

데이터베이스 예

데이터베이스 예

Prim ary key 개인 신상						
이름	주민번호	생년월일	관계	핸드폰번호	우편번호	주소
박찬호	730201-1034343	1973-02-01	후배	(017) 234-2342	(333-333)	구미시 도량동
이기자	680709-1078656	1968-07-09	동생	(018) 234-2342	(444-444)	대구시 달서구
최고야	880101-1027362	1988-01-01	고객	(016) 2342-2342	(222-222)	대구시 남산동
홍순이	600809-1987766	1960-08-09	친구	(019) 234-2342	(555-555)	대구시 검단동
홍길동	771214-1078767	1977-12-14	친구	(011) 234-2342	(111-111)	구미시 공단동

-- 개인신상 - 자동차 테이블 예제

```
CREATE TABLE 개인신상 ( 이름 CHAR(8) ,
                        주민번호 CHAR(14) ,
                        생년월일 CHAR(10) ,
                        관계 CHAR(10) ,
                        핸드폰번호 CHAR(14),
                        우편번호 CHAR(9) ,
                        주소 VARCHAR(75),
                        PRIMARY KEY (이름));
```

Foreign key 자동차				
이름	차종	차명	년식	구입일
이기자	트럭	포토	95	97-5
홍순이	승용차	에쿠스	01	01-8
홍길동	승용차	소나타	98	98-8

```
CREATE TABLE 자동차 ( 이름 CHAR(8),
                        차종 CHAR(6) ,
                        차명 VARCHAR(20) ,
                        년식 CHAR(4) ,
                        구입일 CHAR(7)
                        FOREIGN KEY (이름) REFERENCES 개인신상 );
```

DDL

- Create table
 - 그외 Create database, Create view , Create index
- Alter table
 - 예) Alter table 개인신상 add 이메일 char(10);
- Drop table
 - 예) Drop table 개인신상;

DML

- Select

- select 이름, 주민번호 from 개인신상 where 관계='친구';
- SELECT * FROM 개인신상 ORDER BY 이름 ASC ;
- SELECT * FROM 개인신상 ORDER BY 이름 DESC;
- SELECT 관계 FROM 개인신상;
- SELECT DISTINCT 관계 FROM 개인신상;
- SELECT COUNT(이름) FROM 개인신상 GROUP BY 관계;
- SELECT 이름, 주소 FROM 개인신상 GROUP BY 관계
HAVING COUNT(이름)>=2 ;
- SELECT 이름, 주소 FROM 개인신상 WHERE 이름 LIKE '홍%';
- SELECT SUBSTRING('ABCDE', 2, 3)
- SELECT * FROM 자동차 WHERE 년식 BETWEEN 94 AND 99

- insert into

- insert into 개인신상(이름, 주민번호) values ('홍길이', '901212-1212121');

- update

- update 개인신상 set 주민번호='901212-2212121' where 이름='홍길이';

- delete from

- delete from 개인신상 where 이름='홍길이';

커서

- 커서를 사용하는 이유는 한번 커서를 정의해서 만들면 추가로 SELECT할 필요 없이 바로 커서를 이용하여 레코드에 대한 데이터 처리 작업이 가능하기 때문이다.
- 커서 명령어
 - DECLARE : 커서를 정의 한다.
 - OPEN : 선언된 커서를 사용하기 위해 연다.
 - FETCH : 커서에서 레코드 이동을 위해 사용한다.
 - CLOSE : 커서를 닫는다.
 - DEALLOCATE : 정의한 커서를 제거한다.
- 예
 - DECLARE 학점처리 CURSOR
 - FOR SELECT 번호, 국어, 영어, 수학 FROM 성적
 - FOR UPDATE OF 총점, 평균

 - OPEN 학점처리

트랜잭션의 개념

- 한꺼번에 모두 수행되어야 할 일련의 데이터베이스 연산들
- 트랜잭션=하나 이상의 데이터베이스 연산(SQL 명령)
- 트랜잭션의 성질
 - ① 원자성(Atomicity) : 트랜잭션은 자기의 연산을 전부 실행만이 있지 일부 실행으로 트랜잭션의 기능 갖는 것은 아니다.
 - ② 일관성(Consistency) : 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
 - ③ 격리성(Isolation) : 트랜잭션이 실행 중에 생성하는 연산의 중간 결과는 다른 트랜잭션이 접근할 수 없다.
 - ④ 영속성(Durability) : 트랜잭션이 일단 그 실행을 성공적으로 완료하면 그 결과는 영속적이다.

•

카티션 곱

R			K		
A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b2	c2	b1	c1	d2
a3	b3	c3	b3	c3	d3

조인

좌측 외부조인

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a3	b3	c3	d3
a2	b2	c2	null

R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

$R \bowtie_R K$

K

B	C	D
b1	c1	d1
b1	c1	d2
b3	c3	d3

→

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a3	b3	c3	d3

우측 외부조인

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a3	b3	c3	d3

동등 조인

A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a1	b1	c1	b1	c1	d2
a3	b3	c3	b3	c3	d3

디비전(÷)

A	B
a1	b1
a2	b2
a3	b2
a1	b2
a4	b3

B
b1
b2