logoCC_site.jpg

# Metadados Técnico

**Projeto:** Business Intelligence

**Histórico de Alterações**

|  |  |  |
| --- | --- | --- |
| Data | Autor | Observação |
| 13/03/2011 | Jefferson Ferreira | Versão 1.0 com aceite do projeto |

[Metadados Técnico 1](#_Toc287855435)

[Bancos de dados do projeto 3](#_Toc287855436)

[CI\_MIDDLEWAY 3](#_Toc287855437)

[Datawarehouse 3](#_Toc287855438)

[CI\_DW 3](#_Toc287855439)

[Domínios 4](#_Toc287855440)

[Dimensões 5](#_Toc287855441)

[Dimensões de Tempo 5](#_Toc287855442)

[Dim\_Dia\_semana 5](#_Toc287855443)

[Dimensões de Geografia 5](#_Toc287855444)

[Dim\_País 5](#_Toc287855445)

[Dim\_Regiao\_Geografica 5](#_Toc287855446)

[Dim\_estado 6](#_Toc287855447)

[Dim\_municipio 8](#_Toc287855448)

[Dimensões da Empresa/Produto 9](#_Toc287855449)

[Dim\_bandeira \_cartao 9](#_Toc287855450)

[Dim\_patrocinador 9](#_Toc287855451)

[Dim\_cartao\_patrocinado 10](#_Toc287855452)

[Dimensão Cliente 11](#_Toc287855453)

[Dim\_cliente 11](#_Toc287855454)

[Dimensão Evento 13](#_Toc287855455)

[Dim\_tipo\_local 13](#_Toc287855456)

[Dim\_canal\_venda 13](#_Toc287855457)

[Dim\_local\_evento 14](#_Toc287855458)

[Dim\_tipo\_evento 15](#_Toc287855459)

[Dim\_evento 16](#_Toc287855460)

[Dim\_apresentacao 17](#_Toc287855461)

[Fatos 18](#_Toc287855462)

[Fato\_vendas\_promocionais\_itau 18](#_Toc287855463)

[Fato\_evento\_mes 21](#_Toc287855464)

[Ordem de execução dos ETLs 22](#_Toc287855465)

Este documento descreve as regras de negócio e formas de coleta de dados para alimentar o data warehouse.

## Bancos de dados do projeto

### CI\_MIDDLEWAY

SERVIDOR: 11.11.11.218

USUARIO:

SENHA:

BANCOS:

### Datawarehouse

### CI\_DW

SERVIDOR: 11.11.11.218

USUARIO:

SENHA:

BANCOS:

## Domínios

Domínios são valores fixos cadastrados na base de dados que indicam alguma coisa.

|  |  |  |
| --- | --- | --- |
| Dimensão | Valor | Descrição |
| Dim\_dia\_semana | 1 | Dom |
| 2 | Seg |
| 3 | Ter |
| 4 | Qua |
| 5 | Qui |
| 6 | Sex |
| 7 | Sab |

## 

## Dimensões

### Dimensões de Tempo

Utilizar os scripts abaixo para carga:

#### Dim\_Dia\_semana

|  |  |
| --- | --- |
| Dimensão | Dim\_Dia\_Semana |
| Origem | Script de criação de dados |
| Execução | Apenas 1 vez |
|  | |
| insert into dim\_dia\_semana values (1, 'Dom') | |
| insert into dim\_dia\_semana values (2, 'Seg') | |
| insert into dim\_dia\_semana values (3, 'Ter') | |
| insert into dim\_dia\_semana values (4, 'Qua') | |
| insert into dim\_dia\_semana values (5, 'Qui') | |
| insert into dim\_dia\_semana values (6, 'Sex') | |
| insert into dim\_dia\_semana values (7, 'Sab') | |

carga\_dim\_tempo.sql

### Dimensões de Geografia

#### Dim\_País

|  |  |
| --- | --- |
| Dimensão | Dim\_pais |
| Origem | Script de criação de dados |
| Execução | Apenas 1 vez |
|  | |
| insert into dim\_pais values (1, 'Brasil',getdate()) | |

#### Dim\_Regiao\_Geografica

|  |  |
| --- | --- |
| Dimensão | Dim\_regiao\_geografica |
| Origem | Script de criação de dados |
| Execução | Apenas 1 vez |
|  | |
| insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (1,'Sudeste', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (2,'Sul', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (3,'Norte', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (4,'Nordeste', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (5,'Centro-Oeste', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (6,'Leste', getdate(), 1)  insert into dim\_regiao\_geografica (id\_regiao\_geografica, ds\_regiao\_geografica, dt\_atualizacao, id\_pais) values (7,'Oeste', getdate(), 1) | |

#### Dim\_estado

|  |  |
| --- | --- |
| Dimensão | Dim\_estado |
| Origem | Script de criação de dados |
| Execução | Apenas 1 vez |
|  | |
| insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (1,'Acre','AC',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (2,'Alagoas','AL',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (3,'Amazonas','AM',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (4,'Amapá','AP',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (5,'Bahia','BA',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (6,'Ceará','CE',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (7,'Distrito Federal','DF',getdate(), 5)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (8,'Espirito Santo','ES',getdate(), 1)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (9,'Goiás','GO',getdate(), 5)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (10,'Maranhão','MA',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (11,'Minas Gerais','MG',getdate(), 1)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (12,'Mato Grosso do Sul','MS',getdate(), 5)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (13,'Mato Grosso','MT',getdate(), 5)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (14,'Pará','PA',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (15,'Paraíba','PB',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (16,'Pernambuco','PE',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (17,'Piauí','PI',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (18,'Paraná','PR',getdate(), 2)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (19,'Rio de Janeiro','RJ',getdate(), 1)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (20,'Rio Grande do Norte','RN',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (21,'Rondônia','RO',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (22,'Roraima','RR',getdate(), 3)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (23,'Rio Grande do Sul','RS',getdate(), 2)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (24,'Santa Catarina','SC',getdate(), 2)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (25,'Sergipe','SE',getdate(), 4)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (26,'São Paulo','SP',getdate(), 1)  insert into dim\_estado (id\_estado, ds\_estado, sg\_estado, dt\_atualizacao, id\_regiao\_geografica) values (27,'Tocantis','TO',getdate(), 3) | |

#### Dim\_municipio

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_municipio | |
| Origem | SELECT mun.id\_municipio  , mun.ds\_municipio  , mun.id\_estado  , est.sg\_estado  FROM mw\_municipio as mun  INNER JOIN mw\_estado AS est  ON est.id\_estado = mun.id\_estado | |
| Condição |  | |
| Regra do Negócio | LOOKUP 1:  DIM\_ESTADO.SG\_ESTADO = EST.SG\_ESTADO | |
| Execução | DIARIO | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_municipio |  |
| mun.ds\_municipio | Ds\_municipio |  |
| mun.id\_municipio | Cd\_municipio |  |
| LOOKUP 1: ID\_ESTADO | Id\_estado |  |
| Getdate() | Dt\_atualizacao |  |

### Dimensões da Empresa/Produto

#### Dim\_bandeira \_cartao

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_bandeira\_cartao | |
| Origem | SELECT id\_bandeira\_cartao  ,ds\_bandeira\_cartao  FROM mw\_bandeira\_cartao | |
| Condição |  | |
| Regra do Negócio |  | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_bandeira\_cartao |  |
| Ds\_bandeira\_cartao | Ds\_bandeira\_cartao |  |
| Id\_bandeira\_cartao | Cd\_ bandeira\_cartao |  |
| Getdate() | Dt\_atualizacao |  |

#### Dim\_patrocinador

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_patrocinador | |
| Origem | SELECT id\_Patrocinador  ,ds\_NomPatrocinador  FROM mw\_patrocinador | |
| Condição |  | |
| Regra do Negócio |  | |
| Execução | Diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_ patrocinador |  |
| Ds\_nompatrocinador | Ds\_nompatrocinador |  |
| Id\_patrocinador | Cd\_patrocinador |  |
| Getdate() | Dt\_ atualizacao |  |

#### Dim\_cartao\_patrocinado

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_cartao\_patrocinado | |
| Origem | SELECT cp.id\_cartao\_patrocinado  ,cp.id\_patrocinador  ,cp.ds\_cartao\_patrocinado  ,cp.cd\_bin  ,cp.id\_bandeira\_cartao  FROM mw\_cartao\_patrocinado as cp    inner join mw\_bandeira\_cartao as bc  on bc.id\_bandeira\_cartao = cp.id\_bandeira\_cartao  inner join mw\_patrocinador as p  on p.id\_patrocinador = cp.id\_patrocinador | |
| Condição |  | |
| Regra do Negócio | LOOKUP 1:  Dim\_bandeira\_cartao.cd\_bandeira\_cartao = BC.id\_bandeira\_cartao  LOOKUP 2:  Dim\_patrocinador.cd\_patrocinador = p.id\_patrocinador | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_cartao\_patrocinado |  |
| cp.ds\_cartao\_patrocinado | Ds\_cartao\_patrocinado |  |
| Cp.cd\_bin | Nr\_bin |  |
| Cp.id\_cartao\_patrocinado | Cd\_cartao\_patrocinado |  |
| LOOKUP 1: dim\_bandeira\_cartao.id\_bandeira\_cartao | Id\_bandeira\_cartao |  |
| LOOKUP 2: dim\_patrocinador.id\_patrocinador | Id\_patrocinador |  |
| Getdate() | Dt\_ atualizacao |  |

### Dimensão Cliente

#### Dim\_cliente

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_cliente | |
| Origem | Ci\_middleway  SELECT ds\_nome + ' ' + ds\_sobrenome as ds\_nome  ,ds\_ddd\_telefone  ,ds\_telefone  ,cd\_cpf  ,cd\_email\_login  FROM mw\_cliente  Ci\_????? (todas as bases ativas).  SELECT Nome  ,CPF  ,DDD  ,Telefone  ,EMail  FROM [ci\_coliseu].[dbo].[tabCliente]  Para identificar quais bases estão ativas.  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1 | |
| Condição |  | |
| Regra do Negócio | Vamos ter que ler todos os clientes cadastrados na ci\_middleway e todos os clientes de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  Iniciar a carga pela base middleway.mw\_cliente.  Para identificar quais as bases que teremos que ler, vamos ler a mw\_base (ci\_middleway) e selecionar todas as bases que tiverem ativas, conforme select no quadro de “Origem”.  **Importante:** Só iremos incluir um novo registro na dim\_cliente caso o cliente ainda não exista, para isto iremos utilizar o campo cd\_cpf (mw\_cliente) e CPF (tabcliente) como chave, relacionando com o campo da dim\_cliente.nr\_cpf. Caso o cliente exista, iremos atualizar (update) os dados retornados no select. | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_cliente |  |
| Ds\_nome (mw\_cliente)  Nome (tabcliente) | Ds\_nome |  |
| Ds\_ddd\_telefone (mw\_cliente)  DDD (tabcliente) | Ds\_DDD\_fone |  |
| Ds\_telefone (mw\_cliente)  Telefone (tabcliente) | Ds\_telefone |  |
| Cd\_cpf (mw\_cliente)  CPF (tabcliente) | Nr\_cpf |  |
| Cd\_email\_login (mw\_cliente)  Email (tabcliente) | Ds\_email |  |
| Getdate() | Dt\_ atualizacao |  |

### Dimensão Evento

#### Dim\_tipo\_local

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_tipo\_local | |
| Origem | SELECT id\_tipo\_local  ,ds\_tipo\_local  FROM mw\_tipo\_local | |
| Condição |  | |
| Regra do Negócio |  | |
| Execução | Diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_ tipo\_local |  |
| Ds\_tipo\_local | Ds\_tipo\_local |  |
| Id\_tipo\_local | Cd\_tipo\_local |  |
| Getdate() | Dt\_ atualizacao |  |

#### Dim\_canal\_venda

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_canal\_venda | |
| Origem | SELECT id\_canal\_venda  ,ds\_canal\_venda  FROM mw\_canal\_venda | |
| Condição |  | |
| Regra do Negócio |  | |
| Execução | Diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_ canal\_venda |  |
| Ds\_canal\_venda | Ds\_canal\_venda |  |
| Id\_canal\_venda | Cd\_canal\_venda |  |
| Getdate() | Dt\_ atualizacao |  |

#### Dim\_local\_evento

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_local\_evento | |
| Origem | SELECT le.id\_local\_evento  ,le.ds\_local\_evento  ,le.id\_tipo\_local  ,le.id\_municipio  FROM mw\_local\_evento as le  inner join mw\_municipio as mun  on mun.id\_municipio = le.id\_municipio    inner join mw\_tipo\_local as tl  on tl.id\_tipo\_local = le.id\_tipo\_local | |
| Condição |  | |
| Regra do Negócio | LOOKUP 1:  DIM\_MUNICIPIO.CD\_MUNICIPIO = LE.ID\_MUNICIPIO  LOOKUP 2:  DIM\_TIPO\_LOCAL.CD\_TIPO\_LOCAL =T L.ID\_TIPO\_LOCAL | |
| Execução | DIARIO | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_local\_evento |  |
| Le.ds\_local\_evento | Ds\_local\_evento |  |
| Le.id\_local\_evento | Cd\_local\_evento |  |
| LOOKUP 1: ID\_MUNICIPIO | Id\_municipio |  |
| LOOKUP 2: ID\_TIPO\_LOCAL | Id\_tipo\_local |  |
| Getdate() | Dt\_atualizacao |  |

#### Dim\_tipo\_evento

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim\_tipo\_evento | |
| Origem | Para identificar quais bases estão ativas, utilizar o select abaixo:  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1  Ci\_????? (todas as bases ativas).  SELECT CodTipPeca  ,TipPeca  FROM [ci\_coliseu].[dbo].[tabTipPeca]  WHERE StaTipPeca = 'A' | |
| Condição |  | |
| Regra do Negócio | Definição: As tabelas que estão com a nomenclatura de “PEÇAS” no TSP, foram renomeadas para “EVENTOS”, sendo assim, toda vez que falarmos de PEÇAS e EVENTOS estaremos falando do mesmo conteúdo.  Vamos ter que ler todas as peças de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  Para identificar quais as bases que teremos que ler, vamos ler a mw\_base (ci\_middleway) e selecionar todas as bases que tiverem ativas, conforme select no quadro de “Origem”.  **Importante:** Só iremos incluir um novo registro na dim\_tipo\_evento caso o tipo de evento ainda não exista, para isto iremos utilizar o campo TipPeca (tabTipPeca) como chave, relacionando com o campo da dim\_tipo\_evento.ds\_tipo\_evento. Caso o evento exista, iremos desprezar o resgitro. | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_tipo\_evento |  |
| TipPeca | Ds\_tipo\_evento |  |
| Getdate() | Dt\_ atualizacao |  |

#### Dim\_evento

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim \_evento | |
| Origem | Para identificar quais bases estão ativas, utilizar o select abaixo:  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1  Ci\_????? (todas as bases ativas).  SELECT P.CodPeca  ,P.NomPeca  ,P.DatIniPeca  ,P.DatFinPeca  ,P.id\_local\_evento  ,TP.TipPeca  FROM ci\_coliseu.dbo.tabPeca P  INNER JOIN TabTipPeca as TP  on TP.CodTipPeca = P.CodTipPeca  WHERE P.StaPeca = 'A' | |
| Condição | LOOKUP 1:  DIM\_TIPO\_EVENTO. DS\_TIPO\_EVENTO = TP.TIPPECA  LOOKUP 2:  DIM\_LOCAL\_EVENTO.CD\_LOCAL\_EVENTO = P.ID\_LOCAL\_EVENTO | |
| Regra do Negócio | Vamos ter que ler todas as peças de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  Para identificar quais as bases que teremos que ler, vamos ler a mw\_base (ci\_middleway) e selecionar todas as bases que tiverem ativas, conforme select no quadro de “Origem”.  **Importante:** Só iremos incluir um novo registro na dim\_evento caso o evento ainda não exista, para isto iremos utilizar o campo CodPeca (tabPeca) como chave e o nome da Base que está sendo lida (mw\_base. ds\_nome\_base\_sql), relacionando com o campo da dim \_evento.cd \_evento e dim\_evento.ds\_base. Caso o evento exista, iremos atualizar os campos que retornarem no select. | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_evento |  |
| P.NomPeca | Ds\_evento |  |
| P.DatInipeca | Dt\_inicio\_evento |  |
| P.DatFinipeca | Dt\_final\_evento |  |
| P.CodPeca | Cd\_evento |  |
| Ds\_nome\_base\_sql | Ds\_base |  |
| LOOKUP 1: Id\_tipo\_evento | Id\_tipo\_evento |  |
| LOOKUP 2: Id\_local\_evento | Id\_local\_evento |  |
| Getdate() | Dt\_ atualizacao |  |

#### Dim\_apresentacao

|  |  |  |
| --- | --- | --- |
| Dimensão | Dim \_apresentacao | |
| Origem | Para identificar quais bases estão ativas, utilizar o select abaixo:  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1  Ci\_????? (todas as bases ativas).  SELECT A.CodApresentacao  ,A.DatApresentacao  ,A.HorSessao  ,P.NomPeca  FROM ci\_coliseu.dbo.tabApresentacao as A  inner join tabpeca as P  on p.codpeca = a.codpeca | |
| Condição | LOOKUP 1:  DIM\_EVENTO. DS\_EVENTO = P.NOMPECA | |
| Regra do Negócio | Vamos ter que ler todas as apresentações de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  Para identificar quais as bases que teremos que ler, vamos ler a mw\_base (ci\_middleway) e selecionar todas as bases que tiverem ativas, conforme select no quadro de “Origem”.  **Importante:** Só iremos incluir um novo registro na dim\_apresentacao caso a apresentação ainda não exista, para isto iremos utilizar o campo CodApresentacao (tabApresentacao) como chave, relacionando com o campo da dim \_apresentacao.id \_apresentacao. Caso o evento exista, iremos atualizar os campos que retornarem no select. | |
| Execução | Diário | |
|  |  |  |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Autonumeracao | Id\_apresentacao |  |
| A.DatApresentacao | Dt\_apresentacao |  |
| A.HorSessao | Hr\_sessao |  |
| A.CodApresentacao | Cd\_apresentacao |  |
| LOOKUP 1: Id \_evento | Id \_evento |  |
| Getdate() | Dt\_ atualizacao |  |

## Fatos

#### Fato\_vendas\_promocionais\_itau

|  |  |  |  |
| --- | --- | --- | --- |
| Dimensão | Fato\_vendas\_promocionais\_itau | | |
| Origem | Para identificar quais bases estão ativas, utilizar o select abaixo:  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1  Vamos ter que ler todas as vendas de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  with resultado as (select distinct  tl.numlancamento,  convert(varchar(8), tl.datvenda ,112) as ID\_DIA,  cp.id\_cartao\_patrocinado,  'ci\_coliseu' as nm\_base,  c.codpeca,  c.codapresentacao,  c.CPF,  c.DDD,  c.Telefone,  left(i.bincartao,6) as BIN,  c.codvenda,  p.codtipbilheteBIN,  tc.id\_canal\_venda  from  ci\_coliseu..tablancamento tl  inner join  ci\_coliseu..tabcaixa tc  on tc.codcaixa = tl.codcaixa  inner join  ci\_coliseu..tablugsala ls  on ls.codapresentacao = tl.codapresentacao  and ls.indice = tl.indice  inner join  ci\_coliseu..tabcomprovante c  on c.codvenda = ls.codvenda  inner join  ci\_coliseu..tabpeca p  on p.codpeca = c.codpeca  inner join  ci\_coliseu..tabingresso i  on i.codvenda = c.codvenda  and i.indice = tl.indice  and i.bincartao is not null  inner join  ci\_middleway..mw\_cartao\_patrocinado cp  on cp.id\_cartao\_patrocinado = i.id\_cartao\_patrocinado  where  convert(varchar(8), tl.datvenda ,112) = convert(varchar(8), dateadd(d,-1,getdate()),112)  and tl.codtiplancamento = 1  and not exists (Select 1 from ci\_coliseu..tablancamento tl2  where tl2.numlancamento = tl.numlancamento  and tl2.codtipbilhete = tl.codtipbilhete  and tl2.codtiplancamento = 2  and tl2.codapresentacao = tl.codapresentacao  and tl2.indice = tl.indice) )  select  r.ID\_DIA,  r.id\_cartao\_patrocinado,  r.nm\_base,  r.codpeca,  r.codapresentacao,  r.CPF,  r.DDD,  r.Telefone,  r.BIN,  r.codvenda,  r.id\_canal\_venda,  sum(case when tl.codtipbilhete = r.codtipbilheteBIN then 1 else 0 end) as qt\_bilhete\_promocional,  sum(case when tl.codtipbilhete <> r.codtipbilheteBIN and isnull(tb.perdesconto,0) = 0 then 1 else 0 end) as qt\_bilhete\_inteira,  sum(case when tl.codtipbilhete <> r.codtipbilheteBIN and isnull(tb.perdesconto,0) > 0 then 1 else 0 end) as qt\_bilhete\_meia,  sum(case when tl.codtipbilhete = r.codtipbilheteBIN then i.ValPagto else 0 end) as vl\_bilhete\_promocional,  sum(case when tl.codtipbilhete <> r.codtipbilheteBIN and isnull(tb.perdesconto,0) = 0 then i.ValPagto else 0 end) as vl\_bilhete\_inteira,  sum(case when tl.codtipbilhete <> r.codtipbilheteBIN and isnull(tb.perdesconto,0) > 0 then i.ValPagto else 0 end) as vl\_bilhete\_meia  from  resultado r  inner join  ci\_coliseu..tablancamento tl  on tl.numlancamento = r.numlancamento  inner join  ci\_coliseu..tablugsala ls  on ls.codapresentacao = tl.codapresentacao  and ls.indice = tl.indice  inner join  ci\_coliseu..tabcomprovante c  on c.codvenda = ls.codvenda  inner join  ci\_coliseu..tabingresso i  on i.codvenda = c.codvenda  and i.indice = tl.indice  inner join  ci\_coliseu..tabtipbilhete tb  on tb.codtipbilhete = tl.codtipbilhete  group by  r.ID\_DIA,  r.id\_cartao\_patrocinado,  r.nm\_base,  r.codpeca,  r.codapresentacao,  r.CPF,  r.DDD,  r.Telefone,  r.BIN,  r.codvenda,  r.id\_canal\_venda | | |
| Condição |  | | |
| Regra do Negócio | LOOKUP 1:  DIM\_CARTAO\_PATROCINADO. CD\_CARTAO\_PATROCINADO = R.ID-CARTAO\_PATROCINADO  LOOKUP2:  DIM\_EVENTO.CD\_EVENTO = R.CODPECA AND DIM\_EVENTO.DS\_BASE = MW\_BASE.DS\_NOME\_BASE\_SQL  LOOKUP3:  DIM\_APRESENTACAO.CD\_APRESENTACAO = R.CODAPRESENTACAO  LOOKUP4:  DIM\_CLIENTE.NR\_CPD = R.CPF  LOOKUP5:  DIM\_CANAL\_VENDA.CD\_CANAL\_VENDA = R.ID\_CANAL\_VENDA | | |
| Execução | diário | | |
|  | | | |
| Coluna Origem | | **Coluna Destino** | **Transformação** |
| R.Id\_dia | | Id\_dia |  |
| LOOKUP1:Id\_cartao\_patrocinado | | Id\_cartao\_patrocinado |  |
| LOOKUP2:id\_evento | | Id\_evento |  |
| LOOKUP3:id\_apresentacao | | Id\_apresentacao |  |
| LOOKUP4:id\_cliente | | Id\_cliente |  |
| LOOKUP5:Id\_canal\_venda | | Id\_canal\_venda |  |
| qt\_bilhete\_promocional | | Qt\_ingresso\_promocional |  |
| qt\_bilhete\_inteira | | Qt\_ingresso\_inteira |  |
| qt\_bilhete\_meia | | Qt\_ingresso\_meia |  |
| vl\_bilhete\_promocional | | Vr\_total\_ingressos\_promocionais |  |
| vl\_bilhete\_inteira | | Vr\_total\_ingressos\_inteira |  |
| vl\_bilhete\_meia | | Vr\_total\_ingressos\_meia |  |
| R.DDD | | Ds\_DDD\_fone |  |
| R.tefefone | | Ds\_telefone |  |
| R.Bin | | Nr\_bin |  |

#### Fato\_evento\_mes

|  |  |  |
| --- | --- | --- |
| Dimensão | FATO\_EVENTO\_MES | |
| Origem | Para identificar quais bases estão ativas, utilizar o select abaixo:  Select ds\_nome\_base\_sql  from mw\_base  where in\_ativo = 1  Vamos ter que ler todos os eventos de todas as bases do compreingresso (ci\_coliseu, ci\_vivo, ci\_marejada,...).  Este select foi desabilitado, utilizar abaixo  select distinct  'ci\_coliseu' as nm\_base,  p.codpeca,  convert(varchar(6), dateadd(d,-1, getdate()),112) as ID\_MES  from  ci\_coliseu..tabpeca p  inner join  ci\_coliseu..tabapresentacao a  on a.codpeca = p.codpeca  where  convert(varchar(6), dateadd(d,-1, getdate()) ,112) = convert(varchar(6), a.DatApresentacao,112)  Novo select atualizado 24-03-2011  select distinct (e.id\_evento),  convert(varchar(6), dateadd(d,-1, getdate()),112) as ID\_MES  from  mw\_evento e  inner join  mw\_apresentacao a  on a.id\_evento = e.id\_evento  where  convert(varchar(6), dateadd(d,-1, getdate()) ,112) = convert(varchar(6), a.Dt\_Apresentacao,112)  and e.in\_vende\_itau = 1 and e.in\_ativo = 1 | |
| Condição | Excluir os dados da Fato antes de fazer o ETL  delete fato\_evento\_mes where id\_mes = convert(varchar(6), dateadd(d,-1, getdate()),112) | |
| Regra do Negócio | LOOKUP 1:  Dim\_evento.cd\_evento = p.codpeca | |
| Execução | diário | |
|  | | |
| Coluna Origem | **Coluna Destino** | **Transformação** |
| Id\_mes | Id\_mes |  |
| LOOKUP 1: id\_evento | Id\_evento |  |
| 1 | qt\_count |  |

## Ordem de execução dos ETLs

|  |  |  |
| --- | --- | --- |
| Ordem | ETL | Package Name |
| 1 | Dim \_municipio | 01-Dim\_municipio.dtsx |
| 2 | Dim\_bandeira\_cartao | 02-Dim\_bandeira\_cartao.dtsx |
| 3 | Dim\_patrocinador | 03-Dim\_patrocinador.dtsx |
| 4 | Dim\_cartao\_patrocinado | 04-Dim\_cartao\_patrocinado.dtsx |
| 5 | Dim\_cliente | 05-Dim\_cliente.dtsx |
| 6 | Dim\_tipo\_local | 06-Dim\_tipo\_local.dtsx |
| 7 | Dim\_canal\_venda | 07-Dim\_canal\_venda.dtsx |
| 8 | Dim\_local\_evento | 08-Dim\_local\_evento.dtsx |
| 9 | Dim\_tipo\_evento | 09-Dim\_tipo\_evento.dtsx |
| 10 | Dim \_evento | 10-Dim\_evento.dtsx |
| 11 | Dim \_apresentacao | 11-Dim\_apresentacao.dtsx |
| 12 | Fato\_vendas\_promocionais\_itau | 12-Fato\_vendas\_promocionais\_itau.dtsx |
| 13 | Fato\_evento\_mes | 13-Fato\_evento\_mes.dtsx |

## Alterações em bases

Scripts que devem ser executados em produção

ALTER TABLE TABPECA ADD id\_local\_evento INT

ALTER TABLE TABPECA ADD codtipbilheteBIN TINYINT

ALTER TABLE TABINGRESSO ADD id\_cartao\_patrocinado INT

ALTER TABLE TABCAIXA ADD id\_canal\_venda INT