

## ۱. جدول **employees**

sql

```
CREATE TABLE employees (  
    employee_id INT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    hire_date DATE NOT NULL,  
    salary DECIMAL(10, 2) NOT NULL,  
    department_id INT,  
    FOREIGN KEY (department_id) REFERENCES departments(department_id)  
);
```

## ۲. جدول **customers**

sql

```
CREATE TABLE customers (  
    customer_id INT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    phone VARCHAR(15)  
);
```

## ۳. جدول **products**

sql

```
CREATE TABLE products (  
    product_id INT PRIMARY KEY AUTO_INCREMENT,  
    product_name VARCHAR(100) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    stock_quantity INT NOT NULL  
);
```

## ۴. جدول **sales**

sql

```
CREATE TABLE sales (  
    sale_id INT PRIMARY KEY AUTO_INCREMENT,  
    employee_id INT,  
    product_id INT,  
    sale_date DATE NOT NULL,  
    quantity INT NOT NULL,  
    total_amount DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

#### ۵. جدول departments

sql

```
CREATE TABLE departments (  
    department_id INT PRIMARY KEY AUTO_INCREMENT,  
    department_name VARCHAR(100) NOT NULL  
);
```

#### ۶. جدول orders

sql

```
CREATE TABLE orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_id INT,  
    order_date DATE NOT NULL,  
    status VARCHAR(50),  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

#### ۷. جدول order\_items

sql

```
CREATE TABLE order_items (  
    order_id INT,  
    product_id INT,  
    quantity INT,  
    price DECIMAL(10, 2)
```

```
order_item_id INT PRIMARY KEY AUTO_INCREMENT,  
order_id INT,  
product_id INT,  
quantity INT NOT NULL,  
price DECIMAL(10, 2) NOT NULL,  
FOREIGN KEY (order_id) REFERENCES orders(order_id),  
FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

#### ۸. جدول **suppliers**

sql

```
CREATE TABLE suppliers (  
    supplier_id INT PRIMARY KEY AUTO_INCREMENT,  
    supplier_name VARCHAR(100) NOT NULL,  
    contact_name VARCHAR(50),  
    phone VARCHAR(15)  
);
```

#### ۹. جدول **inventory**

sql

```
CREATE TABLE inventory (  
    inventory_id INT PRIMARY KEY AUTO_INCREMENT,  
    product_id INT,  
    supplier_id INT,  
    quantity_in_stock INT NOT NULL,  
    last_restock_date DATE,  
    FOREIGN KEY (product_id) REFERENCES products(product_id),  
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id)  
);
```

#### ۱۰. جدول **reviews**

sql

```
CREATE TABLE reviews (  
    review_id INT PRIMARY KEY AUTO_INCREMENT,  
    product_id INT,  
    customer_id INT,  
    rating INT CHECK (rating >= 1 AND rating <= 5),  
    review_text TEXT,  
    review_date DATE NOT NULL,  
    FOREIGN KEY (product_id) REFERENCES products(product_id),  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

تمرین ۱: انتخاب تمامی رکوردها

هدف: به دست آوردن تمام رکوردها از جدول employees.

- دستور: یک Query بنویسید که تمام ستون‌ها و رکوردها را از جدول employees انتخاب کند.

تمرین ۲: انتخاب ستون‌های خاص

هدف: انتخاب اطلاعات خاص از جدول.

- جزئیات: از جدول customers تنها first\_name و last\_name را انتخاب کنید.

- دستور: یک Query بنویسید که فقط نام و نام خانوادگی مشتریان را نمایش دهد.

تمرین ۳: فیلتر کردن داده‌ها

هدف: جستجوی داده‌های خاص بر اساس شرایط.

- جزئیات: از جدول employees، کارمندانی را که دستمزد آن‌ها بیش از ۵۰۰۰۰ است، انتخاب کنید.

- دستور: یک Query بنویسید که به شما اجازه دهد فقط کارمندان با دستمزد بالا را ببینید.

تمرین ۴: مرتب‌سازی داده‌ها

هدف: مرتب‌سازی داده‌ها بر اساس یک ستون خاص.

- جزئیات: همه رکوردها از جدول products را انتخاب کنید و آن‌ها را بر اساس قیمت به طور صعودی مرتب کنید.

- دستور: یک Query بنویسید که محصولات را که کمترین قیمت را دارند در ابتدا نمایش دهد.

تمرین ۵: استفاده از توابع تجمیعی

هدف: استفاده از توابع تجمیعی برای استخراج اطلاعات.

- جزئیات: از جدول sales، مجموع کل فروش‌ها را محاسبه کنید.
- دستور: یک Query بنویسید که مجموع فروش‌ها را به عنوان یک مقدار واحد برگرداند.

تمرین ۶: گروه‌بندی داده‌ها

هدف: دریافت اطلاعات به صورت تجمیعی بر اساس دسته‌بندی‌ها.

- جزئیات: تعداد کارمندان در هر بخش را از جدول employees محاسبه کنید و نتیجه را بر اساس نام بخش گروه‌بندی کنید.
- دستور: یک Query بنویسید که تعداد کارمندان هر بخش را به شما ارائه دهد.

تمرین ۷: پیوست کردن دو جدول

هدف: ترکیب داده‌ها از دو جدول مختلف.

- جزئیات: نام کارمندان و نام بخش‌های مرتبط با آن‌ها را با پیوست کردن جدول‌های employees و departments به دست آورید.

- دستور: یک Query بنویسید که این اطلاعات را به صورت یکجا نمایش دهد.

تمرین ۸: به روزرسانی داده‌ها

هدف: تغییر رکوردهای موجود در جدول.

- جزئیات: دستمزد کارمندان بخش "Sales" را ۱۰٪ افزایش دهید.
- دستور: یک Query بنویسید که تغییرات را در رکوردهای مربوطه اعمال کند.

تمرین ۹: حذف رکوردها

هدف: حذف رکوردهای خاص از جدول.

- جزئیات: تمام رکوردهای کارمندانی که در سال ۲۰۲۰ استخدام شده‌اند را از جدول employees حذف کنید.
- دستور: یک Query بنویسید که این رکوردها را بر اساس تاریخ استخدام حذف کند.

تمرین ۱۰: زیرپرسش (Subquery)

هدف: استفاده از زیرپرسش برای جستجوی داده‌ها.

- جزئیات: نام‌ها و نام خانوادگی کارمندانی را که دستمزد آن‌ها بالاتر از متوسط دستمزد کل کارمندان است انتخاب کنید.
- دستور: یک Query بنویسید که این اطلاعات را به شما ارائه دهد با استفاده از یک زیرپرسش.