

SQL: histórico e DDL

Carlos A. Heuser
2005

03/1

SQL

- **SQL** ("structured query language") é uma linguagem comercial de definição e manipulação banco de dados relacional
- SQL (inicialmente chamava-se **SEQUEL**) surgiu no centro de pesquisa de San Jose da IBM, dentro do projeto System R
- SQL é padrão de fato (mesmo SGBDs como INGRES que possuíam outras linguagens oferecem uma interface SQL)
- SQL é padrão de direito (ISO):
 - SQL1 aprovado em 1986, com modificações em 1989
 - SQL2 aprovado em 1992
 - SQL3 aprovado em 1999
 - SQL4 aprovado em 2004

03/2

Componentes de SQL

- SQL é uma linguagem completa de manipulação de banco de dados.
- Oferece as seguintes funcionalidades:
 - Uma **DDL** para definição do esquema da base de dados
 - Uma **DML** para programação de consultas e transações que inserem, removem e alteram linhas de tabelas
 - Uma versão de **SQL embutida** em linguagens de programação de 3ª geração (COBOL, C, ...) estendendo-as para a manipulação de banco de dados
 - Uma versão "**call-level interface**" (**ODBC** - open database connectivity) a partir de SQL/3
 - Instruções para definição de **visões** (tabelas virtuais vistas por um usuário ou uma classe de usuários)
 - Instruções para **controle** de autorização de **acesso**
 - Instruções para controle de **transações** e **concorrência**
 - Instruções para especificação de **restrições de integridade**

03/3

Aderência a padrões

- A aderência a padrões de SQL é importante para aqueles que:
 - escrevem comandos de SQL em suas aplicações (SQL não fica oculto por algum gerador de telas, ...)
 - desejam portar aplicações a vários SGBD
- O padrão é irrelevante para aqueles que usam ferramentas como geradores de telas, relatórios, etc. proprietários - o usuário está preso ao fornecedor da ferramenta
- Praticamente todo fornecedor de SGBD afirma que seu SQL é compatível com o padrão

03/4

Validação de padrões

- ❑ Compatibilidade com padrão somente pode ser garantida se verificada por um órgão independente.
- ❑ Nos EUA, há um órgão do governo (**NIST**) que faz a validação de aderência ao padrão SQL.
- ❑ Padrão SQL1 (86/89) é testada através do conjunto de testes FIPS 127-1
- ❑ Padrão SQL2 de 92 é testado através do conjunto de testes de testes (FIPS 127-2)
- ❑ Os testes de aderência valem apenas para uma plataforma de software/hardware
- ❑ Grandes fornecedores (Oracle, Sybase, DB/2, SQL/Server) normalmente têm pelo menos SQL/2 entry level

03/5

Níveis de SQL/2

- ❑ A norma SQL/2 é formada por **vários níveis** que oferecem cada vez mais funções:
 - **entry level**, conjunto mínimo para considerar produto como SQL/2, implementado pelos vários fornecedores
 - **transitional level** (conjunto de testes somente apareceu em 95)
 - **intermediate level**
 - **full level**, norma completa, ainda sem testes, nem produtos
- ❑ SGBD comerciais implementam variados níveis
- ❑ Não existe portabilidade real entre diferentes SGBD
- ❑ SQL/3 começa a aparecer em alguns produtos (DB2)

03/6

DDL - Criação de banco de dados

- ❑ SQL/2 não oferece instruções para criação de BD.
- ❑ Alguns produtos (SQL/Server) têm instruções de DDL:
 - **Create Database**
cria uma base de dados vazia
 - **Drop Database**
elimina uma base de dados
- ❑ Outros têm abordagens variadas
 - Oracle cria o BD como parte da instalação do software
 - INGRES tem um utilitário

03/7

Instruções da DDL

- ❑ SQL oferece três instruções para definição do esquema da base de dados:
 - **Create Table**
define a estrutura de uma tabela, suas restrições de integridade e cria a tabela vazia
 - **Drop Table**
elimina a tabela da base de dados
 - **Alter Table**
permite modificar a definição de uma tabela

03/8

Criação de Tabelas (Peca)

```
CREATE TABLE Peca
(CodPeca          CHAR(4)      NOT NULL,
 NomePeca         VARCHAR(50)  NOT NULL,
 CorPeca          CHAR(10)     NULL,
 PesoPeca         INTEGER      NULL,
 CidadePeca       VARCHAR(40)  NULL,
 PRIMARY KEY (CodPeca))
```

03/9

Criação de Tabelas (Fornec)

```
CREATE TABLE Fornec
(CodFornec        CHAR(4)      NOT NULL,
 NomeFornec       VARCHAR(50)  NOT NULL,
 StatusFornec     INTEGER,
 CidadeFornec     VARCHAR(40),
 CGC              NUMBER(14)   NULL,
 PRIMARY KEY (CodFornec),
 UNIQUE KEY (CGC))
```

03/10

Criação de Tabelas (Embarq)

```
CREATE TABLE Embarq
(CodPeca          CHAR(4)      NOT NULL,
 CodFornec        CHAR(4)      NOT NULL,
 QtdeEmbarc       INTEGER      NOT NULL,
 PRIMARY KEY (CodPeca,CodFornec),
 FOREIGN KEY (CodPeca) REFERENCES Peca (CodPeca),
 FOREIGN KEY (CodFornec) REFERENCES Fornec)
```

03/11

Observações sobre a definição de tabelas:

- ☐ Em SQL/2 o conjunto de **domínios** de valores de atributos é **fixo**.
- ☐ Desejável (SQL/3):
 - domínio definível pelo usuário (exemplo: dias da semana, meses do ano, ...)
- ☐ Nos SGBD comerciais são oferecidos **domínios adicionais** aos do padrão (CHAR, VARCHAR, INTEGER, REAL,...) destinados a aplicações especiais como DATE, CURRENCY e domínios para armazenar campos longos (BLOBS, até 2 gigabytes) destinados a conter imagens, sons, vídeos, etc. (maioria aparece no SQL/2)
- ☐ A cláusula NOT NULL especifica que uma coluna não admite o valor vazio (requerido para colunas que sejam chave primária)
- ☐ *Default* é NULL permitido (exceto Sybase e SQL/Server antigos)
- ☐ As colunas de uma tabela são classificadas na ordem de sua definição (linhas não tem classificação)

03/12

Restrições de chave

- ❑ No SQL original (System R) e no SQL padrão original (86) não havia cláusulas para especificar chaves.
- ❑ A única maneira de definir chave primária era através da criação de um índice sem duplicatas sobre a coluna.
- ❑ Não havia forma declarativa de definir chaves estrangeiras.
 - SGBD não dava suporte a integridade referencial.
 - Usuário é obrigado a programar os testes de chaves em sua aplicação

03/13

Restrições de chave

- ❑ Padrão (86/89) foi estendido para especificar chaves:
 - **primária**
 - **estrangeira**
 - **alternativa** (*unique key*)
- ❑ Praticamente todos produtos comerciais incluem a definição de chaves

03/14

Chave estrangeira - cláusula ON DELETE

- ❑ Caso nada seja especificado na definição da chave estrangeira, uma linha que contenha uma chave primária referenciada em chave estrangeira não pode ser excluída (regra `RESTRICT`)
- ❑ Pode ser definida a propagação da exclusão da linha para as linhas que a referenciam

```
CREATE TABLE Embarq
(CodPeca          CHAR(4)      NOT NULL,
 CodFornec        CHAR(4)      NOT NULL,
 QtdeEmbarc       INTEGER      NOT NULL,
 PRIMARY KEY (CodPeca,CodFornec)
 FOREIGN KEY (CodPeca) REFERENCES Peca
 ON DELETE CASCADE,
 FOREIGN KEY (CodFornec) REFERENCES Fornec)
```

03/15

Chave estrangeira - cláusula ON DELETE

- ❑ Sintaxe completa:

```
...
FOREIGN KEY ... REFERENCES ...
    ON DELETE {CASCADE | SET NULL | RESTRICT}
```

03/16

ON DELETE SET NULL

```
CREATE TABLE Professor
(CodProf      ...,
 NomeProf     ...,
 CodTit       INTEGER      NULL,
 PRIMARY KEY (CodProf),
 FOREIGN KEY (CodTit) REFERENCES Titulacao
    ON DELETE SET NULL
    ON UPDATE CASCADE)
```

03/17

Chave estrangeira - cláusula ON DELETE

☐ SET NULL

- Especifica que as chaves estrangeiras que referenciam a linha excluída devem ser tornadas vazias
- Válido somente para chaves estrangeiras opcionais

☐ RESTRICT (default)

- Especifica que uma linha não pode ser excluída caso existam chaves estrangeiras que a referenciem

...

```
FOREIGN KEY ... REFERENCES ...
    ON DELETE {CASCADE|SET NULL|RESTRICT}
```

03/18

Chave estrangeira - cláusula ON UPDATE

```
FOREIGN KEY ... REFERENCES ...
    ON UPDATE {CASCADE|SET NULL|RESTRICT}
```

☐ CASCADE

- alteração da chave primária é propagada para as chaves estrangeiras que a referenciam

☐ SET NULL

- chaves estrangeiras que referenciam a linha alterada devem ser tornadas vazias
- válido somente para chaves estrangeiras opcionais

☐ RESTRICT

- uma chave primária não pode ser alterada, caso existam chaves estrangeiras que a referenciem

03/19

Eliminação de Tabelas

- ☐ Para eliminar completamente uma tabela (vazia ou não) da base de dados, é usada a instrução:

```
DROP TABLE Embarq
```

- ☐ SQL/2 inclui cláusulas `RESTRICT` e `CASCADE` (obrigatória) que informam se a exclusão deve ser propagada ou não para objetos definidos com base na tabela (visões)

☐ Exemplo:

```
DROP TABLE Embarq RESTRICT
```

- Exclui a tabela somente se não existirem visões definidas com base na tabela

03/20

Alteração da definição de Tabelas

- ❑ A instrução `ALTER TABLE` serve para modificar a definição original da tabela.
- ❑ Nem todas modificações são permitidas
- ❑ Primeiros SGBDs e SQL/1 somente permitiam adicionar colunas

03/21

Alteração da definição de Tabelas

- ❑ Para modificar a estrutura de tabelas já existentes na base de dados, há uma instrução que permite adicionar colunas a tabelas:

```
ALTER TABLE Embarq ADD DataEmbarq DATE
```

- ❑ Observe-se que:
 - A instrução adiciona uma nova coluna com o valor vazio para todas linhas
 - Os valores para as diversas linhas devem ser adicionadas através de instruções da DML
 - Não pode ser especificada a cláusula `NOT NULL` já que a coluna é criada com o valor vazio (a menos que seja especificado um valor *default*)

03/22

Alteração da definição de tabelas

- ❑ Procedimento caso SGBD não permite a alteração desejada
 1. Armazenar o conteúdo da tabela em tabela temporária ou arquivo do sistema operacional
 2. Eliminar todas referencias à tabela antiga
 3. Eliminar a tabela antiga (`DROP TABLE`)
 4. Definir a nova tabela (`CREATE TABLE`)
 5. Carregar a nova tabela a partir da tabela intermediária ou arquivo do sistema operacional criado no passo 1
 6. Reincluir as referências à tabela

03/23

ALTER TABLE - mudando as chaves

- ❑ `ALTER TABLE` permite incluir ou excluir restrições de chave
- ❑ Para excluir uma chave estrangeira é necessário que ela tenha recebido um nome quando de sua definição

```
ALTER TABLE Representantes  
DROP PRIMARY KEY
```

```
ALTER TABLE Representantes  
ADD FOREIGN KEY  
    (CodFilial) REFERENCES Filiais
```

```
ALTER TABLE Representantes  
DROP FOREIGN KEY  
    (CodFilial) REFERENCES Filiais
```

03/24

ALTER TABLE - definição de chaves

□ ALTER TABLE com definição de chaves permite separar a DDL em duas seções:

1. Primeira seção contém apenas os CREATE TABLE com a estrutura da base de dados
2. Segunda seção contém as restrições de chave através de instruções ALTER TABLE

03/25

SQL/2 - Tratamento generalizado de CONSTRAINTS

□ Sintaxe para chaves vista até aqui é antiga (saindo de uso)

□ A tendência em SQL/2 é tratar de forma uniforme todas restrições de integridade (chave primária, chave única, chave estrangeira, ...)

□ Em SQL restrições de integridade são chamadas de **CONSTRAINT**

03/26

SQL/2 - Tratamento generalizado de CONSTRAINTS

```
create table EMBARQ
(   CODFORNEC      CHAR(4)           not null,
    CODPECA        CHAR(4)           not null,
    QUANT          INTEGER           not null,
    constraint PK_EMBARQ primary key
        (CODFORNEC,CODPECA))

alter table EMBARQ
add constraint FK_EMBARQ_RELATION__PECAS
    foreign key (CODPECA)
        references PECAS (CODPECA)

alter table EMBARQ
add constraint FK_EMBARQ_RELATION__FORNEC
    foreign key (CODFORNEC)
        references FORNEC (CODFORNEC)
```

03/27