

Professores: Calveti e Igor

Aula: 12

Assunto: Revisão do Banco de Dados MySql.

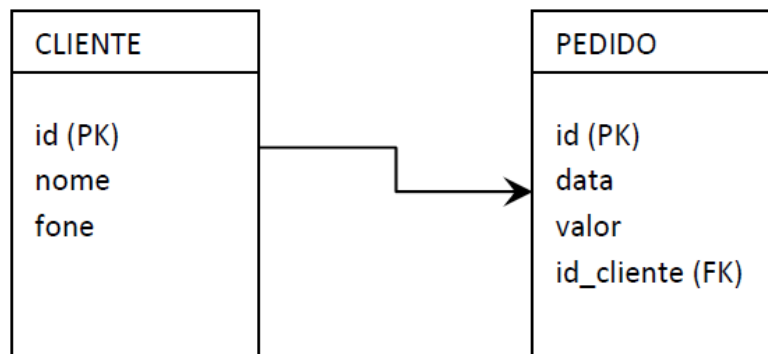
Conceitos Básicos abordados pelo Professor:

- *Banco de Dados MySql;*
- *Entidades;*
- *Databases;*
- *Tabelas;*
- *Dados; e*
- *Comandos Sql.*

Exemplos:

O banco de dados *MySql* é uma ferramenta simples e funcional para que se possam colocar em prática todos os conceitos vistos nas aulas teóricas/práticas da disciplina de Banco de Dados.

Para ilustrar todos os conceitos e operações, será utilizado um modelo de dados muito simples, composto por duas entidades: CLIENTE e PEDIDO.



Para tal, deve-se seguir e realizar todos os passos listados abaixo:

1- Visualizar as *databases* existentes:

```
SHOW DATABASES;
```

2- Criar uma *database* chamada TUTORIAL:

```
CREATE DATABASE TUTORIAL;
```

3- Visualizar novamente as *databases* existentes e selecionar a TUTORIAL para uso:

```
SHOW DATABASES;  
USE TUTORIAL;
```

4- Criar a tabela CLIENTE, conforme especificado no modelo:

```
CREATE TABLE CLIENTE
(
  id SMALLINT UNSIGNED NOT NULL,
  nome VARCHAR(60) NOT NULL,
  fone CHAR(10),
  PRIMARY KEY (id)
) ENGINE = InnoDB;
DESCRIBE CLIENTE;
```

5- Criar a tabela PEDIDO, conforme especificado no modelo (observar a Foreign Key):

```
CREATE TABLE PEDIDO
(
  id SMALLINT UNSIGNED NOT NULL,
  data DATE NOT NULL,
  valor NUMERIC (10, 2) NOT NULL,
  id_cliente SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (id)
) ENGINE = InnoDB;
ALTER TABLE PEDIDO ADD FOREIGN KEY (id_cliente)
REFERENCES CLIENTE (id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
DESCRIBE PEDIDO;
```

6- Inserir uma linha de dados na tabela CLIENTE:

```
INSERT INTO cliente VALUES (101, 'Zé das Couves', '1127991999');
```

7- Inserir várias linhas de dados na tabela CLIENTE:

```
INSERT INTO CLIENTE VALUES
(102, 'Maria das Couves', NULL),
(103, 'Antonio das Couves', '1127991999'),
(104, 'Ana das Couves', '1121213434');
```

8- Recuperar todos os dados da tabela CLIENTE

```
SELECT * FROM cliente;
```

9- Recuperar todos os dados da tabela CLIENTE, em ordem alfabética, pelo nome:

```
SELECT * FROM cliente ORDER BY nome;
```

10- Recuperar os dados dos clientes para os quais não foi atribuído telefone:

```
SELECT id, nome FROM CLIENTE WHERE fone IS NULL;
```

11- Recuperar os dados dos clientes para os quais foi atribuído telefone:

```
SELECT id, nome FROM CLIENTE WHERE fone IS NOT NULL;
```

12- Recuperar os dados dos clientes cujo nome começa com a letra 'A':

```
SELECT * FROM CLIENTE WHERE nome LIKE 'A%';
```

13- Recuperar os dados dos clientes cujo sobrenome é 'Couves':

```
SELECT * FROM CLIENTE WHERE nome LIKE '%Couves';
```

14- Atualizar o telefone do cliente cujo código é 102:

```
UPDATE CLIENTE SET fone = '1121212121' WHERE id = 102;
```

15- Excluir o cliente de código 104:

```
DELETE FROM CLIENTE WHERE id = 104;
```

16- Inserir uma linha de dados na tabela PEDIDO:

```
INSERT INTO PEDIDO VALUES (1001, '2010-06-15', 300.00, 102);
```

17- Execute o comando abaixo e verifique o motivo do erro apresentado (deve ser apresentado um erro de referência da chave estrangeira, já que o cliente de código 104 foi excluído):

```
INSERT INTO PEDIDO VALUES (1002, '2010-06-15', 1250.00, 104);
```

18- Execute o comando abaixo e verifique o motivo do erro apresentado (deve ser apresentado um erro pela tentativa de inserir um pedido com valor NULO):

```
INSERT INTO PEDIDO VALUES (1002, '2010-06-15', NULL, 101);
```

19- Execute o comando abaixo para corrigir os dois problemas anteriores e fazer a inserção normalmente:

```
INSERT INTO PEDIDO VALUES (1002, '2010-06-15', 1250.00, 101);
```

20- Insira vários pedidos de uma só vez:

```
INSERT INTO PEDIDO VALUES  
(1003, '2010-06-16', 200.00, 102),  
(1004, '2010-06-16', 400.00, 101),  
(1005, '2010-06-17', 320.00, 101),  
(1006, '2010-06-17', 780.00, 102),  
(1007, '2010-06-17', 150.00, 102),  
(1008, '2010-06-17', 390.00, 101),  
(1009, '2010-06-18', 650.00, 102),  
(1010, '2010-06-18', 290.00, 101),  
(1011, '2010-06-18', 100.00, 102),  
(1012, '2010-06-19', 315.00, 101),  
(1013, '2010-06-19', 738.00, 101),  
(1014, '2010-06-20', 472.00, 102);
```

21- Execute o comando abaixo e verifique o motivo do erro apresentado (deve ser apresentado um erro por quebra de integridade referencial, haja vista que o cliente de código 101 possui vários pedidos associados a ele):

```
DELETE CLIENTE WHERE id = 101;
```

22- Verifique quantos pedidos foram cadastrados:

```
SELECT count(*) AS total FROM PEDIDO;
```

23- Quais foram os pedidos efetuados a partir do dia 16 de junho:

```
SELECT * FROM PEDIDO WHERE data > '2010-06-16';
```

24- Determine o maior, a média e o menor valor dos pedidos cadastrados:

```
SELECT max(valor) AS maior, avg(valor) AS media,  
min(valor) AS menor FROM PEDIDO;
```

25- Relacione todos os dados de CLIENTE e PEDIDO:

```
SELECT * FROM CLIENTE, PEDIDO  
WHERE CLIENTE.id = PEDIDO.id_cliente;
```

26- Selecione algumas colunas do resultado apresentado anteriormente:

```
SELECT c.id, c.nome, p.data, p.valor FROM CLIENTE c, PEDIDO p
WHERE c.id = p.id_cliente;
```

27- Uma outra sintaxe (padrão ANSI 92) para efetuar a mesma consulta:

```
SELECT * FROM
CLIENTE c INNER JOIN PEDIDO p ON c.id = p.id_cliente;
```

28- Qual a diferença entre INNER JOIN e LEFT JOIN?

(o LEFT JOIN acrescentará ao resultado os clientes que, por ventura, não tenham nenhum pedido a eles associado):

```
SELECT * FROM
CLIENTE c LEFT JOIN PEDIDO p ON c.id = p.id_cliente;
```

29- Qual a diferença entre LEFT JOIN e RIGHT JOIN?

(a diferença está no sentido da declaração. A declaração LEFT ou RIGHT indica qual é a tabela que deve ser considerada para a complementação do resultado. O segundo comando apresentado abaixo produz o mesmo resultado que o comando apresentado no item anterior).

```
SELECT * FROM
CLIENTE c RIGHT JOIN PEDIDO p ON c.id = p.id_cliente;
SELECT * FROM
PEDIDO p RIGHT JOIN CLIENTE c ON p.id_cliente = c.id;
```

30- Para finalizar, montar uma Sub-Query:

```
SELECT c.nome, p.data, p.valor
FROM CLIENTE c INNER JOIN PEDIDO p ON c.id = p.id_cliente
WHERE p.valor = (SELECT max(valor) FROM PEDIDO);
```

Atividade Prática:

Dados os exemplos vistos acima, elaborar um banco de dados composto por:

<u>Tabelas</u>	<u>Campos</u>
PESSOAL	Id(PK); CPF; RG; Nome; DataNascimento;
ACADEMICO	Id(PK) RA; Nome; NotaTeoria; NotaLab; Media; CPF(FK);