Banco de Dados

Aula 1

Na década de 60, O departamento de defesa dos Estados Unidos trabalharam nisso foi ae que surgiu a linguagem COBOL. No CODASY foi discutido o surgimento de uma nova tecnologia . Essa tecnologia ganhou logo um nome de Bancos de Dados.

Bancos de dados é composto por 4 partes como:

- **Base de dados** ,que são os dados propriamente ditos , a estrutura de todo banco de dados.

- **Sistema Gerenciador,** conhecido como SGBD em inglês essa sigla é DNS.

- **Linguagem de Exploração ,** linguagem de acesso ao dado.

- **Programas Adicionais,** como gerencia de usuários, otimizadores de dados, etc ...

A IBM teve uma grande influência para construção e evolução do Bancos de Dados.

A primeira coisa que a IBM propôs foi a criação de Dados Hierárquicos (dados interligados), só que não teria relacionamentos .

Na Década de 70 Edgar F.Codd inovo outro modelo , nesse paradigma os dados eles seriam armazenados Invés de Hierarquia ou ligações de redes, eles teriam ligações mais intrínseca , elas teriam relação. E através dos estudos de Codd , que surgiu o modelo Relacional.

A linguagem especifica , ela já teve alguns nomes : O primeiro delas foi “**Structured Query English Language**” conhecia como “**SEQUEL**” não demorou muito e essa linguagem foi chamada para “**Structured Query Language”** conhecida com “**SQL**”.

Tipos de Bancos de Dados:

* Oracle.
* IBM (DB2).
* Dbase.
* Microsoft (SQLSever).
* My SQL.
* Maria DB.
* Firebird.
* PostgreSQL.

Aula 2

Em 1984 na Suécia , dois programadores, o Michael Widenius e David Axmark, resolveram criar um modelo gratuito, a ideia era criar um Banco de dados gratuito, baseando no modelo Relacional. Surge então o SQL. Em 2007 veio uma empresa (SUN) e comprou o SQL . A SUN deixou de existir e agora ela pertence a Oracle.

O MySql tem :

**DDL** -> É uma linguagem de Definição. Por Exemplo: Criar Banco de dados, Criar uma tabela, Alterar o Banco de dados. Qualquer comando de definição da estrutura da base de dados é mantida dentro do MYSQL pela porção DDL.

**DML** -> É uma linguagem de Manipulação. Por Exemplo: Incluir novos dados, excluir dados , manipular dados de qualquer maneira, alterar a composição deles.

**DQL** -> È uma linguagem de Solicitações. Por exemplo: Fazer um Select, etc...

**DCL** -> É uma linguagem de Controle. Por exemplo: Definir que usuário podem acessar seu banco, que tipo de acesso ele vai poder fazer , que tipo de comando pode executar.

**DTL** -> È uma linguagem de Transações. Por exemplo: Qualquer solicitações que pode ser feita a um banco de dados e ele vai te atender da melhor maneira possível seguindo os 4 princípios que a gente chama de (D.I.C.A)

As característica de uma boa transação:

**D** – **urabilidade**

**I** - **solamento**

**C** - **osistencia**

**A** – **tomicidade**

**Durabilidade –** Todo dado que é colocado ou alterado ou manipulado tem que permanecer durável tem que permanecer dessa maneira, enquanto eu quiser que ele esteja lá.

**Isolamento –** Se eu tenho duas transações feitas ao mesmo tempo ela têm que ser executivas sem uma interferir na outra, precisa que ser isoladas.

**Consistência –** Toda transições tem que levar a banco de dados de um estado consistente a outro consistente. Se tudo estava ok antes, tudo tem continuar ok.

**Atomicidade –** Toda transação tem que ser atônica ou toda ela acontece ou nada acontece, ou tudo da certo ou ele retorna para um estado consistente anteriormente**.**

Aula 3

No mundo do Bancos de dados , vamos agrupar coisas que tem características semelhantes e separar coisas que tem características diferentes essas coisas que têm característica diferentes .

Então o *banco de dados* são coleções de dados, que são de características separadas, mas que estão organizados em locais específicos . Esses locais específicos são minha tabelas. *Tabelas* guardam dados de coisas que tem características semelhantes e se quiser , pode ter varias tabelas dentro de um banco de dados. Os *dados* dentro das tabelas também tem um nome especifico, que são Registros.

Resumindo: Bancos de dados são conjuntos de tabelas e tabelas são conjuntos de registros.

Começando a Programar DBA

Criando um banco de dados:

CREATE DATABASE cadastro;

Criando tabela:

CREATE TABLE pessoas(

nome varchar (30),

idade tinyint,

sexo char(1),

peso float ,

altura float ,

nacionalidade varchar(20)

);

Ver a Estrutura de uma tabela:

DESCRIBE pessoas;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Banco de dados contém tabelas. Tabelas contém Registros. Registros são compostos por campos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Tipos primitivos**

**Numéricos**

Inteiro 🡪 TinyInt, SmallInt , Int, MediumInt, BigInt

Real 🡪 Decimal, Float, Double, Real

Lógico 🡪 Bit, Boolean

**Data/Tempo**

Date, DataTime, TimeStamp, Time, Year

**Literal**

Caractere 🡪 Char, VarChar

Texto 🡪 TinyText, Text, MediumText, LongText

Binário 🡪 TinyBlob, Blob, MediumBlob, LongBlob

Coleção 🡪 Enum, Set

**Espacial**

Geometry, Point, Polygon, MultiPolygon

Aula 4

O formato UTF-8 tem caracteres especiais , inclusive as acentuações que agente trabalha.

Apagando um Banco de dados :

DROP DATABASE cadastro;

(Drop significa largar, apagar, abandonar)

Criando um Banco de dados:

CREATE DATABASE cadastro

DEFAULT CHARACTER SET utf8

DEFAULT COLLATE utf8\_general\_ci;

(O Collate , também serve para definição dos caracteres)

Fazendo a Estrutura da Tabela Pessoas:

(Primary key = a funcionalidade de uma chave primaria é não deixar ter duplas de repetições dentro de uma mesma tabela ).

CREATE TABLE pessoas(

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR (30) NOT NULL,

nascimento DATE,

sexo ENUM ('M' , 'F'),

peso DECIMAL (5,2),

altura DECIMAL (3,2),

nacionalidade VARCHAR (20) DEFAULT 'Brasileiro',

PRIMARY KEY (id)

)DEFAULT CHARSET = utf8;

Mostrar tudo que tem na tabela pessoas:

SELECT \* FROM pessoas;

(select \* from pessoas = seleciona tudo de pessoas)

Aula 5 – Inserindo dados na tabela (INSERT INTO)

Inserindo os dados dentro da tabela:

INSERT INTO pessoas

(nome, nascimento, sexo, peso, altura, nacionalidade)

VALUES

( 'Godofredo', '1984-01-02' ,'M' ,'78.5', '1.83' , 'Brasileiro');

INSERT INTO pessoas

(id ,nome, nascimento, sexo, peso, altura, nacionalidade)

VALUES

(DEFAULT, 'Maria', ' 1999-12-30' ,'F' ,' 55.2, 1.65' , 'Portuguesa');

(default = padrão que está definido na estrutura do banco de dados).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS :** Se a ordem dos campos for exatamente a ordem que foi definida no banco, então não precisa dizer quais são os campos.

**Por exemplo:**

INSERT INTO pessoas

VALUES

(DEFAULT, 'Adalgiza', ' 1930-11-2' ,'F' ,' 63.2’, ’1.75' , 'Irlandesa');

Inserindo vários dados ao mesmo tempo:

INSERT INTO pessoas

(id, nome, nascimento, sexo, peso, altura, nacionalidade)

VALUES

( DEFAULT, 'Ana’,'1975-12-18' ,'F' ,'58.5', '1.43' , 'Americana'),

( DEFAULT, 'Pedro’,'2000-07-15' ,'M' ,'100.5', '2.00' , 'Brasileiro');

Comandos:

DDL (Data Definition Language) :

CREATE DATABASE

CREATE TABLE

DROP TABLE

ALTER TABLE

DML (Data Manipulation Language) :

INSERT INTO

UPDATE

DELETE

TRUNCATE

DQL (Data Query Language) :

SELECT

Aula 6 – Alterando Estrutura da tabela (ALTER TABLE e DROP TABLE)

Comando para Alterar a Estrutura da Tabela:

(Iremos adicionar uma nova coluna).

ALTER TABLE pessoas

ADD COLUMN profissão VARCHAR (10);

**OBS:** Os campos no MySQL são chamados de colunas (COLUMN), referencia aos campos de uma coluna.

Comando para Mostrar como está estruturado a tabela:

DESC pessoas;

OU

DESCRIBE pessoas;

Comando para Alterar a Estrutura da Tabela:

(Iremos Apagar uma nova coluna).

ALTER TABLE pessoas

DROP COLUMN profissão;

Comando para Alterar a Estrutura da Tabela:

(Colocando (Adicionando) Colunas na posição Desejada)

Adicionando após uma Coluna. Nesse caso após a coluna Nome

ALTER TABLE pessoas

ADD COLUMN profissão VARCHAR (10) **AFTER** nome;

Adicionando uma Coluna em primeiro lugar. Nesse caso a Coluna codigo ficara em primeiro lugar.

ALTER TABLE pessoas

ADD COLUMN codigo INT **FIRST** ;

Adicionando Coluna no ultimo lugar . Nesse caso no final de todas as coluna

ALTER TABLE pessoas

ADD COLUMN signo VARCHAR (10);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: Nesse caso de adicionar tabela não necessitaria do parâmetro COLUMN. Também funcionaria sem colocar ele.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comando para Alterar a Estrutura da definição de uma coluna na Tabela:

ALTER TABLE pessoas

**MODIFY** COLUMN profissão VARCHAR (**20**);

Comando para Alterar o nome de uma coluna na Tabela:

ALTER TABLE pessoas

**CHANGE** COLUMN profissão **prof** VARCHAR (**20**);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: Ele muda pode ser usado par editar tanto na Estrutura da definição de uma coluna ou também para modificar o nome de uma coluna.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comando para Alterar o nome de uma Tabela:

ALTER TABLE pessoas RENAME TO gafanhotos;

Criando uma nova tabela

CREATE TABLE IF NOT EXISTS cursos (

nome VARCHAR (30) NOT NULL UNIQUE,

descricao TEXT,

carga INT UNSIGNED,

totalaulas INT,

ano YEAR DEFAULT '2016'

) DEFAULT CHARSET = utf8;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: O IF NOT EXISTS ou IF EXISTS , é um parâmetro muito legal do “CREATE” .

Por exemplo:

IF NOT EXISTS

Você só vai criar uma Tabela, Banco ou Coluna, se ele não existir.

IF EXISTS

Você só vai apagar uma tabela ou um Banco se ele existir

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: O NOT NULL

Por exemplo:

NOT NULL

Obrigatório preencher o campo, não vai aceita o campo em branco

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: A UNIQUE

Por exemplo:

UNIQUE

O campo a ser preenchido vai ser Único, não vai aceitar valores repetitivos (iguais) .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: A UNSIGNED

Por exemplo:

UNSIGNED

O campo a ser preenchido não ira aceitar sinais ou seja valores negativos .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fazendo teste para não criar uma tabela que já existe e também, não sobre escrever a tabela já criada.

CREATE TABLE

**IF NOT EXISTS** gafanhotos (teste INT);

Adicionando a coluna id\_curso em primeiro lugar

ALTER TABLE cursos

**ADD COLUMN** id\_curso **INT FIRST** ;

Adicionando uma Primary key na tabela cursos

ALTER TABLE cursos

ADD **PRIMARY KEY** (id\_curso);

Apagando uma tabela inteira

**DROP** TABLE teste;

Aula 7 – Manipulando Linha (registros , tuplas) (UPDATE, DELETE e TRUNCATE) .

Comando para selecionar tudo da tabela gafanhotos e mostrar os dados de todas as colunas.

SELECT \* FROM gafanhotos;

Comando para manipular linha (dados)

**UPDATE** cursos

SET nome = ‘HTML5’

WHERE id\_curso = ‘1’;

(Update = Atualize||Set=Configure||Where=Onde)

UPDATE cursos

SET nome = 'HTML5'

WHERE id\_curso = '1';

UPDATE cursos

SET nome = 'PHP',ano='2015'

WHERE id\_curso = '4';

UPDATE cursos

SET nome = 'Java',ano='2015',carga = '40'

WHERE id\_curso = '5'

**LIMIT** **1**;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: LIMIT é para limitar quantas linhas vão poder ser afetadas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comando apagar linha (dados)

**DELETE** FROM cursos

WHERE id\_curso=’8’

**DELETE** FROM cursos

WHERE ano =’2018’

LIMITE 3;

Comando para apagar todas as linhas (dados)

**TRUNCATE** TABLE cursos;

Ou

**TRUNCATE** cursos;

Aula 8 – Gerenciamento Cópias de Segurança MySQL.

1-Passo clicar em Server

2-Passo clicar em DataExport

3-Seleciona o Banco de dados que queira exportar

4-Seleciona o tipo (Dump Structure and Date [Estrutura e os dados] || Dump Data Only [Só os dados] || Dump Structure Only [ Só a Estrutura ] )

5- Selecionar qual tipo de exportação:

Export to Dump Project Folder [ Exportação projeto inteiro]

Export to Self-Contained File [Exportação do Arquivo único]

6- Selecionar

Include create Schema [Mesmo que o banco de dados não exista ele irá criar.

7- Clicar em Start Export

8- Vai pedir a senha

Esse Arquivo ira estar no meu computador\documento\pastaDumps

Aula 9 – PHPMY Admin (Parte1)

Entrando no PhpMyAdmin: localhost/phpmyadmin

Aula10 – PHPMY Admin (Parte2)

Aula11 – SELECT (Parte1)

-- Dump completed on 2016-04-17 17:56:27

CREATE DATABASE IF NOT EXISTS `Cadastro2`;

USE `cadastro2`;

-- Table structure for table `cursos`

DROP TABLE IF EXISTS `cursos2`;

CREATE TABLE `cursos2` (

`idcurso2` int(11) NOT NULL DEFAULT '0',

`nome2` varchar(30) NOT NULL,

`descricao2` text,

`carga2` int(10) unsigned DEFAULT NULL,

`totaulas2` int(10) unsigned DEFAULT NULL,

`ano2` year(4) DEFAULT '2016',

PRIMARY KEY (`idcurso2`),

UNIQUE KEY `nome2` (`nome2`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping data for table `cursos`

LOCK TABLES `cursos2` WRITE;

INSERT INTO `cursos2`

VALUES

(1,'HTML5','Curso de HTML5',40,37,2014),

(2,'Algoritmos','Lógica de Programação',20,15,2014),

(3,'Photoshop5','Dicas de Photoshop CC',10,8,2014),

(4,'PHP','Curso de PHP para iniciantes',40,20,2015),

(5,'Java','Introdução à Linguagem Java',40,29,2015),

(6,'MySQL','Bancos de Dados MySQL',30,15,2016),

(7,'Word','Curso completo de Word',40,30,2016),

(8,'Python','Curso de Python',40,18,2017),

(9,'POO','Curso de Programação Orientada a Objetos',60,35,2016),

(10,'Excel','Curso completo de Excel',40,30,2017),

(11,'Responsividade','Curso de Responsividade',30,15,2018),

(12,'C++','Curso de C++ com Orientação a Objetos',40,25,2017),

(13,'C#','Curso de C#',30,12,2017),

(14,'Android','Curso de Desenvolvimento de Aplicativos para Android',60,30,2018),

(15,'JavaScript','Curso de JavaScript',35,18,2017),

(16,'PowerPoint','Curso completo de PowerPoint',30,12,2018),

(17,'Swift','Curso de Desenvolvimento de Aplicativos para iOS',60,30,2019),

(18,'Hardware','Curso de Montagem e Manutenção de PCs',30,12,2017),

(19,'Redes','Curso de Redes para Iniciantes',40,15,2016),

(20,'Segurança','Curso de Segurança',15,8,2018),(

21,'SEO','Curso de Otimização de Sites',30,12,2017),

(22,'Premiere','Curso de Edição de Vídeos com Premiere',20,10,2017),

(23,'After Effects','Curso de Efeitos em Vídeos com After Effects',20,10,2018),

(24,'WordPress','Curso de Criação de Sites com WordPress',60,30,2019),

(25,'Joomla','Curso de Criação de Sites com Joomla',60,30,2019),

(26,'Magento','Curso de Criação de Lojas Virtuais com Magento',50,25,2019),

(27,'Modelagem de Dados','Curso de Modelagem de Dados',30,12,2020),

(28,'HTML4','Curso Básico de HTML, versão 4.0',20,9,2010),

(29,'PHP7','Curso de PHP, versão 7.0',40,20,2020),

(30,'PHP4','Curso de PHP, versão 4.0',30,11,2010);

UNLOCK TABLES;

-- Table structure for table `gafanhotos`

DROP TABLE IF EXISTS `gafanhotos2`;

CREATE TABLE `gafanhotos2` (

`id2` int(11) NOT NULL AUTO\_INCREMENT,

`nome2` varchar(30) NOT NULL,

`profissao2` varchar(20) DEFAULT NULL,

`nascimento2` date DEFAULT NULL,

`sexo2` enum('M','F') DEFAULT NULL,

`peso2` decimal(5,2) DEFAULT NULL,

`altura2` decimal(3,2) DEFAULT NULL,

`nacionalidade2` varchar(20) DEFAULT 'Brasil',

PRIMARY KEY (`id2`)

) ENGINE=InnoDB AUTO\_INCREMENT=62 DEFAULT CHARSET=utf8;

-- Dumping data for table `gafanhotos`

LOCK TABLES `gafanhotos2` WRITE;

INSERT INTO `gafanhotos2`

VALUES

(1,'Daniel Morais','Auxiliar Administrat','1984-01-02','M',78.50,1.83,'Brasil'),

(2,'Talita Nascimento','Farmacêutico','1999-12-30','F',55.20,1.65,'Portugal'),

(3,'Emerson Gabriel','Programador','1920-12-30','M',50.20,1.65,'Moçambique'),

(4,'Lucas Damasceno','Auxiliar Administrat','1930-11-02','M',63.20,1.75,'Irlanda'),

(5,'Leila Martins','Farmacêutico','1975-04-22','F',99.00,2.15,'Brasil'),

(6,'Letícia Neves','Programador','1999-12-03','F',87.00,2.00,'Brasil'),

(7,'Janaína Couto','Auxiliar Administrat','1987-11-12','F',75.40,1.66,'EUA'),

(8,'Carlisson Rosa','Professor','2010-08-01','M',78.22,1.98,'Brasil'),

(9,'Jackson Telles','Programador','1999-01-23','M',55.75,1.33,'Portugal'),

(10,'Danilo Araujo','Dentista','1975-12-10','M',99.21,1.87,'EUA'),

(11,'Andreia Delfino','Auxiliar Administrat','1975-07-01','F',48.64,1.54,'Irlanda'),

(12,'Valter Vilmerson','Ator','1985-10-12','M',88.55,2.03,'Brasil'),(

13,'Allan Silva','Programador','1993-11-11','M',76.99,1.55,'Brasil'),

(14,'Rosana Kunz','Professor','1935-01-16','F',55.24,1.87,'Brasil'),

(15,'Josiane Dutra','Empreendedor','1950-01-20','F',98.70,1.04,'Portugal'),

(16,'Elvis Schwarz','Dentista','2011-05-07','M',66.69,1.76,'EUA'),

(17,'Paulo Narley','Auxiliar Administrat','1997-03-17','M',120.10,2.22,'Brasil'),

(18,'Joade Assis','Médico','1930-12-01','M',65.88,1.78,'França'),

(19,'Nara Matos','Programador','1978-03-17','F',65.90,1.33,'Brasil'),

(20,'Marcos Dissotti','Empreendedor','2010-01-01','M',53.79,1.54,'Portugal'),

(21,'Ana Carolina Mendes','Ator','2000-12-15','F',88.30,1.54,'Brasil'),

(22,'Guilherme de Sousa','Dentista','2001-05-18','M',132.70,1.97,'Moçambique'),

(23,'Bruno Torres','Auxiliar Administrat','2000-01-30','M',44.65,1.65,'Brasil'),

(24,'Yuji Homa','Empreendedor','1996-12-25','M',33.90,1.22,'Japão'),

(25,'Raian Porto','Programador','1989-05-05','M',54.89,1.54,'Brasil'),

(26,'Paulo Batista','Ator','1999-03-15','M',110.12,1.87,'Portugal'),

(27,'Monique Precivalli','Auxiliar Administrat','2013-12-30','F',48.20,1.22,'Brasil'),

(28,'Herisson Silva','Auxiliar Administrat','1965-10-10','M',74.65,1.56,'EUA'),

(29,'Tiago Ulisses','Dentista','1993-04-22','M',150.30,2.35,'Brasil'),

(30,'Anderson Rafael','Programador','1989-12-01','M',64.22,1.44,'Irlanda'),

(31,'Karine Ribeiro','Empreendedor','1988-10-01','F',42.10,1.65,'Brasil'),

(32,'Roberto Luiz Debarba','Ator','2007-01-09','M',77.44,1.56,'Brasil'),

(33,'Jarismar Andrade','Dentista','2000-06-23','F',63.70,1.33,'Brasil'),

(34,'Janaina Oliveira','Professor','1955-03-12','F',52.90,1.76,'Canadá'),

(35,'Márcio Mello','Programador','2011-11-20','M',54.11,1.55,'EUA'),

(36,'Robson Rodolpho','Auxiliar Administrat','2000-08-08','M',110.10,1.76,'Brasil'),

(37,'Daniele Moledo','Empreendedor','2006-08-11','F',101.30,1.99,'Brasil'),

(38,'Neto Sophiate','Ator','1996-05-17','M',59.28,1.65,'Portugal'),

(39,'Neriton Dias','Auxiliar Administrat','2009-10-30','M',48.99,1.29,'Brasil'),

(40,'André Schmidt','Programador','1993-07-26','M',55.37,1.22,'Angola'),

(41,'Isaias Buscarino','Dentista','2001-01-07','M',99.90,1.55,'Moçambique'),

(42,'Rafael Guimma','Empreendedor','1968-04-11','M',88.88,1.54,'Brasil'),

(43,'Ana Carolina Hernandes','Ator','1970-10-11','F',65.40,2.08,'EUA'),

(44,'Luiz Paulo','Professor','1984-11-01','M',75.12,1.38,'Portugal'),

(45,'Bruna Teles','Programador','1980-11-07','F',55.10,1.86,'Brasil'),

(46,'Diogo Padilha','Auxiliar Administrat','2000-03-03','M',54.34,1.88,'Angola'),

(47,'Bruno Miltersteiner','Dentista','1986-02-19','M',77.45,1.65,'Alemanha'),

(48,'Elaine Nunes','Programador','1998-08-15','F',35.90,2.00,'Canadá'),

(49,'Silvio Ricardo','Programador','2012-03-12','M',65.99,1.23,'EUA'),

(50,'Denilson Barbosa da Silva','Empreendedor','2000-01-08','M',97.30,2.00,'Brasil'),

(51,'Jucinei Teixeira','Professor','1977-11-22','F',44.80,1.76,'Portugal'),

(52,'Bruna Santos','Auxiliar Administrat','1991-12-01','F',76.30,1.45,'Canadá'),

(53,'André Vitebo','Médico','1970-07-01','M',44.11,1.55,'Brasil'),

(54,'Andre Santini','Programador','1991-08-15','M',66.00,1.76,'Itália'),

(55,'Ruan Valente','Programador','1998-03-19','M',101.90,1.76,'Canadá'),

(56,'Nailton Mauricio','Médico','1992-04-25','M',86.01,1.43,'EUA'),

(57,'Rita Pontes','Professor','1999-09-02','F',54.10,1.35,'Angola'),

(58,'Carlos Camargo','Programador','2005-02-22','M',124.65,1.33,'Brasil'),

(59,'Philppe Oliveira','Auxiliar Administrat','2000-05-23','M',105.10,2.19,'Brasil'),

(60,'Dayana Dias','Professor','1993-05-30','F',88.30,1.66,'Angola'),

(61,'Silvana Albuquerque','Programador','1999-05-22','F',56.00,1.50,'Brasil');

UNLOCK TABLES;

Comando para Selecionar Todas as Linhas e Colunas

SELECT \* FROM curso;

(Select = Selecione|| \* = Tudo || From = Da)

Comando para Selecionar Todas as Linhas e Colunas

SELECT \* FROM curso;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: ORDER BY Serve para ordenar em ordem alfabética ou do menor para o maior.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: DESC Serve para ordenar em ordem alfabética invertida ou do maior para o menor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: Asc Serve para ordenar em ordem alfabética ou do menor para o maior. Não é necessário colocar ele pois se colocar só o comando ORDER BY ele entenderá que é do menor para o maior ou em ordem alfabética.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Selecionado (mostrando) somente as colunas desejadas.

SELECT nome2, peso2,altura2

FROM gafanhotos2;

Selecionando (mostrando) somente as colunas desejadas, em ordem crescente , ordem alfabética.

SELECT nome2, peso2,altura2

FROM gafanhotos2

**ORDER BY** nome2 ;

OU

SELECT nome2, peso2,altura2

FROM gafanhotos2

**ORDER BY** nome2 **ASC** ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: ORDER BY Serve para ordenar em ordem alfabética ou do menor para o maior.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: Asc Serve para ordenar em ordem alfabética ou do menor para o maior. Não é necessário colocar ele pois se colocar só o comando ORDER BY ele entenderá que é do menor para o maior ou em ordem alfabética.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Selecionando (mostrando) somente as colunas desejadas em ordem decrescente, alfabética Invertida.

SELECT nome2, peso2,altura2

FROM gafanhotos2

ORDER BY nome2 **DESC** ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OBS: DESC Serve para ordenar em ordem alfabética invertida ou do maior para o menor.

SELECT nome2, peso2,altura2

FROM gafanhotos2

ORDER BY nome2 ASC,ano2 DESC ;

Selecionando (mostrando) somente as colunas desejadas, em ordem crescente , ordem alfabética.

SELECT nome2, peso2,altura2

FROM gafanhotos2

ORDER BY nome2,Peso2 ;

(Ele ira mostrar em ordem alfabética nome2 e o Peso2 em ordem crescente , dois campos . Não tem limites de campos para fazer esses parâmetros. Esse exemplo serve nos de mais select que foi citados acima.)

Selecionando (mostrando) as linhas desejadas em ordem crescente, alfabética.

SELECT \* FROM cursos2

**WHERE** ano2 = ‘2016’

ORDER BY nome2;

SELECT \* FROM cursos2

**WHERE** ano2 <= ‘2016’

ORDER BY nome2;

SELECT nome2,descricao2 FROM cursos2

**WHERE** ano2 <= ‘2015’

ORDER BY nome2;

**Operadores**

**<** Menor **>** Maior

**<=** Menor Igual **=>** Maior ou Igual

**=** Igual **!=** Diferente ou **<>**

Comando para seleciona entre uma coisa ou outra

(nesse caso pegar no ano de 2014 até 2016 em outras palavra entre o ano de 2014 até 2016);

SELECT nome2, ano2 FROM cursos2

WHERE ano2 **BETWEEN** 2014 AND 2016;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** BETWEEN = Entre uma coisa ou outra

(Faixas de valores)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comando para seleciona entre uma coisa ou outra

(nesse caso pegar o ano de 2014 , 2016,2017 );

SELECT nome2, ano2 FROM cursos2

WHERE ano2 **IN** (2014 ,2016,2017);

SELECT nome2, ano2 FROM cursos2

WHERE ano2 **IN** (2014 ,2016,2017)

ORDER BY ano2;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** IN = Pode colocar valores específicos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comando para seleciona duas condições verdadeira

(nesse caso ele mostrar se carga2>35 E totaulas2<30);

SELECT nome2,carga2,totaulas2

FROM cursos2

WHERE carga2 >35 AND totaulas2<30;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** AND = Seleciona se as duas codições que você colocou for verdadeira.

SELECT nome2,carga2,totaulas2

FROM cursos2

WHERE carga2 >35 AND totaulas2<30

ORDER BY carga2 ASC, totaulas2 ASC;

Comando para seleciona duas condições ou uma condição verdadeira

(nesse caso ele mostrara se as duas codições ou uma condição for verdadeira. Por Exemplo: Se carga2>35 E totaulas2<30 mostrará. Se carga2>35 E totaulas2>30 mostrara, agora se as duas codiçoes que coloquei não for verdadeira ele não mostrar nenhum, para mostrar tem que ser as duas verdadeiras ou apenas uma.)

SELECT nome2,carga2,totaulas2

FROM cursos2

WHERE carga2 >35 OR totaulas2<30;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** OR = Seleciona se as duas codições que você colocou for verdadeira ou apenas uma.

SELECT nome2,carga2,totaulas2

FROM cursos2

WHERE carga2 >35 OR totaulas2<30

ORDER BY carga2 ASC, totaulas2 ASC;

**Explicando melhor o uso de END e OR**

**p** **q** p **AND** q

V V V

V F F

F V F

F F F

**p** **q** p **OR** q

V V V

V F V

F V V

F F F

Aula12 – SELECT (Parte2)

Comando para selecionar algo que começa com uma letra

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'P%'**;

Comando para selecionar algo que Termina com uma letra

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'%A'**;

Comando para selecionar algo que tenha a letra A .Independente da posição

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'%A%'**;

Comando para selecionar algo que não tenha a letra A

SELECT \* FROM cursos2

WHERE nome2 **NOT** **LIKE**  **'%A%'**;

Comando para selecionar algo que começa com a letra ph e termina com a letra p.

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'ph%p'**;

Comando para selecionar algo que começa com a letra ph e termina com a letra p.

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'ph%p'**;

Comando para selecionar algo que começa com a letra ph podendo ter nada ou outra coisa depois da letra ph e que tenha alguma coisa de pois da letra p.

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'ph%p%'**;

Comando para selecionar algo que começa com a letra ph tenha qualquer coisa ou nada no meio e que depois obrigatoriamente tenha um caractere (letra ou numero) de pois da letra p.

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'ph%p\_'**;

Comando para selecionar algo que começa com a letra p de pois da letra p obrigatoriamente tenha um caractere (numero ou letra) e depois desse caractere tenha a letra p novamente e de pois da letra p tenha alguma coisa ou nada .

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'p\_p%'**;

Comando para selecionar algo que começa com a letra p de pois da letra p obrigatoriamente tenha dois caractere (numero ou letra) e depois desse caractere tenha a letra t e que de pois da letra t tenha alguma coisa ou nada .

SELECT \* FROM cursos2

WHERE nome2 **LIKE**  **'p\_\_t%'**;

Comando para selecionar distinguir algo .

SELECT DISTINCT nacionalidade2 FROM gafanhotos2;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** DISTINCT = Seleciona todos os registros que você selecionou mas de forma sem repetição.

SELECT DISTINCT nacionalidade2 FROM gafanhotos2

ORDER BY nacionalidade2 ASC;

Comando para contar quantidade

(Nesse caso ele ira contar (tudo) quantos cursos (que estão cadastrados) tem.

SELECT **COUNT**(\*) FROM cursos2;

(Nesse caso ele ira contar todos os cursos que estão na carga acima de 40).

SELECT **COUNT**(\*) FROM cursos2 WHERE carga2 >40;

Comando para mostrar o maior .

(nesse caso ele ira mostrar a maior carga do curso)

SELECT MAX(carga2) FROM cursos2 ;

(nesse caso ele ira mostrar o maior totaulas do curso no ano de 2016)

SELECT MAX(totaulas2) FROM cursos2 WHERE ano2 ='2016';

Comando para mostrar o menor .

(nesse caso ele ira mostrar a menor carga do curso)

SELECT MIN(carga2) FROM cursos2 ;

(nesse caso ele ira mostrar o menor totaulas do curso no ano de 2016)

SELECT MIN(totaulas2) FROM cursos2 WHERE ano2 ='2016';

Comando para somar:

(nesse caso ele vai somar o totaula2 da tabela cursos2)

SELECT SUM(totaulas2) FROM cursos2 WHERE ano2 = '2016';

Comando para tirar a media:

(nesse caso ele vai tirar a media do totaula2 da tabela cursos2)

SELECT AVG(totaulas2) FROM cursos2 WHERE ano2 = ‘2016’;

Aula12 – SELECT (Parte3)

Comando de agrupamento:

SELECT totaulas2, COUNT(\*) FROM cursos2

**GROUP BY** totaulas2

ORDER BY totaulas2;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** GROUP BY = Tem por finalidade de agrupar registros. (Criar um grupo para registro que contém mesmo valores.)

SELECT carga2,totaulas2 FROM cursos2

WHEREtotaulas2 = ‘30’

**GROUP BY** carga2;

Aula15 – CHAVE ESTRAGEITA JOIN

SELECT gafanhotos2.nome2,gafanhotos2.cursopreferido,cursos2.nome2,cursos2.ano2

FROM gafanhotos2 JOIN cursos2

ON cursos2.idcurso2 = gafanhotos2.cursopreferido;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** INNER JOIN - Somente vai fazer oque tem relações nas duas tabelas

SELECT gafanhotos2.nome2,gafanhotos2.cursopreferido,cursos2.nome2,cursos2.ano2

FROM gafanhotos2 LEFT JOIN cursos2

ON cursos2.idcurso2 = gafanhotos2.cursopreferido;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** LEFT JOIN - Somente vai fazer oque tem relações nas duas tabelas ,mas dando preferencia na tabela das esquerda nesse caso é a tabela gafanhotos.

SELECT gafanhotos2.nome2,cursos2.nome2,cursos2.ano2

FROM gafanhotos2 RIGHT JOIN cursos2

ON cursos2.idcurso2 = gafanhotos2.cursopreferido;

OU

SELECT g.nome2,c.nome2,c.ano2

FROM gafanhotos2 g2 RIGHT JOIN cursos2 c2

ON c2.idcurso2 = g2.cursopreferido;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OBS:** RIGHT JOIN - Somente vai fazer oque tem relações nas duas tabelas ,mas dando preferencia na tabela da direita nesse caso é a tabela .