

– SQL –
Linguagem de Definição de
Dados

Banco 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

Composição do SQL

- Linguagem de Definição dos Dados
 - comandos para a definição, a modificação e a remoção de relações, além da criação e da remoção de índices
- Linguagem Interativa de Manipulação dos Dados
 - comandos para a consulta, a inserção, a remoção e a modificação de tuplas no banco de dados

Composição do SQL

- Linguagem de Manipulação dos Dados Embutida
 - pode ser utilizada a partir de linguagens de programação de propósito geral
- Definição de visões
 - SQL DDL inclui comandos para a criação e a remoção de visões
- Restrições de integridade
 - SQL DDL possui comandos para a especificação de restrições de integridade

Composição do SQL

- Autorização
 - SQL DDL inclui comandos para a especificação de direitos de acesso a relações e visões
- Gerenciamento de transações
 - introduz comandos para a especificação do início e do fim das transações
- Recuperação de falhas
 - introduz comandos para utilização do arquivo de *log*

SQL DDL

- **CREATE DATABASE | SCHEMA**
 - cria um esquema de BD relacional
- **DROP DATABASE | SCHEMA**
 - remove um esquema de BD relacional

CREATE DATABASE

```
CREATE {DATABASE | SCHEMA} nome  
    [USER `username` [PASSWORD `password` ]  
    ... ;
```

- Cria um esquema de BD relacional
 - agrupa as tabelas e outros comandos que pertencem à mesma aplicação
 - identifica o proprietário do esquema
- Característica
 - o esquema inicial não possui tabelas/dados

DROP DATABASE

```
DROP DATABASE {DATABASE | SCHEMA} nome  
[CASCADE | RESTRICT];
```

- Remove um esquema de BD relacional
 - tabelas/dados
 - índices
 - arquivos de log

} quaisquer elementos associados
- Usuários autorizados
 - proprietário do banco de dados
 - DBA ou usuário com privilégio de *root*

DROP DATABASE

- **CASCADE**
 - remove um esquema de BD, incluindo todas as suas tabelas e os seus outros elementos
- **RESTRICT**
 - remove um esquema de BD somente se não existirem elementos definidos para esse esquema

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_tabela ( A1 D1 R1,  
                             A2 D2 R2,  
                             ...  
                             An Dn Rn );
```

- Cria uma nova tabela (relação)
- Cria os atributos da nova tabela, com
 - nome do atributo: A_i ($1 \leq i \leq n$)
 - tipo de dado (domínio do atributo): D_i
 - restrições que atuam no atributo: R_i

Exemplos de Tipos de Dados

- Numéricos
 - smallint | integer | float | double precision
 - decimal | numeric
- Hora/Data
 - date | time | timestamp
- Strings
 - char | character | varchar | ...
- Outros
 - blob

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 DEFAULT
 - associa um valor *default* para um atributo, caso nenhum outro valor seja especificado
- Cláusula CHECK
 - especifica um predicado que precisa ser satisfeito por todas as tuplas de uma relação
 - exemplos
 - saldo int CHECK (saldo >= 0)
 - nível char(15) CHECK (nível IN ('Bacharelado', 'Mestrado', 'Doutorado'))

Restrições de Integridade

- Integridade referencial
 - dependência existente entre a chave estrangeira de uma relação e a chave primária da relação relacionada
 - problemas
 - atualização ou exclusão de elementos da chave primária sem fazer um ajuste coordenado nas chaves estrangeiras
 - inclusão ou alteração de valores não nulos na chave estrangeira que não existam na chave primária

Restrições de Integridade

- Cláusula FOREIGN KEY

- características

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

- sintaxe

- FOREIGN KEY (atributos)

- REFERENCES nome_relação (atributos)

- [ON UPDATE [NO ACTION | CASCADE | SET NULL | SET DEFAULT]]

- [ON DELETE [NO ACTION | CASCADE | SET NULL | SET DEFAULT]]

DROP TABLE

```
DROP TABLE nome_tabela ;
```

- Remove uma tabela existente do BD
 - dados
 - metadados
 - índices
 - gatilhos que referenciam a tabela
- Usuários autorizados
 - proprietário do banco de dados
 - DBA ou usuário com privilégio de *root*

ALTER TABLE

```
ALTER TABLE nome_tabela;
```

- Altera o esquema de uma tabela do BD
 - adiciona
 - remove
 - altera
- } colunas ou restrições de integridade

Exemplos: ALTER TABLE

```
ALTER TABLE nome_tabela  
  ADD (A1 D1 R1),  
  ...  
  ADD (An Dn Rn)
```

– inclui novas colunas na tabela

```
ALTER TABLE nome_tabela DROP A1
```

– elimina uma coluna já existente da tabela

Exemplos: ALTER TABLE

```
ALTER TABLE nome_tabela ALTER [COLUMN] A1 TO A2
```

- modifica o nome de uma coluna existente de A₁ para A₂

```
ALTER TABLE nome_tabela  
ALTER [COLUMN] A1 TYPE SMALLINT
```

- modifica o tipo de dado de uma coluna

SQL DDL

- **CREATE DOMAIN**
 - cria um domínio para um tipo de dados
- **DROP DOMAIN**
 - remove um domínio existente do BD
- **ALTER DOMAIN**
 - altera a definição de domínio