

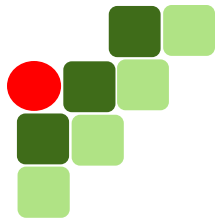


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Araraquara

Banco de Dados II

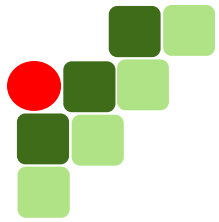
Cristiane Yaguinuma
cristiane.yaguinuma@ifsp.edu.br

► Tipos de JOIN



Roteiro da aula

- ▶ Tipos de JOIN
 - Variações de INNER JOIN
 - NATURAL JOIN
 - CROSS JOIN



Sintaxe padrão SQL:1999

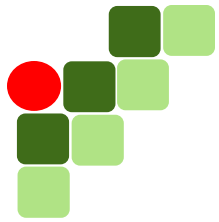
- ▶ Sintaxe SQL padrão definida pela American National Standards Institute (ANSI) em 1999
- ▶ Define o uso do comando JOIN e suas variantes
 - INNER JOIN
 - NATURAL JOIN
 - CROSS JOIN
 - OUTER JOIN



INNER JOIN

- ▶ Conhecido também como JOIN, EQUIJOIN ou junções simples
 - Envolve comparação de igualdade entre atributos chave primária e chave estrangeira
- ▶ Ex.: Recuperar sobrenome dos empregados e o nome do departamento onde trabalham

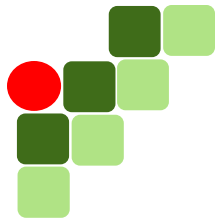
```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME  
FROM EMPLOYEES E  
      INNER JOIN DEPARTMENTS D  
      ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
```



Cláusula USING

- ▶ Usada quando as colunas da cláusula ON de um JOIN possuem o **mesmo nome** nas diferentes tabelas
- ▶ Recuperar sobrenome dos empregados e o nome do departamento onde trabalham

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME  
FROM EMPLOYEES E  
      INNER JOIN DEPARTMENTS D  
      USING (DEPARTMENT_ID) ;
```

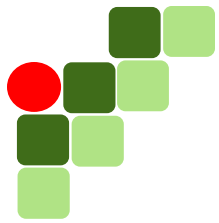


INNER JOIN

- Recuperar sobrenome dos empregados e o título dos cargos que eles exercem

```
SELECT E.LAST_NAME, J.JOB_TITLE  
FROM EMPLOYEES E  
      INNER JOIN JOBS J  
      ON E.JOB_ID = J.JOB_ID;
```

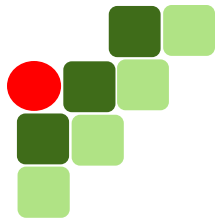
```
SELECT E.LAST_NAME, J.JOB_TITLE  
FROM EMPLOYEES E  
      INNER JOIN JOBS J  
      USING (JOB_ID);
```



INNER JOIN

Consulte sobrenome, título da função e salário dos empregados cujo salário é maior que 5000 e que trabalhem em departamentos com id maior que 50

```
SELECT E.LAST_NAME, J.JOB_TITLE, E.SALARY
FROM EMPLOYEES E
      INNER JOIN JOBS J
      ON E.JOB_ID = J.JOB_ID
WHERE E.SALARY > 5000
AND E.DEPARTMENT_ID > 50;
```

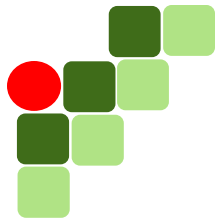


INNER JOIN

- Liste todas as cidades localizadas em países cujo nome contenha a palavra 'United'

```
SELECT L.CITY, C.COUNTRY_NAME
FROM LOCATIONS L
      INNER JOIN COUNTRIES C
      ON L.COUNTRY_ID = C.COUNTRY_ID
WHERE C.COUNTRY_NAME LIKE '%United%';
```

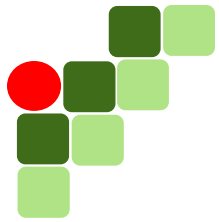
```
SELECT L.CITY, C.COUNTRY_NAME
FROM LOCATIONS L
      INNER JOIN COUNTRIES C
      USING (COUNTRY_ID)
WHERE C.COUNTRY_NAME LIKE '%United%';
```

INNER JOIN

- Recuperar nomes de departamento e sobrenome de seus gerentes

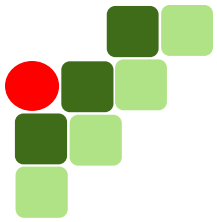
```
SELECT D.DEPARTMENT_NAME, E.LAST_NAME  
FROM DEPARTMENTS D  
      INNER JOIN EMPLOYEES E  
      ON D.MANAGER_ID = E.EMPLOYEE_ID;
```



INNER JOIN

- Recuperar nome dos empregados, o nome do departamento e a cidade onde trabalham

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME, L.CITY  
FROM EMPLOYEES E  
      INNER JOIN DEPARTMENTS D  
      ON E.DEPARTMENT_ID = D.DEPARTMENT_ID  
      INNER JOIN LOCATIONS L  
      ON D.LOCATION_ID = L.LOCATION_ID  
ORDER BY D.DEPARTMENT_NAME, E.LAST_NAME;
```



INNER JOIN

- Recuperar nomes dos departamentos e os nomes dos países onde estão localizados

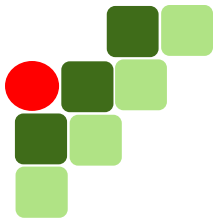
```
SELECT D.DEPARTMENT_NAME, C.COUNTRY_NAME  
FROM DEPARTMENTS D  
      INNER JOIN LOCATIONS L USING (LOCATION_ID)  
      INNER JOIN COUNTRIES C USING (COUNTRY_ID);
```



NATURAL JOIN

- ▶ Realiza junção considerando todas as colunas nas duas tabelas que **possuem o mesmo nome e o mesmo tipo de dados**
- ▶ Recuperar sobrenome dos empregados e o título das funções que eles exercem

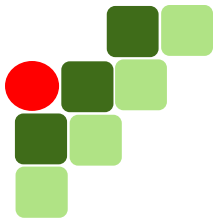
```
SELECT E.LAST_NAME, J.JOB_TITLE  
FROM EMPLOYEES E NATURAL JOIN JOBS J;
```



NATURAL JOIN

- Liste todas as cidades localizadas em países cujo nome contenha a palavra 'United'

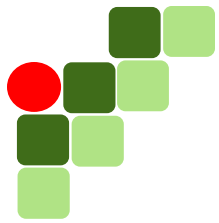
```
SELECT L.CITY, C.COUNTRY_NAME  
FROM LOCATIONS L  
      NATURAL JOIN COUNTRIES C  
WHERE C.COUNTRY_NAME LIKE '%United%';
```



NATURAL JOIN

- Recuperar uma listagem com nome de departamento, nome da cidade, nome do país e nome da região onde está localizado

```
SELECT D.DEPARTMENT_NAME, L.CITY,  
C.COUNTRY_NAME, R.REGION_NAME  
FROM DEPARTMENTS D  
      NATURAL JOIN LOCATIONS L  
      NATURAL JOIN COUNTRIES C  
      NATURAL JOIN REGIONS R;
```

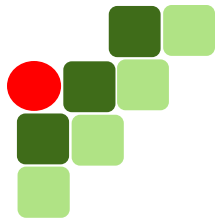


NATURAL JOIN

- ▶ NATURAL JOIN considera **todas as colunas** que possuem o mesmo nome nas duas tabelas
- ▶ Recuperar sobrenome dos empregados e o nome do departamento onde trabalham

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME  
FROM EMPLOYEES E  
      NATURAL JOIN DEPARTMENTS D;
```

- ▶ Essa consulta está correta?

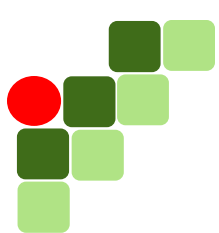


NATURAL JOIN

- ▶ NATURAL JOIN considera **todas as colunas** que possuem o mesmo nome nas duas tabelas
- ▶ Recuperar sobrenome dos empregados e o nome do departamento onde trabalham

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME  
FROM EMPLOYEES E  
      INNER JOIN DEPARTMENTS D  
      USING (DEPARTMENT_ID);
```

- ▶ **Cuidado como o uso de NATURAL JOIN!**

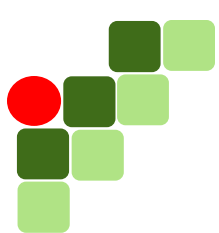


Cláusula USING / NATURAL JOIN

- ▶ Recuperar nomes dos departamentos que estão localizados no país com id = 'US'

```
SELECT D.DEPARTMENT_NAME, C.COUNTRY_ID  
FROM DEPARTMENTS D  
JOIN LOCATIONS L USING (LOCATION_ID)  
JOIN COUNTRIES C USING (COUNTRY_ID)  
WHERE C.COUNTRY_ID = 'US' ;
```

- ▶ Problema: não se deve qualificar uma coluna que é usada em uma cláusula USING ou em um NATURAL JOIN

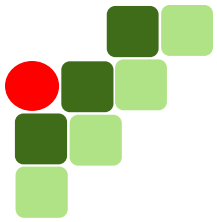


Cláusula USING / NATURAL JOIN

- Recuperar nomes dos departamentos que estão localizados no país com id = 'US'

```
SELECT D.DEPARTMENT_NAME, COUNTRY_ID
FROM DEPARTMENTS D
      INNER JOIN LOCATIONS L USING (LOCATION_ID)
      INNER JOIN COUNTRIES C USING (COUNTRY_ID)
WHERE COUNTRY_ID = 'US' ;
```

```
SELECT D.DEPARTMENT_NAME, COUNTRY_ID
FROM DEPARTMENTS D
      NATURAL JOIN LOCATIONS L
      NATURAL JOIN COUNTRIES C
WHERE COUNTRY_ID = 'US' ;
```



CROSS JOIN

- ▶ **Produto cartesiano**
 - Combina todas as linhas da primeira tabela com todas as linhas da segunda tabela da junção
 - Combinação exaustiva
- ▶ **Pode acontecer quando as condições de junção não são definidas ou estão incorretas**
- ▶ **Gera uma grande quantidade de linhas como resultados**

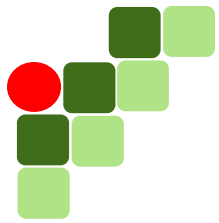


CROSS JOIN

- Comando CROSS JOIN gera produto cartesiano

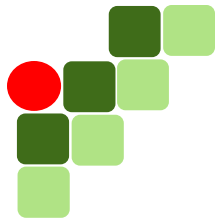
```
SELECT C.COUNTRY_ID, R.REGION_ID  
FROM COUNTRIES C  
      CROSS JOIN REGIONS R;
```

- Quantidade de linhas no resultado = $QT1 * QT2$
 - QT1: Quantidade de linhas da tabela 1
 - QT2: Quantidade de linhas da tabela 2



Dúvidas?





Referências

- ▶ **ELMASRI, Ramez E.; NAVATHE, Shamkant B. Sistemas de Banco de Dados. 6.ed. Pearson, 2011.**
- ▶ **Oracle Database SQL Language Reference**
 - **https://docs.oracle.com/cd/B28359_01/server.111/b28286/queries006.htm**