

SQL

EDUARDO C. DE ALMEIDA
DEPARTAMENTO DE INFORMÁTICA



BANCOS DE DADOS
CI1218

AGENDA

- ▶ Introdução
- ▶ Declaração SELECT
- ▶ Variáveis de tabelas e operadores de conjunto

Conexão: `psql -U ci218 -h bd.c3s1.ufpr.br -d tpch`

HISTORICO

- ▶ Criada início dos anos 1970 pela IBM (“SEQUEL”)
 - Structured English Query Language
- ▶ Adoção comercial
 - Oracle e IBM final dos anos 1970
- ▶ Padrão ANSI SQL em 1986
 - Atual 2016
 - Padrão mínimo suportado 1992

LINGUAGENS

- ▶ Data Manipulation Language (DML)
 - SELECT, INSERT, UPDATE, DELETE
- ▶ Data Definition Language (DDL)
 - CREATE, DROP
- ▶ Data Control Language (DCL)
 - GRANT, REVOKE
- ▶ Outras
 - indices, views, triggers, constraints, transactions ...

MODELO REFERENCIA

Autores(autor, email, idade)

<u>autor</u>	email	idade
JK. Rowling	jk@gmail	20
H. Melville	M@yahoo	69
F. Sabino	S@ig	21
J. Widom	jw@gmail	NULL

Livros(isbn, titulo, editora)

<u>isbn</u>	titulo	editora
1234	Harry Potter 1	Pottermore
1121	Moby Dick	DCL
2222	Grande Mentecapto	Nordica
2223	Inglesa Deslumbrada	Nordica
2224	De Ponta Cabeça	Nordica
2225	Encontra Mercado	Nordica

AutorLivros(autor, isbn)

<u>autor</u>	<u>isbn</u>
JK. Rowling	1234
H. Melville	1121
F. Sabino	2222
F. Sabino	2223
F. Sabino	2224
F. Sabino	2225

SELECT

AutorLivros(autor, isbn)

```
SELECT [DISTINCT] A1, A2, ..., AN
FROM R1, R2, ..., RN
WHERE <CONDIÇÃO>;
```

<u>autor</u>	<u>isbn</u>
JK. Rowling	1234
H. Melville	1121
F. Sabino	2222
F. Sabino	2223
F. Sabino	2224
F. Sabino	2225

$\sigma_{isbn>2000}(autorlivro)$

```
[tpch=# select * from autorlivros where isbn > 2000;
 autor | isbn
-----+-----
 F. Sabino | 2222
 F. Sabino | 2223
 F. Sabino | 2224
 F. Sabino | 2225
(4 rows)
```

$\pi_{autor}(\sigma_{isbn>2000}(autorlivro))$

```
[tpch=# select autor from autorlivros where isbn > 2000;
 autor
-----
 F. Sabino
 F. Sabino
 F. Sabino
 F. Sabino
(4 rows)
```

SELECT

AutorLivros(autor, isbn)

```
SELECT [DISTINCT] A1, A2, ..., AN  
FROM R1, R2, ..., RN  
WHERE <CONDIÇÃO>;
```

<u>autor</u>	<u>isbn</u>
JK. Rowling	1234
H. Melville	1121
F. Sabino	2222
F. Sabino	2223
F. Sabino	2224
F. Sabino	2225

$\sigma_{isbn>2000}(autorlivro)$

```
ltpch=# select * from autorlivros where isbn > 2000;  
 autor | isbn  
-----+-----  
 F. Sabino | 2222  
 F. Sabino | 2223  
 F. Sabino | 2224  
 F. Sabino | 2225  
(4 rows)
```

$\pi_{autor}(\sigma_{isbn>2000}(autorlivro))$

```
ltpch=# select autor from autorlivros where isbn > 2000;  
 autor  
-----  
 F. Sabino  
 F. Sabino  
 F. Sabino  
 F. Sabino  
(4 rows)
```

SELECT não retira duplicatas

SELECT

AutorLivros(autor, isbn)

```
SELECT [DISTINCT] A1, A2, ..., AN  
FROM R1, R2, ..., RN  
WHERE <CONDIÇÃO>;
```

<u>autor</u>	<u>isbn</u>
JK. Rowling	1234
H. Melville	1121
F. Sabino	2222
F. Sabino	2223
F. Sabino	2224
F. Sabino	2225

$\sigma_{isbn>2000}(autorlivro)$

$\pi_{autor}(\sigma_{isbn>2000}(autorlivro))$

```
[tpch=# select distinct autor from autorlivros where isbn > 2000;  
autor  
-----  
F. Sabino  
(1 row)
```

DISTINCT para retirar duplicatas

CLÁUSULA SELECT

PROJETA OS ATRIBUTOS DE INTERESSE

```
[tpch=> select * from autores;
```

autor	email	idade
JK. Rowling	jk@gmail	20
H. Melville	m@yahoo	69
F. Sabino	s@ig	21
J. Widom	jw@gmail	

(4 rows)

```
tpch=> select * from autores;
```

autor	email	idade
JK. Rowling	jk@gmail	20
H. Melville	m@yahoo	69
F. Sabino	s@ig	21
J. Widom	jw@gmail	

(4 rows)

```
[tpch=> select autores.* from autores;
```

autor	email	idade
JK. Rowling	jk@gmail	20
H. Melville	m@yahoo	69
F. Sabino	s@ig	21
J. Widom	jw@gmail	

(4 rows)

“*” retorna todos os atributos.
SELECT permite expressões e aliases

```
SELECT *  
FROM AUTORES;
```

```
SELECT AUTORES.*  
FROM AUTORES;
```

```
SELECT IDADE*12 AS "MESES"  
FROM AUTORES;
```

- ▶ EXPRESSOES: +, -, /, *
- ▶ CONCATENAÇÃO: ||

CLÁUSULA FROM

```
tpch=> select a.autor, b.isbn
tpch-> from autores A, autorlivros b
[tpch-> where a.autor=b.autor;
 autor | isbn
-----+-----
JK. Rowling | 1234
H. Melville | 1121
F. Sabino | 2222
F. Sabino | 2223
F. Sabino | 2224
F. Sabino | 2225
(6 rows)

tpch=> select a.autor, b.isbn
[tpch-> from autores A left outer join autorlivros b on a.autor=b.autor;
 autor | isbn
-----+-----
JK. Rowling | 1234
H. Melville | 1121
F. Sabino | 2222
F. Sabino | 2223
F. Sabino | 2224
F. Sabino | 2225
(6 rows)
```

define as tabelas de acesso (permite alias)

```
SELECT *
FROM AUTORES A;
```

faz o bind de variáveis e registros

```
SELECT A.AUTOR, B.ISBN
FROM AUTORES A, AUTORLIVROS B
WHERE A.AUTOR=B.AUTOR;
```

define as junções entre tabelas

```
SELECT A.AUTOR, B.ISBN
FROM AUTORES A LEFT OUTER JOIN
AUTORLIVROS B ON A.AUTOR=B.AUTOR;
```

CLÁUSULA WHERE

```
tpch=> select *
from autores
where idade > 25 OR idade IS null;
  autor      | email      | idade
-----+-----+-----
 H. Melville | m@yahoo   |    69
 J. Widom    | jw@gmail  |
(2 rows)

tpch=> select *
from autores
where idade between 18 and 24;
  autor      | email      | idade
-----+-----+-----
 JK. Rowling | jk@gmail  |    20
 F. Sabino   | s@ig      |    21
(2 rows)

tpch=> select *
from autores
where idade < 25 AND autor like 'F%';
  autor      | email      | idade
-----+-----+-----
 F. Sabino   | s@ig      |    21
(1 row)

tpch=> select *
from autorlivros
where isbn in(1234,1121,2222);
  autor      | isbn
-----+-----
 JK. Rowling | 1234
 H. Melville | 1121
 F. Sabino   | 2222
(3 rows)
```

expressões complexas
usando AND, OR, NOT e IS

```
SELECT *
FROM AUTORES A
WHERE IDADE > 25 OR IDADE IS NULL;
```

operadores especiais LIKE,
BETWEEN e IN

```
SELECT *
FROM AUTORLIVROS
WHERE ISBN IN(1234,1121,2222);
```

também define as
junções entre tabelas

```
SELECT A.AUTOR, B.ISBN
FROM AUTORES A, AUTORLIVROS B
WHERE A.AUTOR=B.AUTOR;
```

COMPARAÇÃO: =, <, >, <>, <=, >=

OPERADOR LIKE:

- ▶ % (VARIOS CHARACTERES)
- ▶ _ (UM CHARACTER)

CLÁUSULA ORDER BY

ORDENA OS REGISTROS RETORNADOS PELO PROGRAMA

```
SELECT [DISTINCT] A1, A2, ..., AN  
FROM R1, R2, ..., RN  
WHERE <CONDIÇÃO>  
[ORDER BY A1, A2 ASC/DESC];
```

```
tpch=> select *  
from autores  
[order by autor desc;  
autor | email | idade  
-----+-----+-----  
J. Widom | jw@gmail |  
JK. Rowling | jk@gmail | 20  
H. Melville | m@yahoo | 69  
F. Sabino | s@ig | 21  
(4 rows)
```

```
tpch=> select *  
from autores  
[order by autor desc, email asc;  
autor | email | idade  
-----+-----+-----  
J. Widom | jw@gmail |  
JK. Rowling | jk@gmail | 20  
H. Melville | m@yahoo | 69  
F. Sabino | s@ig | 21  
(4 rows)
```

FUNÇÕES STRING

```
tpch=> select autor, strpos(autor, 'ing')
from autores ;
   autor   | strpos
-----+-----
 JK. Rowling |      9
 H. Melville |      0
 F. Sabino   |      0
 J. Widom    |      0
(4 rows)

tpch=> select upper(autor)
from autores ;
   upper
-----
 JK. ROWLING
 H. MELVILLE
 F. SABINO
 J. WIDOM
(4 rows)

tpch=> select substring(autor, 0, 4)
from autores ;
 substring
-----
 JK.
 H.
 F.
 J.
(4 rows)
```

- ▶ SUBSTRING
- ▶ STRPOS
- ▶ LOWER
- ▶ UPPER
- ▶ TRIM
- ▶ LENGTH
- ▶ MD5
- ▶ ...

FUNÇÕES

- ▶ Data: AGE(), CURRENT_DATE
 - select AGE(CURRENT_DATE, '2010-01-01');
- ▶ Numéricas: ABS(), CEIL(), MOD()
 - select ceil(3.35);
- ▶ Conversão: TO_CHAR(), TO_NUMBER(), TO_DATE()
 - select to_date('20200125', 'YYYYMMDD');

OPERADORES DO CONJUNTO

```
tpch=> select autor from autores where idade < 21
union
[select autor from autores where idade > 30;
  autor
-----
JK. Rowling
H. Melville
(2 rows)

tpch=> select autor from autores where idade >= 21
intersect
[select autor from autores where idade < 30;
  autor
-----
F. Sabino
(1 row)

tpch=> select autor from autores where idade < 21
[except
select autor from autores where idade > 30;
  autor
-----
JK. Rowling
(1 row)
```

- ▶ UNION
- ▶ INTERSECT
- ▶ EXCEPT

<u>autor</u>	email	idade
JK. Rowling	jk@gmail	20
H. Melville	M@yahoo	69
F. Sabino	S@ig	21
J. Widom	jw@gmail	NULL

JUNÇÃO

RETORNA OS REGISTROS QUE COINCIDEM ENTRE DUAS TABELAS

```
SELECT [DISTINCT] A1, A2, ..., AN  
FROM R1, [INNER | NATURAL | {LEFT | RIGHT | FULL}] {OUTER}  
JOIN R2  
WHERE <CONDIÇÃO>;
```

JUNÇÃO

RETORNA OS REGISTROS QUE COINCIDEM ENTRE DUAS TABELAS

autor		isbn
JK. Rowling		1234
H. Melville		1121
F. Sabino		2222
F. Sabino		2223
F. Sabino		2224
F. Sabino		2225

(6 rows)

EXPLÍCITO
NO FROM
(SQL99)

```
SELECT A.AUTOR, B.ISBN  
FROM AUTORES A NATURAL JOIN AUTORLIVROS B;
```

```
SELECT A.AUTOR, B.ISBN  
FROM AUTORES A INNER JOIN AUTORLIVROS B ON  
A.AUTOR=B.AUTOR;
```

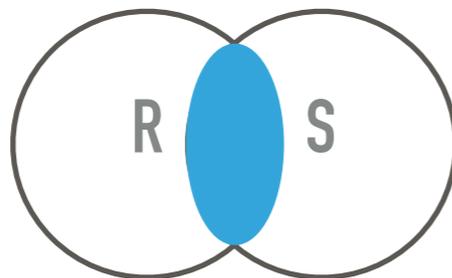
CONDIÇÃO
NO WHERE
(SQL92)

```
SELECT A.AUTOR, B.ISBN  
FROM AUTORES A, AUTORLIVROS B  
WHERE A.AUTOR=B.AUTOR;
```

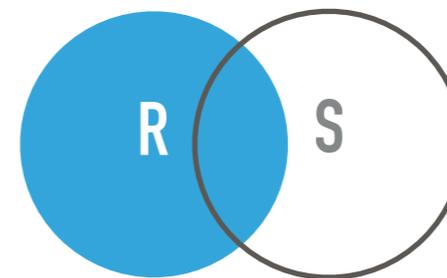
JUNÇÃO

Tipos de join	email
INNER JOIN	junta quando tabelas R e S tem o mesmo valor
LEFT OUTER JOIN	junta quando tabelas R e S tem o mesmo valor E quando somente R tem valor
RIGHT OUTER JOIN	junta quando tabelas R e S tem o mesmo valor E quando somente S tem valor
FULL OUTER JOIN	junta quando tabelas R e S tem o mesmo valor E quando R ou S tem valores únicos

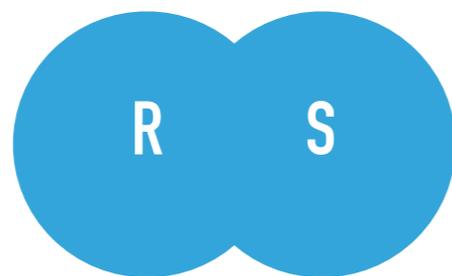
inner



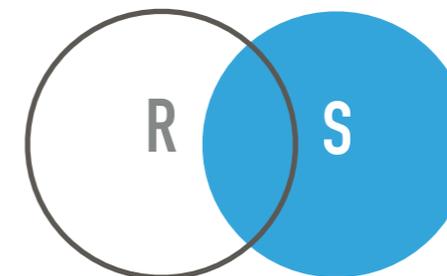
left



full



right



OUTER JOIN

<u>autor</u>	email	idade
JK. Rowling	jk@gmail	20
H. Melville	M@yahoo	69
F. Sabino	S@ig	21
J. Widom	jw@gmail	NULL

<u>autor</u>	<u>isbn</u>
JK. Rowling	1234
H. Melville	1121
F. Sabino	2222
F. Sabino	2223
F. Sabino	2224
F. Sabino	2225

```
SELECT A.AUTOR, B.ISBN  
FROM AUTORES A LEFT OUTER JOIN AUTORLIVROS B  
ON A.AUTOR=B.AUTOR;
```

```
tpch=> select a.autor, b.isbn  
tpch-> from autores A left outer join autorlivros b on a.autor=b.autor;  
autor | isbn  
-----+-----  
JK. Rowling | 1234  
H. Melville | 1121  
F. Sabino | 2222  
F. Sabino | 2223  
F. Sabino | 2224  
F. Sabino | 2225  
J. Widom |  
(7 rows)
```

AGREGAÇÃO

FUNÇÕES DE SUMARIZAÇÃO DE REGISTROS

```
SELECT A1, A2, ..., AN  
FROM R1, R2, ..., RN  
WHERE <CONDIÇÃO>  
GROUP BY COLUNAS  
HAVING <CONDIÇÃO>;
```

- ▶ COUNT(ATRIBUTO)
- ▶ MAX(ATRIBUTO)
- ▶ MIN(ATRIBUTO)
- ▶ AVG(ATRIBUTO)
- ▶ SUM(ATRIBUTO)
- ▶ STDDEV(ATRIBUTO)

AGREGAÇÃO

FUNÇÕES DE SUMARIZAÇÃO DE REGISTROS

```
[tpch=> select avg(idade) from autores;
      avg
-----
 36.666666666666667
(1 row)
```

```
SELECT COUNT(*)
FROM AUTORES;
```

```
SELECT AVG(IDADE)
FROM AUTORES;
```

```
SELECT STDDEV(IDADE)
FROM AUTORES;
```

```
[tpch=> select count(*), avg(idade), stddev(idade), max(idade), min(idade) from autores;
count |      avg      |      stddev      | max | min
-----+-----+-----+-----+-----
     4 | 36.666666666666667 | 28.0059517483932928 | 69 | 20
(1 row)
```

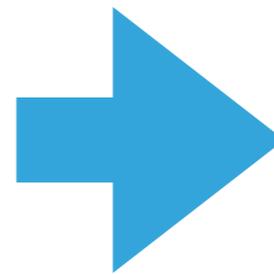
GROUP BY

FUNÇÕES DE SUMARIZAÇÃO DE REGISTROS EM GRUPOS

```
SELECT A.AUTOR, COUNT(*)  
FROM AUTORES A NATURAL JOIN AUTORLIVROS B  
GROUP BY A.AUTOR;
```



```
tpch=> select a.autor, b.isbn  
[from autores A natural join autorlivros b;  
autor | isbn  
-----+-----  
JK. Rowling | 1234  
H. Melville | 1121  
F. Sabino | 2222  
F. Sabino | 2223  
F. Sabino | 2224  
F. Sabino | 2225  
(6 rows)
```



```
tpch=> select a.autor, count(*)  
from autores A natural join autorlivros b  
[group by a.autor;  
autor | count  
-----+-----  
JK. Rowling | 1  
H. Melville | 1  
F. Sabino | 4  
(3 rows)
```

GROUP BY

FUNÇÕES DE SUMARIZAÇÃO DE REGISTROS EM GRUPOS

```
SELECT A.AUTOR, B.ISBN, COUNT(*)  
FROM AUTORES A NATURAL JOIN AUTORLIVROS B  
GROUP BY A.AUTOR;
```



```
SELECT A.AUTOR, B.ISBN, COUNT(*)  
FROM AUTORES A NATURAL JOIN AUTORLIVROS B  
GROUP BY A.AUTOR, B.ISBN;
```

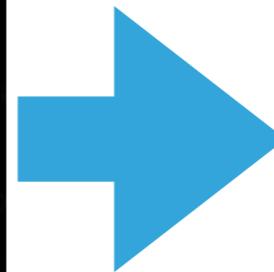


HAVING

FILTRA O SUMÁRIO

```
SELECT A.AUTOR, COUNT(*)  
FROM AUTORES A NATURAL JOIN AUTORLIVROS B  
GROUP BY A.AUTOR  
HAVING COUNT(*) > 1;
```

```
tpch=> select a.autor, count(*)  
from autores A natural join autorlivros b  
group by a.autor;  
autor | count  
-----+-----  
JK. Rowling | 1  
H. Melville | 1  
F. Sabino | 4  
(3 rows)
```



```
tpch=> select a.autor, count(*)  
from autores A natural join autorlivros b  
group by a.autor  
having count(*) > 1;  
autor | count  
-----+-----  
F. Sabino | 4  
(1 row)
```

INSERT

2 FORMAS DE INCLUIR REGISTROS

```
INSERT INTO TABELA  
VALORES (A1, A2, ..., AN);
```

```
[tpch=# insert into autores values('E. Almeida','eduardo@inf',28);  
INSERT 0 1
```

```
INSERT INTO TABELA  
SELECT <COMANDO SELECT>;
```

```
[tpch=# insert into livros  
[tpch=# select * from livrosnovos;  
INSERT 0 1
```

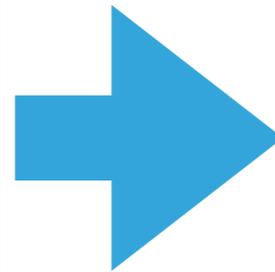
DELETE / TRUNCATE

REMOVER REGISTROS

```
DELETE FROM TABELA  
WHERE <CONDIÇÃO>;
```

```
[tpch=# delete from livros where isbn = 3333;  
DELETE 1
```

```
[tpch=# select * from livros;  
 isbn |      titulo      | editora  
-----+-----+-----  
 1234 | Harry Potter 1   | Pottermore  
 1121 | Mobi Dick        | DCL  
 2222 | Grande Mentecapto | Nordica  
 2223 | Inglesa Deslumbrada | Nordica  
 2224 | De Ponta Cabeça  | Nordica  
 2225 | Encontro Mercado | Nordica  
 3333 | Sistemas de BD   | XPTO  
(7 rows)
```



```
[tpch=# select * from livros;  
 isbn |      titulo      | editora  
-----+-----+-----  
 1234 | Harry Potter 1   | Pottermore  
 1121 | Mobi Dick        | DCL  
 2222 | Grande Mentecapto | Nordica  
 2223 | Inglesa Deslumbrada | Nordica  
 2224 | De Ponta Cabeça  | Nordica  
 2225 | Encontro Mercado | Nordica  
(6 rows)
```

REMOVER **TODOS** REGISTROS DA TABELA. EQUIVALENTE AO
DELETE FROM TABELA; (SEM WHERE)

```
TRUNCATE TABELA;
```

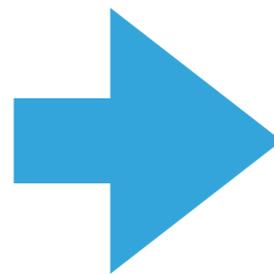
UPDATE

ATUALIZA REGISTROS

```
UPDATE TABELA  
SET A1 = EXPRESSÃO_1, A2 = EXPRESSÃO_2, ...  
WHERE <CONDIÇÃO>;
```

```
tpch=# update livros set isbn=777,titulo='Bancos de dados' where isbn=3333;  
UPDATE 1
```

```
tpch=# select * from livros;  
 isbn |      titulo      | editora  
-----+-----+-----  
 1234 | Harry Potter 1   | Pottermore  
 1121 | Mobi Dick        | DCL  
 2222 | Grande Mentecapto | Nordica  
 2223 | Inglesa Deslumbrada | Nordica  
 2224 | De Ponta Cabeça  | Nordica  
 2225 | Encontro Mercado | Nordica  
 3333 | Sistemas de BD   | XPTO  
(7 rows)
```



```
tpch=# select * from livros;  
 isbn |      titulo      | editora  
-----+-----+-----  
 1234 | Harry Potter 1   | Pottermore  
 1121 | Mobi Dick        | DCL  
 2222 | Grande Mentecapto | Nordica  
 2223 | Inglesa Deslumbrada | Nordica  
 2224 | De Ponta Cabeça  | Nordica  
 2225 | Encontro Mercado | Nordica  
  777 | Bancos de dados  | XPTO  
(7 rows)
```

DATA DEFINITION LANGUAGE(DDL)

- ▶ CREATE TABLE
- ▶ ALTER TABLE
- ▶ DROP TABLE
- ▶ CREATE VIEW
- ▶ CREATE INDEX

CREATE TABLE

```
CREATE TABLE <NOME>(
  <DEFINIÇÃO DE COLUNAS>,
  <DEFINIÇÃO DE RESTRIÇÕES>
)
<OPCIONAIS>;
```

- ▶ <DEFINIÇÃO DE COLUNAS> LISTAS DE ATRIBUTOS E TIPOS
- ▶ <DEFINIÇÃO DE RESTRIÇÕES> CHAVES PRIMÁRIAS, CHAVES ESTRANGEIRAS
- ▶ <OPCIONAIS> INFORMAÇÕES DE ARMAZENAMENTO

```
CREATE TABLE AUTORES(
  AUTOR VARCHAR(30),
  EMAIL VARCHAR(20),
  IDADE INTEGER,
  PRIMARY KEY(AUTOR)
);
```

ALGUNS TIPOS DE DADOS (PADRÃO)

- ▶ INTEGER, SMALLINT
- ▶ NUMERIC(P,D), DOUBLE
- ▶ VARCHAR, CHAR
- ▶ DATE, TIME
- ▶ ...

PRIMARY KEY

RESTRIÇÃO DE INTEGRIDADE

```
CREATE TABLE AUTORES(  
  AUTOR VARCHAR(30),  
  ...  
  PRIMARY KEY(AUTOR)  
);
```

UM ATRIBUTO

```
CREATE TABLE AUTORES(  
  AUTOR VARCHAR(30),  
  CPF VARCHAR(11),  
  ...  
  PRIMARY KEY(AUTOR, CPF)  
);
```

MÚLTIPLOS ATRIBUTOS

FOREIGN KEY REFERENCES

RESTRIÇÃO DE INTEGRIDADE

```
CREATE TABLE AUTORLIVROS(  
  AUTOR VARCHAR(30),  
  ....  
  FOREIGN KEY(AUTOR)  
  REFERENCES AUTORES(AUTOR)  
);
```

UM ATRIBUTO

```
CREATE TABLE AUTORLIVROS(  
  AUTOR VARCHAR(30),  
  CPF VARCHAR(11),  
  ....  
  FOREIGN KEY(AUTOR, CPF)  
  REFERENCES AUTORES(AUTOR, CPF)  
);
```

MÚLTIPLOS ATRIBUTOS

VALUE CONSTRAINTS

RESTRIÇÃO DE INTEGRIDADE

```
CREATE TABLE AUTOR(  
  AUTOR VARCHAR(30) UNIQUE,  
  CPF VARCHAR(11) NOT NULL,  
  IDADE INTEGER CHECK (IDADE > 0)  
);
```

AVALIAÇÃO DE RESTRIÇÕES PODE SER
COMPUTACIONALMENTE CARA

PRÓXIMA AULA

MAPEAMENTO ER-RELACIONAL