



Aula # 1 - Coletando dados

# Aula	11
<input checked="" type="checkbox"/> Ready	<input checked="" type="checkbox"/>

▼ 1. O armazenamento dos dados

▼ 1.1. Tabela completa

ID da Transação	Data da compra	Valor da transação	Moeda	Nome do Comprador	Nome do Vendedor	Nome	Idade
AZ102931	29/04/2022	650	Real	Meigarom	Amazon	Meigarom	33

▼ 1.2. Tabela de clientes

Nome	Idade	Estado Civil	Cidade residência	Salário mensal	Número de filhos
Meigarom	33	Casado	Indaiatuba	35000	0

▼ 1.3. Tabela de produtos

Nome do Produto	Marca	Preço	Material	Peso	Dimensões	Custo de fabricação	Garantia
Fone de Ouvido	Bozer	650	Alumínio	30g	40cm x 2mm	100	1 ano

▼ 1.4. Tabela de empresa

Nome da Empresa	Faturamento Anual	Custo anual	Valor de mercado
Amazon	1000000	500000	22000000

▼ 1.5. Tabela de transações

ID da Transação	Data da compra	Valor da transação	Moeda	Nome do Comprador	Nome do Vendedor
AZ102931	29/04/2022	650	Real	Meigarom	Amazon

▼ 2. Banco de Dados

Um banco de dados relacional é um banco de dados que modela os dados de uma forma que eles sejam percebidos pelo usuário como tabelas, ou mais formalmente relações. (Wikipedia)

Banco de Dados



Relacional

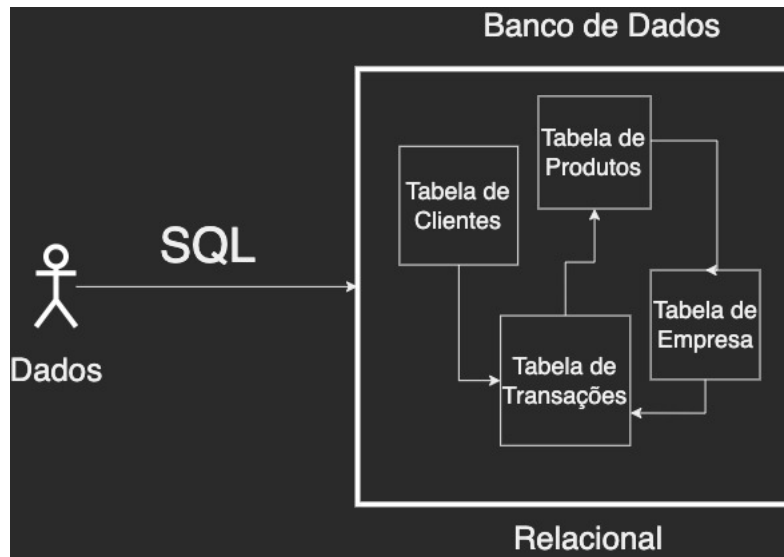
▼ 3. O que é a linguagem SQL

A linguagem SQL (Structured Query Language) é uma linguagem de programação utilizada para gerenciar e manipular bancos de dados relacionais.

Com o SQL, é possível:

1. Criar, alterar e consultar dados em tabelas de um banco de dados.
2. Atualizar linhas, criar colunas, modificar estruturas de tabelas.
3. Fazer um cópia dos dados do banco de dados para o computador.

A sintaxe do SQL é baseada em comandos simples e intuitivos, permitindo a interação com o banco de dados de forma eficiente e organizada.



▼ 4. A anatomia da consulta SQL

Para realizar qualquer consulta SQL, é necessário fornecer 3 informações ao banco de dados:

1. **Nome da tabela(s)** no qual os dados estão armazenados
2. **Nome das colunas** que deseja consultar
3. **Filtro de seleção** das linhas

▼ 4.1. Uma típica consulta SQL:

```
SELECT nome das colunas
FROM   tabela de origem
WHERE  filtro de linhas
```

▼ 4.1.1. Por exemplo

▼ 1.1.2. Tabela de transações (tabela_transacoes)

ID da Transação	Data da compra	Valor da transação	Moeda	Nome do Comprador	Nome do Vendedor
AZ102931	29/04/2022	650	Real	Meigarom	Amazon

▼ 1.1.3. Consulta na tabela

```
SELECT
    ID da Transação,
    Data da compra,
    Valor da transação,
    Moeda,
    Nome do Comprador,
    Nome do Vendedor
FROM
    tabela_transacoes
WHERE
    Valor da transação > 650
```

▼ 4.2. Cuidados

1. A última linha do SELECT, **NÃO TEM VÍRGULA!**.
2. Obrigatório **NOMEAR A TABELA!**.
3. Sempre **INDICAR A TABELA FONTE** da qual a coluna está sendo extraída.
4. As strings são **INDICADAS POR ASPAS SIMPLES**.
5. O símbolo de * indica **TODAS AS COLUNAS**

▼ 5. Coletando dados do banco de dados

1. Selecione as colunas: *loyalty_number*, *year*, *month*, *flights_booked*, *total_flights*, *distance* e *points_accumulated* da tabela *"flight_activity"*.
2. Selecione as mesmas colunas, porém, recupere somente as linhas cuja coluna *distance* é maior do que 2000, da tabela *"flight_activity"*
3. Selecione as mesmas colunas, porém, recupere somente as linhas cuja coluna *distance* é maior do que 2000 e a coluna *month* é igual a 9, da tabela *"flight_activity"*
4. Selecione as mesmas colunas, porém, recupere somente as linhas cuja coluna *distance* é maior do que 2000 ou a coluna *points_accumulated* é menor que 100, da tabela *"flight_activity"*.
5. Selecione as mesmas colunas, porém, recupere somente as linhas cuja coluna *loyalty_card* é igual a *Star* da tabela *"flight_loyalty_history"*.

▼ 5.1. Exercícios da Aula 01

- ▼ 1. Selecione as colunas: *loyalty_number*, *country*, *city*, *gender* e *loyalty_card*, *salary* da tabela *"flight_loyalty_history"*.

```
# consulta dos dados no banco de dados
consulta_atividade = """

SELECT
    flh.loyalty_number,
    flh.country,
    flh.city,
    flh.gender,
    flh.loyalty_card
FROM flight_loyalty_history flh
"""

df_atividade = pd.read_sql_query( consulta_atividade, conn )

# visualiza os dados
df_atividade.head()
```

- ▼ 2. Selecione as mesmas colunas, porém, recupere somente as linhas cuja coluna *gender* é igual a *Female* da tabela *"flight_loyalty_history"*.

```
# consulta dos dados no banco de dados
consulta_atividade = """

SELECT
    flh.loyalty_number,
    flh.country,
    flh.city,
    flh.gender,
    flh.loyalty_card
```

```

FROM
    flight_loyalty_history flh
WHERE
    flh.gender = 'Female'
"""

df_atividade = pd.read_sql_query( consulta_atividade, conn )

# visualiza os dados
df_atividade.head()

```

▼ 3. Selecione as colunas: *loyalty_number*, *month*, *year*, *distance*, *flights_booked* e *total_flights* da tabela “*flight_activity*”, e recupere somente as linhas cuja coluna “*flights_booked*” é maior do que 10 e a coluna “*flights_booked*” é menor do que 12, da tabela “*flight_activity*”

```

# consulta dos dados no banco de dados
consulta_atividade = """

SELECT
    fa.loyalty_number,
        fa.month,
    fa.year,
    fa.distance,
    fa.flights_booked,
        fa.total_flights
FROM
    flight_activity fa
WHERE
    fa.flights_booked > 10 AND fa.flights_booked < 12
"""

df_atividade = pd.read_sql_query( consulta_atividade, conn )

# visualiza os dados
df_atividade.head()

```

▼ 6. Resumo

1. Uma **consulta SQL** seleciona colunas e filtra linhas a partir de um tabela origem.
2. A cláusula SELECT é responsável por **selecionar as colunas**.
3. A cláusula FROM é responsável por **apontar a tabela origem**.
4. A cláusula WHERE é responsável por **filtrar linhas**.
5. **Tome cuidado** com a sintaxe do SQL.

▼ 7. Próximos passos

1. **Coleta de Dados**: Unir as duas tabelas e coletar os dados
2. **Análise de Dados**: Inspeccionar os dados e selecionar colunas e filtrar linhas
3. **Preparar dos dados**: Treinamento do algoritmo de Machine Learning