

cin.ufpe.br



UNIVERSIDADE FEDERAL DE PERNAMBUCO

Fundamentos de SQL

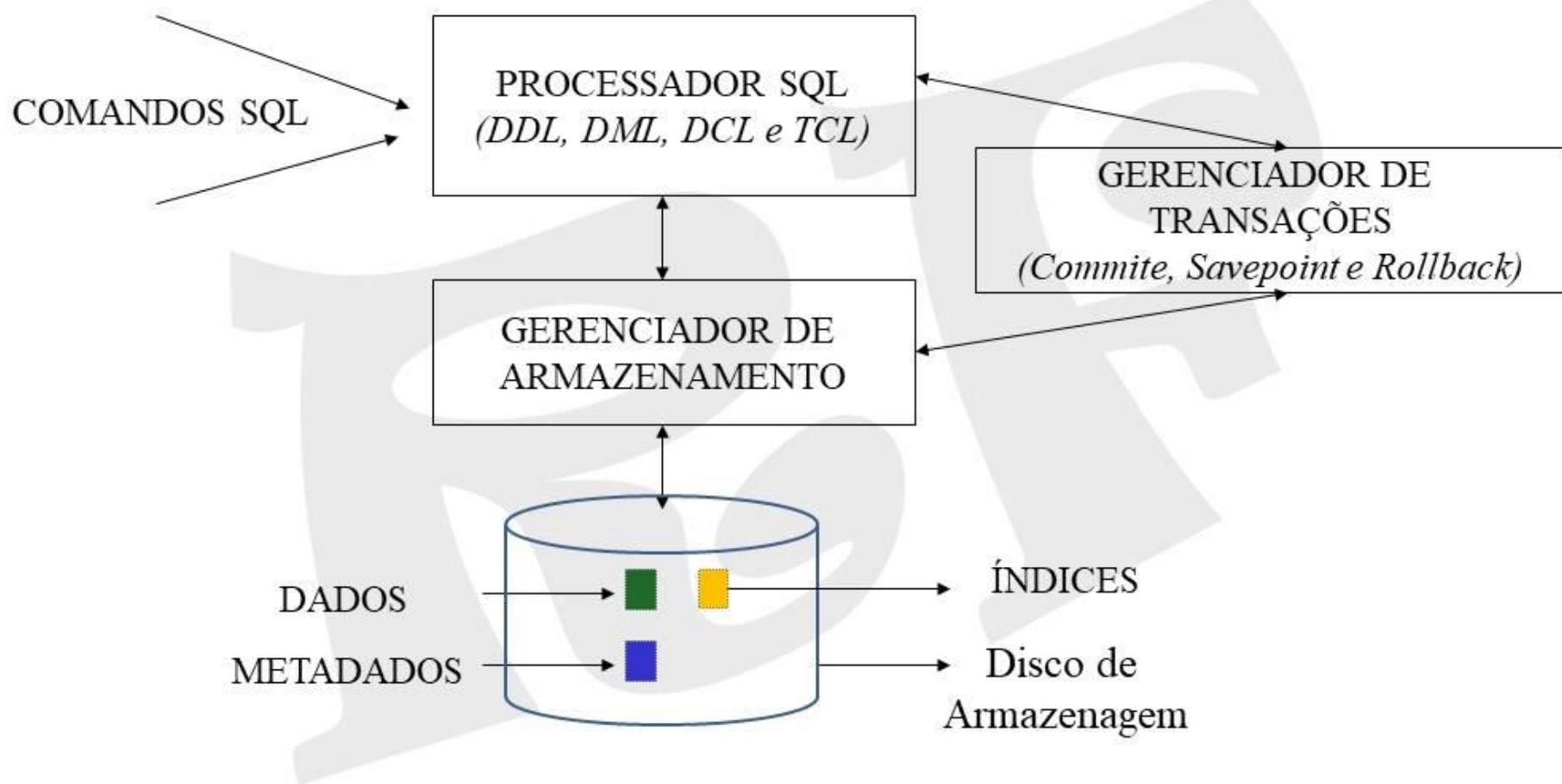
Por: Prof. Robson do Nascimento Fidalgo

rdnf@cin.ufpe.br

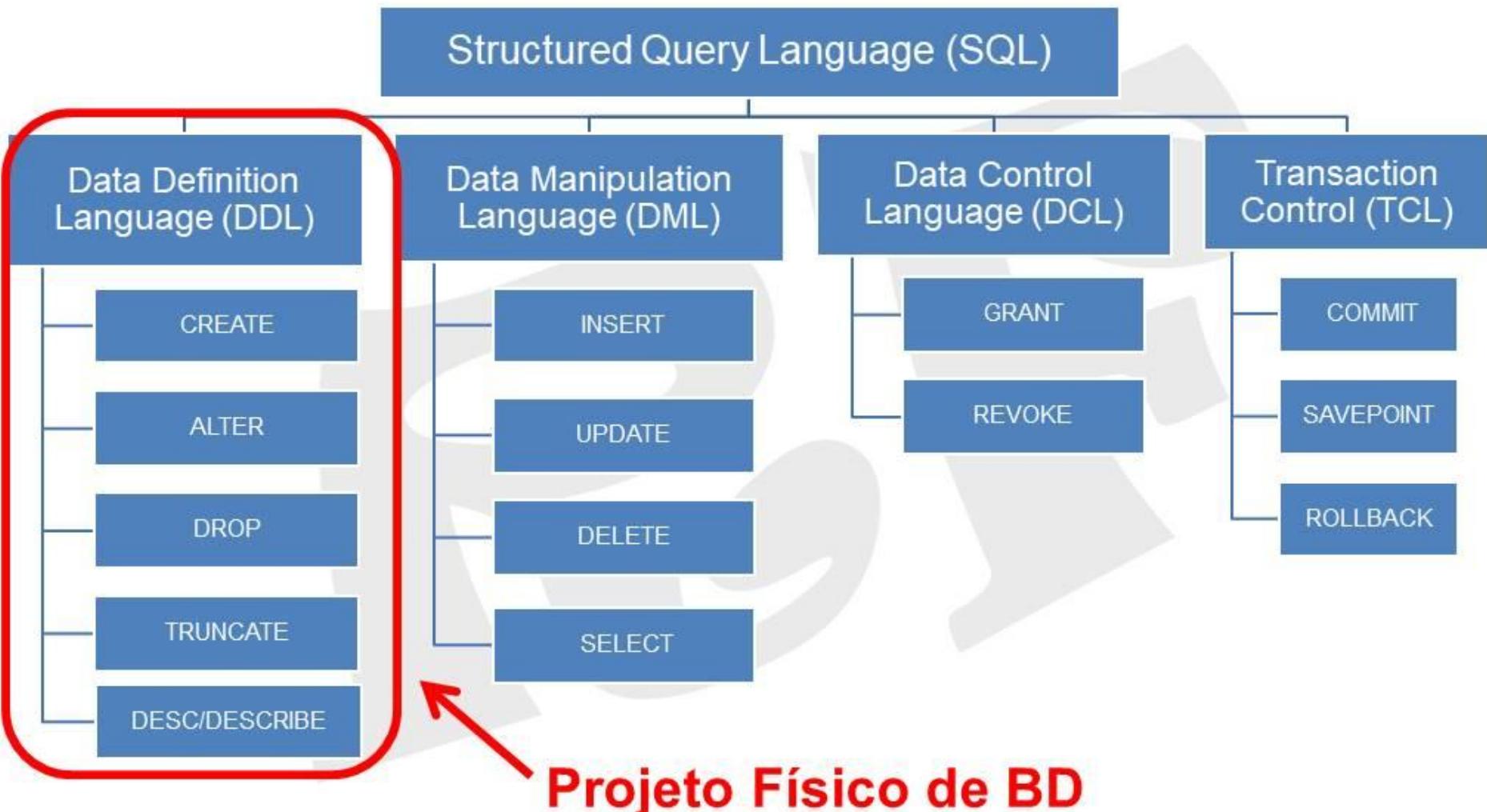
robson.fidalgo@cin.ufpe.br

<https://sites.google.com/site/robsonfidalgo/>

Arquitetura Básica de um SGBDR



Linguagens para SGBDR – Comandos Básicos



DDL – Conceitos Básicos

- ▶ A DDL permite a criação, manutenção e eliminação de objetos do banco de dados. Por exemplo:
 - ▶ Tabelas
 - ▶ Índices
 - ▶ Sequências
 - ▶ Visões
- ▶ Convenções de nomes
 - ▶ Devem começar com uma letra
 - ▶ Pode ter de 1 a 30 caracteres
 - ▶ Pode conter somente A-Z, a-z, 0-9, _, \$ e #
 - ▶ Os nomes devem ser únicos por usuário
 - ▶ Não podem ser utilizadas palavras reservadas (salvo se entre aspas)

(Dicas para uma boa Chave Primária)

- ▶ Prefira usar números inteiros, pois são mais performáticos do que strings
 - ▶ Quanto menor a chave, menor o índice, menos páginas no cache
- ▶ Chaves naturais só devem ser usadas quando forem pequenas e raramente sofrerem atualizações
 - ▶ Atualizações na chave primária impactam nos índices e nas chaves estrangeiras (ex. evitar num_passaporte ou num_contrato)
- ▶ Prefira chaves artificiais (ex. auto incremento) se a chave natural for grande ou passível de atualização

DDL – Conceitos Básicos

- ▶ Tipos de Dados Básicos
 - ▶ Oracle
 - ▶ CHAR(tamanho)
 - ▶ VARCHAR(tamanho)
 - ▶ NUMBER(total, decimais)
 - ▶ DATE
 - ▶ TIMESTAMP
 - ▶ SQL Server
 - ▶ CHAR(tamanho)
 - ▶ VARCHAR(tamanho)
 - ▶ DECIMAL(total, decimais)
 - ▶ DATE
 - ▶ DATETIME

DDL – Conceitos Básicos

- ▶ Restrições de Integridade de Tabelas
 - ▶ CONSTRAINT **PK_NOME_DA_RESTRIÇÃO** PRIMARY KEY (**COLUNAS**)
 - ▶ CONSTRAINT **FK_NOME_DA_RESTRIÇÃO** FOREIGN KEY (**COLUNAS**)
REFERENCES **NOME_DA_TABELA_PAIS**
[ON DELETE **REFERENCE_OPTION**]
[ON UPDATE **REFERENCE_OPTION**] -- *não tem no Oracle*
 - ▶ CONSTRAINT **AK_NOME_DA_RESTRIÇÃO** UNIQUE (**COLUNAS**)
 - ▶ CONSTRAINT **CK_NOME_DA_RESTRIÇÃO** CHECK (**EXPRESSAO**)
 - ▶ **REFERENCE_OPTION**
 - ▶ **RESTRICT|NO ACTION|CASCADE|SET NULL|SET DEFAULT**

DDL – Conceitos Básicos

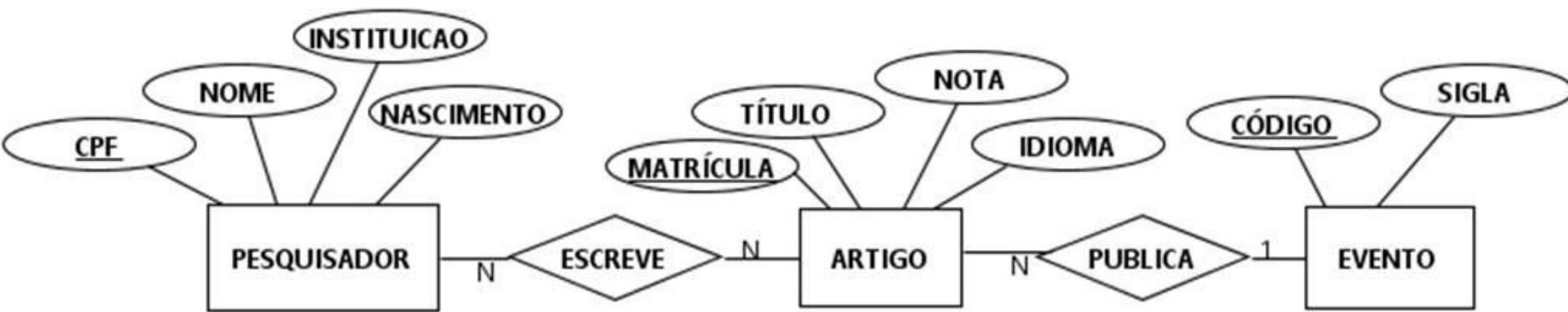
- ▶ Restrições de Integridade de Colunas
 - ▶ CONSTRAINT **NN_NOME_DA_RESTRIÇÃO** NOT NULL
 - ▶ CONSTRAINT **PK_NOME_DA_RESTRIÇÃO** PRIMARY KEY
 - ▶ CONSTRAINT **NOME_DA_RESTRIÇÃO**
REFERENCES **NOME_DA_TABELA_PAIS**
[ON DELETE **REFERENCE_OPTION**]
[ON UPDATE **REFERENCE_OPTION**] -- *não tem no Oracle*
 - ▶ CONSTRAINT **NOME_DA_RESTRIÇÃO** UNIQUE
 - ▶ CONSTRAINT **NOME_DA_RESTRIÇÃO** CHECK (**EXPRESSAO**)
- ▶ **REFERENCE_OPTION**
 - ▶ **RESTRICT|NO ACTION|CASCADE|SET NULL|SET DEFAULT**

(Restrições de Integridade Referencial)

▶ REFERENCE_OPTION

- ▶ RESTRICT: Rejeita a atualização ou exclusão de um registro da tabela pai, se houver registros na tabela filha (**não tem no oracle**)
- ▶ NO ACTION (DEFAULT): Equivalente ao RESTRICT
- ▶ CASCADE: na atualização ou exclusão de um registro da tabela pai, atualiza ou exclui os registros da tabela filha automaticamente
- ▶ SET NULL: na atualização ou exclusão de um registro da tabela pai, define como null o valor do campo na tabela filha
 - ▶ NULL não é considerado um valor, portanto, não fere as restrições de integridade referencial
- ▶ SET DEFAULT: semelhante ao SET NULL, mas o valor da coluna na tabela filha é o seu valor default (**não tem no oracle**)

DDL - Estudo de Caso



PESQUISADOR (CPF, NOME, INSTITUICAO, NASCIMENTO);

EVENTO (COD, NOME, SIGLA, ANO);

ARTIGO(MAT, TITULO, NOTA, IDIOMA, COD)

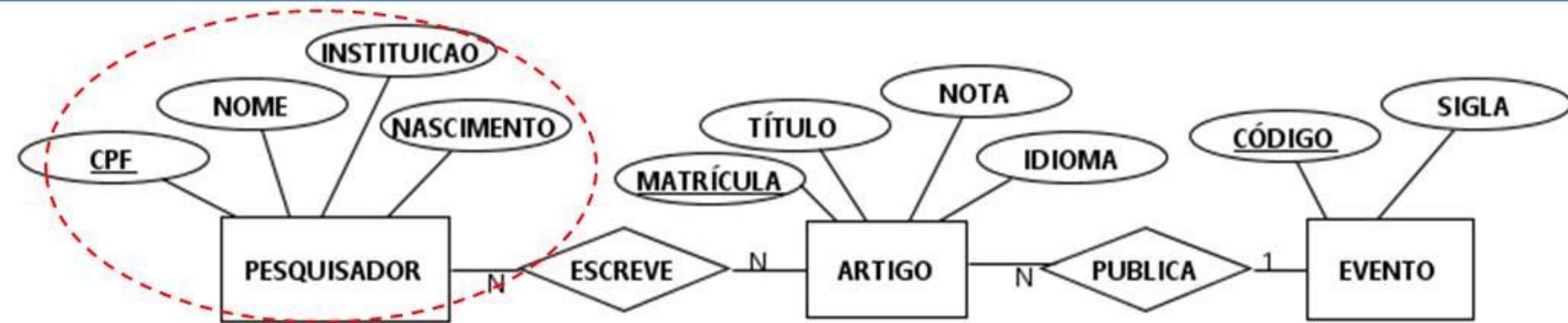
COD REFERENCIA EVENTO (COD);

ESCREVE (CPF, MAT)

CPF REFERENCIA PESQUISADOR (CPF),

MAT REFERENCIA ARTIGO (MAT);

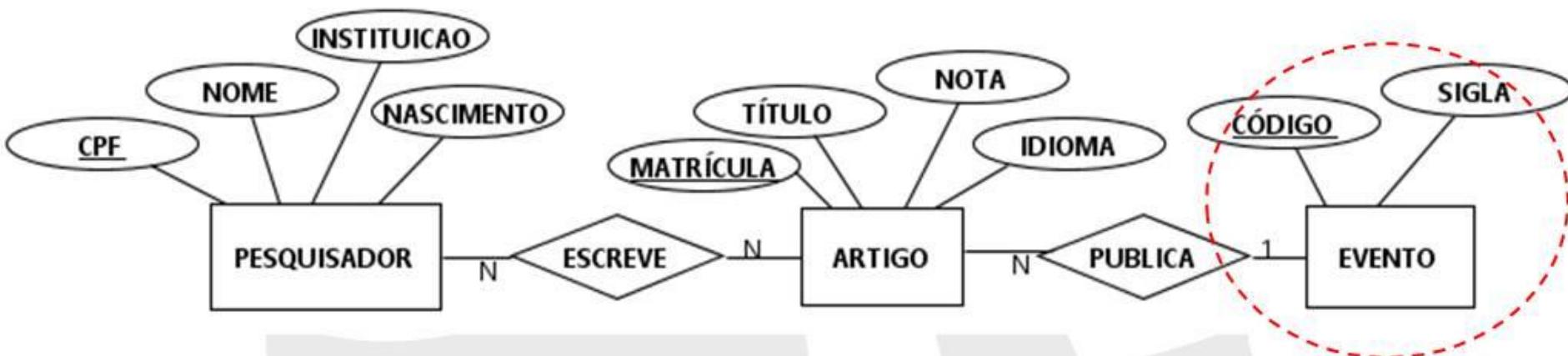
DDL - Criando Tabelas



-- CRIA TABELA PESQUISADOR

```
CREATE TABLE PESQUISADOR(  
    CPF VARCHAR(4),  
    NOME VARCHAR(80) CONSTRAINT NN_PESQ_NOME NOT NULL,  
    INSTITUICAO VARCHAR(40) NOT NULL,  
    NASCIMENTO DATE,  
    CONSTRAINT PK_USUARIOS PRIMARY KEY (CPF),  
    CONSTRAINT AK_USU_CPF UNIQUE (NOME, NASCIMENTO));
```

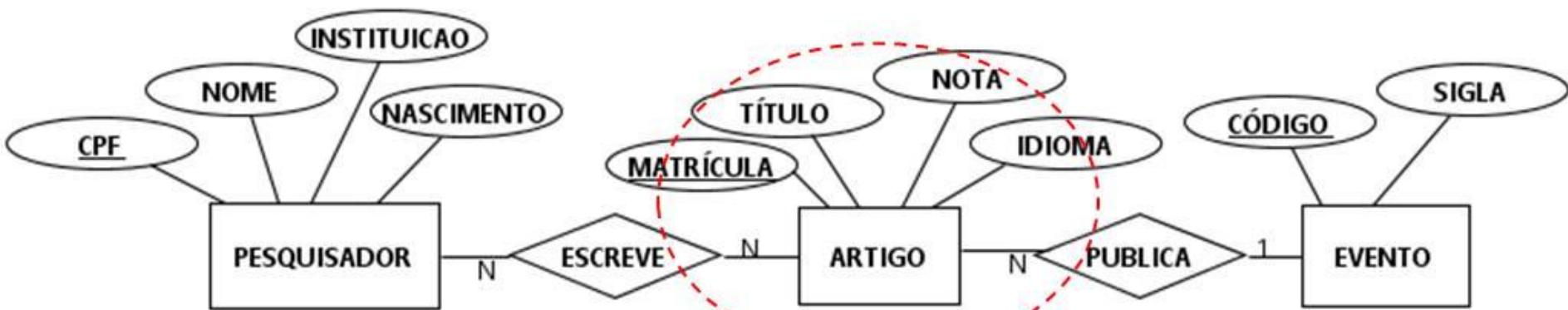
DDL - Criando Tabelas



-- CRIA TABELA EVENTO

```
CREATE TABLE EVENTO(  
COD VARCHAR(4) PRIMARY KEY ,  
NOME VARCHAR(80) NOT NULL,  
SIGLA VARCHAR(10) NOT NULL UNIQUE,  
ANO NUMBER(4));
```

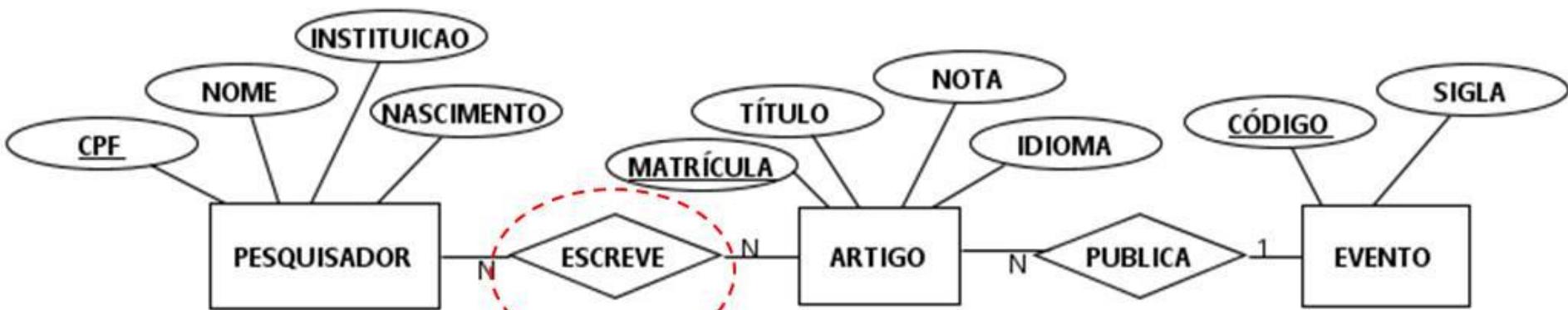
DDL - Criando Tabelas



-- CRIA TABELA ARTIGO

```
CREATE TABLE ARTIGO(  
    MAT VARCHAR(4),  
    TITULO VARCHAR(80) NOT NULL,  
    NOTA NUMBER(4,2) NOT NULL,  
    IDIOMA VARCHAR(15) DEFAULT 'PORTUGUES',  
    COD VARCHAR(4),  
    CONSTRAINT PK_ARTIGO PRIMARY KEY (MAT),  
    CONSTRAINT FK_ART_EVE FOREIGN KEY (COD) REFERENCES EVENTO (COD),  
    CONSTRAINT CHK_ART_NOTA CHECK (NOTA BETWEEN 0 AND 10));
```

DDL - Criando Tabelas



-- CRIA TABELA ESCREVE

```
CREATE TABLE ESCREVE(  
    CPF VARCHAR(4),  
    MAT VARCHAR(4),  
    CONSTRAINT ESCREVE_PK PRIMARY KEY (CPF, MAT),  
    CONSTRAINT ESCREVEPESQUISADOR_FK FOREIGN KEY  
        (CPF) REFERENCES PESQUISADOR ON DELETE CASCADE,  
    CONSTRAINT ESCREVEARTIGO_FK FOREIGN KEY (MAT)  
        REFERENCES ARTIGO ON DELETE CASCADE);
```

DDL - Criando Tabelas

--Criando uma Tabela a Partir de uma Consulta:

```
CREATE TABLE MELHORES_ARTIGOS AS  
SELECT *  
FROM ARTIGO  
WHERE NOTA>=9 ;
```

DDL - Alterando Tabelas

- ▶ **ALTER TABLE** - permite inserir/eliminar/modificar colunas nas tabelas já existentes

ORACLE

```
ALTER TABLE Tabela
{ADD [CONSTRAINT] { coluna | restrição }|
 DROP {COLUMN | CONSTRAINT} { coluna | restrição }|
 MODIFY coluna}
```

SQL SERVER

```
ALTER TABLE Tabela
{ADD [CONSTRAINT] { coluna | restrição }|
 DROP {COLUMN | CONSTRAINT} { coluna | restrição }|
 ALTER COLUMN coluna}
```

DDL - Alterando Tabelas

► Exemplos:

-- ADICIONAR O CAMPO E-MAIL NA TABELA PESQUISADOR

```
ALTER TABLE PESQUISADOR  
ADD EMAIL VARCHAR(40);
```

-- MODIFICAR O CAMPO E-MAIL NA TABELA PESQUISADOR

```
ALTER TABLE PESQUISADOR  
MODIFY EMAIL (OU ALTER COLUMN) CHAR(50);
```

-- ELIMINAR O CAMPO E-MAIL NA TABELA PESQUISADOR

```
ALTER TABLE PESQUISADOR  
DROP COLUMN EMAIL;
```

-- ADICIONAR UMA RESTRIÇÃO NA TABELA PESQUISADOR

```
ALTER TABLE PESQUISADOR  
ADD CONSTRAINT AK_PESQ_EMAIL UNIQUE (EMAIL);
```

DDL – Excluindo ou Limpando uma Tabela

- ▶ **DROP:** Exclui uma tabela por completo
 - ▶ **DROP TABLE nome_da_tabela [CASCADE CONSTRAINTS];**
 - ▶ **DROP TABLE MELHORES_ARTIGOS CASCADE CONSTRAINTS;**
 - **CASCADE CONSTRAINTS** → forçar a remoção das restrições de integridade referencial
- ▶ **TRUNCATE:** Preserva a tabela, mas, de forma rápida e liberando o espaço alocado, exclui todas as suas linhas.
 - ▶ **TRUNCATE TABLE nome_da_tabela;**
 - ▶ **TRUNCATE TABLE MELHORES_ARTIGOS**
- ▶ Comandos DDL não permitem ROLLBACK!

DDL – Criando e Excluindo Índices

- ▶ CREATE INDEX
 - ▶ CREATE [UNIQUE] INDEX nome ON tabela(colunas);
 - ▶ CREATE INDEX idx_usu_nome on PESQUISADOR (nome);
- ▶ DROP INDEX
 - ▶ DROP INDEX nome_do_índice;
 - ▶ DROP INDEX idx_usu_nome

DDL – Dicas sobre Índices

► Quando Criar Índices?

- ▶ A tabela tem muitas linhas
- ▶ A coluna contém inúmeros valores distintos (alta seletividade – ex. CPF)
- ▶ A coluna é muito usada para fazer filtros
- ▶ A coluna sofre pouca atualização

► Quando NÃO Criar Índices?

- ▶ A tabela é pequena
- ▶ A coluna contém muitos valores repetidos (baixa seletividade – ex. Sexo)
- ▶ A coluna dificilmente é usada para fazer filtros
- ▶ A coluna é atualizada frequentemente

DDL – Criando e Excluindo Sequencias

► Exemplos:

```
CREATE SEQUENCE Contar1  
START WITH 1  
INCREMENT BY 1 ;
```

```
CREATE SEQUENCE ContarNegativo1  
START WITH 0  
INCREMENT BY -1 ;
```

```
DROP SEQUENCE ContarNegativo1;
```

DDL – Usando Sequencias

- ▶ Exemplos:

```
INSERT INTO ARTIGO (MAT, TITULO, NOTA, IDIOMA) VALUES  
(Contar1.Nextval, 'XPTO', 9, 'PORTUGUES'); -- ORACLE
```

```
INSERT INTO ARTIGO (MAT, TITULO, NOTA, IDIOMA) VALUES  
(NEXT VALUE FOR Contar1, 'XPTO', 9, 'PORTUGUES'); -- SQL SERVER
```

- ▶ Uma SEQUENCE pode ser utilizada em várias tabelas diferentes, mas isto não é uma boa pratica, pois é mais organizado criar uma SEQUENCE por tabela.

DDL – Criando e Excluindo Visões

► Exemplos:

```
CREATE VIEW MELHORES_ARTIGOS_PT AS  
SELECT *  
FROM ARTIGO  
WHERE NOTA>=9 AND IDIOMA = 'PORTUGUES';
```

```
DROP VIEW MELHORES_ARTIGOS_PT;
```

DDL – Criando e Excluindo Papeis/Usuários

► Exemplos:

```
CREATE ROLE GERENCIA;
```

--NO ORACLE

```
CREATE USER TESTE IDENTIFIED BY TESTE  
DEFAULT TABLESPACE USERS;
```

--NO SQL SERVER

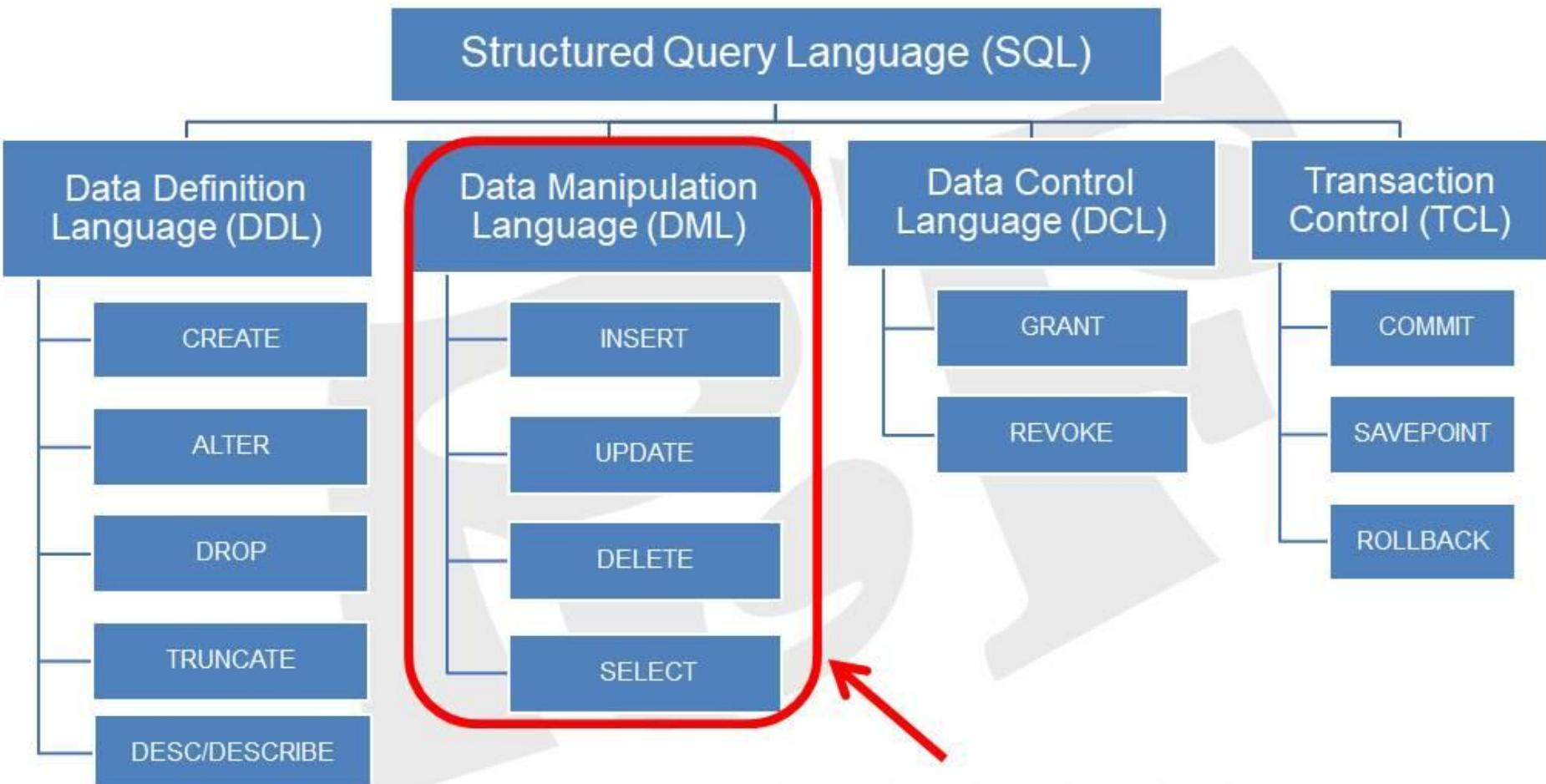
```
CREATE LOGIN TESTE WITH PASSWORD = 'TESTE';  
CREATE USER [TESTE] FOR LOGIN [TESTE] WITH DEFAULT_SCHEMA=[dbo]
```

```
DROP LOGIN/USER/ROLE;
```

DDL - Consultando Metadados

- ▶ Oracle
 - ▶ USER_OBJECTS → SELECT * FROM USER_OBJECTS;
 - ▶ DESC → DESC PESQUISADOR;
- ▶ SQL Server
 - ▶ SYS.ALL_OBJECTS → SELECT * FROM SYS.ALL_OBJECTS;
 - ▶ SP_HELP → SP_HELP PESQUISADOR;

Linguagens para SGBDR – Comandos Básicos



Manipulando Dados

DML - Inserindo Dados em Tabelas

- ▶ **INSERT INTO** - Adiciona uma linha ou várias linhas na tabela.

```
INSERT INTO Tabela [(Coluna1[, Coluna2 [, ...]])]
{VALUES (Valor1[, Valor2[, ...]) | SELECT Cláusula }
```

- ▶ Exemplos

--Insere um registro usando apenas alguns campos da tabela Artigo

```
Insert Into Artigo(CodArtigo, Titulo, Nota)
```

```
Values ('1111', 'Normalização Morreu? ', 10.00);
```

--Insere um registro completo na tabela Artigo

```
Insert Into Artigo
```

```
Values ('2222', 'Desafios em Banco de Dados', 10.00, 'Português', '1010');
```

DML - Inserindo Dados em Tabelas

► Exemplos

--Criar a tabela PesquisadorVeterano e em seguida inserir nesta tabela os Pesquisadores com nascimento < 01/01/1970

```
Create Table PesquisadorVeterano(  
    CPF Varchar(4),  
    Nome Varchar(80) NOT NULL,  
    Instituicao Varchar(40) NOT NULL,  
    Nascimento Date,  
    Primary Key (CPF),  
    Unique (Nome, Nascimento));
```

```
Insert Into PesquisadorVeterano  
SELECT CPF, Nome, Instituicao, Nascimento  
FROM Pesquisador  
WHERE Nascimento < '01/01/1970';
```

DML - Atualizando Dados em Tabelas

- ▶ **UPDATE** - Com base nos critérios especificados, altera valores de campos de uma tabela.

```
UPDATE Tabela
SET Coluna = Valor
[WHERE Condição];
```

Exemplo:

```
/* ALTERAR O NOME E ANO DO EVENTO
COM COD = '1111' */
UPDATE EVENTO
SET NOME = 'NOVO NOME',
ANO = 2000
WHERE COD = '1111';
```

```
/* TIRAR 1 PONTO DE TODOS
OS ARTIGOS */
UPDATE ARTIGO
SET NOTA = NOTA - 1;
```

DML - Excluindo Dados em Tabelas

- ▶ **DELETE** - Exclui as linhas de uma ou mais tabelas que satisfaçam a condição. Diferente do TRUNCATE, o DELETE permite ROLLBACK.

```
DELETE  
FROM Tabela  
[WHERE Condição]
```

Exemplo:

--*Excluir os registros da tabela Artigos, onde CodArtigo = '1111'*

```
DELETE  
FROM Artigo  
WHERE CodArtigo = '1111';
```

--*Excluir todos os registros da tabela Artigo*

```
DELETE  
FROM Artigo;
```

DML - Consultando Dados em Tabelas

- Estrutura Genérica

```
SELECT [DISTINCT | ALL] { * | [Tabela.]Coluna1[AS Alias1]
    [ [Tabela.]Coluna2[AS Alias2] [, ...]]}
FROM Tabela1 [, Tabela2 [, ... ]]
[WHERE {Condição Simples | Condição de Sub-consulta}]
[ORDER BY Coluna1 [ASC | DESC] [, Coluna2[ASC | DESC] [, ... ]]]
[GROUP BY Coluna1 [, Coluna2 [, ... ]]] [HAVING Condição ]
[ {UNION | INTERSECT | EXCEPT} SELECT ... ]
```

DML - Consultando Dados em Tabelas

- Estrutura Básica

SELECT → PROJEÇÃO

FROM → TABELA OU PRODUTO CARTESIANO DELAS

WHERE → SELEÇÃO

$$\Pi_{Coluna1[, Coluna2 [, \dots]]} (\sigma_{Condição}(Tabela1 [\times Tabela2 [\times \dots]]))$$

SELECT *Coluna1[, Coluna2 [, ...]]*

FROM *Tabela1,[Tabela2 [, ...]]*

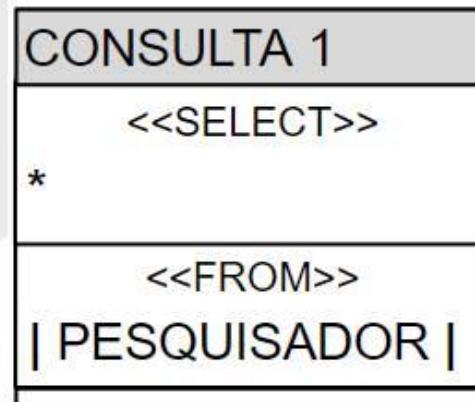
WHERE *Condição*

DML - Consultando Dados em Tabelas

▶ Exemplos:

--CONSULTA 1 - PROJETAR TODAS AS INFORMAÇÕES DOS PESQUISADORES

```
SELECT *  
FROM PESQUISADOR ;
```



DML - Consultando Dados em Tabelas

▶ Exemplos:

-- CONSULTA 2 - PROJETAR A MÉDIA DAS NOTAS DOS ARTIGOS

```
SELECT AVG (NOTA)  
FROM ARTIGO;
```

CONSULTA 2
<<SELECT>>
AVG(NOTA)
<<FROM>>
ARTIGO

DML - Consultando Dados em Tabelas

▶ Exemplos:

```
/* CONSULTA 3 - PROJETAR TODOS OS ARTIGOS(TÍTULOS) E SUAS NOTAS  
COM -1 PONTO */
```

```
SELECT TITULO, NOTA - 1  
FROM ARTIGO;
```



DML - Consultando Dados em Tabelas

► Exemplos:

```
/* CONSULTA 4 - PROJETAR A QUANTIDADE DE PESQUISADORES  
CADASTRADOS*/
```

```
SELECT COUNT (*) AS QUANTIDADE, 'UD' AS UNIDADE  
FROM PESQUISADOR;
```

CONSULTA 4
<<SELECT>> COUNT(*) : QUANTIDADE 'UD' : UNIDADE
<<FROM>> PESQUISADOR

DML - Consultando Dados em Tabelas

▶ Exemplos:

-- CONSULTA 5 - PROJETAR TODOS OS IDIOMAS DOS ARTIGOS

SELECT ALL IDIOMA -- ou **SELECT IDIOMA**
FROM ARTIGO;

CONSULTA 5
<<SELECT>>
IDIOMA
<<FROM>>
ARTIGO

DML - Consultando Dados em Tabelas

▶ Exemplos:

```
/* CONSULTA 6 - PROJETAR TODOS OS IDIOMAS, SEM DUPLICATAS, DOS ARTIGOS */
```

```
SELECT DISTINCT IDIOMA  
FROM ARTIGO;
```

CONSULTA 6 #
<<SELECT>>
IDIOMA
<<FROM>>
ARTIGO

DML - Consultando Dados em Tabelas

▶ Exemplos:

```
/* CONSULTA 7 - PROJETAR TODOS OS NOMES E INSTITUIÇÕES DA TABELA  
PESQUISADOR */
```

```
SELECT P.NOME, P.INSTITUICAO  
FROM PESQUISADOR P;
```

CONSULTA 7
<<SELECT>>
P.NOME
P.INSTITUICAO
<<FROM>>
PESQUISADOR:P

DML - Consultando Dados em Tabelas

▶ Exemplos:

/ CONSULTA 8 - PROJETAR TODOS OS TÍTULOS DOS ARTIGOS E SUAS NOTAS COM -1 PONTO */*

```
SELECT TITULO, NOTA -1 AS REDUCAO  
FROM ARTIGO;
```

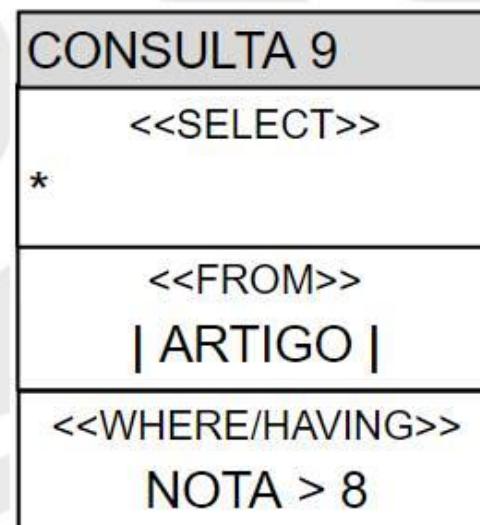
CONSULTA 8
<<SELECT>>
TITULO
NOTA -1:REDUCAO
<<FROM>>
ARTIGO

DML - Consultando Dados em Tabelas

▶ Exemplos:

--CONSULTA 9 - PROJETAR ARTIGOS COM NOTA ACIMA DE 8

```
SELECT *  
FROM ARTIGO  
WHERE NOTA > 8;
```



DML - Consultando Dados em Tabelas

▶ Exemplos:

-- CONSULTA 10 - PROJETAR ARTIGOS COM NOTA ENTRE 7 e 9

```
SELECT *  
FROM ARTIGO  
WHERE NOTA BETWEEN 7 AND 9;
```

CONSULTA 10
<<SELECT>>
*
<<FROM>> ARTIGO
<<WHERE/HAVING>> NOTA BETWEEN 7 AND 9

DML - Consultando Dados em Tabelas

► Exemplos:

```
/* CONSULTA 11 - PROJETAR OS EVENTOS QUE OCORRERAM EM  
2010, 2012 OU 2014 */
```

```
SELECT *  
FROM EVENTO  
WHERE ANO IN (2010, 2012, 2014);
```

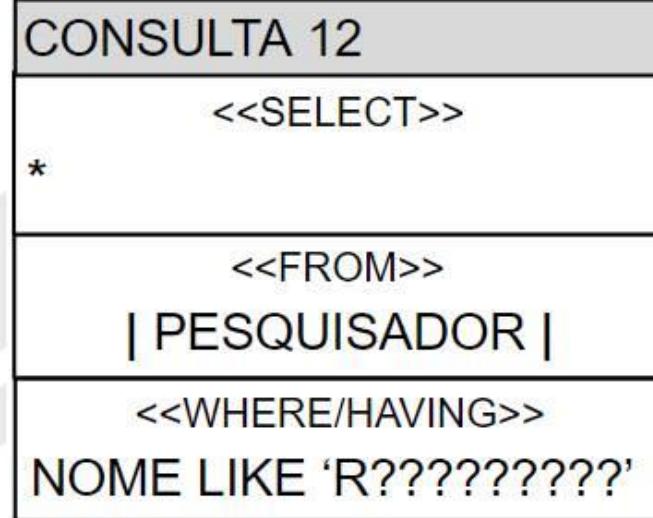
CONSULTA 11
<<SELECT>>
*
<<FROM>> EVENTO
<<WHERE/HAVING>> ANO IN (2010, 2012, 2014)

DML - Consultando Dados em Tabelas

▶ Exemplos

**/* CONSULTA 12 - PROJETAR TODOS OS PESQUISADORES CUJO NOME
TENHA 10 CARACTERES E INICIE COM R */**

```
SELECT *  
  
FROM PESQUISADOR  
  
WHERE NOME LIKE 'R??????????';
```



DML - Consultando Dados em Tabelas

▶ Exemplos

/*CONSULTA 13 - PROJETAR TODOS OS ARTIGOS QUE TENHAM BANCO DE DADOS NO SEU TÍTULO */

```
SELECT *  
FROM ARTIGO  
WHERE TITULO LIKE '%BANCO DE DADOS%';
```

CONSULTA 13
<<SELECT>>
*
<<FROM>> PESQUISADOR
<<WHERE/HAVING>>
TITULO LIKE '%BANCO DE DADOS%'

DML - Consultando Dados em Tabelas

▶ Exemplos

```
/*CONSULTA 14 - PROJETAR TODOS OS ARTIGOS QUE ESTÃO SEM  
NOTA DEFINIDA */
```

```
SELECT *  
FROM ARTIGO  
WHERE A IS NULL;
```

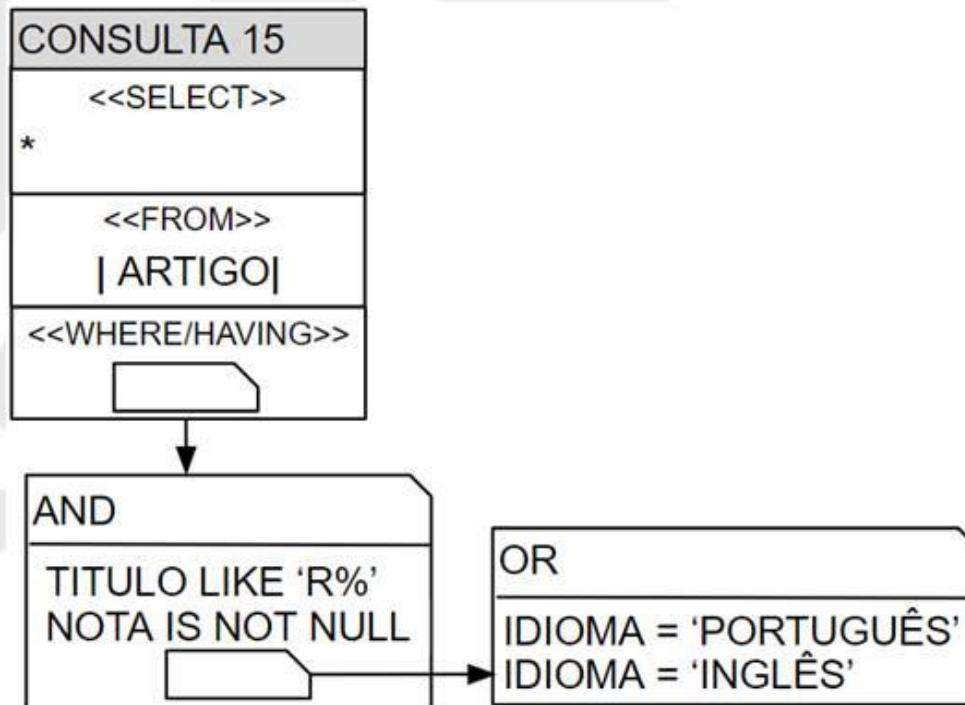
CONSULTA 14
<<SELECT>>
*
<<FROM>> ARTIGO
<<WHERE/HAVING>> NOTA IS NULL

DML - Consultando Dados em Tabelas

▶ Exemplo

```
/* CONSULTA 15 - PROJETAR TODOS OS ARTIGOS QUE INICIAM COM R, ESTÃO  
COM NOTA DEFINIDA E FORAM ESCRITOS EM PORTUGUÊS OU INGLÊS*/
```

```
SELECT *  
FROM ARTIGO  
WHERE TITULO LIKE 'R%' AND  
      NOTA IS NOT NULL AND  
      (IDIOMA = 'PORTUGUÊS' OR  
       IDIOMA = 'INGLÊS');
```



DML - Consultando Dados em Tabelas

▶ Exemplo

```
/* CONSULTA 16 - PROJETAR , EM LETRAS MINÚSCULAS, OS NOMES E, EM  
LETRAS MAIÚSCULAS, AS INSTITUIÇÕES DA TABELA PESQUISADOR */
```

```
SELECT LOWER(P.NOME), UPPER (P.INSTITUICAO)  
FROM PESQUISADOR P;
```

CONSULTA 16
<<SELECT>>
LOWER(P.NOME)
UPPER(P.INSTITUICAO)
<<FROM>>
PESQUISADOR:P

DML - Consultando Dados em Tabelas

▶ Exemplo (Funções para manipular strings)

CONCAT('Hello','World');

RESULTADO: HelloWorld

SUBSTR/SUBSTRING('HelloWorld',1,5)

RESULTADO Hello

LENGTH/LEN('HelloWorld')

RESULTADO: 10

INSTR('HelloWorld', 'W')

CHARINDEX('W', 'HelloWorld')

RESULTADO: 6

LPAD(99999, 10, '') -- Só ORACLE

RESULTADO: *****99999

INITCAP('teste') -- Só ORACLE

RESULTADO: Teste

RPAD(99999, 10, '') -- Só ORACLE

RESULTADO: 99999*****

REPLACE('Jack and Jue', 'J', 'Bl')

RESULTADO: Black and Blue

TRIM(' um teste doido ')

RESULTADO: [um teste doido]

TRIM('*' FROM '**teste**')

RESULTADO: [teste]

DML - Consultando Dados em Tabelas

- ▶ Exemplo (Funções para manipular strings)

```
/* CONSULTA 17 - PROJETAR A CONCATENAÇÃO DE DUAS STRINGS E  
MOSTRÁ-LAS EM MAIÚSCULO */
```

```
SELECT 'Hello World' ORIGINAL, UPPER(CONCAT('Hello World ', 'Teste') ) MODIFICADO  
FROM DUAL;
```

CONSULTA 17
<<SELECT>>
'HELLO WORLD':ORIGINAL
UPPER(CONCAT('HELLO WORLD', 'TESTE')):MODIFICADOR
<<FROM>>
DUAL

ORIGINAL

MODIFICADO

Hello World

HELLO WORLD TESTE

DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (Função para converter string em número)

-- CONSULTA 18 - PROJETAR A SOMA DE UM NÚMERO E UMA STRING

```
SELECT 100+TO_NUMBER('100') -- (ou CAST('100' as INT) no SQL Server)  
FROM DUAL;
```

CONSULTA 18
<<SELECT>>
100+TO_NUMBER('100')
<<FROM>>
DUAL

DML - Consultando Dados em Tabelas

▶ Exemplo (Funções para manipular de números)

Arredondar com 2 casas

ROUND (45.926, 2)

RESULTADO 45.93

Eliminar a terceira casa decimal

TRUNC(45.926, 2) -- Só ORACLE

RESULTADO 45.92

Obter o resto da divisão

MOD(1600,300) -- Só ORACLE

RESULTADO 100

Obter a dezena

ROUND(45.923, -1)

RESULTADO 50

DML - Consultando Dados em Tabelas

▶ Exemplo (Manipulando datas)

-- CONSULTA 19 - PROJETARA DATA DO SISTEMA

```
SELECT SYSDATE -- (ou GETDATE())
FROM DUAL
```

CONSULTA 19
<<SELECT>>
SYSDATE
<<FROM>>
DUAL

DML - Consultando Dados em Tabelas

▶ Exemplo (Manipulando datas)

-- CONSULTA 20 - PROJETAR O DIA, MÊS E ANO DE DATA DO SISTEMA

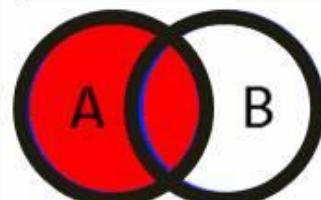
```
SELECT DAY(SYSDATE), MONTH(SYSDATE), YEAR(SYSDATE),  
FROM DUAL;
```

CONSULTA 20
<<SELECT>>
DAY(SYSDATE)
MONTH(SYSDATE)
YEAR(SYSDATE)
<<FROM>>
DUAL

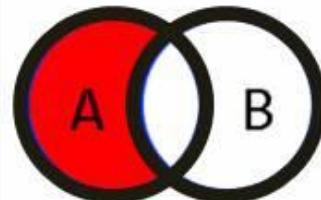
DML - Consultando Dados em Tabelas

▶ Junções

LEFT OUTER JOIN

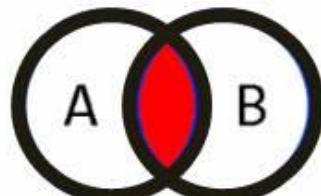


```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY
```



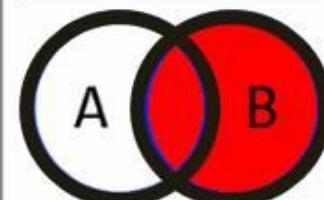
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE b.KEY IS NULL
```

INNER JOIN

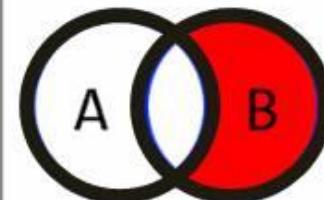


```
SELECT *  
FROM TableA a  
INNER JOIN TableB b  
ON a.KEY = b.KEY
```

RIGHT OUTER JOIN

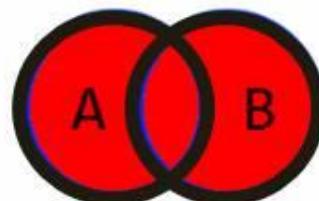


```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY
```

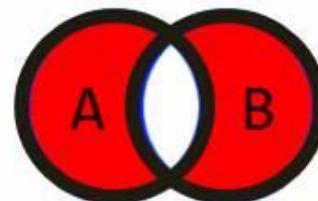


```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL
```

FULL OUTER JOIN



```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY
```



```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL  
OR b.KEY IS NULL
```

DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (junção interna – Old School)

/ CONSULTA 21 - PROJETAR O TÍTULO DE TODOS OS ARTIGOS **PUBLICADOS** E
A SIGLA DE SEUS RESPECTIVOS EVENTOS - SEM USAR “INNER JOIN” */*

```
SELECT A.TITULO, E.SIGLA  
FROM ARTIGO A, EVENTO E  
WHERE A.COD = E.COD;
```

CONSULTA 21
<<SELECT>>
A.TITULO E.SIGLA
<<FROM>> ARTIGO:A EVENTO:E
<<WHERE/HAVING>> A.COD = E.COD

DML - Consultando Dados em Tabelas

► Exemplo (junção interna – New School)

/* CONSULTA 22 - PROJETAR O TÍTULO DE TODOS OS ARTIGOS PUBLICADOS E A SIGLA DOS SEUS RESPECTIVOS EVENTOS - USANDO “INNER JOIN” */

```
SELECT A.TITULO, E.SIGLA  
FROM ARTIGO A INNER JOIN  
      EVENTO E ON A.COD = E.COD;
```

CONSULTA 22

<<SELECT>>

A.TITULO
E.SIGLA

<<FROM>>

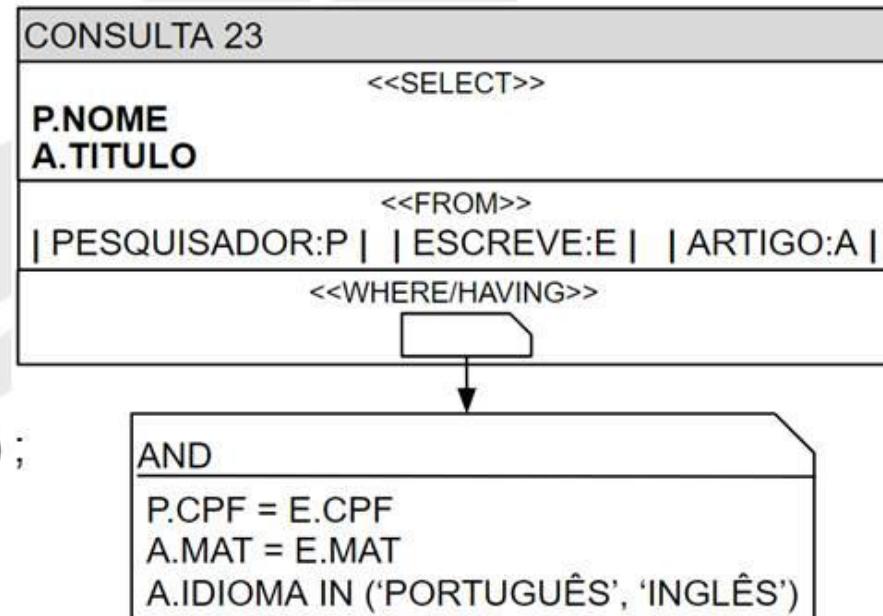
| ARTIGO:A |----->| EVENTO:E |

DML - Consultando Dados em Tabelas

► Exemplo (junção interna encadeada – Old School)

```
/* CONSULTA 23 - PROJETAR OS NOMES DE TODOS OS PESQUISADORES E OS  
TÍTULOS DE SEUS RESPECTIVOS ARTIGOS PUBLICADOS NOS IDIOMAS  
PORTUGUÊS OU INGLÊS - SEM USAR "INNER JOIN" */
```

```
SELECT P.NOME, A.TITULO  
  
FROM PESQUISADOR, ESCREVE, ARTIGO  
  
WHERE P.CPF = E.CPF AND  
  
E.MAT = A.MAT AND  
  
A.IDIOMA IN ('PORTUGUÊS','INGLÊS');
```

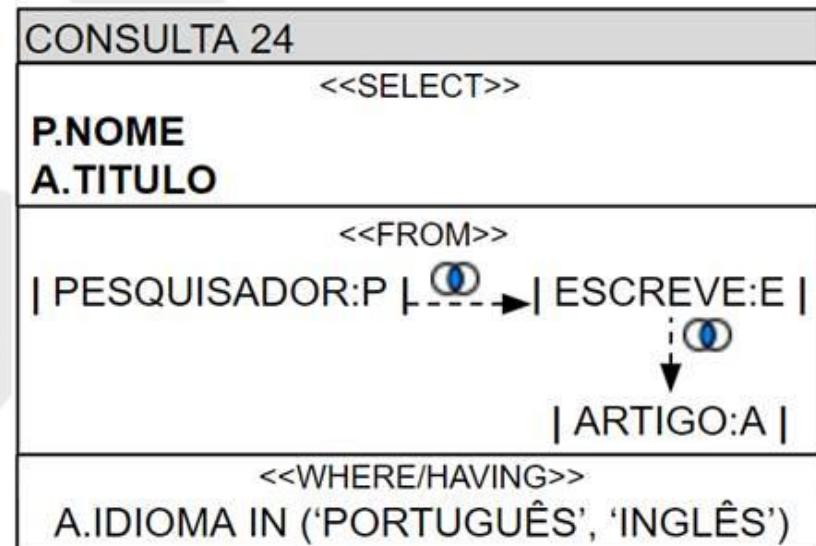


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção interna encadeada – New School)

```
/* CONSULTA 24 - PROJETAR OS NOMES DE TODOS OS PESQUISADORES E OS  
TÍTULOS DOS SEUS RESPECTIVOS ARTIGOS PUBLICADOS NOS IDIOMAS  
PORTUGUÊS OU INGLÊS - USANDO "INNER JOIN" */
```

```
SELECT P.NOME, A.TITULO  
  
FROM PESQUISADOR P INNER JOIN  
  
      ESCREVE E ON P.CPF = E.CPF INNER JOIN  
  
      ARTIGO A ON E.MAT = A.MAT  
  
WHERE A.IDIOMA IN ('PORTUGUÊS','INGLÊS');
```

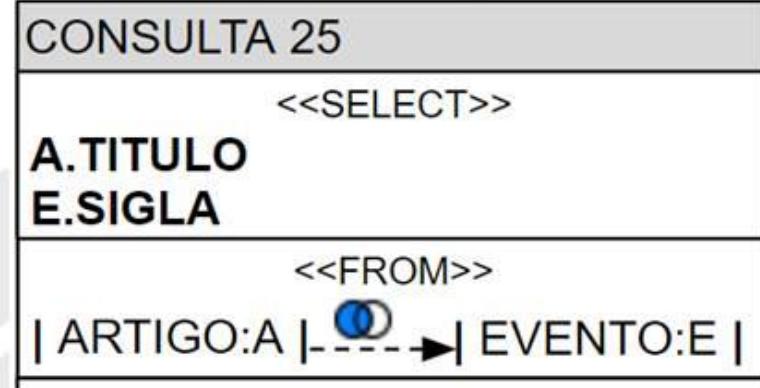


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa à esquerda)

```
/* CONSULTA 25 - PROJETAR OS TÍTULOS DE TODOS OS ARTIGOS E SEUS  
RESPECTIVOS EVENTOS (SIGLAS) */
```

```
SELECT A.TITULO, E.SIGLA  
FROM ARTIGO A LEFT OUTER JOIN  
      EVENTO E ON A.COD = E.COD;
```

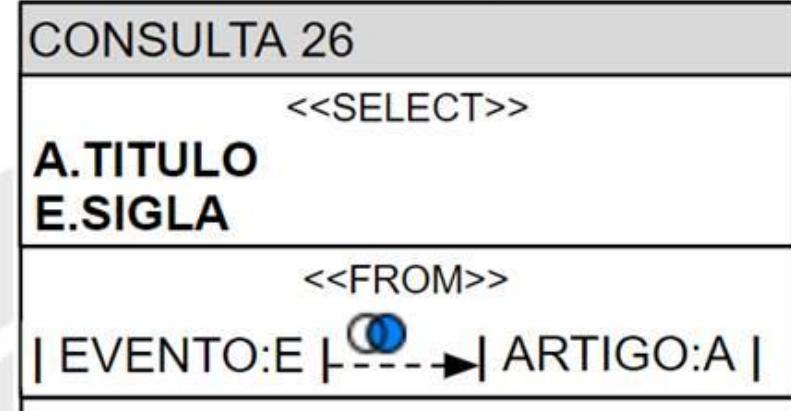


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa à direita)

```
/* CONSULTA 26 - PROJETAR TODOS OS ARTIGOS (TÍTULOS) E SEUS  
RESPECTIVOS EVENTOS (SIGLAS) */
```

```
SELECT A.TITULO, E.SIGLA  
FROM EVENTO E RIGHT OUTER JOIN  
      ARTIGO A ON A.COD = E.COD;
```

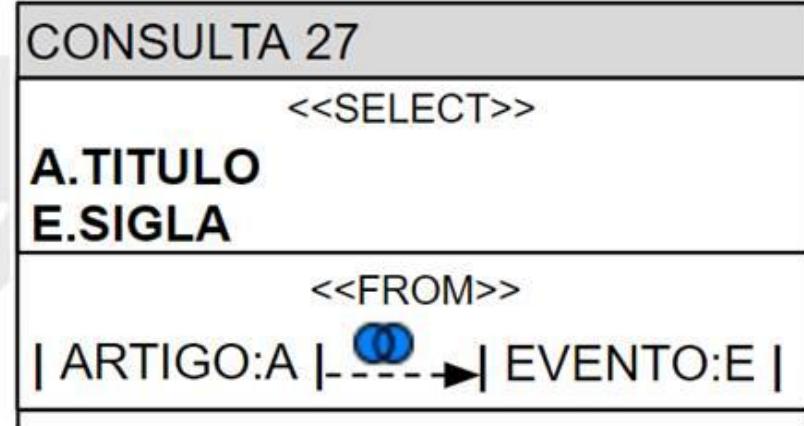


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa completa)

```
/* CONSULTA 27 - PROJETAR OS ARTIGOS (TÍTULOS) E SEUS RESPECTIVOS  
EVENTOS (SIGLAS), INCLUINDO OS ARTIGOS NÃO PUBLICADOS E OS EVENTOS  
SEM ARTIGOS */
```

```
SELECT A.TITULO, E.SIGLA  
FROM ARTIGO A FULL OUTER JOIN  
      EVENTO E ON A.COD = E.COD;
```

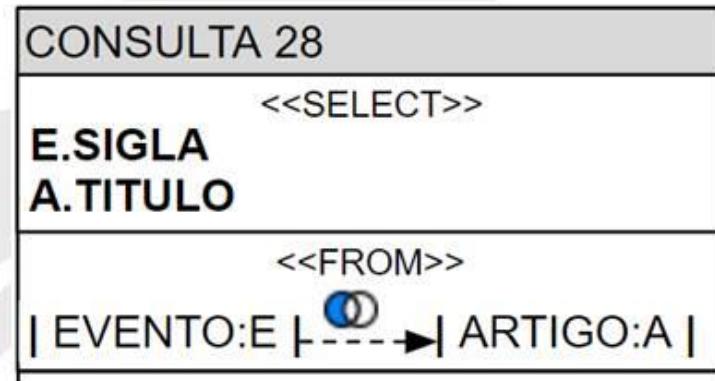


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa à esquerda exclusiva)

```
/* CONSULTA 28 - PROJETAR OS EVENTO (SIGLAS) SEM ARTIGOS PUBLICADOS  
(NULL PARA TÍTULO DO ARTIGO) - USANDO “LEFT OUTER JOIN” */
```

```
SELECT E.SIGLA, A.TITULO  
FROM EVENTO E LEFT OUTER JOIN  
    ARTIGO A ON E.COD = A.COD  
WHERE A.COD IS NULL;
```

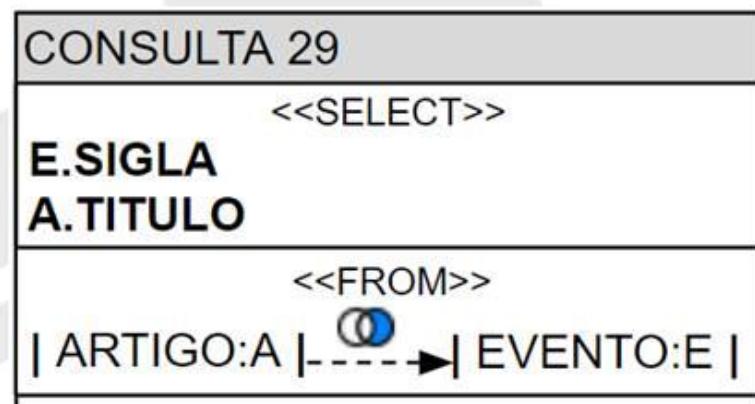


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa à direita exclusiva)

**/* CONSULTA 29 - PROJETAR OS EVENTO (SIGLAS) SEM ARTIGOS PUBLICADOS
(NULL PARA TÍTULO DO ARTIGO) - USANDO “RIGTH OUTER JOIN” */**

```
SELECT E.SIGLA, A.TITULO  
FROM ARTIGO A RIGHT OUTER JOIN  
      EVENTO E ON E.COD = A.COD  
WHERE A.COD IS NULL;
```

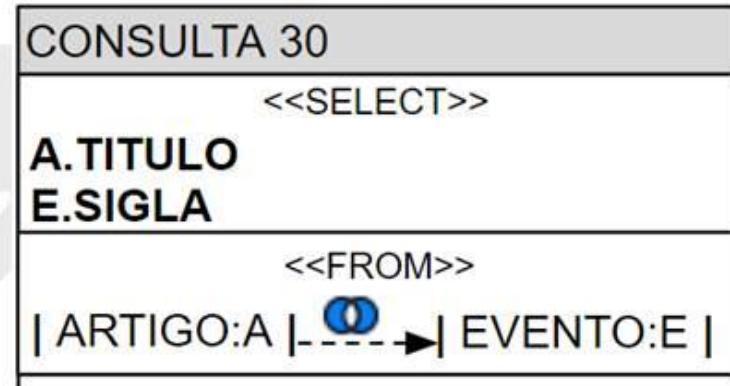


DML - Consultando Dados em Tabelas

- ▶ Exemplo (junção externa completa exclusiva)

```
/* CONSULTA 30 - PROJETAR OS ARTIGOS (TÍTULOS) NÃO PUBLICADOS E OS  
EVENTOS (SIGLAS) SEM ARTIGOS - USANDO “FULL OUTER JOIN” */
```

```
SELECT A.TITULO, E.SIGLA  
  
FROM ARTIGO A FULL OUTER JOIN  
  
      EVENTO E ON A.COD = E.COD  
  
WHERE A.COD IS NULL OR  
  
      E.COD IS NULL;
```

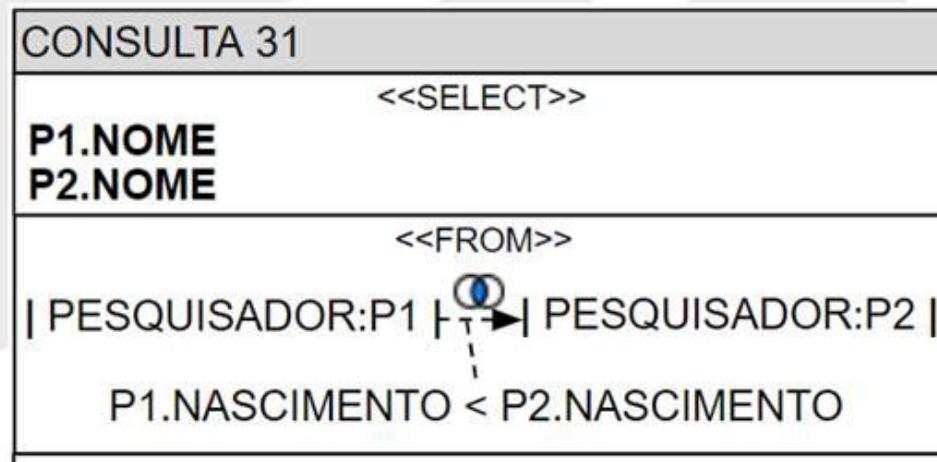


DML - Consultando Dados em Tabelas

- ▶ Exemplo (auto-junção com theta join/non-equi join)

```
/* CONSULTA 31 - PARA CADA PESQUISADOR, PROJETAR SEU NOME E OS NOMES  
DOS PESQUISADORES MAIS NOVOS DO QUE ELE */
```

```
SELECT P1.NOME, P2.NOME  
FROM PESQUISADOR P1 INNER JOIN  
      PESQUISADOR P2 ON P1.NASCIMENTO < P2.NASCIMENTO
```



DML - Consultando Dados em Tabelas

▶ Exemplo (ordenando o resultado)

```
/* CONSULTA 32 - PROJETAR O NOME E O NASCIMENTO DOS  
PESQUISADORES EM ORDEM DECRESCENTE DO NASCIMENTO. PARA DATAS  
IGUAIS, CONSIDERAR A ORDEM ALFABÉTICA DO NOME DO PESQUISADOR */
```

```
SELECT NOME, NASCIMENTO  
FROM PESQUISADOR  
ORDER BY NASCIMENTO DESC, NOME ASC;
```

CONSULTA 32
<<SELECT>>
[2↑] NOME
[1↓] NASCIMENTO
<<FROM>>
PESQUISADOR

DML - Consultando Dados em Tabelas

▶ Exemplo (agrupando/sumarizando o resultado)

-- CONSULTA 33 - PROJETAR A MÉDIA DAS NOTAS DOS ARTIGOS POR EVENTO

```
SELECT E.SIGLA, AVG (A.NOTA) AS MEDIA_NOTA  
FROM ARTIGO A INNER JOIN  
      EVENTO E ON A.COD = E. COD  
GROUP BY E.SIGLA;
```

CONSULTA 33	
<<SELECT>>	
(1) E.SIGLA	AVG(A.NOTA):MEDIA_NOTA
<<FROM>>	
ARTIGO:A	---> EVENTO:E

DML - Consultando Dados em Tabelas

▶ Exemplo (restringindo os grupos do resultado)

```
/* CONSULTA 34 - PROJETAR AS SIGLAS DOS EVENTOS CUJAS MÉDIAS DAS  
NOTAS DOS ARTIGOS SÃO MAIORES DO QUE 8 */
```

```
SELECT E.SIGLA, AVG (A.NOTA)  
FROM EVENTO E INNER JOIN  
      ARTIGO A ON E.COD = A. COD  
GROUP BY E.SIGLA  
HAVING AVG (A.NOTA) > 8;
```

CONSULTA 34
<<SELECT>> (1) E.SIGLA AVG(A.NOTA)
<<FROM>> EVENTO:E ---> ARTIGO:A
<<WHERE/HAVING>> AVG(A.NOTA) > 8

DML - Consultando Dados em Tabelas

▶ Exemplo (agrupando/sumarizando o resultado)

```
/* CONSULTA 35 - PROJETAR OS NOMES DOS PESQUISADORES QUE  
PUBLICARAM MAIS DE 3 ARTIGOS */
```

```
SELECT P.NOME, COUNT(*)  
FROM PESQUISADOR P INNER JOIN  
      ESCREVE E ON P.CPF = E.CPF INNER JOIN  
GROUP BY P.NOME  
HAVING COUNT(A.TITULO) > 3;
```

CONSULTA 35
<<SELECT>>
(1) P.NOME COUNT(*)
<<FROM>> PESQUISADOR:P  ESCREVE:E
<<WHERE/HAVING>> COUNT(A.TITULO) > 3

DML - Consultando Dados em Tabelas

▶ Tipos de Subconsulta

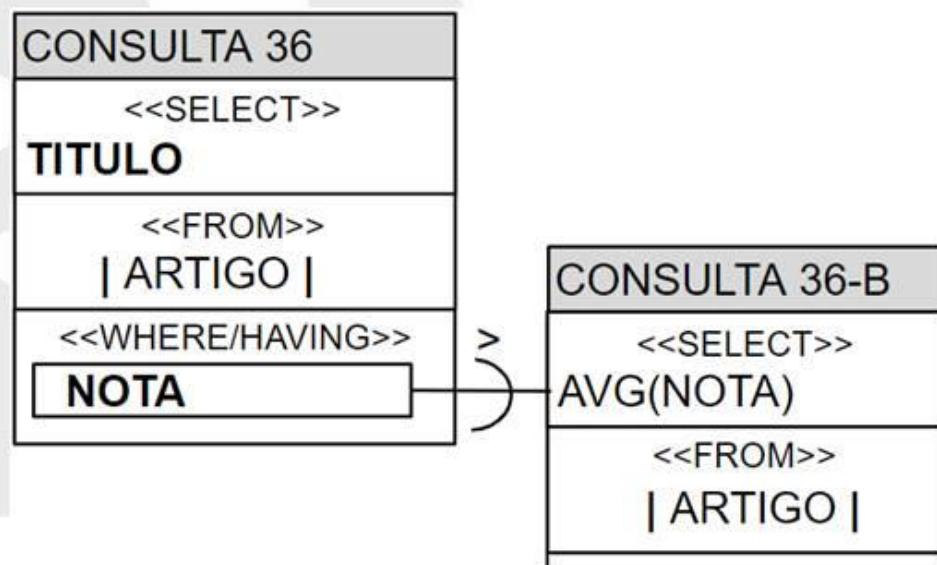
- ▶ Quanto a quantidade de linhas e colunas retornadas
 - ▶ ESCALAR → Retornam um único valor (i.e., uma única linha e coluna)
 - ▶ LINHA → Retornam várias colunas, mas apenas uma linha
 - ▶ TABELA → Retornam uma ou mais colunas e múltiplas linhas
- ▶ Quanto a dependências entre as subconsultas
 - ▶ SIMPLES
 - ▶ CORRELACIONADA
 - SEMI JOIN
 - ANTI JOIN (ou ANTI SEMI JOIN – negação do SEMI JOIN)

DML - Consultando Dados em Tabelas

▶ Exemplo (subconsulta simples e escalar)

```
/* CONSULTA 36 - PROJETAR OS TÍTULOS DOS ARTIGOS COM NOTA  
ACIMA DA MÉDIA */
```

```
SELECT TITULO  
FROM ARTIGO  
WHERE NOTA >  
    (SELECT AVG (NOTA)  
     FROM ARTIGO) ;
```

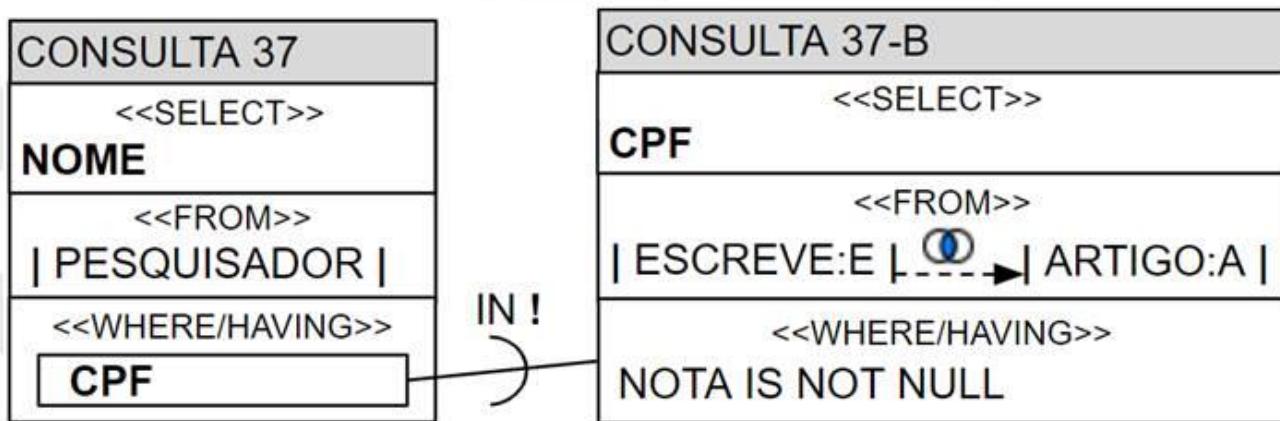


DML - Consultando Dados em Tabelas

► Exemplo (subconsulta simples e tabela)

```
/* CONSULTA 37 - PROJETAR OS NOMES DOS PESQUISADORES QUE POSSUEM  
ARTIGOS SEM NOTA - USANDO “IN” E “INNER JOIN”) */
```

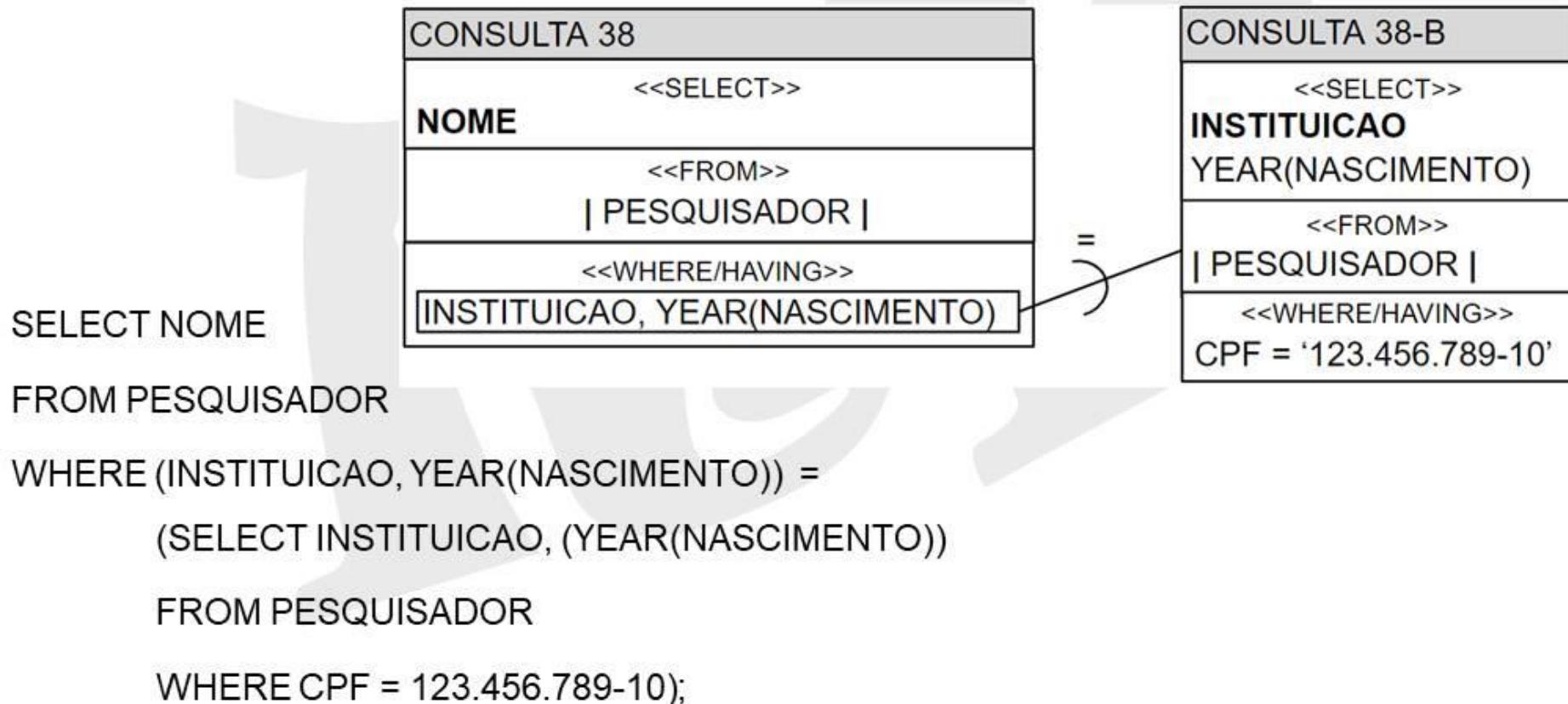
```
SELECT NOME  
FROM PESQUISADOR  
WHERE CPF NOT IN  
(SELECT CPF  
FROM ESCREVE E INNER JOIN  
ARTIGO A ON E.MAT = A.MAT  
WHERE NOTA IS NOT NULL);
```



DML - Consultando Dados em Tabelas

▶ Exemplo (subconsulta simples e linha)

```
/* CONSULTA 38 - PROJETAR OS NOMES DOS PESQUISADORES QUE NASCERAM NO MESMO  
ANO E TRABALHAM NA MESMA INSTITUIÇÃO DO PESQUISADOR 123.456.789-10 */
```

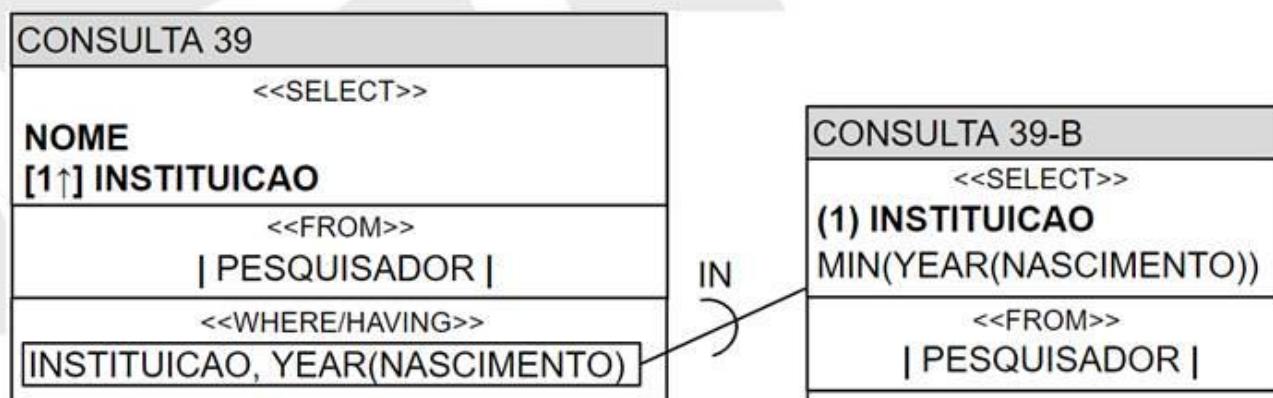


DML - Consultando Dados em Tabelas

▶ Exemplo (subconsulta simples e tabela)

```
/* CONSULTA 39 - POR INSTITUIÇÃO, PROJETAR O NOME E A INSTITUIÇÃO DOS  
PESQUISADORES MAIS VELHOS */
```

```
SELECT NOME, INSTITUICAO  
FROM PESQUISADOR  
WHERE (INSTITUICAO, YEAR(NASCIMENTO)) IN  
(SELECT INSTITUICAO, MIN(YEAR(NASCIMENTO))  
FROM PESQUISADOR  
GROUP BY INSTITUICAO)  
ORDER BY INSTITUICAO;
```

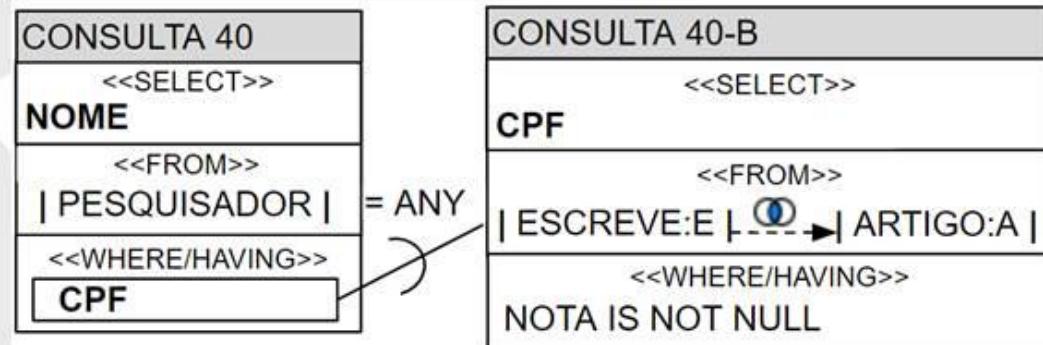


DML - Consultando Dados em Tabelas

▶ Exemplo (subconsulta simples e tabela)

/* CONSULTA 40 - PROJETAR OS NOMES DOS PESQUISADORES QUE POSSUEM ARTIGOS SEM NOTA - USANDO “= ANY”, pois é equivalente ao “IN” */

```
SELECT NOME  
FROM PESQUISADOR  
WHERE CPF = ANY  
(SELECT CPF  
FROM ESCREVE E INNER JOIN  
    ARTIGO A ON E.MAT = A.MAT  
WHERE NOTA IS NULL);
```

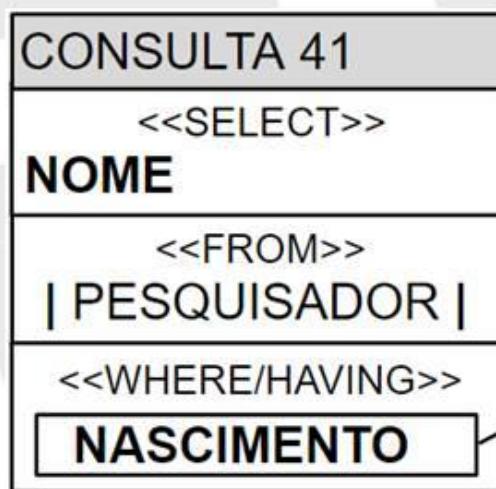


DML - Consultando Dados em Tabelas

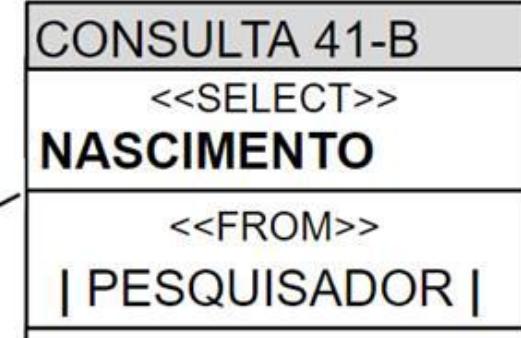
▶ Exemplo (subconsulta simples e tabela)

```
/* CONSULTA 41 - PROJETAR OS NOMES DOS PESQUISADORES, EXCETO O  
DO MAIS VELHO */
```

```
SELECT NOME  
FROM PESQUISADOR  
WHERE NASCIMENTO > ANY  
  (SELECT NASCIMENTO  
   FROM PESQUISADOR)
```



> ANY

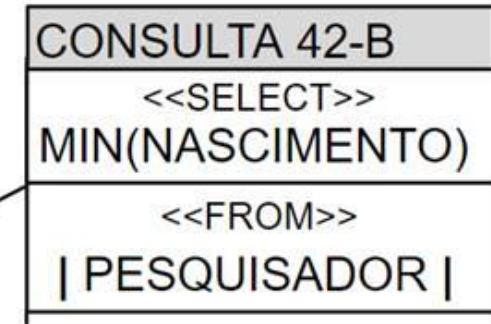
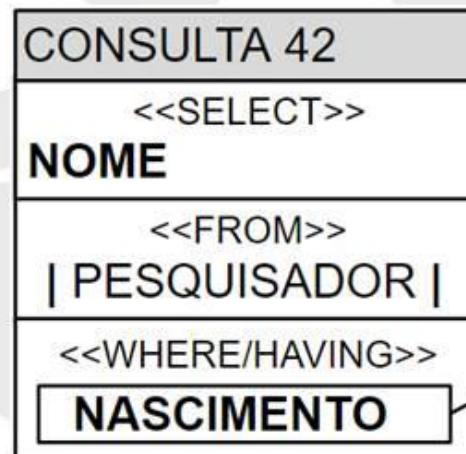


DML - Consultando Dados em Tabelas

▶ Exemplo (subconsulta simples e escalar)

```
/* CONSULTA 42 - PROJETAR OS NOMES DOS PESQUISADORES, EXCETO O  
DO MAIS IDOSO */
```

```
SELECT NOME  
FROM PESQUISADOR  
WHERE NASCIMENTO <>  
  (SELECT MIN(NASCIMENTO)  
   FROM PESQUISADOR)
```



DML - Consultando Dados em Tabelas

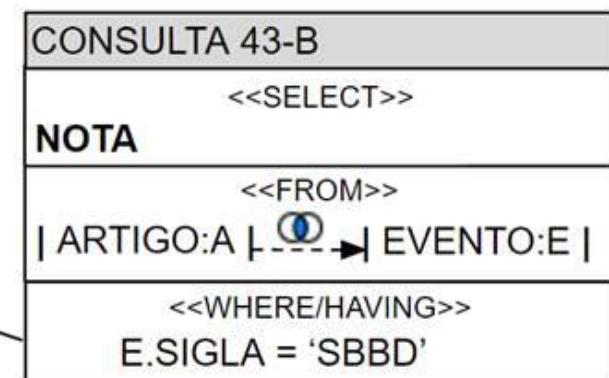
▶ Exemplo (subconsulta simples e tabela)

```
/* CONSULTA 43 - PROJETAR OS TÍTULOS DOS ARTIGOS COM NOTA MAIOR DO  
QUE TODOS OS ARTIGOS DO EVENTO 'SBBD' */
```

```
SELECT TITULO  
FROM ARTIGO  
WHERE NOTA > ALL  
(SELECT NOTA  
FROM ARTIGO A INNER JOIN  
EVENTO E A.COD = E.COD  
WHERE E.SIGLA = 'SBBD');
```



> ALL

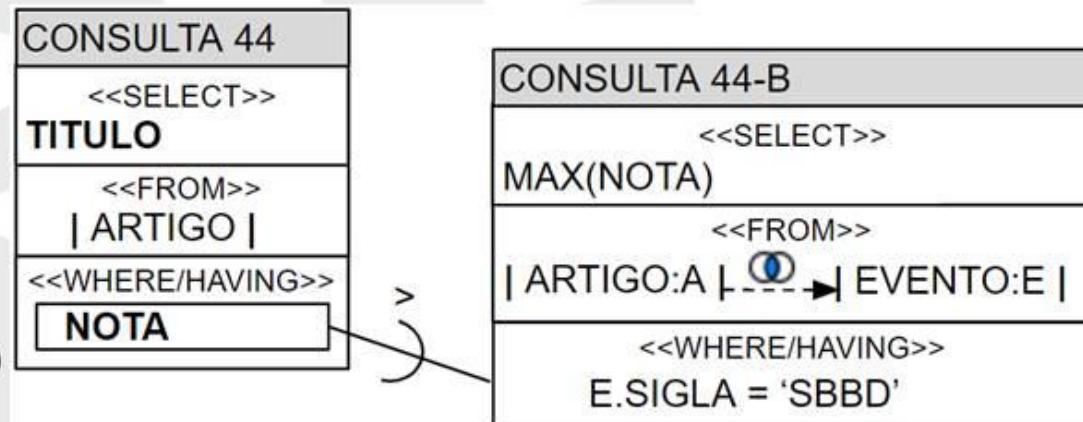


DML - Consultando Dados em Tabelas

- ▶ Exemplo (subconsulta simples e escalar)

```
/* CONSULTA 44 - PROJETAR OS TÍTULOS DOS ARTIGOS COM NOTA MAIOR DO  
QUE TODOS OS ARTIGOS DO EVENTO 'SBBD' */
```

```
SELECT TITULO  
FROM ARTIGO  
WHERE NOTA >  
(SELECT MAX(NOTA)  
FROM ARTIGO A INNER JOIN  
    EVENTO E A.COD = E.COD  
WHERE E.SIGLA = 'SBBD');
```

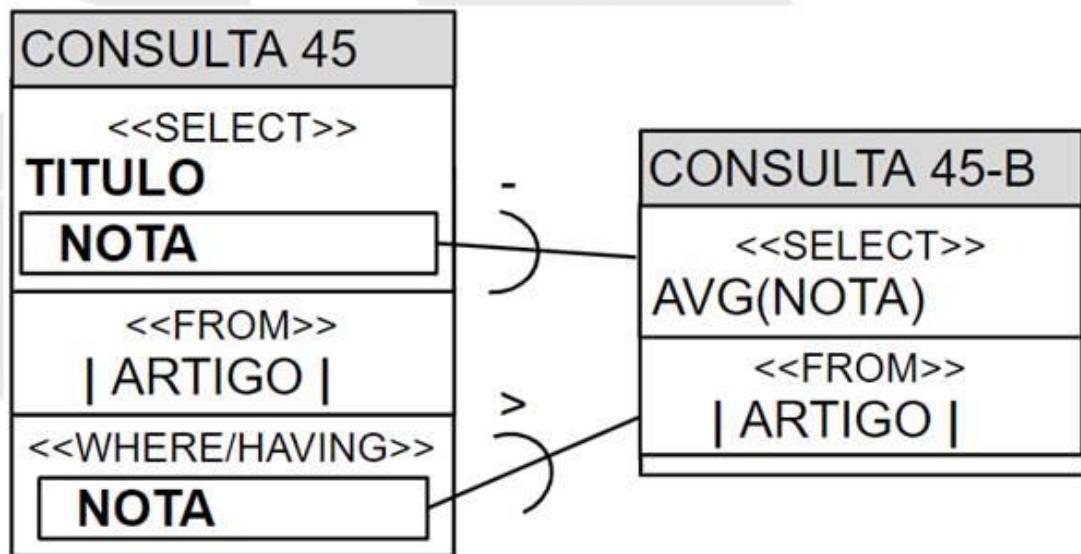


DML - Consultando Dados em Tabelas

- ▶ Exemplo (subconsultas simples e escalares – no SELECT/WHERE)

```
/* CONSULTA 45 - PROJETAR OS TÍTULOS DOS ARTIGOS COM NOTA ACIMA DA  
MÉDIA GERAL, MOSTRANDO QUANTO A NOTA ESTÁ ACIMA DA MÉDIA */
```

```
SELECT TÍTULO, NOTA – (SELECT AVG(A.NOTA) FROM ARTIGO A)  
FROM ARTIGO  
WHERE NOTA >  
( SELECT AVG(A.NOTA)  
FROM ARTIGO A);
```

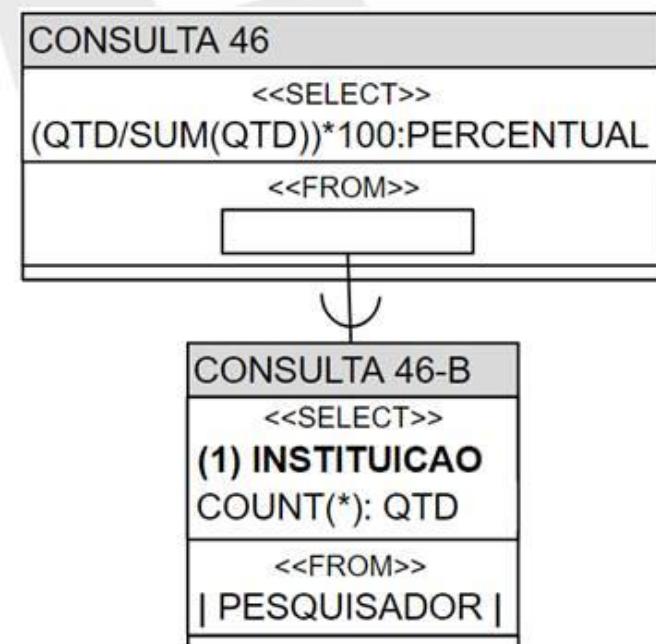


DML - Consultando Dados em Tabelas

- ▶ Exemplo (subconsulta simples e tabela – no FROM)

```
/* CONSULTA 46 - PROJETAR O PERCENTUAL DE PESQUISADORES POR  
INSTITUIÇÃO */
```

```
SELECT INSTITUICAO, (QTD/SUM(QTD))*100 AS PERCENTUAL  
  
FROM  
  
(SELECT INSTITUICAO, COUNT(*) AS QTD  
  
FROM PESQUISADOR  
  
GROUP BY INSTITUICAO);
```

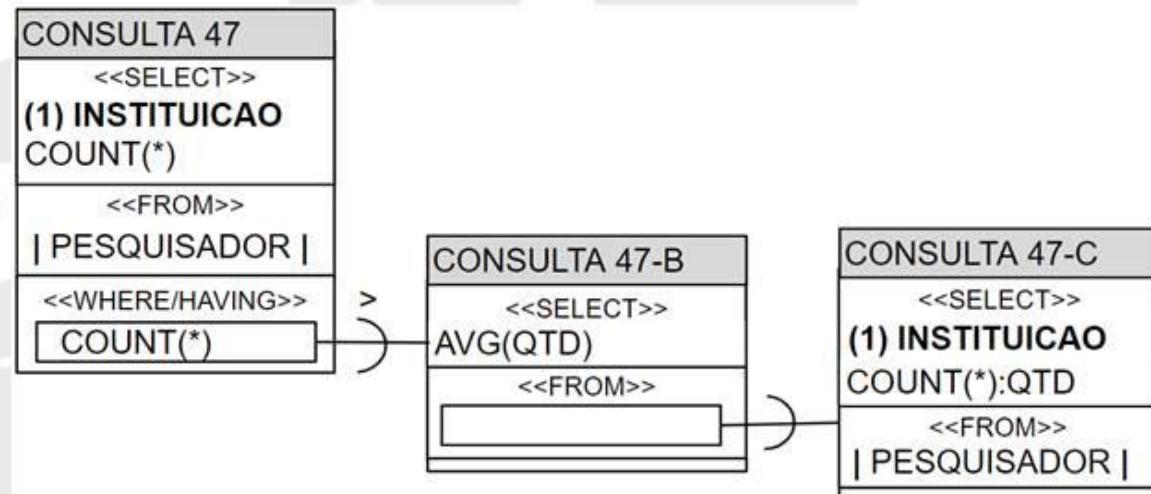


DML - Consultando Dados em Tabelas

- ▶ Exemplo (subconsulta simples e tabela – no HAVING)

/* CONSULTA 47 - PROJETAR QUAIS SÃO AS INSTITUIÇÕES QUE TÊM MAIS PESQUISADORES DO QUE A MÉDIA */

```
SELECT INSTITUICAO, COUNT(*)  
FROM PESQUISADOR  
GROUP BY INSTITUICAO  
HAVING COUNT(*) >  
( SELECT AVG(QTD)  
    FROM (SELECT INSTITUICAO, COUNT (*) AS QTD  
          FROM PESQUISADOR  
          GROUP BY INSTITUICAO))
```

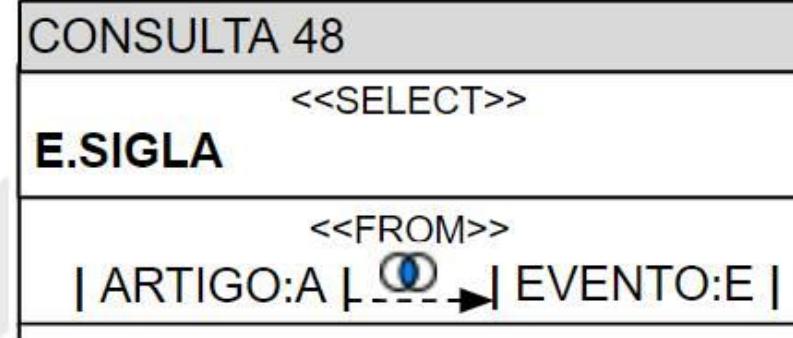


DML - Consultando Dados em Tabelas

- ▶ Exemplo (consulta com junção interna)

```
/* CONSULTA 48 - PROJETAR AS SIGLAS DOS EVENTOS COM ARTIGOS  
PUBLICADOS
```

```
SELECT E.SIGLA  
FROM ARTIGO A INNER JOIN  
      EVENTO E A.COD = E.COD;
```

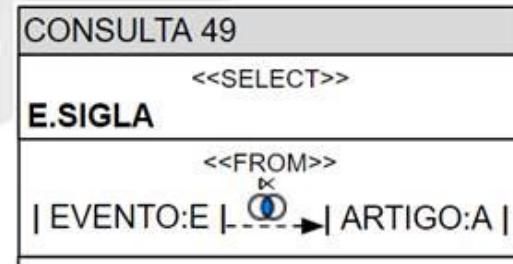
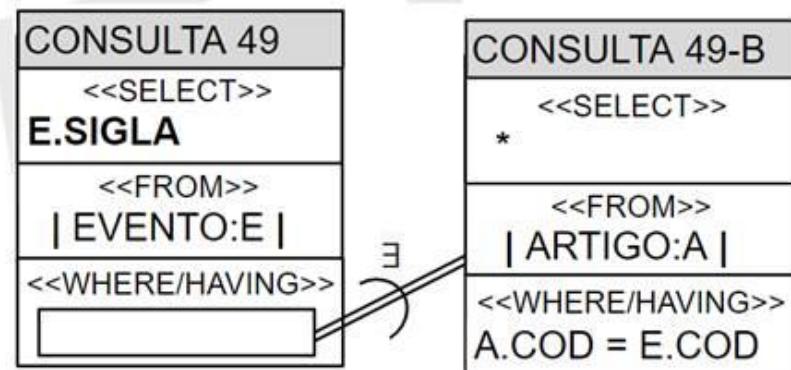


DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (semi join a partir de subconsulta correlacionada e escalar)

```
/* CONSULTA 49 - PROJETAR AS SIGLAS DOS  
EVENTOS COM ARTIGOS PUBLICADOS  
- USANDO “EXISTS” */
```

```
SELECT E.SIGLA  
  
FROM EVENTO E  
  
WHERE EXISTS  
  
(SELECT *  
  
FROM ARTIGO A  
  
WHERE A.COD = E.COD);
```



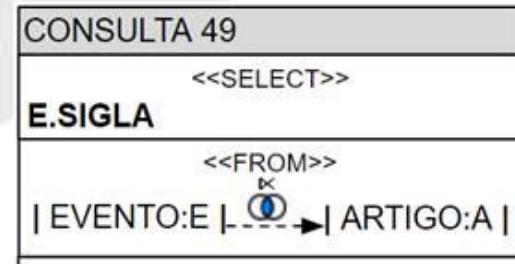
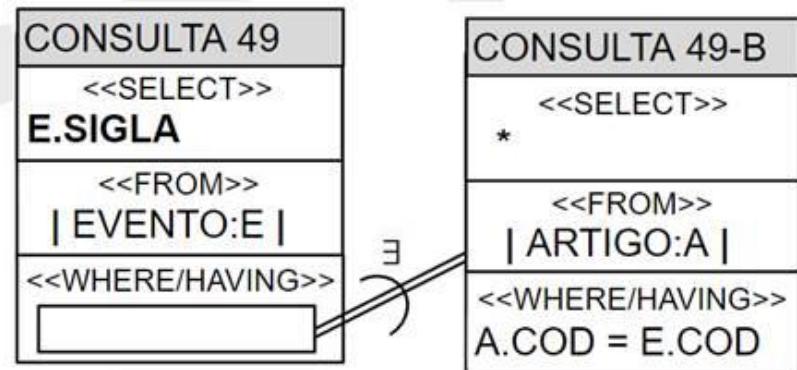
Notação alternativa

DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (semi join a partir de subconsulta correlacionada e escalar)

ATENÇÃO:

SEMI JOIN = exibir os dados da tabela a esquerda a partir de uma comparação entre duas tabelas até encontrar o primeiro resultado verdadeiro. Isto é, **não há produto cartesiano, tampouco resultados repetidos.**



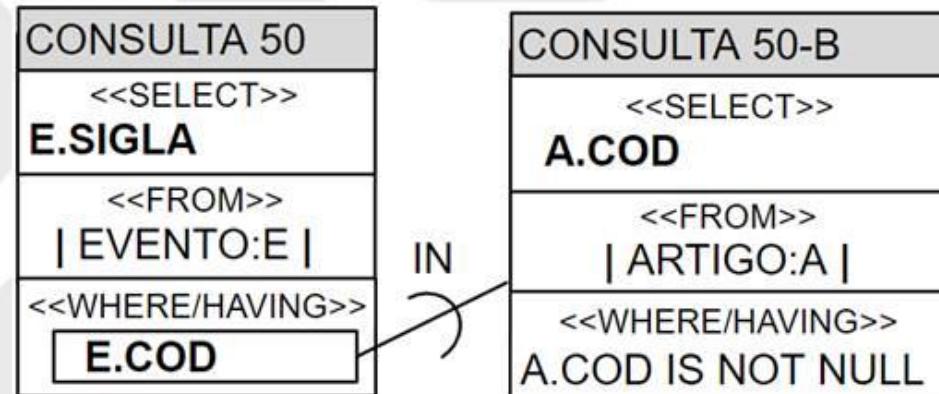
Notação alternativa

DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (semi join a partir de subconsulta simples e tabela)

```
/* CONSULTA 50 - PROJETAR AS SIGLAS DOS EVENTOS COM  
ARTIGOS PUBLICADOS – USANDO “IN” */
```

```
SELECT E.SIGLA  
FROM EVENTO E  
WHERE E.COD IN  
(SELECT A.COD  
FROM ARTIGO A  
WHERE A.COD IS NOT NULL);
```



ATENÇÃO:

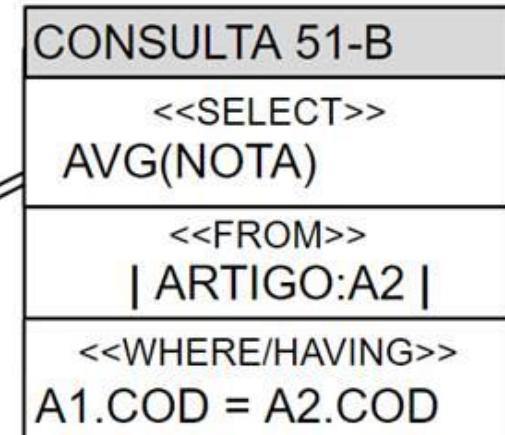
Subconsultas com EXIST usam índice,
por isso são mais performática do que com IN.

DML - Consultando Dados em Tabelas

▶ Exemplo (consulta correlacionada)

```
/* CONSULTA 51 - PROJETAR OS TÍTULOS DOS ARTIGOS COM NOTA ABAIXO DA MÉDIA  
DOS ARTIGOS PUBLICADOS NO MESMO EVENTO*/
```

```
SELECT A1.TÍTULO  
FROM ARTIGO A1  
WHERE A1.NOTA <  
    ( SELECT AVG(NOTA))  
    FROM ARTIGO A2  
    WHERE A1.COD = A2.COD);
```



DML - Consultando Dados em Tabelas

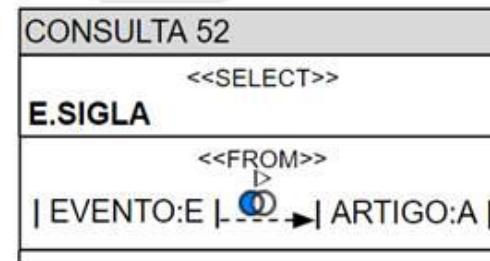
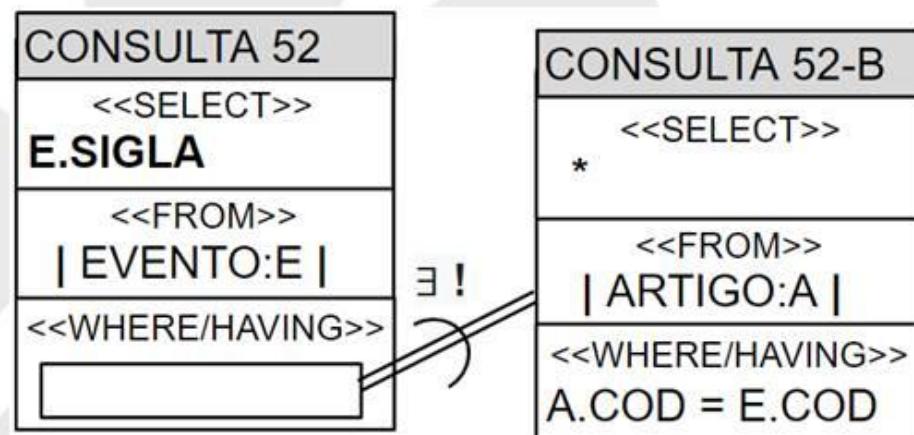
- ▶ **Exemplo** (negação do semi join ou anti join)

/* CONSULTA 52 - PROJETAR AS SIGLAS DOS EVENTOS **SEM ARTIGOS PUBLICADOS** –
USANDO “NOT EXISTS” */

```
SELECT E.SIGLA  
FROM EVENTO E  
WHERE NOT EXISTS  
(SELECT *  
  FROM ARTIGO A  
 WHERE A.COD = E.COD);
```

ATENÇÃO:

É possível fazer com LEFT JOIN +
FK IS NULL, mas não é tão performático
quanto o ANTI JOIN.



Notação alternativa

DML - Consultando Dados em Tabelas

- ▶ **Exemplo** (negação do semi join ou anti join)

/* CONSULTA 53 - PROJETAR AS SIGLAS DOS EVENTOS SEM ARTIGOS PUBLICADOS – USANDO “NOT IN” */

```
SELECT E.SIGLA  
FROM EVENTO E  
WHERE E.COD NOT IN  
    (SELECT A.COD  
     FROM ARTIGO A  
     WHERE A.COD IS NOT NULL);
```

CONSULTA 53
 <<SELECT>>
 E.SIGLA
 <<FROM>>
 | EVENTO:E |
 <<WHERE/HAVING>>
 | E.COD |

CONSULTA 53-B
 <<SELECT>>
 A.COD
 <<FROM>>
 | ARTIGO:A |
 <<WHERE/HAVING>>
 | A.COD IS NOT NULL |

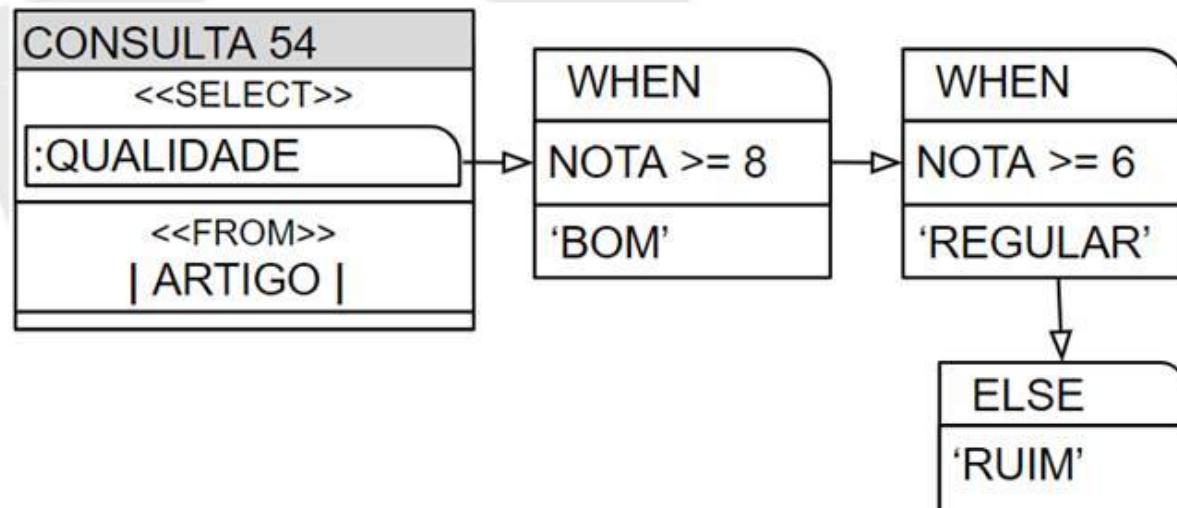
IN !

DML - Consultando Dados em Tabelas

▶ Exemplo (avaliação condicional)

-- CONSULTA 54 - PROJETAR A QUALIDADE DO ARTIGO

```
SELECT  
CASE  
    WHEN NOTA >= 8 THEN 'BOM'  
    WHEN NOTA >= 6 THEN 'REGULAR'  
    ELSE 'RUIM'  
END AS QUALIDADE  
FROM ARTIGO
```

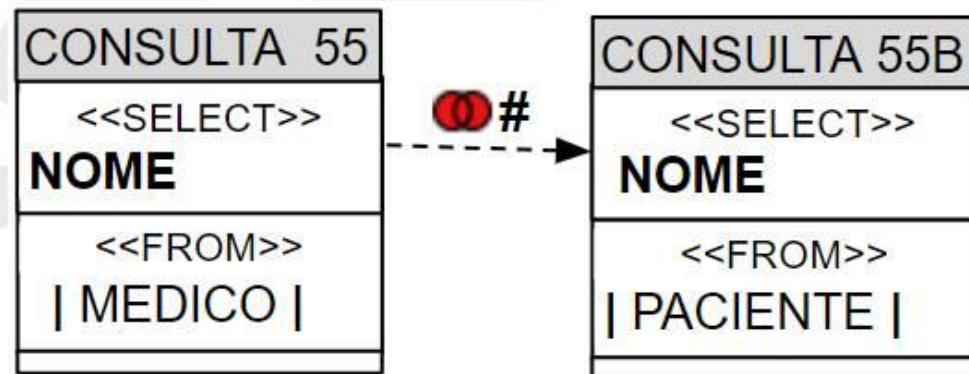


DML - Consultando Dados em Tabelas

- Considere:
 - MEDICO (CODMEDICO, NOME, CRM)
 - PACIENTE (CODPACIENTE, NOME)
- UNION

```
/* CONSULTA 55 - PROJETAR O NOME DE TODAS  
AS PESSOAS CADASTRADAS NO HOSPITAL */
```

```
(SELECT NOME  
FROM MEDICO)  
  
UNION  
  
(SELECT NOME  
FROM PACIENTE);
```



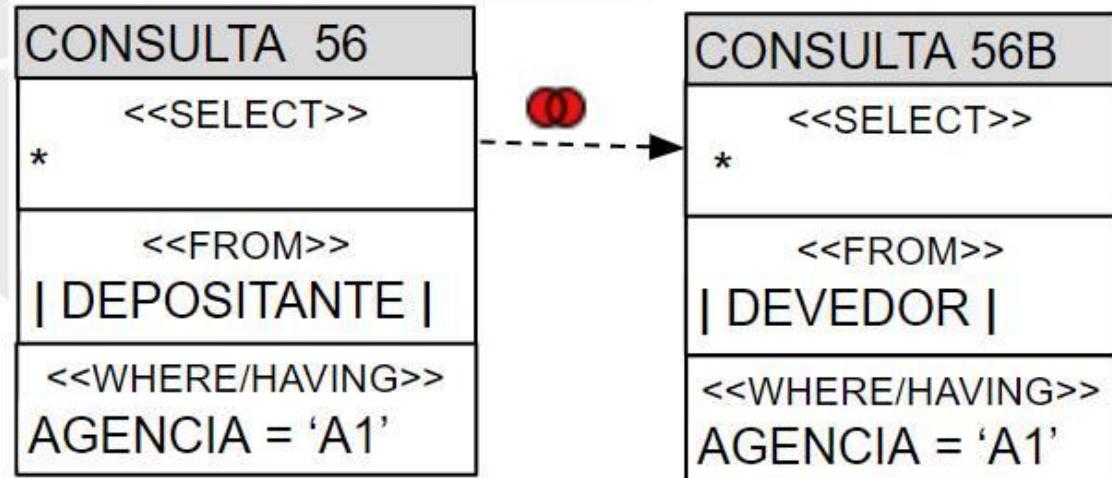
DML - Consultando Dados em Tabelas

- Considere:

- DEPOSITANTE (CPF, NOME, AGENCIA, CONTA)
- DEVEDOR (CPF, NOME, AGENCIA, CONTA)

- UNION

```
/* CONSULTA 56 - Projetar todos os clientes  
da agência A1 com empréstimo ou depósito */  
  
(SELECT *  
FROM DEPOSITANTE  
WHERE AGENCIA = 'A1')  
  
UNION ALL  
  
(SELECT *  
FROM DEVEDOR  
WHERE AGENCIA = 'A1');
```



DML - Consultando Dados em Tabelas

- Considere:
 - MEDICO (CODMEDICO, NOME, CRM)
 - PACIENTE (CODPACIENTE, NOME)
- **INTERSECT**

```
/* CONSULTA 57- PROJETAR O NOME DE TODAS AS PESSOAS QUE SÃO  
MÉDICOS E PACIENTES AO MESMO TEMPO */
```

```
(SELECT NOME  
FROM MEDICO)  
  
INTERSECT  
  
(SELECT NOME  
FROM PACIENTE);
```

CONSULTA 57	
<<SELECT>>	NOME
<<FROM>>	MEDICO

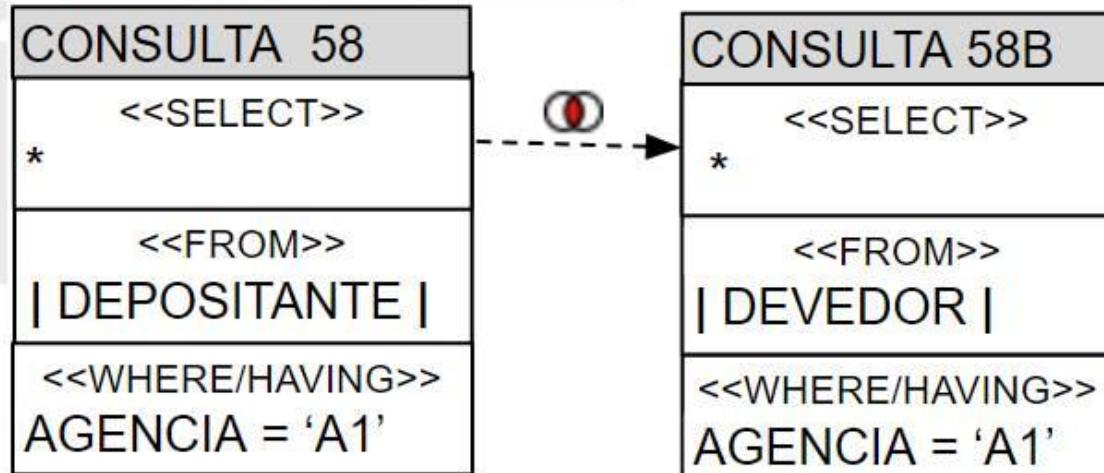
CONSULTA 57B	
<<SELECT>>	NOME
<<FROM>>	PACIENTE

DML - Consultando Dados em Tabelas

- Considere:
 - DEPOSITANTE (CPF, NOME, AGENCIA, CONTA)
 - DEVEDOR (CPF, NOME, AGENCIA, CONTA)
- **INTERSECT**

```
(SELECT *  
FROM DEPOSITANTE  
WHERE AGENCIA = 'A1')  
  
INTERSECT  
  
(SELECT *  
FROM DEVEDOR  
WHERE AGENCIA = 'A1');
```

**/* CONSULTA 58 - PROJETAR TODOS OS CLIENTES
DA AGÊNCIA A1 COM EMPRÉSTIMO E DEPÓSITO**

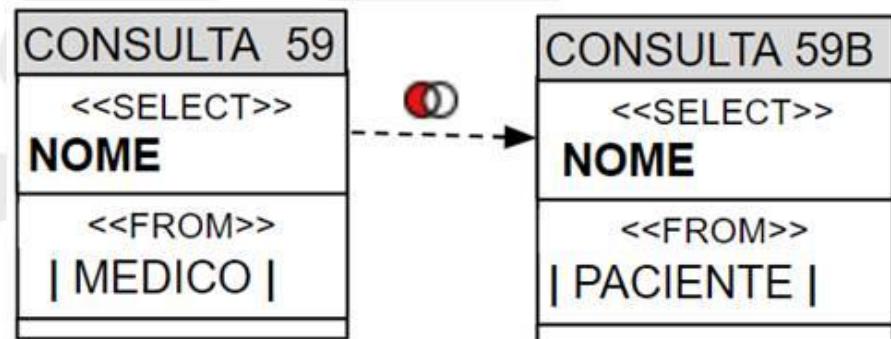


DML - Consultando Dados em Tabelas

- Considere:
 - MEDICO (CODMEDICO, NOME, CRM)
 - PACIENTE (CODPACIENTE, NOME)
- **EXCEPT**

```
/* CONSULTA 59 - PROJETAR O NOME DE TODAS AS  
PESSOAS QUE SÃO MÉDICOS E NÃO SÃO PACIENTES */
```

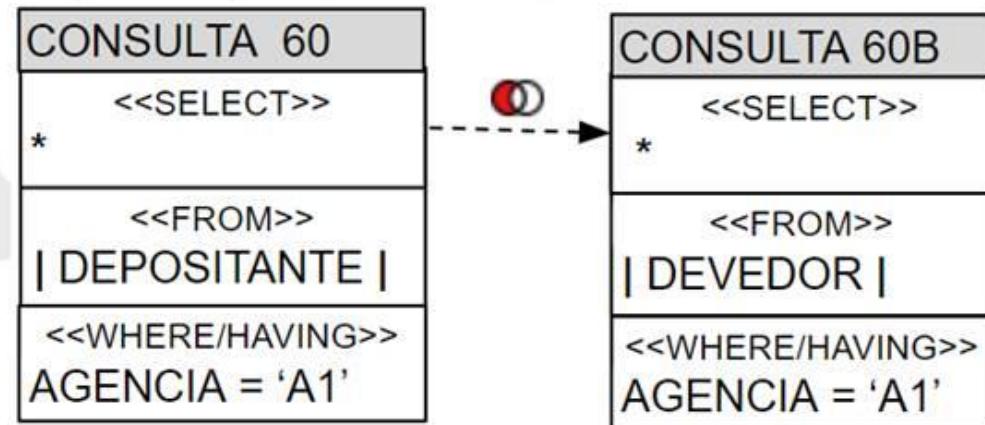
```
(SELECT NOME  
FROM MEDICO)  
  
EXCEPT  
(SELECT NOME  
FROM PACIENTE);
```



DML - Consultando Dados em Tabelas

- Considere:
 - DEPOSITANTE (CPF, NOME, AGENCIA, CONTA) E
 - DEVEDOR (CPF, NOME, AGENCIA, CONTA).
- **EXCEPT**

```
/* CONSULTA 60 - PROJETAR TODOS OS CLIENTES  
DA AGÊNCIA A1 COM CONTA E SEM EMPRÉSTIMO */  
  
(SELECT *  
FROM DEPOSITANTE  
WHERE AGENCIA = 'A1')  
  
EXCEPT  
  
(SELECT *  
FROM DEVEDOR  
WHERE AGENCIA = 'A1');
```



DML - Consultando Dados em Tabelas

- Considere:
 - MEDICO (CODMEDICO, NOME, CRM) E
 - PACIENTE (CODPACIENTE, NOME);
- **UNION EXCLUSIVA (OU COMPLEMENTO)**

```
((SELECT NOME          /* CONSULTA 61 - PROJETAR O NOME DE TODAS AS
   FROM MEDICO)           PESSOAS QUE APENAS SÃO MÉDICO OU PACIENTE */
UNION
  (SELECT NOME
   FROM PACIENTE))
EXCEPT
  ((SELECT NOME
   FROM MEDICO)
INTERSECT
  (SELECT NOME
   FROM PACIENTE));
```

```
CONSULTA 61
  <<SELECT>>
  NOME
  <<FROM>>
  | MEDICO |

CONSULTA 61B
  <<SELECT>>
  NOME
  <<FROM>>
  | PACIENTE |

CONSULTA 61C
  <<SELECT>>
  NOME
  <<FROM>>
  | MEDICO |

CONSULTA 61D
  <<SELECT>>
  NOME
  <<FROM>>
  | PACIENTE |
```

DML - Consultando Dados em Tabelas

- Considere:
 - MEDICO (CodMedico, Nome, CRM) e
 - PACIENTE (CodPaciente, Nome);
- **UNION EXCLUSIVA (OU COMPLEMENTO)**

```
SELECT NOME  
FROM MEDICO          /* CONSULTA 61 - PROJETAR O NOME DE TODAS AS  
WHERE NOME NOT IN      PESSOAS QUE APENAS SÃO MÉDICO OU PACIENTE */  
(SELECT NOME  
     FROM PACIENTE)  
  
UNION  
      SELECT NOME  
      FROM MEDICO  
      WHERE NOME NOT IN  
            (SELECT NOME  
                 FROM PACIENTE);
```

CONSULTA 61

<<SELECT>>
NOME
<<FROM>>
MEDICO

CONSULTA 61B

<<SELECT>>
NOME
<<FROM>>
PACIENTE

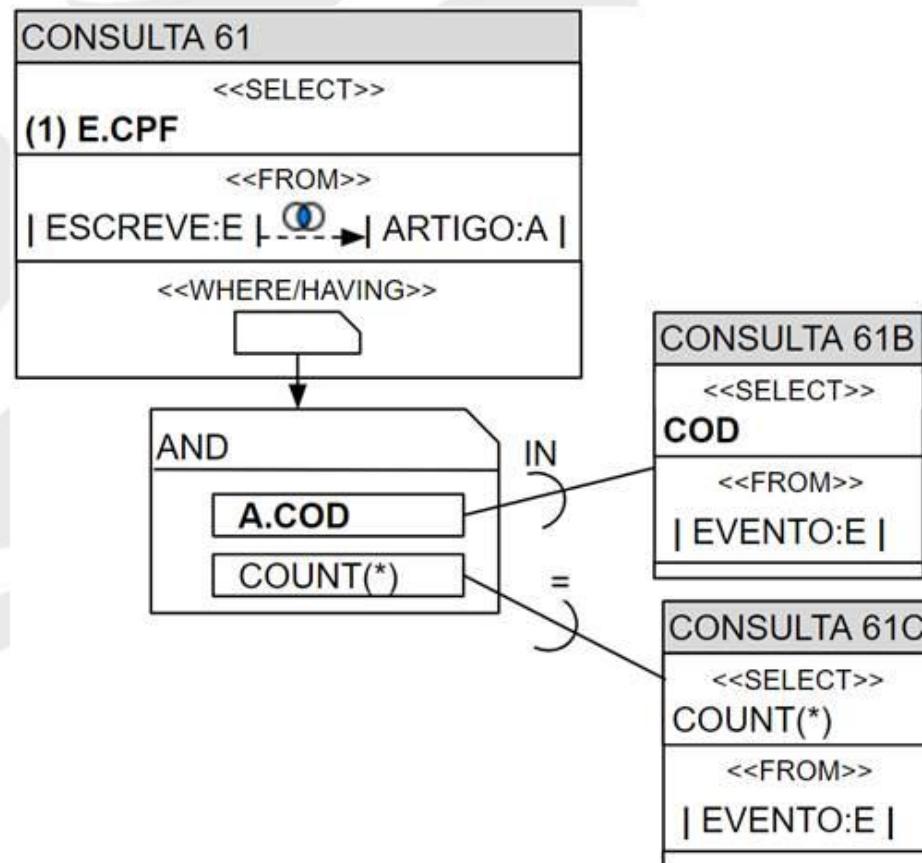
Notação alternativa

DML - Consultando Dados em Tabelas

- **DIVISION**

```
/* PROJETAR OS PROFESSORES QUE  
PUBLICARAM EM TODOS OS EVENTOS */
```

```
SELECT E.CPF  
FROM ESCREVE E INNER JOIN  
      ARTIGO A ON (E.MAT = A.MAT)  
WHERE A.COD IN  
      (SELECT COD  
       FROM EVENTO)  
GROUP BY E.CPF  
HAVING COUNT(*) =  
      (SELECT COUNT(*)  
       FROM EVENTO)
```



DML - Consultando Dados em Tabelas

• DIVISION

```
/* PROJETAR OS PROFESSORES QUE  
PUBLICARAM EM TODOS OS EVENTOS */
```

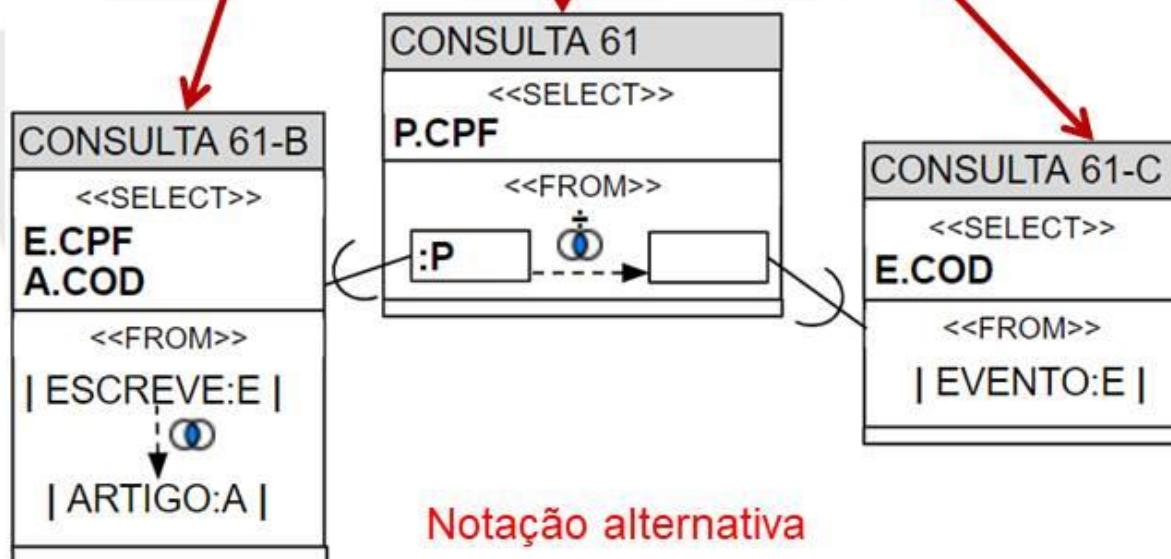
```
SELECT E.CPF  
FROM ESCREVE E INNER JOIN  
    ARTIGO A ON (E.MAT = A.MAT)
```

```
WHERE A.COD IN  
(SELECT COD  
FROM EVENTO)
```

```
GROUP BY E.CPF  
HAVING COUNT(*) =  
(SELECT COUNT(*)  
FROM EVENTO)
```

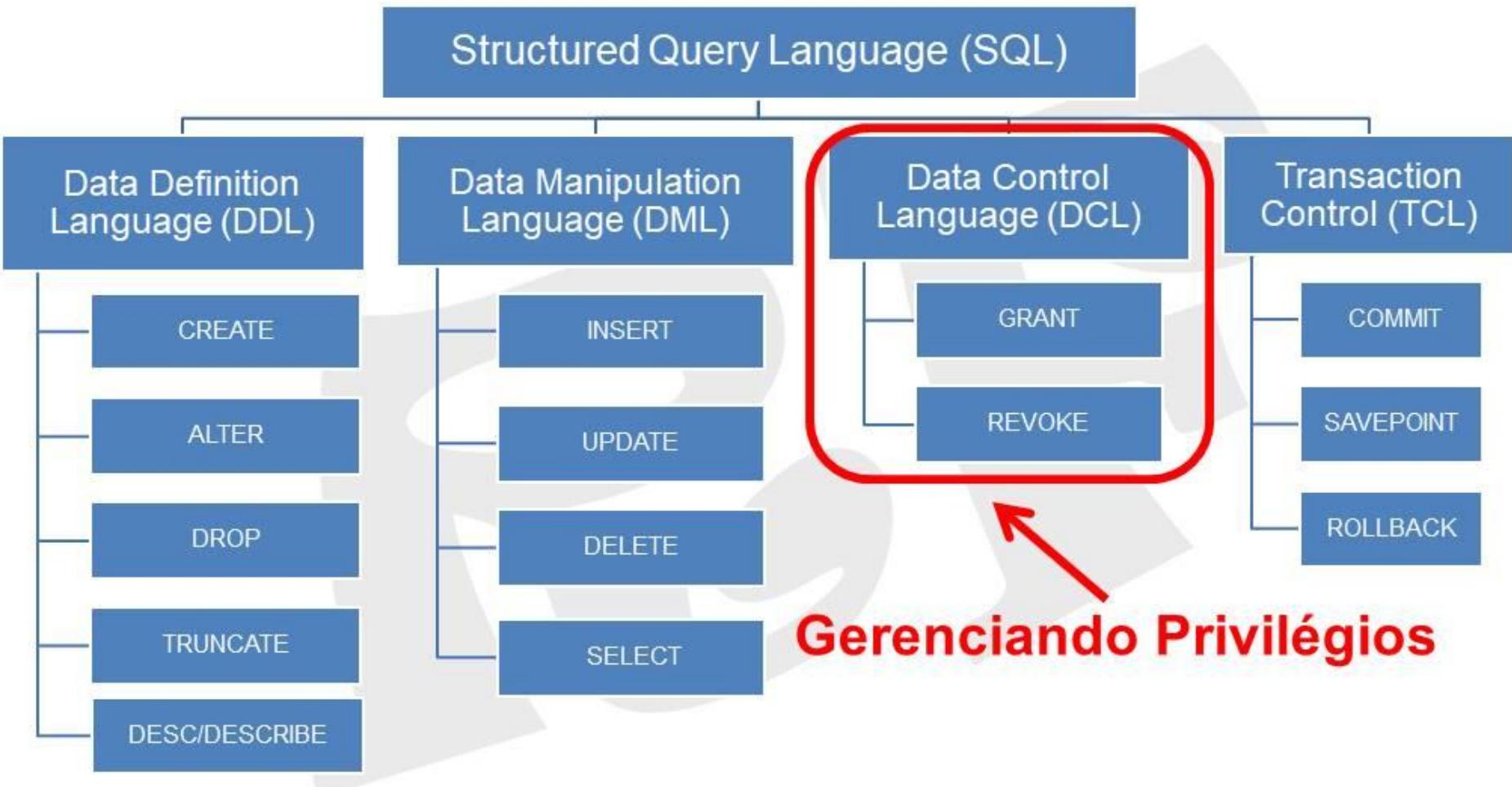
Lembrando: $R(X) = R(A) \div R(B)$, onde:

- 1) $B \subseteq A$
- 2) $X = A - B$
- 3) $R_1(A) \rightarrow \text{Dividendo}$
- 4) $R_2(B) \rightarrow \text{Divisor}$
- 5) $R(X) \rightarrow \text{Quociente}$



Notação alternativa

Linguagens para SGBDR – Comandos Básicos



Gerenciando Privilégios

DCL – Concedendo Privilégios

Exemplos:

```
GRANT CONNECT TO PAULO, PEDRO, ANA
```

```
GRANT SELECT ON ARTIGO, PESQUISADOR TO PAULO, PEDRO;
```

```
GRANT SELECT, INSERT, UPDATE ON ARTIGO TO ANA;
```

```
GRANT ALL PRIVILEGES ON EVENTO TO PUBLIC;
```

```
GRANT CREATE TABLE, CREATE VIEW TO GERENCIA;
```

```
GRANT GERENCIA TO PUBLIC;
```

DDL – Revogando Privilégios

- Exemplos:

```
REVOKE CONNECT FROM PAULO, PEDRO, ANA
```

```
REVOKE SELECT ON ARTIGO, PESQUISADOR FROM PAULO, PEDRO;
```

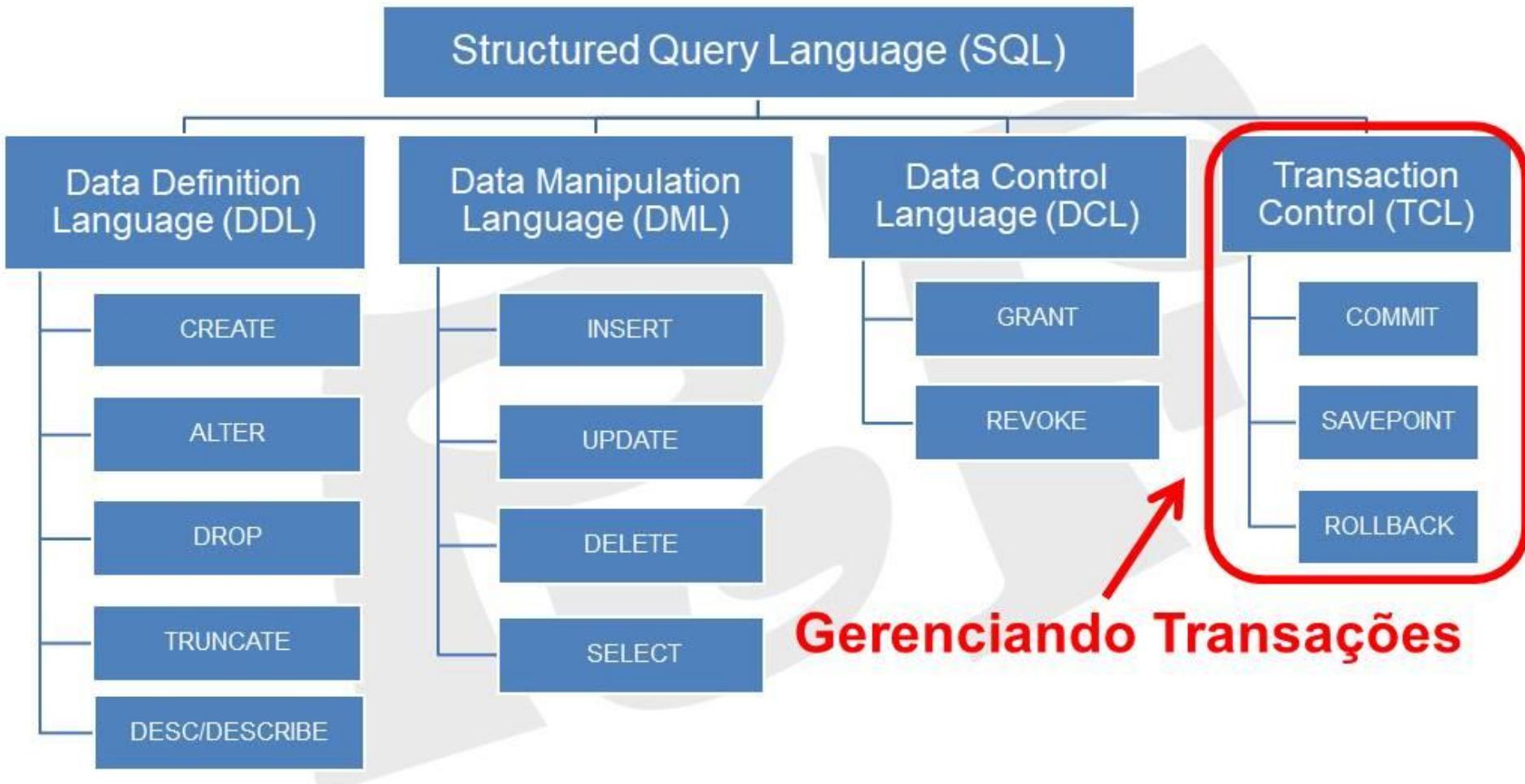
```
REVOKE SELECT, INSERT, UPDATE ON ARTIGO FROM ANA;
```

```
REVOKE ALL PRIVILEGES ON EVENTO FROM PUBLIC;
```

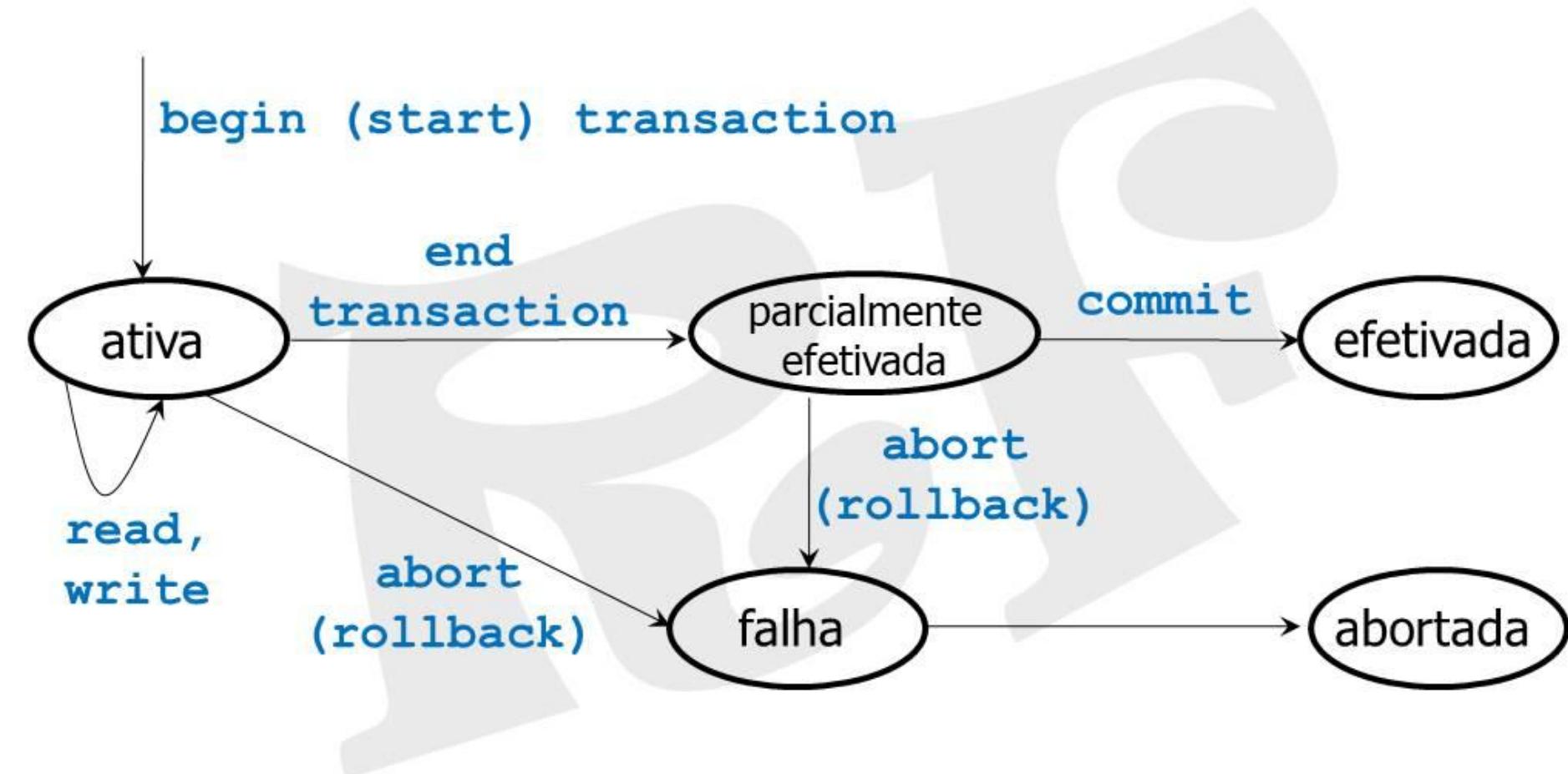
```
REVOKE CREATE TABLE, CREATE VIEW FROM GERENCIA;
```

```
REVOKE GERENCIA FROM PUBLIC;
```

Linguagens para SGBDR – Comandos Básicos



Transações – Conceitos Básicos



Transações – Conceitos Básicos

▶ Propriedades ACID

- ▶ **Atomicidade:** todas as operações de uma transação devem ser efetivadas; ou, na ocorrência de uma falha, nada deve ser efetivado
 - ▶ “tudo ou nada” – não se admite parte de uma operação
- ▶ **Consistência:** transações preservam a consistência da base
 - ▶ Estado inicial consistente \Rightarrow Estado final consistente
- ▶ **Isolamento:** a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- ▶ **Durabilidade:** uma vez consolidada (committed) a transação, suas alterações permanecem no banco até que outras transações aconteçam

Commit, Savepoint e Rollback

Exemplo

```
UPDATE ARTIGO SET NOTA = 10 WHERE MAT = '123';
COMMIT; -- ou COMMIT TRANSACTION;
UPDATE ARTIGO SET NOTA = 5 WHERE MAT = '123';
ROLLBACK; -- ou ROLLBACK TRANSACTION;
```

```
SELECT *
FROM ARTIGOS
WHERE MAT = '123';
```

```
INÍCIO_BLOCO
LAÇO;
    UPDATE, INSERT, DELETE ...
    COMMIT; -- ou COMMIT TRANSACTION;
FIM LAÇO;
FIM_BLOCO;
```

Commit, Savepoint e Rollback

Exemplo

```
INICIO_BLOCO
```

```
...
```

```
SAVEPOINT MEU_SAVEPOINT;
```

```
-- ou SAVE TRANSACTION MEU_SAVEPOINT;
```

```
UPDATE ...
```

```
DELETE ...
```

```
SAVEPOINT MEU_SAVEPOINT;
```

```
-- ou SAVE TRANSACTION MEU_SAVEPOINT;
```

```
INSERT ...
```

```
ROLLBACK TO MEU_SAVEPOINT;
```

```
-- ou ROLL BACK TRANSACTION MEU_SAVEPOINT;
```

```
FIM_BLOCO;
```