

멀티미디어 데이터베이스

프로젝트 보고서

과목명	멀티미디어 데이터베이스
담당교수	이강만
학과	멀티미디어공학과
학년	2학년
학번	2018112552
이름	이다정
제출일	2019.12.13

목 차

1. 요구사항 수집 및 분석

2. 데이터 모델링

- 1) 개념적 데이터 모델
- 2) 논리적 데이터 모델
- 3) 물리적 데이터 모델
- 4) 관계 데이터 모델
- 5) 테이블 명세서

3. 데이터베이스 구축

- 1) 테이블 생성
- 2) 프로시저 생성
- 3) 샘플 데이터 생성

4. 운영 프로그램

- 1) 프로그램 상세 화면 및 사용 매뉴얼
- 2) 프로그램 실행 화면

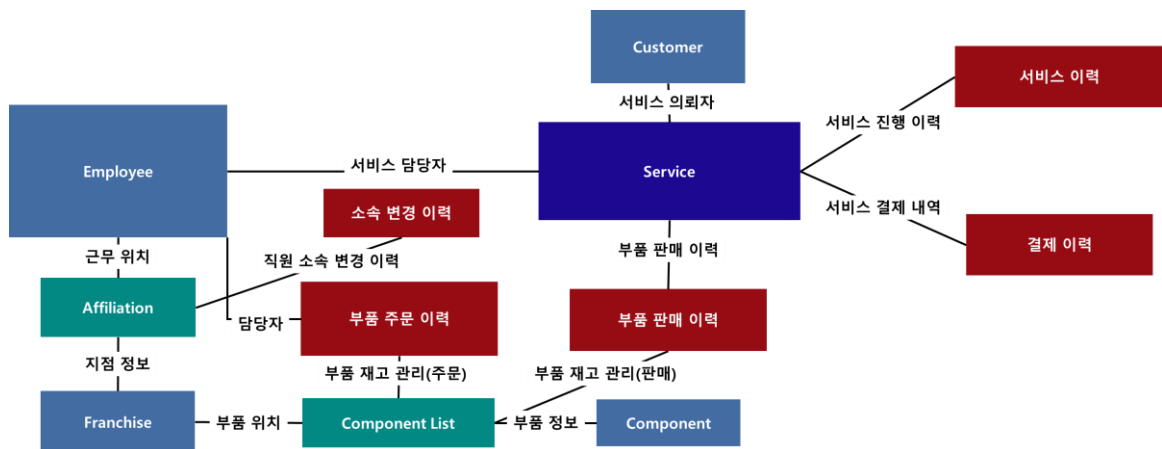
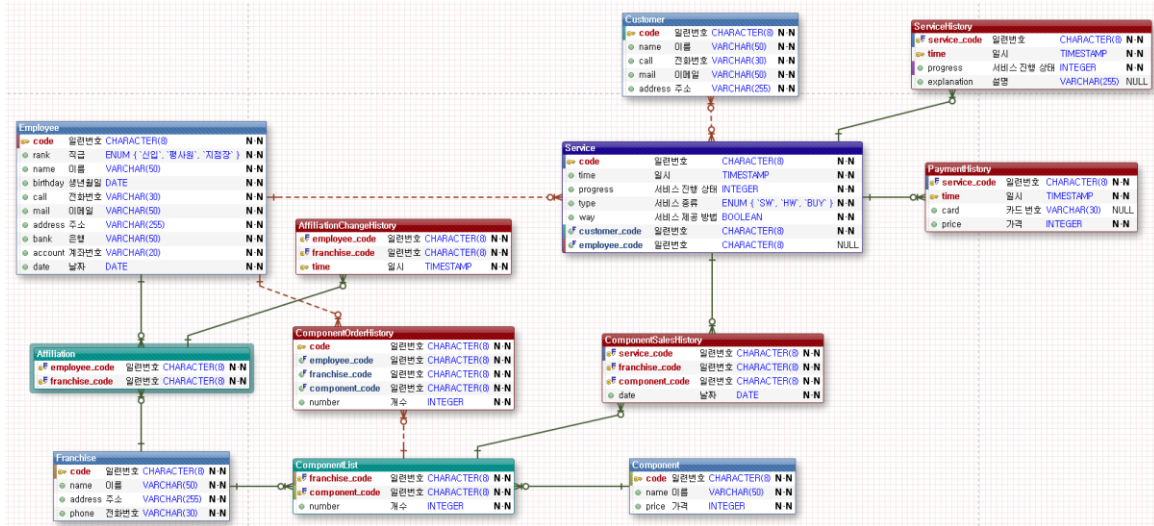
요구사항 수집 및 분석	
분류	서비스 관련
<p>1) 서비스의 종류</p> <ul style="list-style-type: none"> - 소프트웨어 점검(자사 점검 프로그램)으로 해결 가능한 건 - 부품(하드웨어) 교체로 해결 가능한 건 - 새 제품을 구매해야 하는 건 혹은 소모성 주변기기(배터리, 잉크 등) 구입 건 <p>2) 서비스 접수 방법</p> <ul style="list-style-type: none"> - 홈페이지 접수(인터넷/모바일) - 전화 접수 - 현장 접수 <p>3) 서비스 접수 시 필요한 내용</p> <ul style="list-style-type: none"> - 접수 일자 - 고객 이름 - 전화번호 - 문제 상황 <p>4) 서비스 제공 방법</p> <ul style="list-style-type: none"> - 고객이 서비스센터로 방문하는 건 - 직원이 자택에 출장 방문하는 건 <p>5) 서비스 진행 과정</p> <ul style="list-style-type: none"> - 서비스 접수 - 담당직원 매칭 - 서비스 제공 - 정산 <p>6) 서비스 비용 계산</p> <ul style="list-style-type: none"> - 하드웨어 교체 비용 - 인건비 - 출장비 <p>7) 서비스 비용 과정이 남아있어야 함</p> <p>ex)</p> <p>2019.12.05 서비스 접수 - 출장 접수</p> <p>2019.12.06 출장 방문1 - 소프트웨어 점검 진행</p> <p>2019.12.06 부품 신청 - AC001T 2개 신청</p> <p>2019.12.07 부품 도착</p> <p>2019.12.08 출장 방문2 - 부품 교체</p> <p>2019.12.08 정산 - 135,000원</p> <p>2019.12.08 서비스 완료</p>	

요구사항 수집 및 분석	
분류	직원 관련
1) 직급 <ul style="list-style-type: none"> - 신입 - 평사원 - 지점장 2) 직원 정보 <ul style="list-style-type: none"> - 이름 - 생년월일 - 이메일 - 주소 - 계좌은행 - 계좌번호 - 입사일 3) 직원 급여 <ul style="list-style-type: none"> - 연차별로 호봉 지급 	

요구사항 수집 및 분석	
분류	지점 관련
1) 모든 직원은 하나의 지점에 소속됨 2) 지점 정보 <ul style="list-style-type: none"> - 지점 명 - 전화번호 - 주소 - 근무자 - 영업 시간 	

요구사항 수집 및 분석	
분류	부품 관련
1) 모든 지점에는 일부 부품들이 구비되어 있음 2) 부족한 부품은 본사로 주문하여 받아야 함	

물리적 데이터 모델



관계 데이터 모델

Employee	<u>code</u>	rank	name	birthday	call	mail	address	bank	account	date
	AAAAAAA	지점장	김가가	1980/10/24	010-0000-0000	r@naver.com	안산	국민	217019-56-105804	1999/02/03
	BBBBBBB	지점장	이나라	1983/03/07	010-1111-1111	s@naver.com	수원	신한	079801-04-061985	2003/04/05
	CCCCCCC	지점장	박다다	1978/08/04	010-2222-2222	e@gmail.com	대구	기업	1005-802-832006	2007/07/02
	DDDDDDD	평의원	최라라	1984/02/21	010-3333-3333	f@gmail.com	부산	우리	675001-04-034789	2012/04/22
	EEEEEEE	평의원	강마마	1990/04/22	010-4444-4444	a@gmail.com	창원	국민	041301-04-058980	2016/05/28
	FFFFFFF	신입	서바바	1994/01/03	010-5555-5555	q@naver.com	일산	기업	1005-701-464885	2018/12/02

Customer	<u>code</u>	name	call	mail	address
	OOOOOOO	김김김	010-1234-1234	kkk@naver.com	서울시 용산구
	PPPPPPP	이이이	010-2345-2345	lll@naver.com	서울시 용산구
	QQQQQQQ	박박박	010-3456-3456	ppp@naver.com	서울시 종로구
	RRRRRRR	최최최	010-4567-4567	ccc@naver.com	경기도 수원시

Franchise	<u>code</u>	name	address	phone
	GGGGGGG	용산본점	서울시 용산구	02-5678-8765
	HHHHHHH	종로점	서울시 종로구	02-1234-4321
	IIIIIII	수원점	경기도 수원시	031-1357-2468

Component	<u>code</u>	name	price
	JJJJJJJ	육각회전나사	₩ 3,000
	KKKKKKK	7.5V 4080mAh 배터리	₩ 58,000
	LLLLLLL	19V 3.16A 60W 어댑터	₩ 74,000
	MMMMMMM	R380 메인보드	₩ 102,000
	NNNNNNN	RV409 그래픽 카드	₩ 208,000

ComponentList	<u>franchise_code</u>	<u>component_code</u>	number
	GGGGGGG	JJJJJJJ	10
	GGGGGGG	KKKKKKK	3
	GGGGGGG	LLLLLLL	2
	GGGGGGG	MMMMMMM	3
	GGGGGGG	NNNNNNN	2
	HHHHHHH	KKKKKKK	5
	IIIIIII	NNNNNNN	3

Affiliation	<u>employee_code</u>	<u>franchise_code</u>
	AAAAAAA	GGGGGGG
	BBBBBBB	HHHHHHH
	CCCCCCC	IIIIIII
	DDDDDDD	GGGGGGG
	EEEEEEE	HHHHHHH
	FFFFFFF	GGGGGGG

Service	<u>code</u>	<u>time</u>	<u>progress</u>	<u>type</u>	<u>way</u>	<u>customer_code</u>	<u>employee_code</u>
	SERV_001	2019-01-06 15:30:22	4	SW	True	OOOOOOOO	AAAAAAA
	SERV_002	2019-02-01 16:04:03	4	HW	True	PPPPPPPP	DDDDDDDD
	SERV_003	2019-03-08 18:33:21	4	SW	False	QQQQQQQQ	BBBBBBBB
	SERV_004	2019-05-10 20:33:21	4	BUY	False	QQQQQQQQ	EEEEEEEE
	SERV_005	2019-08-11 22:33:21	4	SW	False	OOOOOOOO	FFFFFFF
	SERV_006	2019-10-23 10:33:21	4	SW	True	RRRRRRRR	CCCCCCCC

AffiliationChangeHistory	<u>employee_code</u>	<u>franchise_code</u>	<u>time</u>
	BBBBBBBB	HHHHHHHH	2017-08-12 11:24:50
	FFFFFFF	GGGGGGGG	2018/12/02 14:51:34

ComponentOrderHistory	<u>code</u>	<u>employee_code</u>	<u>franchise_code</u>	<u>component_code</u>	<u>number</u>	<u>time</u>
	19102201	DDDDDDDD	GGGGGGGG	JJJJJJJJ	6	2019-10-22 13:57:24
	19102701	EEEEEEEE	HHHHHHHH	KKKKKKKK	5	2019-10-27 14:20:11
	19102702	CCCCCCCC	IIIIIIII	NNNNNNNN	2	2019-10-27 15:37:23

ComponentSalesHistory	<u>service_code</u>	<u>franchise_code</u>	<u>component_code</u>	<u>date</u>
	SERV_002	GGGGGGGG	KKKKKKKK	2019-02-01

ServiceHistory	<u>service_code</u>	<u>time</u>	<u>Progress</u>	<u>explanation</u>
	SERV_006	2019-10-22 11:23:57	0	인터넷 서비스 접수 완료
	SERV_006	2019-10-22 14:02:13	1	담당 직원 매칭 완료
	SERV_006	2019-10-23 09:43:27	2	고객 방문 완료
	SERV_006	2019-10-23 10:31:24	3	정산
	SERV_006	2019-10-23 10:33:21	4	서비스 제공 완료

PaymentHistory	<u>service_code</u>	<u>time</u>	<u>card</u>	<u>price</u>
	SERV_006	2019-10-23 10:31:24	0000-0000-0000-0000-0000	₩ 24,000

테이블 명세서							Page
테이블명		Component					1/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		부품 일련번호	
2	name	VARCHAR(50)	NN			부품명	
3	price	INTEGER	NN			가격	
비고							
부품의 기본 정보를 담는 테이블							

테이블 명세서							Page
테이블명		Franchise					2/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		지점 일련번호	
2	name	VARCHAR(50)	NN			지점명	
3	address	VARCHAR(255)	NN			지점 위치	
4	phone	VARCHAR(30)	NN			지점 전화번호	
비고							
지점의 정보를 담는 테이블							

테이블 명세서							Page
테이블명		Customer					3/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		고객 일련번호	
2	name	VARCHAR(50)	NN			고객 이름	
3	call	VARCHAR(30)	NN			고객 전화번호	
4	mail	VARCHAR(50)	NN			고객 이메일	
5	address	VARCHAR(255)	NN			고객 주소	
비고							
고객 기본 정보를 담는 테이블							

테이블 명세서							Page
테이블명		Employee					4/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		직원 일련번호	
2	rank	ENUM{'신입', '직원', '지점장' }	NN		'신입'	직급	
3	name	VARCHAR(50)	NN			이름	
4	birthday	DATE	NN			생년월일	
5	call	VARCHAR(30)	NN			전화번호	
6	mail	VARCHAR(50)	NN			메일	
7	address	VARCHAR(255)	NN			주소	
8	bank	VARCHAR(50)	NN			은행	
9	account	VARCHAR(20)	NN			계좌번호	
10	date	DATE	NN		NOW()	입사일	
비고							
직원의 기본 정보를 담는 테이블							

테이블 명세서							Page
테이블명		Affiliation					5/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	employee_code	CHARACTER(8)	NN	P · FK		직원 일련번호	
2	franchise_code	CHARACTER(8)	NN	P · FK		지점 일련번호	
비고							
현재 회사 내 사원의 근무지 정보를 담는 테이블							

테이블 명세서							Page
테이블명		AffiliationChangeHistory					8/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	employee_code	CHARACTER(8)	NN	P · FK		직원 일련번호	
2	franchise_code	CHARACTER(8)	NN	P · FK		지점 일련번호	
3	time	TIMESTAMP	NN	PK		변경시간	
비고							
직원의 근무지 변경 이력을 담는 테이블							

테이블 명세서							Page
테이블명		ComponentList					5/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	franchise_code	CHARACTER(8)	NN	P · FK		지점일련번호	
2	component_code	CHARACTER(8)	NN	P · FK		부품일련번호	
3	number	INTEGER	NN		0	개수	
비고							
부품 재고를 담는 리스트 어느 지점에 어떤 부품이 몇 개 있는지를 나타냄							

테이블 명세서							Page
테이블명		Service					6/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		서비스 일련번호	
2	time	TIMESTAMP	NN		NOW()	서비스 접수 시간	
3	progress	INTEGER	NN		0	서비스 진행 상태	
4	type	ENUM{ 'SW', 'HW', 'BUY' }	NN			서비스 종류	
5	way	BOOLEAN	NN			서비스 제공 방법	
6	customer_code	CHARACTER(8)	NN	FK		고객 일련번호	
7	employee_code	CHARACTER(8)	-	FK		담당직원 일련번호	
비고							
서비스 정보를 담는 테이블 서비스 진행 상황에 따라 값이 변경됨 가장 최근 정보가 담겨있음							

테이블 명세서							Page
테이블명		Component Order History					7/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	code	CHARACTER(8)	NN	PK		부품 주문 일련번호	
2	employee_code	CHARACTER(8)	NN	FK		직원 일련번호	
3	franchise_code	CHARACTER(8)	NN	FK		지점 일련번호	
4	component_code	CHARACTER(8)	NN	FK		부품 일련번호	
5	number	INTEGER	NN		1	개수	
비고							
부품 주문 내역을 담는 테이블 부품 주문은 각 지점에 소속된 직원들이 할 수 있음							

테이블 명세서							Page
테이블명		Component Sales History					8/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	service_code	CHARACTER(8)	NN	P · FK		서비스 일련번호	
2	franchise_code	CHARACTER(8)	NN	P · FK		지점 일련번호	
3	component_code	CHARACTER(8)	NN	P · FK		부품 일련번호	
4	date	DATE	NN		NOW()	판매 날짜	
비고							
부품 판매 이력을 담는 테이블 서비스 진행 단계 중 마지막 단계에서 판매한 부품이 있을 경우 직원이 작성							

테이블 명세서							Page
테이블명		Service History					9/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	service_code	CHARACTER(8)	NN	P · FK		서비스 일련번호	
2	time	TIMESTAMP	NN	PK	NOW()	일시	
3	progress	INTERGER	NN			서비스 진행 상태	
4	explanation	VARCHAR(255)	-			설명	
비고							
서비스 진행 이력을 담는 테이블 Service 테이블의 내용이 변경될 때마다 이력을 남김							

테이블 명세서							Page
테이블명		Payment History					10/10
No	Attribute	Data Type	NN	Ky	Default	Description	
1	service_code	CHARACTER(8)	NN	P · FK		서비스 일련번호	
2	time	TIMESTAMP	NN	PK	NOW()	결제 시간	
3	card	VARCHAR(30)	-			결제 카드	
4	price	INTEGER	NN			가격	
비고							
결제 이력을 담는 테이블							

테이블 생성

```
USE s2018112552;
```

```
-- 직원
```

```
CREATE TABLE Employee (
    `code`    CHARACTER(8)                NOT NULL, -- 직원코드
    `rank`    ENUM('신입', '평사원', '지점장') NOT NULL DEFAULT '신입', --
    `name`    VARCHAR(50)                  NOT NULL, -- 이름
    `birthday` DATE                       NOT NULL, -- 생년월일
    `call`    VARCHAR(30)                  NOT NULL, -- 전화번호
    `mail`    VARCHAR(50)                  NOT NULL, -- 이메일
    `address` VARCHAR(255)                 NOT NULL, -- 주소
    `bank`    VARCHAR(50)                  NOT NULL, -- 계좌은행
    `account` VARCHAR(20)                  NOT NULL, -- 계좌번호
    `date`    DATE                       NOT NULL, -- 입사날짜

    PRIMARY KEY(`code`)
);
```

```
-- 부품
```

```
CREATE TABLE Component (
    `code`    CHARACTER(8) NOT NULL, -- 부품코드
    `name`    VARCHAR(50)  NOT NULL, -- 제품명
    `price`   INTEGER      NOT NULL DEFAULT 0, -- 판매가격

    PRIMARY KEY(`code`)
);
```

```
-- 지점
```

```
CREATE TABLE Franchise (
    `code`    CHARACTER(8) NOT NULL, -- 지점코드
    `name`    VARCHAR(50)  NOT NULL, -- 지점명
    `address` VARCHAR(255) NOT NULL, -- 주소
    `phone`   VARCHAR(30)  NOT NULL, -- 전화번호

    PRIMARY KEY(`code`)
);
```

```

-- 고객
CREATE TABLE Customer (
    `code` CHARACTER(8) NOT NULL, -- 고객코드
    `name` VARCHAR(50) NOT NULL, -- 이름
    `call` VARCHAR(30) NOT NULL, -- 전화번호
    `mail` VARCHAR(50) NOT NULL, -- 이메일
    `address` VARCHAR(255) NOT NULL, -- 주소

    PRIMARY KEY(`code`)
);

-- 부품재고
CREATE TABLE ComponentList (
    `franchise_code` CHARACTER(8) NOT NULL, -- 대리점고유번호
    `component_code` CHARACTER(8) NOT NULL, -- 부품고유번호
    `number` INTEGER NOT NULL DEFAULT 0, -- 개수

    PRIMARY KEY(`franchise_code`, `component_code`),
    FOREIGN KEY(`franchise_code`) REFERENCES Franchise(`code`),
    FOREIGN KEY(`component_code`) REFERENCES Component(`code`)
);

-- 서비스 접수 정보
CREATE TABLE Service (
    `code` CHARACTER(8) NOT NULL, -- 서비스 접수 코드
    `time` TIMESTAMP NOT NULL, -- 접수 시간
    `progress` INTEGER NOT NULL DEFAULT 0, -- 서비스 진행 상태
    `type` ENUM('SW', 'HW', 'BUY') NOT NULL DEFAULT 'SW', -- 서비스 분류
    `way` BOOLEAN NOT NULL DEFAULT false, -- 서비스 제공
    방법
    `customer_code` CHARACTER(8) NOT NULL, -- 고객고유번호
    `employee_code` CHARACTER(8) NULL, -- 담당직원고유번호

    PRIMARY KEY(`code`),
    FOREIGN KEY(`customer_code`) REFERENCES Customer(`code`),
    FOREIGN KEY(`employee_code`) REFERENCES Employee(`code`)
);

```

```

-- 서비스 이력
CREATE TABLE ServiceHistory (
    `service_code` CHARACTER(8) NOT NULL, -- 서비스 접수 코드
    `time`          TIMESTAMP    NOT NULL, -- 시간
    `progress`      INTEGER      NOT NULL DEFAULT 0, -- 이력 번호
    `explanation`    VARCHAR(255) NULL,      -- 서비스 내용

    PRIMARY KEY(`service_code`, `time`),
    FOREIGN KEY(`service_code`) REFERENCES Service(`code`)
);

-- 결제 이력
CREATE TABLE PaymentHistory (
    `service_code` CHARACTER(8) NOT NULL, -- 서비스 접수 코드
    `time`          TIMESTAMP    NOT NULL, -- 결제 시간
    `card`          VARCHAR(30)  NULL,    -- 결제카드번호
    `price`         INTEGER      NOT NULL DEFAULT 0, -- 결제 금액

    PRIMARY KEY(`service_code`, `time`),
    FOREIGN KEY(`service_code`) REFERENCES Service(`code`)
);

-- 부품 주문 이력
CREATE TABLE ComponentOrderHistory (
    `code`          CHARACTER(8) NOT NULL, -- 부품 주문 코드
    `time`          TIMESTAMP    NOT NULL, -- 주문 시간
    `employee_code` CHARACTER(8) NOT NULL, -- 직원코드
    `franchise_code` CHARACTER(8) NOT NULL, -- 대리점고유번호
    `component_code` CHARACTER(8) NOT NULL, -- 부품고유번호
    `number`        INTEGER      NOT NULL DEFAULT 0, -- 개수

    PRIMARY KEY(`code`),
    FOREIGN KEY(`employee_code`) REFERENCES Employee(`code`),
    FOREIGN KEY(`franchise_code`) REFERENCES Franchise(`code`),
    FOREIGN KEY(`component_code`) REFERENCES Component(`code`)
);

```



```

-- 부품 판매 이력
CREATE TABLE ComponentSalesHistory (
    `service_code` CHARACTER(8) NOT NULL, -- 서비스 접수 코드
    `franchise_code` CHARACTER(8) NOT NULL, -- 대리점고유번호
    `component_code` CHARACTER(8) NOT NULL, -- 부품고유번호
    `date` DATE NOT NULL, -- 판매 날짜

    PRIMARY KEY(`service_code`, `franchise_code`, `component_code`),
    FOREIGN KEY(`service_code`) REFERENCES Service(`code`),
    FOREIGN KEY(`franchise_code`) REFERENCES Franchise(`code`),
    FOREIGN KEY(`component_code`) REFERENCES Component(`code`)
);

-- 직원 소속
CREATE TABLE Affiliation (
    `employee_code` CHARACTER(8) NOT NULL, -- 직원코드
    `franchise_code` CHARACTER(8) NOT NULL, -- 지점코드

    PRIMARY KEY(`employee_code`, `franchise_code`),
    FOREIGN KEY(`employee_code`) REFERENCES Employee(`code`),
    FOREIGN KEY(`franchise_code`) REFERENCES Franchise(`code`)
);

-- 소속 변경 이력
CREATE TABLE AffiliationChangeHistory (
    `employee_code` CHARACTER(8) NOT NULL, -- 직원코드
    `franchise_code` CHARACTER(8) NOT NULL, -- 지점코드
    `time` TIMESTAMP NOT NULL, -- 시간

    PRIMARY KEY(`employee_code`, `franchise_code`, `time`),
    FOREIGN KEY(`employee_code`) REFERENCES Employee(`code`),
    FOREIGN KEY(`franchise_code`) REFERENCES Franchise(`code`)
);

```

프로시저 추가

Employee Table	<pre>-- add new Employee DROP PROCEDURE IF EXISTS AddNewEmployee; DELIMITER \$\$ CREATE PROCEDURE AddNewEmployee(IN `code` CHARACTER(8), IN `name` VARCHAR(50), IN `birthday` DATE, IN `call` VARCHAR(30), IN `mail` VARCHAR(50), IN `address` VARCHAR(255), IN `bank` VARCHAR(50), IN `account` VARCHAR(20)) BEGIN INSERT INTO Employee(`code`, `name`, `birthday`, `call`, `mail`, `address`, `bank`, `account`, `date`) VALUES(`code`, `name`, `birthday`, `call`, `mail`, `address`, `bank`, `account`, CURRENT_TIMESTAMP()); END \$\$ DELIMITER ;</pre>
Franchise Table	<pre>-- add new franchise DROP PROCEDURE IF EXISTS AddNewFranchise; DELIMITER \$\$ CREATE PROCEDURE AddNewFranchise(IN `code` CHARACTER(8), IN `name` VARCHAR(50), IN `address` VARCHAR(255), IN `phone` VARCHAR(30)) BEGIN INSERT INTO Component(`code`, `name`, `address`, `phone`) VALUES(`code`, `name`, `address`, `phone`); END \$\$ DELIMITER ;</pre>

<p>Component Table</p>	<pre>-- add new Component DROP PROCEDURE IF EXISTS AddNewComponent; DELIMITER \$\$ CREATE PROCEDURE AddNewComponent(IN `code` CHARACTER(8), IN `name` VARCHAR(50), IN `price` INTEGER) BEGIN INSERT INTO Component(`code`, `name`, `price`) VALUES(`code`, `name`, `price`); END \$\$ DELIMITER ;</pre>
<p>Customer Table</p>	<pre>-- add new customer DROP PROCEDURE IF EXISTS AddNewCustomer; DELIMITER \$\$ CREATE PROCEDURE AddNewCustomer(IN `code` CHARACTER(8), IN `name` VARCHAR(50), IN `call` VARCHAR(30), IN `mail` VARCHAR(50), IN `address` VARCHAR(255)) BEGIN INSERT INTO Component(`code`, `name`, `call`, `mail`, `address`) VALUES(`code`, `name`, `call`, `mail`, `address`); END \$\$ DELIMITER ;</pre>

<p>Affiliation Table + Affiliation Change History Table</p>	<pre>-- set affiliation -- Add log -> AffiliationChangeHistory DROP PROCEDURE IF EXISTS SetAffiliation; DELIMITER \$\$ CREATE PROCEDURE SetAffiliation(IN `employee` CHARACTER(8), IN `franchise` CHARACTER(8)) BEGIN IF EXISTS (SELECT `employee_code` FROM Affiliation WHERE `employee_code` = `employee`) THEN UPDATE Affiliation SET `franchise_code` = `franchise` WHERE `employee_code` = `employee`; ELSE INSERT INTO Afiliation(`employee_code`, `franchise_code`) VALUES(`employee`, `franchise`); INSERT INTO AffiliationChangeHistory(`employee_code`, `time`, `franchise_code`) VALUES(`employee`, CURRENT_TIMESTAMP(), `franchise`); END IF; END \$\$ DELIMITER ;</pre>
---	--

<p>Component List Table + Component Order History Table</p>	<pre>-- order component in list DROP PROCEDURE IF EXISTS OrderComponent; DELIMITER \$\$ CREATE PROCEDURE OrderComponent(IN `code` CHARACTER(8), IN `employee` CHARACTER(8), IN `franchise` CHARACTER(8), IN `component` VARCHAR(50), IN NUMBER INTEGER) BEGIN DECLARE COUNT INTEGER; IF EXISTS (SELECT `number` FROM ComponentList WHERE `franchise_code` = `franchise` AND `component_code` = `component`) THEN SELECT `number` INTO COUNT FROM ComponentList WHERE `franchise_code` = `franchise` AND `component_code` = `component`; UPDATE ComponentList SET `number` = COUNT + NUMBER WHERE `franchise_code` = `franchise` AND `component_code` = `component`; ELSE INSERT INTO ComponentList(`franchise_code`, `component_code`, `number`) VALUES(`franchise`, `component`, NUMBER); END IF; INSERT INTO ComponentOrderHistory(`code`, `time`, `employee_code`, `franchise_code`, `component_code`, `number`) VALUES (`code`, CURRENT_TIMESTAMP(), `employee`, `franchise`, `component`, NUMBER); END \$\$ DELIMITER ;</pre>
---	--

<p>Component List Table + Component Sales History Table</p>	<pre>-- sold component DROP PROCEDURE IF EXISTS SellComponent; DELIMITER \$\$ CREATE PROCEDURE SellComponent(IN `code` CHARACTER(8), IN `employee` CHARACTER(8), IN `franchise` CHARACTER(8), IN `component` VARCHAR(50)) BEGIN DECLARE COUNT INTEGER; SELECT `number` INTO COUNT FROM ComponentList WHERE `franchise_code` = `franchise` AND `component_code` = `component`; IF COUNT > 1 THEN UPDATE ComponentList SET `number` = COUNT - 1 WHERE `franchise_code` = `franchise` AND `component_code` = `component`; INSERT INTO ComponentSalesHistory(`code`, `date`, `employee_code`, `franchise_code`, `component_code`) VALUES (`code`, CURRENT_TIMESTAMP(), `employee`, `franchise`, `component`); ELSEIF COUNT = 1 THEN DELETE FROM ComponentList WHERE `franchise_code` = `franchise` AND `component_code` = `component`; INSERT INTO ComponentSalesHistory(`code`, `date`, `employee_code`, `franchise_code`, `component_code`) VALUES (`code`, CURRENT_TIMESTAMP(), `employee`, `franchise`, `component`); END IF; END \$\$ DELIMITER ;</pre>
---	---

Service Table + Service History Table (ADD)	<pre> DROP PROCEDURE IF EXISTS AddService; DELIMITER \$\$ CREATE PROCEDURE AddService(IN `code` CHARACTER(8), IN `way` BOOLEAN, IN `customer` CHARACTER(8), IN `employee` CHARACTER(8), IN `explanation` VARCHAR(255), OUT `result` BOOLEAN) BEGIN DECLARE NOW DATE; SET NOW = CURRENT_TIMESTAMP(); IF `employee` = NULL THEN INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`) VALUES (`code`, NOW, 0, 'SW', `way`, `customer`); ELSE INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES (`code`, NOW, 0, 'SW', `way`, `customer`, `employee`); SET `result` = TRUE; INSERT INTO ServiceHistory(`service_code`, `time`, `progress`, `explanation`) VALUES (`code`, CURRENT_TIMESTAMP(), 0, `explanation`); END IF; END \$\$ DELIMITER ; </pre>
---	--

Service Table + Service History Table (UPDATE)	<pre> DROP PROCEDURE IF EXISTS MatchingEmployee; DELIMITER \$\$ CREATE PROCEDURE MatchingEmployee(IN `code` CHARACTER(8), IN `employee` CHARACTER(8), OUT `result` BOOLEAN) BEGIN DECLARE NOW DATE; SET NOW = CURRENT_TIMESTAMP(); UPDATE Service SET `employee_code` = `employee` WHERE `code` = `code`; INSERT INTO ServiceHistory(`service_code`, `time`, `progress`, `explanation`) VALUES (`code`, CURRENT_TIMESTAMP(), 1, `explanation`); SET `result` = TRUE; END \$\$ DELIMITER ; </pre>
Service Table + Service History Table (UPDATE) 2	<pre> DROP PROCEDURE IF EXISTS ChangeProgress; DELIMITER \$\$ CREATE PROCEDURE ChangeProgress(IN `code` CHARACTER(8), IN `progress` INTEGER) BEGIN UPDATE Service SET `progress` = `progress` WHERE `code` = `code`; INSERT INTO ServiceHistory(`service_code`, `time`, `progress`, `explanation`) VALUES (`code`, CURRENT_TIMESTAMP(), `progress`, `explanation`); END \$\$ DELIMITER ; </pre>

Service Table + Service History Table (UPDATE) 3	DROP PROCEDURE IF EXISTS ChangeServiceType; DELIMITER \$\$ CREATE PROCEDURE ChangeServiceType(IN 'code' CHARACTER(8), IN 'type' INTEGER) BEGIN UPDATE Service SET 'type' = 'type' WHERE 'code' = 'code'; END \$\$ DELIMITER ;
Service Table + Service History Table (UPDATE) 4	DROP PROCEDURE IF EXISTS PaymentService; DELIMITER \$\$ CREATE PROCEDURE PaymentService(IN 'code' CHARACTER(8), IN 'card' VARCHAR(30), IN 'price' INTEGER) BEGIN IF 'card' = NULL THEN INSERT INTO PaymentHistory('service_code', 'time', 'price') VALUES('code', CURRENT_TIMESTAMP(), 'price'); ELSE INSERT INTO PaymentHistory('service_code', 'time', 'card', 'price') VALUES('code', CURRENT_TIMESTAMP(), 'card', 'price'); END IF; CALL ChangeProgress('code', 5); INSERT INTO ServiceHistory('service_code', 'time', 'progress', 'explanation') VALUES ('code', CURRENT_TIMESTAMP(), 'progress', 'explanation'); END \$\$ DELIMITER ;

샘플 데이터 생성

Employee
Table

-- add employee

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("AAAAAAA", "지점장", "김가가", "1980-10-24",
"010-0000-0000", "r@naver.com", "안산", "국민", "217019-56-105804", "1999-02-
03");
```

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("BBBBBBBB", "지점장", "이나나", "1983-03-07",
"010-1111-1111", "s@naver.com", "수원", "신한", "079801-04-061985", "2003-04-
05");
```

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("CCCCCCCC", "지점장", "박다다", "1978-08-04",
"010-2222-2222", "e@gmail.com", "대구", "기업", "1005-802-832006", "2007-07-
02");
```

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("DDDDDDDD", "평사원", "최라라", "1984-02-21",
"010-3333-3333", "f@gmail.com", "부산", "우리", "675001-04-034789", "2012-04-
22");
```

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("EEEEEEEE", "평사원", "강마마", "1990-04-22",
"010-4444-4444", "a@gmail.com", "창원", "국민", "041301-04-058980", "2016-05-
28");
```

```
INSERT INTO Employee(`code`, `rank`, `name`, `birthday`, `call`, `mail`, `address`,
`bank`, `account`, `date`) VALUES ("FFFFFFF", "신입", "서바바", "1994-01-03", "010-
5555-5555", "g@naver.com", "일산", "기업", "1005-701-464885", "2018-12-02");
```

code	rank	name	birthday	call	mail	address	bank	account	date
AAAAAAA	지점장	김가가	1980-10-24	010-0000-0000	r@naver.com	안산	국민	217019-56-105804	1999-02-03
BBBBBBBB	지점장	이나나	1983-03-07	010-1111-1111	s@naver.com	수원	신한	079801-04-061985	2003-04-05
CCCCCCCC	지점장	박다다	1978-08-04	010-2222-2222	e@gmail.com	대구	기업	1005-802-832006	2007-07-02
DDDDDDDD	평사원	최라라	1984-02-21	010-3333-3333	f@gmail.com	부산	우리	675001-04-034789	2012-04-22
EEEEEEEE	평사원	강마마	1990-04-22	010-4444-4444	a@gmail.com	창원	국민	041301-04-058980	2016-05-28
FFFFFFF	신입	서바바	1994-01-03	010-5555-5555	g@naver.com	일산	기업	1005-701-464885	2018-12-02
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer Table	<pre>-- add customer INSERT INTO Customer(`code`, `name`, `call`, `mail`, `address`) VALUES ("OOOOOOOO", "김김김", "010-1234-1234", "kkk@naver.com", "서울시 용산구"); INSERT INTO Customer(`code`, `name`, `call`, `mail`, `address`) VALUES ("PPPPPPPP", "이이이", "010-2345-2345", "lll@naver.com", "서울시 용산구"); INSERT INTO Customer(`code`, `name`, `call`, `mail`, `address`) VALUES ("QQQQQQQQ", "박박박", "010-3456-3456", "ppp@naver.com", "서울시 종로구"); INSERT INTO Customer(`code`, `name`, `call`, `mail`, `address`) VALUES ("RRRRRRRR", "최최최", "010-4567-4567", "ccc@naver.com", "경기도 수원시");</pre> <table><tr><th></th><th>code</th><th>name</th><th>call</th><th>mail</th><th>address</th></tr><tr><td></td><td>OOOOOOOO</td><td>김김김</td><td>010-1234-1234</td><td>kkk@naver.com</td><td>서울시 용산구</td></tr><tr><td></td><td>PPPPPPPP</td><td>이이이</td><td>010-2345-2345</td><td>lll@naver.com</td><td>서울시 용산구</td></tr><tr><td></td><td>OOOOOOOO</td><td>박박박</td><td>010-3456-3456</td><td>ppp@naver.com</td><td>서울시 종로구</td></tr><tr><td></td><td>RRRRRRRR</td><td>최최최</td><td>010-4567-4567</td><td>ccc@naver.com</td><td>경기도 수원시</td></tr><tr><td></td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>		code	name	call	mail	address		OOOOOOOO	김김김	010-1234-1234	kkk@naver.com	서울시 용산구		PPPPPPPP	이이이	010-2345-2345	lll@naver.com	서울시 용산구		OOOOOOOO	박박박	010-3456-3456	ppp@naver.com	서울시 종로구		RRRRRRRR	최최최	010-4567-4567	ccc@naver.com	경기도 수원시		NULL	NULL	NULL	NULL	NULL
	code	name	call	mail	address																																
	OOOOOOOO	김김김	010-1234-1234	kkk@naver.com	서울시 용산구																																
	PPPPPPPP	이이이	010-2345-2345	lll@naver.com	서울시 용산구																																
	OOOOOOOO	박박박	010-3456-3456	ppp@naver.com	서울시 종로구																																
	RRRRRRRR	최최최	010-4567-4567	ccc@naver.com	경기도 수원시																																
	NULL	NULL	NULL	NULL	NULL																																
Franchise Table	<pre>-- add franchise INSERT INTO Franchise(`code`, `name`, `address`, `phone`) VALUES ("GGGGGGGG", "용산본점", "서울시 용산구", "02-5678-8765"); INSERT INTO Franchise(`code`, `name`, `address`, `phone`) VALUES ("HHHHHHHH", "종로점", "서울시 종로구", "02-1234-4321"); INSERT INTO Franchise(`code`, `name`, `address`, `phone`) VALUES ("IIIIIIII", "수원점", "경기도 수원시", "031-1357-2468");</pre> <table><tr><th></th><th>code</th><th>name</th><th>address</th><th>phone</th></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>서울시 용산구</td><td>02-5678-8765</td></tr><tr><td></td><td>HHHHHHHH</td><td>종로점</td><td>서울시 종로구</td><td>02-1234-4321</td></tr><tr><td></td><td>IIIIIIII</td><td>수원점</td><td>경기도 수원시</td><td>031-1357-2468</td></tr><tr><td></td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>		code	name	address	phone		GGGGGGGG	용산본점	서울시 용산구	02-5678-8765		HHHHHHHH	종로점	서울시 종로구	02-1234-4321		IIIIIIII	수원점	경기도 수원시	031-1357-2468		NULL	NULL	NULL	NULL											
	code	name	address	phone																																	
	GGGGGGGG	용산본점	서울시 용산구	02-5678-8765																																	
	HHHHHHHH	종로점	서울시 종로구	02-1234-4321																																	
	IIIIIIII	수원점	경기도 수원시	031-1357-2468																																	
	NULL	NULL	NULL	NULL																																	
Component Table	<pre>-- add component INSERT INTO Component(`code`, `name`, `price`) VALUES ("JJJJJJJJ", "육각회전나사", 3000); INSERT INTO Component(`code`, `name`, `price`) VALUES ("KKKKKKKK", "7.5V 4080mAh 배터리", 58000); INSERT INTO Component(`code`, `name`, `price`) VALUES ("LLLLLLLL", "19V 3.16A 60W 어댑터", 74000); INSERT INTO Component(`code`, `name`, `price`) VALUES ("MMMMMMMM", "R380 메인보드", 102000); INSERT INTO Component(`code`, `name`, `price`) VALUES ("NNNNNNNN", "RV409 그래픽카드", 208000);</pre> <table><tr><th></th><th>code</th><th>name</th><th>price</th></tr><tr><td></td><td>JJJJJJJJ</td><td>육각회전나사</td><td>3000</td></tr><tr><td></td><td>KKKKKKKK</td><td>7.5V 4080mAh 배터리</td><td>58000</td></tr><tr><td></td><td>LLLLLLLL</td><td>19V 3.16A 60W 어댑터</td><td>74000</td></tr><tr><td></td><td>MMMMMMMM</td><td>R380 메인보드</td><td>102000</td></tr><tr><td></td><td>NNNNNNNN</td><td>RV409 그래픽카드</td><td>208000</td></tr><tr><td></td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>		code	name	price		JJJJJJJJ	육각회전나사	3000		KKKKKKKK	7.5V 4080mAh 배터리	58000		LLLLLLLL	19V 3.16A 60W 어댑터	74000		MMMMMMMM	R380 메인보드	102000		NNNNNNNN	RV409 그래픽카드	208000		NULL	NULL	NULL								
	code	name	price																																		
	JJJJJJJJ	육각회전나사	3000																																		
	KKKKKKKK	7.5V 4080mAh 배터리	58000																																		
	LLLLLLLL	19V 3.16A 60W 어댑터	74000																																		
	MMMMMMMM	R380 메인보드	102000																																		
	NNNNNNNN	RV409 그래픽카드	208000																																		
	NULL	NULL	NULL																																		

Set Affiliation procedure	-- add affiliation with log CALL SetAffiliation("AAAAAAA", "GGGGGGGG"); CALL SetAffiliation("BBBBBBBB", "HHHHHHHH"); CALL SetAffiliation("CCCCCCCC", "IIIIIII"); CALL SetAffiliation("DDDDDDDD", "GGGGGGGG"); CALL SetAffiliation("EEEEEEEE", "HHHHHHHH"); CALL SetAffiliation("FFFFFFF", "GGGGGGGG");																																							
	Affiliation Table	<table><tr><th></th><th>employee_code</th><th>직원</th><th>지점</th><th>franchise_code</th></tr><tr><td></td><td>AAAAAAA</td><td>김가가</td><td>용산본점</td><td>GGGGGGGG</td></tr><tr><td></td><td>DDDDDDDD</td><td>최라라</td><td>용산본점</td><td>GGGGGGGG</td></tr><tr><td></td><td>FFFFFFF</td><td>서바바</td><td>용산본점</td><td>GGGGGGGG</td></tr><tr><td></td><td>BBBBBBBB</td><td>이나라</td><td>종로점</td><td>HHHHHHHH</td></tr><tr><td></td><td>EEEEEEEE</td><td>강마마</td><td>종로점</td><td>HHHHHHHH</td></tr><tr><td></td><td>CCCCCCCC</td><td>박다다</td><td>수원점</td><td>IIIIIII</td></tr></table>					employee_code	직원	지점	franchise_code		AAAAAAA	김가가	용산본점	GGGGGGGG		DDDDDDDD	최라라	용산본점	GGGGGGGG		FFFFFFF	서바바	용산본점	GGGGGGGG		BBBBBBBB	이나라	종로점	HHHHHHHH		EEEEEEEE	강마마	종로점	HHHHHHHH		CCCCCCCC	박다다	수원점	IIIIIII
		employee_code	직원	지점	franchise_code																																			
	AAAAAAA	김가가	용산본점	GGGGGGGG																																				
	DDDDDDDD	최라라	용산본점	GGGGGGGG																																				
	FFFFFFF	서바바	용산본점	GGGGGGGG																																				
	BBBBBBBB	이나라	종로점	HHHHHHHH																																				
	EEEEEEEE	강마마	종로점	HHHHHHHH																																				
	CCCCCCCC	박다다	수원점	IIIIIII																																				
Affiliation Change History Table	<table><tr><th></th><th>employee_code</th><th>franchise_code</th><th>time</th></tr><tr><td></td><td>AAAAAAA</td><td>GGGGGGGG</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>DDDDDDDD</td><td>GGGGGGGG</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>FFFFFFF</td><td>GGGGGGGG</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>BBBBBBBB</td><td>HHHHHHHH</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>EEEEEEEE</td><td>HHHHHHHH</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>CCCCCCCC</td><td>IIIIIII</td><td>2019-12-13 04:46:10</td></tr><tr><td></td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>					employee_code	franchise_code	time		AAAAAAA	GGGGGGGG	2019-12-13 04:46:10		DDDDDDDD	GGGGGGGG	2019-12-13 04:46:10		FFFFFFF	GGGGGGGG	2019-12-13 04:46:10		BBBBBBBB	HHHHHHHH	2019-12-13 04:46:10		EEEEEEEE	HHHHHHHH	2019-12-13 04:46:10		CCCCCCCC	IIIIIII	2019-12-13 04:46:10		NULL	NULL	NULL				
	employee_code	franchise_code	time																																					
	AAAAAAA	GGGGGGGG	2019-12-13 04:46:10																																					
	DDDDDDDD	GGGGGGGG	2019-12-13 04:46:10																																					
	FFFFFFF	GGGGGGGG	2019-12-13 04:46:10																																					
	BBBBBBBB	HHHHHHHH	2019-12-13 04:46:10																																					
	EEEEEEEE	HHHHHHHH	2019-12-13 04:46:10																																					
	CCCCCCCC	IIIIIII	2019-12-13 04:46:10																																					
	NULL	NULL	NULL																																					

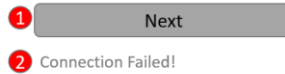
Order Component procedure	-- add component with log CALL OrderComponent("19121301", "AAAAAAA", "GGGGGGGG", "JJJJJJJ", 10); CALL OrderComponent("19121302", "AAAAAAA", "GGGGGGGG", "KKKKKKKK", 3); CALL OrderComponent("19121303", "AAAAAAA", "GGGGGGGG", "LLLLLLLL", 2); CALL OrderComponent("19121304", "AAAAAAA", "GGGGGGGG", "MMMMMMMM", 3); CALL OrderComponent("19121305", "AAAAAAA", "GGGGGGGG", "NNNNNNNN", 2); CALL OrderComponent("19121306", "BBBBBBBB", "HHHHHHHH", "KKKKKKKK", 5); CALL OrderComponent("19121307", "CCCCCCCC", "IIIIIII", "NNNNNNNN", 3);																																																																		
	Component List Table	<table><tr><th></th><th>code</th><th>지점명</th><th>code</th><th>부품명</th><th>개수</th></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>JJJJJJJ</td><td>육각회전나사</td><td>10</td></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>KKKKKKKK</td><td>7.5V 4080mAh 배터리</td><td>3</td></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>LLLLLLLL</td><td>19V 3.16A 60W 어댑터</td><td>2</td></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>MMMMMMMM</td><td>R380 메인보드</td><td>3</td></tr><tr><td></td><td>GGGGGGGG</td><td>용산본점</td><td>NNNNNNNN</td><td>RV409 그래픽카드</td><td>2</td></tr><tr><td></td><td>HHHHHHHH</td><td>종로점</td><td>KKKKKKKK</td><td>7.5V 4080mAh 배터리</td><td>5</td></tr><tr><td></td><td>IIIIIII</td><td>수원점</td><td>NNNNNNNN</td><td>RV409 그래픽카드</td><td>3</td></tr></table>					code	지점명	code	부품명	개수		GGGGGGGG	용산본점	JJJJJJJ	육각회전나사	10		GGGGGGGG	용산본점	KKKKKKKK	7.5V 4080mAh 배터리	3		GGGGGGGG	용산본점	LLLLLLLL	19V 3.16A 60W 어댑터	2		GGGGGGGG	용산본점	MMMMMMMM	R380 메인보드	3		GGGGGGGG	용산본점	NNNNNNNN	RV409 그래픽카드	2		HHHHHHHH	종로점	KKKKKKKK	7.5V 4080mAh 배터리	5		IIIIIII	수원점	NNNNNNNN	RV409 그래픽카드	3														
		code	지점명	code	부품명	개수																																																													
	GGGGGGGG	용산본점	JJJJJJJ	육각회전나사	10																																																														
	GGGGGGGG	용산본점	KKKKKKKK	7.5V 4080mAh 배터리	3																																																														
	GGGGGGGG	용산본점	LLLLLLLL	19V 3.16A 60W 어댑터	2																																																														
	GGGGGGGG	용산본점	MMMMMMMM	R380 메인보드	3																																																														
	GGGGGGGG	용산본점	NNNNNNNN	RV409 그래픽카드	2																																																														
	HHHHHHHH	종로점	KKKKKKKK	7.5V 4080mAh 배터리	5																																																														
	IIIIIII	수원점	NNNNNNNN	RV409 그래픽카드	3																																																														
Component Order History Table	<table><tr><th></th><th>code</th><th>time</th><th>employee_code</th><th>franchise_code</th><th>component_code</th><th>number</th></tr><tr><td></td><td>19121301</td><td>2019-12-13 04:47:38</td><td>AAAAAAA</td><td>GGGGGGGG</td><td>JJJJJJJ</td><td>10</td></tr><tr><td></td><td>19121302</td><td>2019-12-13 04:47:38</td><td>AAAAAAA</td><td>GGGGGGGG</td><td>KKKKKKKK</td><td>3</td></tr><tr><td></td><td>19121303</td><td>2019-12-13 04:47:38</td><td>AAAAAAA</td><td>GGGGGGGG</td><td>LLLLLLLL</td><td>2</td></tr><tr><td></td><td>19121304</td><td>2019-12-13 04:47:38</td><td>AAAAAAA</td><td>GGGGGGGG</td><td>MMMMMMMM</td><td>3</td></tr><tr><td></td><td>19121305</td><td>2019-12-13 04:47:38</td><td>AAAAAAA</td><td>GGGGGGGG</td><td>NNNNNNNN</td><td>2</td></tr><tr><td></td><td>19121306</td><td>2019-12-13 04:47:38</td><td>BBBBBBBB</td><td>HHHHHHHH</td><td>KKKKKKKK</td><td>5</td></tr><tr><td></td><td>19121307</td><td>2019-12-13 04:47:38</td><td>CCCCCCCC</td><td>IIIIIII</td><td>NNNNNNNN</td><td>3</td></tr><tr><td></td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>					code	time	employee_code	franchise_code	component_code	number		19121301	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	JJJJJJJ	10		19121302	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	KKKKKKKK	3		19121303	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	LLLLLLLL	2		19121304	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	MMMMMMMM	3		19121305	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	NNNNNNNN	2		19121306	2019-12-13 04:47:38	BBBBBBBB	HHHHHHHH	KKKKKKKK	5		19121307	2019-12-13 04:47:38	CCCCCCCC	IIIIIII	NNNNNNNN	3		NULL	NULL	NULL	NULL	NULL	NULL
	code	time	employee_code	franchise_code	component_code	number																																																													
	19121301	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	JJJJJJJ	10																																																													
	19121302	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	KKKKKKKK	3																																																													
	19121303	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	LLLLLLLL	2																																																													
	19121304	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	MMMMMMMM	3																																																													
	19121305	2019-12-13 04:47:38	AAAAAAA	GGGGGGGG	NNNNNNNN	2																																																													
	19121306	2019-12-13 04:47:38	BBBBBBBB	HHHHHHHH	KKKKKKKK	5																																																													
	19121307	2019-12-13 04:47:38	CCCCCCCC	IIIIIII	NNNNNNNN	3																																																													
	NULL	NULL	NULL	NULL	NULL	NULL																																																													

Add Service	-- add service INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_001", "2019-01-06 15:30:22", 4, "SW", True, "OOOOOOOO", "AAAAAAA"); INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_002", "2019-02-01 16:04:03", 4, "HW", True, "PPPPPPPP", "DDDDDDDD"); INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_003", "2019-03-08 18:33:21", 4, "SW", True, "QQQQQQQQ", "BBBBBBBB"); INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_004", "2019-05-10 20:33:21", 4, "BUY", True, "QQQQQQQQ", "EEEEEEEE"); INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_005", "2019-08-11 22:33:21", 4, "SW", True, "OOOOOOOO", "FFFFFFF"); INSERT INTO Service(`code`, `time`, `progress`, `type`, `way`, `customer_code`, `employee_code`) VALUES("SERV_006", "2019-10-23 10:33:21", 4, "SW", True, "RRRRRRRR", "CCCCCCCC");									
		code	time	progress	type	way	customer_code	고객명	employee_code	담당직원
		SERV_001	2019-01-06 15:30:22	4	SW	1	OOOOOOOO	김김김	AAAAAAA	김가가
		SERV_005	2019-08-11 22:33:21	4	SW	1	OOOOOOOO	김김김	FFFFFFF	서바바
		SERV_002	2019-02-01 16:04:03	4	HW	1	PPPPPPPP	이이이	DDDDDDDD	최라라
		SERV_003	2019-03-08 18:33:21	4	SW	1	OOOOOOOO	박박박	BBBBBBBB	이나라
		SERV_004	2019-05-10 20:33:21	4	BUY	1	OOOOOOOO	박박박	EEEEEEEE	강마마
		SERV_006	2019-10-23 10:33:21	4	SW	1	RRRRRRRR	최최최	CCCCCCCC	박다다

프로그램 상세 화면 및 사용 매뉴얼

1 시작 화면

A 전자 서비스 센터 통합 관리 프로그램



서버와의 mySQL 연결이 성공해야 Next 버튼이 활성화된다.
 서버와의 연결이 실패할 경우 (2)에 Connection Failed! 라는 라벨이 나타난다.
 연결에 성공한 경우 Next 버튼을 눌러 다음 화면으로 이동하면 된다.

2-1 지점 조회 화면

화면은 좌측 메뉴와 우측 메인 영역으로 나뉩니다. 좌측 메뉴에는 '지점조회', '직원조회', '부품조회', '서비스 조회', '부품 주문', '서비스 신청', '지점 변경 이력', '부품 판매 이력', '결제 이력'이 있습니다. 메인 영역 상단에는 '지점 조회' 탭과 '지점명' (1), '주소 address' (2), '전화번호 phone' (3)의 검색 필터가 있습니다. '지점명' (4)은 드롭다운 메뉴입니다. 필터 아래에는 '소속직원' 섹션과 '소속 직원 표' (이름, 직급, 생년월일, 전화번호, 이메일, 주소, 은행, 계좌번호, 입사일)가 있습니다. 그 다음에는 '부품 재고' 섹션과 '부품 재고표' (부품명, 가격, 개수)가, 마지막으로 '서비스 이력' 섹션과 '서비스 이력표' (부품명, 가격, 개수)가 있습니다.

화면 1에서 이동했거나, 지점조회 버튼을 누르면 지점 조회 창이 나온다.
 (1)에서 지점을 선택하면, 지점 정보가 알맞게 표시된다.
 (2)~(4)를 각각 클릭하여 주소, 전화번호, 지점명을 바꿀 수 있다.

2-2
직원 조회 화면

직원조회 버튼을 누르면 직원 조회 창이 나온다.
 (1)에서 지점명을 선택한 뒤, (2)에서 직원명을 선택하면 해당 직원의 정보가 표시된다. 직원이 많아지는 것을 고려하여 편의상 지점명->직원명으로 접근하게끔 했다.
 (3)~(9)를 각각 클릭하여 직원명, 직급, 생년월일, 전화번호, 이메일, 주소, 은행, 계좌번호를 바꿀 수 있다.

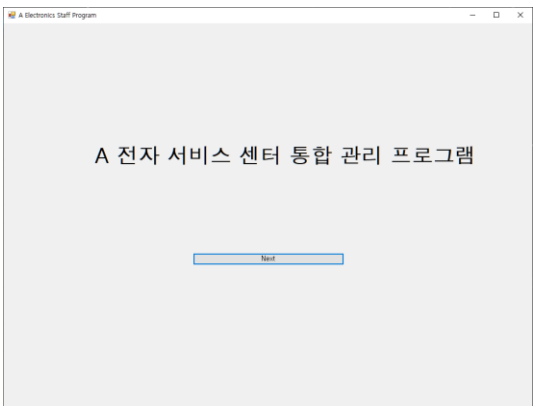
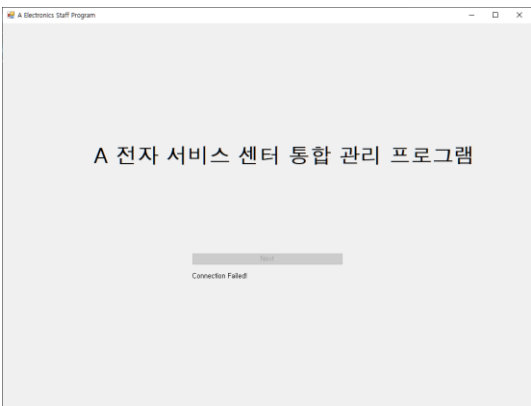
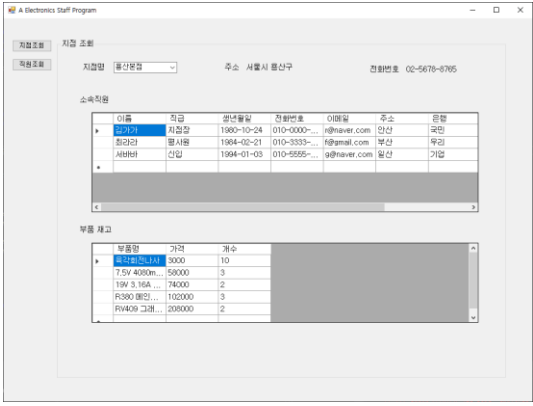
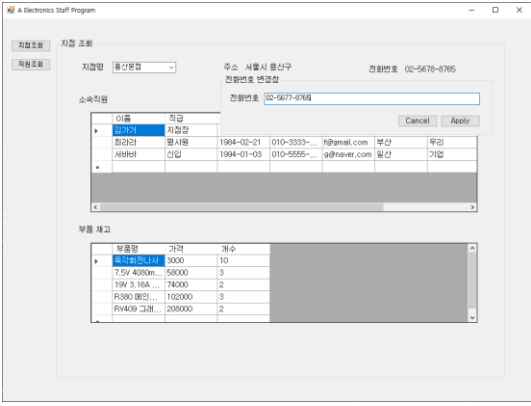
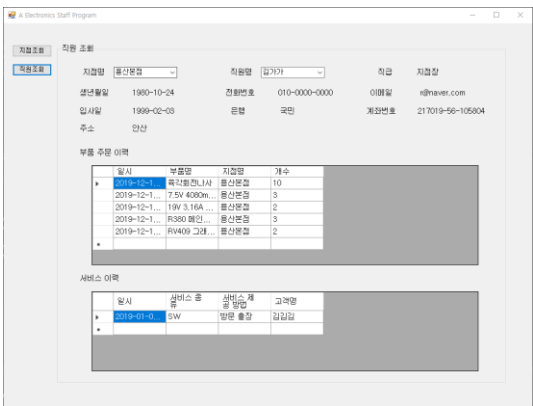
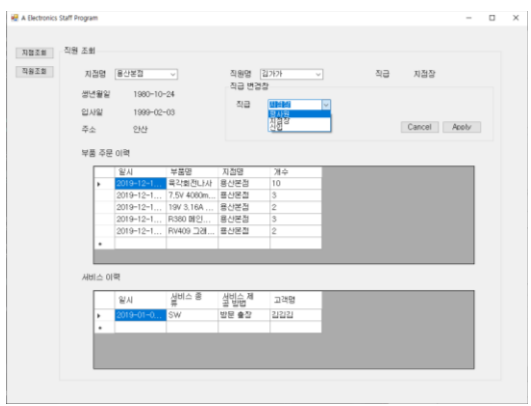
2-3
부품 조회 화면

부품조회 버튼을 누르면 부품 조회 창이 나온다.
 (1)에서 지점명을 선택한 뒤, (2)에서 부품명을 선택하면 해당 부품의 정보가 표시된다. 지점에 존재하지 않는 부품은 나오지 않는다.
 (3)~(4)를 클릭하여 부품명과 가격을 변경할 수 있다

<p>2-4 부품 주문 화면</p>	<div> <div> 지점조회 직원조회 부품조회 서비스 조회 <hr/> 부품 주문 서비스 신청 <hr/> 지점 변경 이력 부품 판매 이력 결제 이력 </div> <div> <div>부품 주문</div> <div> 지점명 1 <input type="text"/> 부품명 3 <input type="text"/> 담당자 2 <input type="text"/> 개수 4 <input type="text"/> 5 <input type="button" value="Order"/> </div> <div>지점 부품 재고</div> <div> 부품 재고 (부품명, 개수) </div> <div>지점 부품 주문 이력</div> <div> 부품 주문 이력 (일시, 지점명, 담당자, 부품명, 개수) </div> </div> </div>
	<p>부품 주문 버튼을 누르면 부품 주문 창이 나온다. (1)에서 지점명을, (2)에서 담당자를, (3)에서 부품명을, (4)에서 개수를 선택하고 (5)를 누르면 부품 주문을 할 수 있다. 부품 주문이 들어가면, 주문 이력에 새로운 이력이 추가된다.</p>
<p>2-5 서비스 신청 화면</p>	<div> <div> 지점조회 직원조회 부품조회 서비스 조회 <hr/> 부품 주문 서비스 신청 <hr/> 지점 변경 이력 부품 판매 이력 결제 이력 </div> <div> <div>서비스 신청</div> <div> 서비스 종류 1 <input type="text"/> 이름 3 <input type="text"/> 메일 5 <input type="text"/> 출장여부 2 <input type="text"/> 전화번호 4 <input type="text"/> 주소 6 <input type="text"/> 7 <input type="button" value="Apply"/> </div> <div>서비스 이력</div> <div> 서비스 이력 (일시, 서비스 진행 상태, 서비스 종류, 서비스 제공 방법, 담당 직원, 설명) </div> </div> </div>
	<p>서비스 신청 버튼을 누르면 서비스 신청 창이 나온다. (1)에서 서비스 종류, (2)에서 출장 여부를 선택한다. (3)에서 이름 입력, (4)에서 전화번호 입력, (5)에서 메일, (6)에서 주소를 입력하고 7을 누르면 적용된다. 서비스가 접수되면, 서비스 이력에 새로운 이력이 추가된다.</p>

<p>2-6 지점 변경 이력</p>	<div data-bbox="517 253 616 389"> 지점조회 직원조회 부품조회 서비스 조회 </div> <div data-bbox="517 432 616 629"> 부품 주문 서비스 신청 지점 변경 이력 부품 판매 이력 결제 이력 </div> <div data-bbox="638 253 1270 797"> 지점 변경 이력 <div data-bbox="699 327 759 353">대리점</div> <div data-bbox="786 315 911 353"> <div>1</div> <input type="text"/> ▼ </div> <div data-bbox="962 327 1007 353">직원</div> <div data-bbox="1026 315 1150 353"> <div>2</div> <input type="text"/> ▼ </div> <div data-bbox="675 432 805 459">지점 변경 이력</div> <div data-bbox="695 474 1233 593"> 지점 변경 이력 (대리점 입장에서 이력 : 일시, 직원) </div> <div data-bbox="675 613 805 640">지점 변경 이력</div> <div data-bbox="695 656 1233 775"> 지점 변경 이력 (직원 입장에서 이력 : 일시, 대리점) </div> </div> <p>(1)에서 대리점을 선택하고, (2)에서 소속 직원을 선택하면 지점 변경 이력에 각각 정보가 갱신된다.</p>
<p>2-7 부품 판매 이력</p>	<div data-bbox="517 940 616 1077"> 지점조회 직원조회 부품조회 서비스 조회 </div> <div data-bbox="517 1120 616 1317"> 부품 주문 서비스 신청 지점 변경 이력 부품 판매 이력 결제 이력 </div> <div data-bbox="638 940 1270 1485"> 부품 판매 이력 <div data-bbox="699 1014 759 1041">대리점</div> <div data-bbox="786 1003 911 1041"> <div>1</div> <input type="text"/> ▼ </div> <div data-bbox="951 1014 1011 1041">부품명</div> <div data-bbox="1026 1003 1150 1041"> <div>2</div> <input type="text"/> ▼ </div> <div data-bbox="675 1120 805 1146">부품 판매 이력</div> <div data-bbox="695 1162 1233 1281"> 지점 변경 이력 (대리점 입장에서 이력 : 일시, 부품명, 판매자, 구매자) </div> <div data-bbox="675 1301 805 1328">부품 판매 이력</div> <div data-bbox="695 1344 1233 1462"> 지점 변경 이력 (부품 입장에서 이력 : 일시, 대리점, 판매자, 구매자) </div> </div> <p>(1)에서 대리점을 선택하고, (2)에서 부품명을 선택하면 부품 판매 이력에 각각 정보가 갱신된다.</p>

<div data-bbox="244 551 352 627"> <p>2-8 결제 이력</p> </div>	<div data-bbox="411 241 1225 851"> <div data-bbox="438 264 534 409"> <div>지점조회</div> <div>직원조회</div> <div>부품조회</div> <div>서비스 조회</div> </div> <div data-bbox="438 443 534 647"> <div>부품 주문</div> <div>서비스 신청</div> <div>지점 변경 이력</div> <div>부품 판매 이력</div> <div>결제 이력</div> </div> <div data-bbox="571 264 1190 813"> <div>결제 이력</div> <div> <div>대리점 <div data-bbox="707 331 831 376">1</div></div> <div>기간 <div data-bbox="948 331 1072 376">2</div></div> </div> <div>기간 내 대리점 결제 이력</div> <div data-bbox="616 490 1153 609"> <div>기간 내 대리점 결제 이력 (일시, 결제자, 가격)</div> </div> <div>전체 이력</div> <div data-bbox="616 674 1153 792"> <div>전체 결제 이력 (일시, 대리점, 담당자, 결제자, 가격)</div> </div> </div> </div>
	<div data-bbox="411 869 1390 945"> <p>(1)에서 대리점을 선택하고, (2)에서 기간을 선택하면 선택한 기간 내의 대리점 결제 이력이 갱신된다.</p> </div>

프로그램 실행 화면		
1 시작 화면		
	연결 성공	
2-1 지점 조회 화면		
	기본 화면	
2-2 직원 조회 화면		
	기본 화면	
2-3 부품 조회 화면		
2-4		

서비스 조회 화면		
2-5 부품 주문 화면		
2-6 지점 변경 이력 화면		
2-7 부품 판매 이력 화면		
2-8 결제 이력 화면		