**DOCUMENTAÇÃO BANCO DE DADOS MYSQL UNINTER**

* **AULA PRATICA 1**

**Estudo de Caso, Exemplo 1**

**Escola de Musica**

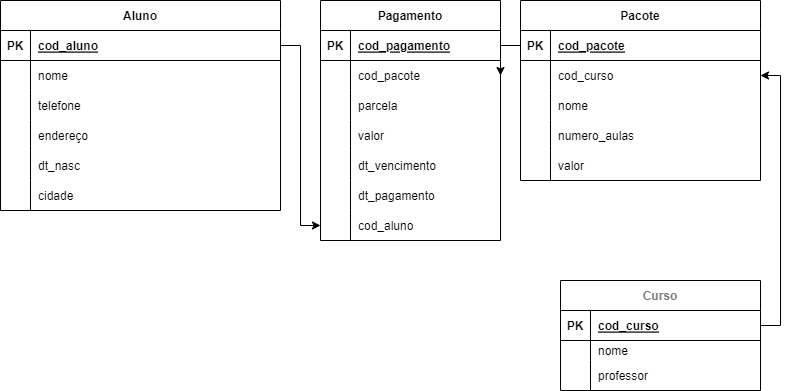
**A escola Saber Ouvir é uma pequena escola de música e possui apenas dois cursos, um para cada instrumento (violão e piano). Cada curso possui apenas um professor.**

**O aluno é cadastrado com os seguintes dados pessoais: nome, data de nascimento, telefone, endereço e cidade.**

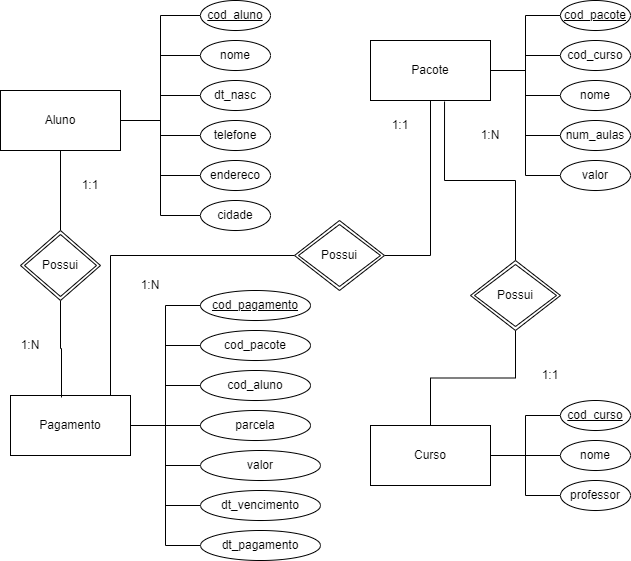
**A escola possui pacotes de aulas predefinidos, e em cada um deles é informado o curso, a opção de 10, 15 ou 20 aulas e o valor do pacote. O alunos escolhe qual pacote de aulas vai comprar e, quando compra um pacote, este pode ser parcelado em até quatro vezes.**

**O aluno tem acesso a um portal financeiro, por meio do qual é possível saber qual foi o pacote adquirido, a parcela, o valor, a data de vencimento e a data de pagamento.**

**Modelo Logico no Draw.io**

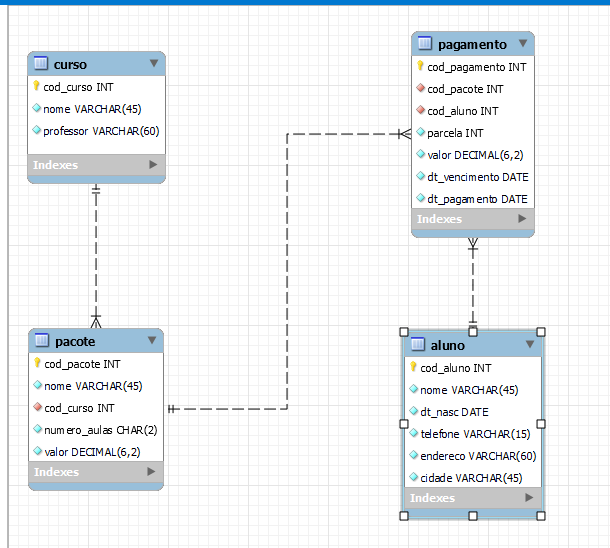
****

**Modelo Relaciona no Draw.io**

****

**MODELO FEITO NO MYSQL**

**Banco de dados fisico**

****

* **AULA PRATICA 2**

**LISTA DE EXERCICIOS:**

* **1. Criar uma base de dados chamada sistema**
* **2. Criar uma tabela chamada cad\_cidade com as seguintes colunas:**
* **codigo inteiro 2 AI PK**
* **nome varchar 30**
* **uf varchar 2**
* **3. Inserir os seguintes registros na tabela cad\_cidade:**
* **Curitiba, PR**
* **Rio de Janeiro, RJ**
* **Ponta Grossa, PR**
* **São Paulo, SP**
* **Ribeirão, SP**
* **4. Visualize a estrutura da tabela cad\_cidade**
* **5. Renomeie a tabela cad\_cidade para cidad**

* **6. Visualize todos os registros da tabela cidade**

* **7. Visualize só as cidades do estado de São Paulo**
* **8. Criar uma tabela chamada funcionário com as seguintes colunas:**
* **codigo inteiro 3 AI PK**
* **nome varchar 30**
* **endereco varchar 40**
* **numero inteiro 6**
* **salario decimal 6,2**
* **codigo\_cidade inteiro 2**
* **9. Adicione a coluna sexo (Char 1) na tabela funcionário**
* **10. Inserir os seguintes registros na tabela funcionário:**
* **Pedro, Rua Flores, 30, 1500.00, 2, M**
* **Maria, Av Brasil, 400, 1960.70, 1, F**
* **Jose, Rua do João, 759, 3800.00, 4, F**
* **Marco, Av Santa Rita, 2, 3450.50, 2, M**
* **11. Mostre somente a coluna nome e sexo dos funcionários**
* **12. Mostre somente os funcionários que possuem o salário maior que 2.000**

* **13. Mostre somente os funcionários do sexo masculino que ganham menos de 1.500**

* **14. Mostre uma lista dos funcionários ordenados pelo salário (primeiro quem ganha menos)**
* **15. Mostre todos os funcionários que recebem entre 2.000 e 3.000**
* **16. Listar os funcionários cujo nome começa com a letra M**
* **17. Listar os funcionários cujo nome contenha a letra A**
* **18. Listar os funcionários cujo nome NÃO começa com a letra M**

* **19. Criar uma nova coluna na tabela funcionário:**
* **setor varchar 10**
* **20. Atualizar a coluna setor com a palavra produção**
* **21. Atualizar a coluna setor do registro 3 para gerente**
* **22. Apagar os registros da tabela funcionário que possuem o código maior que 3**
* **23. Apagar os registros da tabela cidade que são do estado do MS**
* **24. Listar somente o nome dos funcionários e, ao lado, o nome da cidade**
* **25. Mostrar a média dos salários**
* **26. Mostrar a média dos salários agrupados por sexo**
* **27. Mostrar a soma dos salários**

* **28. Mostrar a quantidade de registros na tabela cidade**

**COMANDOS UTILIZADOS NO My SQL WORKBENCH**

**EM ORDEM DOS EXERCICIOS:**

**create database sistema;**

**use sistema;**

**create table cad\_cidade (codigo int(2) not null auto\_increment, nome varchar(30) not null, uf varchar(2) not null, primary key (codigo) );**

**show tables;**

**insert into cad\_cidade (nome, uf) values ('Curitiba', 'PR');**

**insert into cad\_cidade (nome, uf) values ('Rio de Janeiro', 'RJ');**

**insert into cad\_cidade (nome, uf) values ('Ponta Grossa', 'PR');**

**insert into cad\_cidade (nome, uf) values ('Sao Paulo', 'SP');**

**insert into cad\_cidade (nome, uf) values ('Ribeirao', 'SP');**

**select \* from cad\_cidade;**

**describe cad\_cidade;**

**alter table cad\_cidade rename cidade;**

**select \* from cidade;**

**select \* from cidade where uf='SP';**

**create table funcionario (codigo int(3) not null auto\_increment, nome varchar (30) not null, endereco varchar (40) not null, numero int(6) not null, salario decimal(6,2) not null, codigo\_cidade int(2) not null, primary key(codigo) );**

**alter table funcionario add sexo char(1);**

**describe funcionario;**

**insert into funcionario (nome, endereco, numero, salario, codigo\_cidade, sexo) values ('Pedro', 'Rua Flores', 30, 1500.00, 2, 'M');**

**insert into funcionario (nome, endereco, numero, salario, codigo\_cidade, sexo) values ('Maria', 'Av Brasil', 400, 1960.70, 1, 'F');**

**insert into funcionario (nome, endereco, numero, salario, codigo\_cidade, sexo) values ('Jose', 'Rua do Joao', 759, 3800.00, 4, 'F');**

**insert into funcionario (nome, endereco, numero, salario, codigo\_cidade, sexo) values ('Marco', 'Av Santa Rita', 2, 3450.50, 2, 'M');**

**select \* from funcionario;**

**select nome, sexo from funcionario;**

**select \* from funcionario where salario > 2000;**

**select \* from funcionario where sexo='M' and salario < 1500;**

**select \* from funcionario order by salario asc;**

**select \* from funcionario where salario >= 2000 and salario <= 3000;**

**select \* from funcionario where nome like 'M%';**

**select \* from funcionario where nome like '%A%';**

**select \* from funcionario where nome not like 'M%';**

**alter table funcionario add setor varchar(10);**

**describe funcionario;**

**update funcionario set setor='producao' where codigo >= 1 ;**

**select \* from funcionario;**

**update funcionario set setor='gerente' where codigo = 3;**

**delete from funcionario where codigo > 3;**

**delete from cidade where uf = 'MS';**

**select \* from cidade;**

**select funcionario.nome, cidade.nome from funcionario join cidade on funcionario.codigo\_cidade = cidade.codigo;**

**select avg(salario) as 'Media salarial' from funcionario;**

**select sexo, avg(salario) from funcionario group by sexo;**

**select sum(salario) from funcionario;**

**select count(\*) as 'qtd cidades' from cidade;**

**#posso usar as antes de um comando para ficar melhor sinalizado.**

* **AULA PRATICA 3**

**LISTA DE EXERCICIOS:**

* **1. Criar um banco de dados chamado aula**
* **2. Criar a tabela usuario (codigo, login e senha)**
* **3. Criar a tabela cidade (codigo, nome)**
* **4. Criar um usuário chamado gerencia com a senha 123**
* **5. Atribuir ao usuário gerencia o direito de visualizar os dados da tabela cidade**
* **6. Logar com o usuário novo (gerencia) e fazer uma consulta na tabela cidade, depois consultar a tabela usuario e tentar inserir algum dado na tabela usuario**
* **7. Logar novamente com o usuário principal e atribuir ao usuário gerenciar todos os direitos a tabela cidade do banco de dados aula**
* **8. Logar com o usuário gerencia e inserir a cidade “irati” na tabela cidade**
* **9. Atribuir ao usuário gerenciar todos os direitos ao banco de dados aula**
* **10. Logar com o usuário gerencia e testar os direitos**
* **11. Criar um índice na tabela cidade com a coluna código**
* **12. Criar um índice na tabela usuario com a coluna código**
* **13. Criar uma tabela resumo\_cadastro (id, nome, sexo) e criar um índice na coluna id.**
* **14. Remover os direitos do usuário gerencia**
* **15. Apagar o usuário gerencia**

**RESOLUÇAO:**

**show databases**

**create database aula;**

**use aula;**

**create table usuario (codigo int(3) not null auto\_increment, login varchar(30) not null, senha varchar(12) not null, primary key (codigo) );**

**show tables**

**create table cidade (codigo int(3) not null auto\_increment, nome varchar(45) not null, primary key (codigo) );**

**create user gerencia@localhost identified by '123';**

**grant select on aula.cidade to gerencia@localhost;**

**grant all privileges on aula.cidade to gerencia@localhost;**

**grant all privileges on aula.\* to gerencia@localhost;**

**# \* SIGNIFICA igual a TODOS OU TODAS (sem especificar)**

**revoke all privileges on aula.\* from gerencia@localhost;**

**drop user gerencia@localhost;**

* **DEPOIS DE CRIADO O USUÁRIO GERENCIA:**

**show databases;**

**use aula;**

**show tables;**

**select \* from cidade;**

**select \* from usuario;**

**insert into cidade (nome) values ('Irati');**

**insert into cidade (nome) values ('Ponta Grossa');**

**update cidade set nome = 'Ponta Grossa' where codigo = 2;**

**delete from cidade where codigo = 1;**

**describe usuario;**

**show tables;**

**insert into usuario (senha, login) values ('3456', 'joaquim');**

**insert into usuario (senha, login) values ('888', 'joao');**

**update usuario set senha = 1234 where codigo = 2;**

**describe cidade;**

**create index cod\_cidade on cidade (codigo) ;**

**create index cod\_usuario on usuario (codigo) ;**

**create table resumo\_cadastro (id int(4) not null auto\_increment, nome varchar(45) not null, sexo char(1) not null, primary key (id), index cadastro (id) );**

**describe resumo\_cadastro;**

* **AULA PRATICA 4**
* **Treinando stored procedures, functions e triggers**

**LISTA DE EXERCICIOS:**

* **1. Criar a stored procedure FUNC\_MINIMO para retornar o número de funcionários que ganham menos de x reais (passar como parâmetro o valor desejado)**
* **2. Criar a stored procedure FUNC\_SALARIO para retornar o salário de um funcionário específico**
* **3. Criar a stored procedure AUMENTO para aumentar 10% o salário dos funcionários**
* **4. Apagar a stored procedure FUNC\_SALARIO**
* **5. Criar a function SOMA para retornar a soma de dois números inteiros (passar por parâmetros)**
* **6. Criar function SALARIO para retornar o salário de um funcionário específico (passar por parâmetro)**
* **7. Criar uma function FUNC\_SEXO que passe o sexo como parâmetro e retorne o número de funcionários do sexo informado**
* **8. Apagar a function SOMA**
* **9. Criar a tabela backup com: codigo inteiro 3 AI PK nome varchar 30 salario decimal 6,2**
* **10. Criar uma trigger (FAZ\_BACKUP) para quando deletar um funcionário gravar o nome e o salario na tabela backup**
* **11. Criar a function num\_BACKUP para retornar quantos funcionários existem na tabela backup**
* **12. Criar a stored procedure NUM\_SEXO passando como parâmetro o sexo e retornar o número de funcionários daquele sexo (utilizar a condicional SE)**
* **13. Criar uma procedure passando como parâmetro o sexo e retornar os funcionários daquele sexo (utilizar a condicional SE)**
* **14. Apagar a trigger FAZ\_BACKUP**

**RESOLUÇAO:**

**show databases;**

**use aula;**

**create table funcionario (cod int(3) not null auto\_increment, nome varchar(45) not null, sexo char(1) not null, salario decimal(6,2) not null, primary key (cod) );**

**insert into funcionario values (null, 'Pedro', 'M', 7820.00);**

**insert into funcionario values (null, 'Maria', 'F', 9320.00);**

**insert into funcionario values (null, 'Joao', 'M', 2390.80);**

**insert into funcionario values (null, 'Marta', 'F', 3450.75);**

**select \* from funcionario;**

**delimiter $$**

**create procedure func\_minimo (sal decimal)**

**select count(\*) from funcionario where salario < sal;**

**$$**

**call func\_minimo(5000);**

**delimiter $$**

**create procedure func\_salario (id smallint)**

**select \* from funcionario where cod = id;**

**$$**

**call func\_salario(2);**

**delimiter $$**

**create procedure aumento()**

**update funcionario set salario = (salario \* 1.1) where cod > 0;**

**$$**

**call aumento();**

**drop procedure func\_salario;**

**delimiter $$**

**create function soma(a int, b int)**

**returns int**

**return a + b ;**

**$$**

**select soma(3,4);**

**select soma(8,2) as 'Soma';**

**delimiter $$**

**create function salario (id smallint)**

**returns decimal(6,2)**

**return (select salario from funcionario where cod = id);**

**$$**

**select salario (1) as 'Salario';**

**select \* from funcionario;**

**delimiter $$**

**create function func\_sexo (p\_sexo char)**

**returns int**

**return (select count(\*) from funcionario where sexo + p\_sexo);**

**$$**

**select func\_sexo ('M');**

**drop function soma;**

**create table backup (codigo int (3) not null auto\_increment, nome varchar (30) not null, salario decimal(7,2) not null, primary key (codigo) );**

**show tables;**

**delimiter $$**

**create trigger faz\_backup before delete**

**on fucionario**

**for each row**

**begin**

**insert into backup (nome, salario) values (old.nomes, old.salario);**

**end**

**$$**

**select \* from backup;**

**select \* from funciorario;**

**delete from funcionario where cod = 1;**

**delimiter $$**

**create function num\_backup()**

**returns int**

**return (select count(\*) from backup);**

**$$**

**delimiter $$**

**create procedure num\_sexo(p\_sexo char)**

**begin**

**if (p\_sexo = 'M') then**

**select count(\*) from funcionario where sexo = 'M';**

**elseif (p\_sexo = 'F') then**

**select count (\*) from funcionario where sexo = 'F';**

**end if;**

**end**

**$$**

**call num\_sexo ('M');**

**delimiter $$**

**create procedure ver\_func (p\_sexo char)**

**begin**

**if (p\_sexo = 'M') then**

**select \* from funcionario where sexo = 'M';**

**else if (p\_sexo = 'F') then**

**select \* from funcionario where sexo= 'F';**

**end if;**

**end**

**$$**

**call ver\_func('M');**

**call ver\_func('F');**

**drop trigger faz\_backup;**