

DOCUMENTO TÉCNICO – ESTRUTURA E OTIMIZAÇÃO DO BANCO DE DADOS

Projeto: SoulWare – Instituto Alma

Autor: Nelson dos Reis Gomes Souza

Entrega 1 – Projeto Interdisciplinar (2º Semestre ADS 2025)

1. Objetivo

Apresentar a estrutura inicial e as otimizações do banco de dados do projeto SoulWare, atendendo aos requisitos da 1ª entrega do Projeto Interdisciplinar. Este documento detalha a definição das tabelas, o uso de chaves primárias e estrangeiras para garantir a integridade dos dados e, principalmente, a criação de índices estratégicos para melhorar a performance de consultas, com a justificativa técnica para cada decisão.

2. Estrutura das Tabelas e Índices

A seguir, são apresentadas as definições das tabelas do banco de dados com foco nos índices criados para otimização.

2.1. Tabela **usuarios**

(Responsável pelo cadastro e autenticação de usuários no sistema)

MySQL

```
CREATE TABLE `usuarios` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `nome` varchar(100) NOT NULL,  
  `email` varchar(150) NOT NULL,  
  `telefone` varchar(20) DEFAULT NULL,  
  `permissoes` enum('admin','voluntario') NOT NULL DEFAULT 'voluntario',  
  `status` enum('pendente','aprovado') NOT NULL DEFAULT 'pendente',  
  `senha_hash` varchar(255) NOT NULL,  
  `criado_em` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `atualizado_em` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP  
ON UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `idx_usuarios_email` (`email`));
```

Justificativa do Índice (`idx_usuarios_email`): O índice único (**UNIQUE KEY**) na coluna `email` cumpre duas funções vitais:

1. **Garantia de Unicidade:** Impede que múltiplos usuários se cadastrem com o mesmo endereço de e-mail, mantendo a integridade da base de dados.
 2. **Otimização de Performance:** Acelera drasticamente a performance de consultas de autenticação (login) e recuperação de senha, que utilizam o e-mail como principal critério de busca, evitando varreduras completas na tabela (*full table scans*).
-

2.2. Tabela `eventos`

(Gerencia o controle de eventos promovidos pelo Instituto Alma)

MySQL

```
CREATE TABLE `eventos` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(150) NOT NULL,  
  `descricao` text,  
  `data_evento` date NOT NULL,  
  `horario` time DEFAULT NULL,  
  `local` varchar(200) DEFAULT NULL,  
  `status` enum('planejado','finalizado','cancelado') NOT NULL DEFAULT  
  'planejado',  
  `criado_em` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `atualizado_em` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON  
  UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  KEY `idx_eventos_titulo` (`titulo`)  
);
```

Justificativa do Índice (`idx_eventos_titulo`): A criação de um índice na coluna `titulo` visa otimizar as operações de busca e filtragem de eventos. Consultas que utilizam o título como critério são muito comuns, tanto em painéis administrativos quanto nas áreas públicas do site. Este índice garante que essas buscas sejam executadas de forma rápida e eficiente.

2.3. Tabela `inscricoes_eventos`

(Armazena o vínculo entre usuários e eventos)

MySQL

```
CREATE TABLE `inscricoes_eventos` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `usuario_id` int NOT NULL,  
  `evento_id` int NOT NULL,  
  `data_inscricao` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `idx_inscricoes_usuario_evento` (`usuario_id`, `evento_id`),  
  CONSTRAINT `fk_inscricoes_usuarios` FOREIGN KEY (`usuario_id`)  
    REFERENCES `usuarios` (`id`) ON DELETE CASCADE,  
  CONSTRAINT `fk_inscricoes_eventos` FOREIGN KEY (`evento_id`)  
    REFERENCES `eventos` (`id`) ON DELETE CASCADE  
);
```

Justificativa do Índice (`idx_inscricoes_usuario_evento`): O índice composto e único (**UNIQUE KEY**) sobre as colunas `usuario_id` e `evento_id` é uma peça fundamental na arquitetura do sistema por três motivos principais:

1. **Prevenção de Duplicidade:** Garante que um mesmo usuário não possa se inscrever mais de uma vez no mesmo evento, reforçando a regra de negócio diretamente no banco de dados.
2. **Performance em Consultas:** Acelera significativamente consultas que verificam a inscrição de um usuário em um evento específico ou que listam todos os eventos em que um determinado usuário está inscrito.

Integridade Referencial: As chaves estrangeiras (**FOREIGN KEY**) associadas garantem que apenas usuários e eventos existentes possam ser vinculados. O **ON DELETE CASCADE** assegura que, se um usuário ou um evento for removido, todas as inscrições associadas também sejam limpas automaticamente, evitando dados órfãos.

ANTES DO ÍNDICE

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other database functions. The status bar at the top right indicates a execution time of 0,024 segundos. The main window is titled 'Query Builder' and contains the following SQL query:

```
SELECT email FROM USUARIOS WHERE email = 'carla.mendes@example.com';
```

Below the query editor, the 'Plano de Explicação' (Execution Plan) tab is active. It shows the execution plan for the query, which is a full table scan of the 'USUARIOS' table. The plan is displayed in a tree view on the left and a table on the right.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1
TABLE ACCESS	USUARIOS	FULL		1

The tree view on the left shows the following structure:

- SELECT STATEMENT
 - TABLE ACCESS (USUARIOS)
 - Filter Predicates
 - EMAIL='carla.mendes@example.com'
- Other XML
 - {info}
 - info type="db_version"
 - 11.2.0.2
 - info type="parse_schema"
 - "ESTUDOS"

DEPOIS DO ÍNDICE

ESTUDOS x

0,005 segundos

Planilha Query Builder

```
SELECT email FROM USUARIOS WHERE email = 'carla.mendes@example.com';  
  
CREATE UNIQUE INDEX idx_usuarios_email ON usuarios (email);
```

Plano de Explicação x

SQL | 0,005 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1
INDEX	IDX_USUARIOS_EMAIL	UNIQUE SCAN		1

Access Predicates
EMAIL='carla.mendes@example.com'

Other XML
{info}
info type="db_version"
11.2.0.2
info type="parse_schema"

TABELA EVENTOS

ANTES DO ÍNDICE

ESTUDOS x EVENTOS x

0,022 segundos

Planilha Query Builder

```
SELECT email FROM USUARIOS WHERE email = 'carla.mendes@example.com';  
  
CREATE UNIQUE INDEX idx_usuarios_email ON usuarios (email);  
  
SELECT titulo FROM EVENTOS WHERE titulo = 'Feira de Voluntariado';
```

Resultado da Consulta x Saída do Script x Plano de Explicação x

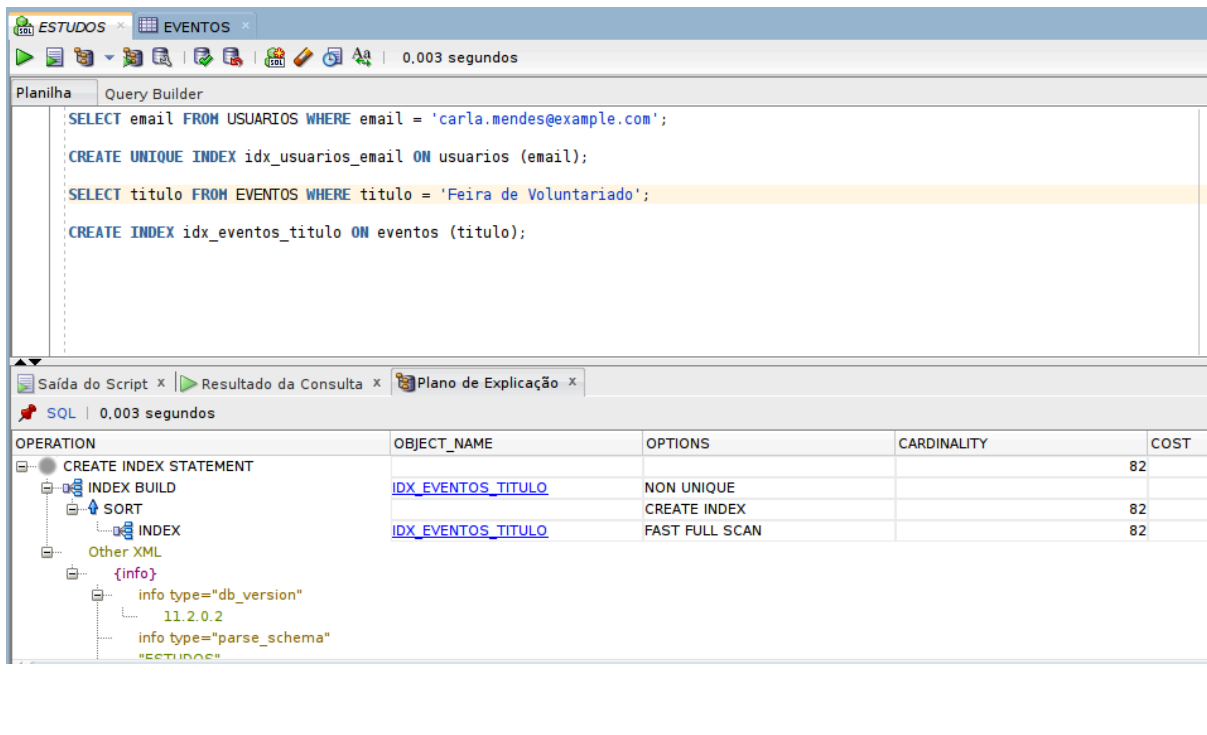
SQL | 0,022 segundos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				1
TABLE ACCESS	EVENTOS	FULL		1

Filter Predicates
TITULO='Feira de Voluntariado'

Other XML
{info}
info type="db_version"
11.2.0.2
info type="parse_schema"
"ESTUDOS"

DEPOIS DO ÍNDICE



The screenshot shows a database query editor with the following SQL commands:

```
SELECT email FROM USUARIOS WHERE email = 'carla.mendes@example.com';  
CREATE UNIQUE INDEX idx_usuarios_email ON usuarios (email);  
SELECT titulo FROM EVENTOS WHERE titulo = 'Feira de Voluntariado';  
CREATE INDEX idx_eventos_titulo ON eventos (titulo);
```

The execution plan for the second query is shown below:

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
CREATE INDEX STATEMENT				82
INDEX BUILD	IDX_EVENTOS_TITULO	NON UNIQUE		
SORT		CREATE INDEX		82
INDEX	IDX_EVENTOS_TITULO	FAST FULL SCAN		82

Additional information from the execution plan:

- Other XML: {info}
- info type="db_version": 11.2.0.2
- info type="parse_schema": "ESTUDOS"

3. Observações Finais

Os índices foram criados de forma estratégica, focando em colunas utilizadas em cláusulas **WHERE**, **JOIN** e **UNIQUE**, para garantir máxima performance em operações de leitura sem sobrecarregar as operações de escrita (**INSERT**, **UPDATE**, **DELETE**).

- As chaves estrangeiras foram implementadas para assegurar a integridade referencial entre as tabelas relacionadas, um pilar para a consistência dos dados.