TÉCNICO LISBOA

Bases de Dados

Aula 07: Tradução Modelo Relacional para SQL

Prof. Paulo Carreira

Sumário

- Definição de tabelas
- Restrições
  - Chave
  - Chave estrangeira
  - Unicidade
- Tipos de dados
- Inserção, remoção e actualização de valores

Sumário

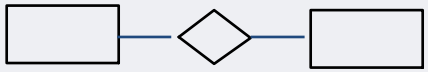
- Tradução de restrições
  - Restrições de coluna
    - Chave
    - Unicidade
    - Chave estrangeira
  - Restrições decorrentes de obrigatoriedades
  - Restrições de domínio
  - Restrições de linha

Concepção de uma Base de Dados

Especificação de Requisitos

- requisito funcional 1:
- requisito funcional 2:
- …
- **regra de integridade 1:**
- **regra de integridade 2:**
- …

Modelo Conceptual
Modelo EA



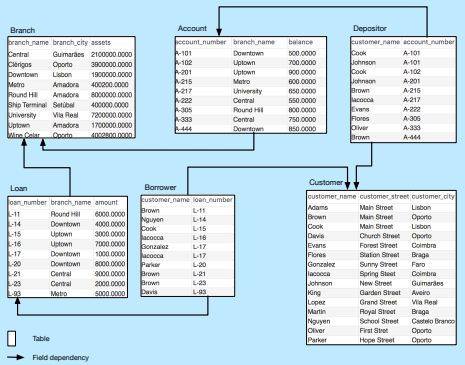
TRADUÇÃO

Modelo Relacional

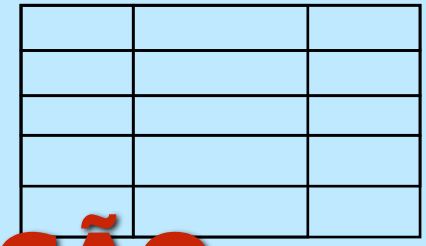
R1(X, Y, Z)

R2(A, B, C)

TRADUÇÃO



Esquema Relacional



Relacional vs. SQL

- Mapping of concepts
  - Relations ⟶ Tables
  - Attributes ⟶ Fields/Columns
  - Integrity Constraints ⟶ Integrity Constraints
- Tables can have duplicate records (unless we specify a key)

Tradução relacional-SQL

Tradução de Relações e Atributos

Relações e atributos

relation_name(a, b, c, d)

```
create table relation_name(
        a integer,
        b varchar(80),
        c numeric(12,4),
        d date
);
```

- **Passo 1**: Cada relação é convertida na tabela correspondente
- **Passo 2**: Cada atributo é convertido numa coluna com o tipo apropriado

SQL/Database Field Types

Text Field Types

- **varchar(*n*)** – cadeia de caracteres de tamanho variável, máximo *n (n < 4000)*
- **char(*n*)** – cadeia de caracteres de tamanho fixo *n*.
  - Utilizado para códigos de tamanho fixo.
  - Preenchida com espaços em branco à direita caso o tamanho da cadeira inserida seja <*n*
- **text** – campo de texto de tamanho variável indeterminado (tipicamente limitado a 65535 caracteres)
  - Common use English 4.79 chars/word (7.6 chars/ word in the dictionary)

Integer numeric types

- **integer** – An integer with up to 9 digits
  - -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647)
  - 4 bytes
- **smallint** – An integer with up to 4 digits
  - 2^15 (-32,768) to 2^15-1 (32,767)
  - 2 bytes
- **bigint** – An integer with up to 18 digits
  - -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807)
  - 8 bytes

Concepts

- A fixed point number always has a fixed number of digits in the form **`IIII.DDDD`** `where` **`IIII`** represents integer part and **DDDD** represents the decimals
- With 4 integer digits and four decimals:
  - Largest fixed point number number is **`9999.9999`**
  - Smallest fixed point number number is **`0000.0001`**
- Floating point would admit **`9999999.9`**, as well as **`0.0000001`**

Fixed point numeric types

- **numeric(*p*,*s*)** – *p* digits, *s* scale
  - A sufficiently but not arbitrarily large number
  - Precision: total number of digits
  - Scale: number of digits to right of the decimal point
- Inserting in numeric with a precision or scale larger than specified will result in an error or in a warning; the value is often be truncated.

Floating point types

- Números de vírgula flutuante permitem escrever números arbitrariamente grandes (distância entre galáxias), ou arbitrariamente pequenos (diâmetro do núcleo de um átomo)
- **float/real** (4 bytes) – número de vírgula flutuante com 6 dígitos de precisão 1E-37 to 1E+37
- **double(*n*)** (8 bytes) – número de vírgula flutuante de dupla precisão (15 dígitos) capaz de representar números entre 1E-307 to 1E+308

Notes

- Floating point calculations **are faster** than fixed-point calculations
- Some values cannot be converted precisely to floating point to are **stored as approximations**
- Testing two floating point numbers for **equality** does **not always work as expected**
- **Never use floats** when exact storage and calculations (for monetary values)

Comportamento do Float

```
create table test(
 x float,
 y float
);

insert into test values(1.2, 1.2);

select x+y from test;
```

```
2.4000000953674316
```

```
select (1.0/3.0)*3.0;
```

Date and Time

- **date** (4 bytes) – permite escrever uma data (sem hora) 4713 AC até 5874897 DC
- **time** (4 bytes) – permite escrever uma hora (sem data) até aos micro-segundos
- **datatetime** (8 bytes) – data e hora (desde as 00:00:00 1753-01-01 até às 23:59:59 9999-12-31
- **timestamp** (8 bytes) – permite escrever uma data e hora (i.e. um instante) até aos micro-segundos
- **interval** (16 bytes) – permite capturar a diferença entre duas datas, duas horas, ou dois timestamps. Gama de valores entre -178.000.000 e 178.000.000 anos

Nota: Não de podem somar datas nem horas mas podem somar-se datas a intervalos e horas a intervalos.

Auto-increment fields

- **serial / auto-increment** (4 bytes) – geram valores sequenciais automaticamente para uma determinada tabela
- Os valores criados não existem no domínio (e não são de facto características das instâncias)
- Tipicamente uma análise cuidada ao domínio revela chaves naturais
- Tornam a sincronização entre bases de dados difícil

Indicative field types

Field	Database Type	Max Size	Min Size	Validation
PERSON/ORGANIZATION DETAILS				
Person Full Name	varchar	80		
Company Name	varchar	200		
Street Address	varchar	255		
City	varchar	30		
Postal Code	varchar	12	2	
Phone Number	varchar	15	3	ITU E.16
Phone Extension	varchar	11		ITU E.16
Language	char	3		ISO 639
Country Name	varchar	70		ISO 3166-1
Latitude	numeric	9,6		
Longitude	numeric	8,6		

Field	Database Type	Max Size	Min Size	Validation
FINANCE				
VAT ID	varchar	20	1	
IBAN	varchar	30		
Credit Card Number	numeric	16		
Money	numeric	16,4		

Field	Database Type	Max Size	Min Size	Validation
ELECTRONIC				
E-mail Address	varchar	254	6	IETF RFC 3696 Checking email addresses
Domain Name	varchar	253	4	
URL	varchar	2083	11	
IP address (incl V6)	varchar	45	11	
GUID	char	36		

Field	Database Type	Max Size	Min Size
SOCIAL NETWORK			
Facebook max name length	varchar	50	
Youtube channel	varchar	20	
Twitter max name length	varchar	15	

Remoção de registos

- Remover registos

```
delete
from tabela
where critério
```

- Remover registos da tabelas **employee** correspondentes a todas as pessoas nascidas a 1 de Jan de 1995

```
delete
from employee
where birthdate = '01-01-1995';
```

Actualização de registos

- Actualizar registos

```
update tabela
set atributo = expr
[where condição]
```

- Actualizar a idade do John Smith na tabela **employee**

```
update employee
set birthdate = '03-03-1993'
where eid = 12345
```

Re-estruturação de tabelas

- Remover colunas

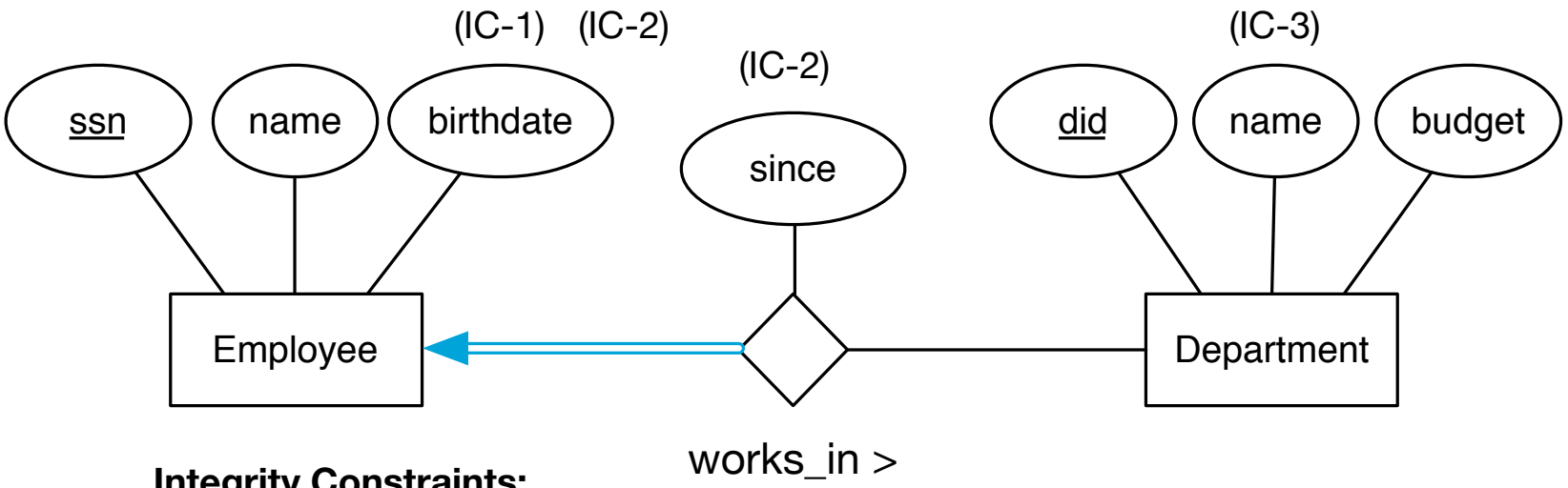
```
alter table tabela
drop nome_coluna
```

- Adicionar colunas

```
alter table tabela
add nome_coluna tipo
```

Tradução de Restrições

Exemplo: E-A



Integrity Constraints:

(IC-1) an employee must be at least 18 years old

(IC-2) **since** must be at least 18 years after **birthdate**

(IC-3) Department names must be unique

Exemplo: Employee

department(did, name, budget)

- **IC-3** Department name is unique

employee(ssn, name, birthdate)

- **IC-1** The difference between today's date and birthdate must be at least 18 years
- ssn: must exist in works_in

works_in(ssn, did, since)

- ssn: FK(employee)
- did : FK(department.did)

Restrições de Coluna

Restrição de chave

Um atributo

relation_name(a, b, c, d)

```
create table table_name(
      a integer,
      b varchar(80),
      c numeric(12,4),
      d date,
      primary key(a)
);
```

Múltiplos atributos

relation_name(a, b, c, d)

```
create table table_name(
      a integer,
      b varchar(80),
      c numeric(12,4),
      d date,
      primary key(a,b)
);
```

- Uma restrição de chave é especificada como a chave primária

primary key (*coluna*$_1$, …, *coluna*$_n$)

Restrição de unicidade

Um atributo "unique"

relation_name(a, **b**, c, d)

```
create table table_name(
        a integer,
        b varchar(80),
        c numeric(12,4),
        d date,
        primary key(a),
        unique(b)
);
```

Múltiplos atributos "unique"/ múltiplas restrições

relation_name(a, b, c, d)

- **unique(b,c)**
- **unique(c,d)**

```
create table table_name(
        a integer,
        b varchar(80),
        c numeric(12,4),
        d date,
        primary key(a),
        unique(b, c),
        unique(c, d)
);
```

- Uma restrição de unicidade have é especificada como uma restrição unique

unique(*coluna*$_1$, …, *coluna*$_n$)

Definição de Esquema – Exemplo 2

```
create table department(
    did integer,
    name varchar(80),
    budget numeric(12,4),
    primary key(did),
    unique(name));
```

```
insert into department values(1, 'Finance', 1000.0);
insert into department values(2, 'Marketing', 2000.0);
```

```
insert into department values(1, 'Engineering', 3000.0);
insert into department values(3, 'Marketing', 4000.0);
```

Definição de Esquema – Exemplo 2

```
create table employee
   (ssn numeric(11),
    name varchar(80),
    birthdate date,
    primary key(ssn)
);
```

```
insert into employee values(12345678901, 'Alice',
   '01-01-1981');
insert into employee values(23456789012, 'Bob',
   '02-02-1992');
```

Restrições de Integridade Referencial

Restrição de chave estrangeira

Um atributo

relation_name(a, **b**, c, d)

- **b : FK(*another_rel*.x)**

```
create table table_name(
      a integer,
      b varchar(80),
      c numeric(12,4),
      d date,
      primary key(a),
      foreign key(b)
         references another_tab(x));
```

Restrição de chave estrangeira

Múltiplos atributos

relation_name(a, **b, c**, d)

- **b,c : FK(*another_rel.x, another_rel.y*)**

```
create table table_name(
      a integer,
      b varchar(80),
      c numeric(12,4),
      d date,
      primary key(a),
      foreign key(b,c)
         references another_tab(x,y));
```

- Uma restrição de chave estrangeira é especificada como uma restrição de chave

foreign key(col_1, …, col_n)
references *nome_tabela*(col_1, …, col_n)

Restrições de Domínio

Null values

- Em SQL todos os campos (não chave), por omissão, podem ter valores a **null**.
- O **null** é um valor especial que significa simultaneamente: não preenchido/ desconhecido/não aplicável
- A utilização do null é ambígua e deve ser evitada

```
insert into employee values(4567890134, 'Daniel',
   null);

select * from employee;
```

Restrição 'not null'

- Para evitar que os campos possam tomar o valor **null**, deve adicionar-se a restrição **not null** à frente do tipo de dados:

```
campo tipo not null
```

```
create table employee
   (ssn numeric(11),
    name varchar(80) not null,
    birthdate date not null,
    primary key(ssn)
);
```

Restrições de domínio

- Para verificar uma restrição de domínio pode acrescentar-se uma instrução **check** que para verifica uma condição sempre que um registo é inserido ou alterado:

check (*condição***)**

- Mais informação sobre a instrução **check**:

https://www.postgresql.org/docs/9.4/static/ddl-constraints.html

Restrições de domínio

```
create table employee(
    ssn numeric(11),
    name varchar(80) not null,
    birthdate date not null,
    gender char(1),
    primary key(ssn),
    check (length(name) > 3),
    check (birthdate > '1920-01-01'),
    check (extract(year from age(birthdate)) > 18),
    check (gender in ('M', 'F'))
);
```

Restrições de domínio

```
create table products (
    product_no integer,
    name text,
    price numeric,
    discounted_price numeric,
    check (price > 0),
    check (discounted_price > 0),
    check (price > discounted_price)
);
```

Validação através de tabelas de referência

- Quando o domínio é grande pode utilizar-se uma tabela com valores pre-preenchidos:

```
create table employee
   (ssn numeric(11),
    name varchar(80) not null,
    birthdate date not null,
    birth_country char(80),
    primary key(ssn),
    check(birth_country
       in (select name from country))
);
```

Sub-query no check faz parte do standard mas não funciona em POSTGRES

Restrições de linha

- Uma **restrição de linha** é uma restrição de integridade que verifica que os dados da linha estão coerentes (por oposição à coluna)

```
create table employee(
    ssn numeric(11),
    name varchar(80) not null,
    birthdate date not null,
    graduation date not null,
    primary key(ssn),
    check (length(name) > 3),
    check (birthdate > '1920-01-01'),
    check (extract(year from age(birthdate)) > 18),
    check graduation > birthdate
);
```

Outras Restrições

Triggers

- Matéria a leccionar mais adiante!