

– SQL –
Linguagem de Definição de
Dados

Laboratório de Bases de Dados
Profa. Dra. Cristina Dutra de Aguiar Ciferri

SQL

Structured Query Language

- Uma das mais importantes linguagens relacionais (se não a mais importante)
- Exemplos de SGBD que utilizam SQL
 - Oracle
 - Informix
 - Ingress
 - SQL Server
 - Interbase
 - SyBase
 - DB2
 - MySQL
 - PostgreSQL

Oracle e SQL DDL

- **CREATE TABLE**
 - cria uma nova tabela (relação) no BD
 - a nova tabela não possui dados
- **DROP TABLE**
 - remove uma tabela (relação) e todas as suas instâncias do BD
- **ALTER TABLE**
 - altera a estrutura de uma tabela (relação) já existente no BD

CREATE TABLE

```
CREATE TABLE nome_da_tabela  
(nome_da_coluna tipo_de_dado [(tamanho)] [DEFAULT expr] [restrições] [,  
nome_da_coluna tipo_de_dado [(tamanho)] [DEFAULT expr] [restrições] ]  
[, restrições] ) ;
```

- Cria uma nova tabela (relação)
- Cria os atributos da nova tabela, com
 - nome do atributo (i.e., nome_da_coluna)
 - tipo de dado e tamanho
 - valor DEFAULT
 - restrições que atuam no atributo

Tipos de Dados

- Char
 - alfanumérico de tamanho fixo
 - tamanho máximo de 2.000 caracteres
- Varchar2
 - alfanumérico de tamanho variável
 - tamanho máximo de 4.000 caracteres
- CLOB (character long object)
 - alfanumérico de tamanho variável
 - tamanho máximo de 4 GB

Tipos de Dados

- BLOB (binary long object)
 - binário de tamanho variável
 - tamanho máximo de 4 GB
- BFILE (binary file)
 - armazena uma referência a um arquivo externo ao banco de dados
 - não permite referência remota
 - o arquivo deve estar localizado na mesma máquina que o banco de dados
 - tamanho gerenciado pelo sistema operacional

Tipos de Dados

- Number (numérico, máx. 38 carac)
 - Number(4)
 - número inteiro
 - tamanho máximo de 4 números
 - Number(12,2)
 - número de ponto flutuante
 - tamanho máximo de 12 números
 - 10 inteiros e 2 decimais

Tipos de Dados

- DATE
 - século, ano, mês, dia, hora, minuto, segundo
- TIMESTAMP
 - dados de DATE + milésimo de segundo
- *Demais tipos de dados*
 - *mantidos para compatibilidade com versões anteriores*
 - *não são indicados para serem utilizados*

Cláusula DEFAULT

- Funcionalidade
 - associa um valor *default* para um atributo, caso nenhum outro valor seja especificado
- `expr`
 - valor fixo
 - expressão
 - função
 - variável do sistema (e.g., `SYSDATE`)

Restrições de Integridade

- Valor nulo
 - representado por NULL
 - membro de todos os domínios
- Restrição NOT NULL
 - especificada quando NULL não é permitido
 - proíbe que o atributo receba valor nulo
- Comparações
 - usar IS NULL e IS NOT NULL

Restrições de Integridade

- Cláusula PRIMARY KEY
 - identifica os atributos da relação que formam a sua chave primária
 - os atributos devem ser definidos como NOT NULL
 - sintaxe
PRIMARY KEY (atributo₁, atributo₂, ..., atributo_x)
- Cláusula UNIQUE
 - não permite valores duplicados para um determinado atributo

Restrições de Integridade

- Cláusula CHECK
 - especifica um predicado que precisa ser satisfeito por todas as tuplas de uma relação
 - exemplos
 - saldo NUMBER (6,2), ...
 - CHECK (saldo >= 0.0),

 - nível char(15), ...
 - CHECK (nível IN ('Bacharelado', 'Mestrado', 'Doutorado'))

Restrições de Integridade

- Cláusula FOREIGN KEY
 - reflete nas chaves estrangeiras todas as alterações na chave primária
 - elimina a possibilidade de violação da integridade referencial
 - sintaxe

```
FOREIGN KEY (atributos)
REFERENCES nome_relação (atributos)
[ON DELETE [ NO ACTION | CASCADE | SET NULL |
SET DEFAULT]]
```

Nomeando Restrições

- Toda restrição possui um nome
 - definido pelo usuário: opção CONSTRAINT
 - atribuído automaticamente pelo sistema
- Definição de restrições
 - chaves primárias (i.e., PRIMARY KEY)
 - chaves estrangeiras (i.e., FOREIGN KEY)
 - chaves únicas (i.e., UNIQUE)
 - condições para valores permitidos para determinado campo (i.e., CHECK)

Declarando Restrições

- Exemplo

```
CREATE TABLE cliente (
```

declaração dos campos

```
cod_cliente      NUMBER(4),  
CPF_cliente      CHAR(12),  
... ,
```

declaração das restrições

```
CONSTRAINT PK_CLIENTE PRIMARY KEY (cod_cliente),  
CONSTRAINT UNIQUE_CPF UNIQUE (CPF_cliente),  
... )
```

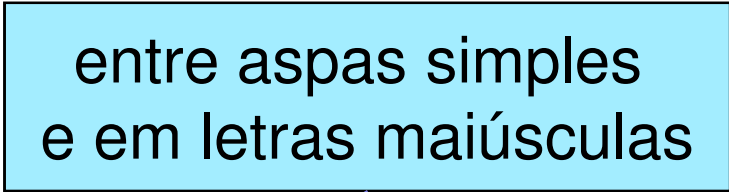
Consultando Restrições

- Tabela **USER_CONSTRAINTS**
 - tabela de controle que armazena dados sobre as restrições

- Consulta

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

entre aspas simples
e em letras maiúsculas



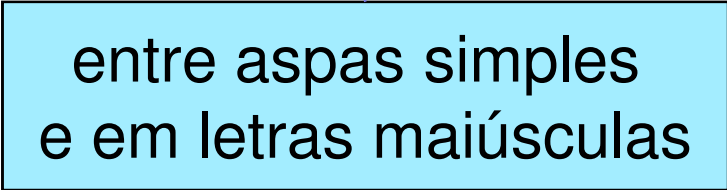
```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME = 'CLIENTE'
```


Consultando Restrições

- Tabela **USER_CONS_COLUMNS**
 - tabela de controle que contém dados sobre as colunas envolvidas em cada restrição

- Consulta

```
SELECT *  
FROM USER_CONS_COLUMNS  
WHERE CONSTRAINT_NAME = 'NOME_DA_RESTRIÇÃO';
```



entre aspas simples
e em letras maiúsculas

DESCRIBE

```
DESC[RIBE] nome_da_tabela;
```

- Mostra a estrutura de uma tabela

DROP TABLE

```
DROP TABLE nome_da_tabela;
```

- Remove uma tabela existente do BD
- Verifica dependências
 - *foreign keys* de outras tabelas associadas devem ser eliminadas anteriormente

ALTER TABLE

- Altera o esquema de uma tabela do BD
 - adiciona (ADD)
 - remove (DROP)
 - modifica (MODIFY)
 - habilita (ENABLE) ou desabilita (DISABLE)
- atributos ou restrições de integridade

ALTER TABLE

ALTER TABLE nome_da_tabela

ADD (nome_da_coluna tipo_de_dado [(tamanho)] [restrições] [,
nome_da_coluna tipo_de_dado [(tamanho)] [restrições]]
[, restrições])

MODIFY (nome_da_coluna [tipo_de_dado [(tamanho)]] [,
nome_da_coluna [tipo_de_dado [(tamanho)]]])

DROP COLUMN nome_da_coluna

DROP CONSTRAINT nome_da_restrição

ENABLE/DISABLE CONSTRAINT nome_da_restrição

Oracle e SQL DDL

- **CREATE INDEX**
 - cria um índice sobre uma ou mais colunas de uma tabela
- **DROP INDEX**
 - remove um índice existente do BD
- **ALTER INDEX**
 - torna um índice ativo ou inativo

CREATE INDEX

```
CREATE [BITMAP] [UNIQUE] INDEX nome_do_índice  
ON nome_da_tabela  
(nome_da_coluna ASC/DESC  
[, nome_da_coluna ASC/DESC] ...)
```

- Cria um índice sobre uma ou mais colunas de uma tabela
- Considerações
 - desempenho das consultas *versus* custos de atualização e de armazenamento

ALTER INDEX

```
ALTER INDEX nome_do_índice REBUILD ;
```

- Recria um índice
 - visa a melhoria de desempenho
 - deve ser realizado periodicamente (dica)
- Não altera um índice
 - para modificá-lo é preciso excluí-lo e depois criá-lo novamente

DROP INDEX

```
DROP INDEX nome_do_índice ;
```

- Remove um índice existente do BD

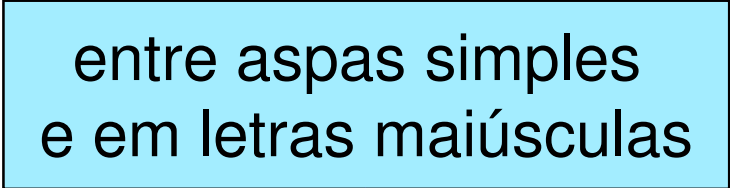
Consultando Índices

- Tabela **USER_INDEXES**
 - tabela de controle que armazena dados sobre os índices
- Tabela **USER_IND_COLUMNS**
 - tabela de controle que armazena dados das colunas dos índices

- Exemplo

```
SELECT *  
FROM USER_INDEXES  
WHERE TABLE_NAME = 'NOME_DA_TABELA';
```

entre aspas simples
e em letras maiúsculas



– SQL –
Linguagem de Manipulação
de Dados

Laboratório de Bases de Dados
Profa. Dra. Cristina Dutra de Aguiar Ciferri

Oracle e SQL DML

- **INSERT INTO ...**
 - insere dados em uma tabela
- **DELETE FROM ... WHERE ...**
 - remove dados de tabelas já existentes
- **UPDATE ... SET ... WHERE ...**
 - altera dados específicos de uma tabela

INSERT

```
INSERT INTO nome_tabela  
VALUES ( V1, V2, ..., VN );
```

- Ordem dos atributos deve ser mantida

```
INSERT INTO nome_tabela (A1, A2, ..., An)  
VALUES ( V1, V2, ..., VN );
```

- Ordem dos atributos não precisa ser mantida

INSERT

```
INSERT INTO nome_tabela  
SELECT ...  
FROM ...  
WHERE ... ;
```

- Tuplas resultantes da cláusula SELECT serão inseridas na tabela nome_tabela

DELETE

```
DELETE FROM nome_tabela  
WHERE predicado ;
```

- Cláusula WHERE
 - é opcional:
 - todas as tuplas da tabela são eliminadas
 - a tabela continua a existir
- Predicado
 - pode ser complexo

UPDATE

```
UPDATE nome_tabela  
  SET coluna = <valor>  
  WHERE predicado ;
```

- Cláusula WHERE
 - é opcional
- Exemplos de <valor>
 - NULL
 - 'string'
 - upper ('string')