

LINGUAGEM SQL

INTRODUÇÃO

A linguagem SQL foi desenvolvida para trabalhar com a manipulação de dados em bancos de dados e tornou-se um padrão internacional de acesso a bancos de dados.

LINGUAGEM SQL

Seus comandos são divididos em categorias, cada qual com funcionalidades específicas. Dependendo a literatura utilizada, possuímos 4 ou 5 categorias. Considerando o material fornecido pela Oracle (empresa que é líder do mercado no segmento de banco de dados), possuímos 4 classes distintas de comandos, sendo:

LINGUAGEM SQL

Comandos DDL (Linguagem de Definição de Dados):

São comandos utilizados para construção dos modelos de dados. Por exemplo, deseja-se criar uma tabela (entidade), deve-se utilizar o comando `CREATE TABLE` para construí-la.

Deve ficar claro que os comandos DDL são utilizados somente para criação de estruturas e não inserção de registros de dados.

LINGUAGEM SQL - DDL

Entre os comandos DDL temos:

CREATE - Utilizado para criação de tabelas, índices, relacionamentos, entre outro. Como exemplo tem-se:

```
CREATE TABLE table_name  
  ( column1 datatype [ NULL | NOT NULL ] [DEFAULT valor],  
    column2 datatype [ NULL | NOT NULL ] [DEFAULT valor],  
    column_n datatype [ NULL | NOT NULL ] );
```

LINGUAGEM SQL - DDL

Principais tipos de dados (DataType)

Char (n) - Caracteres de comprimento fixo n. O valor por omissão é 1 sendo o valor máximo 2000

Varchar2(n) - Armazena uma cadeia de caracteres de comprimento variável. O valor máximo de n são 4000 bytes

Number(n,d) - Número com n dígitos sendo d casas decimais (à direita da vírgula);

Date - Também é armazenada a hora, minuto e segundo

Blob - Binary Large Object. Armazena até 4Gb em formato binário

Clob - Character Large Object. Armazena até 4Gb de texto

LINGUAGEM SQL - DDL

Exemplo:

```
CREATE TABLE tb_cliente (  
    cli_codigo NUMBER(6,0) NOT NULL,  
    cli_nome VARCHAR2(60) NOT NULL,  
    cli_rg VARCHAR2(15) NOT NULL,  
    cli_cpf VARCHAR2(15) NOT NULL,  
    cli_sexo CHAR(1) NOT NULL default 'M',  
    CONSTRAINT PK_cliente PRIMARY KEY (cli_codigo)  
);
```

Linguagem SQL - DDL

Este exemplo, apresenta a criação da entidade `tb_cliente`, com 5 atributos, onde são definidos os tipos de dados e a chave primária (`cli_codigo`) no final da instrução. O que indica o final da instrução corresponde ao (;) ponto e vírgula.

Comando SQL - Relacionamentos

```
CREATE TABLE employees (  
    employee_number number(10) NOT NULL,  
    employee_name varchar2(50) NOT NULL,  
    department_id number(10),  
    salary number(6),  
    CONSTRAINT PK_employees PRIMARY KEY  
        (employee_number),  
    CONSTRAINT fk_departments FOREIGN KEY  
        (department_id) REFERENCES departments  
        (department_id) );
```


Linguagem SQL - Index

Sintaxe de comando para criar um índice para uma tabela.

```
CREATE [UNIQUE] INDEX index_name ON tbl_name  
(index_col_name,...)
```

Exemplo:

```
Create index idx_cadastro on Tb_Cadastro(nome);
```

Linguagem SQL – Índice Único

Para criação do índice único (não permite cadastrar 2 cliente com o mesmo nome e com o mesmo CPF) um comando DDL deve ser utilizado, conforme o exemplo:

```
CREATE UNIQUE INDEX uk_cliente ON tb_cliente  
(cli_nome, cli_cpf);
```

Linguagem SQL - Alter

ALTER - é utilizado para alterar a estrutura de uma tabela por exemplo. Através dele pode-se:

Adicionar colunas:

```
ALTER TABLE TB_CLIENTE ADD CLI_FONE VARCHAR2(14)  
NOT NULL;
```

Linguagem SQL - Alter

Adicionar colunas:

```
ALTER TABLE table_name  
    ADD (column_1 column-definition,  
        column_2 column-definition, ...  
        column_n column_definition);
```

Linguagem Sql - Alter

Modificar colunas:

```
ALTER TABLE nome-tabela
```

```
    MODIFY nome-atributo tipo column_definition;
```

Obs: não é permitido diminuir um atributo que tenha dados armazenados.

Linguagem Sql - Alter

Exemplo:

```
ALTER TABLE TB_CLIENTE  
    MODIFY CLI_FONE VARCHAR2(18) NOT NULL;
```

Obs: não é permitido diminuir um atributo que tenha dados armazenados.

Linguagem SQL – Excluir colunas

Para excluir uma coluna deve-se utilizar o comando abaixo:

```
ALTER TABLE table_name DROP COLUMN  
column_name;
```

Exemplo:

```
ALTER TABLE TB_CLIENTE DROP COLUMN CLI_FONE;
```

Linguagem SQL - Renomear

Para alterar o nome de uma coluna deve-se utilizar o comando abaixo:

```
ALTER TABLE table_name RENAME COLUMN old_name to  
new_name;
```

```
ALTER TABLE TB_CLIENTE RENAME COLUMN  CLI_FONE  
TO CLI_TELEFONE;
```


Linguagem SQL - Renomear

Para alterar o nome de uma tabela deve-se utilizar o comando abaixo:

```
ALTER TABLE table_name RENAME TO new_table_name;
```

```
ALTER TABLE TB_CLIENTE RENAME TO TB_CLIENTES;
```

Linguagem SQL – Alter (primary key)

Pode-se usar o comando Alter Table para incluir uma constraint chave primaria após a tabela ser criada:

```
ALTER TABLE table_name ADD CONSTRAINT  
nome_constraint primary KEY(nome_atributo);
```

Exemplo:

```
alter table tb_cadastro add constraint pk_cadastro primary  
key(cod_cadastro);
```

Linguagem SQL – Alter

Pode-se usar o comando Alter Table para incluir uma constraint após a tabela ser criada:

```
ALTER TABLE table_name ADD CONSTRAINT  
nome_constraint FOREIGN KEY(nome_atributo) REFERENCES  
table_name(nome_atributo)
```

Exemplo:

```
alter table tb_nota add constraint fk_notaxcliente foreign  
key(cod_cadastro) references tb_cadastro(cod_cadastro);
```

Linguagem SQL – Excluir Constraint

Para excluir uma integridade referencial (constraint) seguir os passos abaixo:

Tabela do dicionário de dados onde ficam armazenadas as constraints: (user_constraints);

```
Alter table [nome_tabela] drop constraint [nome_constraint]
```

Exemplo:

```
Alter table tb_estado drop constraint fk_cadastro1;
```

Linguagem SQL – Excluir tabela

DROP - Comando utilizado para apagar tabelas, colunas, índices, entre outros:

Drop Table [nome_da_tabela];

Exemplo:

```
DROP TABLE tb_cliente;
```

Linguagem SQL – Criar Sequencia

Create sequence nome_sequencia
start with [numero inicial]
increment by [numero incremento];

Exemplo:

```
CREATE SEQUENCE sq_cadastro  
START WITH 1000  
INCREMENT BY 1;
```

Linguagem SQL – Excluir Sequencia

Drop sequence nome_sequencia;

Exemplo:

```
DROP SEQUENCE sq_cadastro;
```

Linguagem SQL

Algumas tabelas do dicionário do Oracle importante:

- User_Objects
- User_Tables
- User_Constraints
- User_Source

Comandos DML (Linguagem de Manipulação de Dados)

Os comandos de manipulação de dados server para incluir, alterar, deletar ou selecionar registro das estruturas de dados criadas utilizando os comandos DDL.

Comandos DML (Inserção de dados)

INSERT - Utilizado para inserir registros em uma entidade.

Exemplo:

```
INSERT INTO tb_bairro (bai_codigo, bai_descricao)
VALUES (1, 'CENTRO');
```

```
INSERT INTO tb_bairro(bai_codigo, bai_descricao)
VALUES (sq_bairro.nextval, 'CENTRO');
```

```
INSERT INTO tb_bairro VALUES (1, 'CENTRO');
```

Comandos DML (Alterar de dados)

UPDATE - Utilizado para atualizar registros em uma entidade.

Exemplo:

```
UPDATE tb_funcionário SET fun_salario = 2000  
WHERE fun_codigo = 10;
```

```
UPDATE tb_funcionário SET fun_salario = fun_salario * 1.1;
```

Comandos DML (excluir de dados)

DELETE - Utilizado para apagar registro de uma entidade.

Exemplo:

```
DELETE FROM tb_cliente WHERE cli_codigo = 5;
```

```
DELETE FROM tb_cliente;
```

Comandos DML (buscar dados)

SELECT - Utilizado para selecionar registros em uma ou mais tabelas.

Exemplo:

```
SELECT fun_codigo, fun_nome, fun_salario FROM  
tb_funcionario;
```

```
SELECT * FROM tb_cliente;
```

Comandos DCL (Linguagem de Controle de Dados)

Permite controlar quais usuários tem permissões para quais atribuições no banco de dados. O Comando GRANT dá permissão e o comando REVOKE exclui as permissões concedidas.

Comandos DCL (Linguagem de Controle de Dados)

```
GRANT CREATE_TABLE ON USER_JOSE;  
REVOKE CREATE_TABLE ON USER_JOSE;
```

Respectivamente, dá-se permissão de criação de tabelas ao usuário USER_JOSE e retira-se a permissão concedida

Comandos DTL (Linguagem de Transação de Dados):

Estes comandos são necessários para que os dados sejam gravados fisicamente no disco quando confirma-se uma transação como concluída.

Desta forma, o comando COMMIT, quando executado grava toda a transação no disco, sem a possibilidade de retorno em uma situação anterior a não ser através de mecanismos de backup.

Quando deseja-se desfazer uma transação, executa-se o comando ROLLBACK e o sistema desfaz automaticamente tudo o que não foi comitado até o momento.