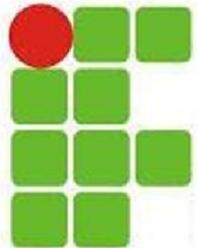
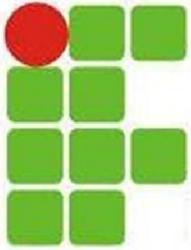

Programação Web

Professor: Diego Oliveira



Conteúdo 13:
Acesso a BD (parte 3)



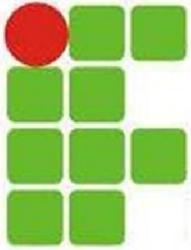


Tipos de Dados

- Nas aulas passadas aprendemos a utilizar o tipo String, salvo como varchar no MySQL
- Hoje vamos aprender a utilizar outros tipos do Java
- A utilização de vários tipos de dados se faz necessária até para sistemas simples, onde são salvos *date* e *time* de ações, valores financeiros, quantidades, responsáveis e etc.



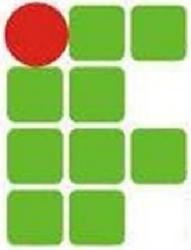
- No próximo slide vemos todos os tipos Java e suas respectivas equivalências em SQL



Tipos de Datos

Tipo de Datos Java	Tipo de Datos SQL
String	VARCHAR, LONGVARCHAR
Java.math.BigDecimal	NUMERIC
boolean	BIT
byte	TINYINT
short	SMALLINT
int	INTEGER
long	BIGINT
float	REAL
double	DOUBLE
byte[]	VARBINARY, LONGVARBINARY
java.sql.Date	DATE
java.sql.Time	TIME
java.sql.Timestamp	TIMESTAMP

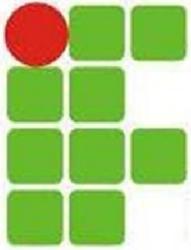




Tipos de Datos

Tipo de Datos SQL	Tipo de Datos Java
CHAR	String
VARCHAR	String
LONGVARCHAR	String
NUMERIC	Java.math.BigDecimal
DECIMAL	Java.math.BigDecimal
BIT	boolean
TINYINT	byte
SMALLINT	short
INTEGER	int

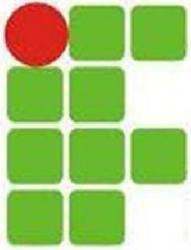




Tipos de Dados

Tipo de Dados SQL	Tipo de Dados Java
BIGINT	long
REAL	float
FLOAT	double
DOUBLE	double
BINARY	byte[]
VARBINARY	byte[]
LONGVARBINARY	byte[]
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp





Tipos de Dados

- No HeidiSQL os tipos são apresentados assim:

Integer

- TINYINT
- SMALLINT
- MEDIUMINT
- INT
- BIGINT
- BIT

Text

- CHAR
- VARCHAR
- TINYTEXT
- TEXT
- MEDIUMTEXT
- LONGTEXT

Binary

- BINARY
- VARBINARY
- TINYBLOB
- BLOB
- MEDIUMBLOB
- LOB

Temporal (time)

- DATE
- TIME
- YEAR
- DATETIME
- TIMESTAMP

Real

- FLOAT
- DOUBLE
- DECIMAL

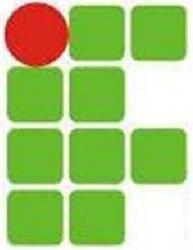
Other

- ENUM
- SET

Spatial (geometry)

- POINT
- LINestring
- POLYGON
- GEOMETRY
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON
- GEOMETRYCOLLECTION

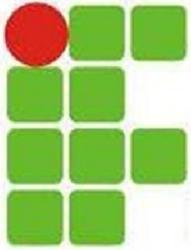




Tipos de Dados

- Ao trabalhar com orientação a objetos e SQL, é importante sempre relacionar tabelas com classes específicas
- Para o caso da biblioteca, temos a tabela livro, então precisamos de uma classe Livro
- Também precisamos de formulários JSP para inserir, atualizar, deletar e consultar esses objetos no banco
- Então fazemos alguns ajustes no projeto:





Tipos de Dados

- Atualize a tabela livro:

```
update livro SET disponivel = 1
```

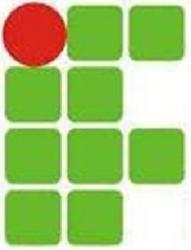
```
update livro SET lancamento = '2015-01-31'
```

```
update livro SET area = 'i'
```

```
update livro SET preco = 50.5
```

#	Nome	Tipo de dados
1	id	INT
2	titulo	VARCHAR
3	autor	VARCHAR
4	edicao	INT
5	editora	VARCHAR
6	ano	INT
7	codigo	VARCHAR
8	exemplares	INT
9	preco	DOUBLE
10	lancamento	DATE
11	disponivel	BIT
12	area	CHAR



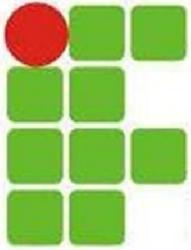


Tipos de Dados

- Crie a classe Livro no Projeto Web:

```
5      public class Livro {  
6          private int id;  
7          private String titulo;  
8          private String autor;  
9          private int edicao;  
10         private String editora;  
11         private int ano;  
12         private String codigo;  
13         private int exemplares;  
14         private double preco;  
15         private Date lancamento;  
16         private boolean disponivel;  
17         private char area;
```





Tipos de Dados

- Crie um formulário JSP correspondente:

Dados do Livro Consultado

ID:

Título:

Autor:

Edição:

Editora:

Ano:

Código:

Exemplares:

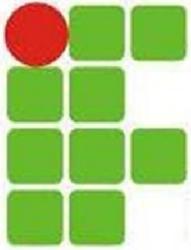
Preço:

Lançamento:

Disponível:

Área:



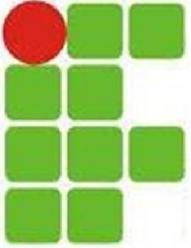


Tipos de Dados

- Código do JSP:

```
15      <title>JSP + MySQL2</title>
16      <%
17          String id = request.getParameter("id");
18      %>
19  </head>
20  <body>
21      <center>
22          <%
23              if(id == null){
24          %>
25          <h1>Digite os dados da pesquisa</h1>
26          <form action="consultaLivro.jsp" method="post">
27              ID: <input type="text" name="id">
28              <input type="submit" value="Pesquisar">
29          </form>
30          <%
31              }else{
```

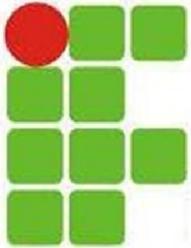




Tipos de Dados

- Código do JSP:

```
32 Banco b = new Banco();
33 ResultSet livros = b.selectBooksById(Integer.parseInt(id));
34
35 Livro l = new Livro();
36 while(livros.next()){
37     l.setId(livros.getInt("id"));
38     l.setTitulo(livros.getString("titulo"));
39     l.setAutor(livros.getString("autor"));
40     l.setEdicao(livros.getInt("edicao"));
41     l.setEditora(livros.getString("editora"));
42     l.setAno(livros.getInt("ano"));
43     l.setCodigo(livros.getString("codigo"));
44     l.setExemplares(livros.getInt("exemplares"));
45     l.setPreco(livros.getDouble("preco"));
46     l.setLancamento(livros.getDate("lancamento"));
47     l.setDisponivel(livros.getBoolean("disponivel"));
48     l.setArea(livros.getString("area").charAt(0));
49 }
50 %>
```

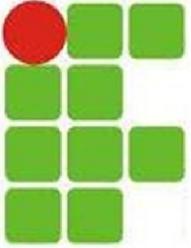


Tipos de Dados

- Código do JSP:

```
<h1>Dados do Livro Consultado</h1>
<form action="consultaLivro.jsp" method="post">
  ID: <input type="text" name="id" value="<%=1.getId()%>"> <br>
  Título: <input type="text" size="50" name="titulo" value="<%=1.getTitulo()%>"> <br>
  Autor: <input type="text" name="autor" value="<%=1.getAutor()%>"> <br>
  Edição: <input type="text" name="edicao" value="<%=1.getEdicao()%>"> <br>
  Editora: <input type="text" name="editora" value="<%=1.getEditora()%>"> <br>
  Ano: <input type="text" name="ano" value="<%=1.getAno()%>"> <br>
  Codigo: <input type="text" name="codigo" value="<%=1.getCodigo()%>"> <br>
  Exemplares: <input type="text" name="exemplares" value="<%=1.getExemplares()%>"> <br>
  Preço: <input type="text" name="preco" value="<%=1.getPreco()%>"> <br>
  Lançamento: <input type="text" name="lançamento" value="<%=1.getLancamento()%>"> <br>
  Disponível: <input type="text" name="disponivel" value="<%=1.isDisponivel()%>"> <br>
  Área: <input type="text" name="area" value="<%=1.getArea()%>"> <br>
  <input type="submit" value="Pesquisar">
</form>
<%=1.isDisponivel()%>
```





Tipos de Dados

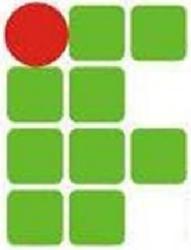
- Código para selecionar um livro pelo ID em Banco.java:

```
public ResultSet selectBooksById(int id) throws SQLException{  
    ResultSet rs = stmt.executeQuery("select * from livro where id=" + id);  
    return rs;  
}
```

- Código para atualizar uma coluna de uma linha específica (**para o exercício**):

```
public void atualizarPreco(double preco, int id) throws SQLException {  
    stmt.executeUpdate("UPDATE livro SET preco = " + preco + " where id = " + id);  
}
```

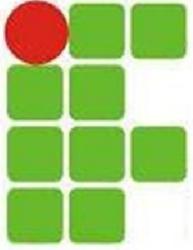




Exercício

- Crie um formulário que permita a consulta de um livro pelo ID
- Ao consultar um livro pelo ID, ele deverá ser resgatado e um novo objeto livro preenchido com todos os seus valores
- Depois todos os dados do livro serão preenchidos no formulário
- **Permita a atualização do banco de dados através deste formulário!**





Perguntas?

