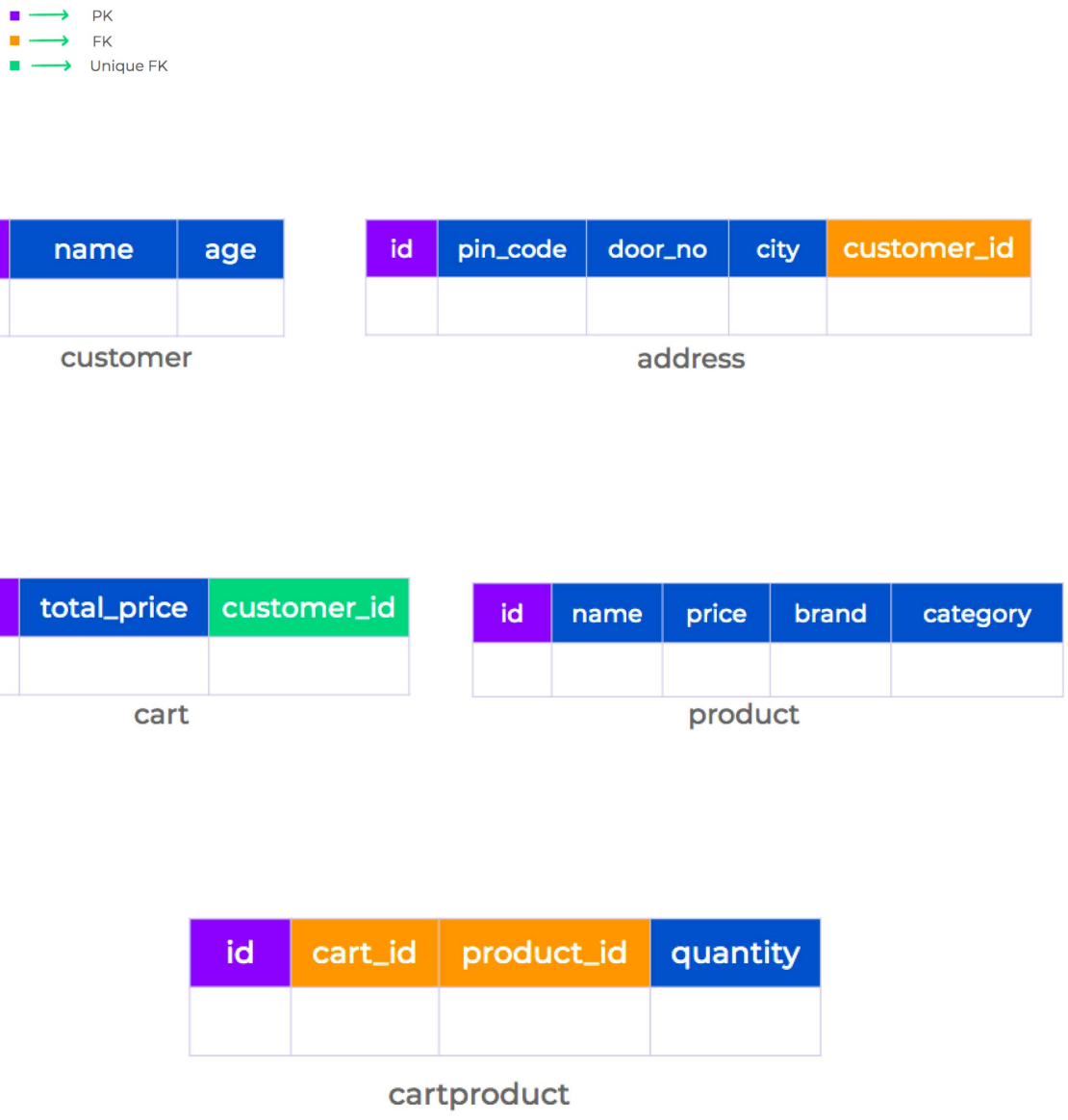
Creating a Relational Database

In the previous sessions, we've explored how to represent an ER model in the form of tables in a relational database.

Now, let's create tables to store the data in the database by defining all the columns and relationships between the tables.

Consider the **e-commerce scenario**. The tables, columns and the relations between them are s follows.



Primary Key

Following syntax creates a table with

`c1` as the primary key.

Syntax

```SQL
CREATE TABLE table_name (
    c1 t1 NOT NULL PRIMARY KEY,
...
    cn tn,
);
```

Foreign Key

In case of foreign key, we just create a foreign key constraint.

Syntax

```SQL
CREATE TABLE table2(
  c1 t1 NOT NULL PRIMARY KEY,
  c2 t2,
  FOREIGN KEY(c2) REFERENCES table1(c3) ON DELETE CASCADE
);
```

Understanding

```SQL
FOREIGN KEY(c2) REFERENCES table1(c3)
```

Above part of the foreign key constraint ensure that foreign key can only contain values that are in the referenced primary key.

```SQL
ON DELETE CASCADE
```

Ensure that if a row in

`table1` is deleted, then all its related rows in `table2` will also be deleted.

Note

To enable foreign key constraints in SQLite, use `PRAGMA foreign_keys = ON;` By default it is enabled in our platform!

Creating Tables in Relational Database

Customer Table

SQL

```sql
CREATE TABLE customer (
    id INTEGER NOT NULL PRIMARY KEY,
    name VARCHAR(250),
    age INT
);
```

Product Table

SQL

```sql
CREATE TABLE product (
  id INTEGER NOT NULL PRIMARY KEY,
  name VARCHAR(250),
  price INT,
  brand VARCHAR(250),
  category VARCHAR(250)
);
```

Address Table

SQL

```sql
CREATE TABLE address(
  id INTEGER NOT NULL PRIMARY KEY,
  pin_code INTEGER,
  door_no VARCHAR(250),
  city VARCHAR(250),
  customer_id INTEGER,
  FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE
);
```

Cart Table

SQL

```sql
CREATE TABLE cart(
  id INTEGER NOT NULL PRIMARY KEY,
  customer_id INTEGER NOT NULL UNIQUE,
  total_price INTEGER,
```

```
    FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE
);
```

Cart Product Table (Junction Table)

SQL

```sql
CREATE TABLE cart_product(
    id INTEGER NOT NULL PRIMARY KEY,
    cart_id INTEGER,
    product_id INTEGER,
    quantity INTEGER,
    FOREIGN KEY (cart_id) REFERENCES cart(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES product(id) ON DELETE CASCADE
);
```

MARKED AS COMPLETE