

SQL

Conceitos básicos

- **Entidades:** entidade é uma “coisa” do mundo real, com uma existência independente.
- **Atributos** são as propriedades dessa coisa.
- Cada atributo está associado a um determinado **tipo**.

Funcionario

SIAPE	Nome	Lotacao	Cargo	Salario	Admissao
--------------	-------------	----------------	--------------	----------------	-----------------

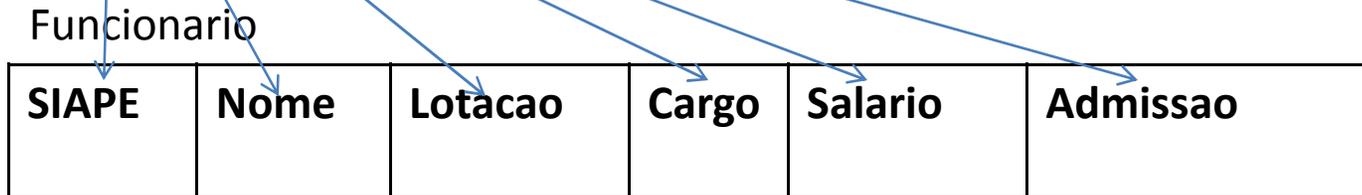
Conceitos básicos

- **Entidades:** entidade é uma “coisa” do mundo real, com uma existência independente.
- **Atributos** são as propriedades dessa coisa.
- Cada atributo está associado a um determinado **tipo**.

Funcionario					
SIAPE	Nome	Lotacao	Cargo	Salario	Admissao

Conceitos básicos

- **Entidades:** entidade é uma “coisa” do mundo real, com uma existência independente.
- **Atributos** são as propriedades dessa coisa.
- Cada atributo está associado a um determinado **tipo**.



Conceitos básicos

- **Entidades:** entidade é uma “coisa” do mundo real, com uma existência independente.
- **Atributos** são as propriedades dessa coisa.
- Cada atributo está associado a um determinado **tipo**.

Funcionario

SIAPE	Nome	Lotacao	Cargo	Salario	Admissao
↑	↑	↑	↑	↑	↑
Texto	Texto	{DPI,DSR,...}	Texto	Real	Data

DataTypes

- Depend on DBMS. Exs do Postgresql:

Table 8-2. Numeric Types

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to 9223372036854775807
decimal	variable	user-specified precision, exact	no limit
numeric	variable	user-specified precision, exact	no limit
real	4 bytes	variable-precision, inexact	6 decimal digits precision
double precision	8 bytes	variable-precision, inexact	15 decimal digits precision
serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

DataTypes

Table 8-4. Character Types

Name	Description
character varying(<i>n</i>), varchar(<i>n</i>)	variable-length with limit
character(<i>n</i>), char(<i>n</i>)	fixed-length, blank padded
text	variable unlimited length

Table 8-9. Date/Time Types

Name	Storage Size	Description	Low Value	High Value	Resolution
timestamp [(p)] [without time zone]	8 bytes	both date and time (no time zone)	4713 BC	294276 AD	1 microsecond / 14 digits
timestamp [(p)] with time zone	8 bytes	both date and time, with time zone	4713 BC	294276 AD	1 microsecond / 14 digits
date	4 bytes	date (no time of day)	4713 BC	5874897 AD	1 day
time [(p)] [without time zone]	8 bytes	time of day (no date)	00:00:00	24:00:00	1 microsecond / 14 digits
time [(p)] with time zone	12 bytes	times of day only, with time zone	00:00:00+1459	24:00:00-1459	1 microsecond / 14 digits
interval [fields] [(p)]	12 bytes	time interval	-178000000 years	178000000 years	1 microsecond / 14 digits

Conceitos básicos

- Esquema x Instância

Funcionario

SIAPE	Nome	Lotacao	Cargo	Salario	Admissao
1258549	Lubia Vinhas	DPI	Pesq.	xxxxx.xx	01/12/1997
1058	Fulano de tal	LAC	Tec.	yyyyy.yy	08/10/2000

Conceitos básicos

- Restrições

Funcionario

<u>SIAPE</u>	Nome	Lotacao	Cargo	Salario	Admissao
--------------	------	---------	-------	---------	----------



Chave Primária: não admite mesmo valor para duas instâncias diferentes.
Não podem ter valores nulos.

Check constraints: estendem a limitação dos valores permitidos para cada atributo:

NOT NULL: mais básica

Outras restrições podem ser construídas (ex. SALARIO > 0)

UNIQUE: aplicável a 1 ou mais colunas a fim de garantir valores únicos para ela(s) na tabela inteira.

Introdução a SQL

- Criação de um banco

```
CREATE DATABASE my_db
```

- Criação de uma tabela

```
CREATE TABLE table_name  
(  
  column_name1 data_type,  
  column_name2 data_type,  
  column_name3 data_type,  
  ....  
)
```

Introdução a SQL

- Restrições podem ser inseridas no comando de criação da tabela ou depois com o comando ALTER TABLE

```
CREATE TABLE table_name  
(  
  column_name1 data_type restriction,  
  column_name2 data_type restriction,  
  column_name3 data_type restriction,  
  ....  
)
```

Introdução a SQL

- Criar a tabela funcionários: usar a GUI mas observar a DDL criada

Funcionario

<u>SIAPE</u>	Nome	Lotacao	Cargo	Salario	Admissao
↑	↑	↑	↑	↑	↑
Inteiro	Texto	{DPI,DSR,...}	Texto	Real	Data

INSERT

```
INSERT INTO table_name  
VALUES (value1, value2, value3,...)
```

Ou

```
INSERT INTO table_name (column1, column2, column3,...)  
VALUES (value1, value2, value3,...)
```

Inserir pelo menos 5 instâncias, tentar disparar as restrições de integridade

SELECT – Seleção + Projeção

```
SELECT column_name(s)
```

```
FROM table_name
```

Ou

```
SELECT * FROM table_name
```

Selecionar empregados

Selecionar o nome dos empregados

SELECT – DISTINCT e ORDER BY

```
SELECT DISTINCT column_name(s) FROM table_name
```

Selecionar quais os diferentes departamentos

```
SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC
```

Selecionar todos os empregados por ordem decrescente de salário

SELECT – WHERE

`SELECT column_name(s) FROM table_name WHERE condição`

Ou

`SELECT * FROM table_name WHERE condição`

Selecionar empregados com salário maior que 10000

Selecionar o nome dos empregados da DPI

Selecionar o nome dos empregados da DPI com salário maior que 1000

UPDATE

UPDATE table_name

SET column1=value, column2=value2,...

WHERE some_column=some_value

Dar um aumento de 10% no salário de todos os empregados;

Dar um aumento extra de mais 10% só para os empregados da DPI!

DELETE

`DELETE FROM table_name WHERE some_column=some_value`

“Demita” todos os funcionários cujos nomes comecem com “L”

ALIAS

```
SELECT column_name(s)  
FROM table_name  
AS alias_name
```

Exemplo:

```
SELECT nome FROM funcionario AS funcname
```

Relacionamentos

- Para estudar relacionamentos é preciso criar uma segunda tabela. Crie conforme abaixo:

Departamento

<u>ID</u> (texto)	Nome (texto)	Chefe (Inteiro)	Descricao (texto)
----------------------	-----------------	--------------------	----------------------

Relacionamentos

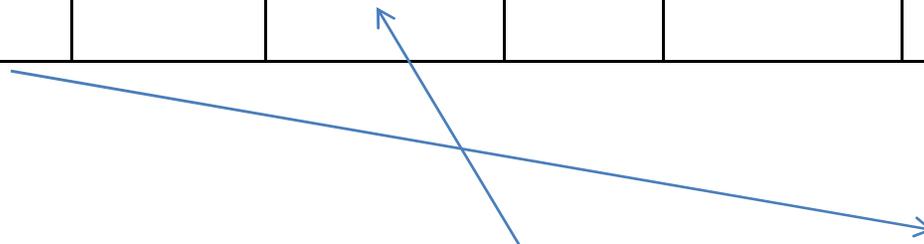
- Use o conceito de chave estrangeira para garantir o relacionamento entre as duas entidades
- Popule a tabela Departamento com pelo menos 3 departamentos: DSR, DPI e LAC
- Revise a tabela de funcionarios e insira mais alguns

Funcionario

<u>SIAPE</u>	Nome	Lotacao	Cargo	Salario	Admissao
--------------	------	---------	-------	---------	----------

Departamento

<u>ID</u>	Nome	Chefe	Descricao
-----------	------	-------	-----------



Foreign Key

```
ALTER TABLE departamento ADD FOREIGN KEY (chefe) REFERENCES funcionario (siape)
```

```
ALTER TABLE departamento  
  ADD CONSTRAINT fk_chefe  
  FOREIGN KEY (chefe)  
  REFERENCES funcionario (siape)
```

```
ALTER TABLE departamento  
  DROP FOREIGN KEY fk_PerOrders
```

JOIN

```
SELECT column_name(s)  
  FROM table_name1  
  INNER|LEFT|RIGHT JOIN table_name2  
  ON table_name1.column_name=table_name2.column_name
```

Mostre os nomes dos chefes de cada departamento

Mostre os funcionarios que não são chefes

Funções

Funções de agregação: retornam um único valor calculado a partir dos valores de uma coluna:

- AVG()
- COUNT()
- MAX()
- MIN()
- SUM()

Funções escalares retornam um valor simples baseado no valor previamente existente

- UCASE()
- LCASE()
- LEN()
- ROUND()
- NOW()

Funções

Descubra a média de salário por departamento

Descubra o nome do chefe cujo salário é mais alto

Descubra quantos funcionários entraram em 1997