뷰

자바 강의실

목차

1. 뷰 생성

- 뷰의 개념
  - 뷰(View)는 한마디로 물리적인 테이블을 근거한 논리적인 가상 테이블이라고 정의할 수 있습니다.
  - 뷰를 가상 테이블이라고 하는 이유를 살펴보겠습니다.
  - 가상이란 단어는 실질적으로 데이터를 저장하고 있지 않기 때문에 붙인 것이고, 테이블이란 단어는 실질적으로 데이터를 저장하고 있지 않더라도 사용자는 마치 테이블을 사용하는 것과 동일하게 뷰를 사용할 수 있기 때문에 붙인 것입니다.
  - 뷰는 기본 테이블에서 파생된 객체로서 기본 테이블에 대한 하나의 쿼리문입니다.
  - 뷰(View)란 '보다'란 의미를 갖고 있는 점을 감안해 보면 알 수 있듯이 실제 테이블에 저장된 데이터를 뷰를 통해서 볼 수 있도록 합니다.
  - 사용자에게 주어진 뷰를 통해서 기본 테이블을 제한적으로 사용하게 됩니다.

1. 뷰 생성

- 뷰의 기본 테이블
  - 뷰는 이미 존재하고 있는 테이블에 제한적으로 접근하도록 합니다.
  - 뷰를 생성하기 위해서는 실질적으로 데이터를 저장하고 있는 물리적인 테이블이 존재해야 하는데 이 테이블을 기본 테이블이라고 합니다.
  - 우선 시스템에서 제공하는 DEPARTMENTS 테이블과 EMPLOYEES 테이블의 내용이 변경되는 것을 막기 위해서 테이블의 내용을 복사한 새로운 테이블을 생성한 후에 이를 기본 테이블로 사용합시다.

1. 뷰 생성

예: 뷰의 기본테이블

질의 : 뷰의 기본 테이블을 생성합시다.

DEPARTMENTS_COPY를 DEPARTMENTS 테이블의 복사본으로 생성합시다.

EMPLOYEES 테이블의 복사본으로 EMPLOYEES_COPY를 생성합시다.

```
CREATE TABLE DEPARTMENTS_COPY
AS
SELECT * FROM DEPARTMENTS;
```

1. 뷰 생성

- 뷰 정의하기
  - 뷰를 생성하여 자주 사용되는 SELECT 문을 간단하게 접근하는 방법을 학습해 봅시다. 다음은 뷰를 생성하기 위한 기본 형식입니다.

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW
view_name [(alias, alias, alias, ...)]
AS subquery
[WITH CHECK OPTION]
[WITH READ ONLY];
```

  - 테이블을 생성하기 위해서 CREATE TABLE 로 시작하지만, 뷰를 생성하기 위해서는 CREATE VIEW로 시작합니다. AS 다음은 마치 서브 쿼리문과 유사합니다.
  - subquery에는 우리가 지금까지 사용하였던 SELECT 문을 기술하면 됩니다.

1. 뷰 생성

- **뷰 정의하기**
  - CREATE OR RELPACE VIEW
    - **뷰를 만들 때** CREATE OR RELPACE VIEW **대신 그냥** CREATE VIEW**만 사용해도 됩니다**.
    - **그러나 그냥** CREATE VIEW**를 통해 만들어진 뷰의 구조를 바꾸려면 뷰를 삭제하고 다시 만들어야 되는 반면**, CREATE OR REPLACE VIEW**는 새로운 뷰를 만들 수 있을 뿐만 아니라 기존에 뷰가 존재하더라도 삭제하지 않고 새로운 구조의 뷰로 변경**(REPLACE)**할 수 있습니다**.
    - **그래서 대부분 뷰를 만들 때는** CREATE VIEW **대신** CREATE OR REPLACE VIEW**를 사용하는 편입니다**.
  - FORCE
    - FORCE**를 사용하면, 기본 테이블의 존재 여부에 상관없이 뷰를 생성합니다**.

1. 뷰 생성

- 뷰 정의하기
  - WITH CHECK OPTION
    - WITH CHECK OPTION을 사용하면, 해당 뷰를 통해서 볼 수 있는 범위 내에서만 UPDATE 또는 INSERT가 가능합니다.
  - WITH READ ONLY
    - WITH READ ONLY를 사용하면 해당 뷰를 통해서는 SELECT만 가능하며 INSERT/UPDATE/DELETE를 할 수 없게 됩니다.
    - 만약 이것을 생략한다면, 뷰를 사용하여 추가, 수정, 삭제 (INSERT/UPDATE/DELETE)가 모두 가능합니다.

1. 뷰 생성

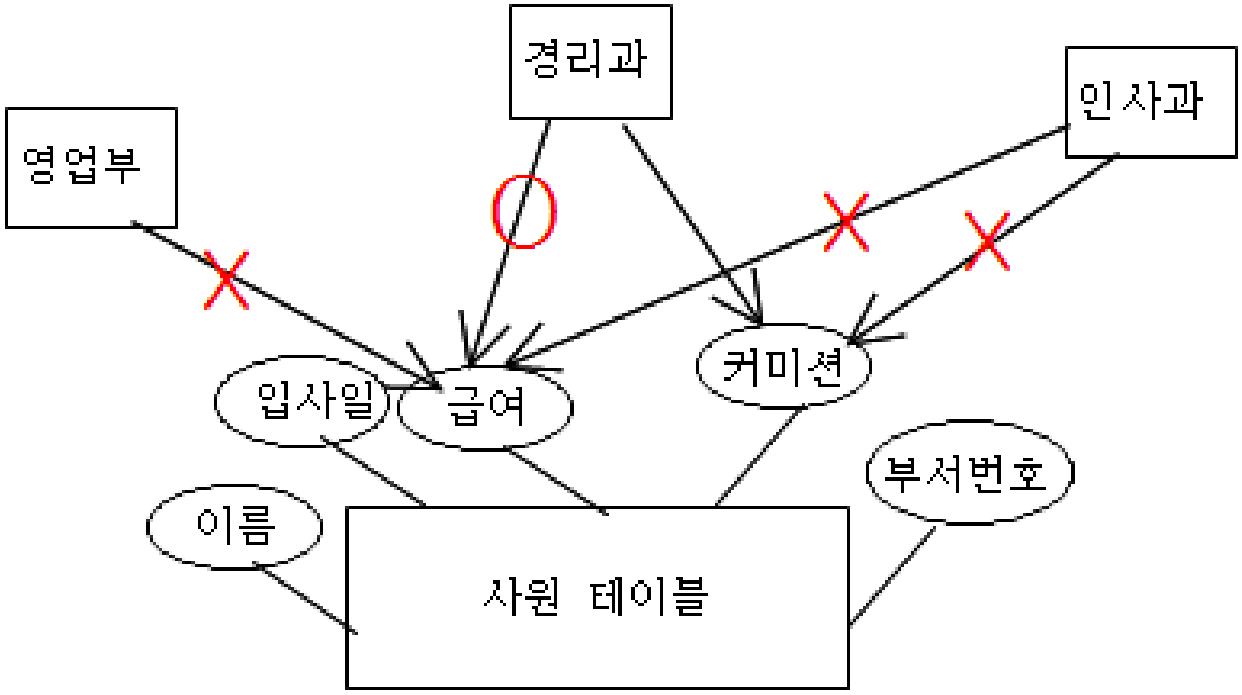
- 뷰를 사용하는 이유
  - 뷰를 사용하는 이유는 두 가지로 설명할 수 있습니다.

    1. 복잡하고 긴 쿼리문을 뷰로 정의하면 접근을 단순화시킬 수 있다.
    2. 보안에 유리하다.

  - 사용자마다 특정 객체만 조회할 수 있도록 권한을 부여를 할 수 있기에 동일한 테이블을 접근하는 사용자마다에 따라 서로 다르게 보도록 여러 개의 뷰를 정의해 놓고 특정 사용자만이 해당 뷰에 접근할 수 있도록 합니다.

1. 뷰 생성

- 뷰를 사용하는 이유
  - 예를 들어 사원 테이블에 개인 적인 정보인 급여와 커미션은 부서에 따라 접근을 제한해야 합니다. 급여나 커미션 모두에 대해서 인사과에서는 조회할 수 없도록 하고 경리과에서는 이 모두가 조회될 수 있도록 하지만 영업부서에서는 경쟁심을 유발하기 위해서 다른 사원의 커미션을 조회할 수 있도록 해야 합니다.



1. 뷰 생성

- 뷰 정의하기
  - 뷰를 만들기 전에 어떤 경우에 뷰를 사용하게 되는지 다음 예를 통해서 뷰가 필요한 이유를 설명해 보도록 하겠습니다.
  - 만일, 30번 부서에 소속된 사원들의 사번과 이름과 부서번호를 자주 검색한다고 한다면 다음과 같은 SELECT 문을 여러 번 입력해야 합니다.

```
SELECT EMPLOYEE_ID, FIRST_NAME, DEPARTMENT_ID
FROM EMPLOYEES_COPY
WHERE DEPARTMENT_ID = 30;
```

  - 위와 같은 결과를 출력하기위해서 매번 SELECT 문을 입력하기란 번거로운 일입니다.
  - 뷰는 이와 같이 번거로운 SELECT 문을 매번 입력하는 대신 보다 쉽게 원하는 결과를 얻고자 하는 바램에서 출발한 개념입니다.

1. 뷰 생성

- 뷰 정의하기
  - 자주 사용되는 30번 부서에 소속된 사원들의 사번과 이름과 부서번호를 출력하기 위한 SELECT문을 하나의 뷰로 정의해 봅시다.

```
CREATE VIEW VIEW_EMPLOYEES_COPY
AS
SELECT EMPLOYEE_ID, FIRST_NAME, DEPARTMENT_ID
FROM EMPLOYEES_COPY
WHERE DEPARTMENT_ID=30;
```

  - 뷰는 테이블에 접근(SELECT)한 것과 동일한 방법으로 결과를 얻을 수 있습니다.

```
SELECT * FROM VIEW_EMPLOYEES_COPY
```

2. 뷰 데이터의 관리

- 뷰의 내부구조와 USER_VIEWS 데이터 딕셔너리
  - 뷰는 물리적으로 데이터를 저장하고 있지 않다고 하였습니다. 그런데도 다음과 같은 질의 문을 수행할 수 있는 이유가 무엇일까요?

```
SELECT * FROM VIEW_EMPLOYEES_COPY
```

  - VIEW_EMPLOYEES_COPY라는 뷰는 데이터를 물리적으로 저장하고 있지 않습니다.
  - CREATE VIEW 명령어로 뷰를 정의할 때 AS 절 다음에 기술한 쿼리 문장 자체를 저장하고 있습니다.
  - 뷰 정의할 때 기술한 쿼리문이 궁금하다면 데이터 딕셔너리 USER_VIEWS 테이블의 TEXT 컬럼 값을 살펴보면 됩니다.

2. 뷰 데이터의 관리

- **뷰의 내부구조와 USER_VIEWS 데이터 딕셔너리**
  - USER_VIEWS에서 테이블 이름과 텍스트만 출력해 보겠습니다.

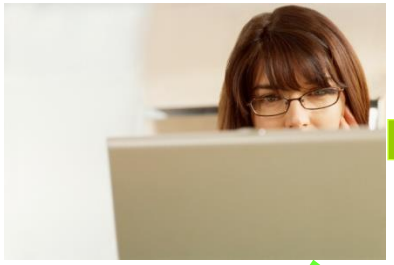
```
SELECT VIEW_NAME, TEXT
FROM USER_VIEWS
```

  - 기본 테이블은 디스크 공간을 할당 받아서 실질적으로 데이터를 저장하고 있지만, 뷰는 데이터 딕셔너리USER_VIEWS 에 사용자가 뷰를 정의할 때 기술한 서브 쿼리문(SELECT 문)만을 문자열 형태로 저장하고 있습니다.

2. 뷰 데이터의 관리

- 뷰의 내부구조와 USER_VIEWS 데이터 딕셔너리
  - 뷰의 동작 원리를 이해하기 위해서 뷰에 대한 질의가 어떻게 내부적으로 처리되는지 자세히 살펴보도록 합시다.

1. 사용자가 뷰에 대해서 질의를 하면 USER_VIEWS에서 뷰에 대한 정의를 조회합니다.
2. 기본 테이블에 대한 뷰의 접근 권한을 살핀다.
3. 뷰에 대한 질의를 기본테이블에 대한 질의로 변환합니다.
4. 기본 테이블에 대한 질의를 통해 데이터를 검색합니다.
5. 검색된 결과를 출력합니다.

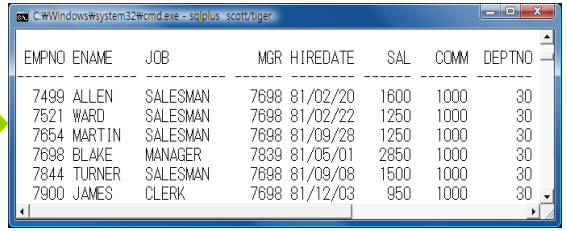


조회/수정/삭제

뷰

SELECT 문

쿼리 수행



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	81/02/20	1600	1000	30
7521	WARD	SALESMAN	7698	81/02/22	1250	1000	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1000	30
7698	BLAKE	MANAGER	7839	81/05/01	2850	1000	30
7844	TURNER	SALESMAN	7698	81/09/08	1500	1000	30
7900	JAMES	CLERK	7698	81/12/03	950	1000	30

쿼리 수행 결과

2. 뷰 데이터의 관리

- 뷰의 내부구조와 USER_VIEWS 데이터 딕셔너리
  - 우리가 앞에서 생성한 뷰인 VIEW_EMPLOYEES_COPY를 SELECT문의 FROM절 다음에 기술하여 질의를 하면 오라클 서버는 USER_VIEWS에서 VIEW_EMPLOYEES_COPY 를 찾아 이를 정의할 때 기술한 서브 쿼리문이 저장된 TEXT 값을 VIEW_EMPLOYEES_COPY 위치로 가져갑니다.

```
SELECT * FROM VIEW_EMPLOYEES_COPY
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, DEPARTMENT_ID
FROM EMPLOYEES_COPY
WHERE DEPARTMENT_ID =30;
```

  - 질의는 기본 테이블인 VIEW_EMPLOYEES_COPY 를 통해 일어납니다. 즉, 기본 테이블인 VIEW_EMPLOYEES_COPY 에 대해서 서브 쿼리문을 수행하게 됩니다. 이러한 동작 원리 덕분에 뷰가 실질적으로 데이터를 저장하고 있지 않더라도 데이터를 검색할 수 있는 것입니다.

2. 뷰 데이터의 관리

- **뷰의 내부구조와 USER_VIEWS 데이터 딕셔너리**
  - INSERT 문에 뷰(VIEW_EMPLOYEES_COPY)를 사용하였지만, 뷰는 쿼리문에 대한 이름일 뿐이기 때문에 새로운 행은 기본 테이블(EMPLOYEES_COPY)에 실질적으로 추가되는 것임을 알 수 있습니다. 뷰(VIEW_EMPLOYEES_COPY)의 내용을 확인하기 위해 SELECT문을 수행하면 변경된 기본 테이블(EMPLOYEES_COPY)의 내용에서 일부를 서브 쿼리한 결과를 보여줍니다.
  - 뷰는 실질적인 데이터를 저장한 기본 테이블을 볼 수 있도록 한 투명한 창입니다, 기본 테이블의 모양이 바뀐 것이고 그 바뀐 내용을 뷰라는 창을 통해서 볼 뿐입니다. 뷰에 INSERT 뿐만 아니라 UPDATE, DELETE 모두 사용할 수 있는데, UPDATE, DELETE 쿼리문 역시 뷰의 텍스트에 저장되어 있는 기본 테이블이 변경하는 것입니다.
  - 이 정도면 뷰가 물리적인 테이블을 근거로 한 논리적인 가상 테이블이란 말의 의미를 이해할 수 있으리라고 생각됩니다.

2. 뷰 데이터의 관리

- **뷰의 종류**
  - **뷰는 뷰를 정의하기 위해서 사용되는 기본 테이블의 수에 따라 단순 뷰(Simple View)와 복합 뷰(Complex View)로 나뉩니다.**

단순 뷰	복합 뷰
하나의 테이블로 생성	여러 개의 테이블로 생성
그룹 함수의 사용이 불가능	그룹 함수의 사용이 가능
DISTINCT 사용이 불가능	DISTINCT 사용이 가능
DML 사용 가능	DML 사용 불가능

2. 뷰 데이터의 관리

- 뷰 생성에 사용되는 다양한 옵션
  - 뷰 정의하는 방법을 살펴보면서 뷰를 생성하기 위한 사용되는 옵션에 대해서 대략적으로 설명을 했습니다.

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW view_name
[(alias, alias, alias, ...)]
AS subquery
[WITH CHECK OPTION]
[WITH READ ONLY];
```

  - 이번 절에서는 옵션에 대해서 예를 들어가면서 보다 자세히 살펴보도록 합시다.

2. 뷰 데이터의 관리

- **뷰 생성에 사용되는 다양한 옵션**
  - CREATE OR REPLACE VIEW 를 사용하면 존재하지 않은 뷰이면 새로운 뷰를 생성하고 기존에 존재하는 뷰이면 그 내용을 변경합니다.
  - 이전에 작성한 VIEW_EMPLOYEES_COPY 뷰는 "EMPLOYEE_ID, FIRST_NAME, DEPARTMENT_ID" 3 개의 컬럼을 출력하는 형태였는데 SALARY 칼럼을 추가로 출력할 수 있도록 하기 위해서 뷰의 구조를 변경합니다.

```
CREATE OR REPLACE VIEW VIEW_EMPLOYEES_COPY
AS
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEES_COPY
WHERE DEPARTMENT_ID = 30;
```

2. 뷰 데이터의 관리

- **기본 테이블 변경 막는** WITH READ ONLY **옵션**
  - WITH READ ONLY **옵션은 뷰를 통해서는 기본 테이블의 어떤 컬럼에 대해서도 내용을 절대 변경할 수 없도록 하는 것입니다.**

2. 뷰 데이터의 관리

- 뷰의 삭제
  - 뷰는 실체가 없는 가상 테이블이기 때문에 뷰를 삭제한다는 것은 USER_VIEWS 데이터 딕셔너리에 저장되어 있는 뷰의 정의를 삭제하는 것을 의미합니다.
  - 따라서 뷰를 삭제해도 뷰를 정의한 기본 테이블의 구조나 데이터에는 전혀 영향을 주지 않습니다.

```
DROP VIEW VIEW_EMPLOYEES_COPY;
```

2. 뷰 데이터의 관리

예: VIEW_DEPARTMENTS_COPY 생성

질의 : VIEW_DEPARTMENTS_COPY 를 생성하라