

## Linguagem de definição de dados

**Linguagem de definição de dados (LDD ou DDL**, do Inglês *Data Definition Language*) é uma linguagem de computador usada para a definição de estruturas de dados. O termo foi inicialmente introduzido em relação ao modelo de banco de dados Codasyl, onde o esquema de banco de dados era escrito em uma Linguagem de Definição de Dados descrevendo os registros, campos e "conjuntos" que construíam o Modelo de dados do usuário. Inicialmente referia-se a um subconjunto da SQL, mas hoje é usada em um sentido genérico para referir-se a qualquer linguagem formal para descrição de estruturas de dados ou informação, assim como esquemas XML.

Uma vez compilados, os parâmetros DDL são armazenados num conjunto de arquivos denominado dicionário de dados (ou catálogo). O dicionário de dados contém os metadados (dados a respeito das estruturas de armazenamento). O SGBD sempre consulta os metadados a cada operação sobre o banco de dados. Por exemplo, um determinado programa precisa recuperar alguns campos (nome, CPF) de um arquivo de clientes. O SGBD irá verificar se os campos "nome" e "CPF" estão definidos para este arquivo. O interpretador DDL processa os comandos alimentados pelos DBAs na definição dos esquemas.

### Sql

Um subconjunto de instruções SQL formam outra '**DDL**'. Estas declarações SQL definem a estrutura de um banco de dados, incluindo linhas, colunas, tabelas, índices e características específicas do banco de dados, tal como localizações de arquivos. Declarações DDL SQL fazem mais parte do SGBD e possuem grandes diferenças entre as variações da SQL. Comandos DDL SQL incluem os seguintes:

#### Declarações Create

**Create** - utilizada para construir um novo banco de dados, tabela, índice ou consulta armazenada. Uma declaração CREATE, em SQL, cria um objeto dentro do Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR). Os tipos de objetos que podem ser criados dependem de qual SGBDR está sendo utilizado, porém a maioria suporta a criação de tabelas, índices, usuários e banco de dados. Alguns sistemas (tais como PostgreSQL) suportam o comando CREATE, e outros comandos DDL, dentro de uma transação e portanto suportam rollback.

- **COMMIT**: indica o término de uma transação bem-sucedida. Ela informa ao gerenciador de transações que uma unidade lógica de trabalho foi concluída com sucesso, o banco de dados já está novamente em um estado consistente e todas as atualizações feitas por essa unidade de trabalho já podem se tornar permanentes.
- **ROLLBACK**: assinala o término de uma transação malsucedida. Ela informa ao gerenciador de transações que algo saiu errado, que o banco de dados pode estar em um estado inconsistente, e que todas as atualizações feitas pela unidade lógica de trabalho até agora devem ser desfeitas.

## Create Table

Talvez o comando mais comum da declaração CREATE seja o comando CREATE TABLE. A sintaxe típica é:

```
CREATE TABLE [nome da tabela] ([definições de colunas]) [parâmetros da tabela]
```

**Definições de colunas:** Uma lista separada por vírgulas consistindo de qualquer uma das seguintes

- Definição de coluna: *[nome da coluna] [tipo de dado] {NULL|NOT NULL} {opções de coluna}*
- Definição de chave primária: *PRIMARY KEY([lista de colunas separadas por vírgulas])*
- CONSTRAINTS: *{CONSTRAINT} [definição de limite]*
- Funcionalidade específica de SGBDR

Por exemplo, o comando para criação de uma tabela chamada **empregados** com algumas colunas seria:

```
CREATE TABLE empregados (  
  id          INTEGER PRIMARY KEY,  
  nome        CHAR(50) NULL,  
  sobrenome    CHAR(75) NOT NULL,  
  data_de_aniversario DATE NULL  
);
```

## Declarações Drop

Drop - remove um banco de dados, tabela, índice ou visão existente.

Uma declaração **DROP** em SQL remove um objeto de um sistema de gerenciamento de banco de dados relacional(SGBDR). Os tipos de objetos que podem ser removidos dependem de qual SGBDR está sendo usado, mas a maioria suporta a exclusão de tabelas, usuários e banco de dados. Alguns sistemas (tais como o PostgreSQL) permitem que DROP e outros comandos ocorram dentro de uma transação e portanto suportem roll back.

Um uso típico é DROP *tipo\_do\_objeto nome\_do\_objeto*. Por exemplo, o comando para excluir uma tabela chamada **empregados** seria:

```
DROP TABLE empregados;
```

A declaração DROP é diferente das declarações DELETE e (não padronizada) TRUNCATE, em que estas não removem a tabela, de fato. Por exemplo, uma declaração DELETE poderia deletar alguns (ou todos) dados da tabela enquanto deixaria a tabela propriamente dita no banco de dados, enquanto que uma declaração DROP removeria a tabela inteira do banco de dados.

## Declarações ALTER

Alter - Modifica um objeto existente do banco de dados.

Uma declaração ALTER em SQL altera as propriedades de um objeto dentro de um sistema de gerenciamento de banco de dados relacional (SGBDR). Os tipos de objetos que podem ser alterados dependem de qual SGBDR está sendo usado.

Um uso comum é ALTER *tipo\_do\_objeto nome\_do\_objeto parametros*. Por exemplo, o comando para adicionar(e então remover) uma coluna chamada **bolhas** para uma tabela existente chamada **pia** seria:

```
ALTER TABLE pia ADD bolhas INTEGER;  
ALTER TABLE pia DROP COLUMN bolhas;
```

## Declarações de integridade referencial

Finalmente, outro tipo de sentença DDL em SQL são as declarações para definição de relacionamentos de integridade referencial, comumente implementadas como tags chaves primárias e chaves estrangeiras em algumas colunas das tabelas. Estas duas declarações podem ser incluídas dentro de uma sentença CREATE TABLE ou ALTER TABLE.