

1. As vantagens de usar um gerenciador de banco de dados (**DBMS**), comparado com bancos tradicionais são:

- Redução de redundância e aumento de consistência ao centralizar os dados, o que também acaba economizando em armazenamento.
- Isolamento entre aplicação e dados. Em sistemas tradicionais baseados em arquivos os dados estão fortemente acoplados com as aplicações, qualquer alteração nos dados requer alteração na aplicação.
- Dados auto descritivos pelo uso de metadados, como o tipo de dado ou restrições no dado. Um dado do tipo CPF (varchar), pode ter uma restrição que evite dados duplicados na base.
- Múltiplas visualizações respondendo a diferentes necessidades.
- Gerenciamento de usuários, aumentando o compartilhamento de dados e segurança.
- Controle de concorrência.

Uma aplicação pode acessar os dados realizando consultas (queries) ou transações (**transactions**). As propriedades de uma transação são: Atomicidade, Consistência, Isolamento e Durabilidade (**ACID**).

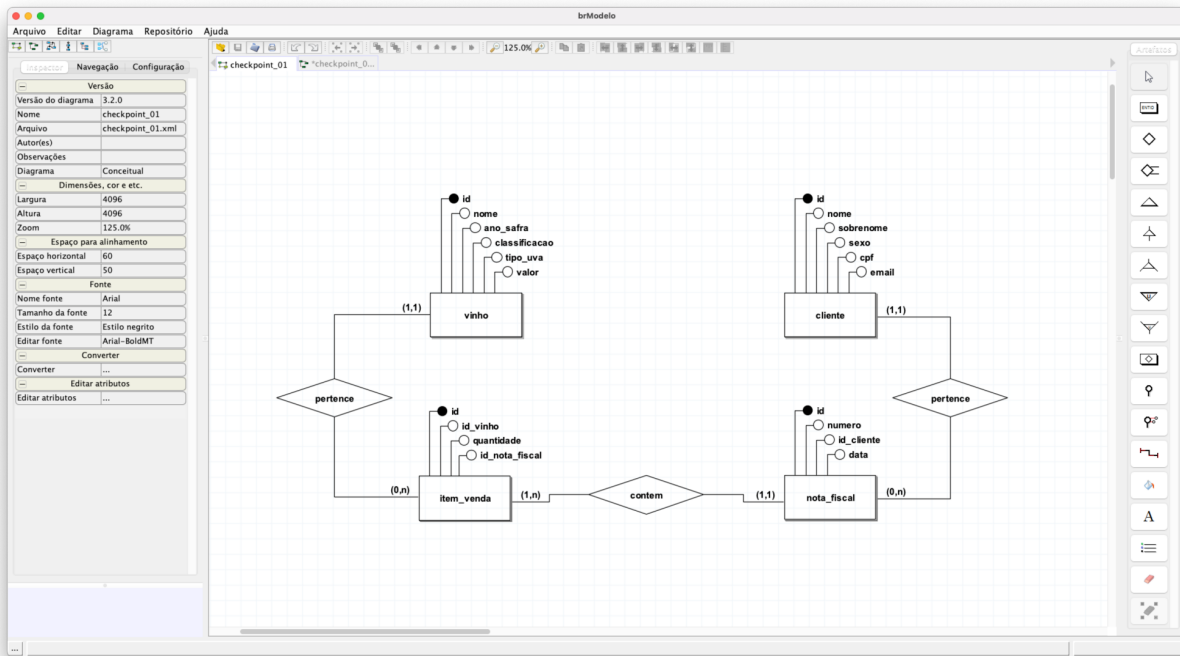
**Atomicidade:** Toda operação é executada completamente ou não é executada.

**Consistência:** A base de dados deve ser consistente antes e depois de uma transação.

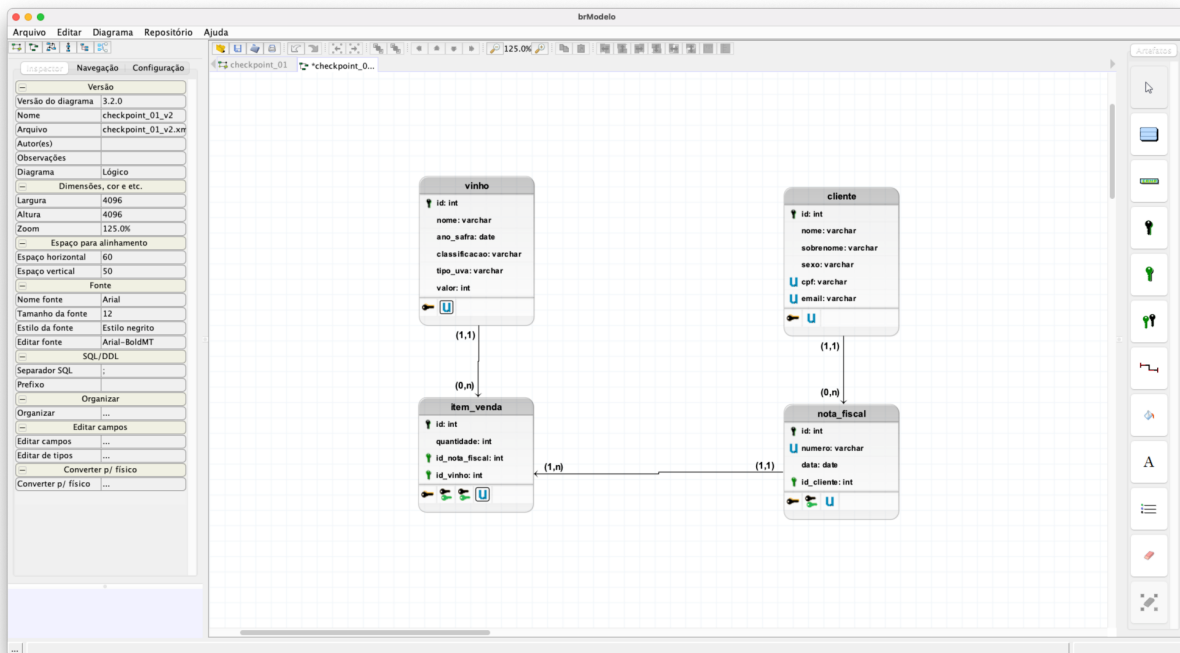
**Isolamento:** Múltiplas transações são executadas independente sem interferência uma da outra.

**Durabilidade:** Alterações de uma transação exitosa são persistentes, não tem perda de dados.

## 2. Modelo conceitual:



## 3. Modelo logico:



#### 4. Schema:

```
sqlite3 checkpoint_01.sqlite
SQLite version 3.32.3 2020-06-18 14:16:19
Enter ".help" for usage hints.
sqlite> .schema
CREATE TABLE vinho
(
  id          INT PRIMARY KEY UNIQUE,
  nome        VARCHAR NOT NULL,
  ano_safrs   DATETIME NOT NULL,
  classificacao VARCHAR NOT NULL,
  tipo_uva    VARCHAR NOT NULL,
  valor       INT      NOT NULL
);
CREATE TABLE cliente
(
  id          INT PRIMARY KEY UNIQUE,
  nome        VARCHAR NOT NULL,
  sobrenome   VARCHAR NOT NULL,
  sexo        VARCHAR NOT NULL,
  cpf         VARCHAR NOT NULL UNIQUE,
  email       VARCHAR NOT NULL UNIQUE
);
CREATE TABLE item_venda
(
  id          INT PRIMARY KEY UNIQUE,
  quantidade  INT NOT NULL,
  id_nota_fiscal INT NOT NULL,
  id_vinho    INT NOT NULL,
  FOREIGN KEY (id_nota_fiscal) REFERENCES nota_fiscal (id),
  FOREIGN KEY (id_vinho) REFERENCES vinho (id)
);
CREATE TABLE nota_fiscal
(
  id          INT PRIMARY KEY UNIQUE,
  numero      VARCHAR NOT NULL UNIQUE,
  data        DATETIME NOT NULL,
  id_cliente  INT      NOT NULL,
  FOREIGN KEY (id_cliente) REFERENCES cliente (id)
);
sqlite> █
```

```
CREATE TABLE vinho
(
  id          INT PRIMARY KEY UNIQUE,
  nome        VARCHAR NOT NULL,
  ano_safrs   DATETIME NOT NULL,
  classificacao VARCHAR NOT NULL,
  tipo_uva    VARCHAR NOT NULL,
  valor       INT      NOT NULL
);

CREATE TABLE cliente
(
  id          INT PRIMARY KEY UNIQUE,
  nome        VARCHAR NOT NULL,
  sobrenome   VARCHAR NOT NULL,
  sexo        VARCHAR NOT NULL,
  cpf         VARCHAR NOT NULL UNIQUE,
  email       VARCHAR NOT NULL UNIQUE
);

CREATE TABLE item_venda
(
  id          INT PRIMARY KEY UNIQUE,
  quantidade  INT NOT NULL,
  id_nota_fiscal INT NOT NULL,
  id_vinho    INT NOT NULL,
  FOREIGN KEY (id_nota_fiscal) REFERENCES nota_fiscal (id),
```

```
    FOREIGN KEY (id_vinho) REFERENCES vinho (id)
);

CREATE TABLE nota_fiscal
(
    id            INT PRIMARY KEY UNIQUE,
    numero        VARCHAR NOT NULL UNIQUE,
    data          DATETIME NOT NULL,
    id_cliente    INT      NOT NULL,
    FOREIGN KEY (id_cliente) REFERENCES cliente (id)
);
```