BANCO DE DADOS

SQL

Prof. Marcos Alexandruk

SUMÁRIO

BREVE HISTÓRIA DA LINGUAGEM SQL

Durante o desenvolvimento do Sistema R, pesquisadores da IBM desenvolveram a linguagem SEQUEL (Structured English Query Language), primeira linguagem de acesso a SGBDR (Sistemas Gerenciadores de Banco de Dados Relacionais).

E. F. Codd elaborou a teoria de que, usando a linguagem SQL, seria possível navegar por grandes quantidades de informações e obter rapidamente uma resposta à consulta.

Com o surgimento de um número cada vez maior de SGBDRs, fez-se necessário especificar um padrão para a linguagem de acesso.

Portanto, em 1986, surgiu o SQL-86, a primeira versão da linguagem SQL (Structured Query Language), em um trabalho conjunto da ISO (International Organization for Standardization) e da ANSI (American National Standards Institute).

A linguagem passou por aperfeiçoamentos e em 1992 foi lançada a SQL-92 ou SQL-2.

Oito anos depois, foi lançado um novo padrão chamado SQL-99 ou SQL-3. Este padrão permitiu a utilização de tipos de dados complexos (exemplo: BLOB - Binary Large Object) e incorporou características de orientação a objetos.

A mais nova versão do padrão SQL é a SQL:2003. Fez-se uma revisão do padrão SQL3 e acrescentou-se uma nova parte que contempla o tratamento de XML.

PRINCIPAIS SISTEMAS DE BANCO DE DADOS QUE USAM SQL

- Apache Derby
- Caché
- DB2
- Ingres
- InterBase
- MySQL
- Oracle
- PostgreSQL
- Microsoft SQL Server
- SQLite
- Sybase
- Informix
- Firebird
- HSQLDB (banco de dados feito em Java)
- PointBase (banco de dados relacional feito em Java)

ORACLE

A palavra *oráculo* significa *profecia* ou *alguém que faz previsões*. Acreditava-se que, se alguém fizesse uma pergunta a um oráculo obteria a resposta de uma divindade.

HISTÓRICO

1977 Larry Ellison, Bob Miner e Ed Oates fundam a SDL (Software Level Laboratories).

1978 O nome da empresa é mudado para RSI (Rational Software Inc.). O Oracle 1.0 é escrito em assembly (utilizando no máximo 128 KB de memória).

1979 A RSI lança o primeiro produto comercial de banco de dados relacional utilizando a linguagem SQL.

1980 O nome da RSI é mudado para Oracle System Corporation (mais tarde: Oracle Corporation). E lançado o Oracle 2.0 em linguagem assembly. É decidido que a próxima versão do SGDB será reescrita em liguagem C.

1983 Lançado o Oracle 3, o primeiro SGBD a rodar em mainframes e em minicomputadores.

1984 Desenvolvida a versão para PC do Oracle (utilizando 256 KB de memória). Lançada a versão 4 do Oracle.

1986 O Oracle 5 é lançado com capacidades distribuídas (SQL*Star). É possível reunir informações de bancos de dados localizados em sites (locais) diferentes.

1988 Lançados pacotes financeiros e CASE (Computer Assisted Software Engeneering). A Oracle muda sua sede para Redwood City.

1989 Lançado o Oracle 6.2.

1993 Lançado o Oracle 7 para UNIX.

1994 Lançado o Oracle 7 para PC.

1996 O conjunto de aplicações comerciais da Oracle inclui os seguintes pacotes: Financials, Suply Change Management, Manufacturing, Project Systems, Human Resources, Market Management. A Oracle adquire a IRI Software e passa a oferecer um conjunto de ferramentas OLAP (Online Analytical Processing), tecnologia que dará suporte ao data warehouse da Oracle.

1997 Lançado o Oracle 8.

1998 A Oracle oferece suporte ao Linux.

1999 Lançado o Oracle 8i.

2000 Lançado o Oracle 9i.

WEBSITES RECOMENDADOS

http://www.oracle.com/technology/documentation/index.html

http://www.ss64.com/ora/

SQL (STRUTURED QUERY LANGUAGE)

Linguagem padrão para manipulação de dados em SGBRD (Sistemas Gerenciadores de Bancos de Dados Relacionais).

Os comandos SQL estão divididos em quatro categorias principais de acordo com sua funcionalidade:

DDL (DATA DEFINITION LANGUAGE)

Linguagem de Definição de Dados: usada para definir dados e objetos de um banco de dados.

COMANDO	FUNÇÃO
CREATE TABLE	Cria uma tabela
CREATE INDEX	Cria um índice
ALTER TABLE	Altera ou insere uma coluna de uma tabela
ALTER INDEX	Altera um índice
DROP TABLE	Elimina uma tabela
DROP INDEX	Elimina um índice

DML (DATA MANIPULATION LANGUAGE)

Linguagem de Manipulação de Dados: usada para manipular dados.

COMANDO	FUNÇÃO
INSERT	Insere uma linha na tabela
DELETE	Exclui as linhas da tabela
UPDATE	Altera o conteúdo de colunas (campos) da tabela

DQL (DATA QUERY LANGUAGE)

Linguagem de Consulta de Dados: usada para recuperar dados.

COMANDO	FUNÇÃO
SELECT	Seleciona (recupera) dados de uma tabela ou visão (view)

Obs.: O SELECT também é considerado um comando DML.

DCL (DATA CONTROL LANGUAGE)

Linguagem de Controle de Dados: usada para conceder ou remover direitos de acesso aos usuários do banco de dados.

COMANDO	FUNÇÃO
CREATE USER	Cria um usuário
ALTER USER	Altera um usuário
GRANT	Concede privilégios de acesso para um usuário
REVOKE	Revoga privilégios de acesso para um usuário

ALGUNS COMANDOS ÚTEIS

Alterar senha do usuário:

```
PASSWORD nome_usuario;
```

Alterar senha do usuário atual:

```
PASSWORD
```

Listar todos os usuários:

```
SELECT * FROM ALL_USERS;
```

Listar todas as tabelas:

```
SELECT TABLE_NAME FROM USER_TABLES;
```

Mostrar usuário atual:

```
SHOW USER
```

Informar constraints (restrições):

```
SELECT * FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'nome_da_tabela';
```

As informações são:

- **OWNER** Usuário que criou a restrição
- **CONSTRAINT_NAME** Nome da restrição
- **CONSTRAINT_TYPE** Tipo de restrição
  - P – PRIMARY KEY
  - R – FOREIGN KEY
  - C – CHECK
  - U – UNIQUE
- **TABLE NAME** Nome da tabela
- **SEARCH_CONDITION** Domínio permitido (check)
- **R_OWNER** Utilizada em restrições FK para indicar qual usuário criou a restrição PK na tabela referenciada
- **R_CONSTRAINT_NAME** Utilizada em restrições FK para indicar qual a restrição PK na tabela referenciada
- **STATUS** Indica se a restrição está ou não habilitada

NOMENCLATURA DE COLUNAS

- Não utilizar caracteres especiais (exceto o underscore "_");
- Começar com uma letra e não com um número;
- Evitar acentuação e "ç";
- Não utilizar espaços.

TIPOS DE DADOS

Tipo	Descrição
char	Cadeia de caracteres de tamanho fixo n. O default é 1 e o máximo é 255.
varchar2 (n)	Cadeia de caracteres de tamanho variável. Tamanho máximo 4.000.
long	Cadeia de caracteres de tamanho variável. Tamanho máximo 2 GB. Só pode existir um campo deste tipo por tabela.
raw e long raw	Equivalente ao varchar2 e long, respectivamente. Utilizado para armazenar dados binários (sons, imagens, etc.) Limite: 2.000 bytes.
number (p,e)	Valores numéricos. Ex: number (5,2) armazena -999,99 a +999,99.
date	Armazena data e hora (inclusive minuto e segundo). Ocupam 7 bytes.

CRIAÇÃO DE TABELAS

Para criar uma tabela utilizamos o comando CREATE TABLE:

```
CREATE TABLE tabela1 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30));
```

VERIFICANDO A ESTRUTURA DE UMA TABELA

```
DESCRIBE tabela1;

DESC tabela1;
```

CONSTRAINTS

As restrições (constraints) estabelecem as relações entre as várias tabelas em um banco de dados e realizam três tarefas fundamentais:

- Mantêm a integridade dos dados
- Não permitem a inclusão de valores "indesejáveis"
- Impedem a exclusão de dados se existirem dependências entre tabelas

CHAVE PRIMÁRIA (PK – PRIMARY KEY)

Coluna ou grupo de colunas que permite identificar uma única linha da tabela.

```
CREATE TABLE tabela1 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30),
CONSTRAINT tabela1_pk PRIMARY KEY(coluna1));

CREATE TABLE tabela1 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30),
CONSTRAINT tabela1_pk PRIMARY KEY(coluna1,coluna2));
```

CHAVE ESTRANGEIRA (FK – FOREIGN KEY)

Coluna que estabelece o relacionamento entre duas tabelas. Corresponde à chave primária da tabela-pai.

ON DELETE SET NULL	Quando for removido um valor da tabela-pai o Oracle define os valores correspondentes da tabela-filho como NULL.
ON DELETE CASCADE	Quando for removido um valor da tabela-pai o Oracle remove os valores correspondentes da tabela-filho.

```
CREATE TABLE tabela2 (
coluna1 NUMBER(5),
coluna2 NUMBER(5),
CONSTRAINT tabela2_pk PRIMARY KEY(coluna1),
CONSTRAINT tabela2_fk FOREIGN KEY(coluna2) REFERENCES tabela1(coluna1));

CREATE TABLE tabela2 (
coluna1 NUMBER(5),
coluna2 NUMBER(5),
CONSTRAINT tabela2_pk PRIMARY KEY(coluna1),
CONSTRAINT tabela2_fk FOREIGN KEY(coluna2) REFERENCES tabela1(coluna1)
ON DELETE SET NULL);
```

```
CREATE TABLE tabela2 (
coluna1 NUMBER(5),
coluna2 NUMBER(5),
CONSTRAINT tabela2_pk PRIMARY KEY(coluna1),
CONSTRAINT tabela2_fk FOREIGN KEY(coluna2) REFERENCES tabela1(coluna1)
ON DELETE CASCADE);
```

DEFAULT

Atribui um conteúdo-padrão a uma coluna da tabela.

```
CREATE TABLE tabela3 (
coluna1 NUMBER(5),
coluna2 NUMBER(5) DEFAULT 1,
coluna3 VARCHAR2(10) DEFAULT 'a',
coluna4 DATE DEFAULT SYSDATE);
```

NOT NULL

Indica que o conteúdo de uma coluna não poderá ser nulo (vazio).

```
CREATE TABLE tabela4 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30) NOT NULL);

CREATE TABLE tabela4 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30) CONSTRAINT tabela4_nn NOT NULL);
```

UNIQUE

Indica que não poderá haver repetição no conteúdo da coluna.

CHAVE PRIMÁRIA	
Permite repetição?	NÃO
Permite valores nulos?	NÃO

UNIQUE	
Permite repetição?	NÃO
Permite valores nulos?	SIM

```
CREATE TABLE tabela5 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30) UNIQUE);

CREATE TABLE tabela5 (
coluna1 NUMBER(5),
coluna2 VARCHAR2(30) CONSTRAINT tabela5_un UNIQUE);
```

CHECK

Define um domínio de valores para o conteúdo da coluna.

```
CREATE TABLE tabela6 (
coluna1 NUMBER(5),
coluna2 CHAR(1) CONSTRAINT tabela6_ch CHECK (coluna2='M' OR coluna2='F'));

CREATE TABLE tabela6 (
coluna1 NUMBER(5),
coluna2 CHAR(2) CONSTRAINT tabela6_ch CHECK (coluna2 IN ('SP','RJ','MG')));

CREATE TABLE tabela6 (
coluna1 NUMBER(5),
coluna2 NUMBER(5) CONSTRAINT tabela6_ch CHECK (coluna2 < 1000));
```

DESATIVANDO CONSTRAINTS

```
ALTER TABLE tabela1 DISABLE CONSTRAINT tabela1_pk;
```

ATIVANDO CONSTRAINTS

```
ALTER TABLE tabela1 ENABLE CONSTRAINT tabela1_pk;
```

Desativar uma CONSTRAINT, inserir valores que violem a restrição de integridade e tentar ativar a restrição posteriormente com o comando ENABLE provocará um erro.

Até a versão 9i não havia possibilidade de incluir a cláusula ON UPDATE CASCADE em uma restrição do tipo Foreign Key. Portanto, para alterar os valores em colunas Primary Key que contém referencias em tabelas-filho é preciso desativar a restrição FK na tabela-filho, alterar os valores na coluna PK na tabela-pai, alterar (se necessário) os valores na coluna FK na tabela-filho e ativar novamente a restrição Foreign Key.

REMOVENDO CONSTRAINTS

```
ALTER TABLE tabela1 DROP CONSTRAINT tabela1_pk;
```

ADICIONANDO CONSTRAINTS

```
ALTER TABLE tabela1 ADD CONSTRAINT tabela1_pk PRIMARY KEY(coluna1);
```

EXCLUSÃO DE TABELAS

Para excluir uma tabela utilizamos o comando DROP TABLE:

```
DROP TABLE tabela1;
```

```
DROP TABLE tabela1 CASCADE CONSTRAINTS;
```

ALTERAÇÃO DE TABELAS

Para alterar as definições de uma tabela utilizamos o comando ALTER TABLE.

ADICIONANDO COLUNAS

```
ALTER TABLE tabela1
   ADD coluna3 VARCHAR2(50);
```

MODIFICANDO COLUNAS

Alterando a **largura** de uma coluna:

```
ALTER TABLE tabela1
   MODIFY coluna2 VARCHAR2(40);
```

Alterando o **nome** de uma coluna (Oracle 9.2.0 e posteriores):

```
ALTER TABLE tabela1
   RENAME COLUMN coluna2 TO coluna4;
```

EXCLUINDO COLUNAS

```
ALTER TABLE tabela1
   DROP coluna3;
```

```
ALTER TABLE tabela1
   DROP coluna3 CASCADE CONSTRAINTS;
```

ALTERANDO O NOME DE UMA TABELA

```
RENAME tabela1 TO tabela10;
```

CRIANDO UMA TABELA COM BASE EM OUTRA

Criando uma tabela com todas as linhas e colunas de outra:

```
CREATE TABLE tabela7 AS SELECT * FROM tabela10;
```

Criando uma tabela com todas as linhas e algumas colunas selecionadas de outra:

```
CREATE TABLE tabela7 (coluna1,coluna2) AS SELECT coluna1,coluna2 FROM tabela10;
```

TRUNCATE

Comando DDL utilizado para 'cortar' uma tabela.

```
TRUNCATE TABLE nome_tabela;
```

- Mais rápido do que executar um DELETE sem a cláusula WHERE.
- Os dados não poderão ser recuperados (a não ser através do backup).

ÍNDICES

Fornecem acesso mais rápido a linhas específicas, eliminando a necessidade de varreduras completas de tabelas.

CRIANDO UM ÍNDICE

Para criar um índice utilizamos o comando CREATE INDEX:

```
CREATE INDEX tabela1_in ON tabela1 (coluna2);
```

EXCLUINDO UM ÍNDICE

Para excluir um índice utilizamos o comando DROP INDEX:

```
DROP INDEX tabela1_in;
```

ROWID

Pseudo coluna (coluna virtual), informa como o Oracle pode localizar em disco as linhas das tabelas (por arquivo, bloco dentro daquele arquivo e linha dentro daquele bloco).

```
SELECT ROWID FROM nome_tabela;
```

DER - DIAGRAMA ENTIDADE RELACIONAMENTO