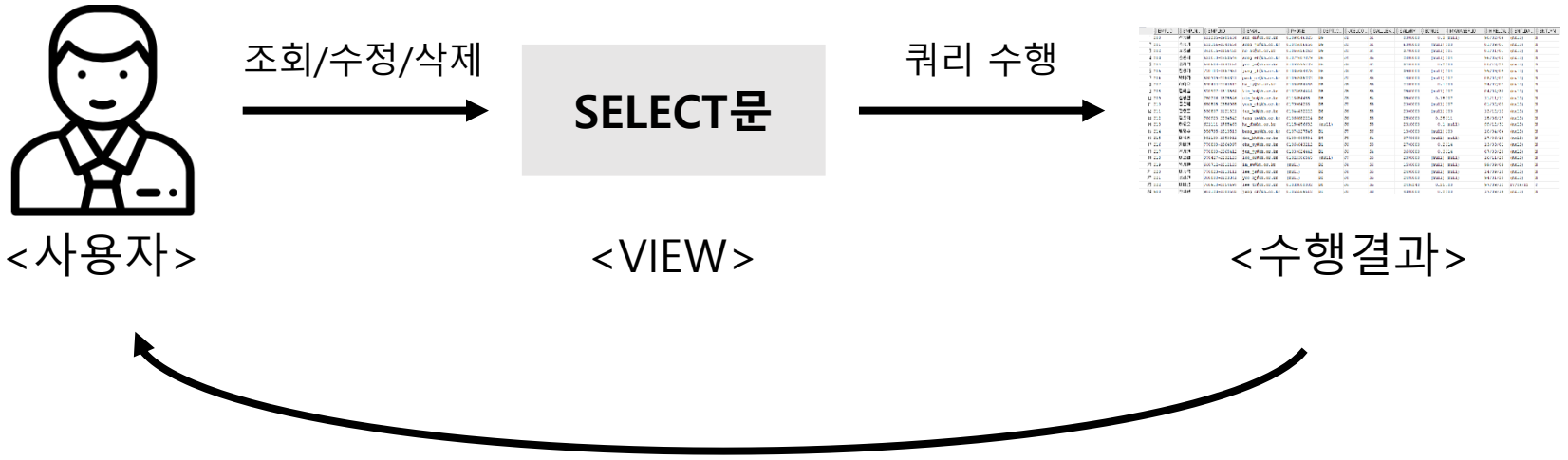
ORACLE OBJECT VIEW

▶ VIEW

SELECT 쿼리의 실행 결과를 화면에 저장한 논리적 가상 테이블
실제 테이블과는 다르게 실질적 데이터를 저장하고 있진 않지만
사용자는 테이블을 사용하는 것과 동일하게 사용 가능



▶ VIEW

✓ 예시1

```
CREATE OR REPLACE VIEW V_EMPLOYEE
AS SELECT EMP_ID, EMP_NAME, DEPT_TITLE, NATIONAL_NAME
    FROM EMPLOYEE
    LEFT JOIN DEPARTMENT ON(DEPT_ID = DEPT_CODE)
    LEFT JOIN LOCATION ON(LOCATION_ID = LOCAL_CODE)
    LEFT JOIN NATIONAL USING(NATIONAL_CODE);

SELECT * FROM V_EMPLOYEE;
```

	EMP_ID	EMP_NAME	DEPT_TITLE	NATIONAL_NAME
1	217	전지연	인사관리부	한국
2	216	차태연	인사관리부	한국
3	214	방명수	인사관리부	한국
4	900	장채현	인사관리부	한국
5	221	유하진	회계관리부	한국
6	220	이중석	회계관리부	한국
7	219	임시환	회계관리부	한국
8	202	노옹철	총무부	한국
9	201	송종기	총무부	한국
10	200	선동일	총무부	한국
11	215	대북혼	해외영업1부	일본
12	210	윤은해	해외영업1부	일본
13	209	심봉선	해외영업1부	일본
14	208	김해술	해외영업1부	일본
15	207	하이유	해외영업1부	일본
16	206	박나라	해외영업1부	일본
17	205	정중하	해외영업2부	중국
18	204	유재식	해외영업2부	중국
19	203	송은희	해외영업2부	중국
20	222	이태림	기술지원부	러시아
21	212	장쯔위	기술지원부	러시아
22	211	전형돈	기술지원부	러시아
23	218	이오리	(null)	(null)
24	213	하동운	(null)	(null)

▶ VIEW

✓ 예시2

```
CREATE OR REPLACE VIEW V_EMP_JOB(사번, 이름, 직급, 성별, 근무년수)
AS SELECT EMP_ID, EMP_NAME, JOB_NAME,
          DECODE(SUBSTR(EMP_NO, 8, 1), 1, '남', 2, '여'),
          EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM HIRE_DATE)
    FROM EMPLOYEE
    JOIN JOB USING(JOB_CODE);
```

* 서브쿼리의 SELECT절에 함수가 사용된 경우 반드시 별칭 지정

	사번	이름	직급	성별	근무년수
1	900	장채현	인턴	남	0
2	200	선동일	대표	남	28
3	201	송종기	부사장	남	17
4	202	노옹철	부사장	남	17
5	203	송은희	차장	여	22
6	204	유재식	부장	남	18
7	205	정중하	부장	남	19
8	206	박나라	사원	여	10
9	207	하이유	과장	여	24
10	208	김해술	과장	남	14
11	209	심봉선	부장	남	7
12	210	윤은해	사원	여	17
13	211	전형돈	대리	남	6
14	212	장쯔위	대리	여	3
15	213	하동운	대리	남	19
16	214	방명수	사원	남	8
17	215	대북혼	과장	남	1
18	216	차태연	대리	남	5
19	217	전지연	대리	여	11
20	218	이오리	사원	여	2
21	219	임시환	차장	남	19
22	220	이중석	차장	남	4
23	221	유하진	차장	남	24
24	222	이태림	대리	여	21

▶ VIEW

✓ 예시3

```
CREATE OR REPLACE VIEW V_JOB
AS SELECT JOB_CODE, JOB_NAME
     FROM JOB;

INSERT INTO V_JOB VALUES('J8', '인턴');

SELECT  * FROM V_JOB;
```

	JOB_CODE	JOB_NAME
1	J1	대표
2	J2	부사장
3	J3	부장
4	J4	차장
5	J5	과장
6	J6	대리
7	J7	사원
8	J8	인턴

```
SELECT * FROM JOB;
```

	JOB_CODE	JOB_NAME
1	J1	대표
2	J2	부사장
3	J3	부장
4	J4	차장
5	J5	과장
6	J6	대리
7	J7	사원
8	J8	인턴

* 생성된 뷰를 가지고 DML구문(INSERT, UPDATE, DELETE) 사용 가능
* 생성된 뷰에 요청한 DML구문이 베이스 테이블도 변경함

▶ DML명령어로 VIEW 조작이 불가능한 경우

1. 뷰 정의에 포함되지 않은 컬럼을 조작하는 경우
2. 뷰에 포함되지 않은 컬럼 중에 베이스가 되는 컬럼이 NOT NULL 제약조건이 지정된 경우
3. 산술 표현식으로 정의된 경우
4. 그룹함수나 GROUP BY절을 포함한 경우
5. DISTINCT를 포함한 경우
6. JOIN을 이용해 여러 테이블을 연결한 경우

▶ DML명령어로 VIEW 조작이 불가능한 경우

1. 뷰 정의에 포함되지 않은 컬럼을 조작하는 경우

```
CREATE OR REPLACE VIEW V_JOB2
AS SELECT JOB_CODE
      FROM JOB;
```

```
INSERT INTO V_JOB2 VALUES('J8', '인턴');
```

* 뷰 정의에 포함되지 않은 컬럼을 INSERT/UPDATE/DELETE하는 경우 에러 발생

	JOB_CODE
1	J1
2	J2
3	J3
4	J4
5	J5
6	J6
7	J7

```
오류 보고 -
SQL 오류: ORA-00913: too many values
00913. 00000 -  "too many values"
*Cause:
*Action:
```

2. 뷰에 포함되지 않은 컬럼 중에 베이스가 되는 테이블 컬럼이 NOT NULL제약조건이 지정된 경우

```
CREATE OR REPLACE VIEW V_JOB3
AS SELECT JOB_NAME
      FROM JOB;
```

```
INSERT INTO V_JOB3 VALUES('인턴');
```

* 뷰에 포함되지 않은 NOT NULL제약조건이 있는 컬럼이 존재하면 INSERT시 에러 발생
* UPDATE/DELETE는 가능

	JOB_NAME
1	대표
2	부사장
3	부장
4	차장
5	과장
6	대리
7	사원

```
오류 보고 -
SQL 오류: ORA-01400: cannot insert NULL into ("EMPLOYEE"."JOB"."JOB_CODE")
01400. 00000 -  "cannot insert NULL into (%s)"
*Cause:    An attempt was made to insert NULL into previously listed objects.
*Action:   These objects cannot accept NULL values.
```

▶ DML명령어로 VIEW 조작이 불가능한 경우

3. 산술 표현식으로 정의된 경우

```
CREATE OR REPLACE VIEW EMP_SAL
AS SELECT EMP_ID, EMP_NAME, SALARY,
          (SALARY + (SALARY*NVL(BONUS, 0)))*12 연봉
   FROM EMPLOYEE;

INSERT INTO EMP_SAL VALUES(800, '정진훈', 3000000, 36000000);
```

* 뷰에 산술 계산식이 포함된 경우 INSERT/UPDATE 시 에러 발생
* 단 DELETE는 가능

```
오류 보고 -
SQL 오류: ORA-01733: virtual column not allowed here
01733. 00000 -  "virtual column not allowed here"
*Cause:
*Action:
```

	EMP_ID	EMP_NAME	SALARY	연봉
1	200	선동일	8000000	124800000
2	201	송종기	6000000	72000000
3	202	노옹철	3700000	44400000
4	203	송은희	2800000	33600000
5	204	유재식	3400000	48960000
6	205	정중하	3900000	46800000
7	206	박나라	1800000	21600000
8	207	하이유	2200000	29040000
9	208	김해술	2500000	30000000
10	209	심봉선	3500000	48300000
11	210	윤은해	2000000	24000000
12	211	전형돈	2000000	24000000
13	212	장쯔위	2550000	38250000
14	213	하동운	2320000	30624000
15	214	방명수	1380000	16560000
16	215	대북혼	3760000	45120000
17	216	차태연	2780000	40032000
18	217	전지연	3660000	57096000
19	218	이오리	2890000	34680000
20	219	임시환	1550000	18600000
21	220	이중석	2490000	29880000
22	221	유하진	2480000	29760000
23	222	이태림	2436240	39467088
24	900	장채현	4300000	61920000

▶ DML명령어로 VIEW 조작이 불가능한 경우

4. 그룹함수 또는 GROUP BY절을 포함한 경우

CREATE OR REPLACE VIEW V_GROUPDEPT

AS **SELECT** DEPT_CODE, SUM(SALARY) 합계, AVG(SALARY) 평균

FROM EMPLOYEE

GROUP BY DEPT_CODE;

	DEPT_CODE	합계	평균
1	(null)	5210000	2605000
2	D1	12120000	3030000
3	D9	17700000	5900000
4	D5	15760000	2626666
5	D6	10100000	3366666
6	D2	6520000	2173333
7	D8	6986240	2328746

INSERT INTO V_GROUPDEPT **VALUES**('D10', 6000000, 4000000);

DELETE FROM V_GROUPDEPT **WHERE** DEPT_CODE = 'D1';

* 그룹함수 또는 GROUP BY를 사용한 경우 INSERT/UPDATE/DELETE 시 에러 발생

```
오류 보고 -
SQL 오류: ORA-01733: virtual column not allowed here
01733. 00000 -  "virtual column not allowed here"
*Cause:
*Action:
```

```
오류 보고 -
SQL 오류: ORA-01732: data manipulation operation not legal on this view
01732. 00000 -  "data manipulation operation not legal on this view"
*Cause:
*Action:
```

▶ DML명령어로 VIEW 조작이 불가능한 경우

5. DISTINCT를 포함한 경우

```
CREATE OR REPLACE VIEW V_DT_EMP
AS SELECT DISTINCT JOB_CODE
      FROM EMPLOYEE;
```

```
INSERT INTO V_DT_EMP VALUES('J9');
DELETE FROM V_DT_EMP WHERE JOB_CODE = 'J1';
```

* DISTINCT를 사용한 경우 INSERT/UPDATE/DELETE 시 에러 발생

	JOB_CODE
1	J2
2	J7
3	J3
4	J6
5	J5
6	J8
7	J1
8	J4

```
오류 보고 -
SQL 오류: ORA-01732: data manipulation operation not legal on this view
01732. 00000 -  "data manipulation operation not legal on this view"
*Cause:
*Action:

오류 보고 -
SQL 오류: ORA-01732: data manipulation operation not legal on this view
01732. 00000 -  "data manipulation operation not legal on this view"
*Cause:
*Action:
```

▶ DML명령어로 VIEW 조작이 불가능한 경우

6. JOIN을 이용해 여러 테이블을 연결한 경우

```
CREATE OR REPLACE VIEW V_JOINEMP
AS SELECT EMP_ID, EMP_NAME, DEPT_TITLE
    FROM EMPLOYEE
    JOIN DEPARTMENT ON (DEPT_CODE = DEPT_ID);
```

```
INSERT INTO V_JOINEMP VALUES(888, '조세오', '인사관리부');
```

* 뷰 정의 시 JOIN을 사용한 경우 INSERT/UPDATE 시 에러 발생
* 단 DELETE는 가능

```
오류 보고 -
SQL 오류: ORA-01776: cannot modify more than one base table through a join view
01776. 00000 -  "cannot modify more than one base table through a join view"
*Cause:    Columns belonging to more than one underlying table were either
           inserted into or updated.
*Action:   Phrase the statement as two or more separate statements.
```

	EMP_ID	EMP_NAME	DEPT_TITLE
1	200	선동일	총무부
2	201	송종기	총무부
3	202	노옹철	총무부
4	203	송은희	해외영업2부
5	204	유재식	해외영업2부
6	205	정중하	해외영업2부
7	206	박나라	해외영업1부
8	207	하이유	해외영업1부
9	208	김해술	해외영업1부
10	209	심봉선	해외영업1부
11	210	윤은해	해외영업1부
12	211	전형돈	기술지원부
13	212	장쯔위	기술지원부
14	214	방명수	인사관리부
15	215	대북혼	해외영업1부
16	216	차태연	인사관리부
17	217	전지연	인사관리부
18	219	임시환	회계관리부
19	220	이중석	회계관리부
20	221	유하진	회계관리부
21	222	이태림	기술지원부
22	900	장채현	인사관리부

▶ VIEW 구조

뷰 정의 시 사용한 쿼리 문장이 TEXT컬럼에 저장되어 있으며

뷰가 실행될 때는 TEXT에 기록된 SELECT에 문장이 다시 실행되면서 결과를 보여주는 구조임

SELECT * **FROM** USER_VIEWS;

	VIEW_NAME	TEXT_LENGTH	TEXT	TYPE_TEXT_LENGTH	TYPE_TEXT	OID_TEXT_LENGTH	OID_TEXT	VIEW_TYPE_OWNER	VIEW_TYPE	SUPERVIE
1	V_EMPLOYEE	199	SELECT EMP_ID, EMP_NAME, DEPT_TITLE, NATIONAL_NA...	(null)	(null)	(null)	(null)	(null)	(null)	(null)
2	V_EMP_JOB	226	SELECT EMP_ID, EMP_NAME, JOB_N...	(null)	(null)	(null)	(null)	(null)	(null)	(null)
3	V_EMP	162	SELECT "EMP_ID","EMP_NAME","EMP_NO","EMAIL","PHO...	(null)	(null)	(null)	(null)	(null)	(null)	(null)
4	V_JOB	34	SELECT JOB_CODE, JOB_NAMEFROM JOB	(null)	(null)	(null)	(null)	(null)	(null)	(null)
5	V_JOB2	24	SELECT JOB_CODEFROM JOB	(null)	(null)	(null)	(null)	(null)	(null)	(null)
6	V_JOB3	24	SELECT JOB_NAMEFROM JOB	(null)	(null)	(null)	(null)	(null)	(null)	(null)
7	V_DT_EMP	38	SELECT DISTINCT JOB_CODE FROM EMPLOYEE	(null)	(null)	(null)	(null)	(null)	(null)	(null)
8	V_GROUPDEPT	101	SELECT DEPT_CODE, SUM(SALARY) 합계, FLOOR(AVG(SAL...	(null)	(null)	(null)	(null)	(null)	(null)	(null)
9	EMP_SAL	93	SELECT EMP_ID, EMP_NAME, SALARY, (SALARY + (SALA...	(null)	(null)	(null)	(null)	(null)	(null)	(null)
10	V_JOINEMP	90	SELECT EMP_ID, EMP_NAME, DEPT_TITLEFROM EMPLOYEE...	(null)	(null)	(null)	(null)	(null)	(null)	(null)

▶ VIEW 옵션

1. OR REPLACE 옵션

    생성한 뷰가 존재하면 뷰를 갱신함

2. FORCE/NOFORCE 옵션

    FORCE옵션은 기본 테이블이 존재하지 않더라도 뷰 생성

    NOFORCE 옵션이 기본 값으로 지정되어 있음

3. WITH CHECK OPTION 옵션

    옵션을 설정한 컬럼의 값을 수정 불가능하게 함(삭제는 가능)

4. WITH READ ONLY 옵션

    뷰에 대해 조회만 가능하고 삽입, 수정, 삭제 등은 불가능하게 함

▶ 인라인 뷰(INLINE-VIEW)

일반적으로 FROM절에 사용된 서브쿼리의 결과 화면에 별칭을 붙인 것을 말함

FROM절에 서브쿼리를 직접 사용해도 되고 따로 뷰를 생성 후 FROM절에 생성한 뷰를 사용해도 됨