Diagrama entidade relacionamento

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Diagrama Entidade-Relacionamento (DER) - Descrição Detalhada**

O modelo é composto por três entidades centrais que se relacionam para registrar as medições dos sensores de um equipamento específico.

**Entidade: T\_EQUIPAMENTO**

Esta tabela armazena o cadastro dos equipamentos que estão sendo monitorados.

* **Propósito:** Manter um registro único para cada máquina ou ativo industrial. Isso permite agrupar e analisar os dados por equipamento.
* **Atributos:**
  + id (NUMBER(15), Chave Primária - PK): Identificador numérico único para cada equipamento.
  + marca (VARCHAR2(255)): A marca do fabricante do equipamento (ex: "Siemens", "WEG").
  + modelo (VARCHAR2(255)): O modelo específico do equipamento.
  + status (VARCHAR2(60)): O estado operacional atual do equipamento (ex: "Ativo", "Em Manutenção", "Inativo").

**Entidade: T\_SENSOR**

Esta tabela armazena o cadastro dos sensores utilizados no monitoramento.

* **Propósito:** Manter um registro de cada sensor individualmente. Isso é crucial, pois um mesmo equipamento pode ter vários sensores, e um sensor pode ser substituído ao longo do tempo.
* **Atributos:**
  + id (NUMBER(15), Chave Primária - PK): Identificador numérico único para cada sensor.
  + nome (VARCHAR2(255)): Nome ou tipo do sensor (ex: "DHT22", "MPU6050").
  + status (VARCHAR2(50)): O estado operacional do sensor (ex: "Operacional", "Com Falha", "Desativado").
  + data\_ativacao (DATE): A data em que o sensor foi instalado e começou a operar.

**Entidade: T\_LEITURA\_SENSOR**

Esta é a tabela principal (tabela fato), que armazena os dados coletados ao longo do tempo.

* **Propósito:** Registrar cada medição individual enviada pelo microcontrolador (ESP32). É a tabela que crescerá mais rapidamente e será a principal fonte de dados para análises e para o modelo de Machine Learning.
* **Atributos:**
  + id (NUMBER(15), Chave Primária - PK): Identificador numérico único para cada registro de leitura.
  + status (VARCHAR2(50)): O status reportado no momento da coleta (ex: "Normal", "Alerta de Vibração").
  + temperatura (NUMBER(6,2)): O valor da temperatura aferido, com duas casas decimais.
  + umidade (NUMBER(6,2)): O valor da umidade aferido, com duas casas decimais.
  + vibracao (NUMBER(6,2)): O valor da vibração aferido, com duas casas decimais.
  + data\_coleta (TIMESTAMP): O registro exato da data e hora da coleta.
  + T\_EQUIPAMENTO\_id (NUMBER(15), Chave Estrangeira - FK): Referencia o id da tabela T\_EQUIPAMENTO, conectando a leitura ao equipamento que a gerou.
  + T\_SENSOR\_id (NUMBER(15), Chave Estrangeira - FK): Referencia o id da tabela T\_SENSOR, conectando a leitura ao sensor que a realizou.

**Relacionamentos**

* **T\_EQUIPAMENTO e T\_LEITURA\_SENSOR:** Relacionamento de **um-para-muitos (1:N)**. Um equipamento pode ter muitas leituras de sensor ao longo do tempo, mas cada leitura pertence a um único equipamento.
* **T\_SENSOR e T\_LEITURA\_SENSOR:** Relacionamento de **um-para-muitos (1:N)**. Um sensor pode realizar muitas leituras, mas cada leitura é feita por um único sensor.

**3. Script SQL de Criação das Tabelas (CREATE TABLE)**

Abaixo está o script SQL, baseado no diagrama, para criar a estrutura do banco de dados. O código inclui a definição das tabelas, chaves primárias, chaves estrangeiras e restrições de não nulidade (NOT NULL) para garantir a integridade dos dados.

-- Tabela para armazenar os equipamentos monitorados

CREATE TABLE T\_EQUIPAMENTO (

id NUMBER(15) NOT NULL,

marca VARCHAR2(255) NOT NULL,

modelo VARCHAR2(255) NOT NULL,

status VARCHAR2(60) NOT NULL,

CONSTRAINT T\_EQUIPAMENTO\_PK PRIMARY KEY (id)

);

-- Tabela para armazenar os sensores individuais

CREATE TABLE T\_SENSOR (

id NUMBER(15) NOT NULL,

nome VARCHAR2(255) NOT NULL,

status VARCHAR2(50) NOT NULL,

data\_ativacao DATE,

CONSTRAINT T\_SENSOR\_PK PRIMARY KEY (id)

);

-- Tabela para armazenar os dados coletados pelos sensores

CREATE TABLE T\_LEITURA\_SENSOR (

id NUMBER(15) NOT NULL,

status VARCHAR2(50) NOT NULL,

temperatura NUMBER(6,2) ,

umidade NUMBER(6,2),

vibracao NUMBER(6,2),

data\_coleta TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,

T\_EQUIPAMENTO\_id NUMBER(15) NOT NULL,

T\_SENSOR\_id NUMBER(15) NOT NULL,

CONSTRAINT T\_LEITURA\_SENSOR\_PK PRIMARY KEY (id),

CONSTRAINT T\_LEITURA\_SENSOR\_T\_EQUIPAMENTO\_FK FOREIGN KEY (T\_EQUIPAMENTO\_id) REFERENCES T\_EQUIPAMENTO(id),

CONSTRAINT T\_LEITURA\_SENSOR\_T\_SENSOR\_FK FOREIGN KEY (T\_SENSOR\_id) REFERENCES T\_SENSOR(id)

);

-- Criar sequências para autoincremento dos IDs

CREATE SEQUENCE SEQ\_EQUIPAMENTO START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_SENSOR START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE SEQ\_LEITURA\_SENSOR START WITH 1 INCREMENT BY 1;

**4. Previsão de Integração Futura com Ferramentas de Visualização**

Esta modelagem de dados é ideal para integração com ferramentas de Business Intelligence (BI) e visualização de dados como **Power BI, Tableau ou Grafana**.

* **Conectividade:** As ferramentas de BI podem se conectar diretamente a este banco de dados via conectores padrão (ODBC/JDBC).
* **Análise de Séries Temporais:** O campo data\_coleta (TIMESTAMP) na tabela T\_LEITURA\_SENSOR é a chave para a criação de gráficos de linha que mostram a evolução da temperatura, umidade e vibração ao longo do tempo.
* **Dashboards:** É possível criar dashboards interativos que filtrem os dados por equipamento (T\_EQUIPAMENTO\_id), permitindo que um gerente de manutenção visualize o status de uma máquina específica em tempo real ou em um período selecionado.
* **Modelos de ML:** Os dados bem estruturados na tabela T\_LEITURA\_SENSOR servem como uma fonte de dados limpa e organizada para treinar os modelos de Machine Learning, que poderão prever falhas com base em padrões históricos de vibração e temperatura.