



# ***Banco de Dados***

Aula 16: SQL

Prof. Fernando Xavier

[fernando.xavier@udf.edu.br](mailto:fernando.xavier@udf.edu.br)

# Banco de Dados

- Programação das Aulas - Preliminar

Data	Tema
07/02/2018	Aula 1: Introdução
14/02/2018	Feriado
21/02/2018	Aula 2: Conceitos Bancos de Dados
28/02/2018	Aula 3: Modelagem de Dados
07/03/2018	Aula 4: Transformação entre Modelos
14/03/2018	Aula 5: Ausência (compensação em 14/04)
21/03/2018	Aula 6: Restrições/Normalização 1
28/03/2018	Aula 7: Revisão
04/04/2018	Aula 8: Prova A21
11/04/2018	Aula 9: Normalização 2
18/04/2018	Aula 10: SQL – Create/Drop/Alter Table

# Banco de Dados

- Programação das Aulas - Preliminar

Data	Tema
25/04/2018	Aula 11: SQL - Create/Drop/Alter Table
02/05/2018	Aula 12: SQL - INSERT/DELETE
09/05/2018	Aula 13: SQL – Dia do Empreendedor
16/05/2018	Aula 14: SQL – SELECT/UPDATE
23/05/2018	Aula 15: SQL – SELECT
<b>30/05/2018</b>	<b>Aula 16: SQL</b>
06/06/2018	Aula 17: PRI
13/06/2018	Aula 18: Apresentação Trabalho Final
20/06/2018	Avaliação Final Campus Virtual – não há aulas
27/06/2018	Aula 19: Avaliação Final

# Banco de Dados

- Sobre tipos de dados

Tipo	Descrição
Varchar(50) (Oracle e MySQL)	String variável com até 50 caracteres
Integer (MySQL)	Número inteiro
Number (oracle)	Números (usam-se decimais mas servem para inteiros)

- Atente para que o tipo de dado depende do SGBD que você utilizar. Assim, um comando DDL para um SGBD pode não funcionar para outro.
- O mesmo vale se você tentar importar o modelo físico gerado pelo BrModelo (não existe um tipo *texto*).



Fonte: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>  
[https://docs.oracle.com/cd/B28359\\_01/server.111/b28318/datatype.htm#CNCPT012](https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm#CNCPT012)

# ***Banco de Dados***

- Esquemas
  - Em alguns SGBDs (p. Ex: PostgreSQL), são tratados como partes de um banco de dados (esquema public, esquema occurrences, etc)
  - Em outros SGBDs, como MySQL, uma schema é um sinônimo para banco de dados

100

- Busca em esquemas
  - Usa-se o ponto entre o nome do schema e a tabela
  - `SELECT <colunas> FROM SCHEMA.TABELA`

Schema  tabela 

```
select count(*) from aulaindice.produto;
```

No select acima, é retornada a quantidade de linhas da tabela produto do banco/schema aulaindice

# Banco de Dados

- Suponha que o usuário queira saber os nomes de todos os produtos que já foram vendidos, independente de quantas vezes

cliente	venda	itensvenda	produto
cliente_id	venda_id	item_id	produto_id
nomecliente	datavenda	venda_id	nomeproduto
	cliente	produto	preco
	vendedor	qtde	estoque

Como fazer?

# Banco de Dados

- Junção entre as tabelas de produto e itensvenda:

```
select iv.venda_id, p.nomeproduto from  
aula16.produto p inner join  
aula16.itensvenda iv on p.produto_id=iv.produto
```

cliente	venda	itensvenda	produto
cliente_id	venda_id	item_id	produto_id
nomecliente	datavenda	venda_id	nomeproduto
	cliente	produto	preco
	vendedor	qtde	estoque

venda_id	nomeproduto
1	Tedado
1	Mouse
2	Monitor 15 polegadas
3	Mouse
3	Monitor 17 polegadas

Mas e se ele quisesse os nomes sem repetir?





# ***Banco de Dados***

- SELECT DISTINCT
  - Usa-se para retornar apenas as linhas cujas colunas tem valores únicos
  - SELECT distinct <colunas> FROM <tabelas>;

# Banco de Dados

- SELECT DISTINCT

```
select distinct p.nomeproduto from  
aula16.produto p inner join aula16.itensvenda iv  
on p.produto_id=iv.produto
```

nomeproduto
Tedado
Mouse
Monitor 15 polegadas
Monitor 17 polegadas

# Banco de Dados

- Por que o select abaixo mostrou o mouse 2 vezes?

```
select distinct p.nomeproduto, iv.venda_id from  
aula16.produto p inner join aula16.itensvenda iv  
on p.produto_id=iv.produto
```

nomeproduto	venda_id
Tedado	1
Mouse	1
Monitor 15 polegadas	2
Mouse	3
Monitor 17 polegadas	3

# Banco de Dados

- Por que o select abaixo mostrou o mouse 2 vezes?

```
select distinct p.nomeproduto, iv.venda_id from  
aula16.produto p inner join aula16.itensvenda iv  
on p.produto_id=iv.produto
```

nomeproduto	venda_id
Tclado	1
Mouse	1
Monitor 15 polegadas	2
Mouse	3
Monitor 17 polegadas	3

O distinct é feito sobre o conjunto das colunas

# ***Banco de Dados***

- Visões
  - São tabelas virtuais geradas a partir de comandos SELECT
  - Contém tabelas e colunas
  - A criação usa as mesmas características de um select (funções, join, etc)
  - Podem ser materializadas, ou seja, os dados são gerados como se fossem em uma tabela física, podendo ser atualizados

# ***Banco de Dados***

- CREATE VIEW
  - CREATE OR REPLACE VIEW <NOME DA VIEW> AS  
    <INSTRUÇÃO SELECT>

# Banco de Dados

- Considere as tabelas

Cliente

cliente_id	nomediente
1	Prof. Fernando
2	Prof. Tite
3	Prof. Parreira
4	Prof. Zacalo

Venda

venda_id	datavenda	cliente	vendedor
1	2018-05-10	1	2
2	2018-05-15	2	4
3	2018-05-16	3	1
NULL	NULL	NULL	NULL

Se quiséssemos dados das duas tabelas, como um relatório de vendas, seria algo como:

```
SELECT c.*, v.venda_id, v.datavenda FROM cliente c  
INNER JOIN venda v ON c.cliente_id=v.cliente
```

# Banco de Dados

- Considere as tabelas

Cliente

cliente_id	nomediente
1	Prof. Fernando
2	Prof. Tite
3	Prof. Parreira
4	Prof. Zacalo

Venda

venda_id	datavenda	cliente	vendedor
1	2018-05-10	1	2
2	2018-05-15	2	4
3	2018-05-16	3	1
NULL	NULL	NULL	NULL

Podemos transformar essa consulta em uma visão:

```
create or replace view vendas_report as SELECT c.*,  
v.venda_id, v.datavenda FROM cliente c INNER JOIN  
venda v ON c.cliente_id=v.cliente
```



# Banco de Dados

- As buscas poderiam ser feitas diretamente na view:

*select \* from vendas\_report where datavenda between '2018-05-01' and '2018-05-31' order by datavenda;*

cliente_id	nomecliente	venda_id	datavenda
1	Prof. Fernando	1	2018-05-10
2	Prof. Tite	2	2018-05-15
3	Prof. Parreira	3	2018-05-16

Fica mais fácil de buscar na view, sem ter que montar várias operações de junção a cada vez que for necessário gerar um relatório

# Banco de Dados

- *Como gerar um relatório que contenha data da venda, nome do cliente e total da venda?*

cliente	venda	itensvenda	produto
cliente_id	venda_id	item_id	produto_id
nomecliente	datavenda	venda_id	nomeproduto
	cliente	produto	preco
	vendedor	qtde	estoque

# Banco de Dados

- Como gerar um relatório que contenha data da venda, nome do cliente e total da venda?

```
SELECT v.venda_id, v.datavenda, c.nomecliente, sum(p.preco*iv.qtde)
FROM
venda v inner join itensvenda iv on v.venda_id=iv.venda_id
        inner join produto p on iv.produto=p.produto_id
        inner join cliente c on v.cliente=c.cliente_id
group by v.venda_id
```

venda_id	datavenda	nomecliente	sum(p.preco*iv.qtde)
1	2018-05-10	Prof. Fernando	1040.00
2	2018-05-15	Prof. Tite	1600.00
3	2018-05-16	Prof. Parreira	1400.00

# Banco de Dados

- *Para criar uma view*

```
create or replace view total_por_venda as
SELECT v.venda_id, v.datavenda, c.nomecliente,
sum(p.preco*iv.qtde) as valor
FROM
venda v inner join itensvenda iv on v.venda_id=iv.venda_id
        inner join produto p on iv.produto=p.produto_id
        inner join cliente c on v.cliente=c.cliente_id
group by v.venda_id
```

# Banco de Dados

- *Buscando nessa view*

```
select * from total_por_venda;
```

venda_id	datavenda	nomecliente	valor
1	2018-05-10	Prof. Fernando	1040.00
2	2018-05-15	Prof. Tite	1600.00
3	2018-05-16	Prof. Parreira	1400.00

Como listar apenas as vendas cujo valor total tenha ultrapassado 1500?

# Banco de Dados

- Como listar apenas as vendas cujo valor total tenha ultrapassado 1500?

```
select * from total_por_venda  
where valor >= 1500;
```

venda_id	datavenda	nomediente	valor
2	2018-05-15	Prof. Tite	1600.00

# ***Banco de Dados***

- Índices
  - São estruturas usadas para otimizar as buscas em tabelas
  - Os usuários não vêem, mas são usadas para tornar as buscas mais rápidas
  - Os índices são mantidos pelo SGBD, logo crie índice apenas para as colunas onde há mais possibilidades de filtros
  - Cada operação de update em uma tabela com índice pode demorar mais do que em uma tabela sem índice

# Banco de Dados

- CREATE INDEX
  - Comando DDL para criar índice em uma tabela
  - CREATE INDEX <nome do índice> ON <Tabela> (colunas)

```
create index produtos_idx ON produto(nomeproduto);
```



# Banco de Dados

- Busca sem criar índice

```
explain select nome from produto where nome like '%B%';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	produto	NULL	ALL	NULL	NULL	NULL	NULL	30391	11.11	Using where

Após criar o índice para a coluna nome

```
create index produtos_idx ON produto(nome);  
explain select nome from produto where nome like '%B%';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	produto	NULL	index	NULL	produtos_idx	602	NULL	30391	11.11	Using where: Using index

# ***Banco de Dados***

- UNION
  - Cláusula associada a dois ou mais selects, quando se deseja unir registros de duas ou mais tabelas
  - Cada select intermediário deve retornar o mesmo número de colunas

# Banco de Dados

- UNION
  - SELECT <colunas1> FROM tabela1  
UNION  
SELECT <colunas2> FROM tabela2
  - A quantidade de colunas em coluna1 deve ser exatamente a mesma da quantidade de colunas de coluna2

# ***Banco de Dados***

- UNION
  - SELECT nomeproduto, preco FROM produtos
  - UNION
  - SELECT nomeservico, preco FROM servicos

# ***Banco de Dados***

- Exercícios: dado o banco de dados da aula passada:
  - Mostre todas as vendas do cliente Prof. Tite
  - Quais os produtos vendidos em cada venda? (nome do produto e data da venda)
  - Imprima os dados dos fornecedores e os produtos que eles fornecem
  - Quantas vendas cada vendedor fez? (nome e quantidade)

# ***Banco de Dados***

- Referências:
  - Elmasri, R., & Navathe, S. (2010). Fundamentals of database systems. Addison-Wesley Publishing Company.
  - Silberschatz, A., Korth, H. F., & Sudarshan, S. (1997). Database system concepts (Vol. 4). New York: McGraw-Hill.
  - Comandos SQL:  
<https://www.w3schools.com/sql/>