

Banco de Dados

Centro Universitário Senac

Prof. Msc. Fabio Versolatto
fabio.rversolatto@sp.senac.br

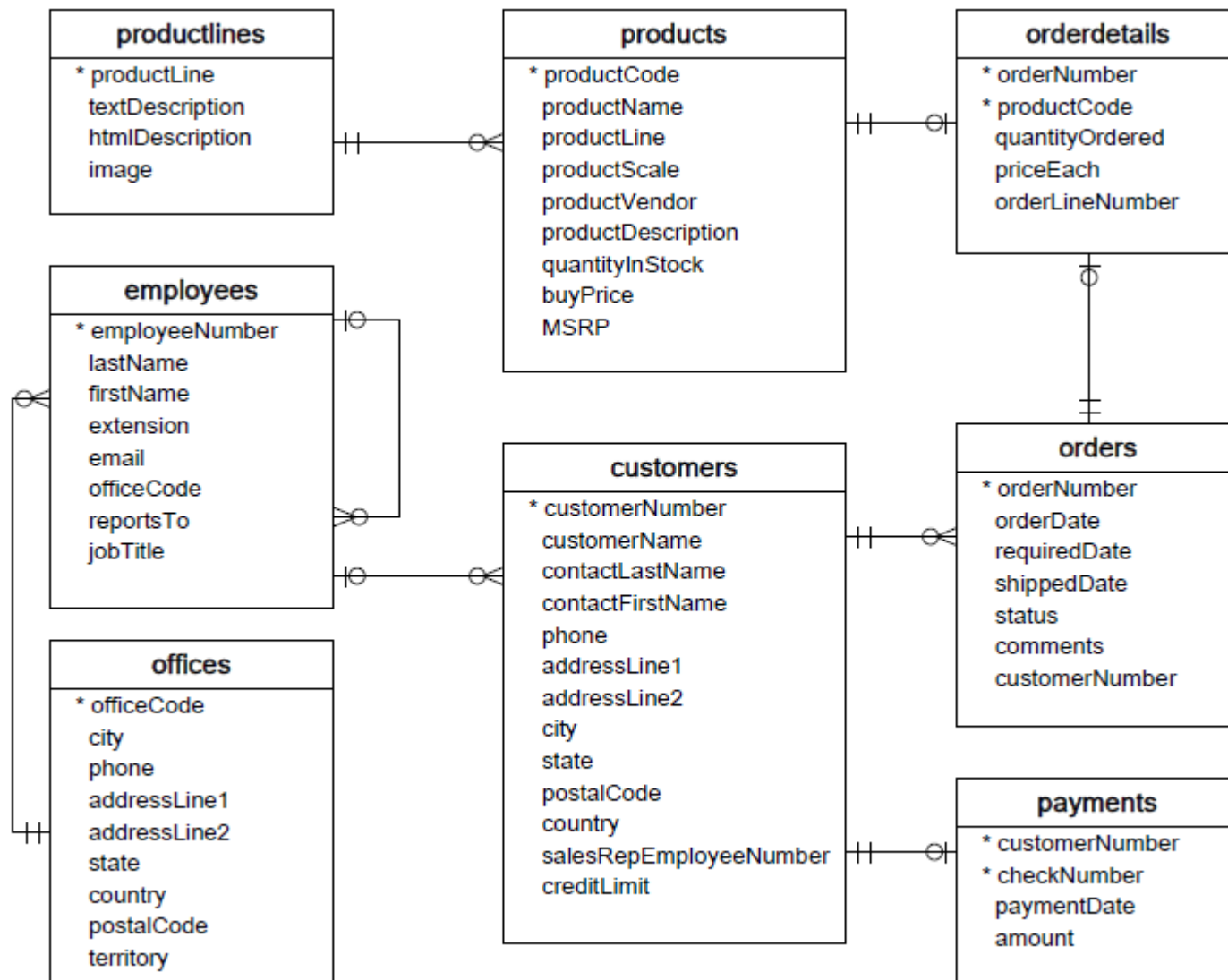
Onde estávamos

- Conceitos Fundamentais de Banco de Dados
- SGBD
- Projeto Lógico – MER
- Introdução ao SQL (implementação do Modelo Físico)
 - ESTRUTURA: CREATE, ALTER
 - CRUD (INSERT, SELECT, UPDATE, DELETE)
 - CLAUSULA WHERE ☺
- Normalização
- Álgebra Relacional

Pra onde vamos...

- Administração de BD

Álgebra Relacional



- 1) Listar o nome, sobrenome e email de todos os funcionários
- 2) Listar o nome, sobrenome e email de todos os funcionários que são representantes de vendas
- 3) Listar todos os cargos (sem repetir)
- 4) Listar todas as cidades (sem repetir) em que a empresa possui escritórios
- 5) Listar o nome, sobrenome e email , cidade, país e telefone do escritório do presidente da empresa
- 6) Listar o produto com o maior preço
- 7) Listar o produto com menor preço
- 8) Listar o produto com menor quantidade em estoque
- 9) Listar o produto com maior quantidade em estoque
- 10) Listar produto e quantidade de vendas daquele produto

```

/*1*/
select e.firstName as nome, e.lastName as sobrenome, e.email as email
from classicmodels.employees e
order by e.firstName;

/*2*/
select e.firstName as nome, e.lastName as sobrenome, e.email as email
from classicmodels.employees e
where e.jobTitle like '%Sales Rep%'
order by e.firstName;

/*3*/
select DISTINCT(e.jobTitle)
from classicmodels.employees e;

/*4*/
select distinct(o.city)
from classicmodels.offices o;

```

```

/*5*/
select e.firstName as nome, e.lastName as sobrenome, e.email as email,
       o.city as cidade, o.country as pais, o.phone as telefone
from classicmodels.employees e
inner join classicmodels.offices o on e.officeCode = o.officeCode
where e.jobTitle = 'President';

/*6*/
select max(p.buyPrice), p.productName
from classicmodels.products p;

/*7*/
select min(p.buyPrice), p.productName
from classicmodels.products p;

```

```

/*8*/
select p.productName
from classicmodels.products p
where p.quantityInStock = (select min(quantityInStock) from classicmodels.products);

/*9*/
select p.productName
from classicmodels.products p
where p.quantityInStock = (select max(quantityInStock) from classicmodels.products);

/*10*/
select p.productName, sum(od.quantityOrdered)
from classicmodels.products p
inner join classicmodels.orderdetails od on od.productCode = p.productCode
group by p.productName;

```

- 1) Listar os clientes com maior quantidade de vendas. Exibir: Nome, Sobrenome, Cidade e Estado do Cliente e a quantidade de vendas dele
- 2) Listar os produtos com maior quantidade de vendas. Exibir: Nome do Produto, Descrição da Linha do Produto, Preço do Produto e Quantidade de Vendas
- 3) Listar os empregados que venderam mais (em quantidade). Exibir: Nome do funcionário, sobrenome, email e o superior dele.
- 4) Listar os clientes que mais gastaram (\$). Exibir: Nome, Sobrenome, Cidade e Estado do Cliente e o limite de créditos deles
- 5) Montar um relatório em que conste todos os dados dos empregados (inclusive do escritório em que eles pertencem) e os clientes que esses atendem (inclua neste relatório também todos os dados dos clientes).

```
/*1*/  
select c.contactFirstName, c.contactLastName, c.state, count(o.orderNumber)  
from classicmodels.customers c  
inner join classicmodels.orders o on c.customerNumber = o.customerNumber  
group by c.contactFirstName, c.contactLastName, c.state;  
/*2*/  
select p.productName, p.productDescription, p.buyPrice, count(o.orderNumber)  
from classicmodels.products p  
inner join classicmodels.orderdetails od on p.productCode = od.productCode  
inner join classicmodels.orders o on o.orderNumber = od.orderNumber  
group by p.productName, p.productDescription, p.buyPrice;  
/*3*/  
select e.firstName, e.lastName, e.email, sup.firstName as superior, count(o.orderNumber)  
from classicmodels.employees e  
inner join classicmodels.customers c on c.salesRepEmployeeNumber = e.employeeNumber  
inner join classicmodels.orders o on c.customerNumber = o.customerNumber  
inner join classicmodels.employees sup on sup.employeeNumber = e.reportsTo  
group by e.firstName, e.lastName, e.email, sup.firstName;
```



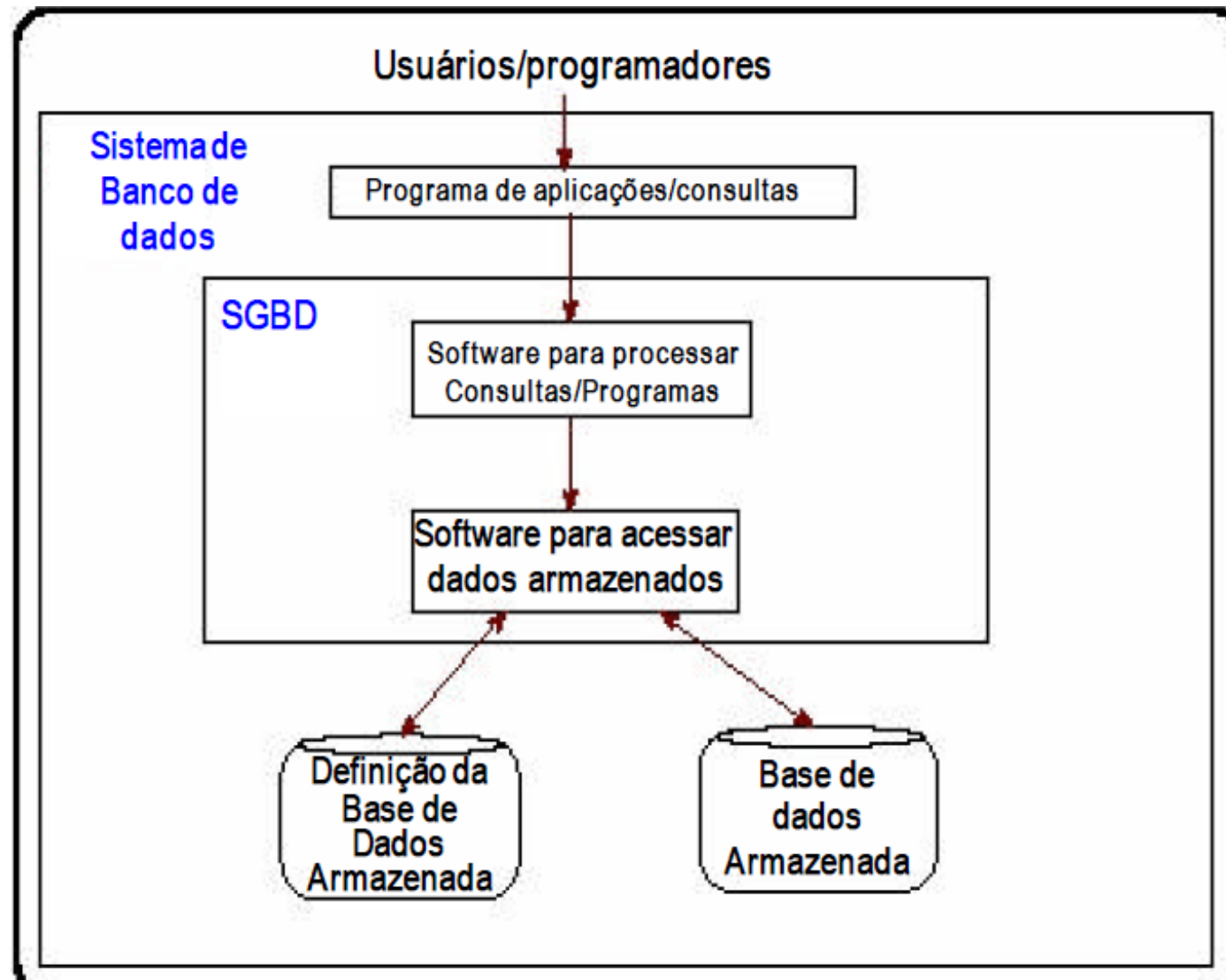
```

/*4*/
select c.contactFirstName, c.contactLastName, c.city, c.state, c.creditLimit, sum(od.quantityOrdered * od.priceEach) as total
from classicmodels.customers c
inner join classicmodels.orders o on c.customerNumber = o.customerNumber
inner join classicmodels.orderdetails od on od.orderNumber = o.orderNumber
group by c.contactFirstName, c.contactLastName, c.state, c.creditLimit
order by total desc;
/*5*/
select e.firstName, e.lastName, e.email, e.jobTitle, e.extension,
       sup.firstName as superior,
       o.city, o.addressLine1, o.addressLine2, o.phone, o.state, o.country, o.postalCode, o.territory,
       c.customerName, c.contactFirstName, c.contactLastName, c.phone, c.addressLine1, c.addressLine2, c.city, c.state, c.postalCode, c.country, c.creditLimit
from classicmodels.employees e
inner join classicmodels.offices o on e.officeCode = o.officeCode
left outer join classicmodels.employees sup on e.reportsTo = sup.employeeNumber
left outer join classicmodels.customers c on c.salesRepEmployeeNumber = e.employeeNumber
order by 1 asc;

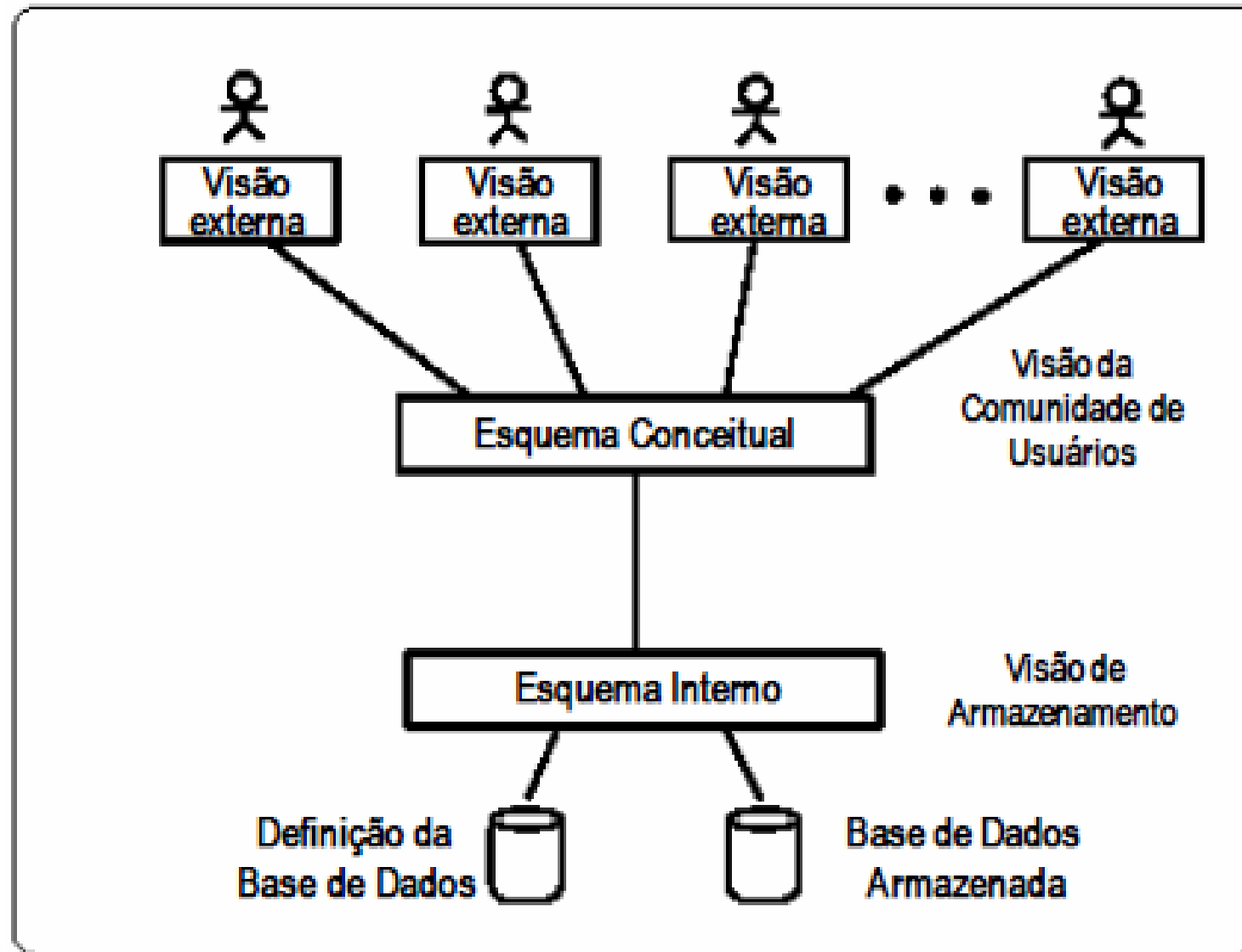
```

left outer... Pega todos os que tem
intersecção E os da esquerda

Administração de BD



Administração de BD



Administração de BD

- Desenvolvedores e Aplicações atuam no nível mais externo
- Administrador do BD atua nas camadas mais internas:
 - segurança e autorização;
 - garantia da disponibilidade e recuperação de dados (backup);
 - otimização do banco de dados para assegurar o desempenho (tuning)

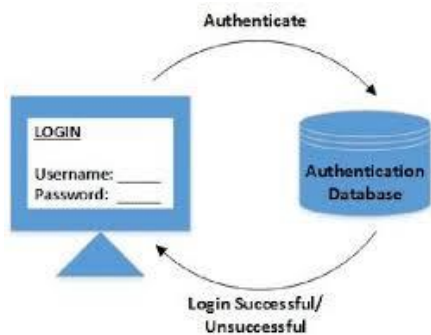
Administrador de BD

- Segurança

CONFIDENCIALIDADE	Garante que só pessoas autorizadas terão acesso aos dados.
AUTENTICAÇÃO	Garante a identidade de quem está utilizando o sistema.
INTEGRIDADE	Garante que os dados não sejam alterados de maneira indevida, seja de forma accidental, seja de forma maliciosa.
DISPONIBILIDADE	Garante que agentes maliciosos não consigam derrubar o sistema, o que impediria o acesso de seus usuários legítimos.
CONTROLE DE ACESSO	Também chamado de autorização, garante que somente os usuários que tiverem os direitos de acesso apropriados possam realizar determinadas operações no sistema.

Administração de BD

- Criação de Logins



```
1 select * from mysql.user;
```

Form Editor | Navigate: << 1 / 5 >> | Edit: [Icons]

Host:	localhost
User:	mysql.infoschema
Select_priv:	Y
Insert_priv:	N
Update_priv:	N
Delete_priv:	N
Create_priv:	N
Drop_priv:	N
Reload_priv:	N
Shutdown_priv:	N
Process_priv:	N

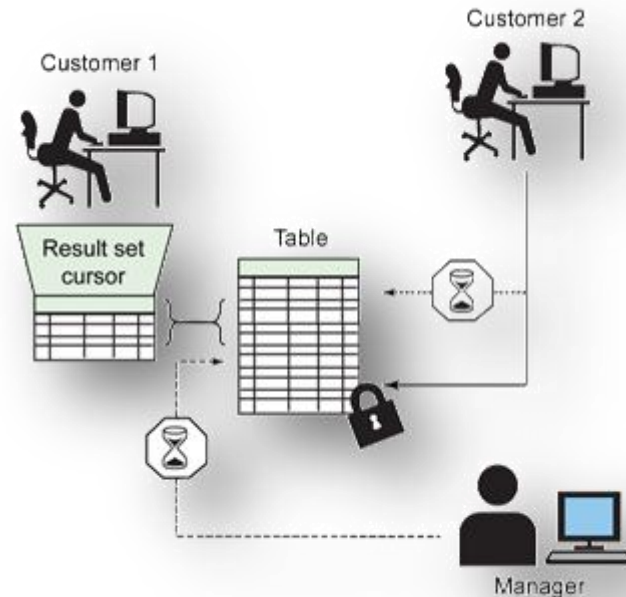
user1 x Apply

Administração de BD

- Criação de Logins

```
CREATE USER 'myapp'@'%' IDENTIFIED BY 'senha'  
WITH MAX_QUERIES_PER_HOUR 200  
    MAX_UPDATES_PER_HOUR 100  
    MAX_CONNECTIONS_PER_HOUR 50  
    MAX_USER_CONNECTIONS 30;
```

- Concorrência



Administração de BD

- Controle de concorrência é quando, em um banco de dados, usuários distintos tentam acessar a mesma informação e então é feito um controle entre essas transações
- ACID
- Atomicidade, Consistência, Isolamento e Durabilidade

Administração de BD

- Bloqueio Compartilhado: quando uma transação recebe este tipo de bloqueio e a instrução é de leitura, então, mais de uma transação poderá acessar o mesmo dado. Se a instrução for de gravação, então ela não poderá participar de um bloqueio compartilhado, ou seja, é permitido que várias transações acessem um mesmo item "A" se todas elas acessarem este item "A" apenas para fins de leitura
- Bloqueio Exclusivo: quando uma transação recebe este tipo de bloqueio, ela fica exclusivamente reservada para a instrução que compõe a transação, não permitindo que outra transação faça uso do dado que está sendo utilizado, logo, um item bloqueado para gravação é chamado de bloqueado exclusivo, pois uma única transação mantém o bloqueio no item

Administração de BD

- Solução mais comum: gerenciamento de transação do SGBD
- Abre uma transação -> Lock da Tabela -> Commit
- Mas não podemos nos limitar apenas isso...

Administração de BD

- Administração de Banco de Dados compreende: Instalar e Manter
- Figura central: DBA (DataBase Administrator) – DBA
- DBA: possui em geral todos os poderes para alteração dos objetos de banco de dados (criação, exclusão, alteração)
- Desenvolvedor de BD: dependente das ações do DBA, somente “usa” os objetos

Administração de BD

- Instalar, uma visão rasa:
 - Configurar a quantidade máxima de usuários que podem se conectar no BD
 - Configurar o tempo (timeout) de uma transação
 - Configurar os usuários/senhas e tipos de acesso
 - Implantar políticas de segurança
 - Implantar mecanismos de contingência, recuperação de falhas e plano de catástrofes.

Administração de BD

- Instalar, uma visão rasa:
 - Configurar o servidor de BD
 - Espaço em disco
 - Memória alocada

Administração de BD

- Manter, uma visão rasa:
 - **Backup**, ou cópia de segurança. *** IMPORTANTE ***
 - **Restore**
 - “O banco caiu”
 - “A tabela está locada”
 - “A performance está ruim”... Será que tudo é culpa do DBA??????

Administração de BD

- Backup
 - **Não é apenas uma cópia de segurança**
 - Necessita de definição de políticas (periodicidade, tempo de armazenamento, tempo de *restore*, etc...)

Tipos de Banco de Dados

- O que vimos aqui até aqui: modelos relacionais
- Existem outras opções
- Quando usar: NECESSIDADE
- Não existe melhor ou pior... Existe aquele que melhor se encaixa a sua necessidade e ao contexto do seu negócio
- Modelo relacional: entidades claras, modelo claro...

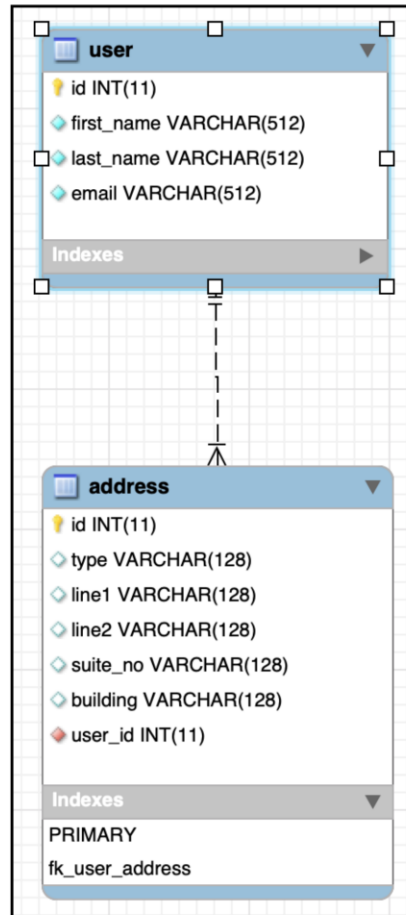
Tipos de Banco de Dados

- NoSQL (não relacional)
- Dados semiestruturados
- Ou seja, enquanto no relacional eu tenho dados estruturados e um schema bem definido, no NoSQL eu tenho modelos que não possuem responsabilidade a estrutura e possui até um certo grau de desestruturação das informações
- Exemplo: Informações vistas como documentos

Tipos de Banco de Dados

- Documento
 - Hierarquia de diretório
 - Coleções
 - Etiquetas
 - Metadados
- JSON

SQL



NoSQL

User & Address

```
{
  "_id": ObjectID,
  "first_name": String,
  "last_name": String,
  "email": String,
  "address": [
    {
      "type": String,
      "line1": String,
      "line2": String,
      "suite_no": String,
      "building": String
    }
  ]
}
```

Tipos de Banco de Dados

- Linguagem para acesso muda também...
- Não é mais T-SQL


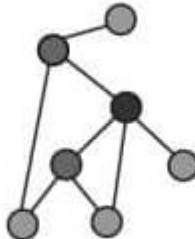
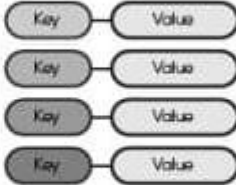
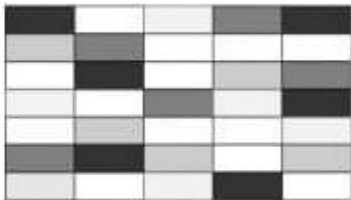
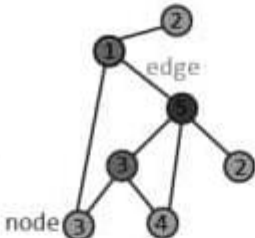





Find all contacts with at least one work phone

SQL CLI `select * from contact A, phones B where
 A.did = B.did and B.type = 'work';`

MongoDB CLI `db.contact.find({ "phones.type": "work" });`

Referência: <https://www.mongodb.com/blog/post/mongodb-vs-sql-day-14-queries>

Tipos de Banco de Dados

Document	Graph	Key-Value	Wide-Column													
																
<pre>{ "user": { "id": "143", "name": "improgrammer", "city": "New York" } }</pre>			<table><tr><td>1</td><td>Fruit</td><td>A Foo</td><td>B Baz</td></tr><tr><td>2</td><td>City</td><td>E DC</td><td>D PLA</td><td>G FLD</td></tr><tr><td>3</td><td>State</td><td>A NZ</td><td>C CL</td></tr></table>	1	Fruit	A Foo	B Baz	2	City	E DC	D PLA	G FLD	3	State	A NZ	C CL
1	Fruit	A Foo	B Baz													
2	City	E DC	D PLA	G FLD												
3	State	A NZ	C CL													
																

Referência: <https://www.improgrammer.net/most-popular-nosql-database/>

Tipos de Banco de Dados

- Existem outros tipos que não o SQL e o NoSQL
- Exemplo: Dicionário (Chave-Valor)

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623



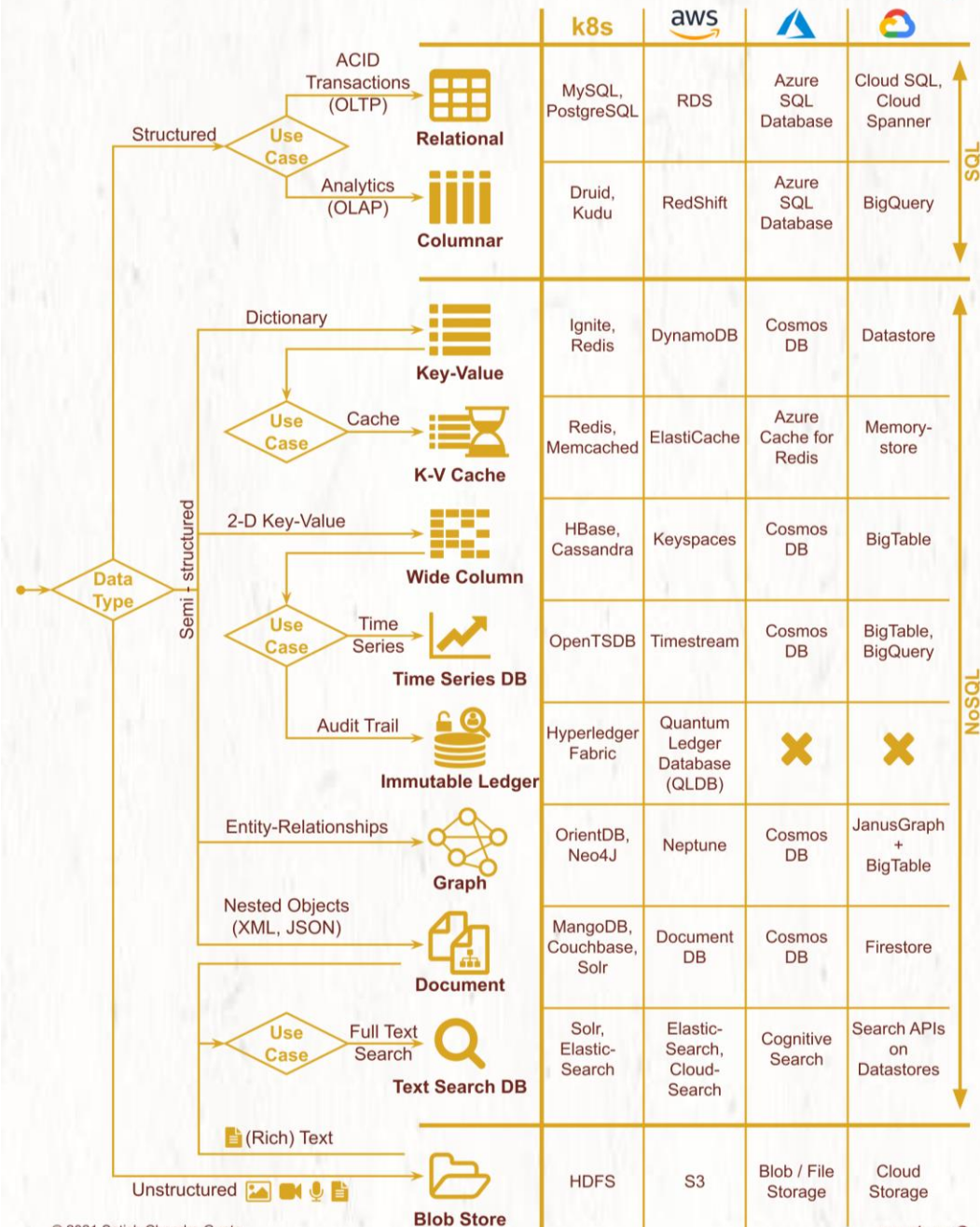
redis

Tipos de Banco de Dados

- Novamente
- Não existe melhor ou pior
- Existe a necessidade !!!
- Não adianta adaptar a tecnologia à necessidade
 - “Eu gosto de MongoDB por que é novo, então tudo vai ser NoSQL”
 - “Eu não conheço NoSQL, então vai ser tudo relacional”

Datastore Choices

scgupta.link/datastores





"That's all Folks!"

[illegible]