



COFFEE – CAFÉ

운영 관리

조원 : 박지원, 이진영, 하경훈

업무 정리



1. 고객의 정보를 등록한다.
2. 직원의 정보를 등록한다.
 - 출근 시간표에 따라 직원의 근무 태도를 관리한다.
 - 근무를 바탕으로 직원들의 월급을 지급한다.
3. 메뉴를 판매한다.
 - 고객의 등급에 따라 할인을 적용한다.
 - 주문 구분에 따라 할인을 적용한다.
 - 특별한 날 이벤트(할인)를 적용하여 판매한다.
4. 결제 금액의 1%를 고객의 포인트로 적립한다.
 - 포인트로 고객의 등급을 구분한다.
 - 고객은 포인트를 사용하여 결제 금액에 할인을 받는다.
5. 필요한 물품 구입 내역을 경비에 기록한다.

업무 정의서



STORE MANAGEMENT

총 수입과 순이익
판매구분에 따른 매출
재고관리
메뉴별 판매 랭킹
배달이 많은 지역

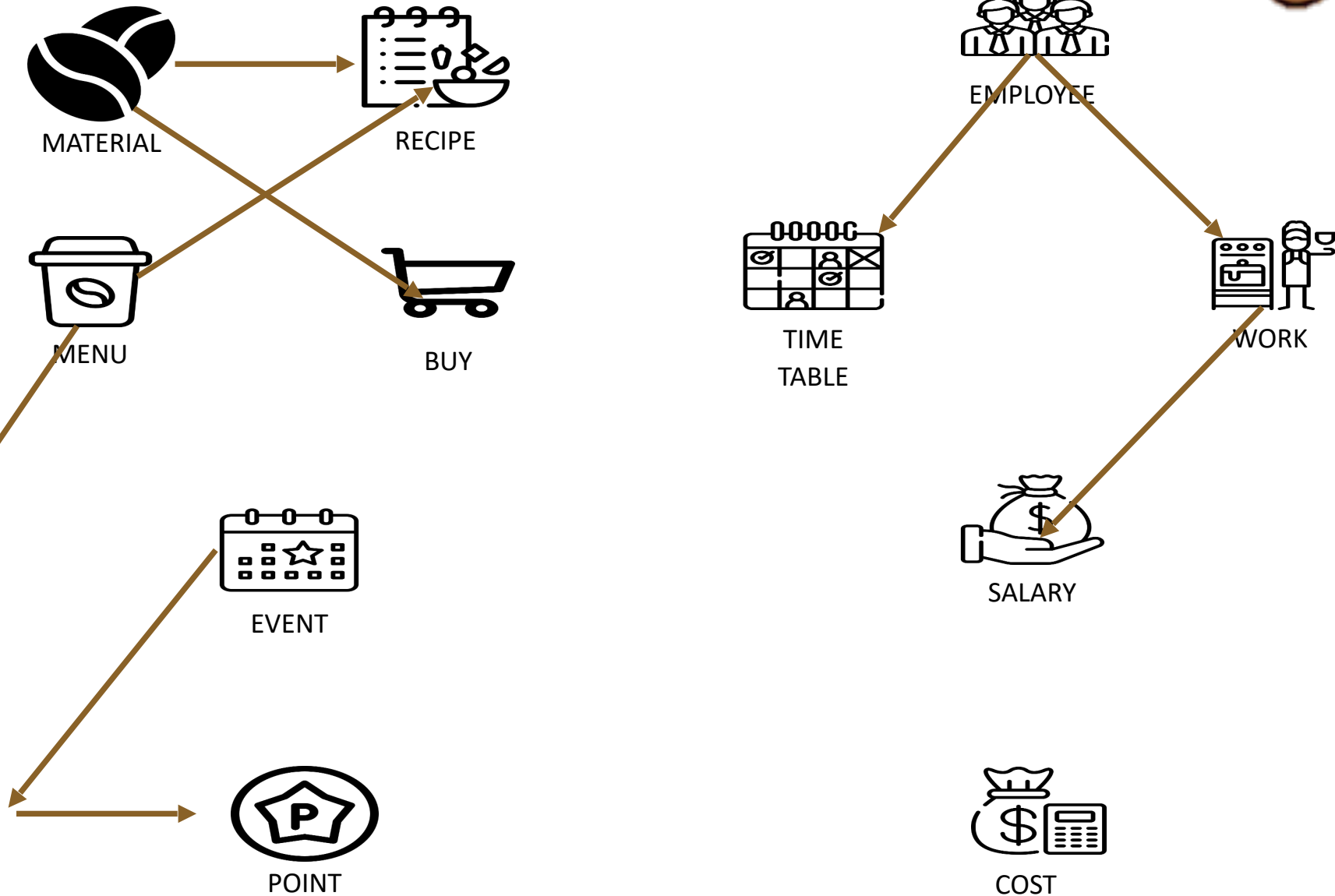
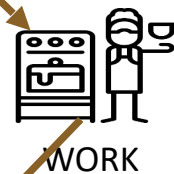
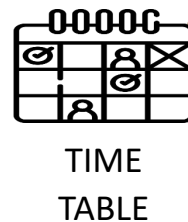
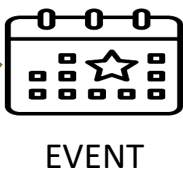
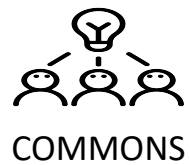
EMPLOYEE MANAGEMENT

고객의 포인트
고객의 환불횟수
고객별 메뉴 선호도
고객별 선호 이용시간

CUSTOMER MANAGEMENT

직원의 근무태도 관리
주휴수당 지급
매출액이 적은 직원

관계도



속성정리



그룹 공통

1. 그룹 ID
2. 그룹 이름
3. 그룹 시퀀스

공통 코드

1. 공통코드 ID
2. 공통코드명
3. 공통코드 레벨
4. 공통코드 순서
5. 부모 ID
6. 그룹 ID
7. 예외 1
8. 예외 2

고객

1. 고객 ID
2. 고객 이름
3. 고객 주소
4. 고객 전화번호
5. 고객 생년월일
6. 고객 등급
7. 비고

메뉴

1. 메뉴 ID
2. 메뉴 구분 ID
3. 메뉴 이름
4. 사이즈
5. HOT / COLD
6. 가격
7. 비고

직원

1. 직원 ID
2. 직원 이름
3. 전화번호
4. 생년월일
5. 직책
6. 시급
7. 입사일
8. 퇴사일
9. 비고

경비

1. 경비 ID
2. 경비 내용
3. 지출 금액
4. 지출 날짜
5. 비고

이벤트

1. 이벤트 ID
2. 이벤트 내용
3. 적용날짜
4. 할인율
5. 고객 등급

판매

1. 주문번호
2. 인덱스
3. 회원 ID
4. 메뉴 ID
5. 수량
6. 주문 구분
7. 주문 시간
8. 환불 여부

가격

1. 주문번호
2. 이벤트 ID
3. 등급할인 코드
4. 사용할 포인트
5. 원 금액
6. 결제금액

포인트

1. 주문번호
2. 고객 ID
3. 포인트 금액
4. 적립날짜
5. 적립 / 사용 구분

속성정리



재료

1. 재료 ID
2. 재료 이름
3. 제조사
4. 원산지
5. 비고

재료 구매

1. 재료 ID
2. 재료 인덱스
3. 가격
4. 수량
5. 구매날짜

근무 시간표

1. 직원 ID
2. 근무 날짜
3. 출근 시각
4. 퇴근 시각
5. 비고

근무

1. 직원 ID
2. 근무 인덱스
3. 근무 날짜
4. 출퇴근 시각
5. 출퇴근 구분

급여

1. 직원 ID
2. 급여 인덱스
3. 한달 급여
4. 해당 급여
지급 대상 월
5. 지급 날짜

테이블 정의



테이블 명 : GROUP_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
GROUP_ID	CHAR(4)	O	O	그룹 ID
GROUP_NAME	VARCHAR2(50)	O		그룹 이름
GROUP_SEQ	NUMBER(3)	O		시퀀스

테이블 정의



테이블 명 : COMMONS_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
COM_ID	CHAR(6)	O	O	공통코드
COM_VAL	VARCHAR2(100)	O		공통코드명
COM_LVL	NUMBER(8)	O		공통코드 레벨
COM_SEQ	NUMBER(8)	O		공통코드 순서
PARENT_ID	CHAR(4))	O		부모코드
GROUP_ID	CHAR(4)	O		그룹코드
EXCEPT1	VARCHAR2(20)			예외1 (할인율)
EXCEPT2	VARCHAR2(20)			예외2 (등급 기준)

테이블 정의



테이블 명 : CUSTOMER_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	기본값	비고
CUS_ID	CHAR(6)	O	O		고객 아이디 CUS001
CUS_NAME	VARCHAR2(30)	O			고객 이름
CUS_ADDR	VARCHAR2(300)	O			고객 주소
CUS_TEL	VARCHAR2(13)	O			고객 전화번호 000-0000-0000
CUS_BIRTH	VARCHAR2(10)	O			고객 생년월일 YYYY-MM-DD
CUS_GRADE	CHAR(6)	O		'COM009'	고객 등급
CUS_CONTENTS	VARCHAR2(100)				비고

테이블 정의



테이블 명 : MENU_TBL

필드명	테이터 타입(크기)	Not Null	Primary KEY	비고
MENU_ID	VARCHAR2(4)	O	O	메뉴 아이디 M001
MENU_PART_ID	VARCHAR2(6)	O		메뉴구분 아이디
MENU_NAME	VARCHAR2(50)	O		메뉴명
MENU_SIZE	CHAR(1)	O		메뉴사이즈(S/ L)
MENU_HC	CHAR(1)	O		HOT& COLD
MENU_PRICE	NUMBER(5)	O		메뉴가격
MENU_CONTENTS	VARCHAR2(100)			비고

테이블 정의



테이블 명 : EMPLOYEE_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	기본값	비고
EMP_ID	CHAR(6)	O	O		직원 아이디 EMP001
EMP_NAME	VARCHAR2(30)	O			직원이름
EMP_TEL	VARCHAR2(13)	O			직원 전화번호 000-0000-0000
EMP_BIRTH	VARCHAR2(10)	O			직원 생년월일 YYYY-MM-DD
EMP_POS	CHAR(6)	O			직원 직책(공통코드)
EMP_SALARY	NUMBER(8)	O			직원 시급
EMP_ENTER_DATE	VARCHAR2(10)	O			직원 입사일
EMP_LEAVE_DATE	VARCHAR2(10)				퇴사일 YYYY-MM-DD (퇴사 전, NULL)
EMP_CONTENTS	VARCHAR2(100)				비고

테이블 정의



테이블 명 : MATERIAL_TBL

필드명	테이터 타입(크기)	Not Null	Primary KEY	기본값	비고
MAT_ID	CHAR(6)	O	O		재료 아이디 MAT001
MAT_NAME	VARCHAR2(30)	O			재료 이름
MAT_COM	VARCHAR2(30)	O			재료 제조사
MAT_ORIGIN	VARCHAR2(30)	O			재료 원산지
MAT_CONTENTS	VARCHAR2(100)				비고

테이블 정의



테이블 명 : EVENT_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
EVENT_ID	CHAR(6)	O	O	이벤트 아이디(EV0001)
EVENT_CONTENTS	VARCHAR2(100)	O		이벤트 내용
EVENT_DATE	VARCHAR2(10)	O		할인이벤트 적용 날짜
EVENT_DIS_RATE	NUMBER(3,2)	O		할인율 EX) 0.15
EVENT_DIS_GRD	CHAR(6)	O		특정 등급(COM005) 이상이벤트적용

테이블 정의



테이블 명 : SALE_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	기본값	비고
ORD_ID	CHAR(6)	O	O		주문번호(A00001)
ORD_IDX	NUMBER(2)	O	O		주문번호 인덱스
CUS_ID	CHAR(6)	O			회원 아이디 (CUS001)
MENU_ID	CHAR(4)	O			메뉴 아이디 (M001)
MENU_QTY	NUMBER(2)	O			메뉴 주문 수량
ORD_PART	CHAR(6)	O			주문 구분COM010
ORD_DATE	DATE	O			주문 시간
REFUND_YN	CHAR(1)	O		N	환불 여부

테이블 정의



테이블 명 : POINT_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	기본값	비고
ORD_ID	CHAR(6)	O	O		주문번호(A00001)
CUS_ID	CHAR(6)	O	O		고객 아이디(CUS001)
SAVE_POINT	NUMBER(5)	O			적립 포인트
SAVE_DATE	VARCHAR2(10)	O			포인트 적립 날짜 (YYYY-MM-DD)
USE_POINT_YN	CHAR(1)	O		N	적립 및 사용 구분

테이블 정의



테이블 명 : BUY_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
MAT_ID	CHAR(6)	O	O	재료 아이디 MAT001
MAT_IDX	NUMBER(3)	O	O	재료 인덱스
MAT_PRICE	NUMBER(10)	O		재료 가격
MAT_QTY	NUMBER(3)	O		재료 수량
MAT_DATE	VARCHAR2(10)	O		재료 구매 날짜 (YYYY-MM-DD)

테이블 정의



테이블 명 : WORK_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
EMP_ID	CHAR(6)	O	O	직원 아이디 EMP001
WORK_IDX	NUMBER(3)	O	O	근무 인덱스
WORK_DATE	VARCHAR2(10)	O		근무 날짜
WORK_TIME	VARCHAR2(5)	O		출퇴근 시간
WORK_IO	CHAR(1)	O		출퇴근 구분 (I,O)

테이블 정의



테이블 명 : SALARY_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
EMP_ID	CHAR(6)	O	O	직원 아이디 EMP001
SAL_IDX	NUMBER(3)	O	O	급여 인덱스
SAL_TLT	NUMBER(8)	O		한 달 총 급여
SAL_MON	VARCHAR2(7)	O		해당 급여 지급 대상 월
SAL_DATE	VARCHAR2(10)	O		급여 지급 일

테이블 정의



테이블 명 : COST_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
COST_ID	CHAR(6)	O	O	경비 아이디 COS001
COST_NAME	VARCHAR2(50)	O		경비 목록
COST_PRICE	NUMBER(10)	O		경비지출 금액
COST_DATE	VARCHAR2(10)	O		경비지출 날짜
COST_CONTENTS	VARCHAR2(10)	O		비고

테이블 정의



테이블 명 : TIMETABLE_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
EMP_ID	CHAR(6)	O	O	직원 아이디 EMP001
WORK_DATE	VARCHAR2(10)	O		근무 날짜
IN_TIME	VARCHAR2(5)	O		출근 시간
OUT_TIME	VARCHAR2(5)	O		퇴근 시간
TT_CONTENTS	VARCHAR2(200)			비고

테이블 정의



테이블 명 : RECIPE_TBL

필드명	테이터 타입(크기)	Not Null	Primary Key	비고
MENU_ID	CHAR(6)	O	O	메뉴 아이디
MAT_ID	CHAR(6)	O	O	재료 아이디 MAT001
MAT_QTY	NUMBER(4)	O		재료 수량
MAT_WATER	NUMBER(4)	O		필요한 물 량
REC_CONTENTS	VARCHAR2(100)	O		비고

패키지 정의



PACKAGE명	PROCEDURE	설명	작성자
PKG_GROUP	PROC_MOD_GROUP_TBL	GROUP_TBL(그룹 테이블)데이터 추가 및 수정 프로시저	하경훈
	PROC_SEL_GROUP_TBL	GROUP_TBL 데이터 조회 프로시저	
	PROC_DEL_GROUP_TBL	GROUP_TBL 데이터 삭제 프로시저	
PKG_COMMONS	PROC_INS_COMMONS_TB	공통 테이블 추가 프로시저	하경훈
	PROC_SEL_COMMONS_TBL	공통 테이블 조회 프로시저	
	PROC_UP_COMMONS_TBL	공통 테이블 수정 프로시저	
	PROC_DEL_COMMONS_TBL	공통 테이블 삭제 프로시저	
PKG_CUSTOMER	PROC_MOD_CUSTOEMR	CUSTOMER_TBL(고객 테이블) 데이터 추가 및 수정 프로시저	박지원
	PROC_SEL_CUSTOMER	CUSTOMER_TBL 조회(검색) 프로시저	
	PROC_DEL_CUSTOMER	CUSTOMER_TBL 삭제 프로시저, 삭제되면 다른 테이블에서는 'DELETE'표시	
PKG_EMPLOYEE	PROC_INS_EMPLOYEE_TBL	EMPLOYEE_TBL(직원 테이블) 추가 프로시저	하경훈
	PROC_SEL_EMPLOYEE_TBL	EMPLOYEE_TBL 조회 프로시저	
	PROC_UP_EMPLOYEE_TBL	EMPLOYEE_TBL 수정 프로시저	
	PROC_LEAVE_EMPLOYEE_TBL	EMPLOYEE_TBL 직원 퇴사일 입력 프로시저	
	PROC_DEL_EMPLOYEE_TBL	EMPLOYEE_TBL 삭제 프로시저	

패키지 정의



PKG_MATRIAL	PROC_MOD_MATRIAL_TBL	재료 테이블 수정,추가 프로시저	이진영
	PROC_SEL_MATRIAL_TBL	재료 테이블 조회 프로시저	
	PROC_DEL_MATRIAL_TBL	재료 테이블 삭제 프로시저	
PKG_BUY	PROC_INS_BUY	BUY_TBL 추가 프로시저	이진영
	PROC_SEL_BUY	BUY_TBL 조회(검색) 프로시저	
	PROC_DEL_BUY	BUY_TBL 삭제 프로시저	
	PROC_UP_BUY	BUY_TBL 수정 프로시저	
PKG_COST	PROC_INS_COST	COST_TBL 추가 프로시저	이진영
	PROC_SEL_COST	COST_TBL 조회 프로시저	
	PROC_UP_COST	COST_TBL 수정 프로시저	
	PROC_DEL_COST	COST_TBL 삭제 프로시저	
PKG_TIMETABLE_TBL	PROC_INS_TIMETABLE_TBL	TIMETABLE_TBL INSERT 프로시저	하경훈
	PROC_UP_TIMETABLE_TBL	TIMETABLE_TBL UPDATE 프로시저	
	PROC_SEL_TIMETABLE_TBL	TIMETABLE_TBL SELECT 프로시저	
	PROC_DEL_TIMETABLE_TBL	TIMETABLE_TBL DELETE 프로시저	

패키지 정의



PKG_WORK	PROC_INS_WORK_TBL	WORK_TBL(출퇴근 테이블) 추가 프로시저	하경훈
	PROC_SEL_WORK_TBL	WORK_TBL 조회 프로시저	
	PROC_UP_WORK_TBL	WORK_TBL 수정 프로시저	
	PROC_DEL_WORK_TBL	WORK_TBL 삭제 프로시저	
PKG_SALARY	PROC_INS_SALARY_TBL	SALARY_TBL(급여테이블) 추가 프로시저	하경훈
	PROC_SEL_SALARY_TBL	SALARY_TBL 조회 프로시저	
	PROC_UP_SALARY_TBL	SALARY_TBL 수정 프로시저	
	PROC_DEL_SALARY_TBL	SALARY_TBL 삭제 프로시저	
PKG_MENU	PROC_MOD_MENU	MENU_TBL(메뉴테이블) 데이터 추가 및 수정 프로시저	이진영
	PROC_SEL_MENU	MENU_TBL 데이터 조회(검색)프로시저	
	PROC_DEL_MENU	MENU_TBL 데이터 삭제 프로시저	
PKG_SALE	PROC_INS_SALE	SALE_TBL(주문 테이블)데이터 추가 프로시저	박지원
	PROC_REFUND_SALE	SALE_TBL 환불 수정 프로시저, PRICE, POINT, CUSTOMER TBL도 UPDATE	
PKG_EVENT	PROC_MOD_EVENT	EVENT_TBL(이벤트테이블) 데이터 추가 및 수정	박지원
	PROC_SEL_EVENT	EVENT_TBL 데이터 조회	
	PROC_DEL_EVENT	EVENT_TBL 데이터 삭제	
PKG_PRICE	PROC_INS_PRICE	PRICE_TBL(가격테이블) 데이터 생성 프로시저	박지원
	PROC_SEL_PRICE	PRICE_TBL 조회 프로시저	
	PROC_DEL_PRICE	PRICE_TBL 삭제 프로시저	
PKG_POINT_TBL	PROC_POINT_TBL_INSERT	POINT_TBL(포인트 테이블) 데이터 추가 프로시저	

패키지 정의



업무프로시저			작성자
PKG_RANK	PROC_MENU_RNK	많이 팔린 메뉴 순위 조회 프로시저	하경훈
	PROC_SEL_REFUND_MN	메뉴마다 환불된 수량과 순위 조회	박지원
	PROC_BUY_REGION_RNK	주문을 많이 한 지역 순위 조회 프로시저	하경훈
	PROC_SALE_BY_TIME	시간대별 매출액 순위 조회 프로시저	하경훈
	PROC_SALE_PERHOUR	시간대별 주문 횟수와 메뉴량 조회	하경훈
PKG_EMP_MANAGE	PROC_SEL_LATE_ATT	직원의 지각 조회(0:요약X , 1:요약)	박지원
	PROC_SEL_ABS_ATT	직원의 결근 조회(0:요약X , 1:요약)	박지원
	PROC_GIVE_BONUS	월별 가장 매출을 많이 올린 직원에게 지급할 보너스 조회 프로시저	하경훈
	PROC_EMP_BY_BESTTIME	특정 날짜의 매출이 가장 높았던 시간대에 근무한 직원 조회 프로시저	하경훈
	PROC_SEL_EMP_WELLWORK	근무시간 중의 일 처리량 조회	하경훈
PKG_CUS_MANAGE	PROC_SEL_W_SALARY	주휴수당을 포함한 주급 계산 (직원 아이디, 직원 이름, 몇주차로 검색 가능)	박지원
	PROC_BEST_MENU_BY_CUS	각 고객들이 가장 많이 주문한 메뉴 조회 프로시저	하경훈
	PROC_SEL_DIS_PRICE	고객들의 할인 받은 금액 조회	박지원
	PROC_SEL_REFUND_PERCUS	고객의 환불 횟수 조회	박지원
	PROC_REFUND_CNT	환불 건수에 따른 고객별 블랙리스트 점수	이진영
PKG_COMPANY_MANAGE	PROC_REVENUE	총 수익과 영업 순이익	이진영
	PROC_ORDER_PART	주문 구분에 따른 매출	이진영
	PROC_STOCK_MNG	재료 구매와 판매에 따른 재고관리	이진영
PKG_ETC	PROC_STOCK_POINT	지역별 쌓은 포인트	이진영

업무 PROCEDURE

PROC_SEL_LATE_ATT (직원의 지각 조회 프로시저) 작성자 : 박지원

--6. 직원의 지각 조회

PROCEDURE PROC_SEL_LATE_ATT

(

IN_EMP_ID IN VARCHAR2,

IN_EMP_NAME IN VARCHAR2,

IN_CHECK_SUMMARY IN NUMBER, --1: 요약 0:전체

O_RST OUT SYS_REFCURSOR

)

AS

BEGIN

-- 요약 선택시의 SELECT

IF IN_CHECK_SUMMARY = 1 THEN

OPEN O_RST FOR

SELECT T1.EMP_ID, T2.EMP_NAME, T2.LATE_CNT, T1.TLT_CNT, ROUND((T2.LATE_CNT/T1.TLT_CNT),2) AS RATE_PCT -- 지각횟수 / 출근 횟수 = 지각률

FROM (

--WORK_TBL 에서 출근한 날만 SELECT

SELECT EMP_ID, COUNT(EMP_ID) AS TLT_CNT

FROM WORK_TBL --TIMETABLE_TBL

WHERE WORK_IO = 'I'

GROUP BY EMP_ID

) T1,

(

--출근한 날중 지각한 날을 집계함수를 활용해 SELECT

SELECT T1.EMP_ID AS EMP_ID, T3.EMP_NAME AS EMP_NAME, COUNT(T2.EMP_ID) AS LATE_CNT

FROM TIMETABLE_TBL T1, WORK_TBL T2, EMPLOYEE_TBL T3

WHERE T1.EMP_ID = T2.EMP_ID(+)

AND T1.WORK_DATE = T2.WORK_DATE(+)

AND T1.EMP_ID(+) = T3.EMP_ID

AND T2.WORK_IO = 'I'

AND TO_DATE(T1.IN_TIME, 'HH24:MI') <= TO_DATE(T2.WORK_TIME, 'HH24:MI')

GROUP BY T1.EMP_ID, T3.EMP_NAME

) T2

WHERE T1.EMP_ID = T2.EMP_ID

AND T2.EMP_ID LIKE '%'||IN_EMP_ID||'%'

AND T2.EMP_NAME LIKE '%'||IN_EMP_NAME||'%'

;

-- 지각 횟수

업무 PROCEDURE

PROC_SEL_LATE_ATT (직원의 지각 조회 프로시저) 작성자 : 박지원

```
ELSIF IN_CHECK_SUMMARY = 0 THEN
    OPEN O_RST FOR
    SELECT T1.EMP_ID,T3.EMP_NAME, T1.WORK_DATE, T1.IN_TIME, T2.WORK_TIME, T2.WORK_IO
    FROM TIMETABLE_TBL T1, WORK_TBL T2, EMPLOYEE_TBL T3
    WHERE T1.EMP_ID = T2.EMP_ID(+)
    AND T1.WORK_DATE = T2.WORK_DATE(+)
    AND T1.EMP_ID(+) = T3.EMP_ID
    AND T2.WORK_IO = 'I'
    AND TO_DATE(T1.IN_TIME,'HH24:MI') <= TO_DATE(T2.WORK_TIME, 'HH24:MI')
    AND T1.EMP_ID LIKE '%'||IN_EMP_ID||'%'
    AND T3.EMP_NAME LIKE '%'||IN_EMP_NAME||'%'
    ;
END IF;
END PROC_SEL_LATE_ATT
;
```

-- 요약이 아니면 전체 지각한 날과 지각한 시각을 나타냄

-- 지각을 하지않아도 표시

-- IN_EMP_ID, IN_EMP_NAME으로 검색 가능

업무 PROCEDURE

PROC_SEL_DIS_PRICE (고객들의 할인 받은 금액 조회) 작성자: 박지원

--고객중 할인을 많이 받은 고객을 SELECT

PROCEDURE PROC_SEL_DIS_PRICE

```
(
    IN_CUS_ID      IN      VARCHAR2,
    IN_CUS_NAME    IN      VARCHAR2,
    O_RST          OUT     SYS_REFCURSOR
)
```

AS

BEGIN

OPEN O_RST FOR

```
SELECT T3.CUS_ID,T3.CUS_NAME ,NVL(SUM(T2.SALE_PRICE),-1) AS DIS_PRICE
      , RANK()OVER(ORDER BY NVL(SUM(T2.SALE_PRICE),-1) DESC) AS DIS_RNK --
```

SUM(할인된 금액), -1은 구매내역 없음

FROM

```
(
    SELECT ORD_ID, CUS_ID
    FROM SALE_TBL
    GROUP BY ORD_ID, CUS_ID
)
```

) T1,

```
(
    SELECT ORD_ID, EVENT_ID, DIS_COM_ID, USE_POINT, ORD_PRICE
           , PAY_PRICE, ORD_PRICE - (PAY_PRICE + USE_POINT) AS SALE_PRICE --
```

주문금액 - (결제된 금액 + 포인트 사용금액) = 할인된 금액

FROM PRICE_TBL

)T2

,CUSTOMER_TBL T3

WHERE T1.ORD_ID = T2.ORD_ID(+)

AND T3.CUS_ID = T1.CUS_ID(+)

AND T3.CUS_ID LIKE '%'||IN_CUS_ID||'%'

AND T3.CUS_NAME LIKE '%'||IN_CUS_NAME||'%'

GROUP BY T3.CUS_ID,T3.CUS_NAME

;

END PROC_SEL_DIS_PRICE

;

-- IN_CUS_ID, IN_CUS_NAME로 검색

업무 PROCEDURE

PROC_SEL_W_SALARY (주휴수당을 포함한 주급 계산) 작성자 : 박지원

--주휴수당을 포함한 주급 계산

PROCEDURE PROC_SEL_W_SALARY

```
(
    IN_EMP_ID      IN      VARCHAR2,
    IN_EMP_NAME    IN      VARCHAR2,
    IN_WEEK_NUM    IN      NUMBER,
    O_RST          OUT     SYS_REFCURSOR
)
AS
BEGIN
    OPEN O_RST FOR
    SELECT A.EMP_ID, A.EMP_NAME, B.WEEK_NUM
        , CASE WHEN B.WORK_HOUR >=15 THEN (B.WORK_HOUR + ROUND(B.WORK_HOUR/B.WEEK_WORK_CNT)) *A.EMP_SALARY
            -- WHEN T2.WORK_HOUT < 15 THEN (T2.WORK_HOUR * T1.EMP_SALARY)
            ELSE B.WORK_HOUR * A.EMP_SALARY
        END AS WEEK_SALARY      --15시간 넘어가면 전체 일한시간의 일한 날을 나뉘서 주휴수당으로 지급
        ,CASE WHEN B.WORK_HOUR >=15 THEN 'Y' ELSE 'N' END AS BONUS_YN
    FROM (
        SELECT EMP_ID, EMP_NAME, EMP_SALARY
        FROM EMPLOYEE_TBL
        WHERE EMP_LEAVE_DATE IS NULL      --아직 퇴사하지 않은 직원들
    ) A,
```

업무 PROCEDURE

PROC_SEL_W_SALARY (주휴수당을 포함한 주급 계산) 작성자 : 박지원

```
(  --출근, 퇴근 데이터 합쳐서 그주에 일했던 시간  SELECT
SELECT T1.EMP_ID,TO_CHAR(TO_DATE(T1.WORK_DATE, 'YYYY-MM-DD'), 'IW') AS WEEK_NUM,
      COUNT(T1.WORK_DATE) AS WEEK_WORK_CNT
      ,SUM( ROUND((TO_DATE(T2.WORK_TIME, 'HH24:MI')- TO_DATE(T1.WORK_TIME, 'HH24:MI')) * 24)) AS WORK_HOUR
FROM
(  --출근 데이터
  SELECT EMP_ID, WORK_DATE, WORK_TIME, WORK_IO
  FROM WORK_TBL
  WHERE WORK_IO = 'I'
)T1,
(  --퇴근데이터
  SELECT EMP_ID,WORK_DATE, WORK_TIME, WORK_IO
  FROM WORK_TBL
  WHERE WORK_IO = 'O'
)T2
--EMPLOYEE_TBL T3
WHERE T1.EMP_ID = T2.EMP_ID
AND T1.WORK_DATE = T2.WORK_DATE
GROUP BY T1.EMP_ID,TO_CHAR(TO_DATE(T1.WORK_DATE, 'YYYY-MM-DD'), 'IW')
)B
WHERE A.EMP_ID = B.EMP_ID(+)
AND A.EMP_ID LIKE '%'||IN_EMP_ID||'%'
AND A.EMP_NAME LIKE '%'||IN_EMP_NAME||'%'
AND B.WEEK_NUM LIKE '%'|| IN_WEEK_NUM ||'%'
;
END PROC_SEL_W_SALARY
;
```

업무 PROCEDURE

PROC_GIVE_BONUS (월별 가장 매출을 많이 올린 직원에게 지급할 보너스 조회 프로시저) 작성자 : 하경훈

```
create or replace NONEDITIONABLE PROCEDURE PROC_GIVE_BONUS
(
    IN_DATE          IN          VARCHAR2,      -- IN_DATE(YYYY-MM)
    O_CUR            OUT         SYS_REFCURSOR
)
AS
BEGIN

    OPEN O_CUR FOR
    SELECT S1.EMP_ID , S2.EMP_NAME ,S1.TLTP2 판매금액, S1.TLTQ2, S1.RNK, S1.TLTQ2*500 지급보너스
    FROM
    (
        SELECT EMP_ID, SUM(SALEPRICE) AS TLTP2 ,SUM(SALEQTY) TLTQ2 ,RANK() OVER(ORDER BY SUM(SALEPRICE) DESC) RNK
        FROM
        (
            SELECT SD, ST, CASE WHEN INTT<= ST AND ST < OUTT THEN
                TLTP
            ELSE 0
            END AS SALEPRICE
            ,B.EMP_ID,B.INTT,B.OUTT ,A.TLTQ,
            CASE WHEN INTT<= ST AND ST < OUTT THEN
                TLTQ
            ELSE 0
            END AS SALEQTY
        FROM
        (
            SELECT SD, ST,SUM(MENU_PRICE * MENU_QTY) TLTP, SUM(MENU_QTY) TLTQ
            FROM
            (
                SELECT T1.MENU_ID,T2.MENU_PRICE, T1.MENU_QTY, TO_CHAR(T1.ORD_DATE,'YYYY-MM-DD')SD,TO_CHAR(T1.ORD_DATE,'HH24') ST
                FROM SALE_TBL T1, MENU_TBL T2
                WHERE T1.MENU_ID = T2.MENU_ID
                AND T2.MENU_HC IS NOT NULL -- 굿즈 제외
                AND TO_CHAR(T1.ORD_DATE,'YYYY-MM') =IN_DATE
                AND REFUND_YN='N'
                ORDER BY T1.ORD_DATE
            )
            GROUP BY SD, ST
            ORDER BY SD, ST
        ) A, -- 날짜별 판매날짜, 판매 시간, 판매금액, 판매 수량
```

업무 PROCEDURE

PROC_GIVE_BONUS (월별 가장 매출을 많이 올린 직원에게 지급할 보너스 조회 프로시저) 작성자 : 하경훈

```
(
    SELECT T1.WORK_DATE,T1.EMP_ID
           ,TO_NUMBER(SUBSTR(T1.WORK_TIME,1,2))+ROUND(TO_NUMBER(SUBSTR(T1.WORK_TIME,4,2))/60*100,-2)/100 INTT
           ,TO_NUMBER(SUBSTR(T2.WORK_TIME,1,2))+ROUND(TO_NUMBER(SUBSTR(T2.WORK_TIME,4,2))/60*100,-2)/100 OUTT
    FROM (SELECT * FROM WORK_TBL WHERE WORK_IO ='I')T1,
         (SELECT * FROM WORK_TBL WHERE WORK_IO ='O')T2
    WHERE T1.WORK_DATE = T2.WORK_DATE AND T1.EMP_ID = T2.EMP_ID
    AND SUBSTR(T1.WORK_DATE,1,7) = IN_DATE
)B -- 직원별 근무 시간
WHERE A.SD = B.WORK_DATE
ORDER BY SD,ST,B.EMP_ID
)
GROUP BY EMP_ID
) S1, EMPLOYEE_TBL S2
WHERE S1.EMP_ID = S2.EMP_ID
AND RNK = 1
;
END PROC_GIVE_BONUS;
```


업무 PROCEDURE

PROC_EMP_BY_BESTTIME (특정 날짜의 매출이 가장 높았던 시간대에 근무한 직원 조회 프로시저) 작성자 : 하경훈

create or replace NONEDITIONABLE PROCEDURE PROC_EMP_BY_BESTTIME -- 지정한 날짜의 가장 매출이 높은 시간대에 근무한 직원 찾기

```
(
    IN_DATE      IN      VARCHAR2,      -- 'YYYY-MM-DD'
    O_CUR        OUT     SYS_REFCURSOR
)
AS

BEGIN

    OPEN O_CUR FOR
    SELECT IN_DATE,S1.TIME1||':00 ~ '||TO_NUMBER(S1.TIME1+1)||':00',TO_CHAR(S1.TLT,'999,999') 매출액,
           S2.EMP_ID,S2.EMP_NAME, S2.INT||':00 ~ '||S2.OUTT||':00' AS 근무시간
    FROM
    (
        SELECT IN_DATE,A.TIME1 , NVL(B.TLTP,0)AS TLT , B.RNK
        FROM (SELECT IN_DATE,10+LEVEL-1 AS TIME1 FROM DUAL CONNECT BY LEVEL<=12) A,
        (
            SELECT TO_CHAR(ORD_DATE,'YYYY-MM-DD')AS SDATE ,TO_CHAR(ORD_DATE,'HH24') AS TIME2,SUM(PAY_PRICE) TLTP,
                   RANK() OVER(ORDER BY SUM(PAY_PRICE) DESC) RNK
            FROM
            (
                SELECT T1.ORD_ID,T2.PAY_PRICE, T1.ORD_DATE
                FROM SALE_TBL T1, PRICE_TBL T2
                WHERE T1.ORD_ID = T2.ORD_ID
                AND T1.REFUND_YN='N'
                AND TO_CHAR(T1.ORD_DATE,'YYYY-MM-DD') =IN_DATE
                GROUP BY T1.ORD_ID,T2.PAY_PRICE, T1.ORD_DATE
                ORDER BY T1.ORD_DATE
            )
            GROUP BY TO_CHAR(ORD_DATE,'YYYY-MM-DD') ,TO_CHAR(ORD_DATE,'HH24')
        ) B
        WHERE A.TIME1 = B. TIME2(+)
        ORDER BY A.TIME1
    )S1,
```

업무 PROCEDURE

PROC_EMP_BY_BESTTIME (특정 날짜의 매출이 가장 높았던 시간대에 근무한 직원 조회 프로시저) 작성자 : 하경훈

```
(
SELECT T1.EMP_ID, T2.EMP_NAME , T1.WORK_DATE ,INT,OUTT
FROM
(
SELECT A.EMP_ID, A.WORK_DATE, TO_NUMBER(SUBSTR(A.WORK_TIME,1,2))+ROUND(TO_NUMBER(SUBSTR(A.WORK_TIME,4,2))/60*100,-2)/100 INT,
TO_NUMBER(SUBSTR(B.WORK_TIME,1,2))+ROUND(TO_NUMBER(SUBSTR(B.WORK_TIME,4,2))/60*100,-2)/100 OUTT

FROM
(SELECT * FROM WORK_TBL WHERE WORK_IO='I')A,(SELECT * FROM WORK_TBL WHERE WORK_IO ='O')B
WHERE A.WORK_DATE = B.WORK_DATE AND A.EMP_ID = B.EMP_ID
AND A.WORK_DATE =IN_DATE
ORDER BY A.WORK_DATE ,A.EMP_ID
) T1, EMPLOYEE_TBL T2
WHERE T1.EMP_ID = T2.EMP_ID
)S2
WHERE S1.RNK =1
AND S1.TIME1 >= S2.INT AND S1.TIME1 < S2.OUTT
;

END FIND_EMPLOYEE_BY_BESTTIME;
```

업무 PROCEDURE

PROC_BEST_MENU_BY_CUS (각 고객이 가장 많이 주문한 메뉴 조회 프로시저) 작성자 : 하경훈

```
create or replace NONEDITIONABLE PROCEDURE PROC_BEST_MENU_BY_CUS-- 고객별 선호 메뉴 ( 굿즈 제외 )
(
    O_CUR          OUT          SYS_REFCURSOR
)
AS

BEGIN
    OPEN O_CUR FOR
    SELECT B.CUS_ID 고객ID, B.CUS_NAME 고객명, A.MENU_ID ,C.MENU_NAME 선호메뉴
    FROM
    (
        SELECT *
        FROM
        (
            SELECT T1.CUS_ID, T1.MENU_ID, COUNT(*) AS QTY , ROW_NUMBER() OVER(PARTITION BY CUS_ID ORDER BY COUNT(*) DESC) RNK
            FROM SALE_TBL T1, MENU_TBL T2
            WHERE T1.MENU_ID = T2.MENU_ID
            AND T1.REFUND_YN ='N'
            AND T2.MENU_HC IS NOT NULL -- 굿즈 제외
            GROUP BY T1.CUS_ID, T1.MENU_ID
        )
        WHERE RNK = 1
    )A, CUSTOMER_TBL B, MENU_TBL C
    WHERE A.CUS_ID(+) = B.CUS_ID AND A.MENU_ID = C.MENU_ID(+)
    ORDER BY A.CUS_ID
    ;
END PROC_BEST_MENU_BY_CUS;
```

업무 PROCEDURE

PROC_ORDER_PART (주문 구분에 따른 매출) 작성자 : 이진영

```
create or replace NONEDITIONABLE PROCEDURE PROC_ORDER_PART -- 판매구분에 따른 매출
(
    O_CUR          OUT          SYS_REFCURSOR
)

AS
BEGIN

    OPEN O_CUR FOR

    SELECT TT1.COM_VAL, TO_CHAR(SUM(TT2.PAY_PRICE), '999,999,999')
    FROM
    (
        SELECT T2.ORD_ID, T2.ORD_PART, T1.COM_VAL, SUM(T2.MENU_QTY)
        FROM COMMONS_TBL T1, SALE_TBL T2
        WHERE T1.COM_ID = T2.ORD_PART
        GROUP BY T2.ORD_ID, T2.ORD_PART, T1.COM_VAL
    )TT1, -- 주문번호별 판매구분
    PRICE_TBL TT2 -- 가격 테이블
    WHERE TT1.ORD_ID = TT2.ORD_ID
    GROUP BY TT1.COM_VAL
;
END PROC_ORDER_PART;
```

업무 PROCEDURE

PROC_STOCK_MNG (재료 구매와 판매에 따른 재고관리) 작성자 : 이진영

```
create or replace NONEDITIONABLE PROCEDURE PROC_STOCK_MNG
(
    O_CUR          OUT    SYS_REFCURSOR
)
AS
BEGIN

    OPEN O_CUR FOR
    SELECT TBL1.MAT_ID, TBL1.MAT_NAME, SUM((TBL1.TLT_BUY - TBL2.SALE_QTY) * 0.001) MATERIAL
    FROM
    (
        SELECT T1.MAT_ID, T1.MAT_NAME, SUM(1000 * T1.MAT_CONTENTS * T2.MAT_QTY) TLT_BUY
        FROM MATERIAL_TBL T1, BUY_TBL T2
        WHERE T1.MAT_ID = T2.MAT_ID
        GROUP BY T1.MAT_ID, T1.MAT_NAME
        ORDER BY MAT_ID
    )TBL1, -- 구매한 재료 수량
    (
        SELECT B.MAT_ID, SUM(SALE_CNT * B.MAT_QTY) SALE_QTY
        FROM
        (
            SELECT MENU_ID, COUNT(MENU_ID) SALE_CNT
            FROM SALE_TBL
            GROUP BY MENU_ID
        )A,
        RECIPE_TBL B -- 메뉴별 필요한 재료 수량
        WHERE A.MENU_ID = B.MENU_ID
        GROUP BY B.MAT_ID
        ORDER BY MAT_ID
    )TBL2 -- 판매한 재료 수량
    WHERE TBL1.MAT_ID = TBL2.MAT_ID
    GROUP BY TBL1.MAT_ID, TBL1.MAT_NAME
    ORDER BY MAT_ID
    ;

END PROC_STOCK_MNG;
```

업무 PROCEDURE

PROC_REVENUE (총 수익과 영업이익) 작성자 : 이진영

```
create or replace NONEDITIONABLE PROCEDURE PROC_REVENUE
```

```
(  
    O_CUR    OUT    SYS_REFCURSOR  
)
```

```
AS
```

```
V_TLT_ORD_PRICE    NUMBER(7);  
V_TLT_COST          NUMBER(7);  
V_TLT_MAT           NUMBER(7);  
V_SAL_TLT           NUMBER(7);
```

```
BEGIN
```

```
-- 총 수입
```

```
SELECT SUM(ORD_PRICE) TLT_ORD_PRICE  
INTO V_TLT_ORD_PRICE  
FROM PRICE_TBL  
;
```

```
-- 경비사용
```

```
SELECT SUM(TLT_COST)  
INTO V_TLT_COST  
FROM  
(  
    SELECT COST_ID, SUM(COST_PRICE * COST_QTY) TLT_COST  
    FROM COST_TBL  
    GROUP BY COST_ID  
)  
;
```


업무 PROCEDURE

PROC_REVENUE (총 수익과 영업이익)

```
-- 재료구매
SELECT SUM(TLT_MAT)
INTO V_TLT_MAT
FROM
(
    SELECT MAT_ID, SUM(MAT_PRICE * MAT_QTY) TLT_MAT
    FROM BUY_TBL
    GROUP BY MAT_ID
)
;

-- 급여
SELECT SUM(SAL_TLT)
INTO V_SAL_TLT
FROM SALARY_TBL
;

OPEN O_CUR FOR
SELECT TO_CHAR(SUM(V_TLT_ORD_PRICE - (V_TLT_COST + V_TLT_MAT + V_SAL_TLT)), '999,999,999') TLT_REVENUR
FROM DUAL
;

END PROC_REVENUE;
```