



BANCO DE DADOS RELACIONAL X NÃO-RELACIONAL

Taxonomia de NoSQL

- Key-value



- Graph database



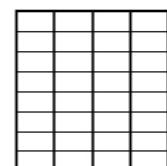
- Document-oriented



- Column family



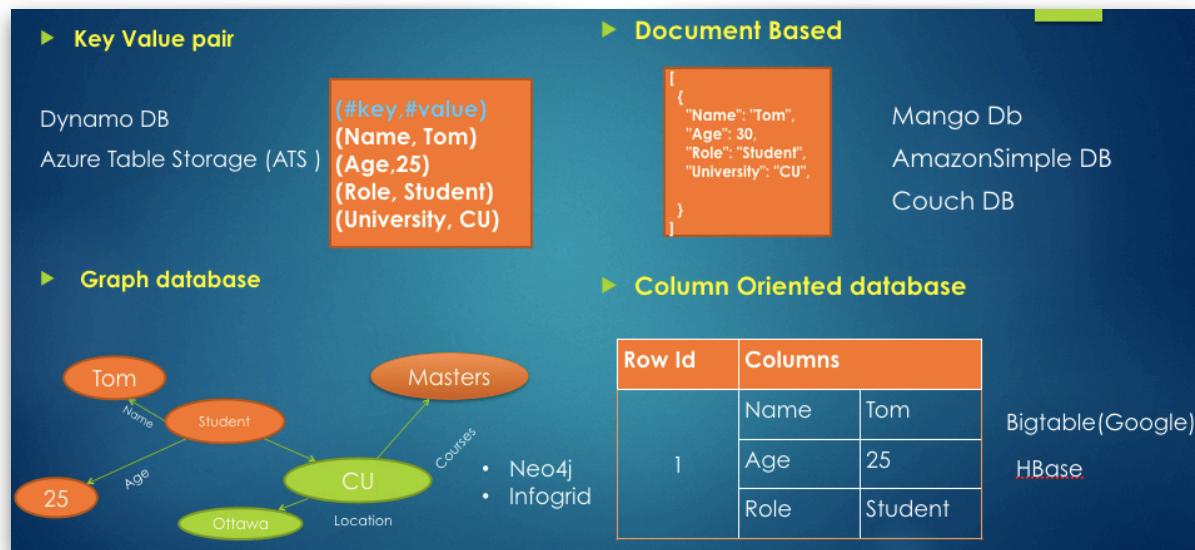
SQL



O CONCEITO DE MODELO
RELACIONAL (SQL) SE BASEIA
NO FATO DE QUE TODOS OS
DADOS SEJAM GUARDADOS EM
TABELAS.

BANCO DE DADOS RELACIONAL X NÃO-RELACIONAL

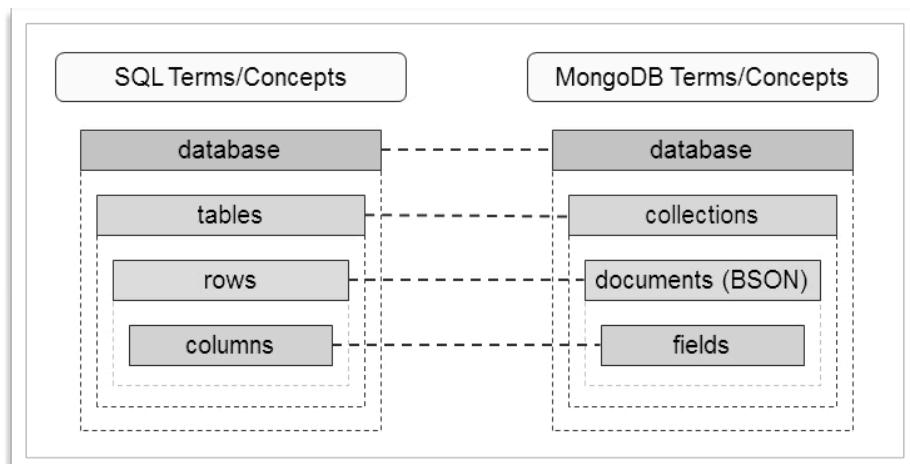
Taxonomia de NoSQL



BANCO DE DADOS RELACIONAL X NÃO-RELACIONAL

Quais as diferenças?

- ▶ **Bancos de Dados NoSQL**
 - ✓ Toda a informação é agrupada e guardada no mesmo registro.
- ▶ **Banco de Dados Relacional**
 - ✓ Precisa ter o relacionamento entre várias **tabelas** para ter a informação, informação esta disposta no modelo **entidade** e **relacionamento**.



BANCO DE DADOS RELACIONAL X NÃO-RELACIONAL

Quais as diferenças ?

► Bancos de Dados NoSQL

- ✓ Toda a informação é agrupada e guardada no mesmo registro.

► Banco de Dados Relacional

- ✓ Precisa ter o relacionamento entre várias **tabelas** para ter a informação, informação esta disposta no modelo **entidade** e **relacionamento**.

Banco de Dados Relacional

```
CREATE TABLE usuarios (
    usuario_id INT NOT NULL,
    primeiro_nome VARCHAR2(50),
    sobre_nome VARCHAR2(50)
);
```

```
INSERT INTO usuario(id_usuario, primeiro_nome, sobre_nome )
VALUES( 1, 'Fernando', 'Monteiro' );
```

Banco de Dados Não-Relacional

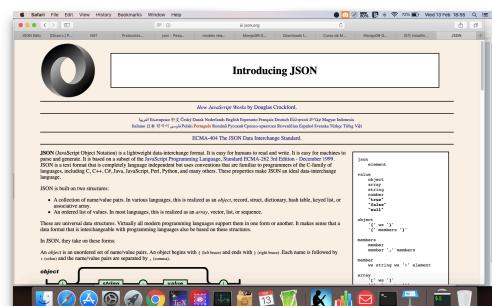
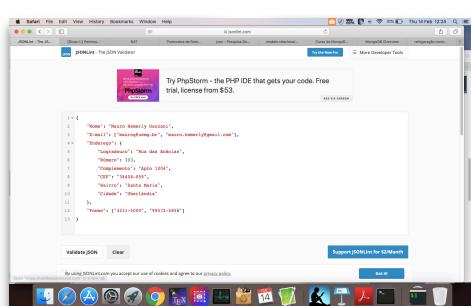
MongoDB

```
db.usuario.insert( { 'usuario_id': 1, 'primeiro_nome': 'Fernando',
'sobre_nome': 'Monteiro' } );
```

JSON (JAVASCRIPT OBJECT NOTATION)

Baseado em um subconjunto da linguagem de programação **JavaScript**, Standard ECMA-262 3a Edição - Dezembro de 1999.

- ◎ Formato leve para troca de informações
- ◎ Para os seres-humanos
 - ✓ fácil leitura e compreensão
- ◎ Para as máquinas
 - ✓ fácil leitura e interpretação
- ◎ Formato independente de linguagem de programação
- ◎ Comparado a XML
 - ◎ Menor
 - ◎ Mais fácil e rápido de escrever.



<https://www.json.org>

<https://jsonlint.com>

JSON (JAVASCRIPT OBJECT NOTATION)

Duas Estruturas Básicas:

- ◎ Uma coleção de pares **nome/valor**
 - ✓ Em outras linguagens: *struct*, *record*, objeto, tabela hash, etc.
 - ✓ Ex: {"nome" : "marcio"}
- ◎ Uma lista ordenada de valores
 - ✓ Em outras linguagens: arrays, listas, sequência, vetor, etc.
 - ✓ Ex.: ["Corolla", "Fusion", "Polo", "Creta"]

JSON (JAVASCRIPT OBJECT NOTATION)

Estruturas e Tipos de Dados:

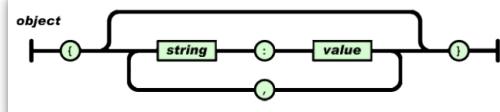
- ◎ Objeto
- ◎ Array
- ◎ Valor
- ◎ Número
- ◎ String

JSON (JAVASCRIPT OBJECT NOTATION)

Objeto

- Conjunto de pares nome/valor

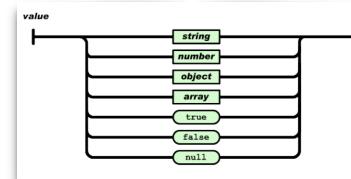
✓ Começa com { e termina com }



✓ Cada nome é seguido por :(dois pontos)

✓ Os pares são separados por , (vírgula)

```
1 {  
2   "Título": "MongoDB in Action",  
3   "Autor": "Kyle Banker",  
4   "Edição": 2,  
5   "Ano": 2016  
6 }
```



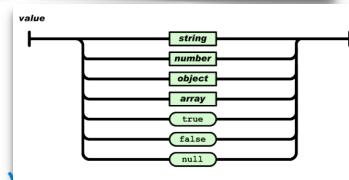
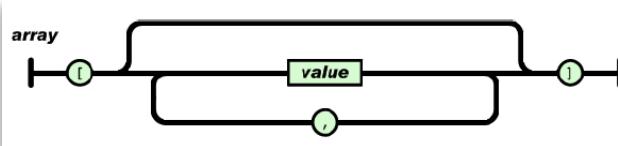
```
1 {  
2   "Segunda-feira": "Banco de Dados II",  
3   "Terça-feira": "Programação Orientada a Objetos",  
4   "Quarta-feira": "Linguagens de Programação",  
5   "Quinta-feira": "Lógica para Computação"  
6 }
```

JSON (JAVASCRIPT OBJECT NOTATION)

Array

- Coleção ordenada de valores

✓ Começa com [e termina com]



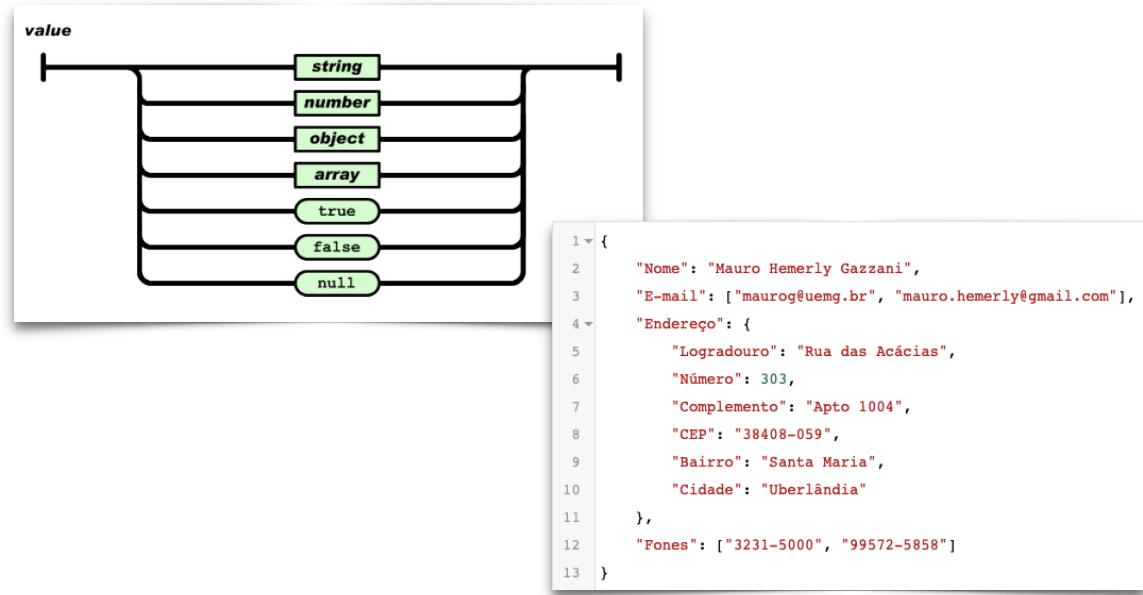
✓ Os valores são separados por , (vírgula)

```
1 {  
2   "Dias da Semana": ["Domingo", "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"],  
3  
4   "funcionários": [  
5     {"  
6       "nome": "John",  
7       "sobrenome": "Doe"  
8     },  
9     {  
10       "nome": "Anna",  
11       "sobrenome": "Smith"  
12     }  
13   ]  
14 }
```

JSON (JAVASCRIPT OBJECT NOTATION)

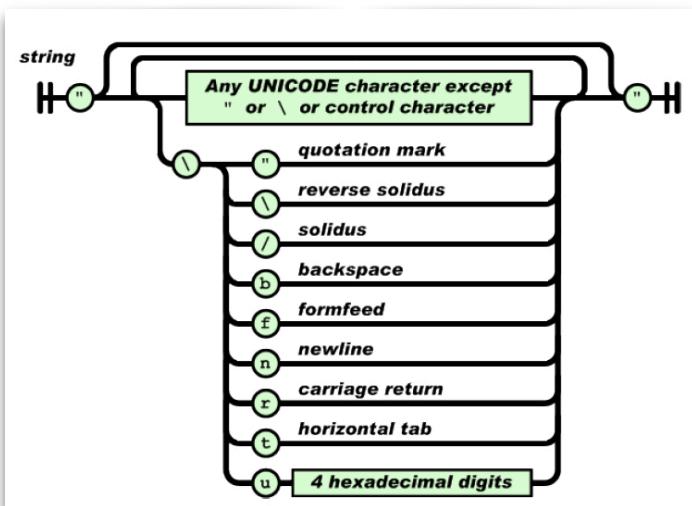
Valor

- Pode ser uma *string*, número (inteiro ou fracionário), objeto, array, ou as palavras *null*, *true* e *false*.



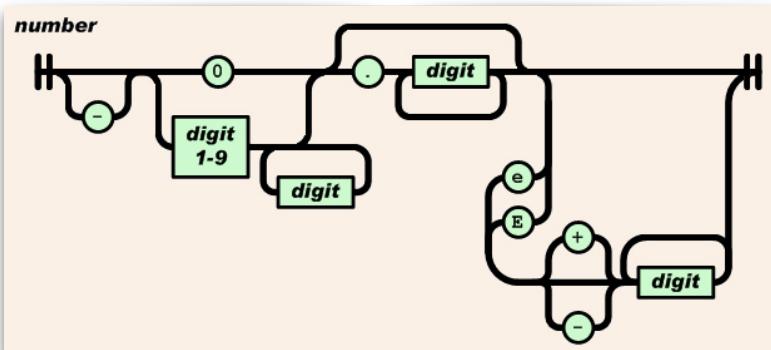
JSON (JAVASCRIPT OBJECT NOTATION)

String



JSON (JAVASCRIPT OBJECT NOTATION)

Números



JSON (JAVASCRIPT OBJECT NOTATION)

Notação em Resumo (BNF)

```
object
  {}
  {members}
members
  pair
  pair . members
pair
  string : value
array
  []
  [elements]
elements
  value
  value , elements
value
  string
  number
  object
  array
  true
  false
  null
```

XML X JSON

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{
  "empinfo" : [
    {
      "employees" : [
        {
          "name" : "James Kirk",
          "age" : 40,
        },
        {
          "name" : "Jean-Luc Picard",
          "age" : 45,
        },
        {
          "name" : "Wesley Crusher",
          "age" : 27,
        }
      ]
    }
  ]
}
```

```
http://localhost:8080/Json/SyncReply/Contacts
{
  - Contacts: [
    - {
      FirstName: "Demis",
      LastName: "Bellot",
      Email: "demis.bellot@gmail.com"
    },
    - {
      FirstName: "Steve",
      LastName: "Jobs",
      Email: "steve@apple.com"
    },
    - {
      FirstName: "Steve",
      LastName: "Ballmer",
      Email: "steve@microsoft.com"
    },
    - {
      FirstName: "Eric",
      LastName: "Schmidt",
      Email: "eric@google.com"
    },
    - {
      FirstName: "Larry",
      LastName: "Ellison",
      Email: "larry@oracle.com"
    }
  ]
}

http://localhost:8080/XML/SyncReply/Contacts
<ContactResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Contacts>
    <Contact>
      <Email>demis.bellot@gmail.com</Email>
      <FirstName>Demis</FirstName>
      <LastName>Bellot</LastName>
    </Contact>
    <Contact>
      <Email>steve@apple.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Jobs</LastName>
    </Contact>
    <Contact>
      <Email>steve@microsoft.com</Email>
      <FirstName>Steve</FirstName>
      <LastName>Ballmer</LastName>
    </Contact>
    <Contact>
      <Email>eric@google.com</Email>
      <FirstName>Eric</FirstName>
      <LastName>Schmidt</LastName>
    </Contact>
    <Contact>
      <Email>larry@oracle.com</Email>
      <FirstName>Larry</FirstName>
      <LastName>Ellison</LastName>
    </Contact>
  </Contacts>
</ContactResponse>
```

JSON: EXERCÍCIO

Banco de Dados II – Aula 2: Tabelas

Composição de uma Base Relacional

Tabela

Veiculo

Atributos				
Placa	Fabricante	Marca	Ano	Cor
IOS-0078	Renault	Sandero	2009	Vermelho
ITO-1314	Volkswagen	Fox	2010	Azul
IIM-1453	Hyundai	I30	2014	Pérola
IVA-2018	Chevrolet	Onix	2015	Branco
MAI-1852	Citroen	C3	2013	Preto

Tuplas

Domínio

codigo	codped	qtd	subtotal
1	1005	10	1200
2	1008	12	720
3	1008	2	240
4	1010	1	500
5	1010	2	335
6	1010	1	185
7	1015	1	125
8	1015	1	630
9	1023	1	400
10	1023	1	300
11	1023	1	200

Tabela Itens

codigo	nome	telefone	limite_compras
202	Ernesto	3222-0809	1500
221	Amélia	3233-2474	2000
233	Luis Alberto	3243-9021	1500
282	José Antônio	3243-0071	800
295	Carlos Silva	3224-5678	800

Tabela Cliente

codigo	subtotal	nome	valor
4	500	Luis Alberto	1020
5	335	Luis Alberto	1020
6	185	Luis Alberto	1020

Tabela Resultado

Banco de Dados II – Aula 2: Tabelas

Composição de uma Base Relacional

Banco de Dados Relacional

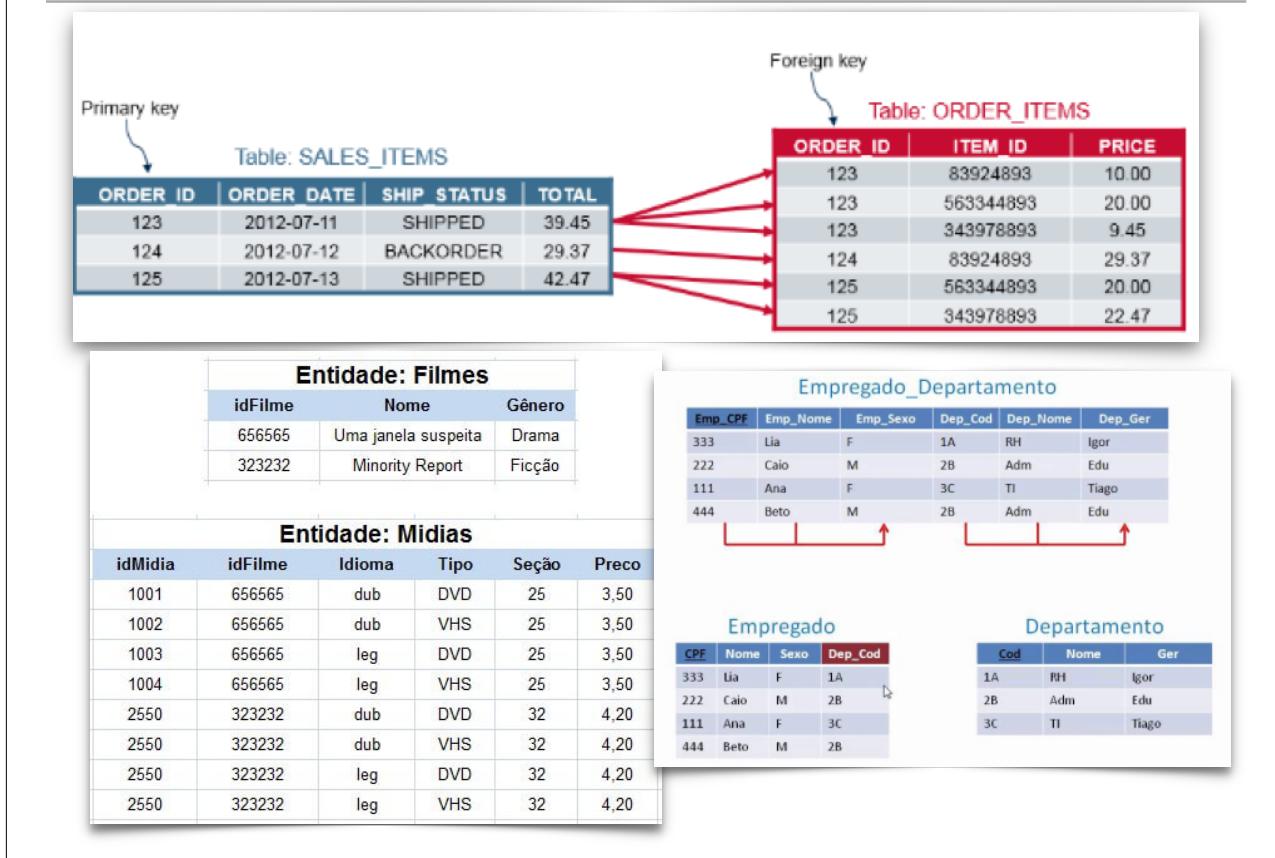
Veiculo	Placa	Fabricante	Marca	Ano	Cor
	IOS-0078	Renault	Sandero	2009	Vermelho
	ITO-1314	Volkswagen	Fox	2010	Azul
	IIM-1453	Hyundai	I30	2014	Pérola
	IVA-2018	Chevrolet	Onix	2015	Branco
	MAI-1852	Citroen	C3	2013	Preto

codCliente	nome	idade	telefone	carroPlaca	Cliente
1	Paulo Freitas	23	5184259863	IOS-0078	
2	Pâmela Silva	35	5196698752	ITO-1314; IVA-2018	
4	Rogério Lins	30	5598633248	IIM-1453; MAI-1852	

Codigos	Nome	DataNascimento
1	Marcio	1/6/1975
2	Marlos	5/8/1980
3	Luciane	10/5/1970
4	Wilkie	12/3/1974

Codigo	Fone	Tipo
1	22548954	Residencial
1	88512547	Celular
3	89665485	Celular
4	26539955	Residencial

JSON: EXERCÍCIO



JSON: EXERCÍCIO

Banco de Dados

Exemplo de um Banco de Dados: "Projeto ABC"

Tabela EMPREGADO					
Nome	RG	CIC	Depto	RG Supervisor	Salário
João Luiz	10101010	11111111	1	NULO	3.000,00
Fernando	20202020	22222222	2	10101010	2.500,00
Ricardo	30303030	33333333	2	10101010	2.500,00
Jorge	40404040	44444444	2	20202020	4.200,00
Renato	50505050	55555555	3	20202020	1.300,00

Tabela DEPENDENTES				
RG Responsável	Nome Dependente	Dt. Nascimento	Relação	Sexo
10101010	Jorge	27/12/86	Filho	Masculino
10101010	Luiz	18/11/79	Filho	Masculino
20202020	Fernanda	14/02/69	Conjuge	Feminino
20202020	Angelo	10/02/95	Filho	Masculino
30303030	Adreia	01/05/90	Filha	Feminina

Tabela DEPARTAMENTO		
Nome	Número	RG Gerente
Contabilidade	1	10101010
Engenharia Civil	2	30303030
Engenharia Mecânica	3	20202020

Tabela PROJETO		
Nome	Número	Localização
Financeiro	5	São Paulo
Motor 3	10	Rio Claro
Pátio Central	20	Campinas

Tabela DEPARTAMENTO_PROJETO		
Número Depto.	Número Projeto	
2	5	
3	10	
2	20	

Tabela EMPREGADO_PROJETO		
RG Empregado	Número Projeto	Horas
20202020	5	10
20202020	10	25
30303030	5	35
40404040	20	50
50505050	20	35

JSON: EXERCÍCIO

Tabela ALUNO:

id_aluno	nome	email	dtcadastro
1	Douglas	douglas@targettrust.com.br	NULL
2	José Carlos	josecarlos@targettrust.com.br	2013-01-21
3	Douglas	dg@targettrust.com.br	2013-01-21
4	Marta	marta@targettrust.com.br	2013-01-21
5	Simone	simone@targettrust.com.br	2013-01-21
6	Rafael	rafael@targettrust.com.br	2013-01-21
7	ROCK	rock@targettrust.com.br	2013-01-21

Tabela CURSO:

id_curso	nome_curso	preco	duracao
1	Java	10.00	20
2	Oracle	30.00	20
3	.NET	60.00	20
4	PHP	40.00	20
5	Linux	40.00	20

Tabela TELEFONE:

id_telefone	id_aluno	tipo	numero
1	1	casa	(51) 3333-4444
2	1	celular	(51) 9999-999

Tabela INSCRICAO:

id_inscricao	avaliacao	dtinicio	id_curso	id_aluno
1	NULL	2011-06-01	1	1
2	NULL	2013-01-21	1	2
3	NULL	2013-01-21	1	3
4	NULL	2013-01-21	1	4
5	NULL	2013-01-21	1	5
6	NULL	2013-01-21	1	6
7	NULL	2013-01-21	2	1
8	NULL	2013-01-21	2	2
9	NULL	2013-01-21	2	3
10	NULL	2013-01-21	3	4
11	NULL	2013-01-21	3	5