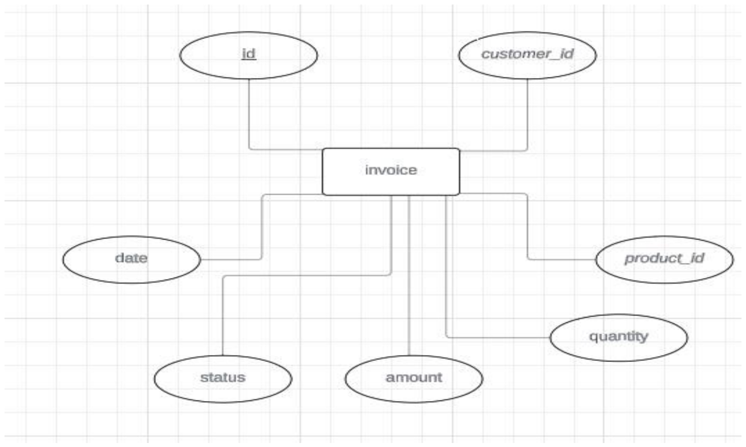
TABLE DESIGN

1. Invoice Table:

```
    id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    product_id INT,
    quantity INT,
    amount DOUBLE,
    status VARCHAR(10),
    date DATE,
    FOREIGN KEY (customer_id) REFERENCES customers(cust_id),
   FOREIGN KEY (product_id) REFERENCES products(prod_id)
```

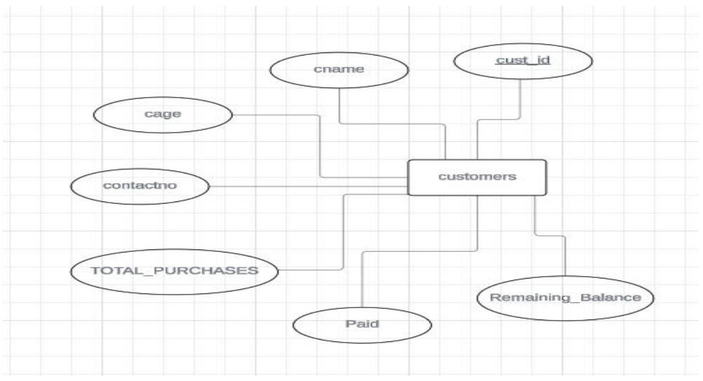
- Primary Key: `id` (Auto-incremented)
- Foreign Keys:
  - `customer_id`: References `cust_id` in the `customers` table.
  - `product_id`: References `prod_id` in the `products` table.



2. Customers Table:

```
CREATE TABLE customers (
    cust_id INT PRIMARY KEY AUTO_INCREMENT,
    cname VARCHAR(50),
    cage INT,
    contactno VARCHAR(15),
    TOTAL_PURCHASES INT DEFAULT 0,
    Paid DOUBLE DEFAULT 0,
    Remaining_Balance DOUBLE DEFAULT 0
```

- Primary Key: `cust_id` (Auto-incremented)



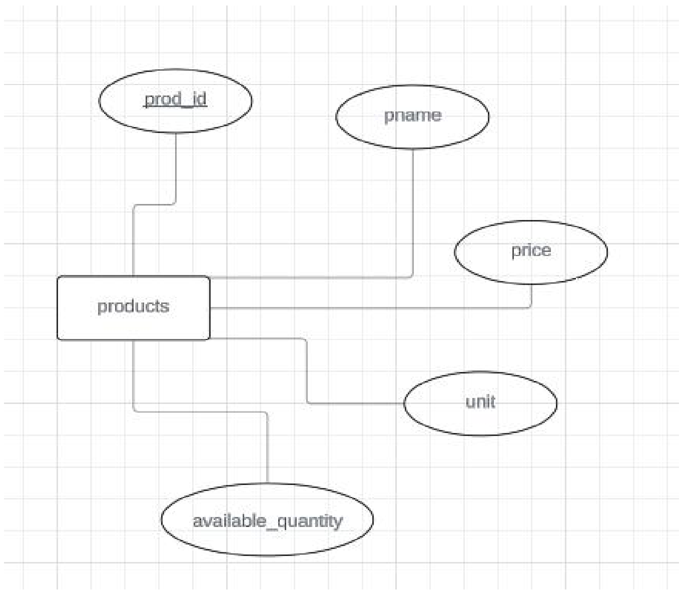
3. Products Table:

```
CREATE TABLE products (
   prod_id INT PRIMARY KEY AUTO_INCREMENT,
   pname VARCHAR(50),
   price INT,
   available_quantity INT,
   unit VARCHAR(10)
```

- Primary Key: `prod_id` (Auto-incremented)

Explanation:

- Primary Key (PK): Each table has a primary key (`cust_id` in `customers`, `prod_id` in `products`, and `id` in `Invoice`) that uniquely identifies each record in the table.
- Foreign Key (FK): The `Invoice` table includes foreign keys (`customer_id` and `product_id`) that establish relationships with the `customers` and `products` tables respectively. These foreign keys enforce referential integrity, ensuring that each invoice record is associated with existing customer and product records.



RELATIONSHIPS BETWEEN THE TABLES:

1. Customer - Invoice Relationship:

Type: One-to-Many
Description: One customer can have multiple invoices, but each invoice belongs to exactly one customer.
Foreign Key: customer_id in the Invoice table references cust_id in the Customers table.
This relationship is represented in your ER diagram by having the customer_id attribute in the Invoice table, which establishes a link to the corresponding cust_id in the Customers table.

2. Product - Invoice Relationship:

Type: One-to-Many
Description: One product can appear in multiple invoices, but each invoice pertains to exactly one product.
Foreign Key: product_id in the Invoice table references prod_id in the Products table.

3. Customer - Product Relationship:

Type: Many-to-Many
Description: A customer can purchase multiple products, and each product can be purchased by multiple customers.