

Banco de Dados Sprint V

Identificando as principais entidades que estão envolvidas em uma empresa de mineração.

Entidades:

"Mina",

"Mineral",

"Funcionário" e

"Produção".

Mina:

MinaID (Chave Primária)

Nome

Localização

Tipo de Minério

Mineral:

MineralID (Chave Primária)

Nome

Tipo

Grau de Pureza

Funcionário:

FuncionarioID (Chave Primária)

Nome

Cargo

Salário

Departamento

Produção:

ProducaoID (Chave Primária)

Data

Alunos: Watiley, Itamar, Victor, Fabio, Lucas e Anderson.

Quantidade Extraída

FK_MinalID (Chave Estrangeira referenciando a Mina)

FK_MineralID (Chave Estrangeira referenciando o Mineral)

FK_FuncionarioID (Chave Estrangeira referenciando o Funcionário)

Relacionamentos:

1. Uma Mina pode produzir vários Minerais (relacionamento 1 para muitos entre Mina e Mineral).
2. Um Funcionário pode estar associado a várias Produções, mas cada Produção está associada a um único Funcionário (relacionamento muitos para um entre Funcionário e Produção).
3. Uma Produção está associada a uma Mina e a um Mineral específicos (relacionamento muitos para um entre Produção e Mina/Mineral).

Terceira Forma Normal 3FN

Explicar como as tabelas estão na Terceira Forma Normal (**3FN**) geralmente envolve demonstrar que:

1. Cada tabela tem uma chave primária única:

1.1 Garanta que cada tabela tenha uma chave primária claramente identificada. Isso garante a identificação única de cada linha na tabela.

2. Eliminação de dependências transitivas:

2.1 Certifique-se de que não há dependências transitivas nos atributos. Por exemplo, se $A \rightarrow B$ e $B \rightarrow C$, então A não deve depender de C.

3. Atributos não chave dependem apenas da chave primária:

3.1 Todos os atributos não chave devem depender apenas da chave primária. Se houver atributos dependentes de outros atributos não chave, isso pode indicar uma violação da **3FN**.

Aqui está uma explicação mais detalhada aplicada a este caso:

Criação do banco de dados:

```
CREATE DATABASE BR_Minerais;
```

Tabela Mina:

A chave primária é MinalID.

Banco de Dados Sprint V

A tabela não possui dependências transitivas e os atributos não chave (Nome, Localização, Tipo de Minério) dependem diretamente da chave primária.

```
CREATE TABLE mina(  
mina_ID int AUTO_INCREMENT PRIMARY KEY,  
nome VARCHAR(50) NOT NULL,  
localizacao VARCHAR(50) NOT NULL,  
tipo_minerio VARCHAR(50) NOT NULL  
);
```

Tabela Mineral:

A chave primária é MineralID.

Não há dependências transitivas, e os atributos não chave (Nome, Tipo, Grau de Pureza) dependem diretamente da chave primária.

```
/** atualizado com a chave estrangeira (FK_Mineral_MinaID) **/
```

```
CREATE TABLE mineral(  
mineral_ID int AUTO_INCREMENT PRIMARY KEY,  
nome_mineral VARCHAR(50) NOT NULL,  
tipo VARCHAR(50) NOT NULL,  
grau_pureza VARCHAR(50) NOT NULL,  
FK_Mineral_MinaID INT NOT NULL,  
CONSTRAINT FOREIGN KEY (FK_Mineral_MinaID)  
REFERENCES mina (mina_ID)  
);
```

Tabela Funcionário:

A chave primária é FuncionarioID.

Não há dependências transitivas, e os atributos não chave (Nome, Cargo, Salário, Departamento) dependem diretamente da chave primária.

```
/** atualizado com a chave estrangeira (FK_Funcionario_MinaID) **/
```

```
CREATE TABLE funcionario(  
funcionario_ID int AUTO_INCREMENT PRIMARY KEY,  
nome_funcionario VARCHAR(50) NOT NULL,  
cargo VARCHAR(50) NOT NULL,  
salario DOUBLE NOT NULL,  
departamento VARCHAR(50) NOT NULL,  
FK_Funcionario_MinaID INT NOT NULL,  
CONSTRAINT FOREIGN KEY (FK_Funcionario_MinaID)  
REFERENCES mina (mina_ID)  
);
```

Banco de Dados Sprint V

Tabela Produção:

A chave primária é ProducaoID.

Não há dependências transitivas, e os atributos não chave (Data, **Hora** e Quantidade Extraída) dependem diretamente da chave primária.

As chaves estrangeiras (FK_MinaID, FK_MineralID, FK_FuncionarioID) referenciam diretamente as chaves primárias nas tabelas Mina, Mineral e Funcionário, respectivamente.

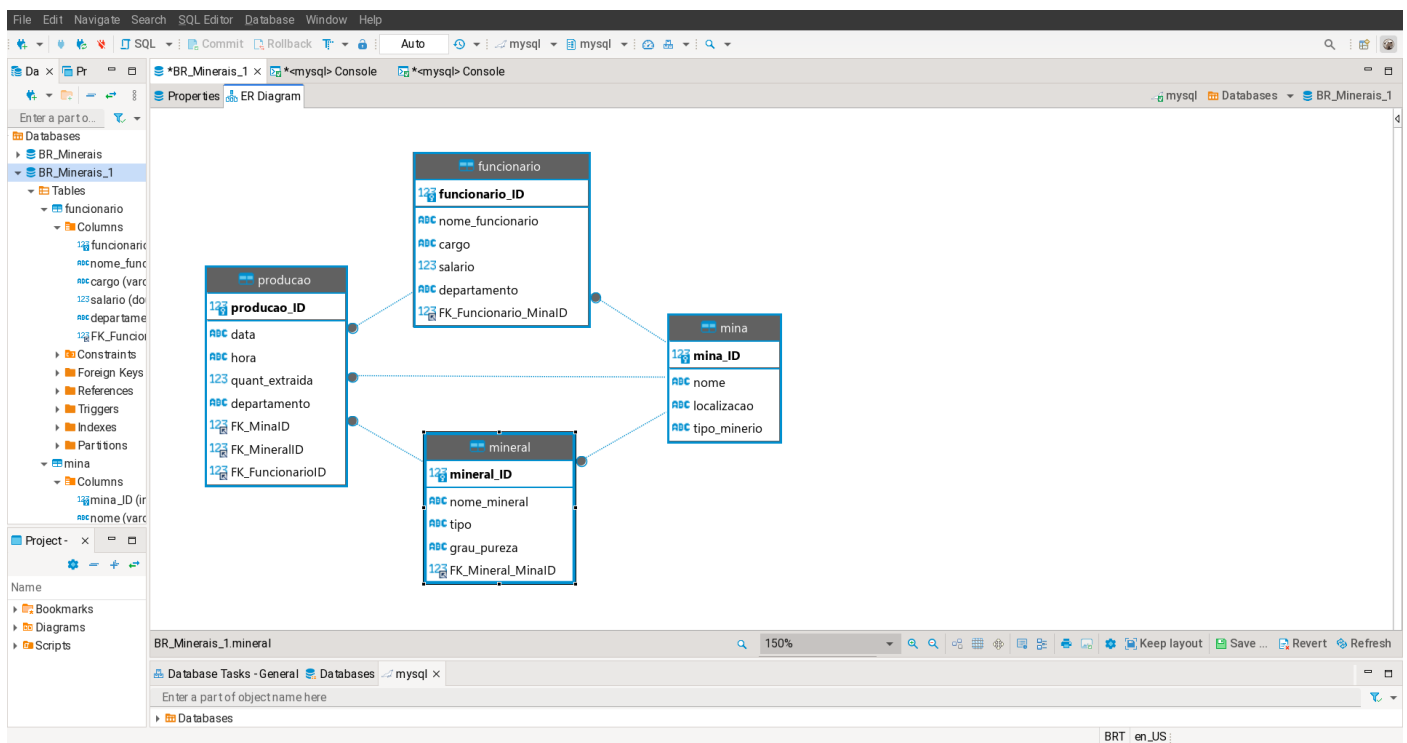
```
CREATE TABLE producao(  
    producao_ID int AUTO_INCREMENT PRIMARY KEY,  
    data VARCHAR(10) NOT NULL,  
    hora VARCHAR(8) NOT NULL,  
    quant_extraida DOUBLE NOT NULL,  
    departamento VARCHAR(50) NOT NULL,  
    FK_MinaID INT NOT NULL,  
    FK_MineralID INT NOT NULL,  
    FK_FuncionarioID INT NOT NULL,  
    CONSTRAINT FOREIGN KEY (FK_MinaID)  
    REFERENCES mina (mina_ID),  
    CONSTRAINT FOREIGN KEY (FK_MineralID)  
    REFERENCES mineral (mineral_ID),  
    CONSTRAINT FOREIGN KEY (FK_FuncionarioID)  
    REFERENCES funcionario (funcionario_ID)  
);
```

Banco de Dados Sprint V

Diagrama Entidade Relacionamento (DER):

/ atualizado com as chaves estrangeiras (tabela funcionário e tabela mineral) **/**

Diagrama:



Alunos: Watiley, Itamar, Victor, Fabio, Lucas e Anderson.

Insert:

```
select * FROM mina;
```

```
INSERT INTO mina (nome, localizacao, tipo_minerio) VALUES ('Teste A','teste local 01',  
'minerio de ferro');
```

```
INSERT INTO mina (nome, localizacao, tipo_minerio) VALUES ('Teste B','teste local 02',  
'calcopirita');
```

```
INSERT INTO mina (nome, localizacao, tipo_minerio) VALUES ('Teste C','teste local 03',  
'bauxita');
```

```
select * FROM mineral;
```

```
/** atualizado com a chave estrangeira referenciando minaID */
```

```
INSERT INTO mineral (nome_mineral, tipo, grau_pureza, FK_Mineral_MinaID) VALUES  
('ferro','2', 'media', 1);
```

```
INSERT INTO mineral (nome_mineral, tipo, grau_pureza, FK_Mineral_MinaID) VALUES  
('cobre','1', 'Alta', 2);
```

```
INSERT INTO mineral (nome_mineral, tipo, grau_pureza, FK_Mineral_MinaID) VALUES  
('Aluminio','1', 'Alta', 3);
```

Banco de Dados Sprint V

```
select * FROM funcionario;
```

```
/** atualizado com a chave estrangeira referenciando minaID */
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Alfa','minerador Chefe', 3000, 'mineracao', 1);
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Beta','minerador Chefe', 3002, 'mineracao', 2);
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Gama','minerador Chefe', 3001, 'mineracao', 3);
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Gama','minerador', 3001, 'mineracao', 2);
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Gama','minerador', 3001, 'mineracao', 1);
```

```
INSERT INTO funcionario (nome_funcionario, cargo, salario, departamento,  
FK_Funcionario_MinaID) VALUES ('Gama','minerador', 3001, 'mineracao', 3);
```

```
SELECT * FROM producao;
```

```
INSERT INTO producao (data, hora, quant_extraida, departamento, FK_MinaID,  
FK_MineralID, FK_FuncionarioID) VALUES ('01-08-2023','08:00hs', 3100, 'M3',1, 1, 1);
```

```
INSERT INTO producao (data, hora, quant_extraida, departamento, FK_MinaID,  
FK_MineralID, FK_FuncionarioID) VALUES ('02-08-2023','09:00hs', 30002, 'M2',2, 2, 2);
```

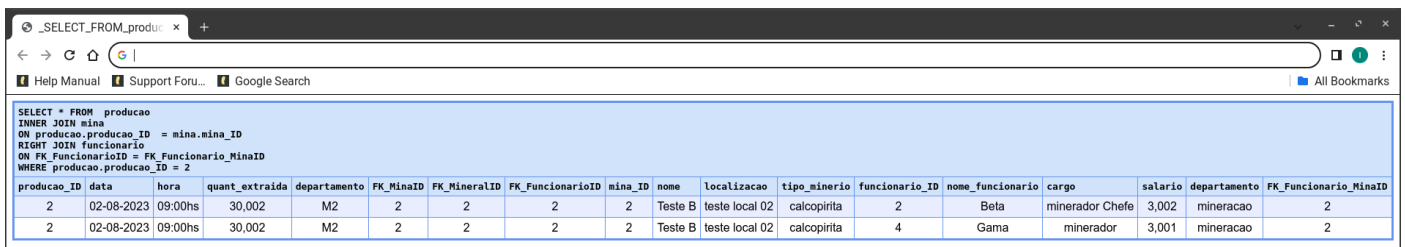
```
INSERT INTO producao (data, hora, quant_extraida, departamento, FK_MinaID,  
FK_MineralID, FK_FuncionarioID) VALUES ('03-09-2023','10:00hs', 30003, 'M3',3, 3, 3);
```


Banco de Dados Sprint V

Select com inner join e right join:

```
/** atualizado com a chave estrangeira FK_Funcionario_MinaID referenciando minaID **/  
/** Todos os dados da producaoID2, tos dos dados da minaID2 e todos os funcionarios  
pertecentes a minaID2 **/
```

```
SELECT * FROM producao  
INNER JOIN mina  
ON producao.producao_ID = mina.mina_ID  
RIGHT JOIN funcionario  
ON FK_FuncionarioID = FK_Funcionario_MinaID  
WHERE producao.producao_ID = 2;
```

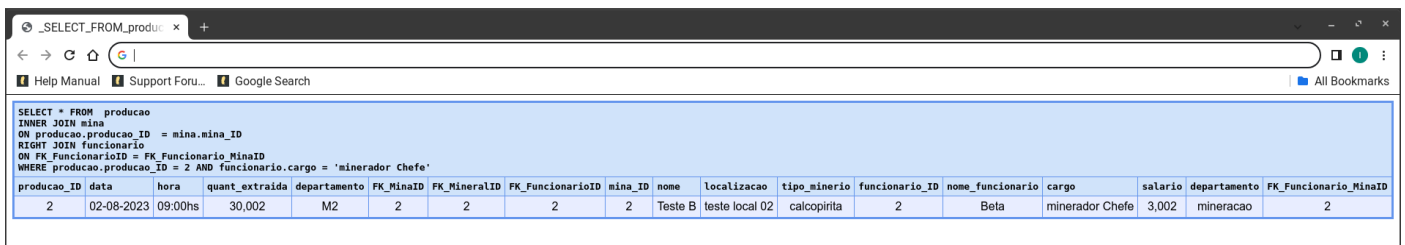


```
SELECT * FROM producao  
INNER JOIN mina  
ON producao.producao_ID = mina.mina_ID  
RIGHT JOIN funcionario  
ON FK_FuncionarioID = FK_Funcionario_MinaID  
WHERE producao.producao_ID = 2
```

producao_ID	data	hora	quant_extraida	departamento	FK_MinaID	FK_MineralID	FK_FuncionarioID	mina_ID	nome	localizacao	tipo_minerio	funcionario_ID	nome_funcionario	cargo	salario	departamento	FK_Funcionario_MinaID
2	02-08-2023	09:00hs	30,002	M2	2	2	2	2	Teste B	teste local 02	calcopirita	2	Beta	minerador Chefe	3,002	mineracao	2
2	02-08-2023	09:00hs	30,002	M2	2	2	2	2	Teste B	teste local 02	calcopirita	4	Gama	minerador	3,001	mineracao	2

```
/** atualizado com a chave estrangeira FK_Funcionario_MinaID referenciando minaID **/  
/** Todos os dados da producaoID2, tos dos dados da minaID2 e apenas o funcionario  
responsavel pertecente a minaID2 **/
```

```
SELECT * FROM producao  
INNER JOIN mina  
ON producao.producao_ID = mina.mina_ID  
RIGHT JOIN funcionario  
ON FK_FuncionarioID = FK_Funcionario_MinaID  
WHERE producao.producao_ID = 2 AND funcionario.cargo = 'minerador Chefe';
```



```
SELECT * FROM producao  
INNER JOIN mina  
ON producao.producao_ID = mina.mina_ID  
RIGHT JOIN funcionario  
ON FK_FuncionarioID = FK_Funcionario_MinaID  
WHERE producao.producao_ID = 2 AND funcionario.cargo = 'minerador Chefe'
```

producao_ID	data	hora	quant_extraida	departamento	FK_MinaID	FK_MineralID	FK_FuncionarioID	mina_ID	nome	localizacao	tipo_minerio	funcionario_ID	nome_funcionario	cargo	salario	departamento	FK_Funcionario_MinaID
2	02-08-2023	09:00hs	30,002	M2	2	2	2	2	Teste B	teste local 02	calcopirita	2	Beta	minerador Chefe	3,002	mineracao	2

Banco de Dados Sprint V