
Garage Cond	0	0	0	0	0	0	0
Paved Drive	0	0	1	0	0	0	0
Wood Deck SF	0	0	1	0	0	0	0
Open Porch SF	0	0	1	0	0	0	0
Enclosed Porch	0	0	1	0	0	0	0
3Ssn Porch	0	0	1	0	0	0	0
Screen Porch	0	0	1	0	0	0	0
Pool Area	0	0	0	0	0	0	0
Pool QC	0	0	0	0	0	0	0
Fence	0	0	1	0	0	0	0
Misc Feature	0	0	0	0	0	0	0
Misc Val	0	0	0	0	0	0	0
Mo Sold	0	0	0	0	0	0	0
Yr Sold	0	0	0	0	0	0	0
Sale Type	0	0	0	0	0	1	0
Sale Condition	0	0	0	0	0	1	0
SalePrice	0	0	0	0	0	1	0

Proceso de Extracción y Transformación de Datos

Una vez se revisaron los datos, estructura y variables encontradas en las diferentes fuentes de datos, se procedió a construir el pipeline, ingresando los inputs requeridos, realizando las transformaciones necesarias y por último, los procesos de unión de tablas para obtener

el documento final con la recopilación del proceso. En la Ilustración 1 se presenta el esquema ETL del proyecto.

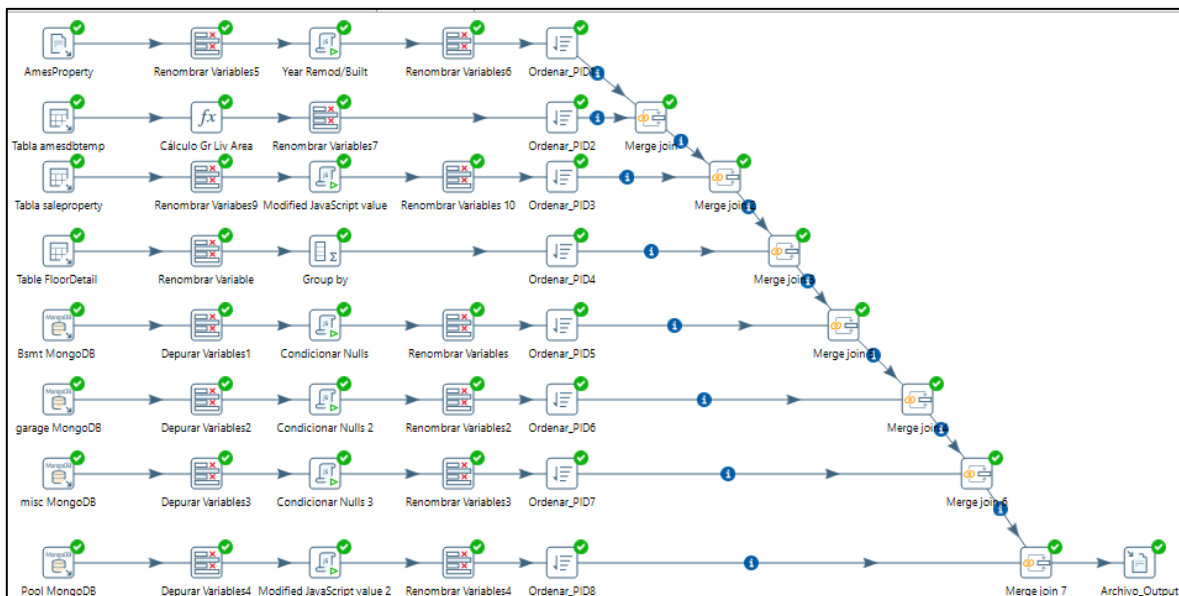


Ilustración 1 Esquema Pipeline ETL Proyecto

Transformaciones

Se describen las transformaciones realizadas para cada una de las fuentes de datos

Amesproperty:

- Select Value-Renombrar Variables: Se modifica el nombre a variables de manera que puedan ser posteriormente llamadas mediante un método en Java: Year_Remod_Add y Year_Built.
- Modified Javascript Value Year Remod/Built: Se define Script en Java para la transformación requerida para "Year Remod", indicando que, si el valor es nulo, la variable toma el dato del campo "Year Built".

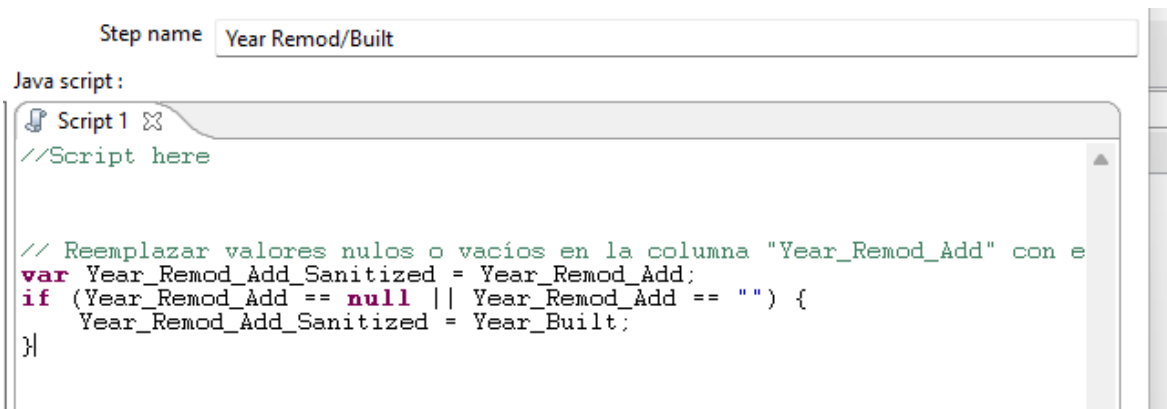


Ilustración 2 Script en Java para transformación Year Remod

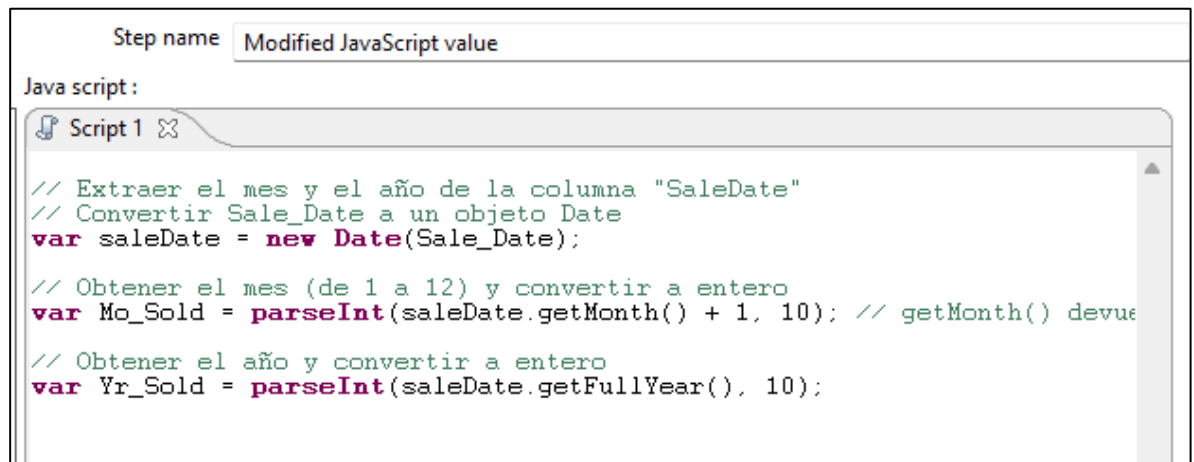
- Select Value- Renombrar Variables: Se renombran las variables conservando su valor del archivo fuente.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge.

Amesdtemp:

- Fórmula- Cálculo GRV LIV AREA: Se establece formula para la definición de la variable GRV LIV AREA, como **[1st Flr SF]+[2nd Flr SF]+[Low Qual Fin SF]**
- Select Value- Renombrar Variables: Se modifica el nombre de la variable “pid” por “PID”
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge

Salesproperty

- Select Value-Renombrar Variables: Se modifica el nombre a variables de manera que puedan ser posteriormente llamadas mediante un método en Java; “Sale_Date”.
- Modified Javascript Value Year Remod/Built: Se define Script en Java para la transformación requerida para “Mo Sold” y “Yr Sold”, los cuales se obtienen como mes y año de la fecha de venta “Sale Date”.



The screenshot shows a software interface with a tab labeled "Step name" and "Modified JavaScript value". Below this, there is a section titled "Java script:" containing a code editor with a tab labeled "Script 1". The code in the editor is as follows:

```
// Extraer el mes y el año de la columna "SaleDate"
// Convertir Sale_Date a un objeto Date
var saleDate = new Date(Sale_Date);

// Obtener el mes (de 1 a 12) y convertir a entero
var Mo_Sold = parseInt(saleDate.getMonth() + 1, 10); // getMonth() devuelve

// Obtener el año y convertir a entero
var Yr_Sold = parseInt(saleDate.getFullYear(), 10);
```

- Select Value- Renombrar Variables; Se modifica el nombre de las variables “Mo Sold”, “Yr Sold” y “Sale_Date”, de acuerdo lo requerido en la salida esperada de los datos.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge

FloorDetail

- Select Value-Renombrar Variables: Se modifica el nombre de la variable “pid” por “PID”

- Group by

Step name

Group by

Include all rows?

☐

Temporary files directory

%%java.io.tmpdir%%

Browse...

TMP-file prefix

grp

Add line number, restart in each group

☐

Line number field name

Always give back a result row

☐

The fields that make up the group:

#	Group field
1	PID

Get Fields

Aggregates :

#	Name	Subject	Type
1	Bsmt Full Bath	Full Bath	Sum
2	Bsmt Half Bath	Half Bath	Sum
3	Bedroom	bedrooms	Sum

Get lookup fields

- ## Bmt MondoDB

- **Select Value-Renombrar Variables:** Se modifica el nombre a variables de manera que puedan ser posteriormente llamadas mediante un método en Java. A continuación se relaciona el método usado:

Step name Depurar Variables1						
Select & Alter Remove Meta-data						
Fields to alter the meta-data for :						
#	Fieldname	Rename to	Type	Length	Precision	Bin
1	Bsmt Unf SF	Bsmt_Unf_SF	None		0	N
2	Bsmt Cond	Bsmt_Cond	None			N
3	Bsmt Half Bath	Bsmt_Half_Bath	None		0	N
4	BsmtFin Type 1	BsmtFin_Type1	None			N
5	PID		None		0	N
6	BsmtFin Type 2	BsmtFin_Type2	None			N
7	Total Bsmt SF	Total_Bsmt_SF	None		0	N
8	Bsmt Exposure	Bsmt_Exposure	None			N
9	Bsmt Full Bath	Bsmt_Full_Bath	None		0	N
10	Bsmt Qual	Bsmt_Qual	None			N
11	BsmtFin SF 2	BsmtFin_SF_2	None		0	N
12	BsmtFin SF 1	BsmtFin_SF_1	None		0	N

- Modified Javascript Value: esta transformación consiste en script de java para modificar valores nulos de las diferentes variables de la base. A continuación el script:

```
// Reemplazar valores nulos en la columna "Bsmt Qual" con "NA"
if (Bsmt_Qual == null) {
    Bsmt_Qual = "NA";
}

// Reemplazar valores nulos en la columna "Bsmt_Cond" con "NA"
if (Bsmt_Cond == null) {
    Bsmt_Qual = "NA";
}

// Reemplazar valores nulos en la columna "Bsmt_Exposure" con "NA"
if (Bsmt_Exposure == null) {
    Bsmt_Exposure = "NA";
}

// Reemplazar valores nulos en la columna "BsmtFin_Type 1" con "NA"
if (BsmtFin_Type1 == null) {
    BsmtFin_Type1 = "NA";
}

// Reemplazar valores nulos en la columna "BsmtFin_Type 2" con "NA"
if (BsmtFin_Type2 == null) {
    BsmtFin_Type2 = "NA";
}

// Reemplazar valores nulos en la columna "BsmtFin_SF_1" con "NA"
if (BsmtFin_SF_1 == null) {
    BsmtFin_SF_1 = 0;
}
```

```
// Reemplazar valores nulos en la columna "BsmtFin_SF_2" con "NA"
if (BsmtFin_SF_2 == null) {
    BsmtFin_SF_2 = 0;
}

// Reemplazar valores nulos en la columna "Bsmt_Unf_SF" con "NA"
if (Bsmt_Unf_SF == null) {
    Bsmt_Unf_SF = 0;
}

// Reemplazar valores nulos en la columna "Total_Bsmt_SF" con "NA"
if (Total_Bsmt_SF == null) {
    Total_Bsmt_SF = 0;
}

// Reemplazar valores nulos en la columna "Bsmt_Full_Bath" con "NA"
if (Bsmt_Full_Bath == null) {
    Bsmt_Full_Bath = 0;
}

// Reemplazar valores nulos en la columna "Bsmt_Half_Bath" con "NA"
if (Bsmt_Half_Bath == null) {
    Bsmt_Half_Bath = 0;
}
```

- Select Value- Renombrar Variables: Se renombrar las variables para mantener su nombre de la fuente original.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge

Garage MongoDB

- Select Value-Renombrar Variables
- Modified Javascript Value: esta transformación consiste en script de java para modificar valores nulos de las diferentes variables de la base. A continuación el script:
- Select Value- Renombrar Variables: Se renombrar las variables para mantener su nombre de la fuente original.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge

```
}
```

Misc Mongo DB

- Select Value-Renombrar Variables
- Modified Javascript Value: esta transformación consiste en script de java para modificar valores nulos de las diferentes variables de la base. A continuación el script:

```
//Script here
```

```
// Reemplazar valores nulos en la columna "Garage_Type" con "NA"
if (Garage_Type == null) {
    Garage_Type = "NA";
}
```

```
// Reemplazar valores nulos en la columna "Garage_Finish" con "NA"
```



```

if (Garage_Finish == null) {
    Garage_Finish = "NA";
}

// Reemplazar valores nulos en la columna "Garage_Qual" con "NA"
if (Garage_Qual == null) {
    Garage_Qual = "NA";
}

// Reemplazar valores nulos en la columna "Garage_Cond" con "NA"
if (Garage_Cond == null) {
    Garage_Cond = "NA";
}

// Reemplazar valores nulos en la columna "Garage_Yr_Blt" con "NA"
if (Garage_Yr_Blt == null) {
    Garage_Yr_Blt = 0;
}

// Reemplazar valores nulos en la columna "Garage_Cars" con "NA"
if (Garage_Cars == null) {
    Garage_Cars = 0;
}

// Reemplazar valores nulos en la columna "Garage_Area" con "NA"
if (Garage_Area == null) {
    Garage_Area = 0;
}

```

- Select Value- Renombrar Variables: Se renombrar las variables para mantener su nombre de la fuente original.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge
-

```

//Script here

// Reemplazar valores nulos en la columna "Misc_Feature" con "NA"
if (Misc_Feature == null) {
    Misc_Feature = "NA";
}

// Reemplazar valores nulos en la columna "Misc_Val" con "NA"
if (Misc_Val == null) {
    Misc_Val = 0;
}

```

Pool Mongo DB

- Select Value-Renombrar Variables
- Modified Javascript Value: esta transformación consiste en script de java para modificar valores nulos de las diferentes variables de la base. A continuación el script:

```

//Script here

// Reemplazar valores nulos en la columna "Pool_QC" con "NA"
if (Pool_QC == null) {
    Pool_QC = "NA";
}

// Reemplazar valores nulos en la columna "Pool_Area" con "NA"

```

```
if (Pool_Area == null) {  
    Pool_Area = 0;  
}
```

- Select Value- Renombrar Variables: Se renombrar las variables para mantener su nombre de la fuente original.
- Sort Rows- Ordenar PID: Se ordena la tabla por PID, llave con la que se realizará posteriormente un proceso merge

Lecciones Aprendidas y conclusiones

1. **Importancia de la Planificación:** La planificación adecuada del proceso ETL es crucial. Definir claramente las etapas de extracción, transformación y carga ayuda a evitar problemas durante la implementación y asegura una integración fluida de los datos.
2. **Conocimiento de las Herramientas:** EL proyecto permitió la interacción con diferentes herramientas claves en los procesos ETL. En este caso, se utilizó Pentaho Data Integration (PDI) para la creación del pipeline, PostgreSQL para la revisión y consulta de datos relacionales, y MongoDB Compass para explorar las colecciones no relacionales. Cada herramienta tiene sus propias fortalezas y limitaciones que deben ser comprendidas para maximizar su uso.
3. **Manejo de Diferentes Tipos de Datos:** Trabajar con fuentes de datos relacionales y no relacionales requiere habilidades para manejar diferentes estructuras y formatos de datos. Comprender cómo integrar datos de múltiples fuentes es esencial para crear un dataset cohesivo y útil.
4. **Transformaciones de Datos:** Las transformaciones de datos son una parte crítica del proceso ETL. La capacidad de limpiar, normalizar y transformar los datos según las necesidades del proyecto es vital para asegurar la calidad y consistencia de los datos finales.
5. **Gestión de Errores y Depuración:** Identificar y resolver errores en el proceso ETL es una habilidad clave. Es importante realizar pruebas exhaustivas y utilizar herramientas de depuración para asegurar que el pipeline funciona correctamente y que los datos se integran sin problemas.
6. **Adaptabilidad y Flexibilidad:** Los proyectos de ETL a menudo requieren adaptabilidad y flexibilidad para ajustarse a cambios en los requisitos o en las fuentes de datos. Es necesario Estar preparado para ajustar el pipeline y las transformaciones según sea necesario es crucial para el éxito del proyecto.