

Implementação de Banco de Dados em Diário de Classe com MySQL E PyQT

José Victor Viriato, Gabriel Haddad

¹Departamento de Informática – Universidade Federal de Santa Maria (UFSM)

{jvviriato, ghvieira}@inf.ufsm.br

Abstract. *This paper exposes the work of UFSM's Computer Science students on the development of a database for an university that needs to computerize its students frequency control. The most important informations, such as functions, triggers, tools, tables and views will be exposed and explained during the paper.*

Resumo. *Este artigo expõe o trabalho de alunos do Curso de Ciência da Computação da UFSM no desenvolvimento de um banco de dados para uma universidade que necessita informatizar no controle de frequência de seus alunos. As informações mais importantes, como funções, triggers, ferramentas, tabelas e views serão expostas e explicadas durante o artigo.*

1. Introdução

O presente artigo apresenta um projeto desenvolvido pelos alunos Gabriel Haddad e José Viriato da Universidade Federal de Santa Maria, docentes do curso de Ciência da Computação. O texto relata em detalhes a ideia básica do projeto, bem como as ferramentas utilizadas ao longo deste, as funções, tabelas, trigger, views e diagrama desenvolvidos.

2. Problema Inicial

O objetivo final do artigo era de conseguir solidificar a ideia de um banco de dados para uma universidade que deseja informatizar seu diário de frequência dos alunos. O banco teria como tabelas: Alunos, Disciplinas, Cursos, Turmas, Professores e Frequência. Desse modo, as tabelas foram criadas e bem como, algumas, interligadas.

3. Ferramentas Utilizadas

Durante a realização do presente trabalho, diversas ferramentas foram utilizadas para completá-lo. Inicialmente a plataforma GitHub[1] foi de extrema necessidade para a dupla, na qual frequentemente atualizava-se os arquivos do trabalho bem como acompanhava seu progresso. Para a Figura 1 - Diagrama de Classe do Banco de Dados - foi utilizada a ferramenta Astah[2]. MySQL, MySQL Workbench[3], Python3[4] juntamente com PyCharm[5] e PyMySQL surgiram a seguir, como itens fundamentais para o projeto. Na sequência, para a construção das interfaces das tabelas, o QtDesigner[6] juntamente com o PyQt foram necessários. Por fim, para a construção do presente artigo, a ferramenta LaTeX[7] foi utilizada.

4. Implementação do Banco de Dados

4.1. Diagrama de Classe

O Diagrama de Classe abaixo (Figura 1) ilustra todas as tabelas presentes no banco de dados que o presente artigo descreve, bem como suas chaves e chaves estrangeiras.

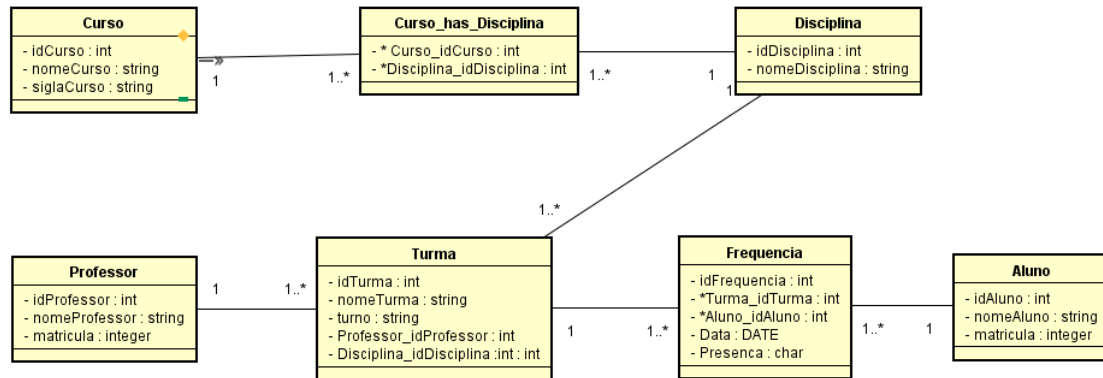


Figura 1. Diagrama de Classe

4.2. Tabelas

```
mysql> describe Aluno;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idAluno    | int(11)   | NO   | PRI | NULL    | auto_increment |
| nomeAluno  | varchar(45) | NO   |     | NULL    |              |
| matricula  | int(11)   | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Curso;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idCurso    | int(11)   | NO   | PRI | NULL    | auto_increment |
| nomeCurso  | varchar(45) | NO   |     | NULL    |              |
| siglaCurso | varchar(10) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Curso_has_Disciplina;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| Curso_idCurso | int(11) | NO   | PRI | NULL    |              |
| Disciplina_idDisciplina | int(11) | NO   | PRI | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Professor;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idProfessor | int(11)   | NO   | PRI | NULL    | auto_increment |
| nomeProfessor | varchar(45) | NO   |     | NULL    |              |
| matricula  | int(11)   | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Turma;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idTurma    | int(11)   | NO   | PRI | NULL    | auto_increment |
| nomeTurma  | varchar(45) | NO   |     | NULL    |              |
| turno     | varchar(45) | NO   |     | NULL    |              |
| Professor_idProfessor | int(11) | NO   | MUL | NULL    |              |
| Disciplina_idDisciplina | int(11) | NO   | MUL | NULL    |              |
| Data      | date      | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Disciplina;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idDisciplina | int(11)   | NO   | PRI | NULL    | auto_increment |
| nomeDisciplina | varchar(45) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+

mysql> describe Frequencia;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idFrequencia | int(11)   | NO   | PRI | NULL    | auto_increment |
| Turma_idTurma | int(11)   | NO   | PRI | NULL    |              |
| Aluno_idAluno | int(11)   | NO   | PRI | NULL    |              |
| Data      | date      | YES  |     | NULL    |              |
| Presenca  | varchar(2) | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
```

Figura 2. Tabelas usadas no Banco de Dados

4.3. Views

```
mysql> describe vw_Turma_has_Alunos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idAluno    | int(11)   | NO   |     | 0       |       |
| nomeAluno  | varchar(45) | NO   |     | NULL    |       |
| idTurma    | int(11)   | NO   |     | 0       |       |
| nomeTurma  | varchar(45) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe vw_Turma;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idTurma    | int(11)   | NO   |     | 0       |       |
| nomeTurma  | varchar(45) | NO   |     | NULL    |       |
| turno     | varchar(45) | NO   |     | NULL    |       |
| idProfessor | int(11)   | NO   |     | 0       |       |
| nomeProfessor | varchar(45) | NO   |     | NULL    |       |
| idDisciplina | int(11)   | NO   |     | 0       |       |
| nomeDisciplina | varchar(45) | NO   |     | NULL    |       |
| YEAR(t.data) | int(4)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe curso_e_disc;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idCurso    | int(11)   | NO   |     | 0       |       |
| nomeCurso  | varchar(45) | NO   |     | NULL    |       |
| idDisciplina | int(11)   | NO   |     | 0       |       |
| nomeDisciplina | varchar(45) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

mysql> describe vw_frequencia;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idFrequencia | int(11)   | NO   |     | 0       |       |
| Turma_idTurma | int(11)   | NO   |     | NULL    |       |
| nomeTurma    | varchar(45) | NO   |     | NULL    |       |
| Aluno_idAluno | int(11)   | NO   |     | NULL    |       |
| nomeAluno    | varchar(45) | NO   |     | NULL    |       |
| Data        | date      | YES  |     | NULL    |       |
| Presenca    | varchar(2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

Figura 3. Views usadas no Banco de Dados

5. Funções Implementadas

Foi utilizada a estrutura MVC (Model-View/Controller), com a linguagem de programação Python3. Para criar a GUI utilizamos o Wrapper PyQt para Qt5, e para a interação com o Banco de Dados foi utilizada a biblioteca PyMySQL, por ser inteiramente escrito em Python, sem dependências, ao contrário dos conectores da Oracle, que são escritos em C.

5.1. Funções de Gestão do Banco de Dados

As funções em cada tabela são extremamente similares, divergindo apenas no campo de suas operações.

— `popularTabela` - Uma função presente em todas as tabelas do banco de dados, cujo nome descreve sua simples função: Preencher a tabela atual com os dados do banco.

— `*-item-Tabela-TABELA` - Base para a maior parte das funções de manipulação da tabela atual. `***` pode ser visto como - adicionar, remover, editar ou buscar.

— `check-if-exists-in-another-table` e `check-if-exists-in-db` - Últimas funções da tabela e essenciais para adição e remoção. A primeira busca se há informações da tabela atual em outra tabela do banco, extremamente importante para a remoção. Na segunda, como mencionado anteriormente, ela busca se a tupla atual já existe no banco, também extremamente importante, porém relacionada à operação de inserção

6. Dificuldades Encontradas

Finalizando, uma das maiores dificuldades encontradas na realização do trabalho foi a edição de tabelas com chaves estrangeiras. Simplesmente não conseguimos fazer a edição de relacionamentos de maneira satisfatória.

Referências

- [1] Viriato J.V., Vieira G.H., fbdqt, <https://github.com/jviriato/fbdqt> (2017).
- [2] Change Vision, Inc., Astah, <http://astah.net/> (2017).
- [3] Oracle Corporation, Mysql workbench, <https://www.mysql.com/products/workbench> (2017).
- [4] Python Software Foundation, Python 3, <https://www.python.org/> (2017).
- [5] Jet Brains, Pycharm, <https://www.jetbrains.com/products.html> (2017).
- [6] The QT Company, Qt designer, <https://www.qt.io/> (2017).
- [7] The LaTeX Project, Latex, <https://www.latex-project.org/get/> (2017).