DDL (for MyShop)

1. **Create Schema:**

```
CREATE SCHEMA myshop;
```

   - `CREATE SCHEMA`: Creates a new schema in the database.
   - Creates a schema named `myshop`.

2. **Create Products Table:**

```
CREATE TABLE myshop.products (
  "product_id" BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  "name" varchar(50),
  "quantity" integer DEFAULT 0
);
```

   - `CREATE TABLE`: Creates a new table.
   - `BIGINT GENERATED BY DEFAULT AS IDENTITY`: Creates an auto-incrementing column for unique identifiers.
   - `PRIMARY KEY`: Defines the primary key for the table.
   - `DEFAULT`: Sets a default value for the column.
   - Creates a `products` table with `product_id`, `name`, and `quantity`.

3. **Create Categories Table:**

```
CREATE TABLE myshop.categories (
  "category_id" BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  "name" varchar(50)
);
```

   - Creates a `categories` table with `category_id` and `name`.

4. **Create Clients Table:**

```
CREATE TABLE myshop.clients (
  "client_id" BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  "name" varchar(50)
);
```

- Creates a `clients` table with `client_id` and `name`.

5. **Create Orders Table:**

```
CREATE TABLE myshop.orders (
  "order_id" bigint PRIMARY KEY,
  "client_id" bigint,
  "order_date" date DEFAULT (now())
);
```

- **`now()`**: A function that returns the current date and time.
- Creates an `orders` table with `order_id`, `client_id`, and `order_date`.

6. **Create Product_Category Table:**

```
CREATE TABLE myshop.product_category (
  "product_id" bigint,
  "category_id" bigint,
  PRIMARY KEY ("product_id", "category_id")
);
```

- **`PRIMARY KEY ("product_id", "category_id")`**: Defines a composite primary key.
- Creates a `product_category` table to associate products with categories using `product_id` and `category_id`.

7. **Create Ordered_Products Table:**

```
CREATE TABLE myshop.ordered_products (
  "product_id" bigint,
  "order_id" bigint,
  "quantity" integer,
  PRIMARY KEY ("product_id", "order_id")
);
```

- Creates an `ordered_products` table to track ordered products using `product_id`, `order_id`, and `quantity`.

8. **Add Foreign Key to Orders Table:**

```
ALTER TABLE myshop.orders ADD FOREIGN KEY ("client_id") REFERENCES
myshop.clients ("client_id");
```

- **`ALTER TABLE`**: Modifies an existing table.
- **`ADD FOREIGN KEY`**: Adds a foreign key constraint to a column.
- Adds a foreign key constraint on `client_id` in the `orders` table referencing `client_id` in the `clients` table.

9. **Add Foreign Key to Product_Category Table (Category ID):**

```
ALTER TABLE myshop.product_category ADD FOREIGN KEY ("category_id")
REFERENCES myshop.categories ("category_id");
```

- Adds a foreign key constraint on `category_id` in the `product_category` table referencing `category_id` in the `categories` table.

10. **Add Foreign Key to Product_Category Table (Product ID):**

```
ALTER TABLE myshop.product_category ADD FOREIGN KEY ("product_id")
REFERENCES myshop.products ("product_id");
```

- Adds a foreign key constraint on `product_id` in the `product_category` table referencing `product_id` in the `products` table.

11. **Add Foreign Key to Ordered_Products Table (Order ID):**

```
ALTER TABLE myshop.ordered_products ADD FOREIGN KEY ("order_id")
REFERENCES myshop.orders ("order_id");
```

- Adds a foreign key constraint on `order_id` in the `ordered_products` table referencing `order_id` in the `orders` table.

12. **Add Foreign Key to Ordered_Products Table (Product ID):**

```
ALTER TABLE myshop.ordered_products ADD FOREIGN KEY ("product_id")
REFERENCES myshop.products ("product_id");
```

- Adds a foreign key constraint on `product_id` in the `ordered_products` table referencing `product_id` in the `products` table.