

데이터베이스(DBMS) 종류 및 특성, 테이블 상세 설계도 작성 (DBMS 종류, 특성 작성)

1. 오픈 소스:

- MariaDB 는 GPL(General Public License) 라이선스를 따르는 오픈 소스 프로젝트로, 누구나 무료로 사용, 수정, 배포할 수 있습니다.

2. MySQL 호환성:

- MariaDB 는 MySQL 과의 호환성을 유지하며, MySQL 에서 사용되는 쿼리 및 스키마 정의 등을 그대로 사용할 수 있습니다. 따라서 MariaDB 로 마이그레이션하는 데 용이합니다.

3. 성능 향상:

- MariaDB 는 성능 향상을 위해 여러 최적화와 기능을 추가하여 MySQL 보다 빠른 처리 속도를 제공합니다. 특히, 인덱스 알고리즘 및 스토리지 엔진의 개선으로 인한 성능 향상이 주목받고 있습니다.

4. 스토리지 엔진:

- MariaDB 는 여러 가지 스토리지 엔진을 지원합니다. InnoDB, Aria, TokuDB 등 다양한 스토리지 엔진을 선택하여 사용할 수 있습니다.

5. 데이터 보안:

- MariaDB 는 데이터 보안을 강화하기 위한 다양한 기능을 제공합니다. SSL/TLS 를 통한 암호화, 데이터베이스 백업 및 복구, 사용자 권한 관리 등이 있습니다.

6. 분산 데이터베이스:

- MariaDB 는 분산 데이터베이스 아키텍처를 지원하여 여러 서버 간에 데이터를 분산하고 병렬 처리를 통해 성능을 향상시킬 수 있습니다.

7. 데이터 정합성과 안정성:

- ACID(원자성, 일관성, 고립성, 지속성) 특성을 준수하여 데이터의 정합성과 안정성을 보장합니다.

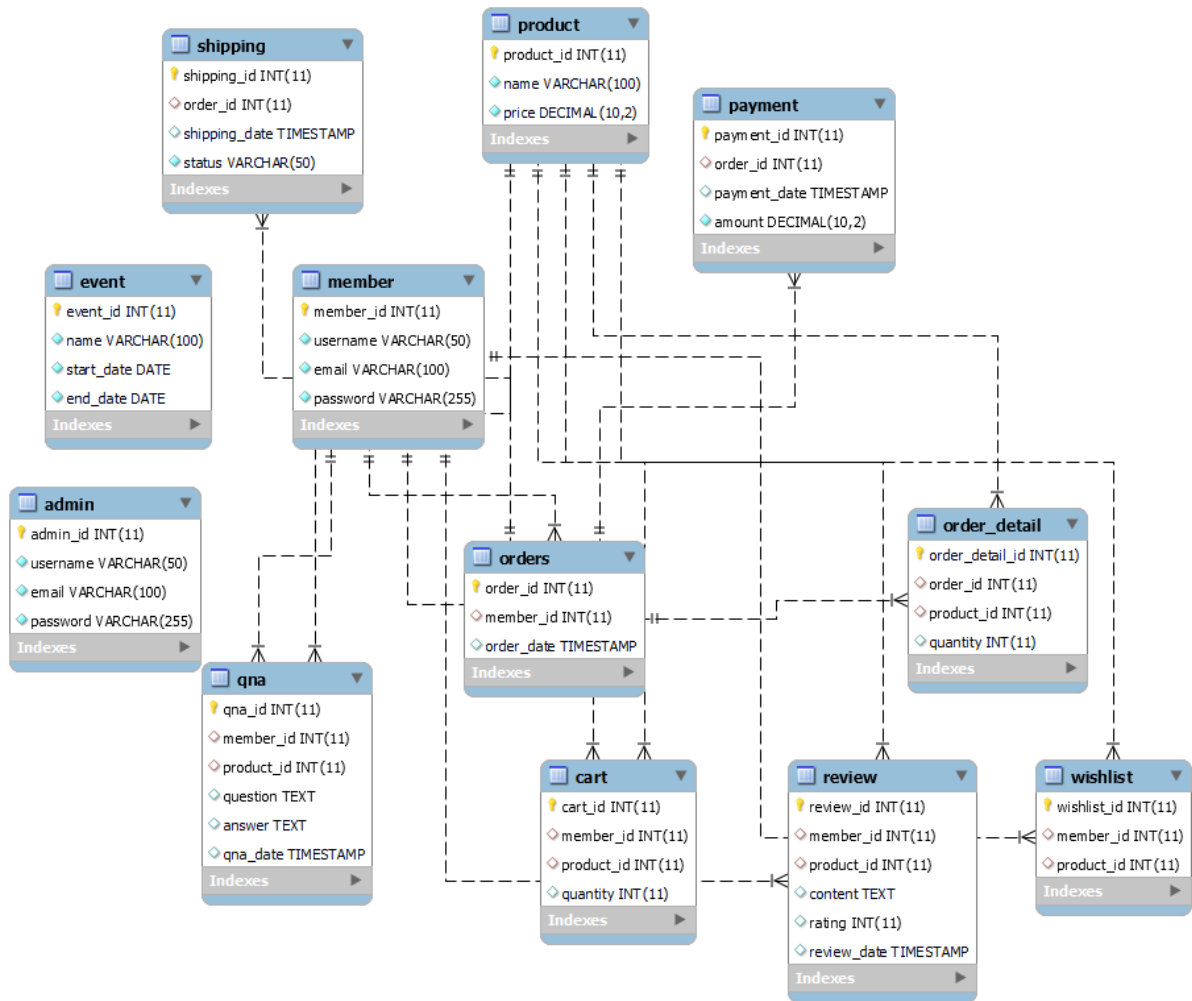
8. 확장성:

- MariaDB 는 다양한 환경 및 규모에 대응하기 위한 확장성을 제공하며, 대규모 트래픽과 데이터 처리에 적합합니다.

9. 활발한 커뮤니티:

- MariaDB 는 활발한 오픈 소스 커뮤니티를 가지고 있어 지속적인 개발과 업데이트가 이루어지고 있습니다.

테이블 상세 설명 작성



1. 회원 테이블 (member):

컬럼명	데이터 타입	속성
member_id	INT	PRIMARY KEY, AUTO_INCREMENT
username	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL
password	VARCHAR(255)	NOT NULL

2. 상품 테이블 (product):

컬럼명	데이터 타입	속성
product_id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(100)	NOT NULL
price	DECIMAL(10, 2)	NOT NULL

3. 주문 테이블 (orders):

컬럼명	데이터 타입	속성
order_id	INT	PRIMARY KEY, AUTO_INCREMENT
member_id	INT	
order_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

4. 주문 상세 테이블 (order_detail):

컬럼명	데이터 타입	속성
order_detail_id	INT	PRIMARY KEY, AUTO_INCREMENT
order_id	INT	
product_id	INT	
quantity	INT	

5. 장바구니 테이블 (cart):

컬럼명	데이터 타입	속성
cart_id	INT	PRIMARY KEY, AUTO_INCREMENT
member_id	INT	
product_id	INT	
quantity	INT	

6. 찜 테이블 (wishlist):

컬럼명	데이터 타입	속성
wishlist_id	INT	PRIMARY KEY, AUTO_INCREMENT
member_id	INT	
product_id	INT	

7. 결제 테이블 (payment):

컬럼명	데이터 타입	속성
payment_id	INT	PRIMARY KEY, AUTO_INCREMENT
order_id	INT	
payment_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
amount	DECIMAL(10, 2)	NOT NULL

8. 배송 테이블 (shipping):

컬럼명	데이터 타입	속성
shipping_id	INT	PRIMARY KEY, AUTO_INCREMENT
order_id	INT	
shipping_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
status	VARCHAR(50)	NOT NULL

9. 리뷰 테이블 (review):

컬럼명	데이터 타입	속성
review_id	INT	PRIMARY KEY, AUTO_INCREMENT
member_id	INT	
product_id	INT	
content	TEXT	
rating	INT	
review_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

10. Q&A 테이블 (qna):

컬럼명	데이터 타입	속성
qna_id	INT	PRIMARY KEY, AUTO_INCREMENT
member_id	INT	
product_id	INT	
question	TEXT	
answer	TEXT	
qna_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

11. 관리자 테이블 (admin):

컬럼명	데이터 타입	속성
admin_id	INT	PRIMARY KEY, AUTO_INCREMENT
username	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL
password	VARCHAR(255)	NOT NULL

12.이벤트 테이블 (event):

컬럼명	데이터 타입	속성
event_id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(100)	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NOT NULL

SQL 스크립트 작성하기

-- 데이터베이스 생성

```
CREATE DATABASE IF NOT EXISTS shopping_mall_db;
```

-- 사용자 생성

```
CREATE USER 'mall_user'@'localhost' IDENTIFIED BY 'password';
```

-- 사용자에게 데이터베이스 권한 부여

```
GRANT ALL PRIVILEGES ON shopping_mall_db.* TO 'mall_user'@'localhost';
```

-- 변경사항 적용

```
FLUSH PRIVILEGES;
```

-- 데이터베이스 사용

```
USE shopping_mall_db;
```

-- 테이블 생성

-- 회원 테이블

```
CREATE TABLE member (
```

```
    member_id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    username VARCHAR(50) NOT NULL,
```

```
    email VARCHAR(100) NOT NULL,
```

```
    password VARCHAR(255) NOT NULL);
```

-- 상품 테이블

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

-- 주문 테이블

```
CREATE TABLE orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    member_id INT,  
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (member_id) REFERENCES member(member_id)  
);
```

-- 주문 상세 테이블

```
CREATE TABLE order_detail (  
    order_detail_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT,  
    product_id INT,  
    quantity INT,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id));
```

-- 장바구니 테이블

```
CREATE TABLE cart (  
    cart_id INT PRIMARY KEY AUTO_INCREMENT,  
    member_id INT,  
    product_id INT,  
    quantity INT,  
    FOREIGN KEY (member_id) REFERENCES member(member_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id)  
);
```

-- 찜 테이블

```
CREATE TABLE wishlist (  
    wishlist_id INT PRIMARY KEY AUTO_INCREMENT,  
    member_id INT,  
    product_id INT,  
    FOREIGN KEY (member_id) REFERENCES member(member_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id)  
);
```


-- 결제 테이블

```
CREATE TABLE payment (  
    payment_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT,  
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    amount DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id)  
);
```

-- 배송 테이블

```
CREATE TABLE shipping (  
    shipping_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT,  
    shipping_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    status VARCHAR(50) NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(order_id)  
);
```

-- 리뷰 테이블

```
CREATE TABLE review (  
    review_id INT PRIMARY KEY AUTO_INCREMENT,  
    member_id INT,  
    product_id INT,  
    content TEXT,  
    rating INT,  
    review_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (member_id) REFERENCES member(member_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id)  
);
```

-- Q&A 테이블

```
CREATE TABLE qna (  
    qna_id INT PRIMARY KEY AUTO_INCREMENT,  
    member_id INT,  
    product_id INT,  
    question TEXT,  
    answer TEXT,  
    qna_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (member_id) REFERENCES member(member_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id));
```


-- 관리자 테이블

```
CREATE TABLE admin (  
    admin_id INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    password VARCHAR(255) NOT NULL  
);
```

-- 이벤트 테이블

```
CREATE TABLE event (  
    event_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    start_date DATE NOT NULL,  
    end_date DATE NOT NULL  
);
```