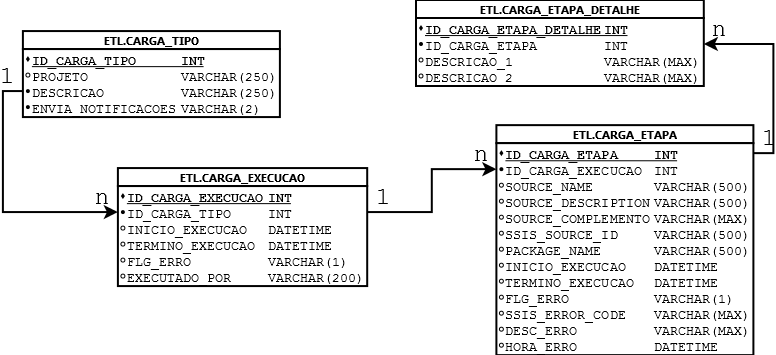
**Template de ETL – Integration Services**

**Purpose**

Establish an ETL pattern that allows to monitor its execution, just as to notify by mail in case of successful or error.

**Data Model**



**Intro**

First of all se few tables and procedures must be created. Most of those tables are going to have data inserted by the template via SSIS Event Handler.

Event Handlers are tasks automatically executed by Integration Services under specific circunstances (for example: the end of a task).

**Tables**

**CARGA\_TIPO**: It’s the type of ETL that is going to be executed. The data insertion in this tables is manually.

Columns:

|  |  |
| --- | --- |
| **Columns** | **Description** |
| ID\_CARGA\_TIPO | Primary key of the table |
| PROJETO | Definition of the ETL |
| DESCRICAO | More detailed explanation of what the ETL is |
| ENVIA\_NOTIFICACOES | Indicates if it is necessary to notify bye mail the ETL execution in case of success or error. |

**CARGA\_EXECUCAO**: It has the execution of the ETL, showing when when it started and when it ended. It’s insertion is made automatically by the ETL template.

Columns:

|  |  |
| --- | --- |
| **Columns** | **Description** |
| ID\_CARGA\_EXECUCAO | Primary key of the table |
| ID\_CARGA\_TIPO | Foreign Key that references the table CARGA\_TIPO |
| INICIO\_EXECUCAO | Indicates the start of the ETL execution |
| TERMINO\_EXECUCAO | Indicates the end of the ETL execution |
| FLG\_ERRO | Flag that shows if the ETL had na error |
| EXECUTADO\_POR | Windows user that executed the ETL |

**CARGA\_ETAPA**: It has the steps of the ETL execution. In this context, steps are defined as each task of Integration Services (SSIS). It’s insertion is made automatically by the ETL template.

Columns:

|  |  |
| --- | --- |
| **Columns** | **Description** |
| ID\_CARGA\_ETAPA | Primary key of the table |
| ID\_CARGA\_EXECUCAO | Foreign Key that references the table CARGA\_EXECUCAO |
| SOURCE\_NAME | Name of the task on SSIS. Corresponds to the variable “System::SourceName” of the SSIS |
| SOURCE\_DESCRIPTION | Description of the task. Corresponds to the variable “System::SourceDescription”. If you use the character “|”, ont the variable, what comes before is going to be inserted on the column SOURCE\_DESCRIPTION, and what comes after is going to be inserted on the column SOURCE\_COMPLEMENTO |
| SOURCE\_COMPLEMENTO | Complementar description of the task. It’s recomended to use when there’s a task inside a loop.  It’s insertion need to be specified when developing the ETL, and the atribute “Description” must chave the delimitator character “|”.  Example: The Task description is “EXECUTE SQL TASK|LOOP NUMBER 12”; the column “SOURCE\_DESCRIPTION” is going to be “EXECUTE SQL TASK” and SOURCE\_COMPLEMENTO is going to be “LOOP NUMBER 12”. |
| SSIS\_SOURCE\_ID | Task key generated by SSIS. Important to know that if the task is executed more than once inside the package (like in a loop, for example) its value is going to repeat. |
| PACKAGE\_NAME | Package name of the ETL |
| INICIO\_EXECUCAO | Start of the task |
| TERMINO\_EXECUCAO | End of the task |
| FLG\_ERRO | Flag that shows if the task had na error |
| SSIS\_ERROR\_CODE | SSIS error code |
| DESC\_ERRO | SSIS error description |
| HORA\_ERRO | Date/Time of the error |

**CARGA\_ETAPA\_COMPLEMENTO**: Optional Filling table to complement a specific task of the ETL (like the amount of rows inserted in a table, for example). It’s insertion need to be implemented for each ETL development.

Columns:

|  |  |
| --- | --- |
| **Columns** | **Description** |
| ID\_CARGA\_ETAPA\_COMPLEMENTO | Primary key of the table |
| ID\_CARGA\_ETAPA | Foreign Key that references the table CARGA\_ETAPA |
| DESCRICAO\_1 | Column for description |
| DESCRICAO\_2 | Other column for description |

**Data Insertions (procedures)**

The ETL template inserts data on the tables by 2 ways:

1. SQL Task that executes procedures;
2. Procedures executed by Event Handlers.

**PR\_CADASTRA\_CARGA\_EXECUCAO:**

* Way of execution: SQL Task on the begining (Register the ETL start of Execution) and SQL Task on the end (to register end of ETL execution)
* Details:

1. If its called on the begging it inserts data on the table CARGA\_EXECUCAO with the ID setted on the SSIS variable ID\_CARGA\_EXECUCAO .
2. If its called ont the end is just made na update on the column TERMINO\_EXECUCAO, just as the e-mails notifications.

**PR\_CADASTRA\_CARGA\_ETAPAS:**

* Way of execution: Event handler “OnPostValidate” before each task execution and also on event handler “OnPostExecute” after each task completion.
* Details:

1. For “OnPostValidate” is basically made an insert on the table CARGA\_ETAPA.
2. For “OnPostExecute” is made na update on the column TERMINO\_EXECUCAO of the table CARGA\_ETAPA.

**PR\_CADASTRA\_CARGA\_ETAPAS\_ERRO:**

* Way of execution: Event handler “OnError” if there is any error during the task execution.
* Details:

Update on the columns FLG\_ERRO, SSIS\_ERROR\_CODE, DESC\_ERRO, HORA\_ERRO of the table CARGA\_ETAPA referencing ID\_CARGA\_ETAPA with error. It also updates the column FLG\_ERRO of the table CARGA\_EXECUCAO.

Besides, it may send na email notyfying the error.