

에어 프로젝트

10조
김혜원
이소연

목차

- 프로젝트 소개
- ERD
- 테이블명세서
- 주요기능과 쿼리
- 쿼리문 증빙
- 주요 전공용어 정리

프로젝트 소개

프로젝트 소개

항공권을 구입하는 회원들의 데이터를 관리하기 위해 생성

항공권 포털시스템으로 항공권을 등록하고,
회원들은 원하는 항공권을 검색하여
구매 및 좌석을 선택하는 시스템이다

-주요기능

주요 기능

1. 회원/항공권/구매내역/좌석정보 등록 기능 : Insert 명령어 사용
2. 회원/항공권/구매내역/좌석정보 현황보기 기능 : select 명령어 사용
3. 회원별 예약좌석 조회 기능 : select 문에서 join 사용
4. 회원별 구매 항공권 및 예약좌석 조회 기능 : view 정의
5. 탈퇴한 회원정보 조회 기능 : trigger 생성
6. 국내선 할인가 적용 기능 : procedure 사용

요구사항

- 항공권을 구매 및 관리한다.
- 항공권은 출발지, 목적지, 출발시간, 도착시간, 항공편명, 가격을 저장한다.
- 고객은 아이디, 이름, 전화번호, 생년월일, 여권번호, 회원등급, 포인트를 저장한다.
- 고객이 항공권을 구매하면 구매내역에 예약번호, 항공편명, 구매날짜, 결제수단, 여권번호가 저장된다.
- 한 사람이 여러 개의 항공권을 구매할 수 있다.
- 동시에 고객이 좌석을 선택하면 항공편명, 예약번호, 좌석등급, 좌석번호의 열과 번호가 저장된다.
- 한 사람이 하나의 좌석을 선택할 수 있다.
- 신규 고객은 포인트 기본값이 100이다.
- 신규 고객의 회원등급은 일반이다.
- 항공권이 취소되면 구매내역과 좌석정보가 삭제된다.
- 고객이 예약을 취소하면 좌석정보도 삭제된다.
- 고객이 탈퇴해도 구매내역 및 좌석정보는 남아있다.

ERD

테이블 찾기

- 항공권을 구매 및 관리한다.
- 항공권은 출발지, 목적지, 출발시간, 도착시간, 항공편명, 가격을 저장한다.
- 고객은 아이디, 이름, 전화번호, 생년월일, 여권번호, 회원등급, 포인트를 저장한다.
- 고객이 항공권을 구매하면 구매내역에 예약번호, 항공편명, 구매날짜, 결제수단, 여권번호가 저장된다.
- 한 사람이 여러 개의 항공권을 구매할 수 있다.
- 동시에 고객이 좌석을 선택하면 항공편명, 예약번호, 좌석등급, 좌석번호의 열과 번호가 저장된다.
- 한 사람이 하나의 좌석을 선택할 수 있다.
- 신규 고객은 포인트 기본값이 100이다.
- 신규 고객의 회원등급은 일반이다.
- 항공권이 취소되면 구매내역과 좌석정보가 삭제된다.
- 고객이 예약을 취소하면 좌석정보도 삭제된다.
- 고객이 탈퇴해도 구매내역 및 좌석정보는 남아있다.

컬럼 찾기

- 항공권을 구매 및 관리한다.
- 항공권은 출발지, 목적지, 출발시간, 도착시간, 항공편명, 가격을 저장한다.
- 고객은 아이디, 이름, 전화번호, 생년월일, 여권번호, 회원등급, 포인트를 저장한다.
- 고객이 항공권을 구매하면 구매내역에 예약번호, 항공편명, 구매날짜, 결제수단, 여권번호가 저장된다.
- 한 사람이 여러 개의 항공권을 구매할 수 있다.
- 동시에 고객이 좌석을 선택하면 항공편명, 예약번호, 좌석등급, 좌석번호의 열과 번호가 저장된다.
- 한 사람이 하나의 좌석을 선택할 수 있다.
- 신규 고객은 포인트 기본값이 100이다.
- 신규 고객의 회원등급은 일반이다.
- 항공권이 취소되면 구매내역과 좌석정보가 삭제된다.
- 고객이 예약을 취소하면 좌석정보도 삭제된다.
- 고객이 탈퇴해도 구매내역 및 좌석정보는 남아있다.

예약조건 찾기

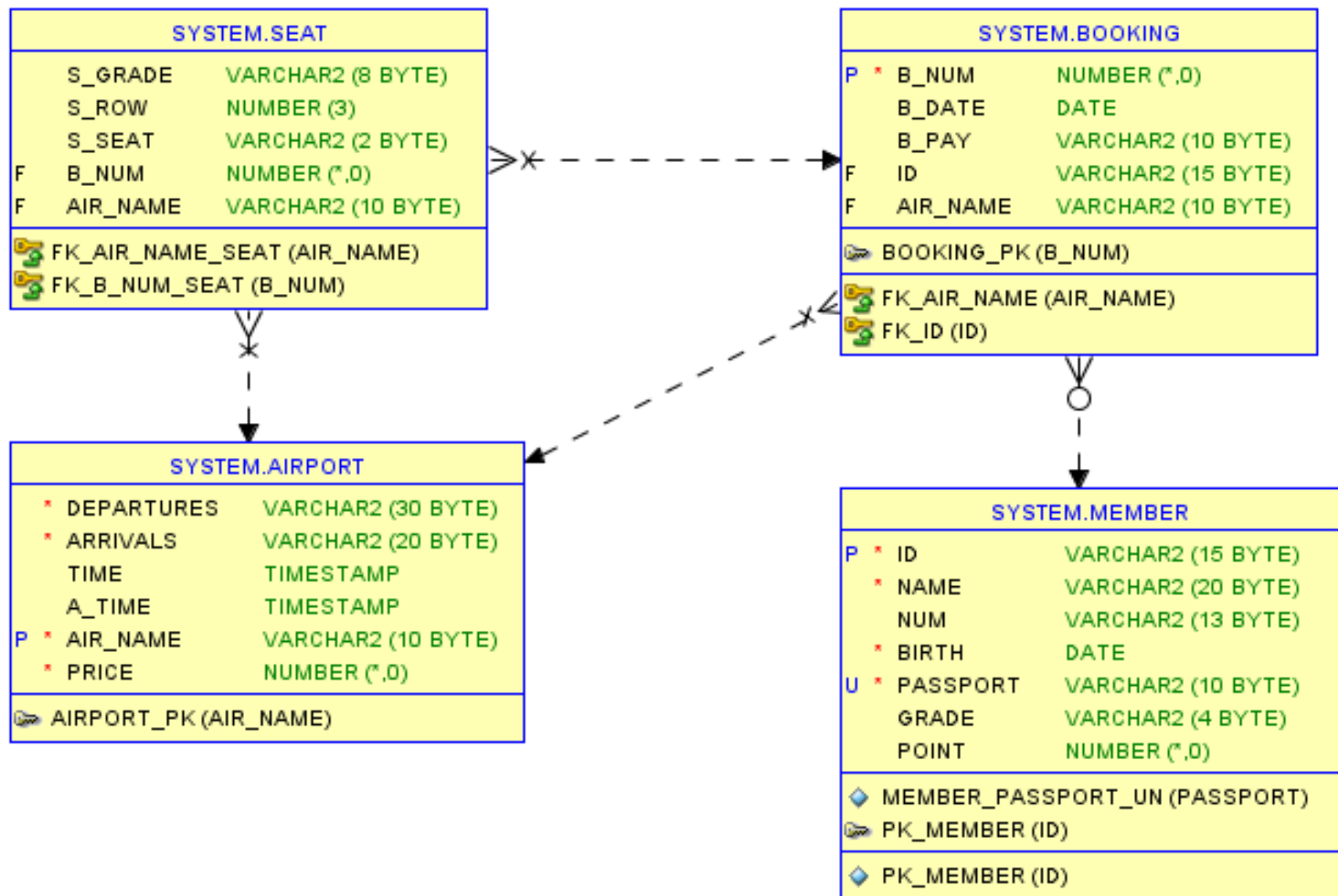
- 항공권을 구매 및 관리한다.
- 항공권은 출발지, 목적지, 출발시간, 도착시간, 항공편명, 가격을 저장한다.
- 고객은 아이디, 이름, 전화번호, 생년월일, 여권번호, 회원등급, 포인트를 저장한다.
- 고객이 항공권을 구매하면 구매내역에 예약번호, 항공편명, 구매날짜, 결제수단, 여권번호가 저장된다.
- 한 사람이 여러 개의 항공권을 구매할 수 있다.
- 동시에 고객이 좌석을 선택하면 항공편명, 예약번호, 좌석등급, 좌석번호의 열과 번호가 저장된다.
- 한 사람이 하나의 좌석을 선택할 수 있다.
- 신규 고객은 포인트 기본값이 100이다.
- 신규 고객의 회원등급 기본값은 일반이다.
- 항공권이 취소되면 구매내역과 좌석정보가 삭제된다.
- 고객이 예약을 취소하면 좌석정보도 삭제된다.
- 고객이 탈퇴해도 구매내역 및 좌석정보는 남아있다.

관계분석

- 한 명의 고객은 여러 개의 항공권을 구매할 수 있다.
- 하나의 항공권은 여러 명의 고객에게 판매될 수 있다.

=> **N:M**

- 하나의 구매항공권에서 한 개의 좌석을 선택할 수 있다
 - 하나의 좌석은 한 개의 구매항공권에서 선택될 수 있다
- => **1:1**



회원

항공권

PK

아이디

이름

전화번호

생년월일

여권번호

회원등급

포인트

PK

출발지

도착지

항공편명

출발시간

도착시간

가격

예약번호

구매날짜

결제수단

FK

회원아이디

FK

항공편명

구매

좌석

항공권

좌석등급

좌석 열

좌석번호

예약번호

FK

항공편명

FK

출발지

도착지

PK

항공편명

출발시간

도착시간

가격

PK

예약번호

구매날짜

결제수단

회원아이디

FK

항공편명

구매

테이블명세서

회원 테이블

빨간색=테이블
초록색=칼럼명
파란색=제약조건

```
create table member(  
id varchar2(15),  
name varchar2(20) not null,  
num varchar2(13),  
birth date not null,  
passport varchar2(10) unique not null,  
grade varchar2(4) default '일반' check(grade in('일반', 'vip', 'vwip')),  
point int default 100,  
constraint PK_member primary key (id)  
);
```

회원 테이블

테이블명세서

Table 명		Member			SYSTEM	
Entity 명		회원			작성자	
Entity 정의		회원의 정보를 관리하는 테이블				
No	Column 명	한글명	자료형	NULL	key	비 고
1	id	회원아이디	문자 15byte	X	primary	unique, index
2	name	회원이름	문자 20byte	X		
3	num	전화번호	문자 13byte	0	=null값 허용	
4	birth	생년월일	날짜 년/월/일	X	=not null	
5	passport	여권번호	문자 10byte	X		unique
6	grade	회원등급	문자 4byte	X		default '일반', check '일반','vip','vvip'
7	point	포인트	숫자(정수)	X		default 100

빨간색=테이블
초록색=칼럼명
파란색=제약조건

항공권 테이블

```
create table airport(  
departures varchar2(30) not null,  
arrivals varchar2(20) not null,  
time timestamp default sysdate,  
a_time timestamp default sysdate,  
air_name varchar2(10) primary key,  
price int not null  
);
```

항공권 테이블

테이블명세서

Table 명		Airport				SYSTEM	
Entity 명		항공권				작성자	
Entity 정의		회원이 구매한 항공권의 정보를 관리하는 테이블					
No	Column 명	한글명	자료형	NULL	key	비 고	
1	departures	출발지	문자 30 byte	X	=not null		
2	arrivals	목적지	문자 20byte	X			
3	time	출발시간	날짜	X		default 시스템상 현재날짜	
4	a_time	도착시간	날짜	X		default 시스템상 현재날짜	
5	air_name	항공편명	문자 10byte	X	primary	unique, index	
6	price	항공권 가격	숫자(정수)	X			

구매내역 테이블

빨간색=테이블
초록색=칼럼명
파란색=제약조건

```
create table booking(  
  b_num int primary key,  
  b_date date default sysdate,  
  b_pay varchar2(10) check(b_pay in('신용카드','무통장입금','네이버페이','카카오  
    페이')),  
  id varchar2(15),  
  air_name varchar2(10),  
  constraint FK_id foreign key (id) references member (id)  
  on delete set null,  
  constraint FK_air_name foreign key (air_name) references airport (air_name)  
  on delete cascade  
);
```

구매내역 테이블

테이블명세서

Table 명		Booking				SYSTEM	
Entity 명		구매정보				작성자	
Entity 정의		회원이 구매한 정보 관리 테이블					
No	Column 명	한글명	자료형	NULL	key	비고	
1	b_num	예약번호	숫자(정수)	X	primary	unique, index	
2	b_date	구매날짜	날짜 년/월/일	X		default 시스템상 현재날짜	
3	b_pay	결제수단	문자 10byte	X		check '신용카드, 무통장입금, 네이버페이, 카카오페이'	
4	id	회원아이디	문자 15byte	X	foreign	on delete set null	
5	air_name	항공편명	문자 10byte	X	foreign	on delete cascade	

좌석정보 테이블

빨간색=테이블
초록색=칼럼명
파란색=제약조건

```
create table seat(  
  s_grade varchar2(8) default '이코노미' check(s_grade in('이코노미', '비즈니스',  
    '퍼스트')),  
  s_row number(3) check(s_row between 1 and 80),  
  s_seat varchar2(2) check(s_seat in('A','B','C','D','E','F')),  
  b_num int,  
  air_name varchar2(10),  
  constraint FK_b_num_seat foreign key (b_num) references booking (b_num)  
  on delete cascade,  
  constraint FK_air_name_seat foreign key (air_name) references airport  
    (air_name)  
  on delete cascade  
);
```

좌석정보 테이블

테이블명세서

Table 명		Seat				SYSTEM	
Entity 명		좌석정보				작성자	
Entity 정의		회원이 구매한 항공권에 포함된 좌석정보 관리 테이블					
No	Column 명	한글명	자료형	NULL	key	비고	
1	s_grade	좌석등급	문자 8byte	X		default '이코노미' check '이코노미,비즈니스,퍼스트'	
2	s_row	좌석열	숫자(정수3자리)	X		Check(s_row between 1 and 80)	
3	s_seat	좌석번호	문자 2byte	X		Check ('A','B','C','D','E','F')	
4	b_num	예약번호	숫자(정수)	X	foreign	on delete cascade	
5	air_name	항공편명	문자 10byte	X	foreign	on delete cascade	

주요기능과 쿼리

회원 등록 기능 : Insert 명령어 사용

- insert into member values
(**'human1'**, '김휴먼', '010-1111-3434', '19920304', 'M000G1111',
default, default);
- insert into member values
(**'code99'**, '박코딩', '010-2222-4455', '20090914', 'M111G2222', default,
300);
- insert into member values
(**'java44'**, '안자바', '010-3578-8236', '19721201', 'M456G4578', 'vip',
5000);
- insert into member values
(**'oracle11'**, '오라클', '010-7897-7787', '20001111', 'M333G0000', 'wvip',
10000);

unique 와 not null 성질을 가지는 primary key로 인해 id(회원아이디) 중복 X
Index 값이 형성되어 검색에 용이

항공권 등록 기능 : Insert 명령어 사용

- insert into airport values
('김포', '제주도', '23/08/28 09:00', '23/08/28 11:00', 'A111', 150000);
- insert into airport values
('인천', '부산', '23/09/29 10:00', '23/09/29 10:50', 'C333', 50000);
- insert into airport values
('부산', '인천', '23/10/03 19:00', '23/10/03 19:50', 'C303', 50000);
- insert into airport values
('인천', '나트랑', '23/07/01 06:30', '23/07/02 11:00', 'B222', 200000);
- insert into airport values
('나트랑', '인천', '23/07/08 22:00', '23/07/09 02:30', 'B202', 210000);
- insert into airport values
('인천', '프랑크푸르트', '23/10/10 15:00', '23/10/12 22:00', 'D444', 500000);

unique 와 not null 성질을 가지는 primary key로 인해 air_name(항공편명) 중복 X
Index 값이 형성되어 검색에 용이

구매내역 등록 기능 : Insert 명령어 사용

- insert into booking values
(`book_seq.nextval`, default, '신용카드', 'oracle11', 'A111');
- insert into booking values
(`book_seq.nextval`, default, '네이버페이', 'oracle11', 'B222');
- insert into booking values
(`book_seq.nextval`, default, '무통장입금', 'human1', 'A111');
- insert into booking values
(`book_seq.nextval`, default, '카카오페이', 'code99', 'A111');
- insert into booking values
(`book_seq.nextval`, default, '신용카드', 'java44', 'D444');

시퀀스를 이용하여 구매순서에 따라 자동 번호 부여
unique 와 not null 성질을 가지는 primary key로 인해 번호 중복 X
Index 값이 형성되어 검색에 용이

구매내역 등록 기능 : Insert 명령어 사용

- insert into booking values
(book_seq.nextval, default, '신용카드', 'oracle11', 'A111');
- insert into booking values
(book_seq.nextval, default, '네이버페이', 'oracle11', 'B222');
- insert into booking values
(book_seq.nextval, default, '무통장입금', 'human1', 'A111');
- insert into booking values
(book_seq.nextval, default, '카카오페이', 'code99', 'A111');
- insert into booking values
(book_seq.nextval, default, '신용카드', 'java44', 'D444');

자식릴레이션이 참조하는 부분인 id(아이디)와 air_name(항공편명) 컬럼에
부모릴레이션의 id(아이디)와 air_name(항공편명) 컬럼의 튜플값 삽입

※부모릴레이션에 존재하지 않는 튜플값 삽입 시 '부모키가 존재하지 않습니다!' 오류 발생※

예약좌석 등록 기능 : Insert 명령어 사용

- insert into seat values ('퍼스트', 1, 'A', 1, 'A111');
- insert into seat values ('비즈니스', 10, 'D', 2, 'B222');
- insert into seat values (default, 25, 'F', 3, 'A111');
- insert into seat values (default, 70, 'C', 4, 'A111');
- insert into seat values (default, 54, 'B', 5, 'D444');

자식릴레이션이 참조하는 부분인 b_num(예약번호)와 air_name(항공편명) 컬럼에 부모릴레이션의 b_num(예약번호)와 air_name(항공편명) 컬럼의 튜플값 삽입

※부모릴레이션에 존재하지 않는 튜플값 삽입 시 '부모키가 존재하지 않습니다!' 오류 발생※

쿼리문 증빙

Table member(회원)

Table MEMBER이(가) 생성되었습니다.

이름	널?	유형
ID	NOT NULL	VARCHAR2(15)
NAME	NOT NULL	VARCHAR2(20)
NUM		VARCHAR2(13)
BIRTH	NOT NULL	DATE
PASSPORT	NOT NULL	VARCHAR2(10)
GRADE		VARCHAR2(4)
POINT		NUMBER(38)

회원 현황보기 기능 : select 명령어 사용

```
select * from member;
```

크립트 출력 x | 실행 결과 x

SQL | 인출된 모든 행: 4(0초)

ID	NAME	NUM	BIRTH	PASSPORT	GRADE	POINT
1 human1	김휴먼	010-1111-3434	92/03/04	M000G1111	일반	100
2 code99	박코딩	010-2222-4455	09/09/14	M111G2222	일반	300
3 java44	안자바	010-3578-8236	72/12/01	M456G4578	vip	5000
4 oracle11	오라클	010-7897-7787	00/11/11	M333G0000	vvip	10000

Table airport(항공권)

Table AIRPORT이(가) 생성되었습니다.

이름	널?	유형
DEPARTURES	NOT NULL	VARCHAR2(30)
ARRIVALS	NOT NULL	VARCHAR2(20)
TIME		TIMESTAMP(6)
A_TIME		TIMESTAMP(6)
AIR_NAME	NOT NULL	VARCHAR2(10)
PRICE	NOT NULL	NUMBER(38)

항공권 현황보기 기능 : select 명령어 사용

```
select * from airport;
```

DEPARTURES	ARRIVALS	TIME	A_TIME	AIR_NAME	PRICE
1 김포	제주도	23/08/28 09:00:00.000000000	23/08/28 11:00:00.000000000	A111	150000
2 인천	부산	23/09/29 10:00:00.000000000	23/09/29 10:50:00.000000000	C333	50000
3 부산	인천	23/10/03 19:00:00.000000000	23/10/03 19:50:00.000000000	C303	50000
4 인천	나트랑	23/07/01 06:30:00.000000000	23/07/02 11:00:00.000000000	B222	200000
5 나트랑	인천	23/07/08 22:00:00.000000000	23/07/09 02:30:00.000000000	B202	210000
6 인천	프랑크푸르트	23/10/10 15:00:00.000000000	23/10/12 22:00:00.000000000	D444	500000

Table booking(구매)

Table BOOKING이 (가) 생성되었습니다.

이름	널?	유형
B_NUM	NOT NULL	NUMBER(38)
B_DATE		DATE
B_PAY		VARCHAR2(10)
B_ID		VARCHAR2(15)
AIR_NAME		VARCHAR2(10)

구매내역 현황보기 기능 :
select 명령어 사용

```
select * from booking;
```

시퀀스를 이용하여 구매순서에 따라 자동 번호 부여

```
create sequence booking_seq;
```

Sequence BOOKING_SEQ이 (가) 생성되었습니다.

	B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	1	23/09/12	신용카드	oracle11	A111
2	2	23/09/12	네이버페이	oracle11	B222
3	3	23/09/12	무통장입금	human1	A111
4	4	23/09/12	카카오페이	code99	A111
5	5	23/09/12	신용카드	java44	D444

Table booking(구매)

Table BOOKING이 (가) 생성되었습니다.

이름	널?	유형
B_NUM	NOT NULL	NUMBER(38)
B_DATE		DATE
B_PAY		VARCHAR2(10)
B_ID		VARCHAR2(15)
AIR_NAME		VARCHAR2(10)

부모 릴레이션에 참조된 데이터 중복 출력 허용

	⚡ B_NUM	⚡ B_DATE	⚡ B_PAY	⚡ B_ID	⚡ AIR_NAME
1	1	23/09/12	신용카드	oracle11	A111
2	2	23/09/12	네이버페이	oracle11	B222
3	3	23/09/12	무통장입금	human1	A111
4	4	23/09/12	카카오페이	code99	A111
5	5	23/09/12	신용카드	java44	D444

Table seat(좌석)

Table SEAT이(가) 생성되었습니다.

이름	널? 유형
S_GRADE	VARCHAR2 (8)
S_ROW	NUMBER (3)
S_SEAT	VARCHAR2 (2)
B_NUM	NUMBER (38)
AIR_NAME	VARCHAR2 (10)

부모 릴레이션에 참조된 데이터 중복 출력 허용

좌석 현황보기 기능 : select 명령어 사용

```
select * from seat;
```

크립트 출력 x 질의 결과 x				
SQL 인출된 모든 행: 5(0초)				
S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1 퍼스트	1	A	1	A111
2 비즈니스	10	D	2	B222
3 이코노미	25	F	3	A111
4 이코노미	70	C	4	A111
5 이코노미	54	B	5	D444

회원 삭제 시 구매테이블

- delete from member where id='human1';

```
create table booking(  
  b_num int primary key,  
  b_date date default sysdate,  
  b_pay varchar2(10) check(b_pay in('신용카드','무통장입금','네이버페이','카카오페이')),  
  b_id varchar2(15),  
  air_name varchar2(10),  
  constraint FK_b_id foreign key (b_id) references member (id)  
  on delete set null,  
  constraint FK_air_name foreign key (air_name) references airport (air_name)  
  on delete cascade  
);
```

On delete set null

=> 부모 릴레이션(member) 데이터 삭제 시 자식 릴레이션(booking) 참조 데이터 null

회원 삭제 시 구매테이블

- delete from member where id='human1';

```
delete from member where id='human1';
select * from member;
```

크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 3(0초)

ID	NAME	NUM	BIRTH	PASSPORT	GRADE	POINT
1 oracle11	오라클	010-7897-7787	00/11/11	M333G0000	vvip	10000
2 code99	박코딩	010-2222-4455	09/09/14	M111G2222	일반	300
3 java44	안자바	010-3578-8236	72/12/01	M456G4578	vip	5000

```
select * from booking;
```

크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 5(0초)

B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	1 23/09/12	신용카드	oracle11	A111
2	2 23/09/12	네이버페이	oracle11	B222
3	3 23/09/12	무통장입금	(null)	A111
4	4 23/09/12	카카오페이	code99	A111
5	5 23/09/12	신용카드	java44	D444

S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1 이코노미	54 B		5 D444	
2 퍼스트	1 A		1 A111	
3 비즈니스	10 D		2 B222	
4 이코노미	25 F		3 A111	
5 이코노미	70 C		4 A111	

구매 취소 시 좌석테이블

- delete from booking where b_num=5;

```
create table seat(  
  s_grade varchar2(8) default '이코노미' check(s_grade in('이코노미', '비즈니스', '퍼스트')),  
  s_row number(3) check(s_row between 1 and 80),  
  s_seat varchar2(2) check(s_seat in('A', 'B', 'C', 'D', 'E', 'F')),  
  b_num int,  
  air_name varchar2(10),  
  constraint FK_b_num_seat foreign key (b_num) references booking (b_num)  
  on delete cascade,  
  constraint FK_air_name_seat foreign key (air_name) references airport (air_name)  
  on delete cascade  
);
```

자식릴레이션이 참조하고 있는 부모릴레이션 삭제X

On delete cascade

=>부모 릴레이션(member) 데이터 삭제 시 자식 릴레이션(booking) 참조 데이터 삭제

구매 취소 시 좌석테이블

- delete from booking where b_num=5;

B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	23/09/12	신용카드	oracle11	A111
2	23/09/12	네이버페이	oracle11	B222
3	23/09/12	무통장입금	human1	A111
4	23/09/12	카카오페이	code99	A111
5	23/09/12	신용카드	java44	D444

SQL | 인출된 모든 행: 4(0초)

B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	23/09/12	신용카드	oracle11	A111
2	23/09/12	네이버페이	oracle11	B222
3	23/09/12	무통장입금	human1	A111
4	23/09/12	카카오페이	code99	A111

부모테이블 데이터 삭제

```
select * from seat;
```

크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 5(0초)

S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1 퍼스트	1 A		1 A111	
2 비즈니스	10 D		2 B222	
3 이코노미	25 F		3 A111	
4 이코노미	70 C		4 A111	
5 이코노미	54 B		5 D444	

```
select * from seat;
```

크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 4(0초)

S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1 퍼스트	1 A		1 A111	
2 비즈니스	10 D		2 B222	
3 이코노미	25 F		3 A111	
4 이코노미	70 C		4 A111	

자식테이블 참조 데이터 삭제

항공권 삭제 시 구매, 좌석테이블

- delete from airport where air_name='B222';

```
create table booking(
  b_num int primary key,
  b_date date default sysdate,
  b_pay varchar2(10) check(b_pay in('신용카드', '무통장입금', '네이버페이', '카카오페이')),
  b_id varchar2(15),
  air_name varchar2(10),
  constraint FK_b_id foreign key (b_id)
on delete set null,
constraint FK_air_name foreign key (air_name)
on delete cascade
);
```

구매테이블

자식릴레이션이 참조하고 있는 부모릴레이션 삭제X

On delete cascade

=>부모 릴레이션(member) 데이터 삭제 시 자식 릴레이션(booking) 참조 데이터 삭제

```
create table seat(
  s_grade varchar2(8) default '이코노미' check(s_grade in('이코노미', '비즈니스', '퍼스트')),
  s_row number(3) check(s_row between 1 and 80),
  s_seat varchar2(2) check(s_seat in('A', 'B', 'C', 'D', 'E', 'F')),
  b_num int,
  air_name varchar2(10),
  constraint FK_b_num_seat foreign key (b_num) references booking (b_num)
on delete cascade,
constraint FK_air_name_seat foreign key (air_name) references airport (air_name)
on delete cascade
);
```

좌석테이블

항공권 삭제 시 구매, 좌석테이블

- delete from airport where air_name='B222';

```
select * from airport;
```

	DEPARTURES	ARRIVALS	TIME
1	김포	제주도	23/08/28 09:00:00.000000000
2	인천	부산	23/09/29 10:00:00.000000000
3	부산	인천	23/10/03 19:00:00.000000000
4	인천	나트랑	23/07/01 06:30:00.000000000
5	나트랑	인천	23/07/08 22:00:00.000000000
6	인천	프랑크푸르트	23/10/10 15:00:00.000000000

```
delete from airport where air_name='B222';
select * from airport;
```

	DEPARTURES	ARRIVALS	TIME
1	김포	제주도	23/08/28 09:00:00.000000000
2	인천	부산	23/09/29 10:00:00.000000000
3	부산	인천	23/10/03 19:00:00.000000000
4	나트랑	인천	23/07/08 22:00:00.000000000
5	인천	프랑크푸르트	23/10/10 15:00:00.000000000

부모테이블 데이터 삭제

	B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	1	23/09/12	신용카드	oracle11	A111
2	2	23/09/12	네이버페이	oracle11	B222
3	3	23/09/12	무통장입금	human1	A111
4	4	23/09/12	카카오페이	code99	A111
5	5	23/09/12	신용카드	java44	D444

	B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
1	1	23/09/12	신용카드	oracle11	A111
2	3	23/09/12	무통장입금	human1	A111
3	4	23/09/12	카카오페이	code99	A111
4	5	23/09/12	신용카드	java44	D444

자식테이블 참조 데이터 삭제

항공권 삭제 시 구매, 좌석테이블

- delete from airport where air_name='B222';

```
select * from airport;
```

	DEPARTURES	ARRIVALS	TIME
1	김포	제주도	23/08/28 09:00:00.000000000
2	인천	부산	23/09/29 10:00:00.000000000
3	부산	인천	23/10/03 19:00:00.000000000
4	인천	나트랑	23/07/01 06:30:00.000000000
5	나트랑	인천	23/07/08 22:00:00.000000000
6	인천	프랑크푸르트	23/10/10 15:00:00.000000000

```
delete from airport where air_name='B222';
```

```
select * from airport;
```

	DEPARTURES	ARRIVALS	TIME
1	김포	제주도	23/08/28 09:00:00.000000000
2	인천	부산	23/09/29 10:00:00.000000000
3	부산	인천	23/10/03 19:00:00.000000000
4	나트랑	인천	23/07/08 22:00:00.000000000
5	인천	프랑크푸르트	23/10/10 15:00:00.000000000

부모테이블 데이터 삭제

```
select * from seat;
```

	S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1	퍼스트	1	A	1	A111
2	바즈니스	10	B	2	B222
3	이코노미	25	F	3	A111
4	이코노미	70	C	4	A111
5	이코노미	54	B	5	D444

```
select * from seat;
```

	S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME
1	퍼스트	1	A	1	A111
2	이코노미	25	F	3	A111
3	이코노미	70	C	4	A111
4	이코노미	54	B	5	D444

자식테이블 참조 데이터 삭제

테이블 기능

-회원의 좌석정보 (inner join)

```
select m.name,b.air_name,s.*
from member m inner join booking b
on m.id=b.b_id
inner join seat s
on b.b_num=s.b_num;
```

스크립트 출력 x | 실행의 결과 x

SQL | 인출된 모든 행: 5(0초)

NAME	AIR_NAME	S_GRADE	S_ROW	S_SEAT	B_NUM	AIR_NAME_1
오라클	A111	퍼스트	1	A	1	A111
오라클	B222	비즈니스	10	D	2	B222
김휴먼	A111	이코노미	25	F	3	A111
박코딩	A111	이코노미	70	C	4	A111
안자바	D444	이코노미	54	B	5	D444

Inner join 을 사용하여 회원정보와 좌석정보가 같이 출력
=>5행 출력 기준 0초

-회원의 결제정보 (inner join)

```
select m.*,b.*
from member m inner join booking b
on m.id=b.b_id;
```

스크립트 출력 x ▶ 질의 결과 x

SQL | 인출된 모든 행 6(0초)

ID	NAME	NUM	BIRTH	PASSPORT	GRADE	POINT	B_NUM	B_DATE	B_PAY	B_ID	AIR_NAME
oracle11	오라클	010-7897-7787	00/11/11	M333G0000	vvip	10000	1	23/09/14	신용카드	oracle11	A111
oracle11	오라클	010-7897-7787	00/11/11	M333G0000	vvip	10000	2	23/09/14	네이버페이	oracle11	B222
human1	김휴먼	010-1111-3434	92/03/04	M000G1111	일반	100	3	23/09/14	무통장입금	human1	A111
code99	박코딩	010-2222-4455	09/09/14	M111G2222	일반	300	4	23/09/14	카카오페이	code99	A111
java44	안자바	010-3578-8236	72/12/01	M456G4578	vip	5100	5	23/09/14	신용카드	java44	D444
java44	안자바	010-3578-8236	72/12/01	M456G4578	vip	5100	47	23/09/14	신용카드	java44	C333

Inner join을 사용하여 회원정보와 결제정보 같이 출력
=>6행 출력 기준 0초

-구매가 없는 항공권 정보 (outer join)

```
select a.*,m.id as 고객명  
from airport a  
left outer join booking b  
on b.air_name=a.air_name  
left outer join member m  
on m.id=b.b_id  
where m.id is null;
```

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 2(0초)

	DEPARTURES	ARRIVALS	TIME	A_TIME	AIR_NAME	PRICE	고객명
1	부산	인천	23/10/03 19:00:00.000000000	23/10/03 19:50:00.000000000	C303	50000	(null)
2	나트랑	인천	23/07/08 22:00:00.000000000	23/07/09 02:30:00.000000000	B202	210000	(null)

outer join을 사용하여 구매가 없는 항공권 정보를 찾아 출력
=>2행 출력 기준 0초

-회원항공권 정보 (view)

```
create view 회원항공권_view as(
select m.name,m.passport,a.departures,a.arrivals,
substr(a.time,1,14) 출발시간,substr(a.a_time,1,14) 도착시간,a.air_name,s.s_grade,s.s_row,s.s_seat
from member m inner join booking b
on m.id=b.b_id
inner join airport a
on b.air_name=a.air_name
inner join seat s
on b.b_num=s.b_num
);
select * from 회원항공권_view;
```

스크립트 출력 x | 실행 결과 x

SQL | 인출된 모든 행: 5(0초)

	NAME	PASSPORT	DEPARTURES	ARRIVALS	출발시간	도착시간	AIR_NAME	S_GRADE	S_ROW	S_SEAT
1	박코딩	M111G2222	김포	제주도	23/08/28 09:00	23/08/28 11:00	A111	이코노미	70	C
2	김휴먼	M000G1111	김포	제주도	23/08/28 09:00	23/08/28 11:00	A111	이코노미	25	F
3	오라클	M333G0000	김포	제주도	23/08/28 09:00	23/08/28 11:00	A111	퍼스트	1	A
4	오라클	M333G0000	인천	나트랑	23/07/01 06:30	23/07/02 11:00	B222	비즈니스	10	D
5	안자바	M456G4578	인천	프랑크푸르트	23/10/10 15:00	23/10/12 22:00	D444	이코노미	54	B

Join을 매번 하지않고 회원정보,항공권정보,좌석정보를 하나의 테이블로 출력해내는
회원항공권 View 테이블 생성
=>5행 출력 기준 0초

-탈퇴한 회원 정보 저장 (trigger)

탈퇴한 회원의 정보를 저장하는 테이블 생성

```
desc delete_member;
```

이름	널?	유형
ID		VARCHAR2(15)
NAME	NOT NULL	VARCHAR2(20)
NUM		VARCHAR2(13)

탈퇴한 회원 테이블에 대한 트리거 형성

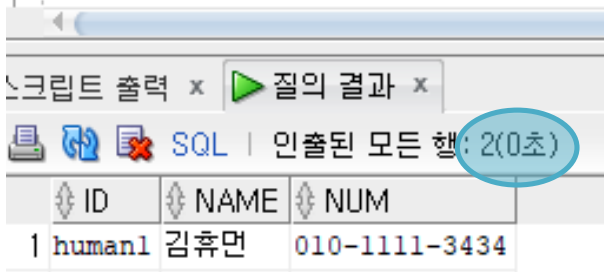
```
create trigger trg_deletemember
after delete
on member
for each row
begin
insert into delete_member
values(:old.id, :old.name, :old.num);
end;
```

-탈퇴한 회원 정보 저장 (trigger)

```
delete member where id='human1';
```

회원id 'human1'인 회원 탈퇴

```
select * from delete_member;
```



ID	NAME	NUM
human1	김휴먼	010-1111-3434

트리거 작동하여
탈퇴한 회원정보 자동으로
탈퇴회원테이블에 데이터 저장
=>2행 출력 기준 0초

-국내선 할인가 적용 (procedure)

빨간색=프로시저명
초록색=칼럼명
파란색=테이블명

```
CREATE OR REPLACE PROCEDURE domestic_discount
IS
BEGIN
UPDATE airport SET price=price*0.9
WHERE departures IN ('인천','부산','제주도','김포')
AND arrivals IN ('인천','부산','제주도','김포');
END;
```

DEPARTURES	ARRIVALS	TIME	A_TIME	AIR_NAME	PRICE
김포	제주도	23/08/28	00:00:00.000000000	A111	150000
인천	부산	23/09/29	00.000000000	C333	50000
인천	나트랑	23/07/01	00.000000000	B222	200000
인천	프랑크푸르트	23/10/10	00.000000000	D444	500000
나트랑	인천	23/07/08	00.000000000	B202	210000
부산	인천	23/10/03	19:00:00.000000000	C303	50000

실행 전

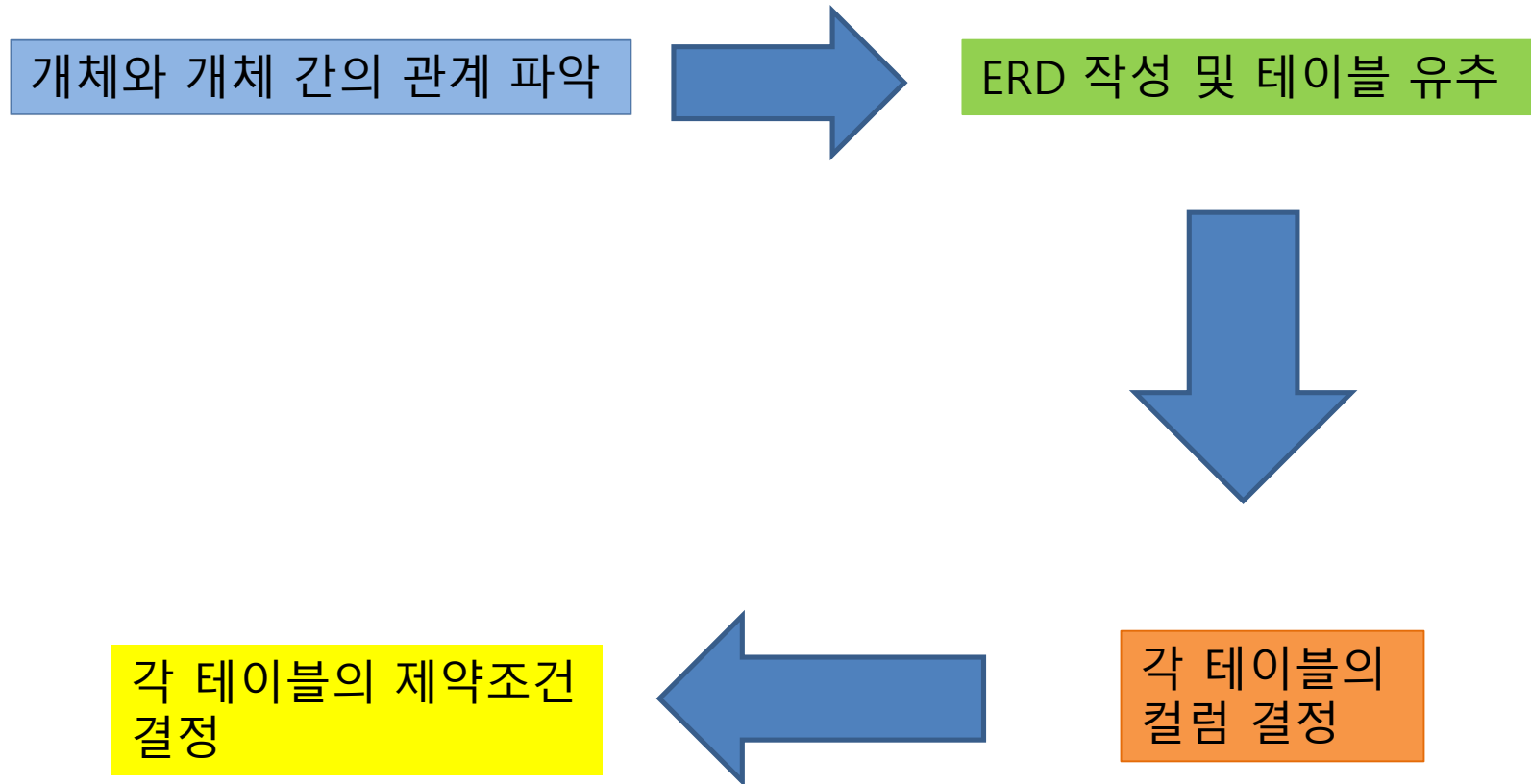
DEPARTURES	ARRIVALS	TIME	A_TIME	AIR_NAME	PRICE
김포	제주도	23/08/28	00.000000000	A111	135000
인천	부산	23/09/29	00.000000000	C333	45000
인천	프랑크푸르트	23/10/10	00.000000000	D444	500000
나트랑	인천	23/07/08	00.000000000	B202	210000
부산	인천	23/10/03	00.000000000	C303	45000

실행 후

주요 전공용어 정리

-요구사항 분석

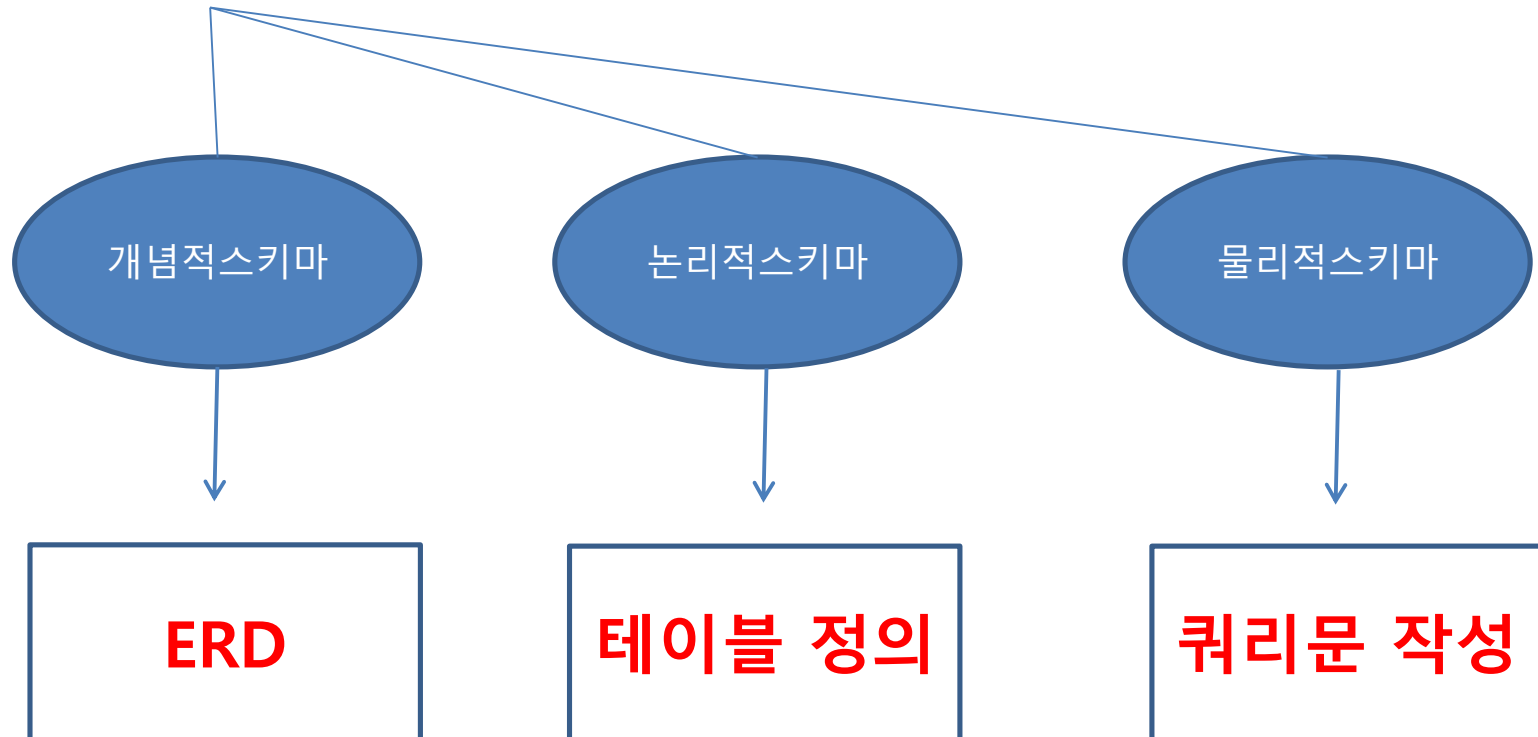
요구사항 분석



-ERD

요구사항 분석

현실세계 이슈 테이블 설계
=> 데이터 모델링



ERD

개체와 개체를 도식화하여
개체들의 관계성을 찾아내는 과정



주어진 요구사항을 바탕으로
개체와 개체의 관계성을 분석한
테이블 형성 전 단계

※ 속성과 제약조건 등을 고려하지 않고 오직
개체간의 관계만 분석

-제약조건

제약조건

테이블에 데이터를
작성/삽입/갱신할 때
조건에 맞지 않는
데이터를 입력시키지
않기 위한 구조

Primary key (기본키)

Foreign key (외래키)

자식 릴레이션의 속성이
부모 릴레이션 속성을 참조

unique

중복 값 금지

Not null

null 값 금지
=반드시 값 입력

check

컬럼에 입력할 수 있는
값의 범위/조건 지정

제약조건

테이블에 데이터를 작성/삽입/갱신할 때 조건에 맞지 않는 데이터를 입력시키지 않기 위한 구조

Primary key (기본키)

Unique & not null & index 자동 생성

설정 방법:

유일성 - 튜플을 유일하게 식별
최소성 - 키를 구성하는 속성 중 불필요한 속성을 빼고 최소한의 속성들로만 키를 구성

Foreign key (외래키)

unique

Not null

check

-Index

Index

Table 검색 시 검색속도를 향상시키기 위해
table의 컬럼에
부여 하는 것

특징1:
table의 검색과 정렬 속도 향상

특징2:
table 컬럼의 고유성 강화

※primary key 기본키에는 unique, not null과 더불어 index가 자동생성
※index는 중요한 정보를 담고 있는 컬럼이 아닌 곳에 부여
※index가 여러 개 있을 경우 오히려 검색속도 저하

Index

특징1:
table의 검색과 정렬 속도 향상

특징2:
table 컬럼의 고유성 강화

Table 검색 시 검색속도를
향상시키기 위해
table의 컬럼에
부여 하는 것

Index 생성O
=> table 컬럼 검색 시
index가 부여된 table만을 검색하여
검색속도 ↑

Index 생성X
=> table 컬럼 검색 시
table의 전체데이터를 검색하여 **검색속도 ↓**

-DDL, DML

DDL

(Data Definition Language)

데이터 정의어
=테이블을 정의하는
명령어

Create : 테이블 생성

Drop : 테이블 삭제

Rename : 테이블명 변경

Alter : 테이블 수정

+ add : 컬럼 추가

+ modify : 컬럼 속성/
제약조건 수정

+ drop : 컬럼 삭제

+ rename : 컬럼명 수정

DML

(Data Manipulation Language)

데이터 조작어
= 튜플을 조작하는
명령어

update : 튜플 수정

insert : 튜플 삽입

시퀀스

delete : 튜플 삭제

select : 튜플 조회

함수

서브쿼리

조인

별칭

-시퀀스

시퀀스

(sequence)

순차적으로 증가하는 순번을
자동으로 반환하는
데이터베이스 객체

Create : 시퀀스 생성

Drop : 시퀀스 삭제

select : 시퀀스 조회

alter : 시퀀스 수정

+ increment by : 증가값 설정

+ minvalue : 최소값 설정

+ maxvalue : 최대값 설정

변수

(Variable)

변수명 규칙

첫 번째 글자는 문자, \$, _ 중 하나

영어 대소문자 구분

Name != name

카멜 케이스

값 저장 :

- | | |
|-----------------------|--------------------------|
| 1. 값을 저장할 데이터가 필요하다 | 1. 변수에 값을 저장해야 변수가 생성된다 |
| 2. 데이터가 저장되는 위치를 지정한다 | 2. 생성 시에 사용한 타입의 값만 저장된다 |

변수명_o, 값 저장_x

변수 선언

변수명_o, 값 저장_o

변수 생성

초기값 : 처음 변수에 저장한 값

초기값 : 처음 변수에 저장한 값

-함수

함수

(function)

매개변수(입력값)에
어떠한 작용을 하여
원하는 리턴값(결과값)을
받기 위해 사용하는 기능

집계함수 : 그룹화

Sum : 그룹의 총합

Max : 그룹의 최대값

Min : 그룹의 최소값

Avg : 그룹의 평균

Count : 그룹의 총 수

문자함수

날짜함수

숫자함수

함수

(function)

매개변수(입력값)에
어떠한 작용을 하여
원하는 리턴값(결과값)을
받기 위해 사용하는 기능

집계함수

문자함수 : 문자형 데이터에 사용

Upper : 문자를 모두 대문자로 변환해 반환

Lower : 문자를 모두 소문자로 변환해 반환

Trim : 특정 문자 삭제

Length : 문자수 반환

Lengthb : 바이트수 반환

Substr : 문자 일부 추출

Instr : 특정 문자의 위치 반환

Replace : 특정 문자를 다른 문자로 대체

L(R)Pad : 문자 맨앞, 맨뒤에 특정 문자 입력

Concat : 문자끼리 합침

날짜함수

숫자함수

함수

(function)

매개변수(입력값)에
어떠한 작용을 하여
원하는 리턴값(결과값)을
받기 위해 사용하는 기능

집계함수

문자함수

날짜함수 : 날짜형 데이터에 사용

Add_months : 몇 개월 이후 날짜 반환

Months_between : 두 날짜 간 개월 수
차이 반환

Round/Trunc : 날짜의 반올림/버림 값
반환

숫자함수

함수

(function)

매개변수(입력값)에
어떠한 작용을 하여
원하는 리턴값(결과값)을
받기 위해 사용하는 기능

집계함수

문자함수

날짜함수

숫자함수 : 숫자형 데이터에 사용

round : 특정 위치에서 반올림

Trunc : 특정 위치에서 버림

mod : 숫자를 나눈 나머지 값

함수

(function)

매개변수(입력값)에
어떠한 작용을 하여
원하는 리턴값(결과값)을
받기 위해 사용하는 기능

집계함수

문자함수

날짜함수

숫자함수 : 숫자형 데이터에 사용



rownum: select 문에 자동으로 번호 부여
1번부터 가져오는 명령어

=> 중간 rownum 출력시
rownum을 생성한 쿼리 결과를
가상의 테이블로 지정하여 인라인쿼리로 활용

-서브쿼리

서브쿼리

(SubQuery)

본쿼리 안에 소괄호로
묶인 select 절

특징1 :
서브쿼리 결과값을
본쿼리에서 사용

특징2 :
서브쿼리 결과값의
단일행/다중행 여부
판단하여 연산자 선택

스칼라 서브쿼리

위치 : select 절

인라인뷰 서브쿼리

위치 : from 절

중첩 서브쿼리

위치 : where/having 절

	단일행	다중행
튜플 수	하나	여러개
연산자	=, <, >	in

-Join

조인

(join)

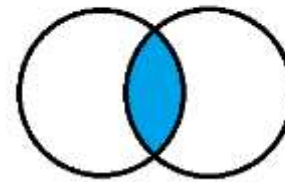
2개 이상의 테이블을
하나의 논리적인
테이블로 합치는 것
(=역정규화)

효과 :
원하는 데이터를
편리하게 볼 수 있음

Inner join

테이블간 매칭되는 튜플만 반영

$A \cap B$

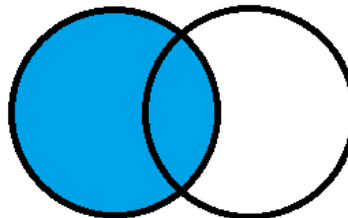


Outer join

테이블간 매칭되지 않는 튜플까지
모두 반영

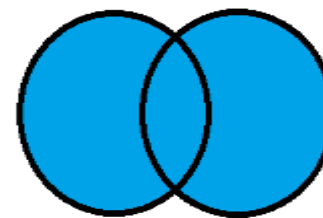
Left outer join

A B



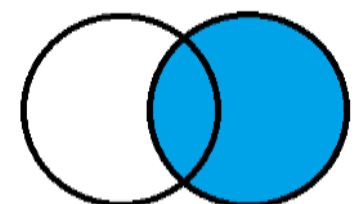
Full outer join

A B



Right outer join

A B



—View

View

Join이 필요한 기능을 실행할 때 물리적인 테이블 여러 개를 검색하지 않고 논리적인 테이블 하나만 검색하도록 하는 것

특징1:

```
Create view 테이블명_view as(  
생성하고 싶은 정보를 담은 select 테이블  
=가상의 테이블  
);  
Select * from 테이블명_view
```

특징2:

물리적인 테이블에 insert로 정보를 추가 저장하여도
View테이블에도 추가 데이터 반영

특징3:

Insert view 정보 추가 가능하지만
무결성의 위배 확인 필요
=>검색의 용도로 view 다용

- 별칭

별칭

(Alias)

Select 문에서
특정 컬럼에
붙이는 이름

용도 :

- 1) 컬럼/테이블명이
너무 길 때
- 2) 함수, 연산 사용해
컬럼명이 없을 때

특징 :

- 1) 30자까지
- 2) 맨 앞 특수문자,
공백은 "로 묶어줌

테이블

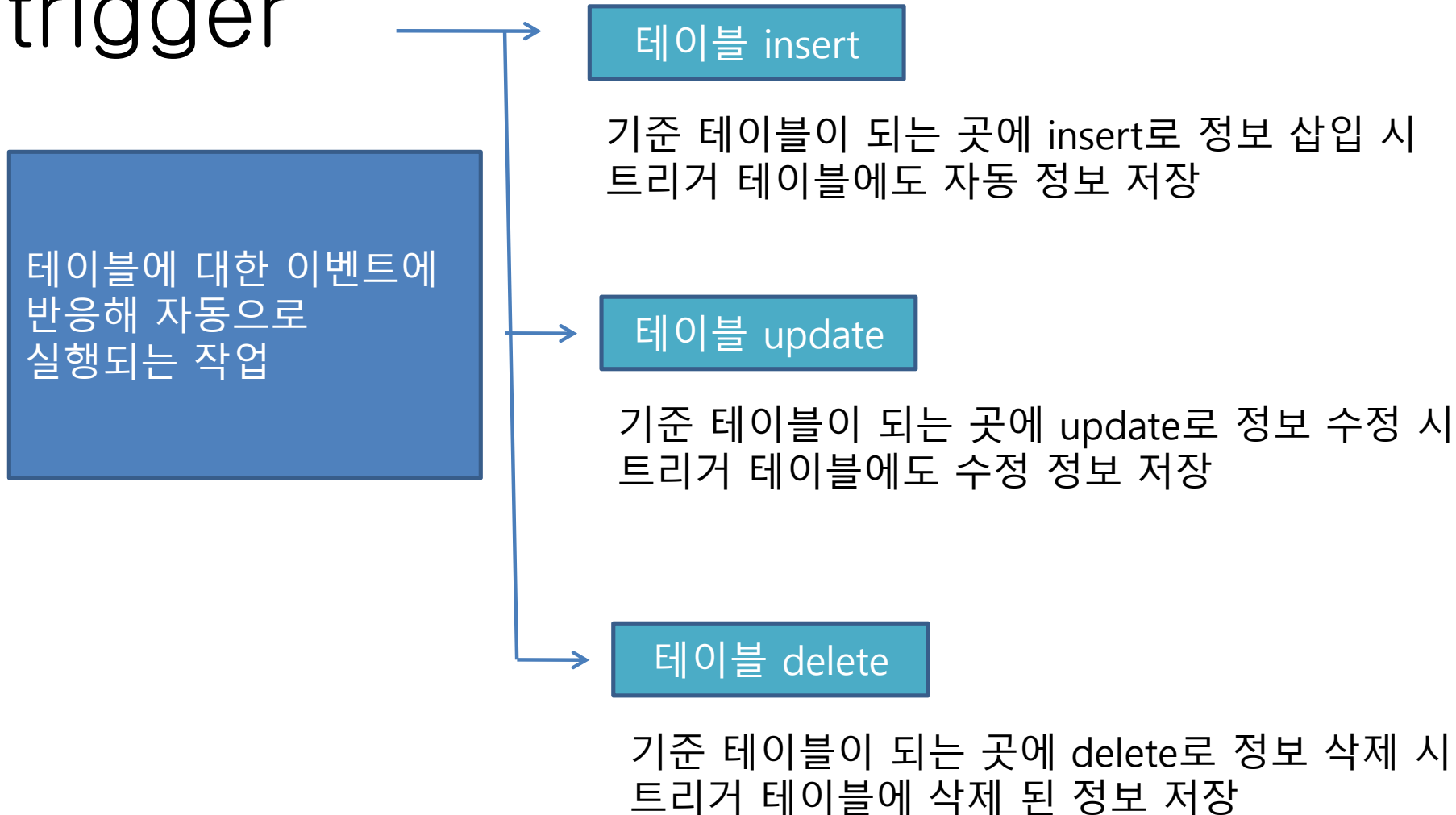
As 선언 X

컬럼

As 선언(생략 가능)

–Trigger

trigger



-프로시저

프로시저

(procedure)

정의 :
데이터베이스에 대한 일련의 작업을 정리한 절차를 저장하여 쿼리 블록을 하나의 함수처럼 실행하는 것

함수와의 차이점 :
함수는 특정 연산 수행 후 결과값 보여주지만 프로시저는 처리만 하고 결과값을 보여주지 않음

변수

사용x

In

실행에 필요한 데이터 값 입력

out

실행 후 호출하면 결과값 반환

In & out

감사합니다