

**MBA  
USP  
ESALQ**

**ENGENHARIA DE  
DADOS I**

Prof. Dr. Jeronymo Marcondes

**\*A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados é do professor.**

**Proibida a reprodução, total ou parcial, sem autorização. Lei nº 9610/98**

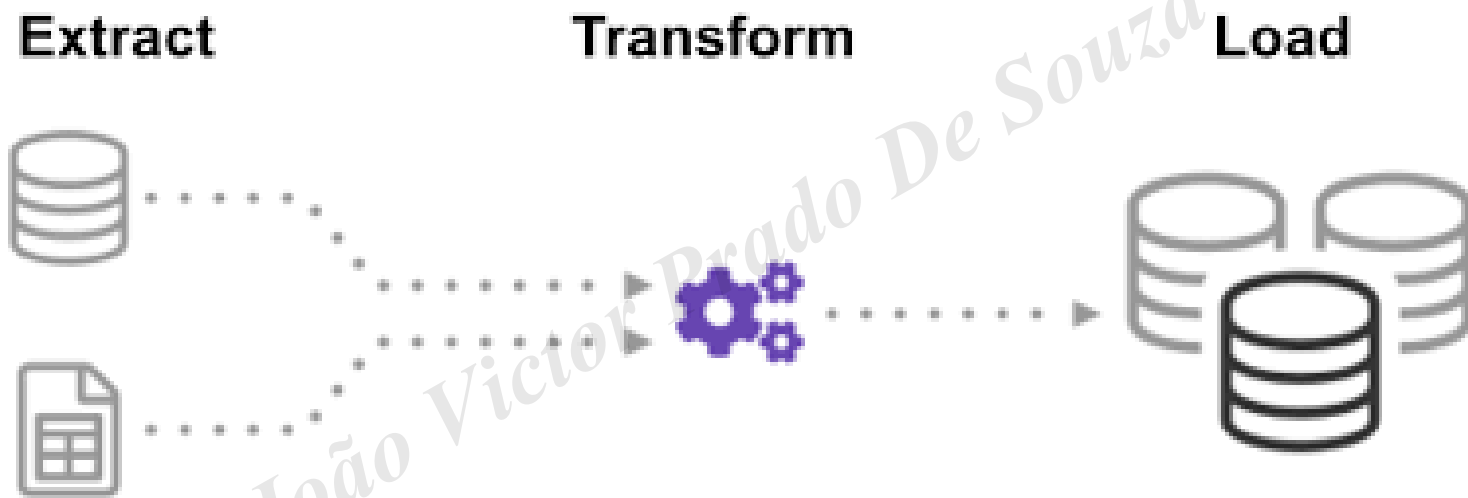
# Introdução

Por que estudar  
Engenharia de Dados?

Engenharia de Dados x  
Ciência de Dados.

O que é Engenharia de  
Dados?

# ETL



# Requisitos em grandes empresas

Um profissional de ciência de dados deve:

- Conhecer sobre estruturas de bancos de dados.
- Saber como funciona um processo de ETL.
- Entender sobre modelagem de bancos de dados.
- Compreender como funciona o uso de dados em produção.
- **Saber SQL.**

# Nosso objetivo

Introdução à  
estrutura de  
dados.

Banco de  
dados  
relacional.

SQL.

Modelo ERD –  
construção e  
interpretação.

O modelo e a  
álgebra  
relacional.

# Dados

- Dados x Informação.
- O que é um banco de dados?

É uma coleção de dados, que descreve, tipicamente, as atividades e relacionamentos de uma ou mais organizações.

Exemplo: MBA USP.

# SGBD

Sistema Gerenciador de Banco de Dados:

Software desenhado para auxiliar na manutenção, organização e coleta dos dados existentes em um banco de dados.

Exemplo: MySQL.



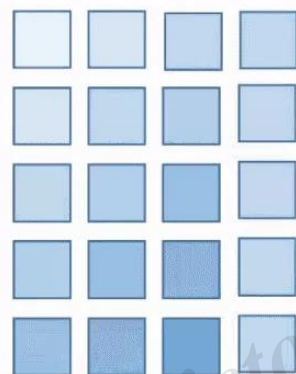
# Estruturas de Dados

Os dados que podemos utilizar dividem-se em:

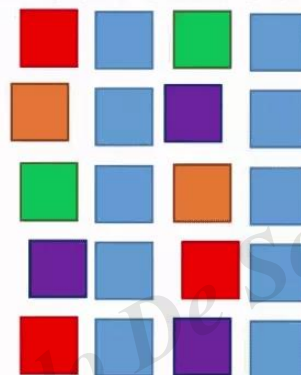
- Dados Estruturados.
- Dados Semiestruturados.
- Dados não estruturados.

## Structured, Unstructured and Semi-Structured

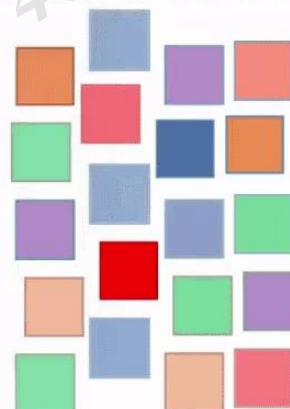
Structured Data



Semi-Structured Data



Unstructured Data



Fonte: <https://www.astera.com/pt/tipo/blog/dados-semiestruturados-e-n%C3%A3o-estruturados-estruturados/>

Estruturados - são os dados que detêm formatos bem definidos, como os extraídos de planilhas ou bancos de dados relacionais no formato SQL.

Semiestruturados – Semelhantes aos dados estruturados, mas não obedientes na totalidade quanto à forma. Nesta linha estão os registros de linguagens baseadas em HTML e XML.

Não estruturados ou NoSQL - não possuem um formato específico, são os dados coletados na sua forma original, como um texto, um vídeo, um fragmento de e-mail, um log de sistema ou ainda uma mera foto.

# Dados Estruturados

CPF	Nome	Nota
x	Zé das couves	10
y	Maria das desgraças	2
h	Silvio Santos	5

# Dados Semiestruturados

```
[
  {
    "CPF": "x",
    "NOME": "Zé das couves",
    "NOTA": "10",
    "TELEFONE" : "não é da sua conta"
  },
  {
    "CPF": "y",
    "NOME": "Maria das desgraças",
    "NOTA": "2"
  },
  {
    "CPF": "h",
    "NOME": "Silvio Santos",
    "NOTA": "5",
    "RENDA": "Muito alta"
  }
]
```

# Dados Não Estruturados



# SGBD Relacional

Nosso foco será em SGBD relacional.

Vantagens no uso de um SGDB:

- Independência.
- Eficiência.
- Integridade e segurança.
- Administração simplificada dos dados.
- Controle de acesso.

# Modelo de dados

- Dados “guardados” no banco de dados conforme modelo. O SGDB nos permitirá olhar este modelo e fazer consultas conforme lógica pré-estabelecida.
- A descrição dos dados em termos de modelos é o que chamamos de **esquema (SCHEMA)**. Conforme exemplo abaixo:

Estudantes(CPF: string, Nome: string, Nota: Integer)



# Tipos de dados

- Os tipos de dados são classificados em diferentes categorias e permitem N formatos. Aqui iremos apresentar somente os mais comuns.
- Integer ou inteiro. Exemplo: 1, 2, etc.
- Float. Exemplo: 0.10, 10.25, etc.
- String. Exemplo: “bom dia”, “meu nome é”, etc.
- Date. Exemplo: 2021-01-01.
- Caso do VARCHAR e CHAR.

# Modelo de dados

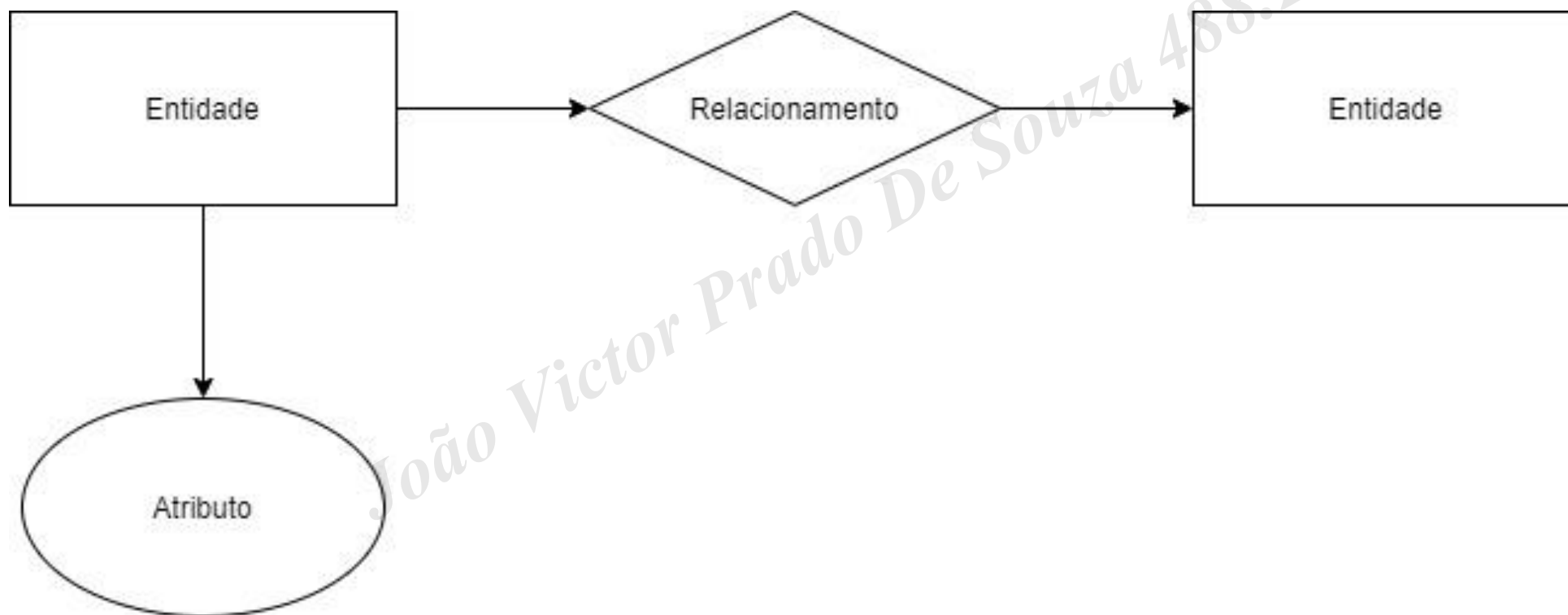
Estudantes(CPF: string, Nome: string, Nota: Integer)

- Isso nos diz que trata-se de uma tabela com três campos.
- Modelo relacional implica que cada registro é único.
- Restrições de integridade!

# Níveis de Abstração

## Modelo conceitual

- Mais alto nível.
- Mais próximo da realidade do negócio.
- Descreve os relacionamentos entre as entidades presentes em um banco de dados.



# Definições ERD

- Entidade: Algo que pode ser definido e que pode ter dados armazenados sobre ele — como uma pessoa, um objeto, conceito ou evento. Pense em entidades como substantivos. Exemplos: um cliente, estudante, carro ou produto.

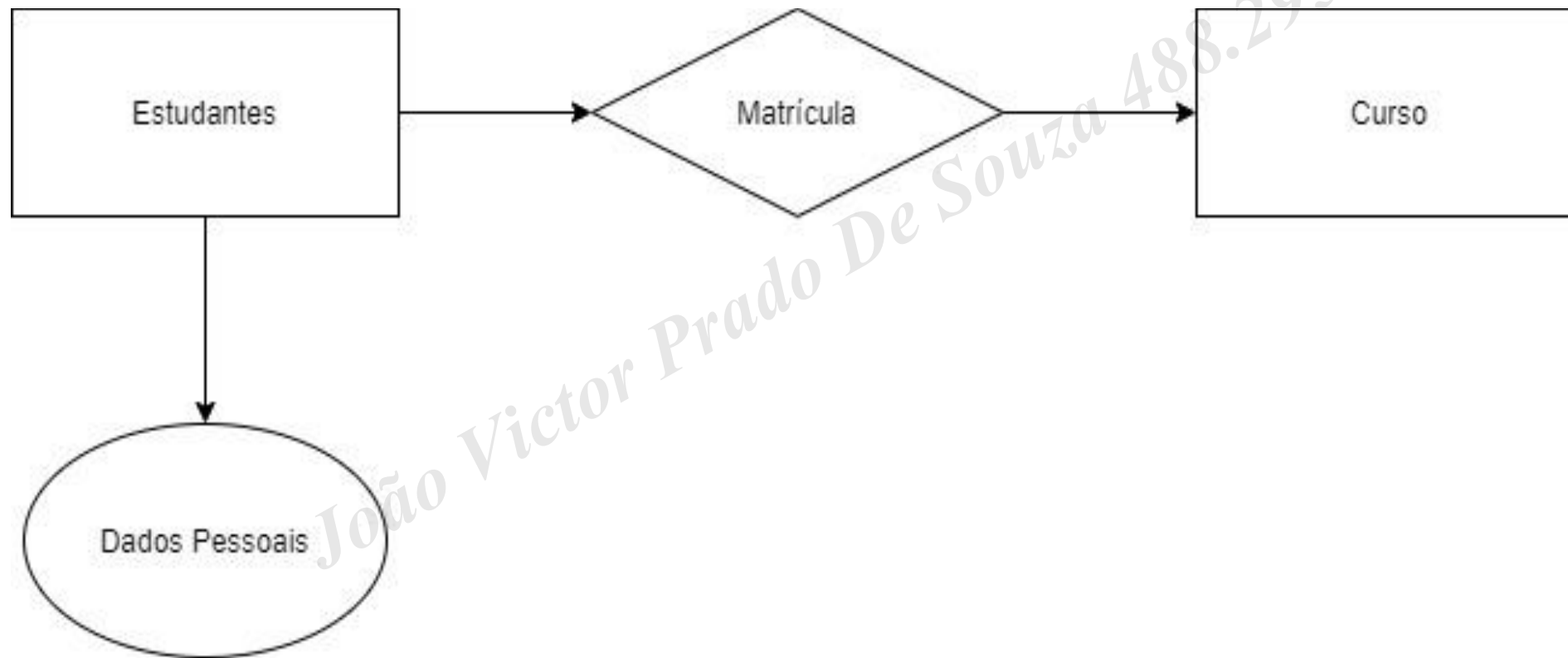
# Definições ERD

- Relacionamento: Como entidades atuam umas sobre as outras ou estão associadas uma com a outra. Pense em relacionamentos como verbos. Por exemplo, o estudante pode se inscrever em um curso. As duas entidades seriam o aluno e o curso, e o relacionamento descrito é o ato de matricular-se, assim conectando as duas entidades.

# Definições ERD

- Atributo: A propriedade ou característica de uma entidade, muitas vezes representada por um oval ou círculo.

# Exemplo ERD



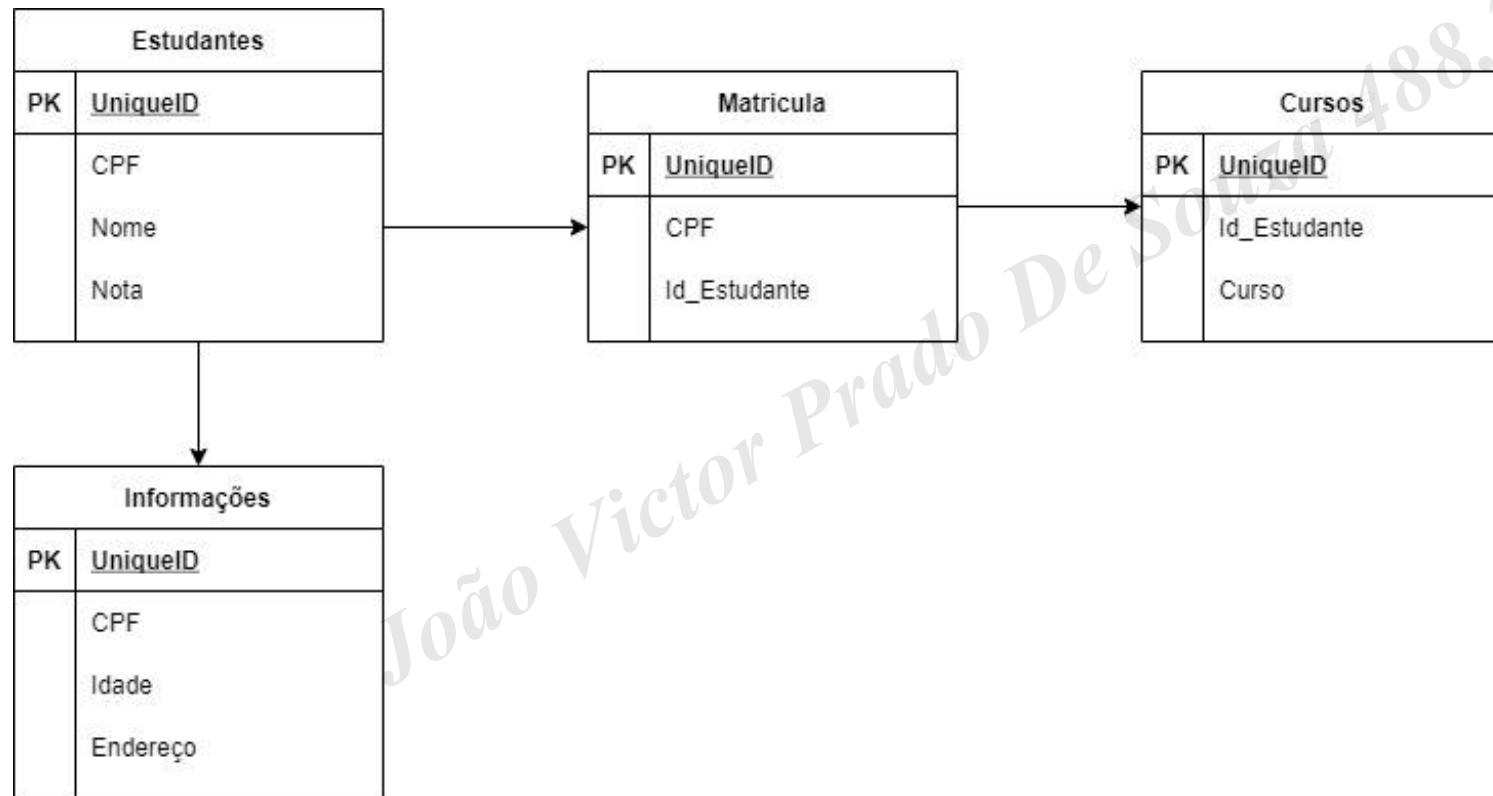


# Níveis de Abstração

## Modelo Lógico

- Como efetivamente os dados estarão dispostos em tabelas no banco de dados.
- Leva em conta limitações do banco e SGBD.
- Define chaves primárias, estrangeiras e restrições de integridade.

# Esquema Lógico



# Níveis de Abstração

## Modelo Físico

- Implementação própria dita. Como inserir os dados e criar tabelas e todo o esquema.
- Mais baixo nível.
- Como serão armazenados os dados.
- Métodos de restauração, backup.

# Query

Dada a existência de um banco de dados, podemos perguntar:

- Quantos estudantes estão matriculados em um curso?
- Quantos cursos estão ativos?
- Qual a idade média dos estudantes?
- Qual a idade média dos estudantes de um determinado curso?

**Entra SQL.**

# SQL

- DML – data manipulation language.
- Universalmente aceita.
- Própria para usar álgebra relacional.

# Introdução ao SQL

João Victor Prado De Souza 488.295.098-76

# SQL

- Structured Query Language.
- Origem – IBM.
- Não precisamos da forma como chegar no resultado – definimos o resultado.
- Linguagem declarativa.

# Aspectos Importantes

DML – manipulação de dados.

DDL – definição de dados.

Acesso remoto a bases de dados.

Gerenciamento de transações.

Segurança.



# Forma básica de uma query

**SELECT** [*DISTINCT*] **lista-seleção**

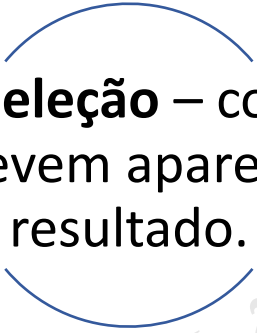
**FROM** **lista-origem**

**WHERE** **qualificação**

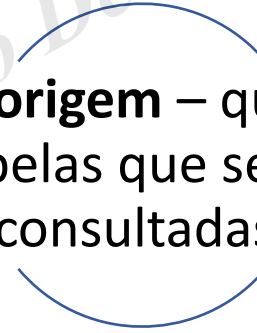
Tabela: ***Alunos***

CPF	Nome	Nota
x	Zé das couves	10
y	Maria das desgraças	2
h	Silvio Santos	5

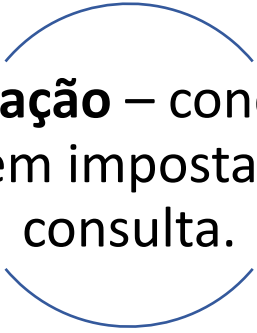
# Componentes



**lista-seleção** – colunas que devem aparecer no resultado.



**lista-origem** – quais as tabelas que serão consultadas.



**Qualificação** – condições a serem impostas na consulta.

# Exemplo 1

- Como obter uma tabela com CPF e notas?

**SELECT CPF, Nota**

**FROM Alunos**

# Observações

- Nome do campo tem que ser exato.
- SQL é case insensitive.
- Separe o nome das colunas por vírgulas.

# Lista- origem e Alias

- Alias é o “apelido”. Muito usado em SQL.
- Você pode utilizá-lo para facilitar o entendimento de sua query.

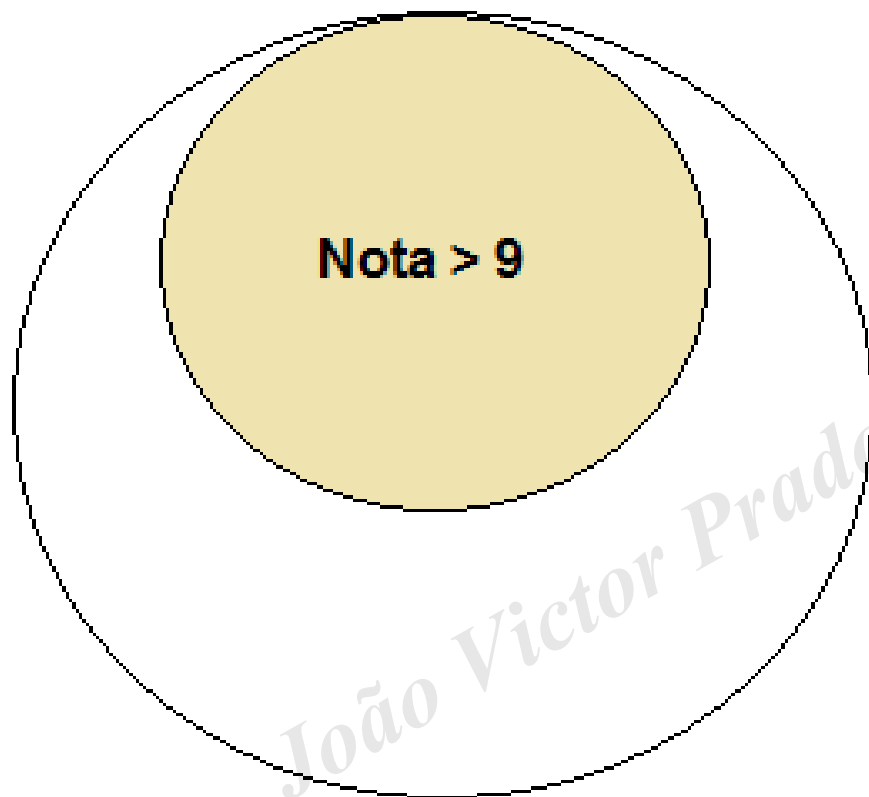
**SELECT A.CPF, A.Nota**

**FROM Alunos A**

# Qualificação

- A qualificação são as famosas cláusulas “where”.
- Essas são uma combinação de expressões booleanas de condições na forma de expressões.
- Em termos de álgebra, são definições de subconjuntos.

**Alunos**





# Operadores de Comparação

Operador	Significado
=	Igual a
> (Maior que)	Maior que
< (Menor que)	Menor que
>= (Maior ou igual a)	Maior ou igual a
<= (Menor ou igual a)	Menor que ou igual a
<> (Diferente de)	É diferente de

## Exemplo 2

- Como obter uma tabela com CPF e notas maiores do que 9?

```
SELECT CPF, Nota  
FROM Alunos  
WHERE Nota > 9
```

# Mais de uma cláusula

- Neste caso, precisamos definir como é a relação entre as cláusulas.
- Suponha que tenhamos 2 condições: condição-1 e condição-2.
- AND => as duas tem de ser verdade ao mesmo tempo.
- OR => uma das duas tem de ser verdade.

Operador	Significado	Exemplo
AND	e	Condição-1 AND condição-2
OR	ou	Condição-1 OR condição-2

## Exemplo 3

CPF	NOME	NOTA	IDADE
XX	JOÃO	10	20
YY	PEDRO	7	30

- Como obter todos os registros com nota maior do que 6 E idade maior do que 25?

**SELECT \***

**FROM Alunos**

**WHERE Nota > 6 AND Idade > 25**

## Exemplo 4

CPF	NOME	NOTA	IDADE
XX	JOÃO	10	20
YY	PEDRO	7	30

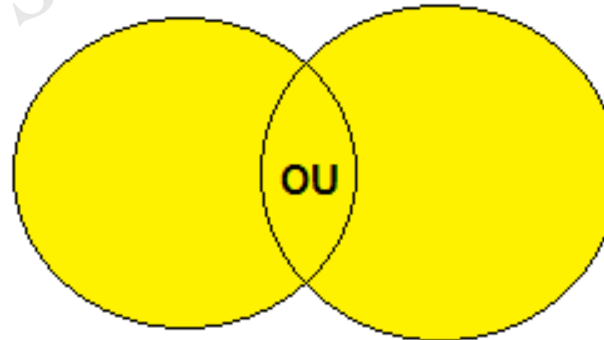
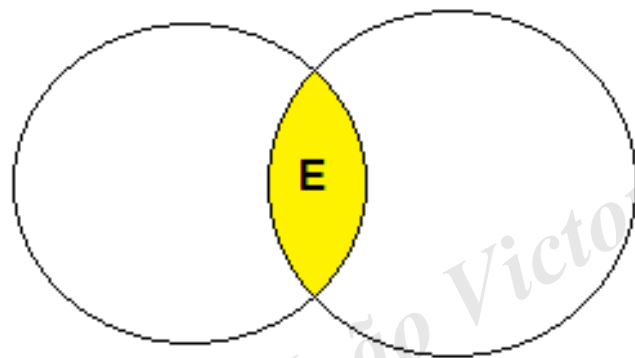
- Como obter todos os registros com nota maior do que 9 **OU** idade maior do que 25?

**SELECT \***

**FROM Alunos**

**WHERE Nota > 9 OR Idade > 25**

# Diagrama de Venn



# Caso de linhas repetidas

PIS	NOME
xxx	Pedro
xxx	Pedro

- Problema na hora de seleção quando ocorrem linhas com alguma coluna de valores repetidos.
- Uso do DISTINCT.

```
SELECT DISTINCT NOME  
FROM Alunos
```

# Agregação

Como agregar valores por operações.

Operações de sumarização de dados:

- Média.
- Mínimo.
- Máximo.
- Etc.



# COUNT

- Conta a quantidade de registros sob determinadas condições.
- Segue a seguinte lógica:

**SELECT COUNT**([Campo Contado]), Campos Agrupados  
**FROM Tabela**  
**GROUP BY Campos Agrupados**

## Exemplo 5

CPF	NOME	ESTADO
XXX	JOÃO	SP
YYY	PEDRO	SP
HHH	MARIANA	AL
JJJ	FLAVIA	RJ

**SELECT COUNT(CPF)**  
**FROM Alunos**

4

## Exemplo 6

CPF	NOME	ESTADO
XXX	JOÃO	SP
YYY	PEDRO	SP
HHH	MARIANA	AL
JJJ	FLAVIA	RJ

**SELECT COUNT(CPF), ESTADO**  
**FROM Alunos**  
**GROUP BY ESTADO**

2	SP
1	AL
1	RJ

# Exemplo 7

CPF	NOME	ESTADO
XXX	JOÃO	SP
YYY	PEDRO	SP
HHH	MARIANA	AL
JJJ	FLAVIA	RJ

**SELECT COUNT(CPF) AS contagem, ESTADO**  
**FROM Alunos**  
**GROUP BY ESTADO**  
**ORDER BY contagem**

1	AL
1	RJ
2	SP

# SUM

- Soma a quantidade de registros sob determinadas condições.
- Segue a seguinte lógica:

```
SELECT SUM([Campo Somado]), Campos Agrupados  
FROM Tabela  
GROUP BY Campos Agrupados
```

# Exemplo 8

CPF	NOME	ESTADO	IDADE
XXX	JOÃO	SP	20
YYY	PEDRO	SP	30
HHH	MARIANA	AL	30
JJJ	FLAVIA	RJ	40

**SELECT SUM(IDADE)**  
**FROM Alunos**

120

# Exemplo 9

CPF	NOME	ESTADO	IDADE
XXX	JOÃO	SP	20
YYY	PEDRO	SP	30
HHH	MARIANA	AL	30
JJJ	FLAVIA	RJ	40

**SELECT SUM**(IDADE), ESTADO  
**FROM** Alunos  
**GROUP BY** ESTADO

50	SP
30	AL
40	RJ

# AVERAGE (AVG)

- Tira a média aritmética dos registros sob determinadas condições.
- Segue a seguinte lógica:

```
SELECT AVG([Campo]), Campos Agrupados  
FROM Tabela  
GROUP BY Campos Agrupados
```



# Exemplo 9

CPF	NOME	ESTADO	IDADE
XXX	JOÃO	SP	20
YYY	PEDRO	SP	30
HHH	MARIANA	AL	30
JJJ	FLAVIA	RJ	40

**SELECT** **AVG**(IDADE), ESTADO  
**FROM** Alunos  
**GROUP BY** ESTADO

25	SP
30	AL
40	RJ

# MIN e MAX

- Tira o menor ou o maior valor dos registros sob determinadas condições.
- Segue a seguinte lógica:

```
SELECT MIN([Campo]), Campos Agrupados  
FROM Tabela  
GROUP BY Campos Agrupados
```

```
SELECT MAX([Campo]), Campos Agrupados  
FROM Tabela  
GROUP BY Campos Agrupados
```

# Exemplo 10

CPF	NOME	ESTADO	IDADE
XXX	JOÃO	SP	20
YYY	PEDRO	SP	30
HHH	MARIANA	AL	30
JJJ	FLAVIA	RJ	40

**SELECT MAX**(IDADE), ESTADO  
**FROM** Alunos  
**GROUP BY** ESTADO

30	SP
30	AL
40	RJ

# Exemplo 11

CPF	NOME	ESTADO	IDADE
XXX	JOÃO	SP	20
YYY	PEDRO	SP	30
HHH	MARIANA	AL	30
JJJ	FLAVIA	RJ	40

**SELECT MIN**(IDADE), ESTADO  
**FROM** Alunos  
**GROUP BY** ESTADO

20	SP
30	AL
40	RJ

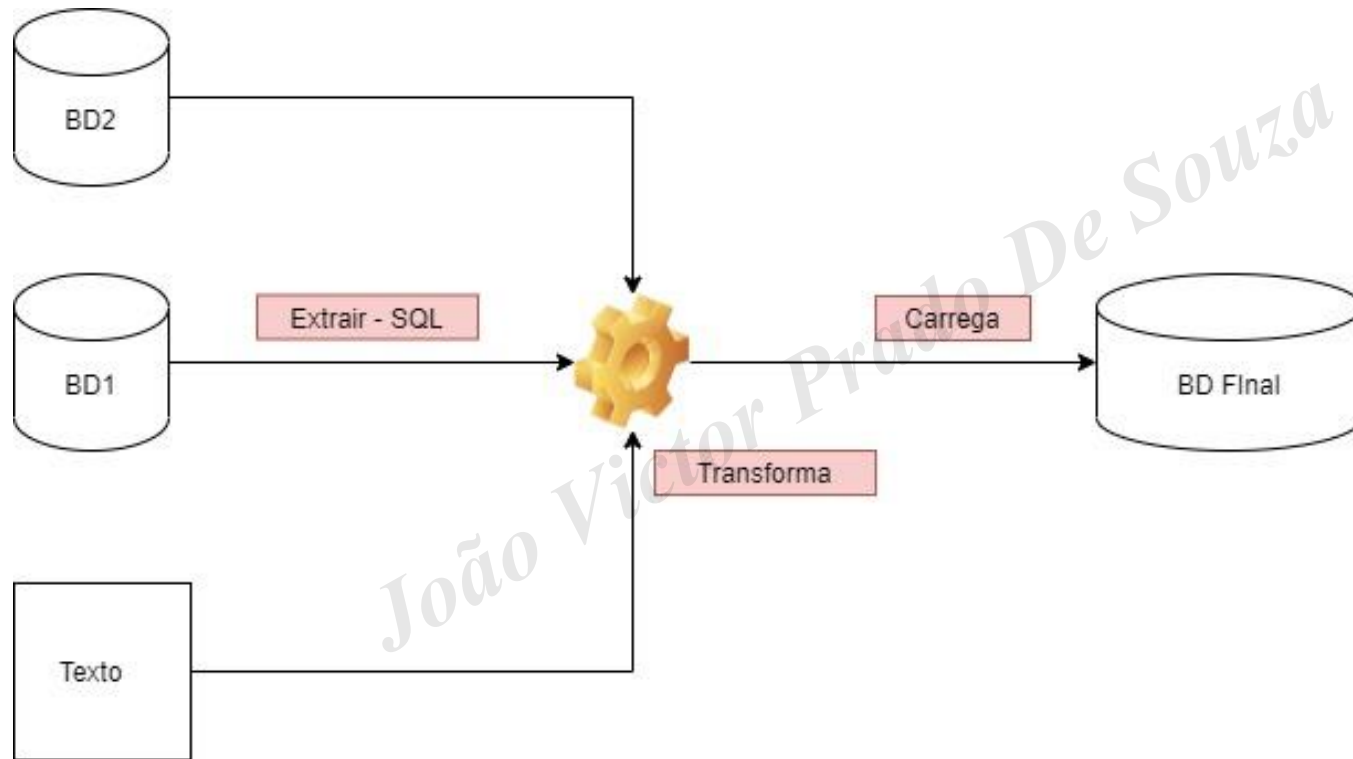
# Introdução ao ETL



Extract, Transformation and Load.

A integração de dados ETL é um processo de três etapas em que os dados são extraídos de uma ou mais fontes de dados, convertidos para o estado necessário e carregados em um banco de dados ou data warehouse em nuvem.

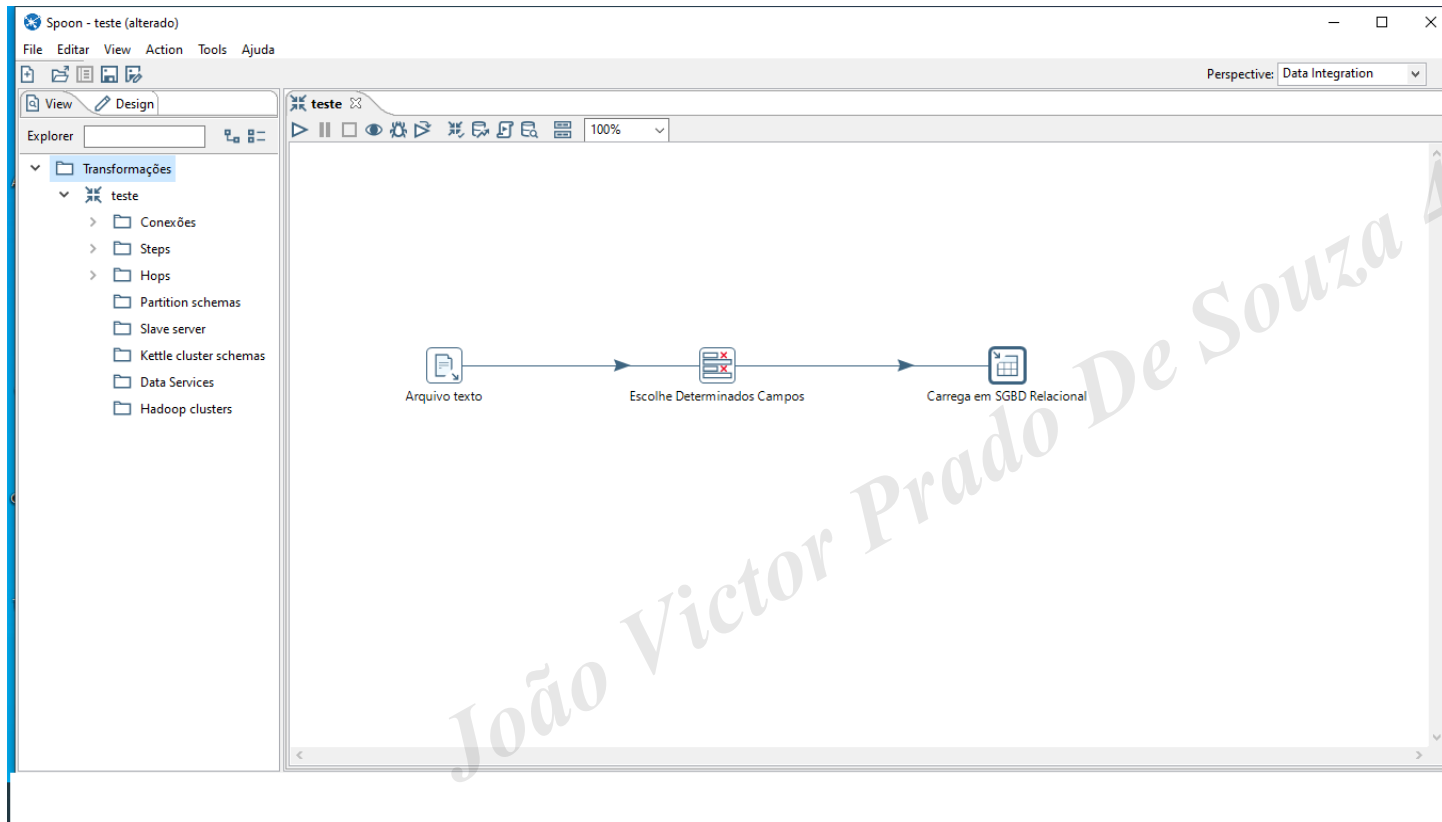
# Estrutura de um ETL



# Frameworks para ETL



# Exemplo Pentaho





# Create Table

- Cria uma Tabela com determinados campos.
- Segue a seguinte lógica:

```
CREATE TABLE ENDERECO  
(  
    Id_Estudante INTEGER,  
    Endereco Varchar(50)  
)
```

# Insert

- Insere determinados campos.
- Segue a seguinte lógica:

```
INSERT INTO ENDERECO (Id_Estudante, Endereco)  
VALUES (2, "RUA PEIXOTO DA SILVA")
```

**OBRIGADO!**

[linkedin.com/in/jeronymo-marcondes-585a26186](https://www.linkedin.com/in/jeronymo-marcondes-585a26186)