

Prática em Banco de Dados

Prof. Saulo Rosa e Silva

Banco de Dados

UNIP

Email: saulo.silva@docente.unip.br

Condições Lógicas

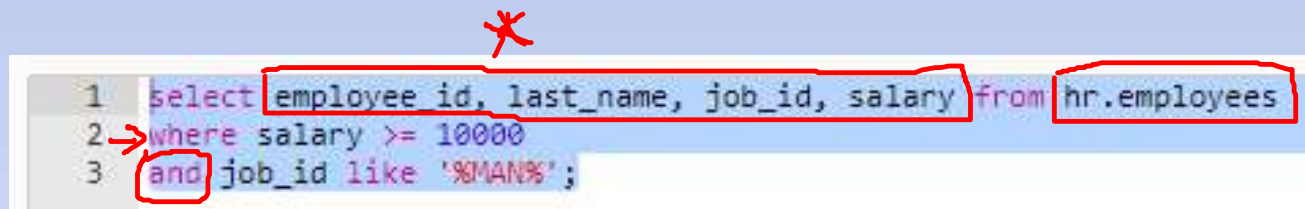
- É possível implementarmos funções lógicas nas consultas em SQL:

Operadores	Descrição
AND	Retorna TRUE se as duas condições forem atendidas
OR	Retornará TRUE se uma das condições for verdadeira
NOT	Retorna TRUE se a condição for falsa

Condições Lógicas

- AND

- Nesse exemplo, o operador AND, força a consulta para ser obrigada a atender as duas condições estabelecidas.
- Para retornar algum resultado, o salário terá que ser maior que 10000 e o job_id conter a palavra %MAN%.



The image shows a SQL query snippet with three lines of code. A red asterisk is placed above the first line. The first line is 'select employee_id, last_name, job_id, salary from hr.employees'. The second line is 'where salary >= 10000'. The third line is 'and job_id like '%MAN%';'. Red boxes highlight the table name 'hr.employees' in the first line, the 'and' keyword in the third line, and the entire third line. A red arrow points to the 'where' keyword in the second line.

```
1 select employee_id, last_name, job_id, salary from hr.employees
2 → where salary >= 10000
3 and job_id like '%MAN%';
```

Condições Lógicas

- O resultado apresentado, será conforme abaixo:

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
114	Raphaely	PU_MAN	11000
145	Russell	SA_MAN	14000
146	Partners	SA_MAN	13500
147	Errazuriz	SA_MAN	12000
148	Cambrault	SA_MAN	11000
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Condições Lógicas

- OR

- O comando OR, diferentemente do AND, necessita que somente uma das condições seja atendida.
- Nesse exemplo, se qualquer uma das condições forem atendidas, os resultados serão exibidos.

```
1 select employee_id, last_name, job_id, salary from hr.employees
2 where salary >= 10000
3 or job_id like '%MAN%';
```

Condições Lógicas

- Notem que o número de resultados apresentados será bem maior do que utilizando o AND, uma vez que uma OU outra condição atendida será considerada na hora da exibição.

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
108	Greenberg	FI_MGR	12008
114	Raphaely	PU_MAN	11000
120	Weiss	ST_MAN	8000
121	Fripp	ST_MAN	8200
122	Kaufling	ST_MAN	7900

156	King	SA_REP	10000
162	Vishney	SA_REP	10500
168	Ozer	SA_REP	11500
169	Bloom	SA_REP	10000
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
204	Baer	PR_REP	10000
205	Higgins	AC_MGR	12008

[Download CSV](#)
24 rows selected.

Regras de precedência

- Podemos também, combinar as condições AND, WHERE e OR para filtrarmos os resultados das consultas:

```
1 select employee_id, last_name, job_id, salary from hr.employees
2 where job_id = 'SA_REP'
3 or job_id = 'AD_PRES'
4 and salary > 15000;
```

Regras de Precedência

- Para a resposta, primeiramente o campo job_id tem que ser igual a 'AS_REP' ou 'AD_PRES', e o salário maior ou igual que 15000.

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
150	<u>Tucker</u>	<u>SA_REP</u>	<u>10000</u>
151	Bernstein	SA_REP	9500
152	Hall	SA_REP	9000
153	Olsen	SA_REP	8000
154	Cambrault	SA_REP	7500
155	Tuvault	SA_REP	7000
156	King	SA_REP	10000

Ordenação das linhas

- Order by
 - É possível com a função order by, ordenar os dados de maneira crescente ou decrescente, com base em uma determinada coluna.
 - **ASC:** ordem crescente, default
 - **DESC:** ordem decrescente

Ordenação das linhas

- No caso, vamos ordenar a consulta abaixo com base na coluna HIRE_DATE:

```
1 select employee_id, last_name, department_id, hire_date
2 from hr.employees
3 order by hire_date;
```

- O resultado, apresenta os empregados pela ordem de contratação.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	HIRE_DATE
102	De Haan	90	13-JAN-01
206	Gietz	110	07-JUN-02
204	Baer	70	07-JUN-02
203	Mavris	40	07-JUN-02
205	Higgins	110	07-JUN-02
109	Faviet	100	16-AUG-02
108	Greenberg	100	17-AUG-02
114	Raphaely	30	07-DEC-02

Ordenação das linhas

- Pode ordenar-se a mesma consulta em ordem decrescente, mediante o uso da partícula **desc** ao final da query:

```
1 select employee_id, last_name, department_id, hire_date
2 from hr.employees
3 order by hire_date desc;
```

- O resultado final, apresentar-se-á também a contento:

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	HIRE_DATE
173	Kumar	80	21-APR-08
167	Banda	80	21-APR-08
166	Ande	80	24-MAR-08
128	Markle	50	08-MAR-08
165	Lee	80	23-FEB-08
136	Philtanker	50	06-FEB-08
183	Geoni	50	03-FEB-08
149	Zlotkey	80	29-JAN-08

Concatenação

- É possível concatenar strings junto as colunas de uma tabela no retorno dos dados de uma consulta sql. O símbolo para utilização da concatenação é o ||.

The diagram shows a SQL query with several annotations explaining the concatenation process:

- `select`: Annotate with a red box and the label "string".
- `funcionario`: Annotate with a red box and the label "string".
- `||`: Annotate with a red box and the label "concatenação".
- `'id '`: Annotate with a red box and the label "string".
- `employee_id`: Annotate with a red box and the label "coluna".
- `||`: Annotate with a red box and the label "simulação de espaço".
- `'last name||'`: Annotate with a red box and the label "coluna".
- `tem o salario de`: Annotate with a red box and the label "string".
- `||s.salary`: Annotate with a red box and the label "coluna".
- `"Concatenado"`: Annotate with a red box and the label "apelido da coluna".

```
select funcionario || 'id ' || employee_id || 'last name||' tem o salario de ||s.salary "Concatenado" from hr.employees s
```

Concatenação

- O resultado final apresentará a consulta como uma sequência de linhas contínuas, com texto e o conteúdo das tabelas:

Concatenado
O funcionário id 100 King Tem o salário de 24000
O funcionário id 101 Kochhar Tem o salário de 17000
O funcionário id 102 De Haan Tem o salário de 17000
O funcionário id 103 Hunold Tem o salário de 9000
O funcionário id 104 Ernst Tem o salário de 6000
O funcionário id 105 Austin Tem o salário de 4800
O funcionário id 106 Pataballa Tem o salário de 4800
O funcionário id 107 Lorentz Tem o salário de 4200

Exercícios!

- 1. Selecionar todas as linhas da tabela Employees.
 - 2. Selecionar, da tabela Employees, todos os registros das colunas first_name e job_id.
 - 3. Selecionar, da tabela Employees, todas as linhas em que o campo job_id tenha o valor IT_PROG.
 - 4. Selecionar, da tabela Employees, todas as linhas em que o campo job_id tenha o valor IT_PROG e o campo salary tenha valor inferior ou igual a 4200.
 - 5. Selecionar, da tabela Employees, todas as linhas em que o campo job_id tenha o valor IT_PROG ou o campo salary tenha valor inferior ou igual a 4200.
-

(mais...) Exercícios!

- 6. Selecionar, da tabela Employees, todas as linhas em que o campo salary tenha valor superior ou igual a 5000.
- 7. Selecionar, da tabela Employees, todas as linhas em que o campo hire_date esteja entre as datas entre 01/01/98 e 01/03/98 – usar obrigatoriamente o conectivo BETWEEN.
- 8. Selecionar, da tabela Employees, todas as linhas em que o campo hire_date esteja entre as datas entre 01/01/98 e 01/12/99 – não usar o BETWEEN.
- 9. Selecionar, da tabela Employees, todas as linhas em que o campo hire_date apresente as datas exatas de 05/02/98 e 07/03/98 – usar obrigatoriamente o conectivo IN.
- 10. Selecionar, da tabela Employees, todas as linhas em que o campo hire_date apresente as datas exatas de 01/01/98 e 07/03/98 – não usar o IN.