

 deti departamento de  
electrónica, telecomunicações  
e informática

## Linguagem SQL - DDL

Base de Dados - 2019/20  
Carlos Costa



Abstractão que cria SQL

1

 deti

## Linguagem SQL

- Structured Query Language (SQL)
  - SEQUEL
- Linguagem para definir, manipular e questionar uma Base de Dados Relacional.
  - É uma linguagem orientada ao processamento de conjuntos
- 2 sublinguagens principais
  - **DDL - Data Definition Language.** *Depois de aprendermos Álgebra Relacional*
  - **DML - Data Manipulation Language.**
- 1 sublinguagem de controlo BD
  - **DCL - Data Control Language** ~sequencia ...

2

deti

## SQL - Versões

- 1986 (SQL-86 e SQL-87)
  - Publicado pela ANSI e ratificado pela ISO.
- 1989 (SQL-89)
- 1992 (SQL-92)
  - conhecido como SQL2.
- 1999 (SQL:1999)
  - conhecido como SQL 3.
  - inclui expressões regulares, queries recursivas, triggers, tipos não escalares, procedimentos, funcionalidades orientadas a objectos, etc.
- 2003 (SQL:2003)
  - Inclui suporte a XML e colunas com numeração automática.
- 2006 (SQL:2006)
  - Define formas de interacção SQL-XML: como importar e armazenar XML em BD SQL, XQuery, etc.
- 2008
 

↑  
JSON predominia agora !!!
- 2011

3

3

deti

## SQL - SQL Server

- Vamos utilizar, como ferramenta de trabalho, a versão SQL Server (>=2012)

Transact-SQL → "T-SQL"

entre SQL

Diferença com "boolean",  
 pois tem de ser "bit"  
 e @algumas  
 coisas...

"Microsoft SQL Server team has extended the ANSI definition with several enhancements and new commands, and has left out a few commands because SQL Server implemented them differently. The result is Transact-SQL, or T-SQL – the dialect of SQL understood by SQL Server"

"Missing from T-SQL are very few ANSI SQL commands, primarily because Microsoft implemented the functionality in other ways."

Microsoft® SQL Server® 2008 Bible

4

4

## SQL - Hierarquia de Objetos

**catalog** = base de Dados

**schema** = "package" de java

**table** = relação

**column** = atributo

Mas há mais elementos como, por exemplo, triggers, vistas, índices, stored procedures, funções, etc.

5

5

## SQL - catalog, schema e database

**MySQL**

Têm alguns diferenciais, depende do objetivo!!!

O significado destes termos varia de acordo com SGBD

SQL Server: `database_name . schema_name . table_name` → Full Qualified Name

6

6

↳ Para aceder a outras base de dados (e.g.)  
→ Dependente dos privilégios

**SQL - Notas introdutórias**

The diagram shows a file named "My.sql" with the following content:

```

GO ;-->
    -- Se houver uma
    -- falha, não queremos
    -- fazer commit...
    GO
    -->
    -- Transaction
    -- não permitida
    -- entre a metade de
    -- base de dados
    -- Podemos
    -- entrar só
    -- uma porta!
    -->
    -- Delimitador de
    -- Bash...

```

Annotations in red and blue highlight specific parts of the code:

- A blue box highlights the first two lines: "GO ;-->" and "-- Se houver uma". A note above it says: "tabela, linha e coluna (table, row and column)".
- A blue box highlights the line "GO". A note above it says: "... para designar os termos formais: relação, tuplo e atributo do modelo relacional".
- A blue box highlights the line "GO". A note above it says: "Cada instrução SQL termina com um ponto e vírgula (";")".
- A blue box highlights the line "-->". A note above it says: "Comentar um linha --".
- A blue box highlights the block starting with "GO". A note above it says: "Comentar um bloco de instruções /\* ... \*/".

Handwritten notes in green at the bottom left of the slide say: "7 Delimitador de Bash...".

**SQL - Data Definition Language (DDL)**

The diagram shows a table with four columns labeled "a", "b", "c", and "d". A handwritten note in red at the top right says: "posso definir regos para as tabelas...".

- Permite definir várias entidades da BD
- Utilizada para especificar a informação acerca de cada relação:
  - O esquema de cada relação.
  - O domínio de valores associados com cada atributo.
  - Restrições de integridade (entidade e referencial) *(A de domínio é automática)*
  - O conjunto de índices a manter para cada relação
  - ...
- Notas importantes:
  - Há comandos não disponíveis em alguns SGBD...
  - Devemos consultar o manual do SGBD para uma sintaxe mais completa dos comandos.

Handwritten notes in green at the bottom left of the slide say: "8".

## Criar e Eliminar uma Base de Dados

- Criar uma base de dados
 

*Ex: CREATE TABLE*

```
CREATE DATABASE dbname;
dbname - nome da base de dados a criar
```

```
CREATE DATABASE COMPANY;
```
- Eliminar uma base de dados
 

*Eliminam dados*

```
DROP DATABASE dbname;
dbname - nome da base de dados a eliminar
```

```
DROP DATABASE COMPANY;
```

**ALTER** → Alterar valores...

9

### Schema

*Depende da escalabilidade,  
→ Para os exercícios dos práticos  
devemos usar SCHEMA*

- Schema é um “namespace” que agrupa tabelas e outros elementos pertencentes à mesma aplicação.
- Criar um Schema

```
CREATE SCHEMA schemaname [AUTHORIZATION username];
```

```
CREATE SCHEMA COMPANY AUTHORIZATION 'CCosta';
```

- Eliminar um Schema

```
DROP SCHEMA schemaname;
```

```
DROP SCHEMA COMPANY;
```

Só existe uma base de dados, mas podem haver vários schemas!„

10

MySQL - sinónimo de “CREATE DATABASE” !

10

 deti

## SQL - Tipo de Dados

- Tipos de dados básicos:
  - Numbers
  - Characters, strings
  - Date e time
  - Binary objects

Standart

Utilizando uma tabela de codificação ele guarda a informação sem juízo de valor (podemos ter a mesma interpretação)

- Os tipos de dados podem variar de acordo com o SGDB!
- Recomendação: Utilizar, na medida do possível, tipos de dados compatíveis com o standard.
  - Aumenta a portabilidade da solução...

11

11

 deti

## SQL - Tipos de dados (SQL:1999)

- Numeric
  - NUMERIC(p,s) e.g. 300.00
  - DECIMAL(p,s)
  - INTEGER (alias: INT) e.g. 32767
  - SMALLINT small integers
  - FLOAT(p) e.g. -1E+03
  - REAL (for short floats) DOUBLE (for long floats)
- String
  - CHARACTER(n) (fixed length)
  - CHARACTER (variable lenght)
  - CHARACTER VARYING(n) (alias: VARCHAR(n))
  - CLOB (Character Large Object, e.g., for large text)
- Date
  - DATE e.g. '1993-01-02'
  - TIME e.g. '13:14:15'
  - TIMESTAMP e.g. '1993-01-02 13:14:15.000001'
- Binary
  - BIT[(n)] e.g. B'01000100'
  - BLOB[(n)] e.g. X'49FE' (Binary Large Objects, e.g., for multimedia)
- Boolean
  - Boolean

Listagem não exaustiva...

12

12

## SQL - Tipo de Dados

Alguns mais utilizados...

- **char(n)**
  - cadeia de caracteres de tamanho fixo n
- **varchar(n)**
  - cadeia de caracteres com tamanho máximo n
- **int**
  - números inteiros (4 bytes)
- **numeric(precisão, escala)**
  - números reais “sem limite” de tamanho
- **date e time**
  - data e hora
- **boolean\***
  - valores booleanos

Não precisamos  
mais do que  
isto!!!  
(\* Para os auto...)

13

13

## SQL Server - Tipos de Dados

Numeric Data Types

Data Type	Description	Length
<u>int</u>	Stores integer values ranging from -2,147,483,648 to 2,147,483,647	4 bytes
<u>tinyint</u>	Stores integer values ranging from 0 to 255	1 byte
<u>smallint</u>	Stores integer values ranging from -32,768 to 32,767	2 bytes
<u>bigint</u>	Stores integer values ranging from -253 to 253-1	8 bytes
<u>money</u>	Stores monetary values ranging from -922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
<u>smallmoney</u>	Stores monetary values ranging from -214,748.3648 to 214,748.3647	4 bytes
<u>decimal(p,s)</u>	Stores decimal values of precision p and scale s. The maximum precision is 38 digits	5-17 bytes
<u>numeric(p,s)</u>	Functionally equivalent to decimal	5-17 bytes
<u>float(n)</u>	Stores floating point values with precision of 7 digits (when n=24) or 15 digits (when n=53)	4 bytes (when n=24) or 8 bytes (when n=53)
<u>real</u>	Functionally equivalent to float(24)	4 bytes

14

14

**SQL Server- Tipos de Dados (cont.)**

Character String Data Types

Data Type	Description	Length
<u>char(n)</u>	Stores $n$ characters	$n$ bytes (where $n$ is in the range of 1–8,000)
<u>nchar(n)</u>	Stores $n$ <u>Unicode</u> characters	$2n$ bytes (where $n$ is in the range of 1–4,000)
<u>varchar(n)</u>	Stores approximately $n$ characters	Actual string length +2 bytes (where $n$ is in the range of 1–8,000)
<u>varchar(max)</u>	Stores up to $2^{31}-1$ characters	Actual string length +2 bytes
<u>nvarchar(n)</u>	Stores approximately $n$ characters	$2n$ (actual string length) +2 bytes (where $n$ is in the range of 1–4,000)
<u>nvarchar(max)</u>	Stores up to $((2^{31}-1)/2)-2$ characters	$2n$ (actual string length) +2 bytes

Binary Data Types

Data Type	Description	Length
<u>bit</u>	Stores a single bit of data	1 byte per 8 bit columns in a table
<u>binary(n)</u>	Stores $n$ bytes of binary data	$n$ bytes (where $n$ is in the range of 1–8,000)
<u>varbinary(n)</u>	Stores approximately $n$ bytes of binary data	Actual length +2 bytes (where $n$ is in the range of 1–8,000)
<u>varbinary(max)</u>	Stores up to $2^{31}-1$ bytes of binary data	Actual length +2 bytes

O mêsse "boolean" →

15

15

**SQL Server- Tipos de Dados (cont.)**

Date and Time Data Types

Data Type	Description	Length	Example
<u>date</u>	Stores dates between January 1, 0001, and December 31, 9999	3 bytes	2008-01-15
<u>datetime</u>	Stores dates and times between January 1, 1753, and December 31, 9999, with an accuracy of 3.33 milliseconds	8 bytes	2008-01-15 09:42:16.142
<u>datetime2</u>	Stores date and times between January 1, 0001, and December 31, 9999, with an accuracy of 100 nanoseconds	6–8 bytes	2008-01-15 09:42:16.1420221
<u>datetimeoffset</u>	Stores date and times with the same precision as datetime2 and also includes an offset from Universal Time Coordinated (UTC) (also known as Greenwich Mean Time)	8–10 bytes	2008-01-15 09:42:16.1420221 +05:00
<u>smalldatetime</u>	Stores dates and times between January 1, 1900, and June 6, 2079, with an accuracy of 1 minute (the seconds are always listed as "00")	4 bytes	2008-01-15 09:42:00
<u>time</u>	Stores times with an accuracy of 100 nanoseconds	3–5 bytes	09:42:16.1420221

Listagem não exaustiva. Há outros tipo como o cursor, sql\_variant, table, xml, ...

16

16

 deti

## SQL - Definição de Domínio

- O comando `create domain` permite definir novos tipos de dados.
- Um domain pode conter um valor de defeito (default) e restrições do tipo not null e check.

```
CREATE DOMAIN domainname
```

**Criação...**

```
CREATE DOMAIN compsalary INTEGER
    NOT NULL CHECK (compsalary > 475);
```

**Utilização...**

```
CREATE TABLE EMPLOYEE (
    ...
    Salary          compsalary,
    ...);
```

*Se inicialmente definirmos restrições, vai aparecer em vários tabelas (e.g., com INT maior que 750), criando um domínio (um gênero de "constante")*

Nota: Não disponível em SQL SERVER. 

17

17

 deti

## SQL - Definição de Novo Tipo

- Como alternativa ao domain, podemos criar só um novo tipo (alias) com o comando `create type`.

```
CREATE Type... em SQL SERVER
```

**Criação...**

```
CREATE TYPE SSN FROM varchar(9) NOT NULL;
```

*Não permite checks*

**Utilização...**

```
CREATE TABLE EMPLOYEE (
    ...
    Ssn           SSN,
    ...);
```

- Nota: Em geral, é mais limitado que o create domain.

18

18

 deti

## DDL - Criar uma Tabela

```
CREATE TABLE tbname ( A1 D1, A2 D2, ..., An Dn,
                      (integrity-constraint1),
                      ...
                      (integrity-constraintK) );
tbname - nome da relação (tabela)
```

CREATE TABLE COMPANY.EMPLOYEE (...)  
CREATE TABLE EMPLOYEE (...)

COMPANY - nome do schema

A1 D1, A2 D2, ..., An Dn  
A1...An - Atributos da relação  
D1...Dn - Domínio dos atributos

Restrições de Integridade  
integrity-constraint1,  
...,  
integrity-constraintN

19

 deti

## Criar uma Tabela (exemplo)

```
CREATE TABLE...
definindo atributos e respectivo domínio.
```

```
CREATE TABLE EMPLOYEE (
    Fname          VARCHAR(15),
    Minit          CHAR,
    Lname          VARCHAR(15),
    Ssn            CHAR(9),
    Bdate          DATE,
    Address        VARCHAR(30),
    Sex             CHAR,
    Salary          DECIMAL(10,2),
    Super_ssn      CHAR(9),
    Dno             INT);
```

20

20

**Atributos - Valores por Omissão**

- Podem ser definidos valores por omissão para cada coluna
  - utilizando o termo “default” 

```
CREATE com default ...
```

```
CREATE TABLE EMPLOYEE (
    Fname           VARCHAR(15),
    ...
    Salary          DECIMAL(10,2)      DEFAULT 0,
    ...
    Dno             INT;            No caso de não inserirmos fica NULL,
```

*Lógica do negócio*

21

21

**Restrições de Integridade**

- check (P)
  - impõr uma regra a um atributo
- not null
  - atributo não pode ser null
- primary key (A<sub>1</sub>, ..., A<sub>n</sub>)
  - definir chave primária
- unique (A<sub>1</sub>, ..., A<sub>n</sub>)
  - chaves candidatas não primárias
- foreign key
  - definir chave estrangeira

As restrições podem ser de:

- coluna - referem-se a apenas uma coluna e são descritas em frente à coluna
- tabela - referem-se a mais do que a uma coluna e ficam separadas da definição das colunas

*Lógica do negócio*

22

22

Foreign Keys  
Primary Keys  
Unique

 deti

## Restrição CHECK

**Restrição CHECK na coluna...**

```
CREATE TABLE EMPLOYEE (
    ...
    Salary           DECIMAL(10,2)      CHECK (Salary > 12),
    ...
)
```

**Restrição CHECK na tabela...**

```
CREATE TABLE DEPARTMENT (
    ...
    Dept_create_date   DATE            NOT NULL,
    Mgr_start_date     DATE,
    ...
    CHECK (Dept_create_date <= Mgr_start_date);
```

*Restrição* → Deveremos por tudo no fim...

**Restrição aplicada a cada atributo referenciado sempre que um tuplo é introduzido ou modificado.**

23

23

 deti

## Restrição PRIMARY KEY

- Só podemos definir uma chave primária na tabela.
  - Por definição, a chave primária não pode conter valores repetidos ou nulos.

**Restrição PRIMARY KEY na coluna...**

```
CREATE TABLE EMPLOYEE (
    ...
    Ssn             CHAR(9)          PRIMARY KEY,
    ...
)
```

**Restrição PRIMARY KEY na tabela...**  
(obrigatório se PK for composta por mais do que um atributo)

```
CREATE TABLE EMPLOYEE (
    ...
    Ssn             CHAR(9),
    ...
    PRIMARY KEY (Ssn));
```

*↑ Mete SEMPRE no fim!*

24

## Restrição UNIQUE

• Utilizada para as chaves candidatas alternativas.

- Não pode conter valores repetidos mas pode ter valores null.

**Restrição UNIQUE na coluna...**

```
CREATE TABLE DEPARTMENT (
    Dname          VARCHAR(15)  UNIQUE NOT NULL, Assim garantimos o mês nulo!
    Dnumber        INT          NOT NULL,
    PRIMARY KEY (Dnumber),
    ... );
```

**Restrição UNIQUE na tabela...**

```
CREATE TABLE DEPARTMENT (
    Dname          VARCHAR(15)      NOT NULL,
    Dnumber        INT              NOT NULL,
    PRIMARY KEY (Dnumber),
    UNIQUE (Dname), ... );
```

25

## Restrição FOREIGN KEY

• Utilizada para declarar chaves estrangeiras.

• Uma chave estrangeira deve referenciar uma chave primária ou única. *Qualquer que seja a candidata a chave primária!»*

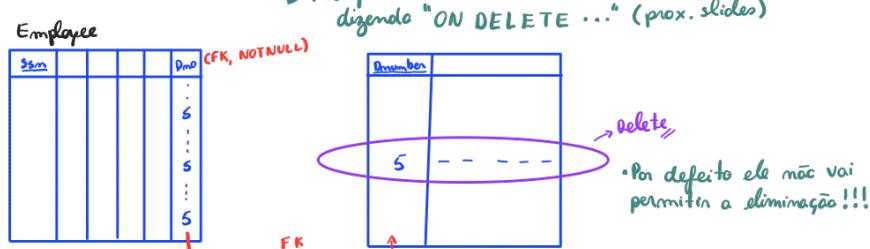
**Restrição FOREIGN KEY na coluna...**

```
CREATE TABLE EMPLOYEE (
    ...
    Super_ssn   CHAR(9)   REFERENCES EMPLOYEE(Ssn),
    Dno         INT       REFERENCES DEPARTMENT(Dnumber) NOT NULL,
    ...);
```

**Restrição FOREIGN KEY na tabela...**

```
CREATE TABLE EMPLOYEE (
    ...
    Ssn         CHAR(9),
    Dno         INT       NOT NULL,
    ...
    FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
    FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber));
```

26



13

 deti

## Restrição FOREIGN KEY

Integridade Referencial

- Pode haver uma violação quando são inseridos ou eliminados tuplos ou quando os atributos chave estrangeira ou primária são modificados, resultando numa rejeição da operação.
- Podemos definir as seguintes ações alternativas: “on delete” e “on update”, com as seguintes opções:
  - restrict - não deixa efetuar a operação
  - cascade - apaga os registos associados (delete) ou altera a chave estrangeira (update)
  - set null - a chave estrangeira passa a null.
  - set default - a chave estrangeira passa a ter o valor por omissão.

27



 deti

## Restrição FOREIGN KEY

Integridade Referencial

**Restrição FOREIGN KEY**

```
CREATE TABLE EMPLOYEE (
    ...
    Ssn          CHAR(9),
    Dno          INT              NOT NULL,
    ...
    FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
        ON DELETE SET NULL  ON UPDATE CASCADE,  

    FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
        ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

Se o tuplo do supervisor é eliminado, a coluna Super\_ssn dos supervisionados passa automaticamente a Null.

Se o Ssn do supervisor é atualizado, a coluna Super\_ssn dos supervisionados é atualizada em cascata.

Aqui o CASCADE vai alterar o valor apenas!,,

Vai mudar todos em cascata!

Pergunta de Teste!,,

28

## Restrições - atribuição de nome

- Imaginando que queremos alterar uma restrição de uma tabela... Como referenciá-la?
- Nestas situações temos de “baptizar” a restrição com um nome próprio.

### Restrições com nome...

```
CREATE TABLE EMPLOYEE (
    ...
    ...
    CONSTRAINT EMPPK
        PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
            ON DELETE SET NULL    ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
            ON DELETE SET DEFAULT ON UPDATE CASCADE);
,
```

*baptizado...*

*(posso fazer drop das restrições)*

29

## Tabela - Drop

- O comando **drop table** remove da base de dados toda a informação sobre a tabela e os dados (tuplos).

### Eliminar a tabela EMPLOYEE

```
DROP TABLE EMPLOYEE;
```

- Caso haja violação de restrições de integridade referencial, a operação é rejeitada.
- No entanto, a opção **CASCADE\*** permite eliminar a tabela e os elementos referenciados na restrição.

### Eliminar a tabela EMPLOYEE com opção CASCADE

```
DROP TABLE EMPLOYEE CASCADE; ⇒ Eliminam todos os "dependências"
```

\* Não está disponível em SQL Server. Solução: eliminar primeiro o constraint.

30

30

**Tabela - Alter**

- O comando `alter table` é utilizado para modificar o esquema da tabela ou restrições existentes.
- Adicionar atributos à tabela:

```
ALTER TABLE tablename ADD Attribute Domain
```

```
ALTER TABLE EMPLOYEE ADD nofiscal INT; → Alterar: Adicionar // Remover (++)
```

- Todos os tuplos existentes ficam com valor null no novo atributo.

- Adicionar restrições à tabela:

```
ALTER TABLE tablename ADD CONSTRAINT name theconstraint
```

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT salarymin CHECK (Salary >475);
```

↑ Se já existir dê erro!

31

**Tabela - Alter**

- Eliminar atributos da tabela:

```
ALTER TABLE tablename DROP COLUMN attributename
```

```
ALTER TABLE EMPLOYEE DROP COLUMN nofiscal;
```

↑ Eliminar atributo

- Eliminar restrições da tabela:

```
ALTER TABLE tablename DROP CONSTRAINT name
```

```
ALTER TABLE EMPLOYEE DROP CONSTRAINT salarymin;
```

↑ Eliminar restrição

- Alterar um atributo de uma tabela:

```
ALTER TABLE tablename ALTER Attribute Domain
```

```
ALTER TABLE EMPLOYEE ALTER COLUMN noFiscal CHAR(9);
```

↑ "Lá dentro"  
↑ Se não existir dê erro!

32

deti

## SQL DDL - Caso de Estudo

Empresa

33

33

deti

## Esquema Relacional da BD da Empresa

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

**DEPT\_LOCATIONS**

Dnumber	Dlocation
---------	-----------

**PROJECT**

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

**WORKS\_ON**

Essn	Pno	Hours
------	-----	-------

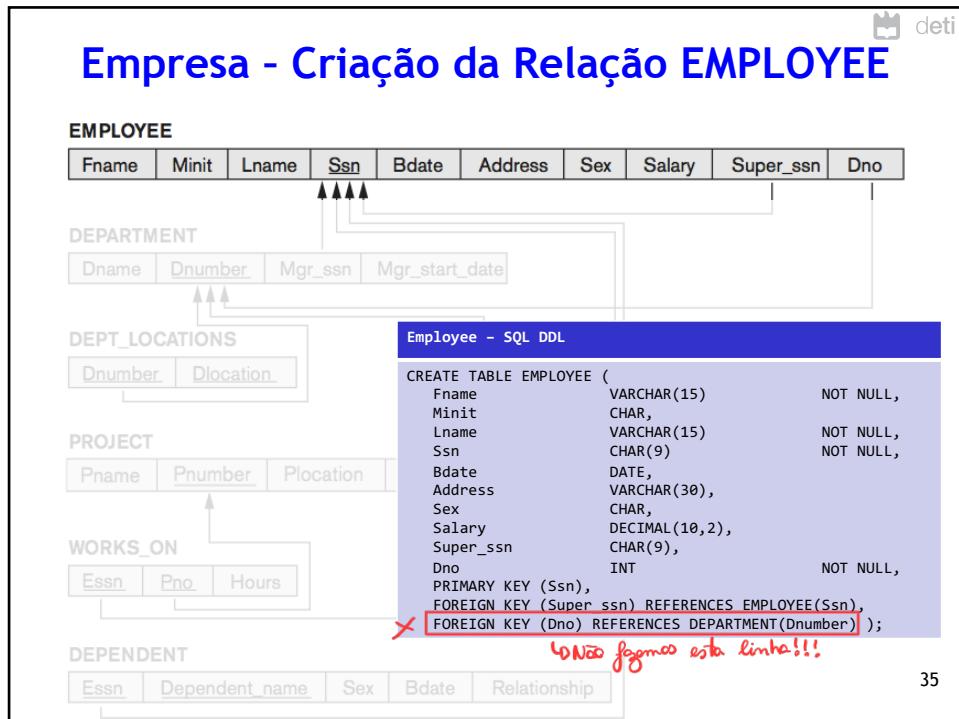
**DEPENDENT**

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

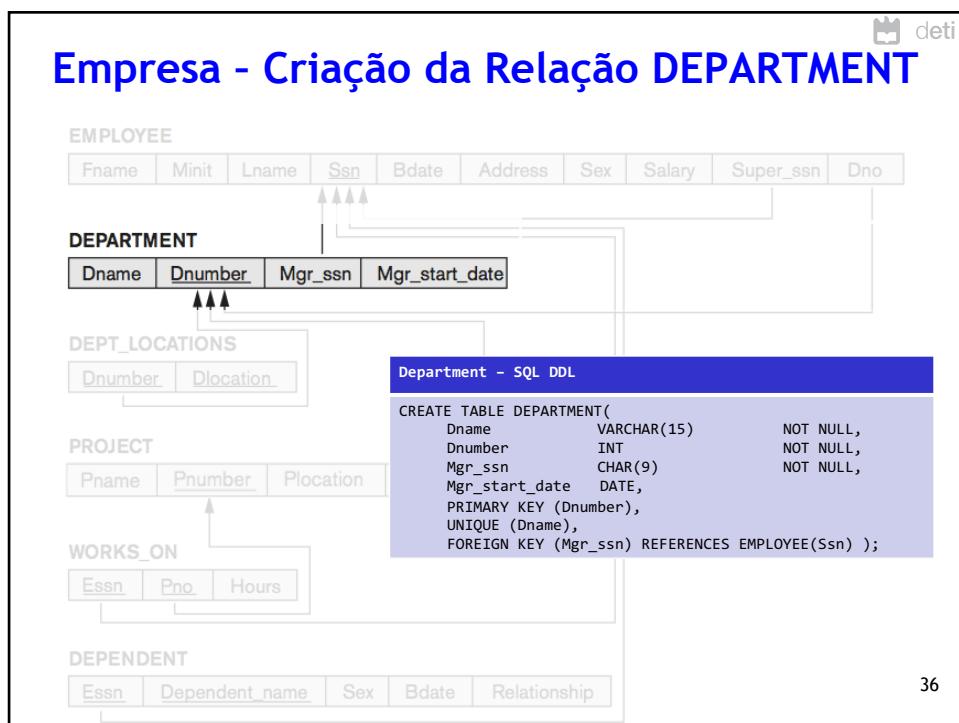
- Primeiro devemos criar os tabelas... (CREATE)
- Depois alteramos (ALTER)
  - (...)
  - Corro em cada uma precisamos de por atributos de outros (...)

34

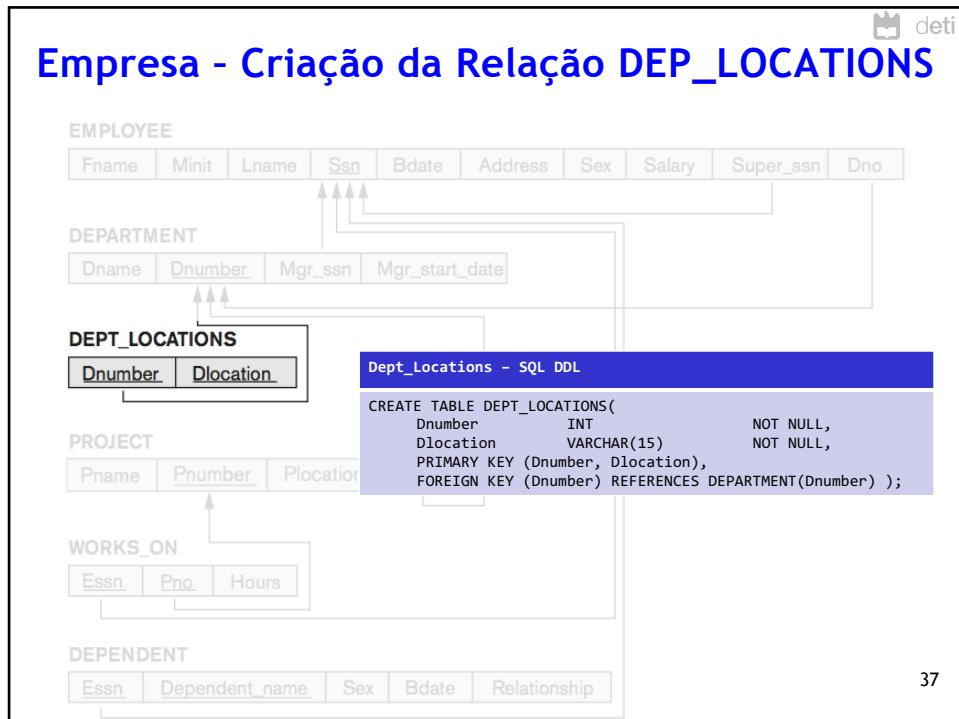
34



35



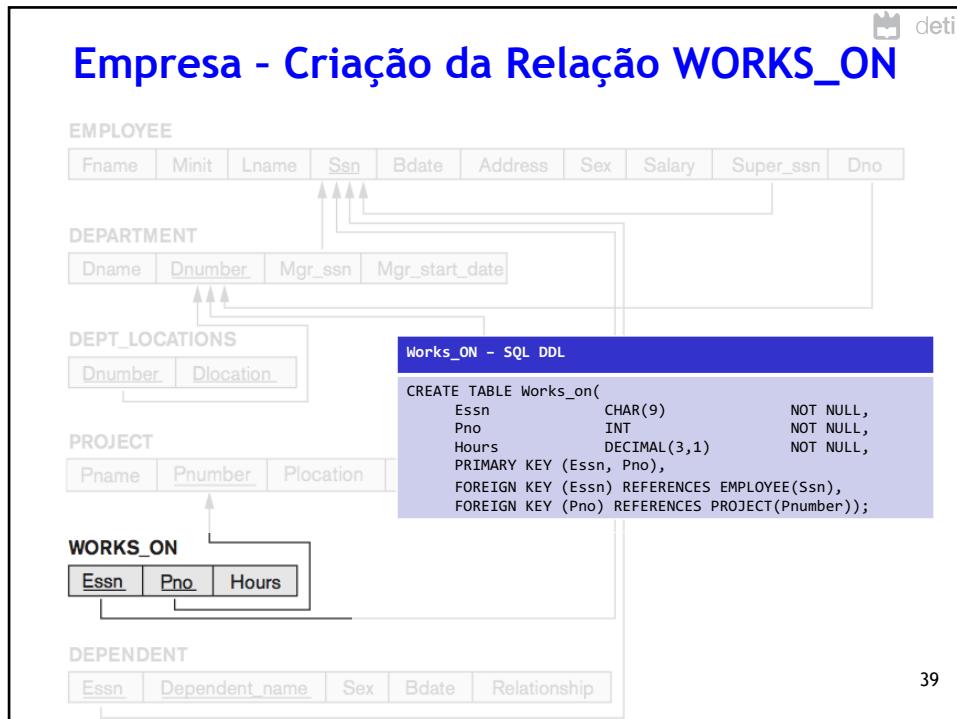
36



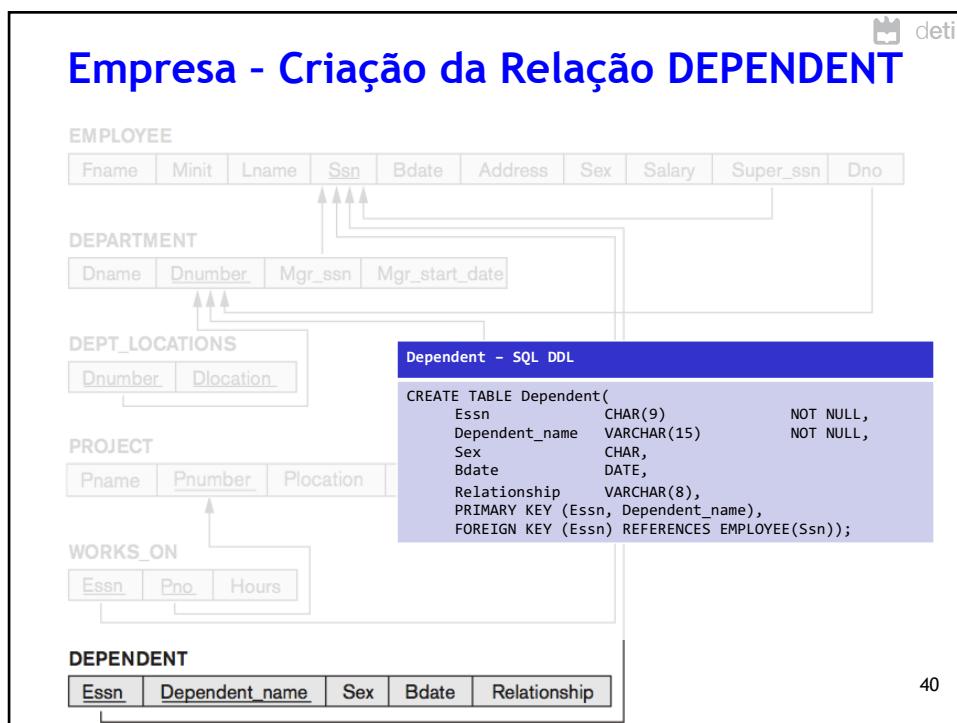
37



38



39



40

**deti**

## Empresa DDL - Considerações Práticas

**EXEMPLO: Employee, Department and Foreign Keys**

```

CREATE TABLE EMPLOYEE (
    Ssn           CHAR(9)          NOT NULL,
    Super_ssn    CHAR(9),
    Dno          INT              NOT NULL,
    ...
    PRIMARY KEY (Ssn),
    FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn));

CREATE TABLE DEPARTMENT(
    Dnumber       INT              NOT NULL,
    ...
    PRIMARY KEY (Dnumber),
    ...);

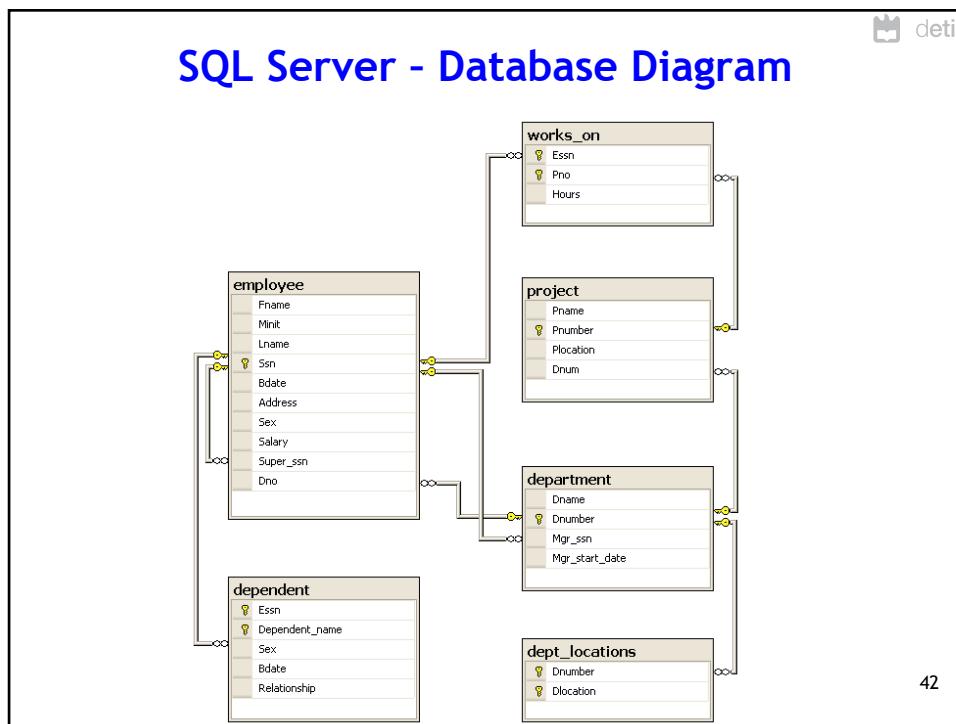
ALTER TABLE EMPLOYEE
    ADD CONSTRAINT EMPDEPTFK FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);
ALTER TABLE DEPARTMENT
    ADD CONSTRAINT DEPTMGRFK FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn);
  
```

- Na prática só podemos criar restrições de integridade referencial, com recurso a chaves estrangeiras, quando temos as duas relações criadas.
- Assim, devemos começar por criar cada uma das relações (tabelas) e só depois definir as restrições.
  - ou pelo menos uma delas...

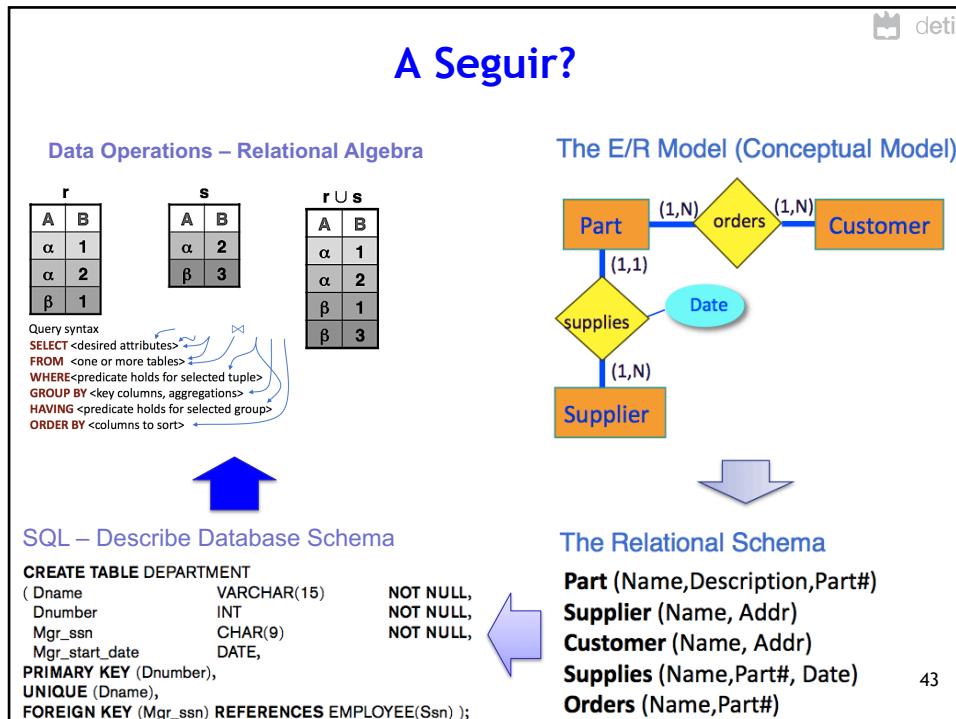
41

41

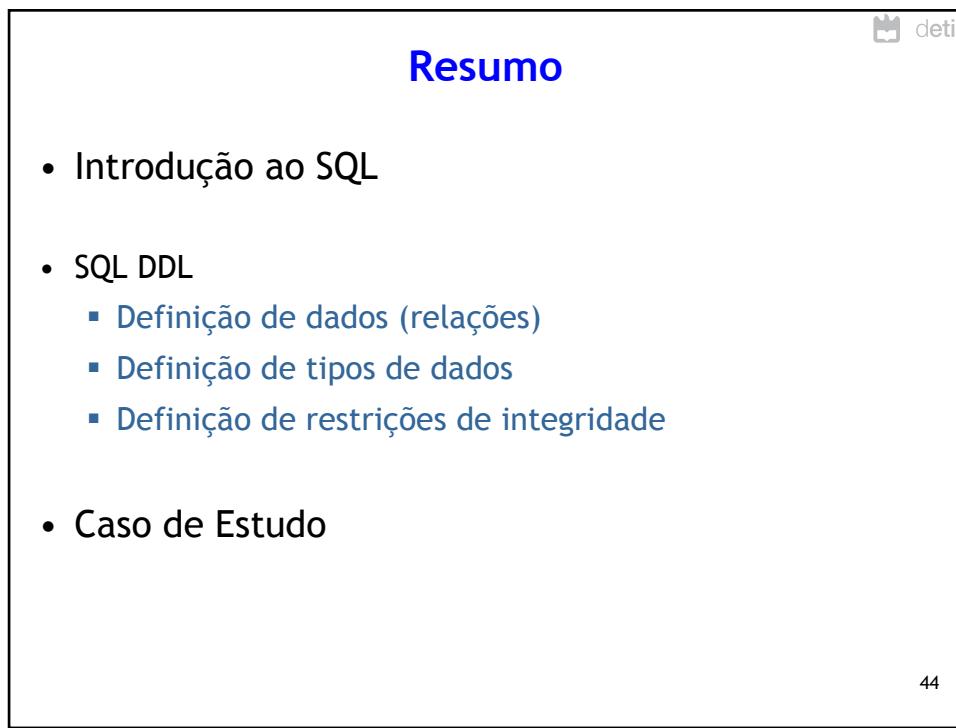
→ Podem também corrigir este problema...



42



43



44

44