



UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPTO. DE INFORMÁTICA E ESTATÍSTICA

CURSO DE SISTEMAS DE INFORMAÇÃO

DATA WAREHOUSE

PROF. JOSÉ LEOMAR TODESCO

Trabalho Final de Data Warehouse

Data Mart do Vestibular da UFSC de 2010 a 2012

Emerson Demetrio Plácido

Florianópolis, 03 de Dezembro de 2015

1 RESUMO

Este trabalho visa analisar os candidatos e resultados dos vestibulares realizados pela UFSC nos anos de 2010, 2011 e 2012, assume-se um papel de uma empresa de recrutamento interessada em buscar talentos na área da engenharia, física e química. Para tanto, são feitas análises de desempenho individual principalmente nas áreas exatas (matemática, física e química), afim de financiar bolsas de estudo em universidades estrangeiras. O escopo se limitará aos alunos que tiveram melhor desempenho nos anos de 2010, 2011 e 2012. Serão feitos também filtros por renda e localização, origem do candidato e a renda per capita de sua família. Os cursos do nosso interesse são física (bacharelado), química, engenharia química e engenharia de materiais.

Palavras-chave: vestibular, UFSC, data warehouse, data mart, formulário socioeconômico, Universidade Federal de Santa Catarina, Comissão Permanente do Vestibular, COPERVE.

2 INTRODUÇÃO

Atualmente, as pessoas que pretendem cursar alguma faculdade na Universidade Federal de Santa Catarina precisam realizar o UFSC/COPERVE.

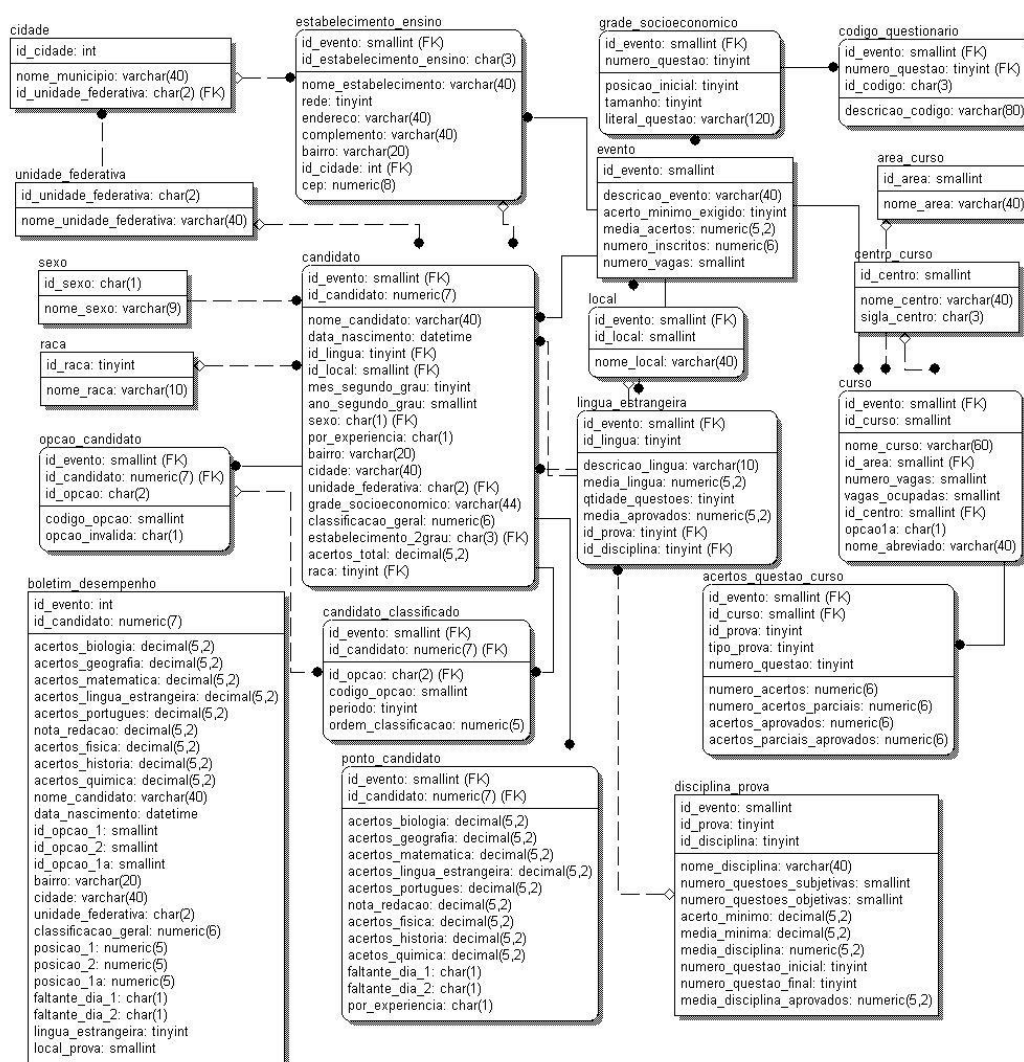
A concorrência nos cursos de engenharia é extremamente alta e além disso, as disciplinas exatas sempre são barreiras para os candidatos.

Este trabalho tem por objetivo buscar a relação de alunos que se saíram bem nas disciplinas de exatas, tais como química, física, matemática e, além disso, língua estrangeira (especialmente, inglês). Uma vez que serão ofertadas bolsas de estudo aos candidatos que foram melhor nas disciplinas acima, será feita uma análise no banco de dados oferecido pela disciplina afim de filtrar os resultados. A seguir, mostra-se o passo a passo para a construção do *data-mart* que será utilizado para transformar os dados da COPERVE em informações, abordando as etapas de planejamento, projeto, modelagem dimensional, perguntas estratégicas, ETL e front-end.

3 OBTENÇÃO DOS DADOS

Os dados utilizados neste trabalho foram disponibilizados pelo professor da disciplina em parceria com a COPERVE/UFSC. É importante ressaltar que os eventos analisados serão aqueles que aconteceram nos anos de 2010, 2011 e 2012. Os dados do vestibular são obtidos através de preenchimento de um cadastro socioeconômico por parte dos alunos, no momento da inscrição. Os formulários não interferem na classificação/admissão dos candidatos. O objetivo principal destes formulários é a compreensão dos aspectos socioeconômico e culturais dos candidatos.

4 MODELO RELACIONAL (COPERVE)



5 MÉTODOS

Para cruzamento dos dados segundo a especificação definida, optou-se por utilizar um *data-mart*, (um nodo de um projeto *data wharehouse*), buscando uma base robusta e confiável para obtenção dos dados e para análise de resultados. O ferramental e as técnicas utilizadas são descritas em seguida.

6 METODOLOGIA

Para construção do data-mart, são necessários os seguintes passos:

1. Planejamento do projeto
2. Administração do projeto
3. Definição dos requisitos de negócio
4. Modelagem dimensional
5. Projeto físico
6. Desenvolvimento e projeto da área de transição
7. Especificação da aplicação do usuário final (Front Room)
8. Desenvolvimento da aplicação do usuário final (Front Room)
9. Projeto e arquitetura técnica
10. Instalação e seleção de produtos
11. Implantação e manutenção

Este trabalho foca nas etapas de planejamento, análise, modelagem dimensional, projeto físico, ETL e front-end.

6.1 Planejamento

Escopo: traçar um perfil dos candidatos dos vestibulares de 2010 a 2012 da UFSC buscando talentos nas áreas exatas (matemática, física e química) e como eles dividem-se dentro das opções oferecidas pela UFSC.

Justificativa: O trabalho servirá para gerar informações relacionadas ao perfil do candidato, a fim de premiar aqueles que melhor se saíram com bolsas de estudo.

Exclusões: Candidatos que prestaram o vestibular por experiência; Todas as avaliações dos anos que não estão na relação.

Riscos: Dados fornecidos podem apresentar inconsistências ou defeitos.

Fatores críticos para o sucesso: desenvolver um *data-mart* que responda as perguntas propostas abaixo.

6.2 Análise e Perguntas

Com o objetivo de gerar informações sobre os candidatos, instituições de ensino e desempenho no vestibular, foram levantadas as seguintes questões para análise:

1. Qual é o número total de candidatos aprovados e reprovados por tipo de rede de ensino e por ano?
2. Qual é o percentual de acertos nas áreas exatas dividindo por origem socioeconômica?
3. Qual é a média de nota das disciplinas das áreas exatas por origem socioeconômica?
4. Quem são os candidatos que melhor se saíram nas áreas exatas?
5. Qual é a renda familiar dos candidatos que melhor se saíram nas áreas exatas?

6.3 Modelagem dimensional

A modelagem dimensional foi definida conforme a especificação da disciplina, sendo:

- **Definição do Processo de negócio:**

O projeto teve como objetivo analisar dados de desempenho e origem socioeconômica dos candidatos.

- **Definição do grão:**

Para poder responder às nossas perguntas estratégicas, foi decidido como grão o desempenho anual de cada candidato em cada matéria.

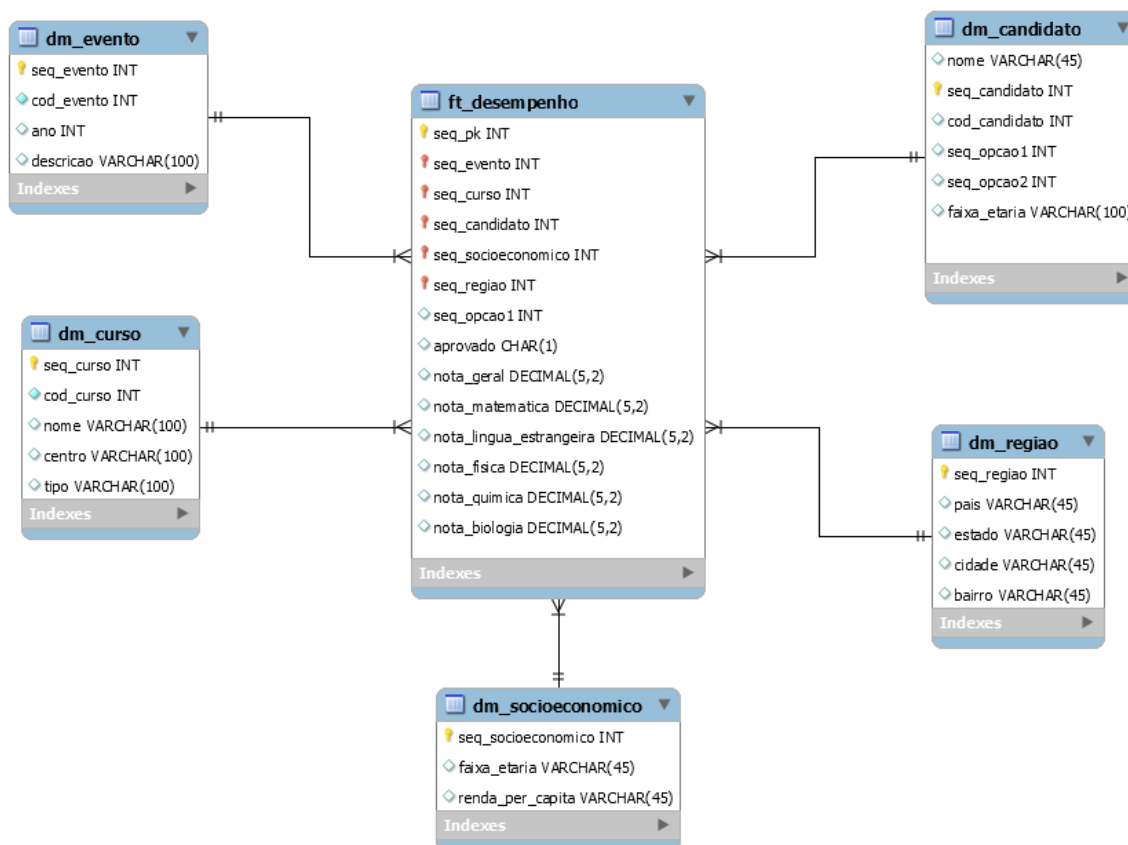
- **Definição do fato:**

O nosso fato “ft_desempenho”, nele é discriminado cada desempenho em cada matéria, e nota total. Também contém uma chave estrangeira (fk) para relacionamento com as dimensões.

- **Definição das dimensões:**

O modelo proposto conta com cinco dimensões:

1. **Dimensão Candidato (*dm_evento*):**
Contém dados sobre o evento vestibular de cada ano.
2. **Dimensão Curso (*dm_curso*)**
Contém as informação sobre o curso escolhido pelo vestibulando.
3. **Dimensão Socioeconômico (*dm_socioeconomico*):**
Contém dados sobre os cadastros socioeconômicos dos candidatos.
4. **Dimensão Candidato (*dm_candidato*):**
Contém as representações das informações gerais de um candidato.
5. **Dimensão Curso (*dm_regiao*)**
Contém as informação sobre o curso escolhido pelo vestibulando.



7 PROJETO FÍSICO

O projeto físico foi implementando no bando de dados MySql.
O Script de criação está disponível no anexo 1.

FERRAMENTAS UTILIZADAS

Para modelagem física, dimensional, ETL, back-end e front-end foram utilizadas as seguintes ferramentas:

Modelagem: MySQL Workbench

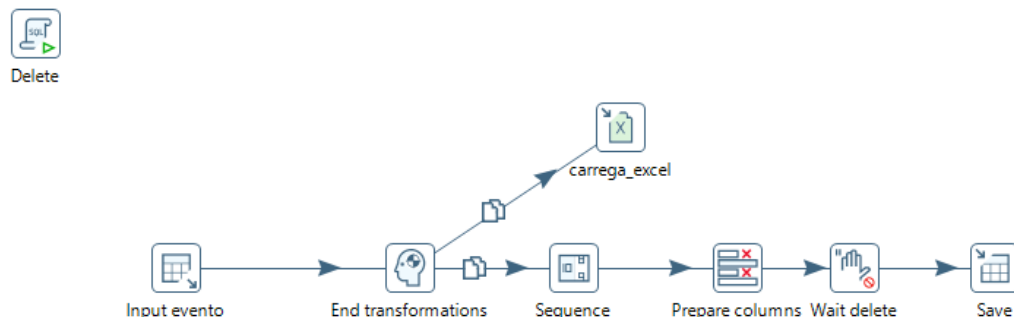
Carregamento do projeto físico, avaliação e validação da modelagem física e dimensional: Navicat

Extração, *back-end*, transformações e carga de dados: Kettle/ Pentaho BI CE.

Front-End: Tableau

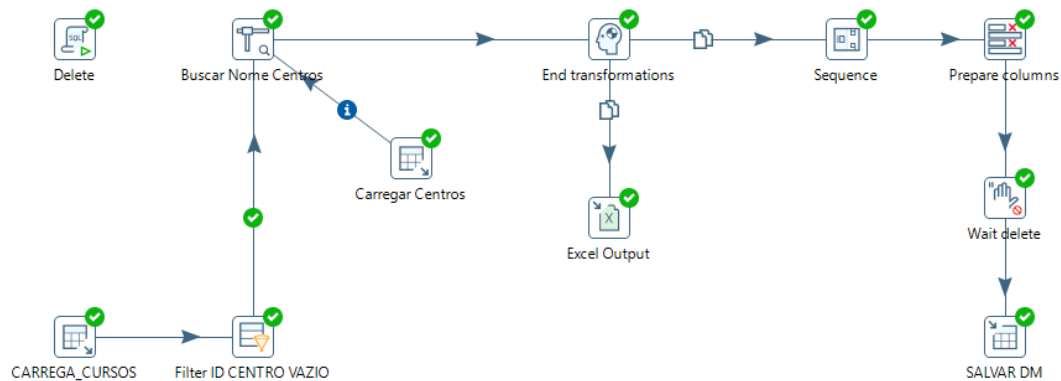
Para o ETL, foram criadas as seguintes dimensões:

Dimensão Evento:



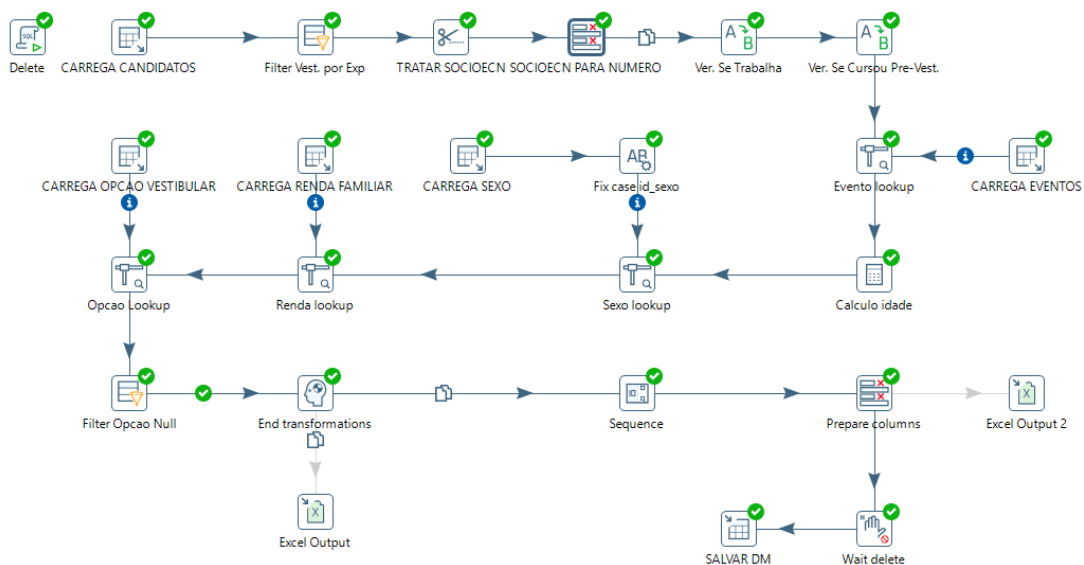
Nesta dimensão, são carregados os eventos cujos anos interessam na modelaegem (2009, 2010 e 2011).

Dimensão Curso:



Nesta dimensão, são carregados os valores padrão dos cursos para serem “linkados” depois com os usuários e demais entidades. Observe a filtragem presente nos nodos e a saída em excel, que serviu como *debug*. Por fim, a saída “SALVAR DM” conecta no banco “dwtf_tb” e salva os dados filtrados.

Dimensão Candidato:



Nesta dimensão...

8 FRONT-END

=====

9 RESULTADOS

9.1 DESAFIOS

9.2 DIFICULDADES ENCONTRADAS

9.3 RESULTADOS

=====

10 VERSIONAMENTO

Embora não fosse parte da especificação do trabalho, foi criado um repositório com as diversas versões do trabalho, assim como a sua evolução, permitindo assim um acompanhamento do mesmo. Como trata-se de um trabalho acadêmico, o acesso é público, disponível no endereço a seguir:

< <https://github.com/emersondemetrio/20152dw> >

11 CONCLUSÕES

12 ANEXOS

Anexo 1 – Script de Criação do projeto físico:

```
-- MySQL Script generated by MySQL Workbench
-- 12/02/15 19:17:56
-- Model: New Model      Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----
-- Schema dwtf_db
-- -----
-- Trabalho Final - Data Warehouse UFSC
-- -----
-- Schema dwtf_db
--
-- Trabalho Final - Data Warehouse UFSC
-- -----

CREATE SCHEMA IF NOT EXISTS `dwtf_db` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `dwtf_db` ;

-- -----
-- Table `dwtf_db`.`dm_evento`
-- -----
CREATE TABLE IF NOT EXISTS `dwtf_db`.`dm_evento` (
  `seq_evento` INT NOT NULL COMMENT '',
  `cod_evento` INT NOT NULL COMMENT '',
  `ano` INT NULL COMMENT '',
  `descricao` VARCHAR(100) NULL COMMENT '',
  PRIMARY KEY (`seq_evento`) COMMENT '')
ENGINE = InnoDB;

-- -----
-- Table `dwtf_db`.`dm_candidato`
-- -----
CREATE TABLE IF NOT EXISTS `dwtf_db`.`dm_candidato` (
  `seq_candidato` INT NOT NULL COMMENT '',
  `cod_candidato` INT NULL COMMENT '',
  `seq_opcao1` INT NULL COMMENT '',
```

```

`seq_opcao2` INT NULL COMMENT '',
`sexo` CHAR(1) NULL COMMENT '',
`trabalha` CHAR(1) NULL COMMENT '',
`pre_vest` CHAR(1) NULL COMMENT '',
`idade` CHAR(2) NULL COMMENT '',
PRIMARY KEY (`seq_candidato`) COMMENT '')
ENGINE = InnoDB;

-----
-- Table `dwtf_db`.`dm_curso`
-----

CREATE TABLE IF NOT EXISTS `dwtf_db`.`dm_curso` (
  `seq_curso` INT NOT NULL COMMENT '',
  `cod_curso` INT NOT NULL COMMENT '',
  `nome` VARCHAR(100) NULL COMMENT '',
  `centro` VARCHAR(100) NULL COMMENT '',
  PRIMARY KEY (`seq_curso`) COMMENT '')
ENGINE = InnoDB;

-----
-- Table `dwtf_db`.`dm_socioeconomico`
-----

CREATE TABLE IF NOT EXISTS `dwtf_db`.`dm_socioeconomico` (
  `seq_socioeconomico` INT NOT NULL COMMENT '',
  `faixa_etaria` VARCHAR(45) NULL COMMENT '',
  `renda_per_capita` VARCHAR(45) NULL COMMENT '',
  PRIMARY KEY (`seq_socioeconomico`) COMMENT '')
ENGINE = InnoDB;

-----
-- Table `dwtf_db`.`dm_regiao`
-----

CREATE TABLE IF NOT EXISTS `dwtf_db`.`dm_regiao` (
  `seq_regiao` INT NOT NULL COMMENT '',
  `pais` VARCHAR(45) NULL COMMENT '',
  `estado` VARCHAR(45) NULL COMMENT '',
  `cidade` VARCHAR(45) NULL COMMENT '',
  `bairro` VARCHAR(45) NULL COMMENT '',
  PRIMARY KEY (`seq_regiao`) COMMENT '')
ENGINE = InnoDB;

-----
-- Table `dwtf_db`.`ft_desempenho`
-----

CREATE TABLE IF NOT EXISTS `dwtf_db`.`ft_desempenho` (

```

```

`seq_pk` INT NOT NULL COMMENT '',
`seq_evento` INT NOT NULL COMMENT '',
`seq_curso` INT NOT NULL COMMENT '',
`seq_candidato` INT NOT NULL COMMENT '',
`seq_socioeconomico` INT NOT NULL COMMENT '',
`seq_regiao` INT NOT NULL COMMENT '',
`seq_opcao1` INT NULL COMMENT '',
`aprovado` CHAR(1) NULL COMMENT '',
`nota_geral` DECIMAL(5,2) NULL COMMENT '',
`nota_matematica` DECIMAL(5,2) NULL COMMENT '',
`nota_lingua_estrangeira` DECIMAL(5,2) NULL COMMENT '',
`nota_fisica` DECIMAL(5,2) NULL COMMENT '',
`nota_quimica` DECIMAL(5,2) NULL COMMENT '',
`nota_biologia` DECIMAL(5,2) NULL COMMENT '',
PRIMARY KEY (`seq_pk`, `seq_evento`, `seq_curso`, `seq_candidato`,
`seq_socioeconomico`, `seq_regiao`) COMMENT '',
INDEX `fk_ft_desempenho_dm_evento_idx` (`seq_evento` ASC) COMMENT
'',
INDEX `fk_ft_desempenho_dm_cursor1_idx` (`seq_curso` ASC) COMMENT
'',
INDEX `fk_ft_desempenho_dm_candidato1_idx` (`seq_candidato` ASC)
COMMENT '',
INDEX `fk_ft_desempenho_dm_socioeconomico1_idx`
(`seq_socioeconomico` ASC) COMMENT '',
INDEX `fk_ft_desempenho_dm_regiao1_idx` (`seq_regiao` ASC) COMMENT
'',
CONSTRAINT `fk_ft_desempenho_dm_evento`
FOREIGN KEY (`seq_evento`)
REFERENCES `dwtf_db`.`dm_evento` (`seq_evento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_ft_desempenho_dm_cursor1`
FOREIGN KEY (`seq_curso`)
REFERENCES `dwtf_db`.`dm_curso` (`seq_curso`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_ft_desempenho_dm_candidato1`
FOREIGN KEY (`seq_candidato`)
REFERENCES `dwtf_db`.`dm_candidato` (`seq_candidato`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_ft_desempenho_dm_socioeconomico1`
FOREIGN KEY (`seq_socioeconomico`)
REFERENCES `dwtf_db`.`dm_socioeconomico` (`seq_socioeconomico`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_ft_desempenho_dm_regiao1`
FOREIGN KEY (`seq_regiao`)
REFERENCES `dwtf_db`.`dm_regiao` (`seq_regiao`)

```

```
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```