

# Banco de dados II

## Aula 1

Agosto de 2024

# Agenda Revisão SQL

- Comandos: SELECT e CREATE.
- Operadores
- Tipos
- Exercícios





01

# Operadores em geral

Lógicos e aritméticos

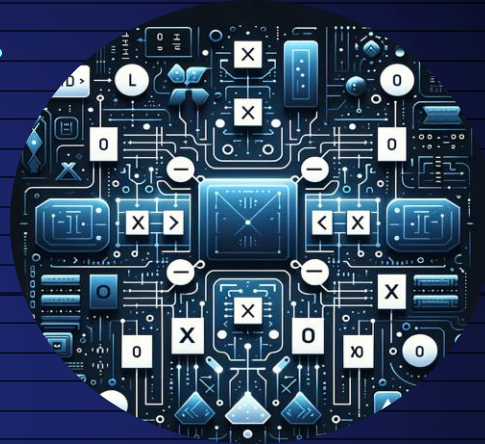
# Operadores aritméticos

- Soma(+)
- Subtração(-)
- Multiplicação(\*)
- Divisão(/)



# Operadores lógicos

- Igual a(=)
- Diferente(<>)
- Entre(Between... And...)
- Na lista(in)
- Maior(>)
- Menor(<)
- É nulo(is null)
- Maior ou igual(>=)
- Menor ou igual(<=)
- Parecido(Like)



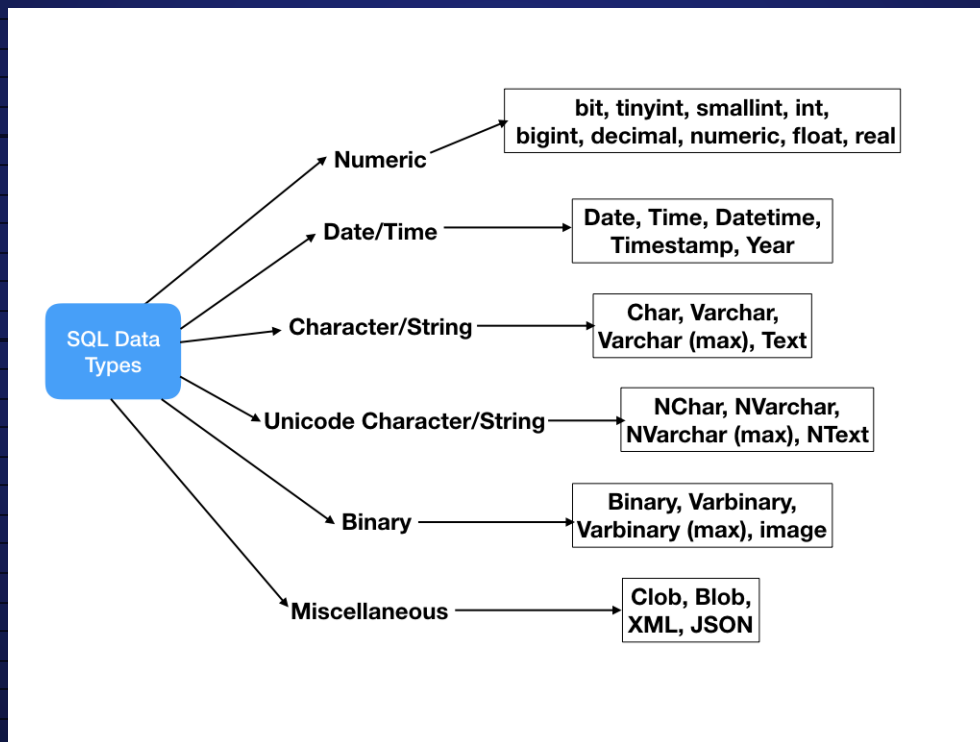


02

# Tipo de dados



# Tipos de dados em geral



# Tipos de dados numéricos

Tipo de Campo	Tamanho de Armazenamento
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 ou 8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D)	M+2 bytes se $D > 0$ , M+1 bytes se $D = 0$
NUMERIC(M,D)	M+2 bytes se $D > 0$ , M+1 bytes se $D = 0$



# Tipos de dados Data

**Date:** tipo data, armazena uma data. A margem de valores vai desde o 1 de Janeiro de 1001 ao 31 de dezembro de 9999. O formato de armazenamento é de ano-mês-dia.

**DateTime:** Combinação de data e hora. A margem de valores vai desde o 1 de Janeiro de 1001 às 0 horas, 0 minutos e 0 segundos ao 31 de Dezembro de 9999 às 23 horas, 59 minutos e 59 segundos. O formato de armazenamento é de ano-mês-dia horas:minutos:segundos

**TimeStamp:** Combinação de data e hora. A margem vai desde o 1 de Janeiro de 1970 ao ano 2037. O formato de armazenamento depende do tamanho do campo:

# Tipos de dados Data

Tamanho	Formato
14	AnoMesDiaHoraMinutoSegundo aaaammddhhmmss
12	AnoMesDiaHoraMinutoSegundo aammddhhmmss
8	AnoMesDia aaaammdd
6	AnoMesDia aammdd
4	AnoMes aamm
2	Ano aa

# Tipos de dados Data

**Time:** armazena uma hora. A margem de horas vai desde -838 horas, 59 minutos e 59 segundos. O formato de armazenamento é 'HH:MM:SS'.

**Year:** armazena um ano. A margem de valores permitidos vai desde o ano 1901 ao ano 2155. O campo pode ter tamanho dois ou tamanho 4 dependendo de se queremos armazenar o ano com dois ou quatro algarismos.

# Tipos de dados Data

Tipo de Campo	Tamanho de Armazenamento
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

# Funções para manipulação de Data

Funções para manipulação de datas no Oracle

Para você conseguir visualizar o conteúdo em outros formatos inclusive visualizando horário o Oracle nos oferece uma função chamada TO\_CHAR e através dela informando o formato desejado conseguimos ter mais controle da informação.

Exemplo:

Select numerodopedido,

To\_char(datadopedido, 'HH24:MI:SS DD-MON-RRRR') From pedidos;

FORMATO	DESCRIÇÃO
YYYY ou RRRR	Ano completo em número
YEAR	Ano por extenso
MM	Mês em número
MON	Mês em letras - formato abreviado
MONTH	Mês em letras - formato completo
DD	Dia em número
DY	Dia da semana em letras - formato abreviado
DAY	Dia da semana em letras - formato completo

# Tipos de dados - Caracteres

**VarChar(n)**: armazena uma cadeia de longitude variável. A cadeia poderá conter desde 0 até 255 caracteres. Dentro dos tipos de cadeia pode-se distinguir dois subtipos, os tipo Text e os tipo Blob (Binary Large Object) A diferença entre um tipo e outro é o tratamento que recebem na hora de ordená-los e compará-los. No tipo text ordena-se sem ter importância as maiúsculas e as minúsculas e no tipo blob ordena-se tendo em conta as maiúsculas e minúsculas. Os tipos blob utilizam-se para armazenar dados binários como podem ser ficheiros.

# Tipos de dados - Caracteres

- **TinyText** e **TinyBlob**: Coluna com uma longitude máxima de 255 caracteres.
- **Blob** e **Text**: um texto com um máximo de 65535 caracteres.
- **MediumBlob** e **MediumText**: um texto com um máximo de 16.777.215 caracteres.
- **LongBlob** e **LongText**: um texto com um máximo de caracteres 4.294.967.295. Há que ter em conta que devido aos protocolos de comunicação os pacotes podem ter um máximo de 16 Mb.
- **Enum**: campo que pode ter um único valor de uma lista que se especifica. O tipo Enum aceita até 65535 valores diferentes.
- **Set**: um campo que pode conter nenhum, um ou vários valores de uma lista. A lista pode ter um máximo de 64 valores.



# Tipos de dados - Caracteres

Tipo de campo	Tamanho de Armazenamento
CHAR(n)	n bytes
VARCHAR(n)	n +1 bytes
TINYBLOB, TINYTEXT	Longitude+1 bytes
BLOB, TEXT	Longitude +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitude +3 bytes
LOB, LONGTEXT	Longitude +4 bytes
ENUM('value1','value2',...)	1 ó dos bytes dependendo do número de valores
SET('value1','value2',...)	1, 2, 3, 4 ó 8 bytes, dependendo do número de valores



03

# Comando CREATE

# Criando Databases

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name [create_option] ...
```

create\_option:

```
{ [DEFAULT] CHARACTER SET [=] charset_name |
```

```
[DEFAULT] COLLATE [=] collation_name |
```

```
[DEFAULT] ENCRYPTION [=] {'Y' | 'N'} }
```

# Criando Databases

CREATE DATABASE TESTE;

O comando USE instrui o SGBD a utilizar o banco de dados especificado para rodar os comandos.

USE TESTE;

Para visualizar o banco de dados selecionado no momento use o comando:

SELECT DATABASE();

Podemos excluir um banco de dados existente com o comando DROP DATABASE;;

DROP DATABASE [IF EXISTS] nome\_BD;

# Usando Constraints

AUTO\_INCREMENT pode ser utilizado para automatizar um código que sirva de chave primária de uma tabela.

PRIMARY KEY define a chave primária da tabela, isto é, o campo que serve como chave da tabela e que não pode ser repetido.

NOT NULL define que um determinado campo seja de preenchimento obrigatório.

# Criando Databases

```
CREATE TABLE IF NOT EXISTS Tb_aluno(  
    id_aluno INT(6) NOT NULL PRIMARY KEY,  
    cd_rm_aluno INT(5),  
    nm_aluno VARCHAR(100) NOT NULL,  
    dt_nascimento_aluno DATE NOT NULL  
);
```

# Restrições de Integridade

NOT NULL - Não aceita valores nulos.

UNIQUE - Valores inseridos devem ser únicos.

PRIMARY KEY - Combinação entre NOT NULL e UNIQUE

FOREIGN KEY - Referencia uma chave estrangeira.

CHECK - Garante que os valores em uma coluna satisfaçam uma condição específica.

DEFAULT - Define um valor padrão para uma coluna se nenhum valor for especificado.

CREATE INDEX - Usado para criar e recuperar dados do banco de dados muito rapidamente.



04

Comando SELECT

# Comando SELECT

Permite recuperar os dados de um objeto do banco de dados, como uma tabela, view e, em alguns casos, uma stored procedure (alguns bancos de dados permitem a criação de procedimentos que retornam valor). A sintaxe mais básica do comando é:

```
SELECT <lista_de_campos> FROM <nome_da_tabela></nome_da_tabela></lista_de_campos>
```

```
SELECT codigo, nome FROM Clientes
```

```
SELECT * FROM Clientes
```



# Comando **SELECT** com **WHERE**

- A cláusula Where permite ao comando SQL passar condições de filtragem.

```
SELECT codigo, nome FROM Clientes
```

```
WHERE codigo = 10;
```

```
SELECT codigo, nome FROM Clientes
```

```
WHERE UF = 'RJ';
```

```
SELECT codigo, nome FROM Clientes
```

```
WHERE codigo >= 100 AND codigo <= 500;
```

```
SELECT codigo, nome FROM Clientes
```

```
WHERE UF = 'MG' OR UF = 'SP';
```

05

# Exercícios síncronos

# Exercício de revisão

1 - Crie a tabela Funcionario abaixo, começando pelo database até a tabela, sendo CodFun chave primária:

<i>Campo</i>	<i>Tipo</i>	<i>Descrição</i>
CodFun	integer	código do funcionário (não nulo)
Nome	varchar(40)	nome do funcionário (não nulo)
Depto	char ( 2 )	departamento onde está locado o funcionário
Funcao	char (20)	função do funcionário
Salario	decimal (10, 2)	salário do funcionário

# Exercício de revisão

2 - Selecione todos os funcionários do departamento de vendas:



.....

# Obrigado!

Alguma dúvida?  
[jheymesso.Cavalcanti@unicap.br](mailto:jheymesso.Cavalcanti@unicap.br)