1.Criando Banco De Dados e Tabelas

Criando um B.D(Banco de Dados):

Create Database Escola;
^^ ^^
Cria um B.D Nome do B.D

Criando uma tabela dentro do B.D criado:

Use Escola;

Seleciona um B.D de sua escolha(no caso, Escola)

Create Table Aluno(
Nome Varchar(50) not null);

Cria uma Tabela, no caso chamada de *"Aluno"*, por enquanto só possui um atributo chamado "*Nome*".

2.Tipos de Atributos:

Int:Valores inteiros de do máximo tamanho 11

Varchar:Valores strings, textos, no máximo de tamanho 255.

Double: valores reais, não possui limite de tamanho.

Date: armazena valores de data no formato (Ano/Mês/dia)

Primary Key:Chave primária, define a diferença entre os dados registrados em cada tabela, por exemplo, 2 alunos não possuem a mesma matricula no mesmo ano.

Foreign Key:Serve para relacionar 2 tabelas, por exemplo o Código que identifica uma Turma em uma escola está presente na tabela do Aluno.

Not null:Define o atributo como obrigatório a se preencher(chaves primárias já são consideradas not null por padrão)

Auto increment: Valor que automaticamente soma 1 a cada valor registrado, por exemplo supunha que Fulano é o primeiro aluno matriculado, ele vai ter matricula de numero 1, já o segundo aluno vai ter matricula 2 automaticamente.

Utilizando cada atributo em uma tabela

Create table Teste(
Inteiro int(11) not null,
Real double not null,
Texto varchar(20) not null,
Data date not null,
Cod int(1) primary key auto_increment);

Utilizando chaves estrangeiras(Foreign Key):

```
create table Aluno(
   Nome varchar(50) not null,
   Matricula int(8) primary key,
   Idade int(2) not null,
   TurmaCod int(2) not null,
   Foreign key(TurmaCod) references Turma(cod) );

create table Turma(
   curso varchar(20) not null,
   cod int(2) primary key,
   serie varchar(2) not null );
```

Uma turma tem vários alunos, mas 1 aluno só pode ser de 1 turma, então numa relação de 1:N, o que for 1 possui a chave estrangeira.

Passo a passo:

1-Cria-se um atributo de mesmo tipo e tamanho dentro da tabela que for conter a Foreign Key

TurmaCod int(2) not null,

2-Utiliza-se o comando da chave estrangeira

Foreign key(nome do atributo do passo 1) references Tabela em que será pega a chave estrangeira(nome da chave primária dessa tabela)

Foreign key(TurmaCod) references Turma(cod)

3-A chave estrangeira sempre será uma CHAVE PRIMÁRIA de outra tabela

3.Inserindo valores em uma tabela:

Insert into Tabela (coluna1,coluna2) values (valor1, valor2);

Nesse modelo é possivel especificar quais colunas preencher em uma tabela, mas caso queira preencher todas de uma vez:

Insert into tabela values(valordacoluna1,valordacoluna2);

Esses valores vão ser preenchidos na ordem em que a tabela está organizada

Exemplos:

insert into turma values ("informatica",4,"2ª");

insert into turma (cod,serie) values (4,"2ª");

Lembrando que valores que forem Varchar precisam estar entre aspas

4.Exibindo valores de uma tabela:

Select * from tabela;

Mostra todos os valores registrados em uma tabela

Exemplo:

Select * from Aluno;

Mostrando valores de colunas especificas:
Select coluna1,coluna2 from tabela;

Exemplo:
Select nome,matricula from aluno;

Mais usos do Select:

Select count(coluna) from tabela; <= Conta quantos valores foram registrados nessa coluna(retorna um numero)

Select sum(coluna) from tabela; <=Soma os valores de todas as colunas registradas

Select max(coluna) from tabela <=Seleciona o maior valor dentre as colunas

Select min(coluna) from tabela <= Seleciona o menor valor dentre as colunas

Select coluna from tabela where valor=x; <=Só mostra valores das colunas se cumprir alguma condição
Exemplo:
select nome from aluno where matricula=1;
select nome from aluno where nome="Fulano";
select nome from aluno where matricula<2;

Select coluna from tabela order by coluna asc; <=Organiza os valores de acordo com a mesma ou outra coluna de modo crescente
Exemplo:
select nome,idade from aluno order by idade asc;
Mostra o nome e a idade dos alunos de forma crescente de acordo com a idade.

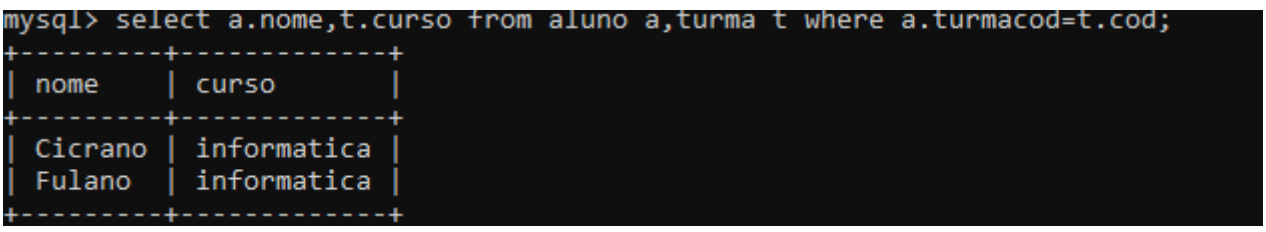
Select coluna from tabela order by coluna desc; <=Organiza os valores de acordo com a mesma ou outra coluna de modo decrescente
Exemplo:
select nome,idade from aluno order by idade asc;
Mostra o nome e a idade dos alunos de forma decresnte de acordo com a idade.

Select avg(coluna) from tabela; <= Realiza uma média simples dos valores das colunas.

Juntando 2 tabelas através das chaves estrangeiras:

select a.nome,t.curso from aluno a,turma t where a.turmacod=t.cod;

Primeiro, dê nome as tabelas que for juntar, Aluno é "a" e Turma é "t", então vou utilizar o nome de a e o curso de t, e só vai ser exibido valores quando a chave estrangeira do aluno for igual a chave primária da turma.



5.Atualizando dados da tabela:

Update <= Utilizado para atualizar dados de uma coluna, geralmente utilizado com o where para especificar onde mudar os dados.

Update tabela set coluna1="novo valor",coluna2=1 where coluna3=2;
Exemplo:
update aluno set nome="Novo nome" where matricula=1;

```
mysql> update aluno set nome="Novo nome" where matricula=1;
Query OK, 1 row affected (0.31 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from aluno;
+-----------+-----------+-------+----------+
| Nome      | Matricula | Idade | TurmaCod |
+-----------+-----------+-------+----------+
| Novo nome |         1 |    15 |        4 |
| Fulano    |         2 |    20 |        4 |
+-----------+-----------+-------+----------+
```

Caso não houvesse o where, todos os alunos cadastrados teriam seu nome modificado

6.Deletando dados da tabela e banco de dados

Drop <= Deleta uma tabela do banco de dados ou o proprio banco de dados
Exemplo:

Drop table aluno; <= Deleta a tabela aluno
Drop databse escola; <=Deleta todo o banco de dados

Delete <= Deleta dados de uma tabela, deve ser usado com o where, a não ser que o usuáio queria que todos os dados da tabela sejam apagados

Delete from aluno;
Delete from aluno where matricula=1;

7.Entendendo melhor os operadores condicionais

WHERE<= "Onde", procura um valor especificado pelo usuário, pode ser utilizado além do Select

LIKE<=Usado junto com o where, procura valores dependendo de como for usado
Exemplo:

Where Nome like "B%" <=Se o nome começar com "B", a condição será verdadeira.
Where Nome like "%B" <=Se o nome terminar com "B", a condição será verdadeira
Where Nome like "%B%" <=Se o nome ter "B" no inicio, meio ou fim, a condição será verdadeira