

# 상품 테이블: 상품에 대한 정보를 저장

```
CREATE TABLE product (  
  id int NOT NULL AUTO_INCREMENT, # 상품 ID, 고유 식별자  
  name varchar(255) NOT NULL, # 상품 이름  
  img_path varchar(255) DEFAULT NULL, # 상품 이미지 경로  
  category varchar(255) DEFAULT NULL, # 상품 카테고리  
  price int NOT NULL, # 상품 가격  
  last_update_time datetime(6) NOT NULL, # 마지막 업데이트 시간  
  PRIMARY KEY (id)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

# 고객 테이블: 고객에 대한 정보를 저장

```
CREATE TABLE customer (  
  id bigint NOT NULL AUTO_INCREMENT, # 고객 ID, 고유 식별자  
  password varchar(128) NOT NULL, # 고객 비밀번호  
  last_login datetime(6) DEFAULT NULL, # 마지막 로그인 시간  
  is_superuser tinyint(1) NOT NULL, # 슈퍼유저 여부  
  username varchar(150) NOT NULL, # 사용자 이름  
  first_name varchar(150) NOT NULL, # 이름  
  last_name varchar(150) NOT NULL, # 성  
  email varchar(254) NOT NULL, # 이메일 주소  
  is_staff tinyint(1) NOT NULL, # 스태프 여부  
  is_active tinyint(1) NOT NULL, # 활성 상태  
  date_joined datetime(6) NOT NULL, # 가입 날짜  
  phone_number varchar(20) DEFAULT NULL, # 전화번호  
  age int DEFAULT NULL, # 나이  
  gender varchar(10) DEFAULT NULL, # 성별  
  address varchar(200) DEFAULT NULL, # 주소  
  last_update_time datetime(6) NOT NULL, # 마지막 업데이트 시간  
  name varchar(255) DEFAULT NULL, # 전체 이름  
  PRIMARY KEY (id),  
  UNIQUE KEY username (username)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf
```

# 주문 테이블: 고객의 주문에 대한 정보를 저장

```
CREATE TABLE orders (  
  id int NOT NULL AUTO_INCREMENT, # 주문 ID, 고유 식별자  
  promo_id varchar(255) DEFAULT NULL, # 프로모션 ID
```

```

order_cnt int NOT NULL, # 주문 수량
order_price int NOT NULL, # 주문 가격
order_dt varchar(255) NOT NULL, # 주문 날짜
last_update_time datetime(6) NOT NULL, # 마지막 업데이트 시간
cust_id bigint NOT NULL, # 고객 ID, 고객 테이블의 외래키
prd_id int NOT NULL, # 상품 ID, 상품 테이블의 외래키
PRIMARY KEY (id),
KEY order_prd_id_f3688dba_fk_product_prd_id (prd_id),
KEY order_cust_id_a1158f81_fk_customer_id (cust_id),
KEY ix_orders_01 (promo_id),
KEY ix_orders_02 (last_update_time),
CONSTRAINT order_cust_id_a1158f81_fk_customer_id FOREIGN KEY (cust_id) REFERENCES
customer (id),
CONSTRAINT order_prd_id_f3688dba_fk_product_prd_id FOREIGN KEY (prd_id) REFERENCES
product (id)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

```

#### # 참고할 SQL

##### 1. 고객과 그들의 주문 정보 조회

```

SELECT customer.*, orders.order_price, orders.order_dt
FROM customer
JOIN orders ON customer.id = orders.cust_id;

```

##### 2. 특정 카테고리 상품을 주문한 고객 목록

```

SELECT DISTINCT customer.id, customer.name
FROM customer
JOIN orders ON customer.id = orders.cust_id
JOIN product ON orders.prd_id = product.id
WHERE product.category = 'electronics';

```

##### 3. 주문된 모든 상품의 정보와 주문한 고객의 정보

```

SELECT product.name, product.price, customer.name AS customer_name, orders.order_dt
FROM orders
JOIN product ON orders.prd_id = product.id
JOIN customer ON orders.cust_id = customer.id;

```

##### 4. 가장 많이 팔린 상품의 이름과 판매 수량

```
SELECT product.name, SUM(orders.order_cnt) AS total_sold
FROM orders
JOIN product ON orders.prd_id = product.id
GROUP BY product.name
ORDER BY total_sold DESC
LIMIT 1;
```

5. 각 고객별 최근 주문 상품 정보

```
SELECT customer.id, customer.name, product.name AS product_name, MAX(orders.last_update_time)
AS last_order_time
FROM customer
JOIN orders ON customer.id = orders.cust_id
JOIN product ON orders.prd_id = product.id
GROUP BY customer.id, product.name;
```

6. 특정 고객의 모든 주문과 그 주문에 포함된 상품 정보

```
SELECT customer.name, orders.id AS order_id, product.name AS product_name, orders.order_price
FROM customer
JOIN orders ON customer.id = orders.cust_id
JOIN product ON orders.prd_id = product.id
WHERE customer.id = 101;
```

7. 각 상품별 총 판매액

```
SELECT product.name, SUM(orders.order_price * orders.order_cnt) AS total_revenue
FROM product
JOIN orders ON product.id = orders.prd_id
GROUP BY product.name;
```

8. 특정 기간 동안 활성 상태인 고객에 의해 주문된 모든 상품

```
SELECT product.name, orders.order_dt
FROM orders
JOIN product ON orders.prd_id = product.id
JOIN customer ON orders.cust_id = customer.id
WHERE customer.is_active = 1 AND orders.order_dt BETWEEN '2022-01-01' AND '2022-12-31';
```

9. 각 고객별로 가장 많이 주문한 상품

```
SELECT customer.name, product.name AS favorite_product, MAX(total_orders) AS max_orders
FROM (
    SELECT orders.cust_id, orders.prd_id, COUNT(*) AS total_orders
```

```
FROM orders
GROUP BY orders.cust_id, orders.prn_id
) AS order_counts
JOIN customer ON order_counts.cust_id = customer.id
JOIN product ON order_counts.prn_id = product.id
GROUP BY customer.name
ORDER BY max_orders DESC;
```

#### 10. 고객별 최근 1년간의 총 주문 금액

```
SELECT customer.name, SUM(orders.order_price) AS total_spent_last_year
FROM orders
JOIN customer ON orders.cust_id = customer.id
WHERE orders.order_dt >= DATE_SUB(NOW(), INTERVAL 1 YEAR)
GROUP BY customer.id;
```

#### 11. 오늘 주문 금액

```
select sum(order_price) as today_total_order_price
from orders
where order_dt = date_format(curdate(), '%Y-%m-%d')
order by last_update_time desc;
```

#### 12. 어제 주문 금액

```
select sum(order_price) as yesterday_total_order_price
from orders
where order_dt = date_format(date_sub(curdate(), interval 1 day), '%Y-%m-%d');
```