

- ▶ Linguagem de Definição dos Dados
 - ▶ Tipos em SQL
 - ▶ Manipular Tabelas (relações)
 - ▶ Integridade e Segurança
- ▶ Linguagem de Manipulação dos Dados
 - ▶ Estrutura básica
 - ▶ Operações com conjuntos
 - ▶ Funções de agregação
 - ▶ Valores nulos
 - ▶ Junções
 - ▶ Sub-consultas embutidas
 - ▶ Relações derivadas
 - ▶ Vistas
 - ▶ Modificação da Base de Dados
 - ▶ União, Diferença, Divisão em SQL
- ▶ SQL Embutido

Tipos em SQL

- ▶ `char(n)`: cadeia de caracteres de comprimento fixo n .
- ▶ `varchar(n)`: cadeia de caracteres de comprimento variável, com o máximo n especificado pelo utilizador.
- ▶ `int`: inteiro (um subconjunto finito dos inteiros, dependente da máquina).
- ▶ `smallint`: inteiro pequeno (um subconjunto do tipo `int`).
- ▶ `numeric(p,d)`: número de **vírgula fixa**, com de p dígitos no total e com d casas decimais.
- ▶ `real`, `double precision`: Números de **vírgula flutuante**, com precisão dependente da máquina.
- ▶ `float(n)`: número de vírgula flutuante, com um mínimo de precisão de n dígitos.

Os valores «nulos» (NULL) são permitidos em todos os tipos de dados.

A declaração de um atributo como NOT NULL proíbe os valores nulos para esse atributo.

Linguagem para especificar a informação acerca de cada relação, incluindo:

- ▶ O esquema de cada relação.
- ▶ O domínio de valores associados com cada atributo.
- ▶ Restrições de integridade.
- ▶ O conjunto de índices a manter para cada relação.
- ▶ Informação de segurança e autorização para cada relação.
- ▶ As estruturas de armazenamento físico em disco de cada relação.

Tipos Data/Tempo em SQL (cont.)

- ▶ `date`: datas, contendo um ano com (4 dígitos), mês e dia, («2001-7-27»);
- ▶ `time`: tempo (diário), em horas, minutos e segundos, («09:00:30.75»);
- ▶ `timestamp`: data mais hora, («2001-7-27 09:00:30.75»).
- ▶ `Interval`: período de tempo.

A subtracção de dois valores de `date/time/timestamp` devolve um intervalo.

Os valores de intervalos podem ser adicionados a valores de `date/time/timestamp`.

Pode-se extrair campos do valor `date/time/timestamp`.

Tipos em MySQL

```
TINYINT[(tamanho)] [UNSIGNED] [ZEROFILL]
SMALLINT[(tamanho)] [UNSIGNED] [ZEROFILL]
MEDIUMINT[(tamanho)] [UNSIGNED] [ZEROFILL]
INT[(tamanho)] [UNSIGNED] [ZEROFILL]
INTEGER[(tamanho)] [UNSIGNED] [ZEROFILL]
BIGINT[(tamanho)] [UNSIGNED] [ZEROFILL]
REAL[(tamanho,decimais)] [UNSIGNED] [ZEROFILL]
DOUBLE[(tamanho,decimais)] [UNSIGNED] [ZEROFILL]
FLOAT[(tamanho,decimais)] [UNSIGNED] [ZEROFILL]
DECIMAL(tamanho,decimais) [UNSIGNED] [ZEROFILL]
NUMERIC(tamanho,decimais) [UNSIGNED] [ZEROFILL]
CHAR(tamanho) [BINARY | ASCII | UNICODE]
VARCHAR(tamanho) [BINARY]
DATE
TIME
TIMESTAMP
DATETIME
TINYBLOB
BLOB
MEDIUMBLOB
LONGBLOB
TINYTEXT
TEXT
MEDIUMTEXT
LONGTEXT
ENUM(value1,value2,value3,...)
SET(value1,value2,value3,...)
```

2014/09/29 (v06)
153/311

Restrições de integridade

- not null
- primary key (A_1, \dots, A_n)
- unique (A_1, \dots, A_n)
- check (P), em que P é um predicado.

Exemplo:

```
CREATE TABLE Balcao
(balcaoNome char(15) NOT NULL,
 balcaoCidade char(30),
 depositos integer,
 PRIMARY KEY (balcaoNome),
 CHECK (depositos >= 0))
```

2014/09/29 (v06)
155/311

Instrução Create Table

- Uma tabela SQL é definida recorrendo ao comando CREATE TABLE:

```
CREATE TABLE  $r$  ( $A_1 D_1, A_2 D_2, \dots, A_n D_n,$   
restrição_de_integridade 1,  
...,  
restrição_de_integridade  $k$ )
```

- i
 - r é o nome da relação;
 - A_i é o nome de um atributo no esquema de relação r ;
 - D_i é o tipo de dados dos valores no domínio do atributo A_i .
- Exemplo:

```
CREATE TABLE Balcao
(balcaoNome char(15) not null,
 balcaoCidade char(30),
 depositos integer)
```

2014/09/29 (v06)
154/311

Instrução Drop Table

O comando DROP TABLE remove da base de dados toda a informação sobre a relação (e não apenas os dados).

```
DROP TABLE Emprestimo
```

Caso haja chaves externas deve-se utilizar a declaração cascade constraints.

```
DROP TABLE Emprestimo CASCADE CONSTRAINTS
```

2014/09/29 (v06)
156/311

Instrução Alter Table

O comando `ALTER TABLE` é utilizado para modificar o esquema, ou as restrições sobre relações já existente.

- ▶ Para adicionar novos atributos:

```
ALTER TABLE r ADD A D
```

em que A é o nome do atributo a adicionar à relação r e D o domínio de A .

Todos os tuplos existentes ficam com `NULL` no novo atributo.

- ▶ Para eliminar atributos de uma relação

```
ALTER TABLE r DROP A
```

em que A é o nome de um atributo na relação r .

2014/09/29 (v06)
157/311

Integridade e Segurança

- ▶ Restrições ao Domínio
- ▶ Integridade Referencial
- ▶ Asserções
- ▶ Segurança e Autorizações

2014/09/29 (v06)
159/311

Instrução Alter Table (cont.)

- ▶ Para adicionar novas restrições:

```
ALTER TABLE r ADD CONSTRAINT N R
```

em que N é um nome dado à nova restrição e R define a restrição. Por exemplo:

```
ALTER TABLE Conta ADD CONSTRAINT saldoPos CHECK (balanco > 0)
```

- ▶ Para remover restrições (definidas previamente com um nome):

```
ALTER TABLE r DROP CONSTRAINT N
```

por exemplo:

```
ALTER TABLE Conta DROP CONSTRAINT saldoPos
```

2014/09/29 (v06)
158/311

Restrições ao Domínio

- ▶ As restrições de integridade impõem-se para garantir que os dados fiquem protegidos contra «estragos» acidentais. Devem assegurar que da actualização dos dados não resulta a perda da consistência.
- ▶ [Restrições ao domínio](#) são a forma mais elementar de restrição de integridade.
- ▶ Testam condições sobre valores a introduzir em atributos. Fazem-no, restringindo o domínio do atributo em causa.
- ▶ No momento em que se define a tabela.
- ▶ A forma mais comum de restrição ao domínio é a proibição do valor `NULL` (`NOT NULL` depois do atributo).
- ▶ Podem-se impor outras restrições usando, depois dum nome de atributo A , `CHECK(condição(A_1, \dots, A_n))` onde `condição(...)` denota uma condição imposta sobre os atributos A_1, \dots, A_n .

2014/09/29 (v06)
160/311

Exemplos de restrições ao domínio

```
create table Alunos (  
  numAluno number(6) not null,  
  nome varchar(30) not null,  
  local varchar(25),  
  dataNsc date not null,  
  sexo char(1) not null check (sexo in ('F','M')),  
  codCurso number(3) not null);
```

```
create table Produtos (  
  idProduto number(6) not null,  
  nome varchar2(30) not null,  
  iva number(2) not null check (iva in (5,12,21)));
```

```
create table Balcao (  
  balcaoNome char(15),  
  balcaoCidade char(30),  
  depositos number check (depositos >= 0));
```

2014/09/29 (v06)
161/311

Modificação da Base de Dados

Os testes abaixo devem ser efectuados de modo a preservar-se a seguinte restrição de integridade referencial:

$$\Pi_{\alpha}(r_2) \subseteq \Pi_K(r_1)$$

Inserção: se um tuplo t_2 é inserido em r_2 , o sistema tem de garantir que existe um tuplo t_1 em r_1 tal que $t_1[K] = t_2[\alpha]$.

Ou seja

$$t_2[\alpha] \in \Pi_K(r_1)$$

Remoção: se um tuplo t_1 é removido de r_1 , o sistema deve calcular o conjunto de tuplos em r_2 que referenciam t_1 :

$$\sigma_{\alpha=t_1[K]}(r_2)$$

Se este conjunto não é vazio, ou o comando de remoção é rejeitado, ou os tuplos que referenciam t_1 devem ser eles próprios removidos (remoções em cascata são possíveis).

2014/09/29 (v06)
163/311

Integridade Referencial

Chaves Externas (também designadas por «estrangeiras»)

- Garante que um valor que ocorre numa relação para um certo conjunto de atributos também ocorre num outro conjunto de atributos de outra relação.

Exemplo: Se «Coimbra-central» é o nome de uma agência que ocorre num dos tuplos da relação *Conta*, então existe um tuplo na relação *Balcao* para o balcão «Coimbra-central».

- Definição Formal:

Sejam $r_1(R_1)$ e $r_2(R_2)$ duas relações com chaves primárias K_1 e K_2 respectivamente.

O subconjunto α de R_2 é uma chave externa referindo K_1 na relação r_1 , se para todo t_2 em r_2 existe um tuplo t_1 em r_1 tal que $t_1[K_1] = t_2[\alpha]$.

Aquando da escrita do modelo relacional as chaves externas são distinguidas colocando-as a sublinhado a tracejado.

2014/09/29 (v06)
162/311

Modificação da Base de Dados (Cont.)

Actualização. Existem duas situações:

Se um tuplo t_2 é actualizado na relação r_2 em que é modificado o valor da chave externa α , então é efectuado um teste similar ao da inserção. Seja t_2' o novo valor do tuplo t_2 . O sistema deve garantir que

$$t_2'[\alpha] \in \Pi_K(r_1)$$

Se o tuplo t_1 é actualizado em r_1 , e a operação altera o valor da chave primária (K), então é efectuado um teste semelhante ao da remoção. O sistema deve calcular

$$\sigma_{\alpha=t_1[K]}(r_2)$$

usando o valor anterior de t_1 (o valor antes da actualização). Se o conjunto é não vazio, a actualização pode ser rejeitado, ou a actualização pode ser propagada em cascata, ou os tuplos podem ser removidos.

2014/09/29 (v06)
164/311

Integridade Referencial em SQL

As chaves **primárias**, **candidatas** e **externas** podem ser especificadas na instrução SQL `create table`:

- ▶ A cláusula `primary key` inclui a lista dos atributos que formam a chave primária.
- ▶ A cláusula `unique key` inclui a lista dos atributos que formam a chave candidata.
- ▶ A cláusula `foreign key` inclui a lista de atributos que constituem a chave externa assim como o nome da relação referenciada pela chave externa.

2014/09/29 (v06)
165/311

Integridade Referencial em SQL — Exemplos

```
CREATE TABLE Cursos (  
    codCurso number(3) NOT NULL,  
    nome varchar(35) NOT NULL,  
    PRIMARY KEY (codCurso));
```

```
CREATE TABLE Cadeiras (  
    codCadeira number(3) NOT NULL, ...,  
    PRIMARY KEY (codCadeira));
```

```
CREATE TABLE CursoCadeira (  
    codCurso number(3) NOT NULL,  
    codCadeira number(3) NOT NULL,  
    semestre number(2) NOT NULL,  
    PRIMARY KEY (codCurso, codCadeira),  
    FOREIGN KEY (codCurso) REFERENCES Cursos,  
    FOREIGN KEY (codCadeira) REFERENCES Cadeiras);
```

2014/09/29 (v06)
166/311

Acções em Cascata em SQL

```
CREATE TABLE Conta  
...  
    FOREIGN KEY (balcaoNome) REFERENCES Balcao  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
...)
```

- ▶ Com as cláusulas `ON DELETE CASCADE`, se a remoção de um tuplo na relação `Balcao` resulta na violação da restrição da integridade referencial, a remoção propaga-se em «cascata» para a relação `Conta`, removendo o tuplo que referia a agência que tinha sido eliminada.
- ▶ Actualizações em cascata são semelhantes.

2014/09/29 (v06)
167/311

Acções em cascata em SQL (cont.)

- ▶ Se existe uma cadeia de dependências de chaves externas através de várias relações, com um `ON DELETE CASCADE` especificado em cada dependência, uma remoção ou actualização num dos extremos pode-se propagar através de toda a cadeia.
- ▶ Se uma remoção ou actualização em cascata origina uma violação de uma restrição que não pode ser tratada por uma outra operação em cascata, o sistema aborta a transacção. Como resultado, todas as alterações provocadas pela transacção e respectivas acções em cascata serão anuladas.
- ▶ A integridade referencial é verificada apenas no final da transacção
 - ▶ Passos intermédios podem violar a integridade referencial desde que passos posteriores a reponham.
 - ▶ Caso contrário seria impossível criar alguns estados da base de dados, por exemplos, inserir dois tuplos cujas chaves externas apontam um para o outro.

2014/09/29 (v06)
168/311

Integridade Referencial (cont.)

- ▶ Alternativas às operações em cascata:
 - ON DELETE SET NULL
 - ON DELETE SET DEFAULT
- ▶ Valores nulos em atributos de chaves externas complicam a semântica de integridade referencial da SQL, devendo-se evitar recorrendo a `NOT NULL`.
- ▶ Se algum atributo de uma chave externa é nulo, o tuplo satisfaz automaticamente a restrição de integridade referencial!

2014/09/29 (v06)
169/311

Autorizações

Diferentes formas de autorização, em dados da bases de dados:

- ▶ Autorização de leitura - permite ler, mas não modificar dados.
- ▶ Autorização de inserção - permite inserir novos tuplos, mas não modificar tuplos existentes.
- ▶ Autorização de modificação - permite modificar tuplos, mas não apagá-los.
- ▶ Autorização de apagar («delete») - permite apagar tuplos

2014/09/29 (v06)
171/311

Segurança

Segurança — ao contrário das restrições de integridade, que pretendiam proteger a base de dados contra estragos acidentais, a segurança preocupa-se com proteger a base de dados de estragos propositados.

- ▶ A nível do sistema operativo
- ▶ A nível da rede
- ▶ A nível físico
- ▶ A nível humano
- ▶ A nível da base de dados

Mecanismos de autenticação e autorização para permitir acessos selectivos de (certos) utilizadores a (certas) partes dos dados

2014/09/29 (v06)
170/311

Autorizações (Cont.)

Diferentes formas de autorização, para alterar esquemas:

- ▶ Autorização de índice - permite criar e apagar ficheiros de indexação.
- ▶ Autorização de recursos - permite criar novas relações.
- ▶ Autorização de alteração - permite criar e apagar atributos duma relação.
- ▶ Autorização de remoção («drop») - permite apagar relações.

2014/09/29 (v06)
172/311