

Banco de Dados



SQL – DDL

Data Definition Language

Linguagem de Definição de Dados



MySQL – Data Definition Language



Tipos de Dados



MySQL – Data Definition Language



| Tipos Numéricos | | | | |
|-----------------|-------------------------------|----------|----------------------|----------------------|
| Tipo | Uso | | Tamanho | |
| | | Atributo | MIN | MAX |
| TINYINT | Um inteiro muito pequeno | Signed: | -128 | 127 |
| | | Unsigned | 0 | 255 |
| SMALLINT | Um inteiro pequeno | Signed: | -32768 | 32767 |
| | | Unsigned | 0 | 65535 |
| MEDIUMINT | Um inteiro de tamanho mediano | Signed: | -8388608 | 8388607 |
| | | Unsigned | 0 | 16777215 |
| INT or INTEGER | Um inteiro de tamanho normal | Signed: | -2147483648 | 2147483647 |
| | | Unsigned | 0 | 4294967295 |
| BIGINT | Um inteiro de tamanho grande | Signed: | -9223372036854775808 | 9223372036854775807 |
| | | Unsigned | 0 | 18446744073709551615 |



MySQL – Data Definition Language



| | | | | |
|---|---|-----------------------|--|--|
| FLOAT | Um pequeno número de ponto flutuante (precisão simples) | Signed | -3.402823466E+38 | -1.175494351E-38, 0 |
| | | | 1.175494351E-38 | 3.402823466E+38 |
| | | Não pode ser unsigned | - | |
| | | OBS | Se o número de decimais não for especificado ou for ≤ 24 será de precisão simples | |
| DOUBLE, DOUBLE PRECISION, REAL | Um número de ponto flutuante de tamanho normal (precisão dupla) | Signed | -1.7976931348623157E+308 | -2.2250738585072014E-308, 0 |
| | | | 2.2250738585072014E-308 | 1.7976931348623157E+308 |
| | | Não pode ser unsigned | - | |
| | | OBS | Se o número de decimais não for especificado ou for $25 \leq \text{Decimais} \leq 53$ será de precisão dupla | |
| DECIMAL, NUMERIC | Um número de ponto flutuante descompactado . | Signed | Se comporta como um campo CHAR: "descompactado" significa que o número é armazenado como uma string, usando um caractere para cada dígito do valor. O ponto decimal e, para números negativos, o sinal '-' não é contado. Se o decimal for 0, os valores não terão ponto decimal ou parte fracionária. | O alcance máximo de valores decimais é o mesmo que para o DOUBLE, mas a faixa atual para um campo DECIMAL dado pode ser limitado pela escolha de comprimento e decimais. |
| | | Não pode ser unsigned | - | |
| | | OBS | Se Decimais é deixado de fora ele é definido como 0. Se o comprimento é deixado de fora ele é definido como 10. Note que no MySQL 3,22 o comprimento inclui o sinal eo ponto decimal | |



MySQL – Data Definition Language



| Campos de Datas | | | | |
|-----------------|---|-----------|--|-----------------------|
| | | Formato | MIN | MAX |
| DATE | Data | | '1000-01-01' | '9999-12-31' |
| | | OBS | Formato: 'YYYY-MM-DD' | |
| DATETIME | Data e horário | | '1000-01-01 00:00:00' | '9999-12-31 23:59:59' |
| | | OBS | Formato: 'YYYY-MM-DD HH:MM:SS' | |
| TIMESTAMP | Timestamp | | '1970-01-01 00:00:00' | aproximadamente 2037 |
| | | OBS | Formato: YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD ou YYMMDD, dependendo se M é 14 (ausente), 12, 8 ou 6, podendo ser strings ou números. Este tipo é recomendável para instruções de INSERT ou UPDATE pois é automaticamente marcado com os valores da operação mais recente quando não informado. | |
| TIME | A time | | '-838:59:59' | '838:59:59' |
| | | OBS | formato: 'HH:MM:SS', podem ser strings ou números | |
| YEAR | Anos com 2 ou 4 dígitos. O padrão é 4 dígitos | 4 dígitos | 1901 | 2155 e 0000 |
| | | 2 dígitos | 1970 | 2069 |
| | | OBS | Formato: YYYY podem ser strings ou números. | |



MySQL – Data Definition Language



| Campos Texto | | | | |
|--------------|--|-----|---|--|
| | | | MIN | MAX |
| CHAR | String de tamanho fixo. Sempre é completada com espaços a direita até o tamanho definido | | 1 | 255 caracteres |
| | | OBS | Espaços excessivos são removidos quando o valor é trazido. Os valores são ordenados e comparados ignorando caixas altas e baixas de acordo com a codificação padrão, a menos que seja fornecido uma chave binária. | |
| VARCHAR | String de tamanho variável | | Values in <code>VARCHAR</code> columns are variable-length strings. The length can be specified as a value from 0 to 65,535. The effective maximum length of a <code>VARCHAR</code> is subject to the maximum row size (65,535 bytes, which is shared among all columns) and the character set used. See Section 8.4.7, “Limits on Table Column Count and Row Size” . | |
| | | OBS | | |
| TINYTEXT | | | 0 | 255 ($2^8 - 1$) caracteres |
| TEXT | | | 0 | 65535 ($2^{16} - 1$) caracteres |
| MEDIUMTEXT | | | 0 | 16777215 ($2^{24} - 1$) caracteres |
| LONGTEXT | | | 0 | 4294967295 ($2^{32} - 1$) caracteres |



MySQL – Data Definition Language



| Dados Binários | | | | |
|----------------|------------|--|--|--|
| TINYBLOB | | | 0 | 255 ($2^8 - 1$) caracteres |
| BLOB | | | 0 | 65535 ($2^{16} - 1$) caracteres |
| MEDIUMBLOB | | | 0 | 16777215 ($2^{24} - 1$) caracteres |
| LOB | | | 0 | 4294967295 ($2^{32} - 1$) caracteres |
| Listas | | | | |
| | | | MIN | MAX |
| ENUM | Enumeração | | String que pode conter apenas um valor ou zero | 65535 valores distintos. |
| SET | Lista | | String que pode conter zero ou mais valores | 64 itens |



MySQL – Data Definition Language



```
1 # String de conexão a ser executada no terminal
2 mysql -h localhost -u root -p
3
4 # Comando DDL para criação de um banco de dados
5
6 CREATE DATABASE uscsDB;
7
8 /* Comandos de controle do SGBD MySQL*/
9
10 # Responsável por listar os bancos de dados
11
12 SHOW DATABASES;
13
14 # Selecionar o banco de dados de trabalho
15
16 USE uscsDB;
17
18 # Exibir uma listagem das tabelas contidas no banco de dados
19
20 SHOW TABLES uscsDB;
21
22 # Detalhamento da tabela Aluno contida no banco de dados uscsDB
23
24 DESCRIBE Aluno;
```





MySQL – Data Definition Language



```
1  /*USCS - Banco de Dados*/
2
3  CREATE TABLE Aluno (
4      idAluno TINYINT NOT NULL,
5      PRIMARY KEY (idAluno)
6  );
```

| Tipos Numéricos | | | | |
|-----------------|--------------------------|----------|---------|-----|
| Tipo | Uso | | Tamanho | |
| | | Atributo | MIN | MAX |
| TINYINT | Um inteiro muito pequeno | Signed: | -128 | 127 |
| | | Unsigned | 0 | 255 |

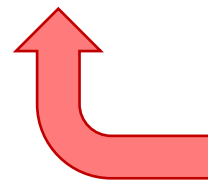
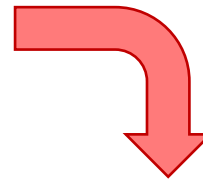


MySQL – Data Definition Language



```
1  /*USCS - Banco de Dados*/  
2  
3  CREATE TABLE Aluno (  
4      idAluno TINYINT NOT NULL,  
5      PRIMARY KEY (idAluno)  
6  );
```

```
8▼ CREATE TABLE Aluno (  
9      idAluno TINYINT(2) NOT NULL,  
10     PRIMARY KEY (idAluno)  
11 );
```



```
13▼ CREATE TABLE Aluno (  
14     idAluno TINYINT(2) SIGNED NOT NULL,  
15     PRIMARY KEY (idAluno)  
16 );
```



MySQL – Data Definition Language



```
18▼ CREATE TABLE Aluno (  
19     idAluno TINYINT(2) UNSIGNED NOT NULL,  
20     PRIMARY KEY (idAluno)  
21 );
```

| Tipos Numéricos | | | | |
|-----------------|--------------------------|----------|---------|-----|
| Tipo | Uso | | Tamanho | |
| | | Atributo | MIN | MAX |
| TINYINT | Um inteiro muito pequeno | Signed: | -128 | 127 |
| | | Unsigned | 0 | 255 |



MySQL – Data Definition Language



```
23 CREATE TABLE Aluno (  
24     idAluno TINYINT(2) UNSIGNED ZEROFILL NOT NULL,  
25     PRIMARY KEY (idAluno)  
26 );
```

| | idAluno |
|---|---------|
| ▶ | 02 |
| | 06 |
| | 87 |
| | 102 |



MySQL – Data Definition Language



```
23 CREATE TABLE Aluno (  
24     idAluno TINYINT(3) UNSIGNED ZEROFILL NOT NULL,  
25     Nome VARCHAR(45) NOT NULL,  
26     Estado CHAR(2),  
27     PRIMARY KEY (idAluno)  
28 );
```

| Campos Texto | | | | |
|--------------|--|-----|---|----------------|
| | | | MIN | MAX |
| CHAR | String de tamanho fixo. Sempre é completada com espaços a direita até o tamanho definido | | 1 | 255 caracteres |
| | | OBS | Espaços excessivos são removidos quando o valor é trazido. Os valores são ordenados e comparados ignorando caixas altas e baixas de acordo com a codificação padrão, a menos que seja fornecido uma chave binária. | |
| VARCHAR | String de tamanho variável | | 1 | 255 caracteres |
| | | OBS | Os valores são ordenados e comparados ignorando caixas altas e baixas de acordo com a codificação padrão, a menos que seja fornecido uma chave binária. Nota: Espaços excessivos são removidos quando o valor é inserido. | |



MySQL – Data Definition Language



| Value | CHAR (4) | Storage Required | VARCHAR (4) | Storage Required |
|------------|----------|------------------|-------------|------------------|
| ' ' | ' ' | 4 bytes | ' ' | 1 byte |
| 'ab' | 'ab ' | 4 bytes | 'ab' | 3 bytes |
| 'abcd' | 'abcd' | 4 bytes | 'abcd' | 5 bytes |
| 'abcdefgh' | 'abcd' | 4 bytes | 'abcd' | 5 bytes |

```
23 CREATE TABLE Aluno (  
24     idAluno TINYINT(3) UNSIGNED ZEROFILL NOT NULL,  
25     Nome VARCHAR(45) NOT NULL,  
26     Estado CHAR(2),  
27     PRIMARY KEY (idAluno)  
28 );
```



MySQL – Data Definition Language



```
23 CREATE TABLE Aluno (  
24     idAluno TINYINT(3) UNSIGNED ZEROFILL NOT NULL,  
25     Nome VARCHAR(45) NOT NULL,  
26     Estado CHAR(2),  
27     Peso DECIMAL(5,2),  
28     PRIMARY KEY (idAluno)  
29 );
```





MySQL – Data Definition Language



```
31 ALTER TABLE Aluno
32 ADD email VARCHAR(80) NOT NULL AFTER Estado;
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ▶ | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Email | varchar(80) | NO | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |





MySQL – Data Definition Language



```
34 ALTER TABLE Aluno
35 ADD NomeMae VARCHAR(45),
36 ADD NomePai VARCHAR(45);
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ► | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Email | varchar(80) | NO | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |
| | NomeMae | varchar(45) | YES | | NULL | |
| | NomePai | varchar(45) | YES | | NULL | |





MySQL – Data Definition Language



```
63 ALTER TABLE Aluno
64 ADD DataNascimento DATE,
65 ADD DataMatricula DATETIME,
66 ADD DataAtualizacaoRegistro TIMESTAMP;
```

| | | Formato | MIN | MAX |
|-----------|----------------|---------|--|-----------------------|
| DATE | Data | | '1000-01-01' | '9999-12-31' |
| | | OBS | Formato: 'YYYY-MM-DD' | |
| DATETIME | Data e horário | | '1000-01-01 00:00:00' | '9999-12-31 23:59:59' |
| | | OBS | Formato: 'YYYY-MM-DD HH:MM:SS' | |
| TIMESTAMP | Timestamp | | '1970-01-01 00:00:00' | aproximadamente 2037 |
| | | OBS | Formato: YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD ou YYMMDD, dependendo se M é 14 (ausente), 12, 8 ou 6, podendo ser strings ou números. Este tipo é recomendável para instruções de INSERT ou UPDATE pois é automaticamente marcado com os valores da operação mais recente quando não informado. | |

| Field | Type | Null | Key | Default |
|-------------------------|------------------------------|------|-----|-------------------|
| idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL |
| Nome | varchar(45) | NO | | NULL |
| Estado | char(2) | YES | | NULL |
| email | varchar(80) | NO | | NULL |
| Peso | decimal(5,2) | YES | | NULL |
| NomeMae | varchar(40) | NO | | NULL |
| NomePai | varchar(40) | YES | | NULL |
| DataNascimento | date | YES | | NULL |
| DataMatricula | datetime | YES | | NULL |
| DataAtualizacaoRegistro | timestamp | NO | | CURRENT_TIMESTAMP |





MySQL – Data Definition Language



| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ▶ | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | email | varchar(80) | NO | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |
| | NomeMae | varchar(45) | YES | | NULL | |
| | NomePai | varchar(45) | YES | | NULL | |

```
54 ALTER TABLE Aluno
55 CHANGE NomeMae NomeMae VARCHAR(40) NOT NULL,
56 CHANGE NomePai NomePai VARCHAR(40);
```





MySQL – Data Definition Language



| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ▶ | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | email | varchar(80) | NO | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |
| | NomeMae | varchar(40) | NO | | NULL | |
| | NomePai | varchar(40) | YES | | NULL | |

```
58 ALTER TABLE Aluno
59 CHANGE Email EmailAluno VARCHAR(80);
```





MySQL – Data Definition Language



```
38 ALTER TABLE Aluno DROP Email;
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ► | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |
| | NomeMae | varchar(45) | YES | | NULL | |
| | NomePai | varchar(45) | YES | | NULL | |



MySQL – Data Definition Language



```
38 ALTER TABLE Aluno
39 DROP Email,
40 DROP NomePai; |
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ► | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |
| | NomeMae | varchar(45) | YES | | NULL | |





MySQL – Data Definition Language



```
38▼ CREATE TABLE Aluno (  
39     idAluno TINYINT(3) UNSIGNED ZEROFILL NOT NULL,  
40     Nome VARCHAR(45) NOT NULL,  
41     Estado CHAR(2),  
42     Peso DECIMAL(5,2)  
43 );
```

O que tem de errado aqui?



| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ► | idAluno | tinyint(3) unsigned zerofill | NO | | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |



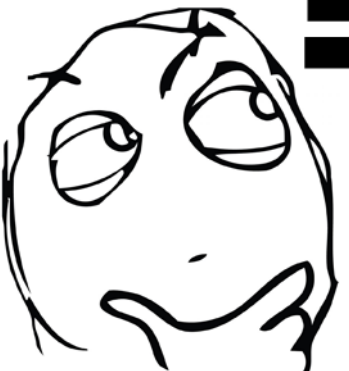
MySQL – Data Definition Language



```
38▼ CREATE TABLE Aluno (  
39     idAluno TINYINT(3) UNSIGNED ZEROFILL NOT NULL,  
40     Nome VARCHAR(45) NOT NULL,  
41     Estado CHAR(2),  
42     Peso DECIMAL(5,2)  
43 );
```

O que tem de errado aqui?

???



| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ▶ | idAluno | tinyint(3) unsigned zerofill | NO | | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |



MySQL – Data Definition Language



```
46 ALTER TABLE Aluno
47 ADD PRIMARY KEY (idAluno);
```

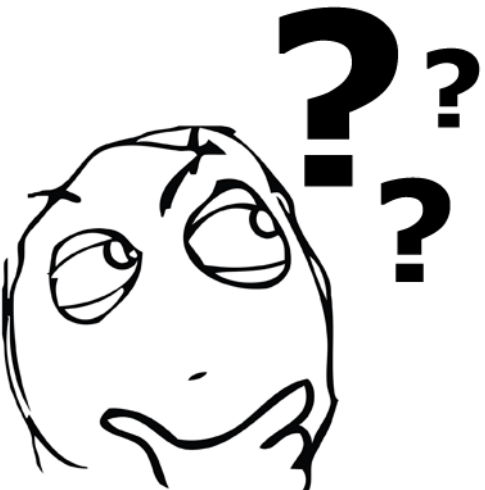
| | Field | Type | Null | Key | Default | Extra |
|---|---------|------------------------------|------|-----|---------|-------|
| ► | idAluno | tinyint(3) unsigned zerofill | NO | PRI | NULL | |
| | Nome | varchar(45) | NO | | NULL | |
| | Estado | char(2) | YES | | NULL | |
| | Peso | decimal(5,2) | YES | | NULL | |



MySQL – Data Definition Language

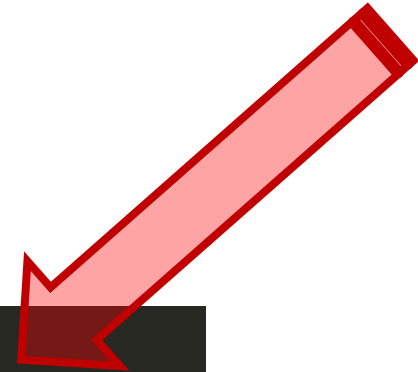


Posso facilitar minha vida se por acaso o idAluno for um código numérico sequencial?





MySQL – Data Definition Language



```
69▼ CREATE TABLE Aluno (  
70     idAluno INT(10) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,  
71     Nome VARCHAR(45) NOT NULL,  
72     PRIMARY KEY (idAluno)  
73 );
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|---------------------------|------|-----|---------|----------------|
| ▶ | idAluno | int(10) unsigned zerofill | NO | PRI | NULL | auto_increment |
| | Nome | varchar(45) | NO | | NULL | |



MySQL – Data Definition Language



```
48  
49 DROP TABLE Aluno;  
50
```





MySQL – Data Definition Language



```
50  
51 DROP DATABASE uscsDB;
```





MySQL – Data Definition Language



Exercício – Primeira Parte

Deseja-se projetar um BD para registro de pessoas físicas:

- a) Cada pessoa deve possuir nome, data de nascimento, nome do pai, nome da mãe, CPF, data e hora de cadastro, estado de nascimento e sexo;
- b) Sendo opcional o registro do nome do pai e estado de nascimento.
- c) É necessário também armazenar no banco de dados registros referentes a profissões, nacionalidades e etnias, sendo que cada um deva possuir um código de identificação único e sequencial e sua descrição.

Realizar a modelagem utilizando o Workbench e informar os comandos de DDL necessário para criação do banco de dados e suas tabelas.



MySQL – Data Definition Language



Exercício – Segunda Parte

Deseja-se alterar a estrutura do banco de dados criado para atender os seguintes requisitos:

- a) O nome do pai deve ser obrigatório.
- b) As tabelas de profissões, nacionalidades deve conter a data e hora de criação dos registros.
- c) A tabela de etnia não será mais necessária.
- d) O atributo estado de nascimento deve ser renomeado para UFNascimento.

Realizar a modelagem utilizando o Workbench e informar os comandos de DDL necessário para criação do banco de dados e suas tabelas.



MySQL – Data Definition Language



Exercício – Terceira Parte

- a) Apague individualmente cada tabela;
- b) Apague o banco de dados;