

SQL (Structured Query Language)

Disciplina: Banco de Dados

Professor: José Antônio

Tópicos de discussão

- ◉ Criando um banco de dados
- ◉ Incluindo, atualizando e excluindo linhas nas tabelas
- ◉ Pesquisa básica em tabelas
- ◉ Cálculos e funções usuais
- ◉ Pesquisa em múltiplas tabelas
- ◉ Subconsultas
- ◉ Pesquisa avançada

Criando um BD

- ◉ Definição de dados
- ◉ Criação de tabelas
- ◉ Integridade referencial – Constraints
- ◉ Alteração de estrutura de tabela
- ◉ Excluindo uma tabela
- ◉ Criação de índices
- ◉ Laboratório

Linguagem DDL

- ◉ Incluindo dados em tabelas
- ◉ Atualizando dados em tabelas
- ◉ Excluindo dados de tabelas
- ◉ Controle básico de transações
- ◉ laboratório

Pesquisa Básica

- ◉ Ordenando o resultado
- ◉ Filtrando linhas
- ◉ laboratório

Cálculo e Funções

- Cálculos
- Numéricos
- Alfanuméricos
- Manipulação de datas
- Funções de grupo
- Conversão de tipo
- Agrupando resultado
- laboratório

Subconsultas

- Subquery de uma linha
- Utilizando subquery em cláusula HAVING
- EXISTS
- Subquery de múltiplas linhas
- Subquery na cláusula FROM
- laboratório

Pesquisa avançada

- UNION
- UNION ALL
- EXCEPT/DISTINCT/MINUS
- INSERTED
- Expressões CASE
- CASE compacto
- Expressões COALESCE
- Laboratório

Introdução

- Em junho de 1970, E. F. Codd, membro do Laboratório de Pesquisa da IBM em San Jose, na Califórnia, publicou um trabalho intitulado “ A Relacional Model of Data for Large Shared Data banks” (Um Modelo Relacional de dados para Grandes bancos de Dados Compartilhados), no jornal Association of Computer Machinery. Nesse trabalho, Codd estabeleceu princípios sobre gerência de banco de dados, denominando-os com o termo relacional. Essa foi a base utilizada na criação de uma linguagem-padrão para manipular informações em Banco de dados Relacionais. E essa linguagem é a SQL.

Introdução – continuação

- Desde o início da utilização dos computadores, sabemos que um sistema é feito para aceitar entrada de dados, realizar processamento e gerar saída das informações processadas. Com o tempo, verificou-se a necessidade de armazenar as informações geradas pelos programas de computadores. O armazenamento e a recuperação das informações passaram a desempenhar um papel fundamental na informática.

Histórico

- Inicialmente chamada de SEQUEL (Structured English Language), a linguagem SQL foi concebida e desenvolvida pela IBM, utilizando os conceitos de Codd (1974). Em 1977 passou a ser chamada de SQL. Em 1979, a Relational Software Inc., hoje Oracle Corporation, lançou a primeira versão comercial da linguagem SQL.

Histórico – continuação

- A SQL pode ser considerada um padrão para manipulação de dados em banco de dados. Duas entidades:
 - ANSI (American National Standards Institute)
 - ISO (International Standards Organization)
 - SQL – 86 (ANSI)
 - SQL – 89 – modificações significativas (BD atuais)
 - SQL – 92
 - SQL – 99 – SQL3 (BD objeto-relacional)

O que é SQL?

SQL é um conjunto de comandos de manipulação de banco de dados utilizados para criar e manter a estrutura desse banco de dados, além de incluir, excluir, modificar e pesquisar informações nas tabelas dele. Ela não é uma linguagem de programação autônoma. Quando desenvolvemos aplicações para banco de dados, é necessário utilizar uma linguagem de programação tradicional (C, Java, Pascal, COBOL, Visual Basic, Delphi, etc.) e embutir comandos SQL para manipular os dados.

SQL

A linguagem SQL é dividida nos seguintes componentes:

- Data Definition Language(DDL): permite a criação, alteração e exclusão de componentes (objetos) do banco de dados.
 - CREATE – ALTER – DROP
- Data Manipulation Language(DML): permite a manipulação dos dados armazenados no banco de dados.
 - INSERT – DELETE – UPDATE

SQL

A linguagem SQL é dividida nos seguintes componentes:

- Data Query Language(DQL): permite extrair dados do banco de dados.
 - SELECT
- Data Control Language(DCL): provê a segurança interna do banco de dados.
 - CREATE USER – ALTER USER
 - GRANT – REVOKE
 - CREATE SCHEMA

Criando um BD

Para iniciar o trabalho de criação do banco de dados, deve-se inicialmente transformar o modelo lógico (DER), no modelo físico que será implementado. O processo consiste em simplesmente atribuir tipos de dados e tamanho para cada um dos atributos que foram identificados.

Definição de Dados

Antes de criar as tabelas no nosso banco de dados, temos que definir quais são as características de cada um dos campos. As características que o SQL exige são o tipo e o tamanho de cada campo.

Tipo de dados

Tipo de dado	Descrição
INTEGER OU INT	Número positivo ou negativo inteiro.
SMALLINT	Mesma função do INT, mas ocupa a metade do espaço.
NUMERIC	Número positivo ou negativo. Deve-se informar o tamanho do campo e casas decimais.
DECIMAL	Semelhante a NUMERIC, em alguns casos tem maior precisão em casas decimais.
REAL	Número de ponto flutuante de simples precisão(Exponencial).
DOUBLE PRECISION	Número de ponto flutuante de dupla precisão.
FLOAT	Número de ponto flutuante em que você define o nível de precisão (número de dígitos significativos).
BIT	Armazenamento de um número fixo de bits.

Tipo de dados

Tipo de dado	Descrição
BIT VARYING	Igual a BIT, permitindo armazenar valores maiores. Normalmente utilizado para armazenar imagens.
DATE	Permite armazenar datas.
TIME	Permite armazenar horários.
TIMESTAMP	Permite armazenar uma combinação de data e hora.
CHAR	Permite armazenar cadeia de caracteres. Tamanho informado será fixo.
VARCHAR	Permite armazenar cadeia de caracteres, mas de tamanho variável.
INTERNAL	Intervalo de data ou hora.

Criando Tabelas

Tabelas são as estruturas mais importantes de um banco de dados. Nas tabelas estará o conteúdo que representa cada objeto do mundo real. No padrão SQL as tabelas em três categorias:

- Tabelas permanentes;
- Tabelas temporárias globais;
- Tabelas temporárias locais.

Criando Tabelas

CREATE TABLE – Sintaxe básica

```
CREATE TABLE nome_tabela (  
    coluna1 tipo_De_Dado constraint,  
    coluna2 tipo_De_Dado constraint,  
    ...  
    colunan tipo_De_dado constraint,  
    constraint_de_tabela )
```

Criando Tabelas

Exemplo:

```
Create table Cliente (  
    Clienteid int not null,  
    Nome varchar(50) not null,  
    Endereco varchar(50) not null,  
    Cidade varchar(25) not null,  
    Estado char(2) not null,  
    primary key (Clienteid) )
```

Constraint mais comuns

- Chave primaria (Primary key)
- Chave estrangeira (Foreign Key)
- Default
- Not null
- Unique
- Check

Constraint mais comuns

- Chave primaria (Primary key)

Digamos que haja uma tabela de cliente cuja chave primária seja o campo Clienteid. A criação da chave primária ficaria assim:

...

Primary key (Clienteid)

...

Constraint mais comuns

- Chave estrangeira (Foreign key)

```
FOREIGN KEY nome_Chave (lista de colunas)
REFERENCES nome_tabela (lista de colunas)
ON UPDATE ação
ON DELETE ação
```

ONDE

Ação – CASCADE – NO ACTION

Constraint mais comuns

- Chave estrangeira (Foreign key)

Como exemplo, vamos fazer referência á tabela de Clientes quando estamos criando a tabela de pedidos.

```
FOREIGN KEY pedido_cliente_fk ( clienteid )  
REFERENCES Cliente ( clienteid )  
ON UPDATE CASCADE  
ON DELETE CASCADE
```

Constraint mais comuns

- DEFAULT

Serve para atribuir um conteúdo-padrão a uma coluna da tabela, sempre que for incluída uma nova linha na tabela.

...

Quantidade INT default 1,

...

Constraint mais comuns

- NOT NULL

Indica que o conteúdo de uma coluna não pode ser Nulo.

...

Nome_Cliente varchar(50) not null,

...

Constraint mais comuns

- UNIQUE

Indica que não pode haver repetição no conteúdo da coluna. Isso é diferente do conceito de chave primária.

...

CPF numeric(11) UNIQUE,

...

Constraint mais comuns

- CHECK

Um domínio é uma expressão de valores possíveis para o conteúdo de uma coluna. Podemos, ao criarmos uma coluna, especificar quais os valores que poderão ser utilizados para preencher a coluna.

...

```
Sexo char(1) CHECK ( UPPER(sexo) = 'M' OR  
UPPER(Sexo) = 'F' ),
```

...

Constraint mais comuns

- ASSERTIVAS

Uma assertiva é utilizada para estabelecer restrição no banco de dados com base em dados de uma ou mais tabelas. Por exemplo, você pode estabelecer que a tabela CD sempre tenha pelo menos uma linha.

```
CREATE ASSERTION há_CD  
check (EXISTS select codigo_cd from CD )
```

Alterar estrutura de tabela

Para alterar a estrutura de uma tabela, utilizamos o comando ALTER TABLE.

- Adicionar novas colunas;
- Acrescentar novas constraints;
- Modificar colunas;
- Excluir elementos;
- Trocar o nome do elemento;

Alterar estrutura de tabela

Adicionar novas colunas:

```
ALTER TABLE nome_tabela
```

```
  ADD nome_coluna tipo_de_dado constraint,
```

```
    nome_coluna tipo_de_dado constraint,
```

```
    ...,
```

```
    nome_coluna tipo_de_dado constraint,
```

Alterar estrutura de tabela

Adicionar novas colunas:

Exemplo:

```
ALTER TABLE Cliente  
ADD email varchar(80) UNIQUE
```

Alterar estrutura de tabela

Adicionar novas constraints:

```
ALTER TABLE nome_tabela  
ADD constraint
```

Exemplo:

```
ALTER TABLE Cliente  
ADD primary key ( clienteid )
```

Alterar estrutura de tabela

modificando colunas:

```
ALTER TABLE nome_tabela  
    MODIFY nome_coluna tipo_de_dados constraint
```

Exemplo:

```
ALTER TABLE Cliente  
    MODIFY email varchar(100) not null
```

Alterar estrutura de tabela

Excluindo elementos:

```
ALTER TABLE tabela  
DELETE/DROP elemento
```

Alterar estrutura de tabela

Excluindo elementos:

Exemplos(SQL Server):

```
ALTER TABLE Cliente  
DROP Column email
```

```
ALTER TABLE Cliente  
DROP Constraint pk_cliente
```

Laboratório

Usando o Query Analyzer do SQL Server, crie o banco de dados Cadastro_CD, com a seguintes estrutura de tabelas:

Gravadora

Codigo_gravadora int
Nome_gravadora varchar(60)
Endereco varchar(60)
Telefone varchar(10)
Contato varchar(20)
URL varchar(80)

CD

codigo_cd int
codigo_grvadora int
nome varchar(60)
preco_venda decimal(14,2)
data_lancamento datetime
cd_indicado int

Faixa

codigo_cd int
codigo_musica int
numero_faixa int

Laboratório

Usando o Query Analyzer do SQL Server, crie o banco de dados Cadastro_CD, com a seguintes estrutura de tabelas e crie o diagrama do slide seguinte com o Enterprise Manager:

CD_Categoria

Codigo_categoria int
Menor_preco decimal(14,2)
Maior_preco decimal(14,2)
cd_indicado int

Autor

Codigo_autor int
Nome_autor varchar(60)

Musica

codigo_musica int
nome_musica varchar(60)
duracao decimal(6,2)

Musica_autor

codigo_musica int
codigo_autor int

Laboratório

