

**INSTITUTO
FEDERAL**

Baiano

Campus
Guanambi

BANCO DE DADOS



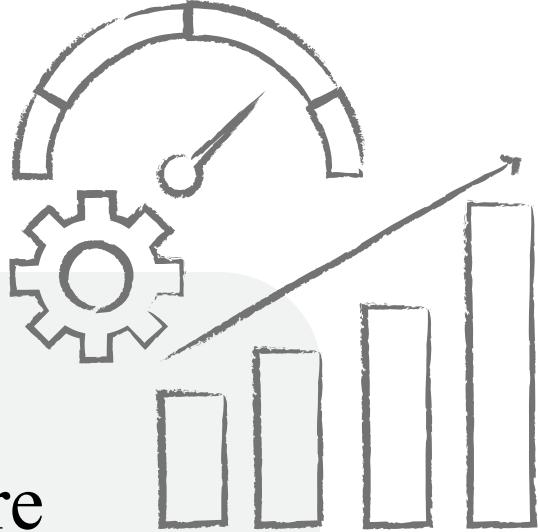
Apresentação por : Arthur Guimarães, Daniel Oliveira, Davi Carvalho e Jâmerson de Souza

Prof^a: Moniky Ribeiro

TÓPICOS

- INTRODUÇÃO;
- ESTRUTURA DAS TABELAS;
- PROBLEMAS IDENTIFICADOS NO BANCO DE DADOS;
- DEFINIÇÃO DA PK;
- DEFINIÇÃO DAS FKS;
- MODIFICAÇÃO NO TIPO DE DADOS;
- CONSULTAS SQL.

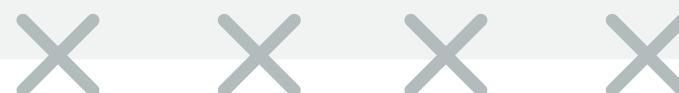
INTRODUÇÃO



O objetivo deste projeto foi explorar e manipular uma base de dados real sobre ocorrências aeronáuticas no Brasil, aplicando os conceitos aprendidos em Fundamentos de Banco de Dados, como modelagem, normalização, integridade referencial e consultas SQL.

Pontos-chave:

- Trabalhar com dados reais do governo brasileiro
- Aplicar conceitos de banco de dados
- Desenvolver habilidades em SQL
- Identificar e corrigir problemas estruturais



INTRODUÇÃO



Os dados foram obtidos no portal de dados abertos do governo brasileiro (dados.gov.br), especificamente do conjunto de dados sobre ocorrências aeronáuticas da aviação civil. A base contém 5 tabelas principais com informações sobre acidentes, aeronaves envolvidas, fatores contribuintes e recomendações de segurança.

Dados importantes:

- Fonte: Portal dados.gov.br
- 5 tabelas CSV: ocorrencia, aeronave, ocorrencia_tipo, fator_contribuinte, recomendacao
- 1 diagrama ER em formato PNG
- Ferramentas utilizadas: DataGrip e MySQL Workbench



ESTRUTURA DAS TABELAS

Estrutura:

tabela_ocorrencia (tabela central)

tabela_tipo_ocorrencia (tipos e categorias)

tabela_aeronave (aeronaves envolvidas)

tabela_fator_contribuinte (causas identificadas)

tabela_recomendacao_seguranca (medidas de segurança)

Relacionamentos:

- 1:N entre tabela_ocorrencia e tabela_aeronave
- 1:N entre tabela_ocorrencia e tabela_tipo_ocorrencia
- 1:N entre tabela_ocorrencia e tabela_fator_contribuinte
- 1:N entre tabela_ocorrencia e tabela_recomendacao_seguranca



PROBLEMAS IDENTIFICADOS NO BANCO DE DADOS

Ao analisar a estrutura inicial dos dados, identifiquei 6 problemas principais que precisavam ser corrigidos para garantir a integridade e qualidade do banco de dados.

1. REDUNDÂNCIA E INCONSISTÊNCIA EM CHAVES DE RELACIONAMENTO

Problema identificado:

- Tabela principal (tabela_ocorrecia) tinha 4 colunas duplicadas: codigo_ocorrecia1, 2, 3, 4
- Todas continham o mesmo valor do codigo_ocorrecia principal
- Exemplo: código 87402, 87403, 87405 → faltava o 87404

Impacto:

- Redundância desnecessária de dados
- Dificulta manutenção e consultas
- Gera confusão sobre qual coluna usar em JOINs

PROBLEMAS IDENTIFICADOS NO BANCO DE DADOS

2. AUSÊNCIA DE REGISTROS E VIOLAÇÃO DE INTEGRIDADE REFERENCIAL

Problema crítico:

- Códigos de ocorrência com lacunas não documentadas (ex: 87404 ausente)
- Tabelas secundárias referenciavam códigos inexistentes na tabela principal
- Exemplo: código 87404 aparecia duplicado em tabela_tipo_ocorrecia, mas não existia em tabela_ocorrecia

Impacto:

- Violação do princípio de integridade referencial
- Impossibilita criação de Foreign Keys
- Resulta em análises incompletas e estatisticamente enviesadas
- Compromete confiabilidade de consultas com JOINs

PROBLEMAS IDENTIFICADOS NO BANCO DE DADOS

3. INCOMPLETITUDE DE DADOS E CAMPOS COM VALORES NULOS

Um dos problemas mais preocupantes foi a presença massiva de valores NULL em campos críticos para análises técnicas.

Campos afetados:

- aeronave_motor_tipo: tipo de motor (pistão, turbina, etc)
- aeronave_pmd: peso máximo de decolagem
- aeronave_ano_fabricacao: ano de fabricação da aeronave
- Campos descritivos em tabela_fator_contribuinte

Impacto:

- Compromete análises técnicas sobre perfis de risco
- Impossibilita estudos sobre características específicas de aeronaves
- Limita análises preditivas e estudos de causalidade
- Dificulta identificação de padrões em fatores contribuintes

PROBLEMAS IDENTIFICADOS NO BANCO DE DADOS

4. INADEQUAÇÃO DO FORMATO DE ARMAZENAMENTO DE DATAS

Problema técnico:

- Datas armazenadas como TEXT/VARCHAR ao invés de DATE
- Formato brasileiro: dd/mm/aaaa (ex: "15/08/2024")
- Campos afetados: ocorrencia_dia, divulgacao_dia_publicacao

Limitações causadas:

- Impossibilita uso direto de funções temporais do MySQL
- Requer conversões com STR_TO_DATE() em todas as consultas
- Degradação de performance em análises de grandes volumes
- Dificulta consultas de tendências sazonais e intervalos

DEFINIÇÃO DA PK



```
ALTER TABLE tabela_ocorrencia ADD PRIMARY KEY (codigo_ocorrencia);
```

Define a coluna `codigo_ocorrencia` como chave primária da tabela.

Isso significa:

- valores únicos
- não podem ser nulos
- usados para identificar cada linha

DEFINIÇÃO DAS FKS

```
-- Criação de chaves estrangeiras nas tabelas externas
SET FOREIGN_KEY_CHECKS = 0;

ALTER TABLE tabela_tipo_ocorrencia
ADD FOREIGN KEY (codigo_ocorrencial) REFERENCES tabela_ocorrencia(codigo_ocorrencial);

ALTER TABLE tabela_aeronave
ADD FOREIGN KEY (codigo_ocorrencia2) REFERENCES tabela_ocorrencia(codigo_ocorrencia2);

ALTER TABLE tabela_fator_contribuinte
ADD FOREIGN KEY (codigo_ocorrencia3) REFERENCES tabela_ocorrencia(codigo_ocorrencia3);

ALTER TABLE tabela_recomendacao_seguranca
ADD FOREIGN KEY (codigo_ocorrencia4) REFERENCES tabela_ocorrencia(codigo_ocorrencia4);
```



MODIFICAÇÃO NO TIPO DE DADOS



- Aqui temos exemplos de alterações que fizemos para corrigir o tipo dos dados:
 - VARCHAR(100) para cidades e nomes
 - VARCHAR(5) para UF
 - INT para contadores
 - DATE para datas

CONSULTAS SQL



```
-- Listar todas as ocorrências  
select * from tabela_ocorrencia;
```

CONSULTAS SQL

RESULTADO:

CONSULTAS SQL



```
-- Listar aeronaves de uma ocorrência específica  
SELECT * FROM tabela_aeronave  
WHERE codigo_ocorrenca2 = 87104;
```

CONSULTAS SQL

RESULTADO:



codigo_ocorrencia2	aeronave_matricula	aeronave_operador_categoria	aeronave_tipo_veiculo	aeronave_fabricante	aeronave_modelo
1	87104 PSLBM	***	AVIÃO	AIRBUS S.A.S.	A321-271N

CONSULTAS SQL



```
-- Listar recomendações para uma ocorrência específica.  
SELECT tabela_recomendacao_seguranca.*  
FROM tabela_recomendacao_seguranca  
INNER JOIN tabela_ocorrencia 1..n<->1: ON tabela_recomendacao_seguranca.codigo_ocorrencia4 = tabela_ocorrencia.codigo_ocorrencia4  
WHERE tabela_ocorrencia.codigo_ocorrencia = 84857;
```

CONSULTAS SQL



RESULTADO:

A screenshot of a database management system interface showing the results of a query. The title bar says "Output" and "acidentes_aeronaves....ecomendacao_seguranca". The interface includes various buttons for navigating through the results and a toolbar with icons like CSV export and search.

Não deu nenhum resultado na tabela porque nas duas tables não tem um codigo_ocorrencia igual, fazendo aparecer isso no output.

CONSULTAS SQL

```
-- Inserir uma aeronave e uma ocorrência (É possível?) sim, é possível e inclusive deu muito trabalho,  
-- basta inserir primeiro a ocorrência e depois a aeronave, sim, eu tentei o contrário primeiro e  
-- obviamente deu errado :( mas depois funcionou :D  
INSERT INTO tabela_aeronave (  
    codigo_ocorrencia2,  
    aeronave_matricula,  
    aeronave_operador_categoria,  
    aeronave_tipo_veiculo,  
    aeronave_fabricante,  
    aeronave_modelo,  
    aeronave_tipo_icao,  
    aeronave_motor_tipo,  
    aeronave_motor_quantidade,  
    aeronave_pmd,  
    aeronave_pmd_categoria,  
    aeronave_assentos,  
    aeronave_ano_fabricacao,  
    aeronave_pais_fabricante,  
    aeronave_pais_registro,  
    aeronave_registro_categoria,  
    aeronave_registro_segmento,  
    aeronave_voo_origem,  
    aeronave_voo_destino,  
    aeronave_fase_operacao,  
    aeronave_tipo_operacao,  
    aeronave_nivel_dano,  
    aeronave_fatalidades_total  
) VALUES (  
    codigo_ocorrencia2 999102,  
    aeronave_matricula 'PT-ABC',
```



CONSULTAS SQL

```
aeronave_tipo_operacao,  
aeronave_nivel_dano,  
aeronave_fatalidades_total  
) VALUES (  
    codigo_ocorrencia2 999102,  
    aeronave_matricula 'PT-ABC',  
    aeronave_operador_categoria 'REGULAR',  
    aeronave_tipo_veiculo 'AVIÃO',  
    aeronave_fabricante 'EMBRAER',  
    aeronave_modelo 'EMB-712',  
    aeronave_tipo_icao 'E712',  
    aeronave_motor_tipo 'PISTÃO',  
    aeronave_motor_quantidade '1',  
    aeronave_pmd 1100,  
    aeronave_pmd_categoria 1,  
    aeronave_assentos '4',  
    aeronave_ano_fabricacao '1998',  
    aeronave_pais_fabricante 'BRASIL',  
    aeronave_pais_registro 'BRASIL',  
    aeronave_registro_categoria 'PRIVADO',  
    aeronave_registro_segmento 'TÁXI AÉREO',  
    aeronave_voo_origem 'SBBR',  
    aeronave_voo_destino 'SBRJ',  
    aeronave_fase_operacao 'DECOLAGEM',  
    aeronave_tipo_operacao 'PARTICULAR',  
    aeronave_nivel_dano 'SUBSTANCIAL',  
    aeronave_fatalidades_total 0  
);
```



CONSULTAS SQL



RESULTADO:

A screenshot of a database query results window. The title bar says "Output" and "acidentes_aeronaves.tabela_aeronave". The window has a dark theme with light-colored text. At the top, there are various icons for navigating between tables and performing operations like DDL and CSV export. Below the icons, there are dropdown menus for each column header, such as "codigo_ocorrencia2", "aeronave_matricula", "aeronave_operador_categoria", "aeronave_tipo_veiculo", "aeronave_fabricante", and "aeronave". The main data area shows a single row of data with the following values:

1	999102	PT-ABC	REGULAR	AVIÃO	EMBRAER	EMB-712
---	--------	--------	---------	-------	---------	---------

CONSULTAS SQL

```
INSERT INTO tabela_ocorrencia (
    codigo_ocorrencia,
    codigo_ocorrencia1,
    codigo_ocorrencia2,
    codigo_ocorrencia3,
    codigo_ocorrencia4,
    ocorrencia_classificacao,
    ocorrencia_latitude,
    ocorrencia_longitude,
    ocorrencia_cidade,
    ocorrencia_uf,
    ocorrencia_pais,
    ocorrencia_aerodromo,
    ocorrencia_dia,
    ocorrencia_hora,
    investigacao_aeronave_liberada,
    investigacao_status,
    divulgacao_relatorio_numero,
    divulgacao_relatorio_publicado,
    divulgacao_dia_publicacao,
    total_recomendacoes,
    total_aeronaves_envolvidas,
    ocorrencia_saida_pista
) VALUES (
    codigo_ocorrencia 999001,
    codigo_ocorrencia1 999101,
    codigo_ocorrencia2 999102,
    codigo_ocorrencia3 999103,
    codigo_ocorrencia4 999104,
```



CONSULTAS SQL

```
) VALUES (
    codigo_ocorrencia 999001,
    codigo_ocorrencia1 999101,
    codigo_ocorrencia2 999102,
    codigo_ocorrencia3 999103,
    codigo_ocorrencia4 999104,
    ocorrencia_classificacao 'ACIDENTE',
    ocorrencia_latitude -15.793889,
    ocorrencia_longitude -47.882778,
    ocorrencia_cidade 'BRASÍLIA',
    ocorrencia_uf 'DF',
    ocorrencia_pais 'BRASIL',
    ocorrencia_aerodromo 'SBBR',
    ocorrencia_dia '2021-08-12',
    ocorrencia_hora '14:32:00',
    investigacao_aeronave_liberada 'SIM',
    investigacao_status 'CONCLUÍDA',
    divulgacao_relatorio_numero 'REL-2021-999',
    divulgacao_relatorio_publicado 'SIM',
    divulgacao_dia_publicacao '2021-12-01',
    total_recomendacoes 2,
    total_aeronaves_envolvidas 1,
    ocorrencia_saida_pista 'NAO'
);
```



CONSULTAS SQL



RESULTADO:

A screenshot of a database management system's output window. The title bar says "Output" and "acidentes_aeronaves.tabela_ocorrencia". The window contains a table with six columns: "codigo_ocorrencia", "codigo_ocorrencia1", "codigo_ocorrencia2", "codigo_ocorrencia3", "codigo_ocorrencia4", and "ocorrencia_classificacao". The data row shows values: 999001, 999101, 999102, 999103, 999104, and ACIDENTE.

codigo_ocorrencia	codigo_ocorrencia1	codigo_ocorrencia2	codigo_ocorrencia3	codigo_ocorrencia4	ocorrencia_classificacao
999001	999101	999102	999103	999104	ACIDENTE

CONSULTAS SQL



```
-- PASSO 5
-- Ocorrências mais graves e total de aeronaves envolvidas (usei SELF JOIN)
SELECT A.ocorrencia_classificacao, B.total_aeronaves_envolvidas
FROM tabela_ocorrencia AS A, tabela_ocorrencia AS B
WHERE A.ocorrencia_classificacao = "INCIDENTE GRAVE";
```

CONSULTAS SQL

RESULTADO:



	ocorrencia_classificacao	total_aeronaves_envolvidas
1	INCIDENTE GRAVE	1
2	INCIDENTE GRAVE	1
3	INCIDENTE GRAVE	1
4	INCIDENTE GRAVE	1
5	INCIDENTE GRAVE	1
6	INCIDENTE GRAVE	1
7	INCIDENTE GRAVE	1
8	INCIDENTE GRAVE	1
9	INCIDENTE GRAVE	1
10	INCIDENTE GRAVE	1
11	INCIDENTE GRAVE	1
12	INCIDENTE GRAVE	1
13	INCIDENTE GRAVE	1
14	INCIDENTE GRAVE	1
15	INCIDENTE GRAVE	1
16	INCIDENTE GRAVE	1

CONSULTAS SQL



```
-- Ocorrências mais graves e matrículas de aeronaves envolvidas
SELECT tabela_aeronave.aeronave_matricula, tabela_ocorrencia.ocorrencia_classificacao
FROM tabela_aeronave
INNER JOIN tabela_ocorrencia 1..n<->1: ON tabela_ocorrencia.codigo_ocorrencia2 = tabela_aeronave.codigo_ocorrencia2
WHERE tabela_ocorrencia.ocorrencia_classificacao = "INCIDENTE GRAVE";
```

CONSULTAS SQL

RESULTADO:

aeronave_matricula	ocorrencia_classificacao
PPZZK	INCIDENTE GRAVE
PSGPP	INCIDENTE GRAVE
PROHY	INCIDENTE GRAVE
PPGNO	INCIDENTE GRAVE
PTCAN	INCIDENTE GRAVE
PPFKW	INCIDENTE GRAVE
PRZND	INCIDENTE GRAVE
PRKLK	INCIDENTE GRAVE
PTYEL	INCIDENTE GRAVE
PTZVV	INCIDENTE GRAVE
PTVVM	INCIDENTE GRAVE
PTIQW	INCIDENTE GRAVE
PPGGV	INCIDENTE GRAVE
PPAIJ	INCIDENTE GRAVE
PRMYM	INCIDENTE GRAVE
PUFAI	INCIDENTE GRAVE
PRZLB	INCIDENTE GRAVE
PPFUL	INCIDENTE GRAVE



CONSULTAS SQL



```
-- Total de ocorrências por estado  
  
SELECT COUNT(*) AS Ocorrencias  
FROM tabela_ocorrecia  
WHERE ocorrecia_UF = "BA";
```

CONSULTAS SQL

RESULTADO:

aeronave_matricula	ocorrencia_classificacao
PPZZK	INCIDENTE GRAVE
PSGPP	INCIDENTE GRAVE
PROHY	INCIDENTE GRAVE
PPGNO	INCIDENTE GRAVE
PTCAN	INCIDENTE GRAVE
PPFKW	INCIDENTE GRAVE
PRZND	INCIDENTE GRAVE
PRKLK	INCIDENTE GRAVE
PTYEL	INCIDENTE GRAVE
PTZVV	INCIDENTE GRAVE
PTVVM	INCIDENTE GRAVE
PTIQW	INCIDENTE GRAVE
PPGGV	INCIDENTE GRAVE
PPAIJ	INCIDENTE GRAVE
PRMYM	INCIDENTE GRAVE
PUFAI	INCIDENTE GRAVE
PRZLB	INCIDENTE GRAVE
PPFUL	INCIDENTE GRAVE



CONSULTAS SQL



```
-- Total de ocorrências por estado  
  
SELECT ocorrencia_uf AS Estado,  
       COUNT(*) AS Total_Ocorrencias  
  
FROM tabela_ocorrencia  
  
GROUP BY ocorrencia_uf  
  
ORDER BY Total_Ocorrencias DESC;
```

CONSULTAS SQL

RESULTADO:

	Estado	Total_Ocorrencias
1	SP	133
2	RJ	84
3	DF	34
4	BA	33
5	MG	28
6	PR	24
7	SC	17
8	PE	16
9	RS	16
10	MT	14
11	GO	13
12	CE	13
13	PA	12
14	AM	12
15	RR	11
16	ES	8
17	TO	7
18	MS	7



CONSULTAS SQL



```
-- Total de ocorrências na cidade de Guanambi  
SELECT COUNT(*) AS Ocurrances  
FROM tabela_ocorrencia  
WHERE ocorrencia_cidade = "GUANAMBI";
```

CONSULTAS SQL

RESULTADO:



Ocurrences
0

Nas 500 linhas de cidades que separamos não tinha nenhuma com ocorrencia em Guanambi

CONSULTAS SQL



```
-- Total de ocorrências na cidade de Belo Horizonte  
SELECT COUNT(*) AS Ocurrences  
FROM tabela_ocorrenca  
WHERE ocorrencia_cidade = "BELO HORIZONTE";
```

CONSULTAS SQL

RESULTADO:



Ocurrences	1
1	1

CONSULTAS SQL



```
-- Partindo de qual origem que mais tem acidentes?  
SELECT aeronave_voo_origem, COUNT(aeronave_voo_origem) as FREQUENCIA  
FROM tabela_aeronave  
GROUP BY aeronave_voo_origem  
ORDER BY FREQUENCIA DESC  
LIMIT 10;
```

CONSULTAS SQL



RESULTADO:

Output Partindo de qual origem tem mais acidentes? ×

W | ⌂ | ⏴ | ⏵ | ⏷ | ⏸ | ⏹

	aeronave_voo_origem	FREQUENCIA
1	GOVERNADOR ANDRÉ FRANCO MONTORO	49
2	ANTONIO CARLOS JOBIM / GALEÃO	38
3	FORA DE AERODROMO	30
4	TANCREDO NEVES	26
5	CONGONHAS	23
6	PRESIDENTE JUSCELINO KUBITSCHEK	22
7	VIRACOPOS	22
8	DEPUTADO LUÍS EDUARDO MAGALHÃES	20
9	JACAREPAGUÁ	18
10	SANTA GENOVEVA/GOIÂNIA	14

CONSULTAS SQL



```
-- ANÁLISES INTERESSANTES
-- FABRICANTES COM MAIS OCORRENCIAS

SELECT
    aeronave_fabricante,
    COUNT(*) as total_ocorrencias,
    SUM(aeronave_fatalidades_total) as total_fatalidades
FROM tabela_aeronave
WHERE aeronave_fabricante IS NOT NULL
GROUP BY aeronave_fabricante
ORDER BY total_ocorrencias DESC
LIMIT 10;
```

CONSULTAS SQL



RESULTADO:

Output Result 47

	aeronave_fabricante	total_ocorrencias	total_fatalidades
1	BOEING	95	0
2	AIRBUS	74	0
3	EMBRAER	57	1
4	BOEING COMPANY	48	0
5	ATR - GIE AVIONS DE TRANSPORT RÉGIONAL	32	0
6	CESSNA AIRCRAFT	28	2
7	AIRBUS S.A.S.	20	0
8	PIPER AIRCRAFT	16	1
9	AGUSTA	12	0
0	SIKORSKY AIRCRAFT	11	0

CONSULTAS SQL



```
-- ANALISES INTERESSANTES
-- fase do voo em que ocorrem mais acidentes
SELECT
    aeronave_fase_operacao,
    COUNT(*) AS total_ocorrencias,
    ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM tabela_aeronave), 2) AS percentual
FROM tabela_aeronave
WHERE aeronave_fase_operacao IS NOT NULL
GROUP BY aeronave_fase_operacao
ORDER BY total_ocorrencias DESC;
```

CONSULTAS SQL

RESULTADO:

aeronave_fase_operacao	total_ocorrencias	percentual
DECOLAGEM	117	23.40
POUSO	114	22.80
CRUZEIRO	103	20.60
APROXIMAÇÃO FINAL	43	8.60
SUBIDA	42	8.40
TÁXI	36	7.20
ESTACIONAMENTO	18	3.60
DESCIDA	11	2.20
MANOBRA	9	1.80
INSPEÇÃO DE TRÂNSITO/INTERVOO	3	0.60
INDETERMINADA	2	0.40
OUTRA FASE	1	0.20
PUSHBACK	1	0.20

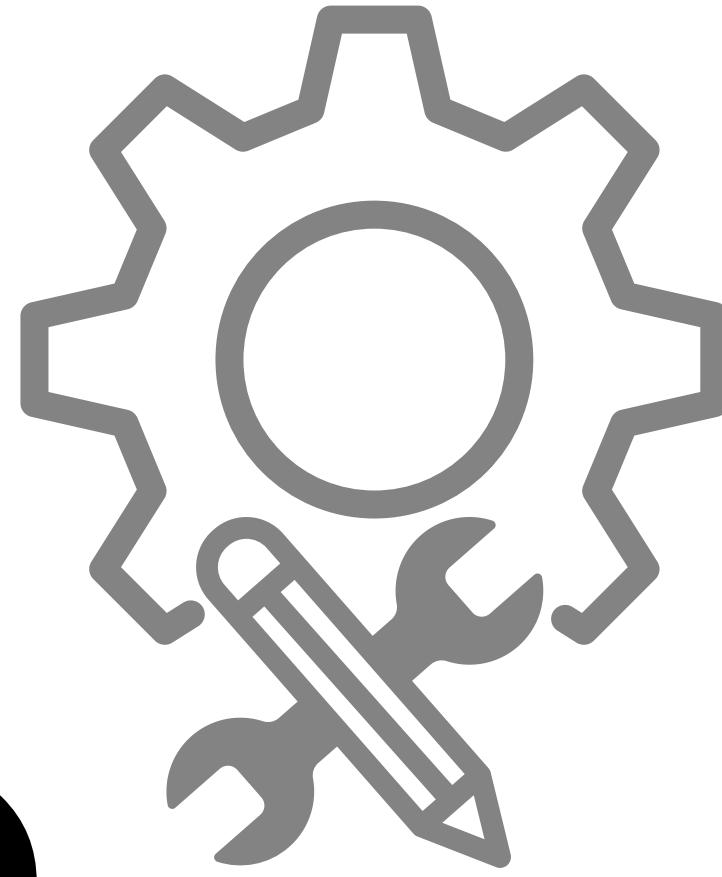
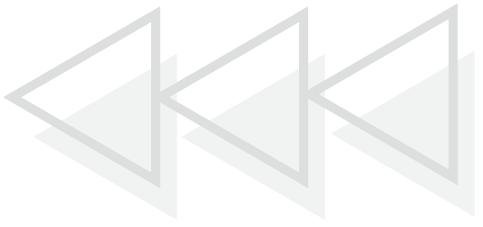
CONCLUSÃO

- Estruturamos o banco de dados definindo PKs e FKs para garantir integridade entre as tabelas.
- Realizamos consultas essenciais: listar ocorrências, aeronaves envolvidas e recomendações específicas.
- **COM AS ANÁLISES ADICIONAIS, IDENTIFICAMOS:**
 - Estados e cidades com mais ocorrências.
 - Fases do voo onde acontecem mais acidentes.
 - Fabricantes com maior número de registros.
 - Origem de voo mais frequente em incidentes.
 - O projeto mostrou como dados reais podem gerar informações relevantes quando bem organizados e consultados com SQL.
 - Consolidou o aprendizado sobre modelagem, relacionamento entre tabelas e análise de dados no MySQL.



OBRIGADO

× × × ×



×