

# 최종 결과 보고서

고객 및 채팅 관리 프로그램

오스팀임플란트

C++ 기반 영상처리 전문가 양성과정

안다미로

# CONTENTS

---

## 개요

- 프로젝트 개요
- 개발 환경
- 프로그램 요약

## 프로젝트 내용

- 3차 프로젝트  
ORACLE을 이용한 DB설계

## 결론

- 보완 사항
- 발전 방향
- 시연

# 1

## 개요

1-1. 프로젝트 개요

1-2. 개발 환경

1-3. 프로그램 요약

# 1-1. 프로젝트 개요

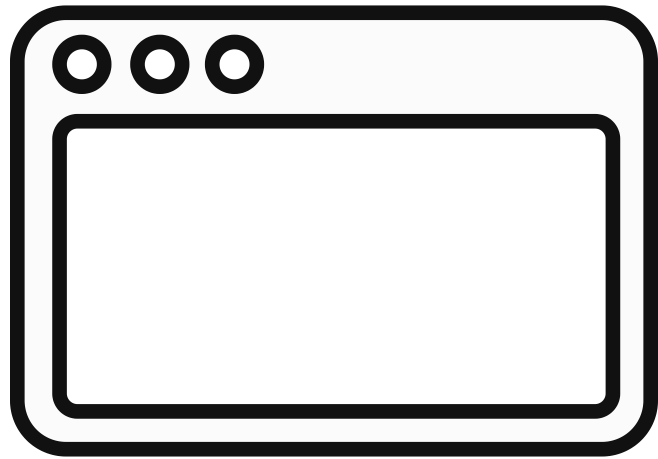
프로젝트	고객 관리 및 채팅 프로그램의 제작	
목적	고객, 상품, 주문 정보를 관리하고, 고객과 채팅할 수 있는 프로그램을 제작	
개발 기간	1차	2022년 09월 01일 ~ 2022년 09월 07일 (7일)
	2차	2022년 10월 17일 ~ 2022년 10월 30일 (14일)
	3차	2022년 11월 10일 ~ 2022년 11월 20일 (11일)
개발자	안다미로	
버전 관리	1차	<a href="https://github.com/AndaCondada/Project_CustomerManagement_CPP">https://github.com/AndaCondada/Project_CustomerManagement_CPP</a>
	2차	<a href="https://github.com/AndaCondada/Project_CustomerManagement_Qt">https://github.com/AndaCondada/Project_CustomerManagement_Qt</a>
	3차	<a href="https://github.com/AndaCondada/Project_CustomerManagementApp">https://github.com/AndaCondada/Project_CustomerManagementApp</a>

## 1-2. 개발 환경

OS	Windows 10
Language	C++
GUI Framework	Qt 6.3.1 Qt 6.3.2(Source for DB)
IDE	Qt Creator 8.0.1
DB	ORACLE XE 11g
Version	Git



## 1-3. 프로그램 요약



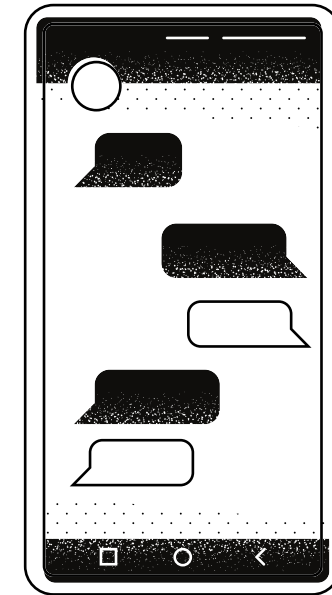
### <관리 프로그램>

- 고객, 상품, 주문 관리
- 채팅 서버
- 채팅 로그 저장



### <관리자용 채팅 프로그램>

- 서버를 통해 고객과 채팅
- 파일 전송
- 공지사항 등록



### <고객용 채팅 프로그램>

- 서버를 통해 관리자와 채팅
- 파일 전송

# 2

## 프로젝트 내용

### 2-1. 3차 프로젝트

- ORACLE DB
- DB 설계\_데이터 관계
- DB 설계\_권한 설정
- DB에 UI 적용\_MVC패턴
- 로그 저장\_생산자-소비자 패턴

## 2-1. 3차 프로젝트

### - ORACLE DB

#### 상황

- 프로그램 구조 상, 다수의 관리 매니저들이 하나의 데이터 베이스에 접근할 수 있음
- 등록되는 고객, 상품, 주문이 많아지면, 유기적으로 연결된 대규모 데이터베이스를 처리해야 하는 수요가 발생할 수 있음
- 임상데이터 등 사내에서 발생하는 데이터는 개인적으로 민감한 정보인 경우가 대다수
- 보안 관리가 철저하게 이루어져야 함



## ORACLE<sup>®</sup>

---

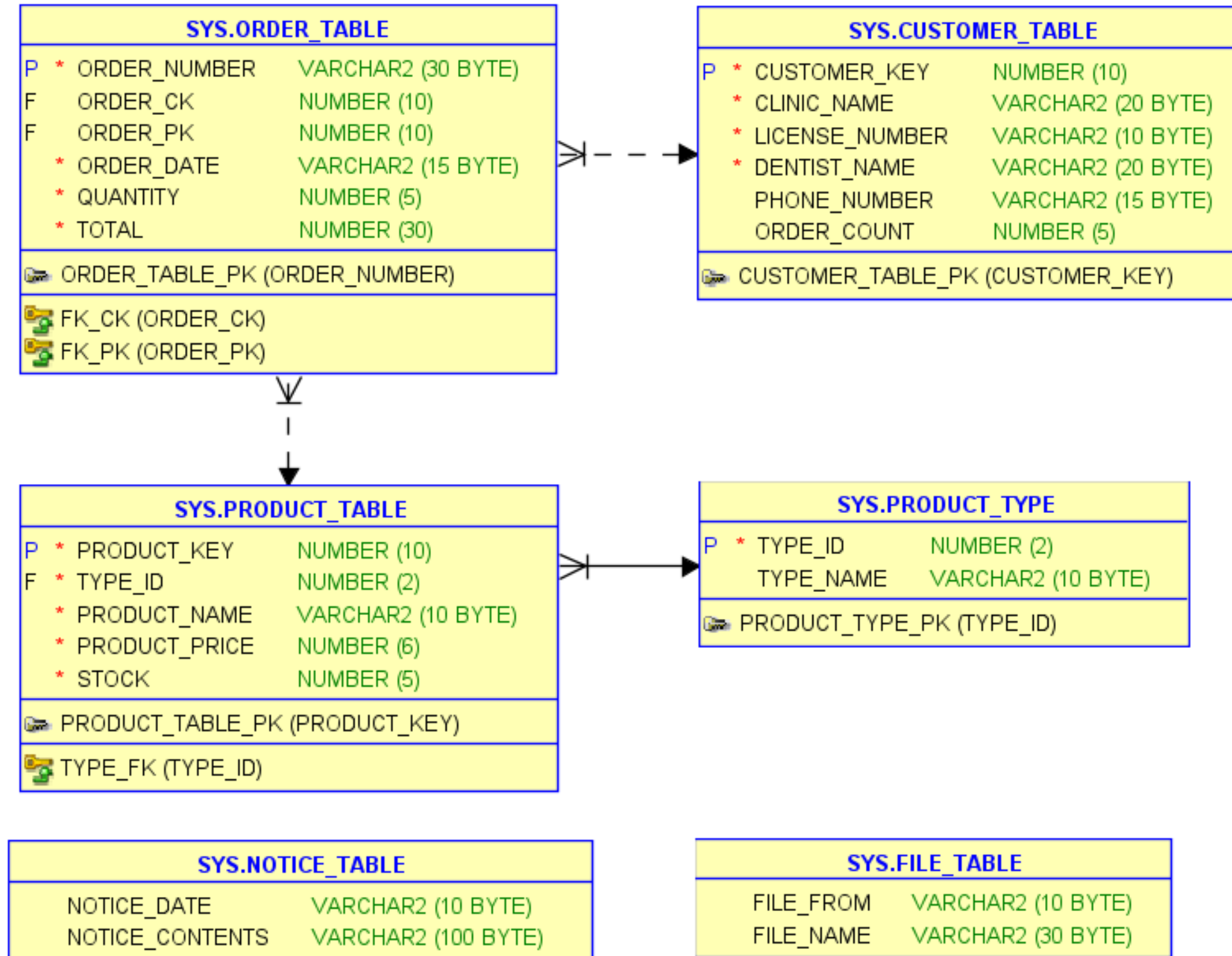
## DATABASE

- 다수의 사용자 접근 가능
- 타 데이터 베이스보다 고성능 트랜잭션 수행
- 대용량 데이터베이스 지원
- 분산처리 시스템 지원
- 높은 보안성



## 2-1. 3차 프로젝트

### - DB 설계\_데이터 관계

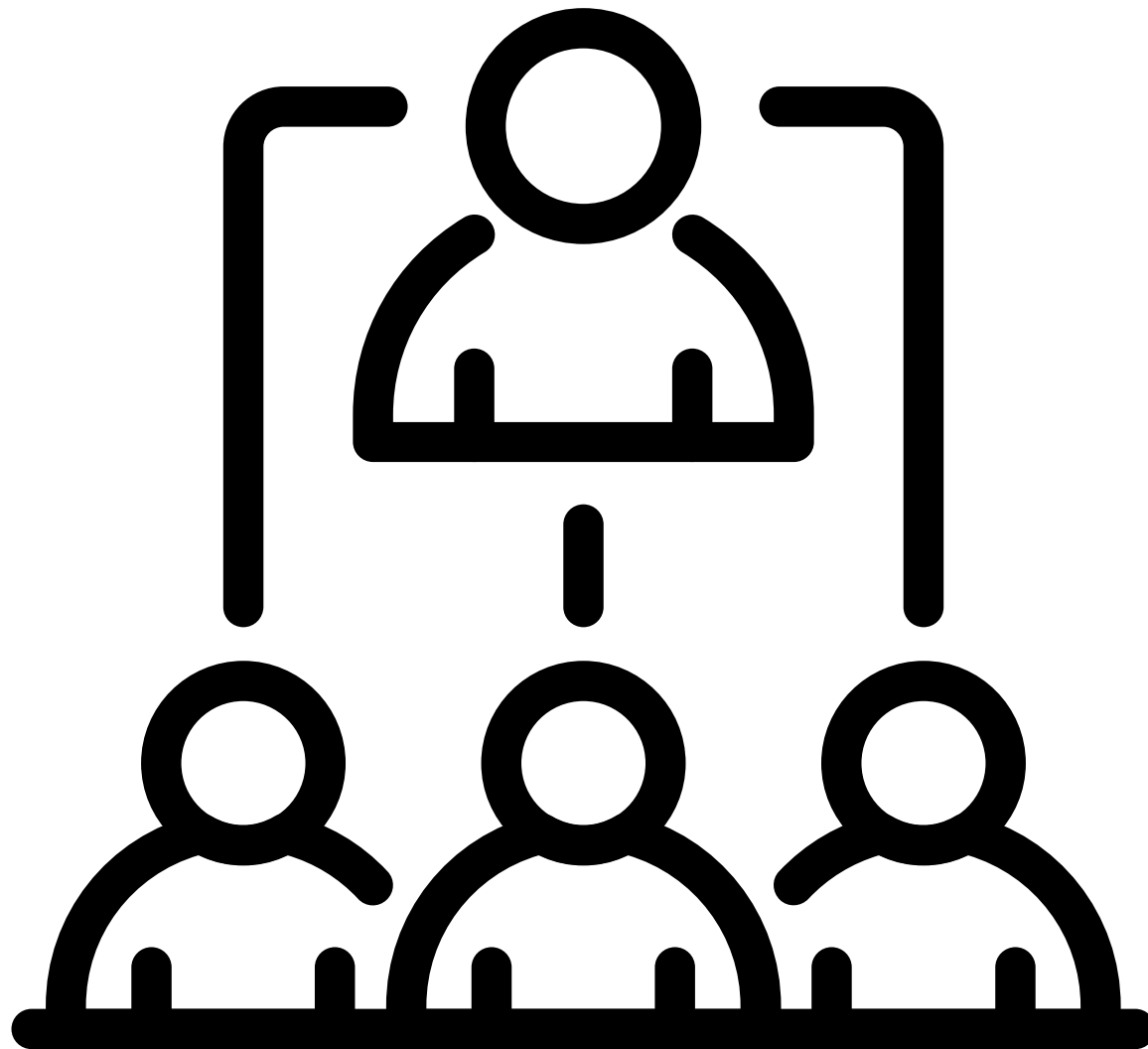


ERD

- 기존 CSV형식의 파일로 저장하던 정보를 DB로 전환
- 필요한 데이터를 RDBMS에 맞게 연관된 테이블로 설계
- 테이블마다 PK설정 및 필요한 경우에 FK를 적절하게 설정하여, 유기적인 데이터 베이스 구조 설계

## 2-1. 3차 프로젝트

### - DB 설계\_권한 설정



DBA

- DataBase를 직접 관리하는 것은 DBA
- 테이블 생성, 삭제, 데이터 입력, 삭제, 조회 등 모든 권한을 가지고 있음
- 이후 다른 매니저가 업무상 특정 테이블에 대한 권한이 필요한 경우, 계정 생성 및 권한 부여 가능

MANAGERS

- DBA로부터 계정과 권한을 부여 받아서, 권한에 해당하는 만큼 DB에 접근할 수 있음
- 기본적으로 특정 테이블에 대한 SELECT 권한만 부여
- 데이터를 추가하는 권한이 필요한 경우, INSERT에 대한 권한을 주는 것은 지양하고, 프로시저나 함수를 생성하여 권한을 부여

## 2-1. 3차 프로젝트

### - DB 설계\_권한 설정

```
GRANT SELECT ON customer_table TO customer_manager;
GRANT SELECT ON product_table TO product_manager;
GRANT SELECT ON product_type TO product_manager;
GRANT SELECT ON order_table TO order_manager;
GRANT SELECT ON notice_table TO chat_manager;
GRANT SELECT, INSERT ON notice_table TO chat_admin;
GRANT SELECT ON notice_table TO client;
GRANT SELECT ON customer_table TO chat_manager;
GRANT SELECT ON customer_table TO chat_admin;
GRANT SELECT, INSERT ON file_table TO chat_manager;
```

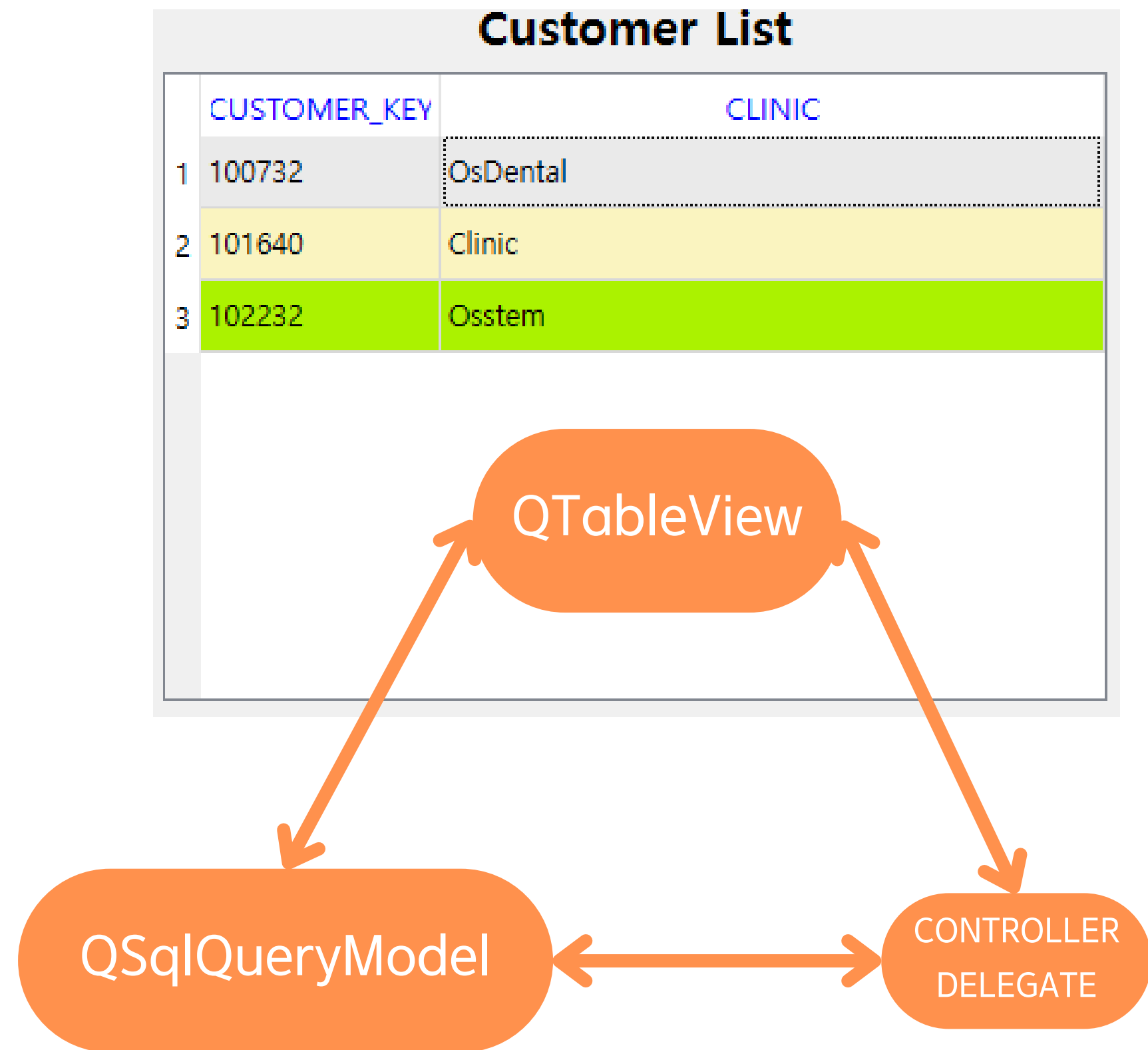
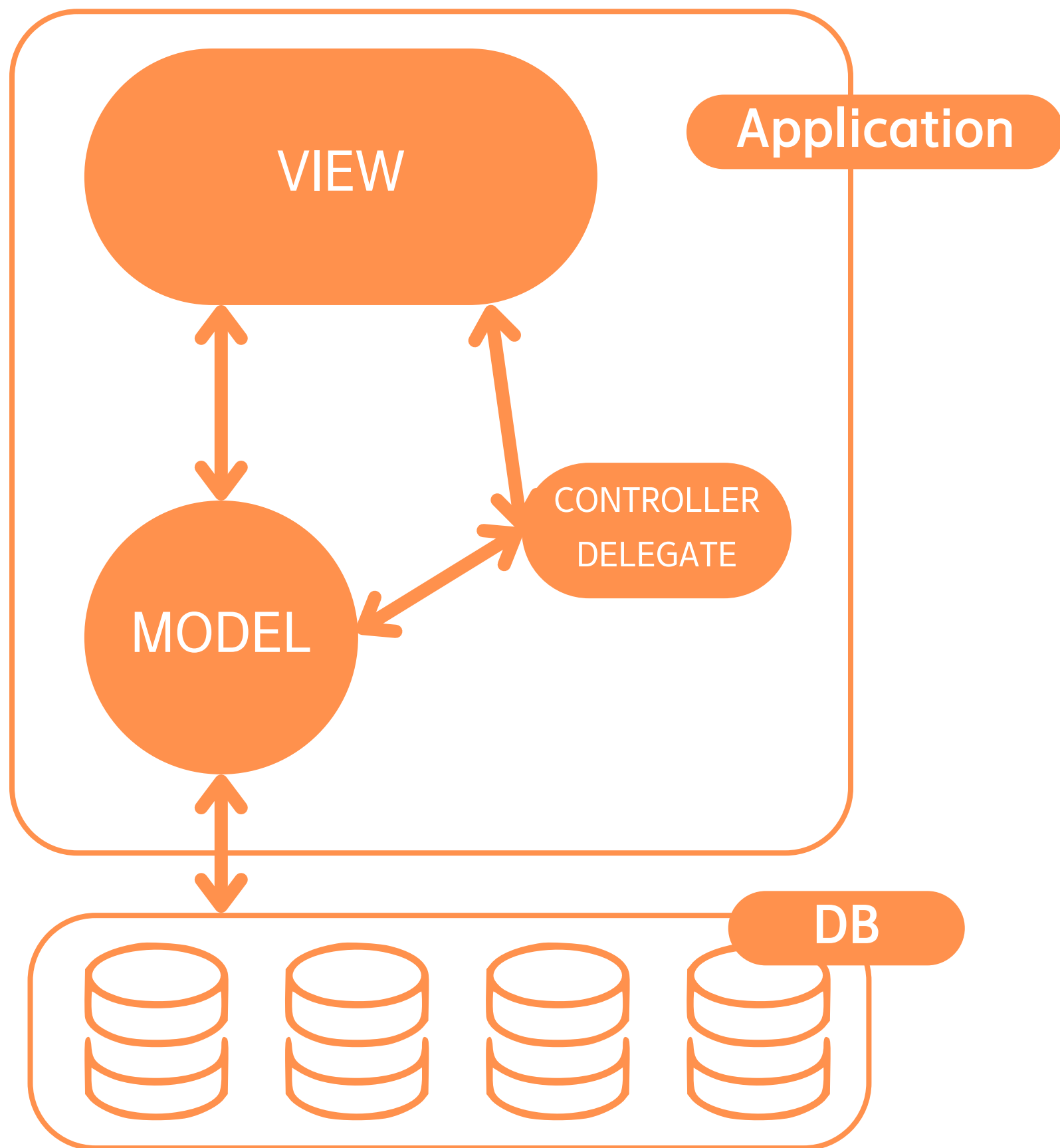
테이블 접근 권한 부여

```
GRANT EXECUTE ON INPUT_CUSTOMER TO customer_manager;
GRANT EXECUTE ON INPUT_PRODUCT TO product_manager;
GRANT EXECUTE ON INPUT_ORDER TO order_manager;
GRANT EXECUTE ON EDIT_CUSTOMER TO customer_manager;
GRANT EXECUTE ON EDIT_PRODUCT TO product_manager;
GRANT EXECUTE ON EDIT_ORDER TO order_manager;
GRANT EXECUTE ON ck_order TO order_manager;
GRANT EXECUTE ON count_customer TO order_manager;
GRANT EXECUTE ON pk_order TO order_manager;
GRANT EXECUTE ON count_product TO order_manager;
GRANT EXECUTE ON check_price TO order_manager;
GRANT EXECUTE ON order_count TO order_manager;
GRANT EXECUTE ON check_stock TO order_manager;
GRANT EXECUTE ON order_stock TO order_manager;
GRANT EXECUTE ON input_notice TO chat_manager;
```

프로시저, 함수 접근 권한 부여

## 2-1. 3차 프로젝트

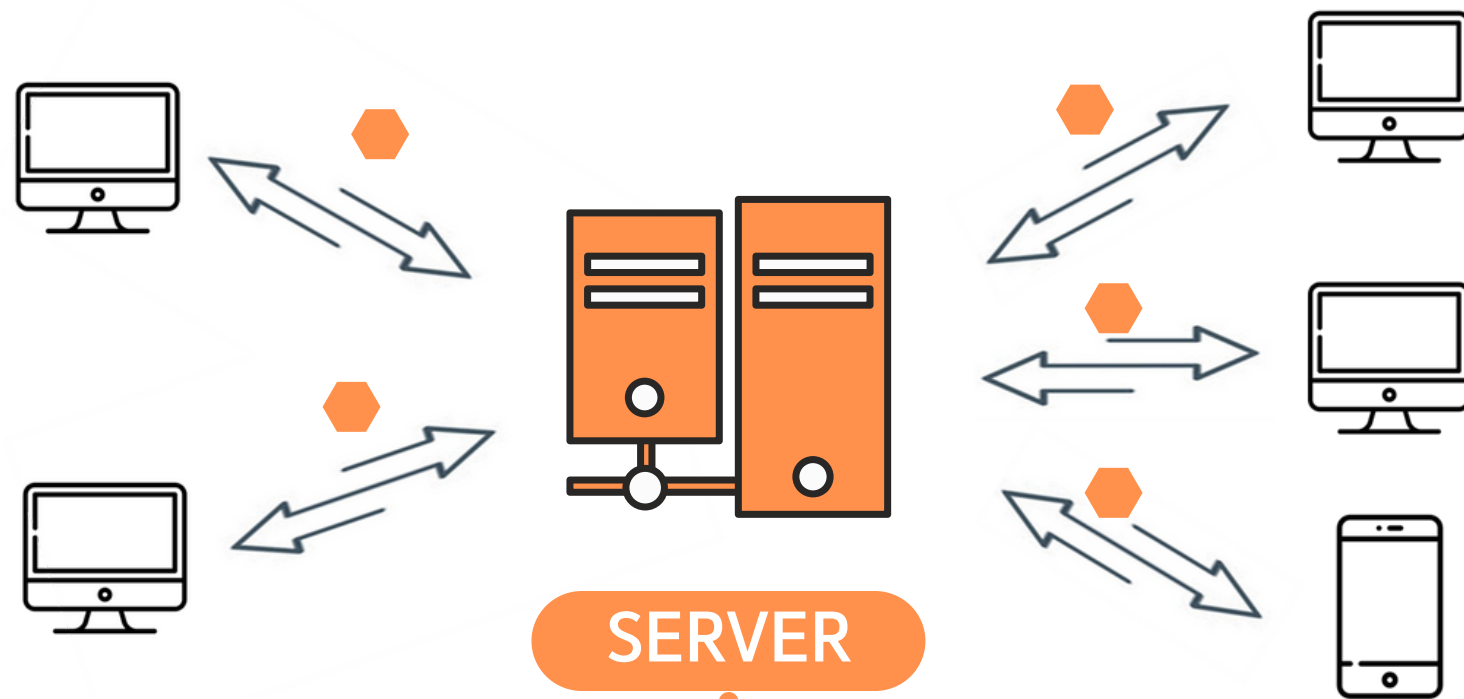
### - DB에 UI 적용\_MVC패턴



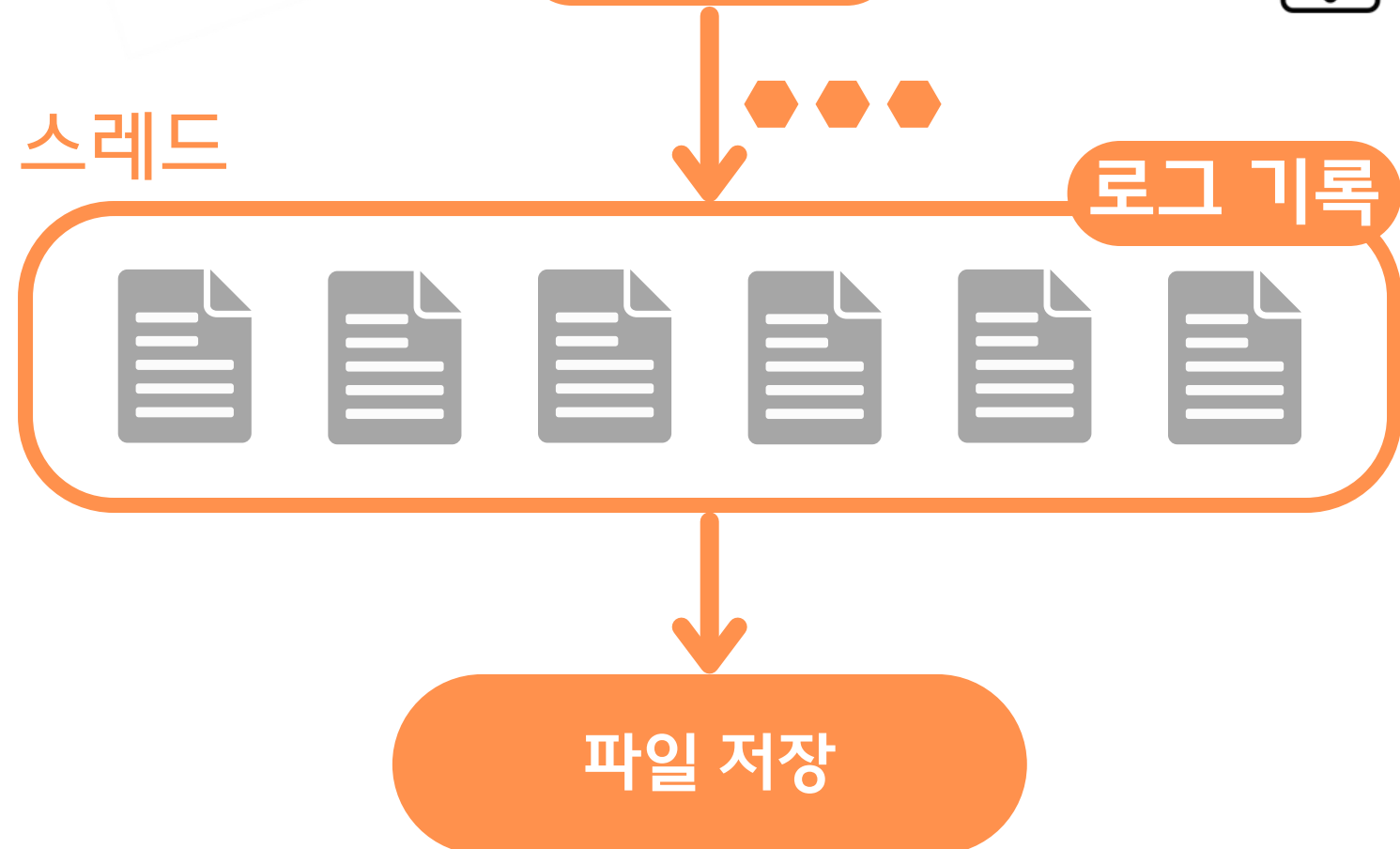
## 2-1. 3차 프로젝트

### - 로그 저장\_생산자-소비자 패턴

#### ◆ 소켓



#### 스레드



```
connect(ui->logSaveButton, SIGNAL(clicked()),  
        logSaveThread, SLOT(saveData()), Qt::QueuedConnection);
```

- 로그를 기록함에 있어서, QueuedConnection을 적용함으로써 Queue형태로 시그널을 처리
- 서버에서는 계속해서 소켓을 읽으면서, 로그 기록을 만들어가고, 사용자가 ui->logSaveButton을 누르면, 로그를 저장하는 시그널이 쌓이게 됨 (생산)
- 스레드가 돌아가면서 차례로 시그널을 처리(소비)

# 3

## 결론

3-1. 보완 사항 및 발전 방향

3-2. 시연

## 3-1. 보완 사항 및 발전 방향

### 보완 사항

- 책임 중심의 설계를 정확하게 구현하기 위해, 인터페이스 함수 등을 적절하게 활용하여 패턴을 적용할 필요가 있음
- Qt에서 ORACLE DB에 접속하기 위해 사용한 ODBC 드라이버는 배포 시 실행될 PC에도 같은 드라이버 환경이 있어야만 실행 가능한 문제 발생
- DB에서 한글 입력 시, 깨지는 문제 해결 필요

### 발전 방향

- OCI 등의 방법을 통해, 배포 시에도 추가적인 오라클 환경설정이 필요 없도록 해결 가능
- 고객 계정에 대한 DB권한을 적절하게 적용하여, 고객의 회원가입, 주문 기능을 추가할 수 있음
- 권한문제로 프로시저와 함수를 많이 사용 -> 튜닝

시연