

# CONSOLIDAÇÃO DE CONCEITOS

**D** **DESENVOLVE**  
programa de formação  
e inclusão de talentos  
em tecnologia



 **Grupo  
Boticário**



**KORU**





# DADOS ESTRUTURADOS

---

## Definição

---

Dados com **modelo ou estrutura pré-determinada**, organizados de forma rígida e consistente.

## Características

---

- Armazenados em tabelas
- Seguem o Modelo Relacional
- Manipulados com SQL

## Componentes

---

- Tabelas (entidades)
- Colunas (atributos)
- Linhas (registros)



# TIPOS DE DADOS EM SQL

## Strings

- CHAR
- VARCHAR
- TEXT
- BLOB

## Numéricos

- INT, SMALLINT, BIGINT
- DECIMAL, NUMERIC
- FLOAT

## Data e Hora

- DATE
- TIME
- DATETIME
- TIMESTAMP

## Lógicos

- BOOLEAN
- 1/0
- 'T'/'F'

## Data Types



# CONSULTAS SQL BÁSICAS

---



## SELECT

Especifica quais colunas recuperar dos dados.

---



## FROM

Identifica as tabelas de origem dos dados.

---



## WHERE

Filtra dados com condições específicas.

---



## ORDER BY

Classifica o conjunto de resultados final.



# OPERADORES DE COMPARAÇÃO

Operador	Significado	Exemplo
=	Igual a	WHERE idade = 30
!= ou <>	Não igual a	WHERE status <> 'Inativo'
< e >	Menor/Maior que	WHERE preco > 100
<= e >=	Menor/Maior ou igual	WHERE data <= '2023-01-01'

# COMBINANDO TABELAS COM JOINS

---



# ORDEN DE EXECUÇÃO SQL

---



## FROM

Identifica tabelas e processa JOINS



## WHERE

Filtra as linhas do conjunto intermediário



## GROUP BY

Agrupa linhas para agregações



## HAVING

Filtra os grupos criados



## SELECT

Seleciona colunas finais



## ORDER BY

Classifica o conjunto de resultados



# SQL VS NOSQL

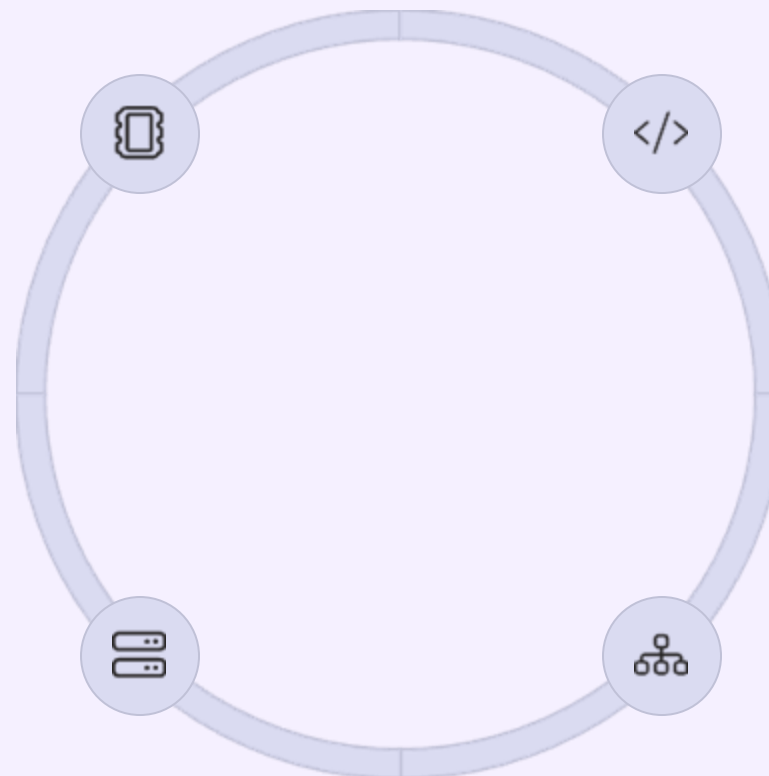
---

## Modelo de Dados

- SQL: Tabelas relacionais
- NoSQL: Chave-valor, documento, colunar, grafo

## Escalabilidade

- SQL: Vertical (principalmente)
- NoSQL: Horizontal (distribuída)



## Linguagem

- SQL: Linguagem SQL padronizada
- NoSQL: Linguagens específicas (MQL, Cypher)

## Esquema

- SQL: Esquema rígido, predefinido
- NoSQL: Schema-free ou flexível

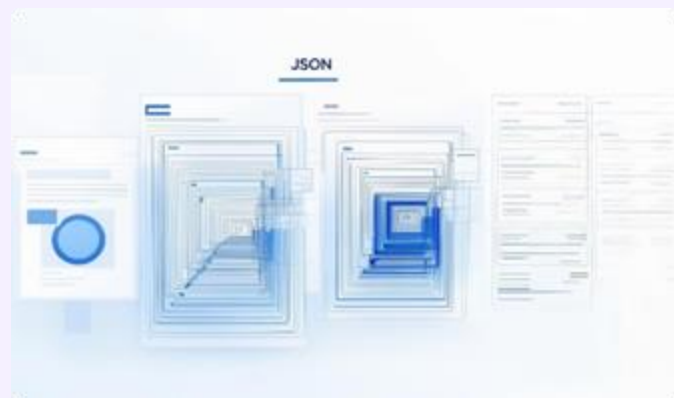
# MODELOS NOSQL

---



## Key-Value Stores

Forma mais simples de armazenar dados. Associa um valor a uma chave única.



## Document Databases

Armazena documentos (JSON) autocontidos em coleções. Extremamente flexível.



## Graph Databases

Focado em relacionamentos. Dados armazenados como nós e arestas.

# INGESTÃO DE DADOS

## Extração

Obtenção de dados de fontes externas em diversos formatos.

- Arquivos (CSV, JSON, XML)
- APIs
- Outros bancos de dados

## Transformação

Preparação dos dados para armazenamento.

- Limpeza
- Normalização
- Enriquecimento

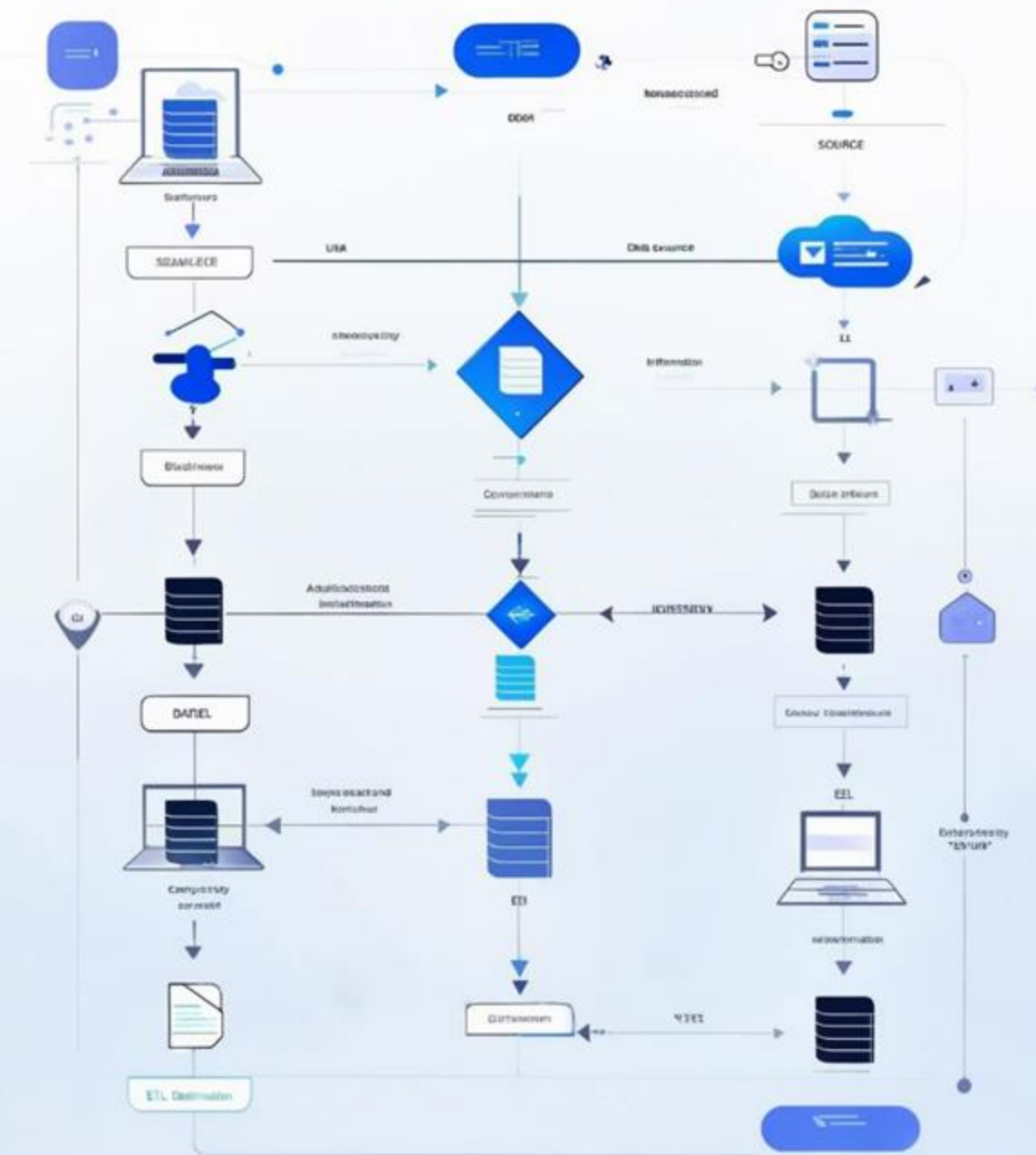
## Carregamento

Inserção dos dados no sistema de destino.

- INSERT INTO (volumes pequenos)
- BULK LOAD (grandes volumes)
- APIs específicas (NoSQL)

## Data Pipeline

Enterprise for Digital Transformation. View how the ETL (Extract, Transform, Load) data integration model processes data from its internal transformation and integration algorithms. Many organizations data starts, process to create them with data wrapping, transformation.



# ETL VS ELT

---

## ETL

Extract, Transform, Load

## Transformação Prévia

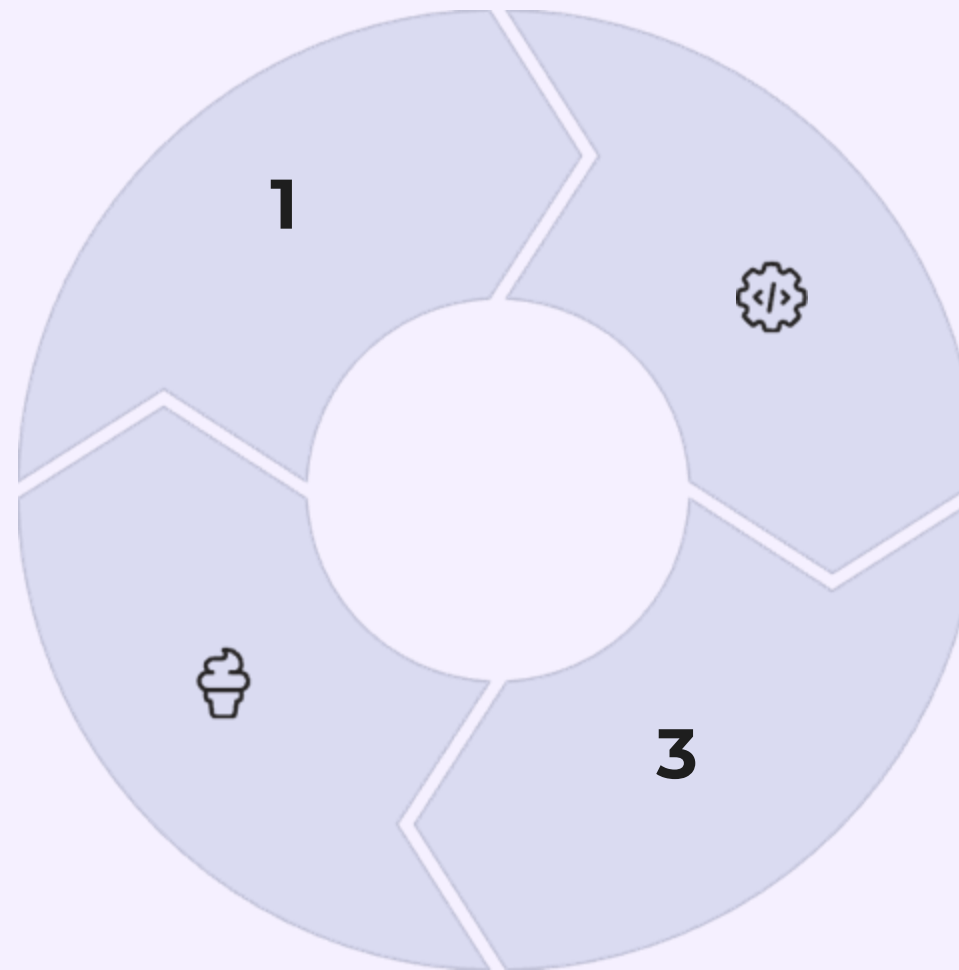
Dados transformados antes do carregamento

## ELT

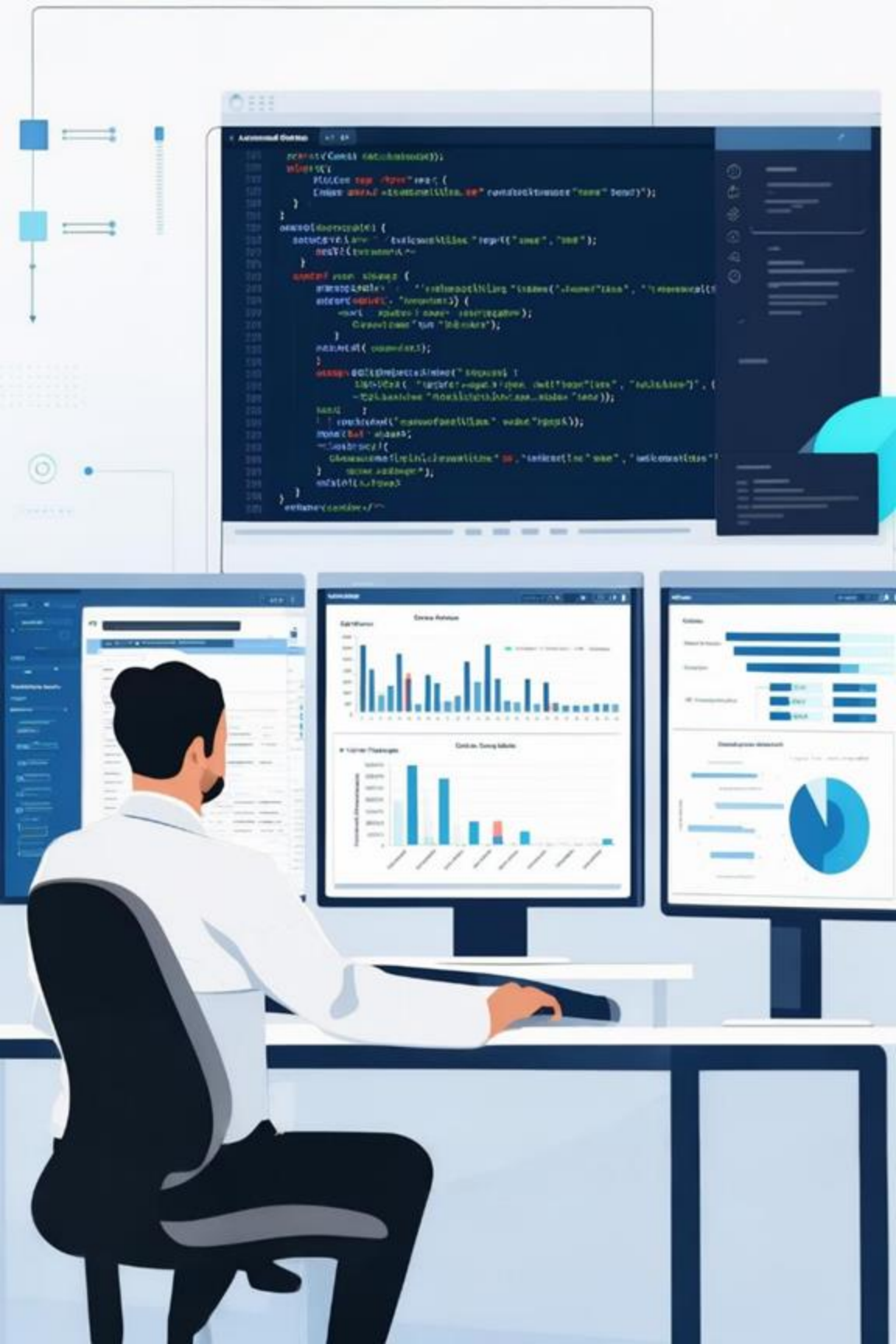
Extract, Load, Transform

## Transformação Posterior

Dados transformados dentro do banco de dados



# AUTOMAÇÃO DE CONSULTAS SQL



## 1

### Scripts SQL

Arquivos de texto com instruções SQL para reuso e versionamento.

## 2

### Views

Consultas SQL salvas no banco de dados como tabelas virtuais.

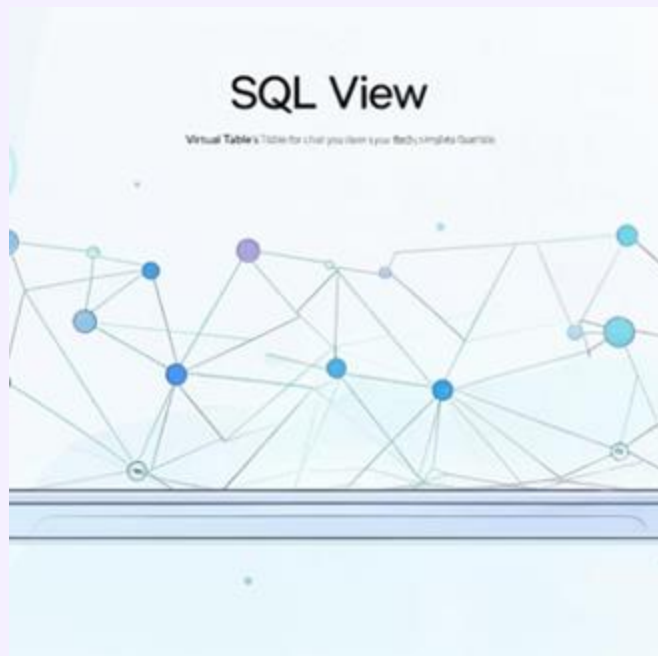
## 3

### Integração

Conexão com ferramentas de BI e linguagens de programação.



# BENEFÍCIOS DAS VIEWS SQL



As Views simplificam consultas complexas, garantem consistência nas definições de dados, criam conjuntos específicos para análise e fornecem uma camada adicional de segurança.



**Momento de  
avaliar a aula  
de hoje!**