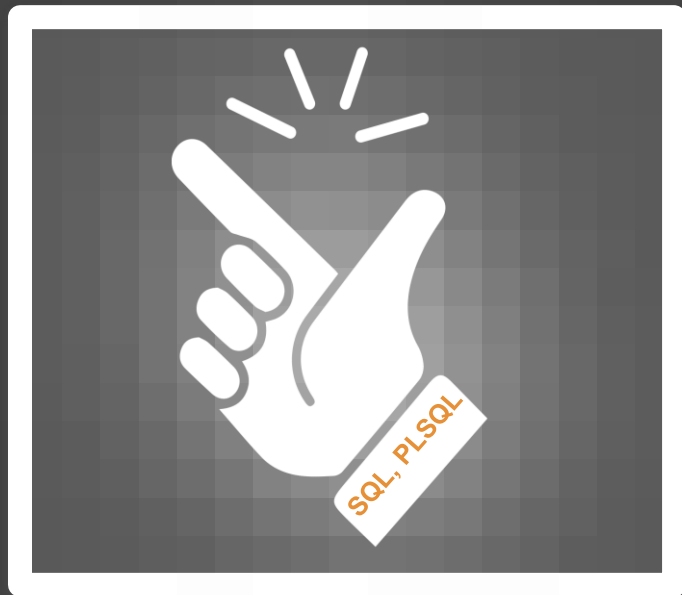


[ 행사관리가이렇게쉽조 ]



**EASY 행사관리**

발표 : 2023年 12月 6日

# T EXT

01 Purpose  
개발 동기

02 Team member & co-work  
조원소개 및 역할 분담

03 Date of project  
프로젝트 진행일정

04 System flow on event management  
거래 및 행사관리 시스템 흐름

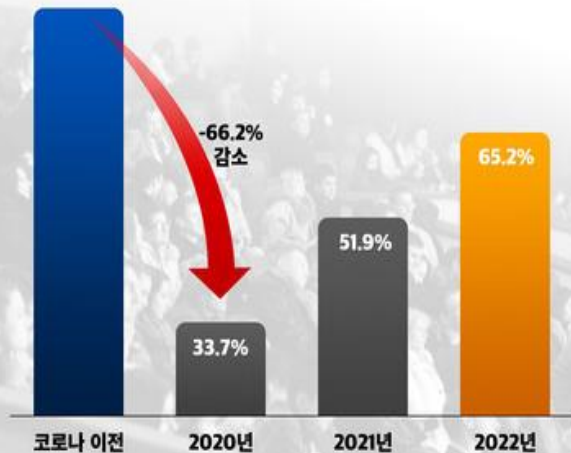
05 Consult of ER win program  
설계구현 ER Diagram

06 Descript to SQL & PLSQL  
SQL, PL/SQL 기능소개

# 1. 개발 동기

행사운영 실무 경험자의 현실적 Data-Base 운영관리 필요성 착안

## 코로나19 이후 오프라인 모임·행사 분석결과

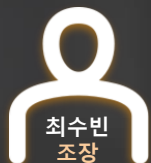


• 온오프믹스 이벤트데이터센터 제공(전국 주요 컨벤션 18개소 및 온오프믹스 행사 데이터 18만개)

- 행사의 효율적 운영관리 프로그램 **필요성**
- 현업 근무자의 고객사, 거래처 정보관리 **효율성**
- 거래업체 견적관리, 계약관리, 정산관리 **편의성**
- 현업의 계약서, 블랙리스트 **관리성**



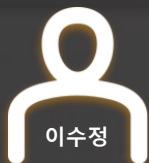
## 2. 조원소개 및 역할 분담



최수빈  
조장



장경희



이수정



이윤범

<b>최수빈</b> Team leader Soo	인력 거래처 건적 및 결제데이터 입력&풀이 직원 및 계약, 결제 관련 데이터 입력&풀이	
<b>이수정</b> Crystal	렌탈 거래처 건적 및 결제데이터 입력&풀이	
<b>이윤범</b> Tigre	행사기관 데이터 입력&풀이	
<b>장경희</b> J kiki	행사정보 데이터 입력&풀이	

※ SQL, PLSQL 구현과 보고서 작성은 조원들의 의견을 수렴해 모두 함께 진행하였습니다.

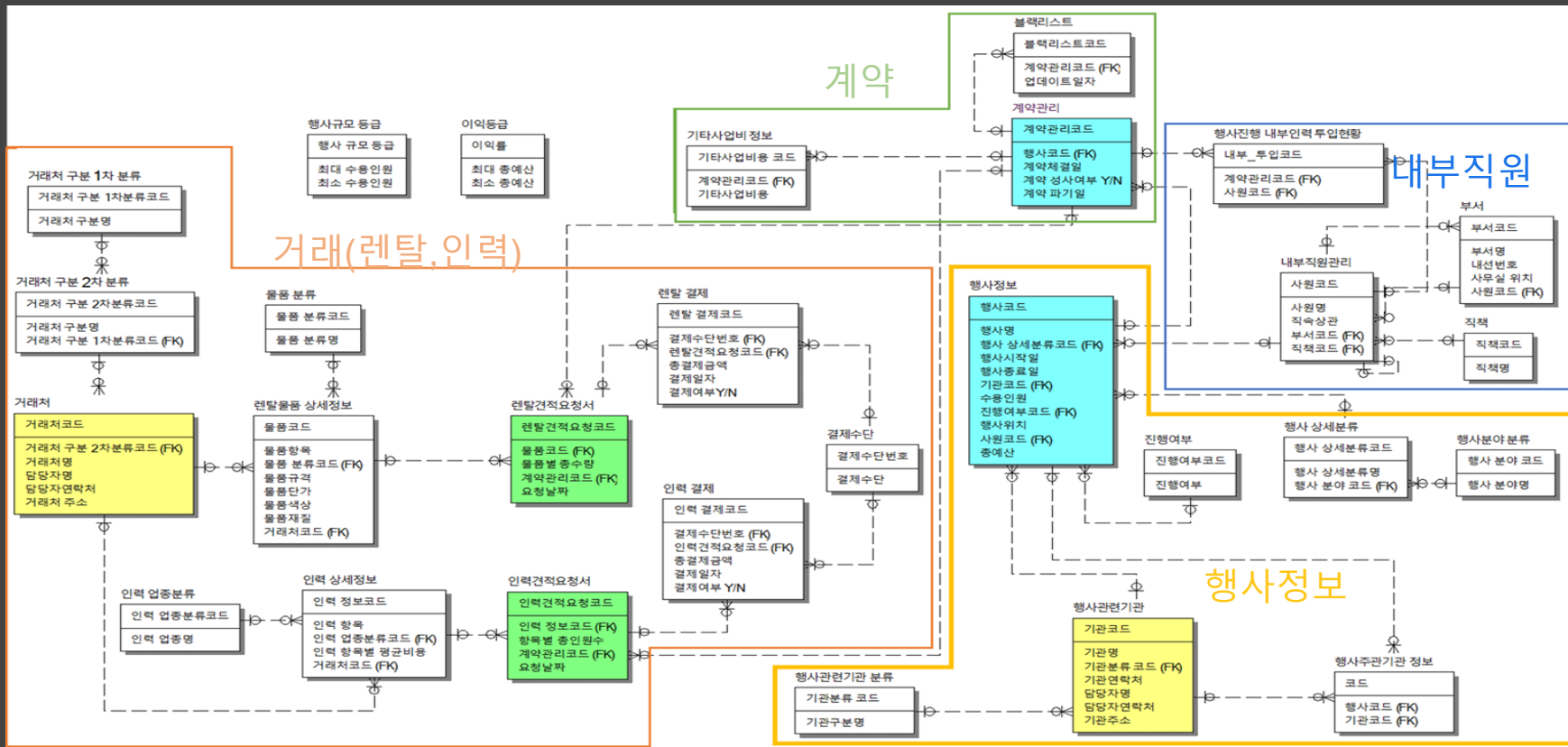
### 3. 프로젝트 진행일정

Sun	Mon	Tue	Wed	Thu	Fri	Sat
11.19	11.20	11.21	11.22	11.23	11.24	11.25
			프로젝트 주제선정 및 검토	설계도 제작 ERwin		SQL, PLSQL 문항 작성
11.26	11.27	11.28	11.29	11.30	12.1	12.2
SQL, PLSQL 문항 작성	데이터 입력		SQL 구현		PLSQL 구현	
12.3	12.4	12.5				
PLSQL 구현	보고서 및 PPT 제작					

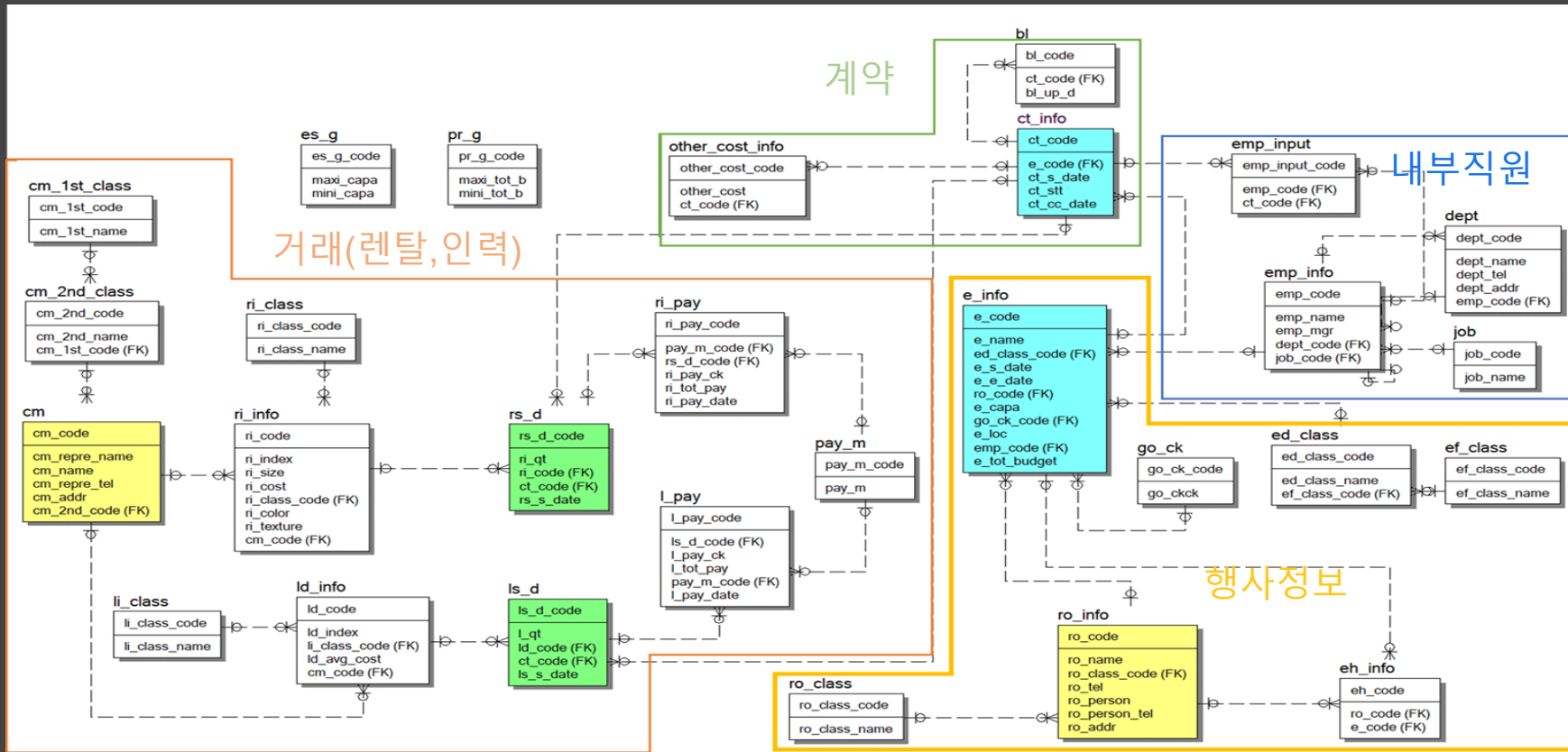
※ 프로젝트 기간: 총 2주, 약14일 소요



# 5. 설계구현 - ER Diagram(logical)



# 5. 설계구현 - ER Diagram(physical)





# SQL, PL SQL 기능 소개



## 1. 행사 관리

- 행사정보 조회
- 행사정보 변경



## 2. 기관 관리

- 기관정보 조회
- 기관등급 관리



## 3. 계약 관리

- 계약정보 변경



## 4. 거래처 관리

- 거래처 정보 조회



## 5. 견적 및 결제관리

- 결제정보 조회
- 견적 및 결제정보 변경



## 6. 회계관리

- 행사 회계정보 조회

# 1. 행사관리 - 행사정보 조회

SQL

Q. 최근 3개월 내 진행한 행사의 행사정보와 행사규모등급 및 내부투입인원수 조회

```
SQL> select a.e_name "행사명", to_char(a.e_s_date, 'yyyy/mm') "행사년월", a.e_tot_budget "행사총예산",  
        (select b.es_g_code from es_g b where a.e_capa<=b.maxi_capa and a.e_capa>=b.mini_capa) "행사규모등급",  
        count(d.ct_code)||'명' "내부투입인원수"  
from e_info a, ct_info c, emp_input d  
where a.e_code=c.e_code and c.ct_code=d.ct_code  
      and a.e_s_date>=add_months(sysdate,-3)  
      and a.e_s_date<=sysdate and d.emp_code is not null  
group by a.e_name, to_char(a.e_s_date, 'yyyy/mm'), a.e_tot_budget, a.e_capa, d.ct_code  
order by 2;
```

행사규모 등급	최대 수용인원	최소 수용인원
E	400	1
D	1,900	401
C	4,000	1901
B	5,500	4,001
A	20,000	5,501

# 1. 행사관리 - 행사정보 조회

SQL

Q. 최근 3개월 내 진행한 행사의 행사정보와 행사규모등급 및 내부투입인원수 조회

## 결과 화면

```
name "행사명", to_char(a.e_s_date, 'yyyy/mm') "행사년월", a.e_tot_budget "행사총예산",
```

행사명	행사년월	행사총예산	행사규모등급	내부투입인원수
2023 GXG 게임문화축제	2023/09	120000000	C	6명
2023 서울스트리트댄스페스티벌	2023/09	198000000	A	8명
2023 피맥무비페스티벌 영화제	2023/09	180000000	B	6명
2023 궁디팡팡 컷페스타	2023/10	165000000	B	7명
2023 빅피플 페스티벌	2023/10	142000000	A	7명
2023 사이버위협 대응 인공지능 보안 컨퍼런스	2023/10	125000000	D	4명
2023 월드 챔피언십	2023/10	185000000	A	8명
2023 정신건강축제 - 내 마음을 가꾸는 정원	2023/10	130000000	C	4명
2023 국제게임전시회 G-STAR	2023/11	150000000	B	7명
2023 서울무용제	2023/11	134000000	B	4명
2023 서울소울페스티벌	2023/11	180000000	C	8명
2023 우리 댕댕이도 981레이서	2023/11	123000000	B	4명
2023 휴애리 동백 축제	2023/11	85000000	C	4명

13 개의 행이 선택되었습니다.

D	1,900	401
C	4,000	1901
B	5,500	4,001
A	20,000	5,501

# 1. 행사관리 - 행사정보 변경

PLSQL

Q. 현재 보류 상태인 행사에 대해 총예산이 입력값(8000만원) 이하인 행사는 진행 취소로 변경하는 커서

```
SQL> accept budget prompt '총예산을 입력하세요 :'  
SQL> declare  
    v_budget e_info.e_tot_budget%type;  
    v_gc_code_ing go_ck.go_ck_code%type;  
    v_gc_code_end go_ck.go_ck_code%type;  
cursor gckcur is  
    select go_ck_code, e_tot_budget from e_info where go_ck_code=v_gc_code_ing;  
begin  
    v_budget:=&budget;  
    select go_ck_code into v_gc_code_ing from go_ck where go_ckck='보류';  
    select go_ck_code into v_gc_code_end from go_ck where go_ckck='진행취소';  
    for s in gckcur loop  
        update e_info set go_ck_code=v_gc_code_end where e_tot_budget<=v_budget;  
    end loop;  
end;  
/
```

# 1. 행사정보 변경

## 변경 전 Data

E_CODE	E_NAME	E_TOT_BUDGET	GO_CK_CODE	GO_CKCK
10041	2024 한국정보과학회 학술대회	154000000	205	보류
10082	2024 청년 정신건강 인식개선 캠페인	96000000	205	보류
10108	2023 기록교 자선행사	80000000	205	보류
10008	2024 대구약령시한방문화축제	65000000	205	보류
10051	2024 전국 테크노파크 워크숍 및 지역기업지원 성과발표회	60000000	205	보류
10049	2024 주소선박 기술포럼	31000000	205	보류
10045	2024 부동산 투자 세미나	25000000	205	보류
10054	2023 학술 컨퍼런스	25000000	205	보류
10059	2024 제3회 HMG 개발자 교육 프로그램	21000000	205	보류
10043	2024 한국생물공학회 추계학술대회	15000000	205	보류

10 개의 행이 선택되었습니다.

## 변경 후 Data

SQL> 04

총예산을 입력하세요 : 80000000

PL/SQL 처리가 정상적으로 완료되었습니다.

```
SQL> select a.e_code, a.e_name, e_tot_budget, a.go_ck_code, b.go_ckck
2 from e_info a, go_ck b
3 where a.go_ck_code=b.go_ck_code and e_code in (10041, 10082, 10108, 10008, 10051, 10049, 10054, 10045, 10059, 10043)
4 order by 3 desc;
```

E_CODE	E_NAME	E_TOT_BUDGET	GO_CK_CODE	GO_CKCK
10041	2024 한국정보과학회 학술대회	154000000	205	보류
10082	2024 청년 정신건강 인식개선 캠페인	96000000	205	보류
10108	2023 기록교 자선행사	80000000	203	보류취소
10008	2024 대구약령시한방문화축제	65000000	203	보류취소
10051	2024 전국 테크노파크 워크숍 및 지역기업지원 성과발표회	60000000	203	보류취소
10049	2024 주소선박 기술포럼	31000000	203	보류취소
10045	2024 부동산 투자 세미나	25000000	203	보류취소
10054	2023 학술 컨퍼런스	25000000	203	보류취소
10059	2024 제3회 HMG 개발자 교육 프로그램	21000000	203	보류취소
10043	2024 한국생물공학회 추계학술대회	15000000	203	보류취소

10 개의 행이 선택되었습니다.

## 2. 기관관리 - 기관정보 조회

SQL

Q. 진행 여부가 '보류'인 행사 중 블랙리스트에 존재하는 주최 기관이 주최하는 행사 정보 조회

```
SQL> select e.e_code "행사코드", e.e_name "행사명", r.ro_name "주최기관", e.e_tot_budget "총예산",  
           e.e_s_date "행사시작일", e.e_e_date "행사종료일"  
from e_info e, go_ck g, ro_info r  
where g.go_ck_code=e.go_ck_code  
      and r.ro_code=e.ro_code  
      and g.go_ckck='보류' and r.ro_name in (select r.ro_name  
                                             from bl b, ct_info c, e_info e, ro_info r  
                                             where c.ct_code=b.ct_code  
                                                    and e.e_code=c.e_code  
                                                    and r.ro_code=e.ro_code);
```

블랙리스트 : 계약을 일방적으로 파기한 계약 및 기관을 별도로 관리하는 테이블

## 2. 기관관리 - 기관정보 조회

SQL

Q. 진행 여부가 '보류'인 행사 중 블랙리스트에 존재하는 주최 기관이 주최하는 행사 정보 조회

```
SQL> select e.e_code "행사코드", e.e_name "행사명", r.ro_name "주최기관", e.e_tot_budget "총예산",  
        e.e_start_date "행사시작일", e.e_e_date "행사종료일"  
from e_info e, go_ckpt, ro_info r
```

블랙리스트 테이블

블랙리스트코드	계약코드	행사명	기관	블랙리스트등록일
40001	13063	2023 삼충음성유방암 의학세미나	제주대학교병원	23/08/15
40002	13070	2023 환경아 놀자	대구광역시	23/09/20
40003	13073	2023 국내 백신개발 사업단 합동 학술대회	국내백신개발 사업단	23/10/02

```
from bl b, ct_info c, e_info e, ro_info r
```

해당 주최기관 행사정보

```
where c.ct_code=b.ct_code  
and e.e_code=c.e_code
```

행사코드	행사명	주최기관	총예산	행사시작일	행사종료일
10008	2024 대구약령시한방문화축제	대구광역시	65000000	24/05/05	24/05/07

## 2. 기관관리 - 기관등급 조회

PLSQL

Q. 주최기관별 행사개최 횟수에 따른 기관 VIP 등급(A등급~D등급)을 출력하는 함수

```
SQL> create or replace function f_vip (code number) return varchar2
is
    v_event_count number;
begin
    select count(*) into v_event_count
    from e_info e, go_ck g
    where e.go_ck_code=g.go_ck_code and go_ckck in ('진행중', '종료') and ro_code=code;
    return case when v_event_count>=0 and v_event_count<3 then 'D'
    when v_event_count>=3 and v_event_count<5 then 'C'
    when v_event_count>=5 and v_event_count<8 then 'B'
    else 'A' end;
end;
/
```



## 2. 기관관리 - 기관등급 조회

PLSQL

Q. 주최기관별 행사개최 횟수에 따른 기관 VIP 등급(A등급~D등급)을 출력하는 함수

### 결과 화면

```
create function f_vip (code number) return varchar2
```

함수가 생성되었습니다.

```
SQL> select * from (select ro_code, ro_name, f_vip(ro_code) from ro_info order by 3) where rownum<=10;
```

RO_CODE	RO_NAME	F_VIP(RO_CODE)
11001	서울특별시	A
11082	부산광역시	B
11053	서울문화재단	C
11072	대구광역시	C
11002	서울시환경운동연합	D
11006	문화체육관광부	D
11007	서울시체육회	D
11008	한국애견연맹	D
11009	서울아트센터	D
11010	명동성당	D

10 개의 행이 선택되었습니다.

### 3. 계약관리 - 계약정보 변경

PLSQL

Q1. 행사 시작 2주 전, 계약을 파기한 계약 건을 블랙리스트로 등록하는 트리거

```
SQL> create or replace trigger t_bl
      before update of ct_stt on ct_info
      for each row
      declare
        v_ckcode e_info.go_ck_code%type;
        v_date e_info.e_s_date%type;
      begin
        select go_ck_code, e_s_date into v_ckcode, v_date
        from e_info where e_code=:new.e_code;
        if :new.ct_stt='N' and :old.ct_stt='Y' and v_ckcode!=205
           and sysdate<=v_date and sysdate>=(v_date-14) then
          insert into bl(bl_code, ct_code, bl_up_d)
          values(seq_bl.nextval, :new.ct_code, sysdate);
        end if;
      end;
      /
```

Q2. 계약 성사 여부가 N으로 변경 시 계약파기일을 오늘로 변경, 행사 진행 상태는 '취소'로 변경하는 트리거

```
SQL> create or replace trigger t_event
      before update of ct_stt on ct_info
      for each row
      begin
        if :new.ct_stt='N' and :old.ct_stt='Y' then
          :new.ct_cc_date:=sysdate;
          update e_info set go_ck_code=203
          where e_code=:new.e_code;
        end if;
      end;
      /
```

### 3. 계약관리 - 계약정보 변경

PLSQL

Q1. 행사 시작 2주 전, 계약을 파기한 계약 건을 블랙리스트로 등록하는

Q2. 계약 성사 여부가 N으로 변경 시 계약파기일을 오늘로 변경, 행사 진행 상태는 '취소'로 변경하는 트리거

#### 변경 전 Data

CT_CODE	E_CODE	E_NAME	CT_S_DAT	CT_CC_DATE	CT_STT	GO_CK_CODE	GO_CKCK
13080	10100	2023 왕궁 수문장 교대의식	23/11/13		Y	201	진행전

#### 변경 후 Data

트리거가 생성되었습니다.

```
SQL> update ct_info set ct_stt='N' where ct_code=13080;
```

1 행이 갱신되었습니다.

```
SQL> select ci.ct_code, ei.e_code, ei.e_name, ci.ct_s_date, ci.ct_cc_date, ci.ct_stt, gc.go_ck_code, gc.go_ckck
2 from ct_info ci, e_info ei, go_ck gc
3 where ei.e_code=ci.e_code and gc.go_ck_code=ei.go_ck_code and ci.ct_code=13080;
```

CT_CODE	E_CODE	E_NAME	CT_S_DAT	CT_CC_DATE	CT_STT	GO_CK_CODE	GO_CKCK
13080	10100	2023 왕궁 수문장 교대의식	23/11/13	23/12/04	N	203	진행취소

```
SQL> select * from bl;
```

BL_CODE	CT_CODE	BL_UP_D
40001	13063	23/08/15
40002	13070	23/09/20
40003	13073	23/10/02
40004	13080	23/12/04

## 4. 거래처 관리 - 거래처정보 조회

SQL

Q. 결제 여부가 N인 결제건 중 2023/12/04 기준 건적요청일이 2개월 이상 지난 건에 대한 거래처 및 거래처담당자 정보, 건적요청일자 조회

```
SQL> select l.l_pay_ck "결제여부", ei.e_name "행사명", o.cm_1st_name "거래처구분", c.cm_name "거래처",
           c.cm_repre_name "담당자", c.cm_repre_tel "담당자전화번호", d.ls_s_date "건적요청일"
from l_pay l, ls_d d, ld_info i, cm c, cm_2nd_class t, cm_1st_class o, ct_info ci, e_info ei
where l_pay_ck='N'
      and d.ls_d_code=l.ls_d_code and i.ld_code=d.ld_code and c.cm_code=i.cm_code and t.cm_2nd_code=c.cm_2nd_code
      and o.cm_1st_code=t.cm_1st_code and ci.ct_code=d.ct_code
      and ei.e_code=ci.e_code and d.ls_s_date<add_months(sysdate,-2)
group by ei.e_name, c.cm_name, c.cm_repre_name, c.cm_repre_tel, d.ls_s_date, l.l_pay_ck, o.cm_1st_name
union all
select r.ri_pay_ck "결제여부", ei.e_name "행사명", o.cm_1st_name "거래처구분", c.cm_name "거래처",
           c.cm_repre_name "담당자", c.cm_repre_tel "담당자전화번호", rd.rs_s_date "건적요청일"
from ri_pay r, rs_d rd, ri_info ri, cm c, cm_2nd_class t, cm_1st_class o, ct_info ci, e_info ei
where ri_pay_ck='N'
      and rd.rs_d_code=r.rs_d_code and ri.ri_code=rd.ri_code and c.cm_code=ri.cm_code and t.cm_2nd_code=c.cm_2nd_code
      and o.cm_1st_code=t.cm_1st_code and ci.ct_code=rd.ct_code
      and ei.e_code=ci.e_code and rd.rs_s_date<add_months(sysdate,-2)
group by ei.e_name, c.cm_name, c.cm_repre_name, c.cm_repre_tel, rd.rs_s_date, r.ri_pay_ck, o.cm_1st_name
order by 2, 7;
```

## 4. 거래처 관리 - 거래처정보 조회

SQL

Q. 결제 여부가 N인 결제건 중 2023/12/04 기준 건적요청일이 2개월 이상 지난 건에 대한 거래처 및 거래처담당자 정보, 건적요청일자 조회

### 결과 화면

SQL: select \* from tcm where ei.e\_name "행사명", o.cm\_1st\_name "거래처구분", c.cm\_name "거래처",  
 "담당자", "담당자전화번호", "건적요청일자"

결제여부	행사명	거래처구분	거래처	담당자	담당자전화번호	건적요청일
N	2023 산악대행	인력	아이디어	고승태	010-2383-4796	23/08/02
N	2023 산악대행	인력	리드캐리	김도원	010-0841-8920	23/08/02
N	2023 산악대행	인력	알에이	한성준	010-2638-6355	23/08/18
N	2023 산악대행	인력	타인	도성민	010-0022-8321	23/08/18
N	2023 산악대행	인력	다오렌탈	양서유	010-0973-5889	23/08/18
N	2023 산악대행	인력	렌탈프로	정대하	010-3498-9337	23/08/18
N	2023 산악대행	인력	렌탈프로	유정	010-3405-5466	23/08/18
N	2023 산악대행	인력	렌탈프로	장태하	010-2962-8183	23/08/18
N	2023 산악대행	인력	렌탈프로	정대하	010-8134-3492	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-0014-9489	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-5246-2566	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-2716-2342	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-0217-6373	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-8462-4197	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-0841-8920	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-2383-4796	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-2962-8183	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-6116-3875	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-3498-9337	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-2638-6355	23/09/28
결제여부	행사명	거래처구분	거래처	담당자	담당자전화번호	건적요청일
N	2023 산악대행	인력	아이디어	고승태	010-2383-4796	23/08/02
N	2023 산악대행	인력	리드캐리	김도원	010-0841-8920	23/08/02
N	2023 산악대행	인력	알에이	한성준	010-2638-6355	23/08/18
N	2023 산악대행	인력	타인	도성민	010-0022-8321	23/08/18
N	2023 산악대행	인력	다오렌탈	양서유	010-0973-5889	23/08/18
N	2023 산악대행	인력	렌탈프로	정대하	010-3498-9337	23/08/18
N	2023 산악대행	인력	렌탈프로	유정	010-3405-5466	23/08/18
N	2023 산악대행	인력	렌탈프로	장태하	010-2962-8183	23/08/18
N	2023 산악대행	인력	렌탈프로	정대하	010-8134-3492	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-0014-9489	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-5246-2566	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-2716-2342	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-0217-6373	23/08/18
N	2023 산악대행	인력	렌탈프로	신상우	010-8462-4197	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-0841-8920	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-2383-4796	23/09/13
N	2023 산악대행	인력	렌탈프로	신상우	010-2962-8183	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-6116-3875	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-3498-9337	23/09/28
N	2023 산악대행	인력	렌탈프로	신상우	010-2638-6355	23/09/28

25 개의 행이 선택되었습니다.

order by 2, 7;

## 5. 건적 및 결제 관리 - 결제정보 조회

PLSQL

Q. 행사명 입력 시, 지출 금액이 가장 높은 렌탈 거래처 및 인력 거래처와 해당 비용을 출력하는 프로시저

```
SQL> create or replace procedure p_cm_cost(v_e_name in varchar2, v_cm out varchar2)
is
v_max_lcm cm.cm_name%type; v_max_lcost number;
v_max_rcm cm.cm_name%type; v_max_rcost number;
begin
select cm_name, cm_tot into v_max_rcm, v_max_rcost
from (select ei.e_name, cm_name, sum(ri_tot_pay) cm_tot from e_info ei, ct_info ci, rs_d rd, ri_pay rp, ri_info ri, cm cm
where ei.e_code=ci.e_code and ci.ct_code=rd.ct_code and rd.rs_d_code=rp.rs_d_code and ri.ri_code=rd.ri_code
and cm.cm_code=ri.cm_code and ei.e_name=v_e_name and ri_pay_ck='Y'
group by ei.e_name, cm.cm_name order by 3 desc)
where rownum=1;
select cm_name, cm_tot into v_max_lcm, v_max_lcost
from (select ei.e_name, cm_name, sum(l_tot_pay) cm_tot from e_info ei, ct_info ci, ls_d ld, l_pay lp, ld_info li, cm cm
where ei.e_code=ci.e_code and ci.ct_code=ld.ct_code and ld.ls_d_code=lp.ls_d_code and li.ld_code=ld.ld_code
and cm.cm_code=li.cm_code and ei.e_name=v_e_name and lp.l_pay_ck='Y'
group by ei.e_name, cm.cm_name order by 3 desc)
where rownum=1;
v_cm:=v_e_name|| '에서 가장 총지출금액이 높은 렌탈거래처는 '||v_max_rcm ||', 인력거래처는 '||v_max_lcm||' 이며,
각각 총 지출금액은 렌탈은 '||v_max_rcost||' 원, 인력은 '||v_max_lcost||' 원 입니다.';
end;
/
```

## 5. 건적 및 결제 관리 - 결제정보 조회

PLSQL

Q. 행사명 입력 시, 지출 금액이 가장 높은 렌탈 거래처 및 인력 거래처와 해당 비용을 출력하는 프로시저

```
SQL> create or replace procedure p_cm_cost(v_e_name in varchar2, v_cm out varchar2)
is
```

### 결과 화면

프로시저가 생성되었습니다.

```
SQL> variable cm varchar2(500)
SQL> exec p_cm_cost('2023 서울스트리트댄스페스티벌', :cm)
```

PL/SQL 처리가 정상적으로 완료되었습니다.

```
SQL> print cm
```

```
CM
```

2023 서울스트리트댄스페스티벌 에서 가장 총지출금액이 높은 렌탈거래처는 압구정렌탈, 인력거래처는 고려대학교 구로병원 이며, 각각 총 지출금액은 렌탈은 11760000 원, 인력은 2000000 원 입니다.

```
select cm_name, cm_tot into v_max_lcm, v_max_lcost
from (select ei.e_name, cm_name, sum(l_tot_pay) cm_tot from e_info ei, ct_info ci, ls_d ld, l_pay lp, ld_info li, cm cm
where ei.e_code=ci.e_code and ci.ct_code=ld.ct_code and ld.ls_d_code=lp.ls_d_code and li.ld_code=ld.ld_code
and cm.cm_code=li.cm_code and ei.e_name=v_e_name and lp.l_pay_ck='Y'
group by ei.e_name, cm.cm_name order by 3 desc)
where rownum=1;

v_cm:=v_e_name||' 에서 가장 총지출금액이 높은 렌탈거래처는 '||v_max_rcm||', 인력거래처는 '||v_max_lcm||' 이며,
각각 총 지출금액은 렌탈은 '||v_max_rcost||' 원, 인력은 '||v_max_lcost||' 원 입니다.;

end;

/
```

## 5. 건적 및 결제 관리 - 건적 및 결제정보 변경

PLSQL

Q. 인력 건적 요청 인원이 변경될 시, 인력 결제 테이블의 총 결제 금액을 변경하는 트리거

```
SQL> create or replace trigger t_update_lpay
      after update of l_qt on ls_d
      for each row
      declare
      begin
        update l_pay set l_tot_pay=:new.l_qt*(select ld_avg_cost from ld_info where ld_code=:old.ld_code)
        where ls_d_code=:new.ls_d_code;
      end;
      /
```



## 5. 건적 및 결제 관리 - 건적 및 결제정보 변경

PLSQL

Q. 인력 건적 요청 인원이 변경될 시, 인력 결제 테이블의 총 결제 금액을 변경하는 트리거

변경 전 Data

create trigger t\_update\_lpay

LS_D_CODE	L_QT	L_PAY_CODE	L_TOT_PAY
27001	2	33001	200000

d.ld\_code)

where ls\_d\_code=:new.ls\_d\_code;

변경 후 Data

트리거가 생성되었습니다.

```
SQL> update ls_d set l_qt=10 where ls_d_code=27001;
```

1 행이 갱신되었습니다.

```
SQL> select d.ls_d_code, d.l_qt, p.l_pay_code, p.l_tot_pay from ls_d d, l_pay p  
2 where d.ls_d_code=p.ls_d_code and d.ls_d_code=27001;
```

LS_D_CODE	L_QT	L_PAY_CODE	L_TOT_PAY
27001	10	33001	1000000

## 6. 회계 관리 - 행사 회계정보 조회

SQL

Q. 연도별 행사진행횟수 및 평균총예산, 평균총비용, 평균총수익 조회

```
SQL> with avg_cost as (select 연도, floor(sum(평균지출비용)) 평균총비용
                        from (select to_char(ei.e_s_date, 'yyyy') "연도", avg(l_tot_pay) "평균지출비용 " from e_info ei, ct_info ci, ls_d ld, l_pay lp, go_ck gc
                              where ei.e_code=ci.e_code and ci.ct_code=ld.ct_code and ld.ls_d_code=lp.ls_d_code and gc.go_ck_code=ei.go_ck_code
                              and gc.go_ckck in ('진행중', '종료') and lp.l_pay_ck='Y'
                              group by to_char(ei.e_s_date, 'yyyy')
                              union all
                              select to_char(ei.e_s_date, 'yyyy') "연도", avg(ri_tot_pay) "평균지출비용 " from e_info ei, ct_info ci, rs_d rd, ri_pay rp,
go_ck gc
                              where ei.e_code=ci.e_code and ci.ct_code=rd.ct_code and rd.rs_d_code=rp.rs_d_code and gc.go_ck_code=ei.go_ck_code
                              and gc.go_ckck in ('진행중', '종료') and rp.ri_pay_ck='Y'
                              group by to_char(ei.e_s_date, 'yyyy'))
                        group by 연도 order by 1),
avg_mon as (select eavg.*, pg.pr_g_code 평균이익률
              from (select to_char(ei.e_s_date, 'yyyy') "연도", count(*) "행사횟수", floor(avg(ei.e_tot_budget)) "평균총예산"
                    from e_info ei, ct_info ci, go_ck gc
                    where ei.e_code=ci.e_code and gc.go_ckck in ('진행중', '종료') and gc.go_ck_code=ei.go_ck_code
                    group by to_char(ei.e_s_date, 'yyyy')) eavg, pr_g pg
              where eavg.평균총예산>=pg.mini_tot_b and eavg.평균총예산<=pg.maxi_tot_b order by 1)
select m.*, c.평균총비용, floor((평균총예산*평균이익률)) "평균총수익"
from avg_cost c, avg_mon m
where c.연도=m.연도;
```

## 6. 회계 관리 - 행사 회계정보 조회

SQL

Q. 연도별 행사진행횟수 및 평균총예산, 평균총비용, 평균총수익 조회

### 결과 화면

```
select 연도, floor(sum(평균지출비용)) 평균총비용
```

```
from (select to_char(ei.e_s_date, 'yyyy') "연도", avg(l_tot_pay) "평균지출비용" from e_info ei, ct_info ci, s_dld l, pay lp, go_ck gc
where ei.go_ck_code=ci.go_ck_code and ci.go_ck_code=l.go_ck_code and l.go_ck_code=lp.go_ck_code and lp.go_ck_code=gc.go_ck_code)
```

연도	행사횟수	평균총예산	평균이익률	평균총비용	평균총수익
2019	24	99500000	.4	1912500	39800000
2020	14	70028571	.3	2820490	21008571
2021	1	180000000	.5	1736630	90000000
2022	15	101800000	.4	1331518	40720000
2023	26	126384615	.5	1634324	63192307

```
avg_mon as (select eavg, pg.pr_g_code 평균이익률
```

```
from (select to_char(ei.e_s_date, 'yyyy') "연도", count(*) "행사횟수", floor(avg(ei.e_tot_budget)) "평균총예산"
```

```
from e_info ei, ct_info ci, go_ck gc
```

```
where ei.e_code=ci.e_code and gc.go_ckck in ('진행중', '종료') and gc.go_ck_code=ei.go_ck_code
```

```
group by to_char(ei.e_s_date, 'yyyy')) eavg, pr_g pg
```

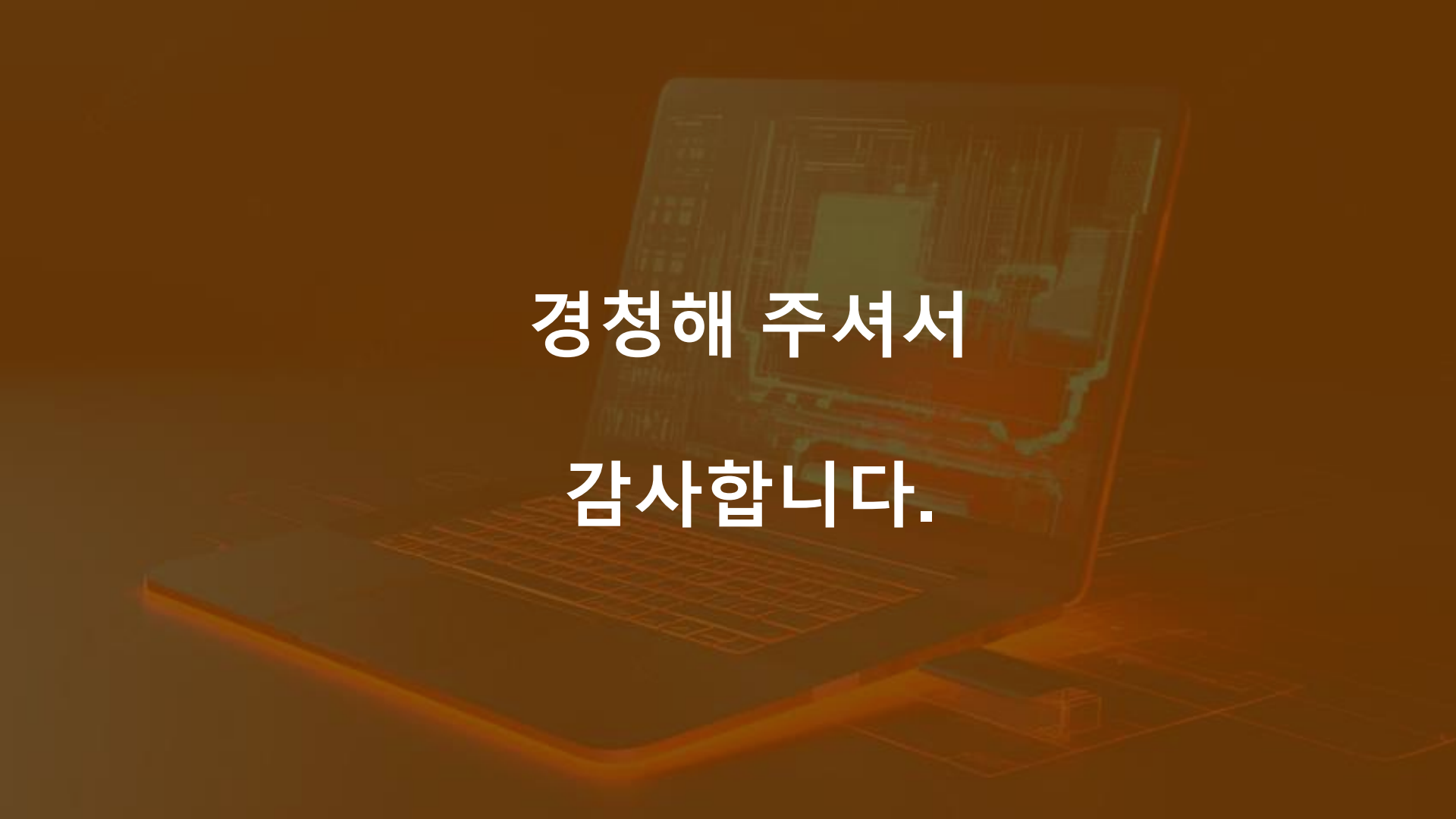
```
where eavg.평균총예산>pg.mini_tot_b and eavg.평균총예산<=pg.maxi_tot_b order by 1)
```

```
select m: "C:평균총비용", floor((평균총예산*평균이익률)) "평균총수익"
```

```
from avg_cost c, avg_mon m
```

```
where c.연도=m.연도;
```

이익률	최대 총예산	최소 총예산
30%	80,000,000	80,000,000 <sup>1</sup>
40%	125,000,000	80,000,001
50%	200,000,000	125,000,001



경청해 주셔서  
감사합니다.