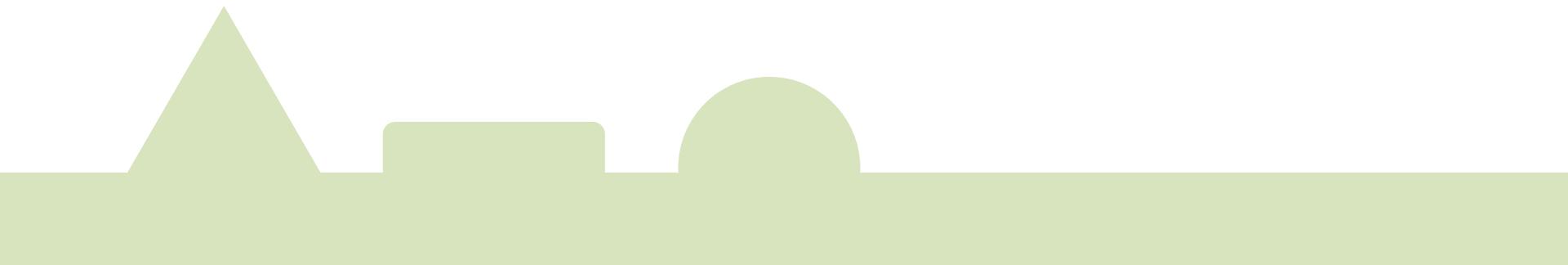




# 10. DML



# 목차

1. DML
2. INSERT
3. UPDATE
4. DELETE
5. MERGE



# 1. DML

## ❖ DML(Data Manipulation Language)

- 테이블에 데이터 검색, 삽입, 수정, 삭제하는 데 사용한다.
- Rollback이 가능하다.

종류	의미
SELECT	데이터를 검색하는 역할
INSERT	데이터를 추가하는 역할
UPDATE	데이터를 수정하는 역할
DELETE	데이터를 삭제하는 역할

## 2. INSERT(1)

- ❖ 테이블에 **새로운 데이터를 입력**하기 위해 사용하는 데이터 조작어.

**형식**

```
INSERT INTO 테이블명  
(컬럼 이름, ...)  
VALUES(컬럼 데이터, ...);
```

\* DEPT테이블의 구조만 복사한 DEPT01 테이블 준비

**예**

```
CREATE TABLE DEPT01  
AS  
SELECT * FROM DEPT WHERE 1=0;
```

## 2. INSERT(2)

### ❖ 데이터 추가

- 새로운 데이터를 추가하기 위해서 사용할 명령어 INSERT INTO ~ VALUES ~는 컬럼 명에 기술된 목록의 수와 VALUES 다음에 나오는 괄호에 기술한 값의 개수가 같아야 한다.

```
INSERT INTO DEPT01  
(DEPTNO, DNAME, LOC)  
VALUES(10, '영업부', '부산');
```

- ❖ 컬럼 DEPTNO에 10을, 컬럼 DNAME에는 '영업부'를, 컬럼 LOC에는 '부산'을 추가한다.

## 2.1 INSERT 구문에서 오류 발생의 예(1)

- ❖ 컬럼 명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호 안에 기술한 값의 개수가 적으면 에러가 발생한다.

예

```
INSERT INTO DEPT01  
(DEPTNO, DNAME, LOC)  
VALUES(10, '영업부');
```

- ❖ 컬럼 명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호에 기술한 값의 개수가 많으면 에러가 발생한다.

예

```
INSERT INTO DEPT01  
(DEPTNO, DNAME, LOC)  
VALUES(10, '영업부', '부산', 20);
```

## 2.1 INSERT 구문에서 오류 발생의 예(2)

- ❖ 컬럼의 이름이 잘못 입력되었을 때에도 에러가 발생한다.

예

```
INSERT INTO DEPT01  
(NUM, DNAME, LOC)  
VALUES(10, '영업부', '부산');
```

- ❖ 컬럼과 입력할 값의 데이터 타입이 서로 맞지 않을 경우에도 에러가 발생한다.

예

```
INSERT INTO DEPT01  
(DEPTNO, DNAME, LOC)  
VALUES(10, 영업부, '부산');
```

## 2.2 컬럼명을 생략한 INSERT 구문

- ❖ 테이블에 로우를 추가할 때 모든 컬럼에 모두 자료를 입력하는 경우에는 굳이 컬럼 목록을 기술하지 않아도 된다.
- ❖ 컬럼 목록이 생략되면 VALUES 절 다음의 값들이 테이블의 기본 컬럼 순서대로 입력된다.
- ❖ 테이블의 컬럼 순서는 CREATE TABLE로 테이블을 생성할 때의 순서를 따른다.
- ❖ 테이블의 기본 컬럼 순서는 DESC 문으로 조회했을 때 보여지는 순서이다.

예

```
INSERT INTO DEPT01  
VALUES(20, '인사부', '천안');
```

# 문제

1. 다음과 같은 구조로 SAM01 테이블을 생성한 후, 데이터를 추가하시오.

이름	널 ?	유형
----- -- -----		
EMPNO	NUMBER (4)	
ENAME	VARCHAR2 (10)	
JOB	VARCHAR2 (9)	
SAL	NUMBER (7, 2)	

EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000

## 2.3 NULL 값 삽입하는 방법(1)

- ❖ 데이터를 입력하는 시점에서 해당 컬럼 값을 모르거나 확정되지 않았을 경우에는 NULL값을 입력해야 한다.
- ❖ NULL 값 삽입은 암시적인 방법과 명시적인 방법이 있다.
- ❖ **암시적 방법**
  - 컬럼명 리스트에 컬럼을 생략하는 방법
  - 다른 컬럼은 값을 입력한다. 이렇게 생략한 컬럼에는 암시적으로 NULL 값이 할당된다.
- ❖ **명시적 방법**
  - VALUES 리스트에 명시적으로 NULL을 입력한다.

## 2.3 NULL 값 삽입하는 방법(2)

### ❖ 암시적 NULL 값 삽입

- 저장할 값을 명확하게 알고 있는 컬럼명만 명시적으로 기술한 후에 그에 매칭되는 값을 VALUES 절 다음에 기술합니다.

예

```
INSERT INTO DEPT01  
(DEPTNO, DNAME)  
VALUES(30, '영업부');
```

## 2.3 NULL 값 삽입하는 방법(3)

### ❖ 명시적 NULL 값 삽입

- 컬럼명을 명시적으로 기술하지 않으면 테이블이 갖고 있는 모든 컬럼에 값을 지정해야 한다.
- LOC 컬럼의 값이 결정되어 지지 않았더라도 반드시 값을 3개 지정해야 하기 때문에 명시적으로 VALUES 리스트에서 LOC 컬럼에 NULL을 입력한다.

예

```
INSERT INTO DEPT01  
VALUES(40, '전산부', NULL);
```

- NULL 대신 “ ” 을 사용할 수 있다.

예

```
INSERT INTO DEPT01  
VALUES(50, '', '부산');
```

## 2.3 NULL 값 삽입하는 방법(4)

- ❖ 부서 테이블에 컬럼이 NULL 값을 허용하는지 살펴보기 위해서 DESC 명령을 실행한다.
- ❖ DEPT 테이블의 DEPTNO 컬럼은 NOT NULL 제약조건이 지정되어 있다.

이름	널?	유형
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (14)
LOC		VARCHAR2 (13)

- ❖ NOT NULL 제약조건이 지정된 DEPTNO 컬럼은 NULL 값을 입력하지 못한다.
- ❖ 오라클이 제공하는 DEPT 테이블의 DEPTNO 컬럼에 NULL 값을 허용하지 못하도록 오라클 내부에서 이미 컬럼에 제약조건을 지정해 놓은 상태이다.
- ❖ 컬럼에 NULL 값을 허용하지 못하도록 하려면 컬럼에 제약조건을 지정해야 한다.

# 문제

2. 문제 1에서 생성한 SAM01 테이블에 다음과 같이 NULL 값을 갖는 행을 추가 하시오.

EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000
1030	베리	(null)	2500
1040	망고	(null)	25000

## 2.4 서브 쿼리로 데이터 삽입하기

- ❖ INSERT INTO 다음에 VALUES 절을 사용하는 대신에 서브 쿼리를 사용할 수 있다.
- ❖ 테이블에 있던 여러 행을 복사해서 다른 테이블에 삽입할 수 있다.
- ❖ INSERT 명령문에서 지정한 컬럼의 개수나 데이터 타입이 서브쿼리를 수행한 결과와 동일해야 한다는 주의점이 있다.
  - \* 빈 테이블 생성

예

```
CREATE TABLE DEPT02  
AS  
SELECT * FROM DEPT WHERE 1=0;
```

- ❖ 테이블 구조만을 복사해서 내용을 갖지 않는 테이블에 서브 쿼리 로우를 입력해 본다.

예

```
INSERT INTO DEPT02  
SELECT * FROM DEPT;
```

# 문제

3. 문제 1에서 생성한 SAM01 테이블에 서브 쿼리문을 이용하여 EMP에 저장된 사원 중 10번 부서 소속 사원의 정보를 추가하시오.

EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000
1030	베리	(null)	2500
1040	망고	(null)	25000
1007	이분세	부장	520
1010	이병현	과장	500
1014	조인성	사원	250

### 3. UPDATE(1)

- ❖ 테이블에 저장된 데이터를 수정하기 위해서 사용한다.

**UPDATE 테이블명**

**SET 컬럼1=값1, 컬럼2=값2, ...**

**WHERE 조건;**

- ❖ UPDATE 문은 기존의 행을 수정하는 것이다. 따라서 어떤 행의 데이터를 수정하는지 WHERE 절을 이용하여 조건을 지정한다.
- ❖ WHERE 절을 사용하지 않을 경우는 테이블에 있는 모든 행이 수정된다.
- ❖ 전체 행을 수정하려고 했던 것이 아니라면 큰 문제가 발생하므로 WHERE 절의 사용 유무를 신중히 판단 하여야한다.

### 3. UPDATE(2)

- ❖ ex) 모든 사원의 부서번호를 30으로 수정한다.

형식

```
UPDATE EMP01  
SET DEPTNO=30;
```

- ❖ ex) 모든 사원의 급여를 10% 인상시킨다.

형식

```
UPDATE EMP01  
SET SAL=SAL*1.1;
```

- ❖ ex) 모든 사원의 입사일을 오늘로 수정한다.

형식

```
UPDATE EMP01  
SET HIREDATE=SYSDATE;
```

### 3. UPDATE(3)

- ❖ UPDATE 문에 WHERE 절을 추가하면 테이블의 특정 행이 변경된다.
  - \* EMP01 테이블을 삭제하고 EMP 테이블의 구조와 데이터를 갖는 EMP01 테이블을 다시 생성한다.

# 문제

4. EMP01 테이블의 데이터 중 부서번호가 10번인 사원의 부서번호를 30번으로 수정하시오.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1001	김사랑	사원	1013	07/03/01	300	(null)	20
1002	한예슬	대리	1005	07/04/02	250	80	30
1003	오지호	과장	1005	05/02/10	500	100	30
1004	이병현	부장	1008	03/09/02	600	(null)	20
1005	신농협	과장	1005	05/04/07	450	200	30
1006	장동건	부장	1008	03/10/09	480	(null)	30
1007	이문세	부장	1008	04/01/08	520	(null)	30
1008	감우성	차장	1003	04/03/08	500	0	30
1009	안성기	사장	(null)	96/10/04	1000	(null)	20
1010	이병현	과장	1003	05/04/07	500	(null)	30
1011	조향기	사원	1007	07/03/01	280	(null)	30
1012	강혜정	사원	1006	07/08/09	300	(null)	20
1013	박승훈	부장	1003	02/10/09	560	(null)	20
1014	조인성	사원	1006	07/11/09	250	(null)	30

# 문제

5. EMP01 테이블의 직원들의 급여 중 400이상인 직원만 급여를 10% 인상하시오.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1001	김사랑	사원	1013	07/03/01	300	(null)	20
1002	한예슬	대리	1005	07/04/02	250	80	30
1003	오지호	과장	1005	05/02/10	550	100	30
1004	이병현	부장	1008	03/09/02	660	(null)	20
1005	신농협	과장	1005	05/04/07	495	200	30
1006	장농건	부장	1008	03/10/09	528	(null)	30
1007	이문세	부장	1008	04/01/08	572	(null)	30
1008	감우성	차장	1003	04/03/08	550	0	30
1009	안성기	사장	(null)	96/10/04	1100	(null)	20
1010	이병현	과장	1003	05/04/07	550	(null)	30
1011	조향기	사원	1007	07/03/01	280	(null)	30
1012	강혜정	사원	1006	07/08/09	300	(null)	20
1013	박충훈	부장	1003	02/10/09	616	(null)	20
1014	조인성	사원	1006	07/11/09	250	(null)	30

# 문제

6. SAM01 테이블에 저장된 직원 중 급여가 10000 이상인 직원들의 급여를 5000씩 삭감하시오.

EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000
1030	베리	(null)	2500
1040	망고	(null)	20000
1007	이분세	부장	520
1010	이병현	과장	500
1014	조인성	사원	250

### 3.1 다중 컬럼 값 변경

- ❖ 테이블에서 하나의 컬럼이 아닌 여러 개 컬럼의 값을 변경하려면 기존 SET 절에 콤마(,)를 추가하고 컬럼=값을 추가 기술하면 된다.
- ❖ ex) 한예슬 직원의 부서번호는 10으로, 직급은 '과장'으로 한꺼번에 수정하도록 한다.

예

```
UPDATE EMP01  
SET DEPTNO=10, JOB='과장'  
WHERE ENAME='한예슬';
```

- ❖ ex) 오지호 직원의 입사일자는 오늘로, 급여를 600으로, 커미션을 200으로 수정하도록 한다.

예

```
UPDATAE EMP01  
SET HIREDATE=SYSDATE, SAL=600, COMM=200  
WHERE ENAME='오지호';
```

## 3.2 서브 쿼리를 이용한 데이터 수정

- ❖ ex) 20번 부서의 지역명을 40번 부서의 지역명으로 변경하기 위해서 서브 쿼리문을 사용한다.

예

```
UPDATE DEPT01  
SET LOC=(SELECT LOC FROM DEPT01  
          WHERE DEPT=40)  
WHERE DEPTNO=20;
```

DEPTNO	DNAME	LOC
10	경리부	서울
20	인사부	수원
30	영업부	용인
40	전산부	수원

# 문제

7. EMP 테이블의 특정 컬럼과 데이터로 구성된 SAM02 테이블을 생성하시오.

이름	널?	유형
ENAME		VARCHAR2 (10)
SAL		NUMBER (7, 2)
HIREDATE		DATE
DEPTNO		NUMBER (2)

8. SAM02 테이블 데이터 중 '인천'에 위치한 부서 소속 직원들의 급여를 100 인상하시오.

ENAME	SAL	HIREDATE	DEPTNO
김사랑	400	07/03/01	20
한예슬	250	07/04/02	30
오지호	500	05/02/10	30
이병현	700	03/09/02	20
신농협	450	05/04/07	30
장동건	480	03/10/09	30
이문세	520	04/01/08	10
감우성	500	04/03/08	30
안성기	1100	96/10/04	20
이병현	500	05/04/07	10
조향기	280	07/03/01	30
강혜정	400	07/08/09	20
박충훈	660	02/10/09	20
조인성	250	07/11/09	10

## 4. DELETE(1)

- ❖ 테이블에 저장된 데이터를 삭제하기 위해서 사용한다.

**DELETE FROM 테이블명  
WHERE 조건;**

- ❖ DELETE 문은 테이블의 기존 행을 삭제하며 특정한 로우(행)을 삭제하기 위해서 WHERE 절을 이용하여 조건을 지정한다.
- ❖ 만약 DELETE 문에 WHERE 절을 사용하지 않을 경우 테이블에 있는 모든 행이 삭제되므로 매우 신중하게 명령문을 사용해야 한다.

## 4. DELETE(2)

### ❖ DELETE, TRUNCATE, DROP 차이점

- DELETE : 데이터는 지워지지만 테이블의 용량은 줄어들지 않는다. 원하는 데이터만 지울 수 있고, 잘못 삭제한 경우 되돌릴 수 있다(Rollback)
- TRUNCATE : 일반 데이터 및 인덱스 등도 모두 삭제되고, 테이블의 용량도 줄어든다. 원하는 데이터를 선택해서 지울 수 없으며, 삭제 후 되돌릴 수 없다.
- DROP : 테이블 전체를 삭제. 삭제 후 되돌릴 수 없다.

### ❖ ex) DELETE 문으로 DEPT01 테이블의 모든 행을 삭제한다.

예

**DELETE FROM DEPT01;**

DEPTNO	DNAME	LOC
10	경리부	서울
20	인사부	수원
30	영업부	용인
40	전산부	수원



DEPTNO	DNAME	LOC

# 문제

9. SAM01 테이블에서 직급이 정해지지 않은 직원을 삭제하시오.

EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000
1030	베리	(null)	2500
1040	망고	(null)	20000
1007	이문세	부장	520
1010	이병현	과장	500
1014	조인성	사원	250



EMPNO	ENAME	JOB	SAL
1000	사과	경찰	1000
1010	바나나	간호사	1500
1020	오렌지	의사	2000
1007	이문세	부장	520
1010	이병현	과장	500
1014	조인성	사원	250

## 4.1 서브 쿼리를 이용한 데이터 삭제

- ❖ EMP01 테이블에서 부서명이 '영업부' 인 직원을 모두 삭제해보도록 한다.
- ❖ 부서명은 부서 테이블에 기록되어 있으므로 부서 테이블에서 부서명이 '영업부' 인 부서 번호부터 알아내야 한다.
- ❖ 이렇게 알아낸 부서번호를 사원 테이블에 적용하기 위해 서브쿼리를 이용한다.

# 문제

10. SAM02 테이블에서 '영업부' 부서 소속 직원들만 삭제하시오.

ENAME	SAL	HIREDATE	DEPTNO
김사랑	300	07/03/01	20
한예슬	250	07/04/02	30
오지호	500	05/02/10	30
이병현	600	03/09/02	20
신농협	450	05/04/07	30
장농건	480	03/10/09	30
이문세	520	04/01/08	10
감우성	500	04/03/08	30
안성기	1000	96/10/04	20
이병현	500	05/04/07	10
조향기	280	07/03/01	30
강혜정	300	07/08/09	20
박충훈	560	02/10/09	20
조인성	250	07/11/09	10



ENAME	SAL	HIREDATE	DEPTNO
김사랑	300	07/03/01	20
이병현	600	03/09/02	20
이문세	520	04/01/08	10
안성기	1000	96/10/04	20
이병현	500	05/04/07	10
강혜정	300	07/08/09	20
박충훈	560	02/10/09	20
조인성	250	07/11/09	10

## 5. MERGE(1)

### ❖ MERGE

- 합병이라는 의미로 구조가 같은 두 개의 테이블을 하나의 테이블로 합치는 기능을 한다.
- 수행하는 테이블에 기존에 존재하는 행이 있다면 새로운 값으로 갱신(UPDATE)되고, 존재하지 않으면 새로운 행으로 추가(INSERT)된다.
- 오라클 9i 버전 이상부터 사용이 가능하다.

형식

**MERGE INTO [테이블명 or 뷰명]  
USING [조회 서브쿼리]**

## 5. MERGE(2)

형식

**MERGE INTO** [테이블명 or 뷰명] – Update 또는 Insert할 테이블 명 혹은 뷰  
**USING** [조회 서브쿼리] (INTO절의 동일 테이블이라면 dual 사용)  
**ON**[INTO절과 USING의 조인 조건] (Update/Insert 조건은 ON 절에서 결정)  
**WHEN MATCHED THEN** – 일치되는 경우 UPDATE  
**UPDATE SET** – ON절에 사용한 컬럼은 UPDATE가 불가능  
[컬럼1]=[값1],  
[컬럼2]=[값2]...  
**WHEN NOT MATCHED THEN** – 일치 안되는 경우 INSERT  
**INSERT**(컬럼1, 컬럼2...)  
**VALUES**(값1, 값2...)

## 5. MERGE(2)

- ❖ ex) MERGE문에 의해서 UPDATE 되고 INSERT 되는지 살펴보도록 한다.
- ❖ 1. 새로운 테이블 EMP01을 생성한다.

예

```
CREATE TABLE EMP01  
AS  
SELECT * FROM EMP;
```

- ❖ 2. 직급이 '과장'인 사원들로만 구성된 EMP02 테이블을 생성한다.

예

```
CREATE TABLE EMP02  
AS  
SELECT * FROM EMP  
WHERE JOB='과장';
```

## 5. MERGE(3)

- 3. MERGE 문에 의해서 기존에 존재하는 행이 갱신(UPDATE) 되는지 살펴보기 위해서 새로 생성된 EMP02 테이블의 JOB을 '인턴'로 변경한다.

예

```
UPDATE EMP02  
SET JOB='인턴';
```

- 4. MERGE 문에 의해서 새로운 행이 추가(INSERT) 되는지 살펴보기 위해서 새로운 행을 추가한다.

예

```
INSERT INTO EMP02  
VALUES(2000, '이건희', '회장', NULL, '2009/01/12', 1200, 10,  
20);
```

## 5. MERGE(4)

- ❖ 5. EMP01 테이블에 EMP02 테이블을 합병해본다.

예

```
MERGE INTO EMP01
  USING EMP02
ON(EMP01.EMPNO=EMP02.EMPNO)
WHEN MATCHED THEN
  UPDATE SET
    EMP01.ENAME=EMP02.ENAME,
    EMP01.JOB=EMP02.JOB,
    EMP01.MGR=EMP02.MGR,
    EMP01.HIREDATE=EMP02.HIREDATE,
    EMP01.SAL=EMP02.SAL,
    EMP01.COMM=EMP02.COMM,
    EMP01.DEPTNO=EMP02.DEPTNO
WHEN NOT MATCHED THEN
  INSERT VALUES(EMP02.EMPNO, EMP02.ENAME, EMP02.JOB,
EMP02.MGR, EMP02.HIREDATE, EMP02.SAL, EMP02.COMM,
EMP02.DEPTNO);
```

## 5. MERGE(5)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1001	김사랑	사원	1013	07/03/01	300	(null)	20
1002	한예슬	대리	1005	07/04/02	250	80	30
1003	오지호	인턴	1005	05/02/10	500	100	30
1004	이병현	부장	1008	03/09/02	600	(null)	20
1005	신농협	인턴	1005	05/04/07	450	200	30
1006	장농건	부장	1008	03/10/09	480	(null)	30
1007	이문세	부장	1008	04/01/08	520	(null)	10
1008	감우성	차장	1003	04/03/08	500	0	30
1009	안성기	사장	(null)	96/10/04	1000	(null)	20
1010	이병현	인턴	1003	05/04/07	500	(null)	10
1011	조향기	사원	1007	07/03/01	280	(null)	30
1012	강혜정	사원	1006	07/08/09	300	(null)	20
1013	박충훈	부장	1003	02/10/09	560	(null)	20
1014	조인성	사원	1006	07/11/09	250	(null)	10
2000	이건희	회장	(null)	09/01/12	1200	10	20