

초보자를 위한
Oracle SQL Database



시퀀스 및 테이블 관리



한국기술교육대학교
온라인평생교육원

학습내용

- 시퀀스
- 테이블 관리

학습목표

- 시퀀스를 생성하고 활용할 수 있다.
- 테이블 관리를 활용할 수 있다.

시퀀스



시퀀스 및 테이블 관리

시퀀스

시퀀스란?

시퀀스의 필요성

- Auto Increment와 같은 자동 증가 칼럼을 지원하는 데이터베이스가 있으나 Oracle은 지원을 하지 않음
- Oracle에서는 칼럼의 값을 증가시키기 위해서는 MAX(칼럼) + 1 또는 시퀀스를 사용하여 일련번호를 지정할 수 있음



시퀀스 및 테이블 관리

시퀀스

시퀀스란?

시퀀스란?

- Oracle에서 유일(UNIQUE)한 값을 생성해주는 객체
- 기본키(PRIMARY KEY)와 같이 순차적으로 증가하는 칼럼을 자동으로 생성하여 중복 값이 생성되지 않음
- 시퀀스는 테이블과는 독립적으로 저장되고 생성됨



시퀀스



시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.]시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

INCREMENT BY : 시퀀스 실행 시 증가시킬 값



시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.]시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

START WITH : 시퀀스 시작값
(MINVALUE와 같거나 커야 한다)



시퀀스

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.]시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

MINVALUE : 시퀀스 시작 시 최소값

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.]시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

MAXVALUE : 시퀀스가 가질 수 있는 최대값

시퀀스

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.] 시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

NOCYCLE | CYCLE : 시퀀스의 최대값 도달 시
최소값 값으로 다시 반복 여부.

NOCYCLE(반복 안함. 시퀀스 생성 중지),
CYCLE(최대값 도달 시 최소값으로 다시 시작)

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [スキマ.] 시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

NOCACHE | CACHE : 캐시를 사용하여 메모리에 미리 값을
계산하여 속도를 개선, 동시 사용자가 많을 경우 장점

NOCACHE(사용 안 함), CACHE 값 : 기본값 20 지정한 값만큼
시퀀스를 미리 생성

시퀀스



시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

시퀀스 문 작성

CREATE SEQUENCE [스키마.] 시퀀스명

INCREMENT BY 증감값

START WITH 시작값

[MINVALUE | NOMINVALUE] 최소값

[MAXVALUE | NOMAXVALUE] 최대값

[CYCLE | NOCYCLE]

[CACHE | NOCACHE]

[ORDER | NOORDER]

NOORDER | ORDER : 옵션에서 지정한 순서로 값을 생성
NOORDER(사용 안함), ORDER(사용)

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스를 테스트할 테이블 생성

```
워크시트 질의 작성기
CREATE TABLE SEQ_TEST_TABLE
    (ID NUMBER, NAME VARCHAR2(20));
```

스크립트 출력 x 질의 결과 x
작업이 완료되었습니다.(0.038초)

Table SEQ_TEST_TABLE이 (가) 생성되었습니다.

**CREATE TABLE SEQ_TEST_TABLE
(ID NUMBER, NAME VARCHAR2(20));**

시퀀스

시퀀스 및 테이블 관리

시퀀스

❖ 시퀀스 문

✚ 시퀀스 생성

워크시트 질의 작성기

```
CREATE SEQUENCE SEQ_MYNO
    INCREMENT BY 1
    START WITH 1;
```

SEQ_MYN0라는 시퀀스를 생성하는 예
증가값을 1로 지정
시작값을 1로 지정

스크립트 출력 x
작업이 완료되었습니다.(0.046초)

Sequence SEQ_MYN0이 (가) 생성되었습니다.

**CREATE SEQUENCE SEQ_MYN0
INCREMENT BY 1
START WITH 1;**

시퀀스 및 테이블 관리

시퀀스

❖ 시퀀스 문

✚ 테이블에 시퀀스를 사용하여 데이터 입력

ID	NAME
1	강감찬

INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYN0.NEXTVAL, '강감찬');

시퀀스

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

- + 테이블에 시퀀스를 사용하여 데이터 입력

The screenshot shows two windows of Oracle SQL Developer. The left window displays the following SQL code:

```
INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '강감찬');
INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '이순신');
```

The right window shows the results of the query in a table:

ID	NAME
1	강감찬
2	이순신

A purple callout box points to the table with the text "자동으로 ID의 값이 101 증가하여 2로 입력" (The value of ID increases automatically from 101 to 2). A red arrow points from the text "데이터를 insert하면" (When inserting data) to the first INSERT statement.

INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '이순신');

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

- + 테이블에 시퀀스를 사용하여 데이터 입력

The screenshot shows two windows of Oracle SQL Developer. The left window displays the following SQL code:

```
INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '강감찬');
INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '이순신');
```

Below the code, a red box highlights the DELETE statement:

```
DELETE FROM SEQ_TEST_TABLE WHERE NAME = '이순신';
```

The right window shows the results of the query in a table:

ID	NAME
1	강감찬

A purple callout box points to the table with the text "시퀀스명.NEXTVAL은 지정한 시퀀스명의 값을 INCREMENT BY에서 지정한 증감값만큼 크기를 변경" (The value of SEQ_NAME.NEXTVAL is changed by the increment value specified in INCREMENT BY).

A red arrow points from the text "NEXTVAL을 테스트 하기 위해 테이블에서 마지막으로 입력한 데이터를 삭제" (Delete the last inserted data from the table to test NEXTVAL) to the DELETE statement.

NEXTVAL을 테스트 하기 위해 테이블에서 마지막으로 입력한 데이터를 삭제

시퀀스

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

+ 테이블에 시퀀스를 사용하여 데이터 입력

The screenshot shows the Oracle SQL Developer interface with a script editor window. The code is as follows:

```

-- 시작 페이지 x 19c remote x SEQ_TEST_TABLE x
-- 워크시트 | 질의 작성기
-- INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '이순신');
-- DELETE FROM SEQ_TEST_TABLE WHERE NAME = '이순신';
-- INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '유관순');

-- 스크립트 출력 x | 질의 결과 x
-- 작업이 완료되었습니다.(0.03초)

1 행 이 (가) 삽입되었습니다.

1 행 이 (가) 삭제되었습니다.

1 행 이 (가) 삽입되었습니다.

```

A callout bubble points to the second insert statement with the text "데이터 삭제 후 다시 데이터를 insert".

INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO.NEXTVAL, '유관순');

시퀀스 및 테이블 관리

시퀀스

시퀀스 문

+ 테이블에 시퀀스를 사용하여 데이터 입력

The screenshot shows the Oracle SQL Developer interface with a table editor window for the SEQ_TEST_TABLE. The table has two columns: ID and NAME. The data is:

ID	NAME
1	이순신
2	강감찬
3	유관순

A callout bubble points to the row with ID 3 and NAME 유관순 with the text "데이터 탭을 선택하여 SEQ_TEST_TABLE의 데이터를 확인하면, ID의 값이 3으로 된 것이 보임".

시퀀스는 데이터의 삭제와 관계없이 순차적으로 계속 증가

시퀀스

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

✚ 시퀀스명.CURRVAL

시퀀스명.CURRVAL로 현재 시퀀스 번호를 확인할 수 있음

```
SELECT SEQ_MYNO.CURRVAL FROM DUAL;
```

CURRVAL은 NEXTVAL과 달리
여러 번 실행해도 번호는 증가하지 않음

CURRVAL
1
3

SELECT SEQ_MYNO.CURRVAL FROM DUAL;

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

✚ 시퀀스명.CURRVAL

SELECT 문으로 SEQ_MYNO.NEXTVAL을 여러 번 실행한 결과

NEXTVAL
1
4

NEXTVAL
1
5

NEXTVAL
1
6

실행할 때마다 SEQ_MYNO.NEXTVAL의 값이 증가하는 것을 볼 수 있음

시퀀스

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

✚ 시퀀스명.CURRVAL

```

CREATE SEQUENCE SEQ_MYNO2
INCREMENT BY 2
START WITH 1
MAXVALUE 35
CYCLE
CACHE 10;

```

Sequence SEQ_MYNO2이(가) 생성되었습니다.

**CREATE SEQUENCE
SEQ_MYNO2
INCREMENT BY 2
START WITH 1
MAXVALUE 35
CYCLE
CACHE 10;**

초기값을 1로 하고 최대 35까지 2씩 증가를 하며, 최대값에 도달하면 다시 초기값으로 돌아가도록 설정

속도 향상을 위해 캐시를 10개로 지정한 시퀀스

SEQ_MYNO2라는 시퀀스를 하나 더 생성

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

✚ 시퀀스명.CURRVAL

```

SELECT SEQ_MYNO2.CURRVAL FROM DUAL;

```

ORA-08002: 시퀀스 SEQ_MYNO2.CURRVAL은 이 세션에서는 정의 되어 있지 않습니다
08002, 00000 - "sequence %s.CURRVAL is not yet defined in this session"
*Cause: sequence CURRVAL has been selected before sequence NEXTVAL
*Action: select NEXTVAL from the sequence before selecting CURRVAL

SEQ_MYNO2를 생성한 상태에서 다음과 같이 CURRVAL을 실행하면 오류가 발생

SELECT SEQ_MYNO2.CURRVAL FROM DUAL;

시퀀스

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스명.CURRVAL

```

시작 페이지 × 19c remote × SEQ_TEST_TABLE ×
▶ 워크시트 | 질의 작성기
CYCLE
CACHE 10;

SELECT SEQ_MYNO2.CURRVAL FROM DUAL;
SELECT SEQ_MYNO2.NEXTVAL FROM DUAL;

SQL | 인출된 모든 행: 1(0.007초)
NEXTVAL
1 1

```

CURRVAL은 NEXTVAL을 최소 한번은 실행한 세션에서만 사용 가능

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스명.CURRVAL

```

시작 페이지 × 19c remote × SEQ_TEST_TABLE ×
▶ 워크시트 | 질의 작성기
CACHE 10;

SELECT SEQ_MYNO2.CURRVAL FROM DUAL;
SELECT SEQ_MYNO2.NEXTVAL FROM DUAL;
INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO2.NEXTVAL, '박문수');

SQL | 작업이 완료되었습니다.(0.041초)
Sequence SEQ_MYNO2이 (가) 생성되었습니다.

```

SEQ_TEST_TABLE에 SEQ_MYNO2.NEXTVAL 시퀀스를 이용하여 데이터를 입력

INSERT INTO SEQ_TEST_TABLE VALUES(SEQ_MYNO2.NEXTVAL, '박문수');

시퀀스

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스명.CURRVAL

SEQ_TEST_TABLE을 보면 새로 추가된 “박문수” 데이터가 보임

ID	NAME
1	강감찬
2	유관순
3	박문수

ID를 보면 3으로 되어 있는데,
이는 SEQ_MYN02 시퀀스의 값

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스명.CURRVAL

SEQ_MYN02.NEXTVAL 시퀀스가 2씩 증가를 하며,
최대값에 도달하면 다시 초기값으로 돌아가는지 확인한 결과

ID	NAME
3	박문수
5	박문수
7	박문수
9	박문수
11	박문수
13	박문수
15	박문수
17	박문수
19	박문수
21	박문수
23	박문수
25	박문수
27	박문수
29	박문수
31	박문수
33	박문수
35	박문수
1	박문수
3	박문수
5	박문수
7	박문수

자동 증가 칼럼을 지원하는 데이터베이스와 달리
Oracle에서는 시퀀스를 이용하여
필요에 따라 하나의 테이블에도 여러 개의 시퀀스를 사용하여
데이터를 저장할 수 있음

시퀀스



시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스 조회

```
SELECT * FROM USER_SEQUENCES;
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_BY
SEQ_MYNO	1	9999999999999999999999999999999999	1	N	N
SEQ_MYNO2	1		35	Y	N

SELECT * FROM USER_SEQUENCES;

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스 수정

```

ALTER SEQUENCE SEQ_MYNO INCREMENT BY 3;
ALTER SEQUENCE SEQ_MYNO MAXVALUE 999999;
ALTER SEQUENCE SEQ_MYNO CYCLE;

```

스크립트 출력 x | 질의 결과 x
작업이 완료되었습니다.(0.036초)

Sequence SEQ_ID01 (가) 삭제되었습니다.

Sequence SEQ_MYNO01 (가) 변경되었습니다.

Sequence SEQ_MYNO01 (가) 변경되었습니다.

시퀀스 증가값을 1 → 3로 변경
ALTER SEQUENCE SEQ_MYNO INCREMENT BY 3;

시퀀스 최대값을 999,999로 변경
ALTER SEQUENCE SEQ_MYNO MAXVALUE 999999;

시퀀스가 최대값에 도달하면 다시 최소값으로 다시 반복
ALTER SEQUENCE SEQ_MYNO CYCLE;

정의되어 있는 시퀀스는 START WITH를 제외한 모든 값은 변경이 가능

시퀀스

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스 수정

The screenshot shows the Oracle SQL Developer interface. In the top-left, there are tabs for '시퀀스 및 테이블 관리' and '시퀀스'. The main area has a toolbar with various icons. Below the toolbar is a '워크시트' (Worksheet) tab and a '질의 작성기' (Query Builder) tab. The worksheet contains the following SQL code:

```
ALTER SEQUENCE SEQ_MYNO MAXVALUE 999999;
ALTER SEQUENCE SEQ_MYNO CYCLE;

SELECT * FROM USER_SEQUENCES;
```

Below the worksheet is a results window titled 'SQL | 인출된 모든 행: 2(0,002초)'. It displays the following table:

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
SEQ_MYNO	1	999999	3	Y	N	20	
SEQ_MYNO2	1	35	2	Y	N	10	

A purple callout box points to the '변경된 SEQ_MYNO 시퀀스를 조회한 결과' (Result of querying the modified SEQ_MYNO sequence) section, which corresponds to the second row of the table.

A red callout box highlights the SQL query:

```
SELECT * FROM USER_SEQUENCES;
```

시퀀스 및 테이블 관리

시퀀스

◆ 시퀀스 문

+ 시퀀스 삭제

The screenshot shows the Oracle SQL Developer interface. In the top-left, there are tabs for '시퀀스 및 테이블 관리' and '시퀀스'. The main area has a toolbar with various icons. Below the toolbar is a '워크시트' (Worksheet) tab and a '질의 작성기' (Query Builder) tab. The worksheet contains the following SQL code:

```
SELECT * FROM USER_SEQUENCES;

DROP SEQUENCE SEQ_MYNO2;

SELECT * FROM USER_SEQUENCES;
```

Below the worksheet is a results window titled 'SQL | 인출된 모든 행: 1(0,004초)'. It displays the following table:

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
SEQ_MYNO	1	999999	3	Y	N	20	

A red callout box highlights the SQL command:

```
DROP SEQUENCE SEQ_MYNO2;
```

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

▣ 테이블 생성

CREATE TABLE 명령문의 특징

- 실행 즉시 COMMIT되는 데이터 정의어(DDL)
- 데이터 조작어(DML)과 달리 ROLLBACK될 수 없음



시퀀스 및 테이블 관리

테이블 관리

▣ 테이블 생성

CREATE TABLE 문

CREATE [GLOBAL TEMPORARY] **TABLE** [SCHEMA.]

테이블명

(

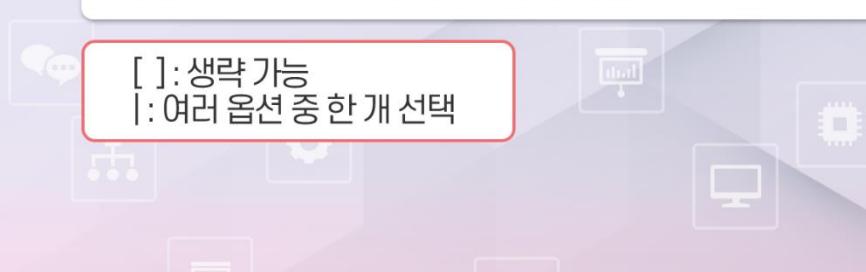
칼럼1 데이터 타입 [DEFAULT 표현식] [칼럼 제약],
칼럼2 데이터 타입 [DEFAULT 표현식] [칼럼 제약],

...

CONSTRAINT [제약이름] [제약종류] 칼럼1 [, 컬럼2] ...
) [TABLESPACE 테이블 스페이스명];

[]: 생략 가능

|: 여러 옵션 중 한 개 선택



테이블 관리

시퀀스 및 테이블 관리

테이블 관리

◆ 테이블 생성

CREATE TABLE 문

CREATE [GLOBAL TEMPORARY] TABLE [SCHEMA.]

테이블명

(

칼럼1 데이터 타입 [DEFAULT 표현식] [칼럼 제약],
칼럼2 데이터 타입 [DEFAULT 표현식] [칼럼 제약],

...

CONSTRAINT [제약이름] [제약종류] 칼럼1 [, 컬러2] ...
) [**TABLESPACE** 테이블 스페이스명];

GLOBAL TEMPORARY : 임시 테이블을 만들기 위한
키워드로서 테이블 구조는 모든 세션에서 볼 수 있지만,
데이터는 테이블을 생성한 세션에서만 조회 가능

시퀀스 및 테이블 관리

테이블 관리

◆ 테이블 생성

CREATE TABLE 문

CREATE [GLOBAL TEMPORARY] TABLE [SCHEMA.]

테이블명

(

칼럼1 데이터 타입 [DEFAULT 표현식] [칼럼 제약],
칼럼2 데이터 타입 [DEFAULT 표현식] [칼럼 제약],

...

CONSTRAINT [제약이름] [제약종류] 칼럼1 [, 컬러2] ...
) [**TABLESPACE** 테이블 스페이스명];

TABLESPACE : 테이블이 저장되는
테이블스페이스를 지정

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

◆ 테이블 생성

워크시트 질의 작성기

```
DESC 교과목;
```

DESC[RIBE] 명령어로 테이블의 생성 여부와 테이블의 구조를 확인할 수 있음

스크립트 출력 x

작업이 완료되었습니다.(0.031초)

이름	널?	유형
교과목코드	NOT NULL	CHAR (6)
교과목명		NVARCHAR2 (30)
이수구분		NVARCHAR2 (3)
학점		NUMBER (2)
학부코드		NUMBER (2)

확인 가능한 정보: 이름, NULL 여부, 데이터 유형(타입과 크기), 무결성 제약 조건

교과목 테이블 확인 예제 : DESC 교과목;

시퀀스 및 테이블 관리

테이블 관리

◆ 서브 쿼리를 이용한 테이블 생성

서브 쿼리를 이용한 테이블 생성

- CREATE TABLE 명령문의 서브 쿼리 절을 이용하여 다른 테이블의 구조와 데이터를 복사하여 새로운 테이블을 생성할 수 있음
- CREATE TABLE 명령문에서 지정한 칼럼의 수와 데이터 타입은 서브 쿼리 절에서 지정한 칼럼 수와 데이터 타입이 일치해야 함
- 테이블 생성 시 칼럼명을 지정하지 않을 경우에는 서브 쿼리의 칼럼명과 동일한 칼럼명으로 생성
- 서브 쿼리 절을 이용하여 테이블을 생성하면 서브 쿼리의 출력 결과가 생성되는 테이블의 초기 데이터로 삽입됨
- 무결성 제약조건은 NOT NULL 조건만 복사되므로 기본키, 참조키와 같은 무결성 제약조건은 사용자가 다시 정의해야 함
- DEFAULT 옵션에서 정의한 기본값은 그대로 복사됨

테이블 관리

시퀀스 및 테이블 관리 테이블 관리

서브 쿼리를 이용한 테이블 생성

```
CREATE TABLE 교과목2
AS SELECT * FROM 교과목;
```

작업이 완료되었습니다.(0,209초)

Table 교과목2이 (가) 생성되었습니다.

예1) 교과목 테이블의 구조와 데이터를 복사하여 교과목2 테이블 생성

시퀀스 및 테이블 관리 테이블 관리

서브 쿼리를 이용한 테이블 생성

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 교과목코드	CHAR (6 BYTE)	No	(null)	1 (null)	
2 교과목명	NVARCHAR2 (30 C...	Yes	(null)	2 (null)	
3 이수구분	NVARCHAR2 (3 CHAR)	Yes	(null)	3 (null)	
4 학점	NUMBER (2, 0)	Yes	(null)	4 (null)	
5 학부코드	NUMBER (2, 0)	Yes	(null)	5 (null)	

교과목2 테이블 구조와 데이터

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

	교과목코드	교과목명	이수구분	학점	학부코드
1	IMA360	생산관리	전공	3	40
2	IMA371	국제금융론	전공	3	40
3	IMA391	투자론	전공	3	40
4	IMA430	경영전략 (캡스톤디자인)	전공	3	40
5	IMA442	국제경영론	전공	3	40
6	IMA450	민법	전공	3	40
7	IMA507	비즈니스커뮤니케이션1	전공	4	40
8	IMA615	관리회계실무	전공	3	40
9	IMA731	하이테크마케팅 (PBL)	전공	4	40
10	IMA851	소비자행동론 (특수)	전공	3	40
11	IMA860	기업법	전공	3	40
12	IMA900	경영정보시스템개론	전공	3	40
13	IMA910	혁신글로벌리더십	전공	3	40

교과목2 테이블 구조와 데이터

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

```

워크시트    질의 작성기
CREATE TABLE 교과목3
AS SELECT 교과목코드, 교과목명, 이수구분 FROM 교과목;

```

스크립트 출력 x | 작업이 완료되었습니다.

CREATE TABLE 교과목3
AS SELECT 교과목코드, 교과목명, 이수구분 FROM 교과목;

Table 교과목3이 (가) 생성되었습니다.

예2) 교과목 테이블의 교과목코드, 교과목명, 이수구분 칼럼과 데이터를 복사하여 교과목3 테이블 생성

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

The screenshot shows the Oracle SQL Developer interface with a table editor open. The table is named '교과목3'. It has three columns: 'COLUMN_NAME', 'DATA_TYPE', and 'COMMENTS'. The data is as follows:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 교과목코드	CHAR (6 BYTE)	No	(null)	1	(null)
2 교과목명	NVARCHAR2 (30 C...	Yes	(null)	2	(null)
3 이수구분	NVARCHAR2 (3 CHAR)	Yes	(null)	3	(null)

교과목3 테이블 구조와 데이터

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

The screenshot shows the Oracle SQL Developer interface with a table editor open. The table is named '교과목3'. It has three columns: '교과목코드', '교과목명', and '이수구분'. The data is as follows:

교과목코드	교과목명	이수구분
1 IMA360	생산관리	전공
2 IMA371	국제금융론	전공
3 IMA391	투자론	전공
4 IMA430	경영전략 (캡스톤디자인)	전공
5 IMA442	국제경영론	전공
6 IMA450	민법	전공
7 IMA507	비즈니스커뮤니케이션1	전공
8 IMA615	관리회계실무	전공
9 IMA731	하이테크마케팅 (PBL)	전공
10 IMA851	소비자행동론 (특수)	전공
11 IMA860	기업법	전공
12 IMA900	경영정보시스템개론	전공
13 IMA910	혁신글로벌리더십	전공

교과목3 테이블 구조와 데이터

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

**CREATE TABLE 교과목4
AS SELECT * FROM 교과목
WHERE 1=2;**

데이터 구조만 복사하려면 서브 쿼리의 WHERE 조건절의
조건을 항상 거짓이 되게 함 (WHERE 1=2)
• 조건을 만족하는 결과가 없어 데이터가 발생하지 않음

Table 교과목4이 (가) 생성되었습니다.

WHERE 조건을 어떻게 지정하느냐에 따라
일부 데이터만 복사하여 테이블을 생성할 수도 있음

예3) 교과목 테이블의 구조만 복사하여 교과목4 테이블 생성

시퀀스 및 테이블 관리

테이블 관리

서브 쿼리를 이용한 테이블 생성

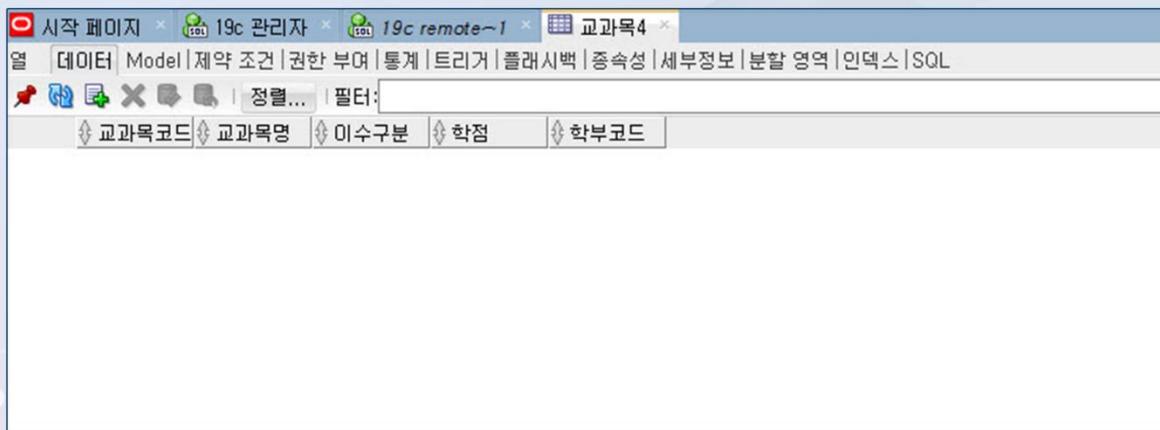
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 교과목코드	CHAR (6 BYTE)	No	(null)	1 (null)	
2 교과목명	NVARCHAR2 (30 C...	Yes	(null)	2 (null)	
3 이수구분	NVARCHAR2 (3 CHAR)	Yes	(null)	3 (null)	
4 학점	NUMBER (2, 0)	Yes	(null)	4 (null)	
5 학부코드	NUMBER (2, 0)	Yes	(null)	5 (null)	

교과목4 테이블 구조와 데이터

테이블 관리

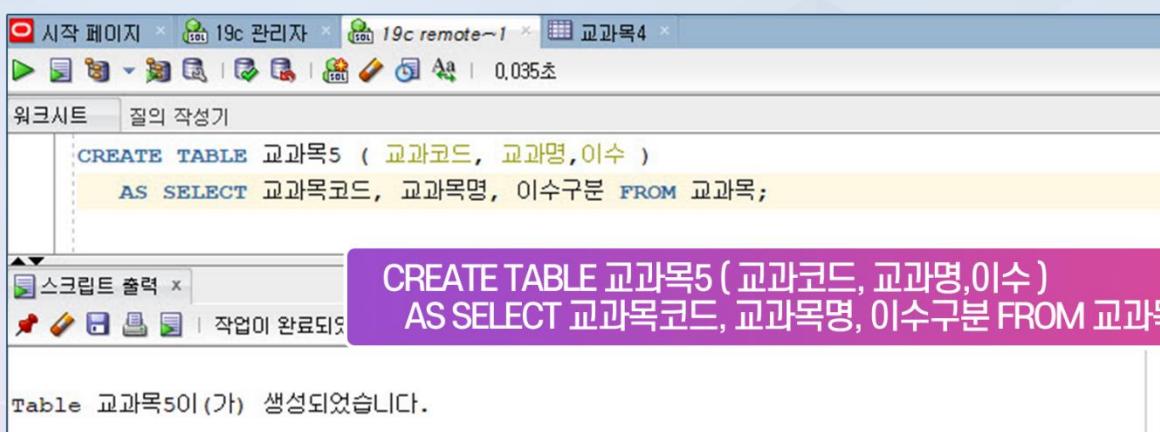
시퀀스 및 테이블 관리 테이블 관리

◆ 서브 쿼리를 이용한 테이블 생성



시퀀스 및 테이블 관리 테이블 관리

◆ 서브 쿼리를 이용한 테이블 생성



예4) 교과목 테이블의 **교과목코드, 교과목명, 이수구분** 칼럼을
교과목5 테이블의 **교과코드, 교과명, 이수** 칼럼으로 복사하고 데이터도 복사

테이블 관리

시퀀스 및 테이블 관리 테이블 관리

서브 쿼리를 이용한 테이블 생성

The screenshot shows the Oracle SQL Developer interface with the '교과목5' table selected. The table has three columns: COLUMN_NAME, DATA_TYPE, and COMMENTS. The data is as follows:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 교과코드	CHAR (6 BYTE)	No	(null)	1 (null)	
2 교과명	NVARCHAR2 (30 C...	Yes	(null)	2 (null)	
3 이수	NVARCHAR2 (3 CHAR)	Yes	(null)	3 (null)	

교과목5 테이블 구조와 데이터

시퀀스 및 테이블 관리 테이블 관리

서브 쿼리를 이용한 테이블 생성

The screenshot shows the Oracle SQL Developer interface with the '교과목5' table selected. The table has three columns: 교과코드, 교과명, and 이수. The data is as follows:

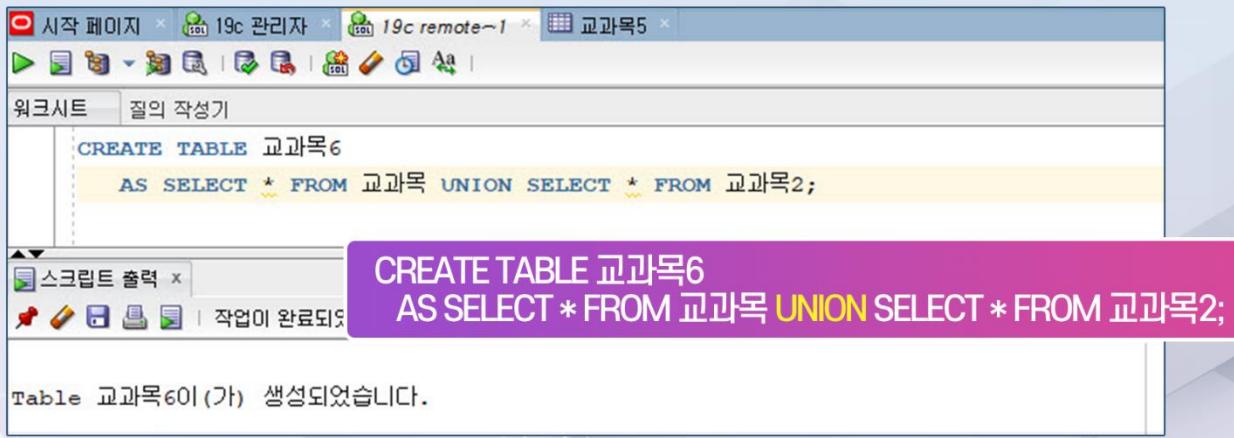
교과코드	교과명	이수
1 IMA360	생산관리	전공
2 IMA371	국제금융론	전공
3 IMA391	투자론	전공
4 IMA430	경영전략 (캡스톤디자인)	전공
5 IMA442	국제경영론	전공
6 IMA450	민법	전공
7 IMA507	비즈니스커뮤니케이션1	전공
8 IMA615	관리회계실무	전공
9 IMA731	하이테크마케팅 (PBL)	전공
10 IMA851	소비자행동론 (특수)	전공
11 IMA860	기업법	전공
12 IMA900	경영정보시스템개론	전공
13 IMA910	혁신글로벌리더십	전공

교과목5 테이블 구조와 데이터

테이블 관리

시퀀스 및 테이블 관리 테이블 관리

서브 쿼리를 이용한 테이블 생성



```
CREATE TABLE 교과목6
AS SELECT * FROM 교과목 UNION SELECT * FROM 교과목2;
```

Table 교과목6이 (가) 생성되었습니다.

**예5) 두 개의 테이블(교과목, 교과목2)을 합쳐서 교과목6 테이블 만들기
이때 테이블의 칼럼 구조와 크기는 같아야 함**

시퀀스 및 테이블 관리 테이블 관리

테이블 삭제

DROP TABLE 명령문의 특징

- 실행 즉시 COMMIT되는 데이터 정의어(DDL)
- 데이터 조작어(DML)와 달리 ROLLBACK될 수 없음
- 자신이 소유한 테이블에 대해 저장된 데이터를 포함해서 테이블 전체를 삭제할 수 있음

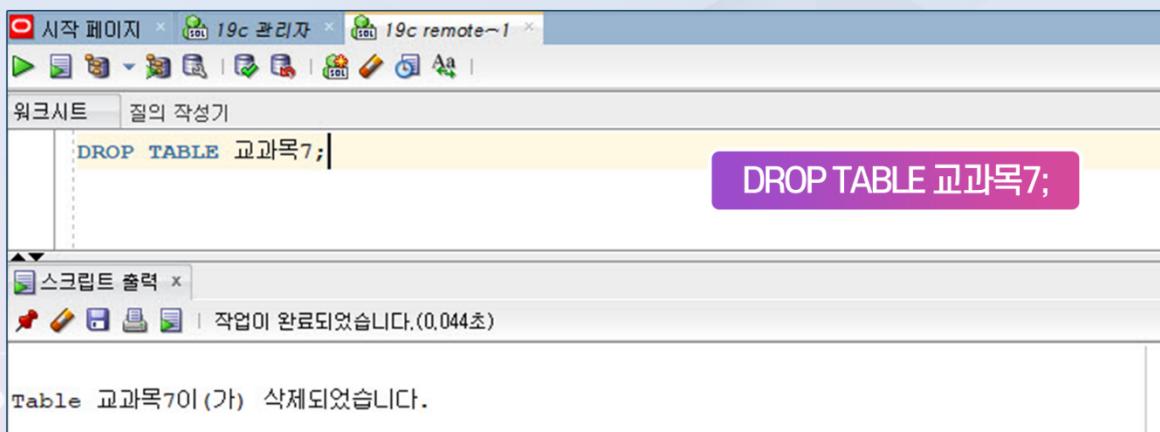
DROP TABLE 명령문

DROP TABLE [SCHEMA.] 테이블명;

테이블 관리

시퀀스 및 테이블 관리 테이블 관리

◆ 테이블 삭제



예) 교과목7 테이블 삭제

시퀀스 및 테이블 관리 테이블 관리

◆ 데이터 딕셔너리

데이터 딕셔너리란

- 생성된 모든 테이블에 대한 모든 정보는 데이터 딕셔너리에 저장
- 사용자와 데이터베이스의 자원을 효율적으로 관리하기 위한 데이터를 저장하는 시스템 테이블의 집합
- 데이터 딕셔너리에 저장된 데이터는 Oracle 서버에 의해 자동으로 갱신
- 데이터베이스 관리자나 일반 사용자는 읽기 전용 뷰에 의해 데이터 딕셔너리의 내용을 조회만 할 수 있음

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

데이터 딕셔너리에서 관리하는 데이터 정보

- DB의 물리적 구조와 객체의 논리적 구조
- 스키마 객체에 할당된 공간의 크기와 사용중인 공간의 크기 정보
- 객체 접근 및 간접에 대한 감사 정보
- 데이터베이스명, 버전, 생성날짜, 시작모드, 인스턴스명 정보
- 오라클 사용자명과 스키마 객체명
- 사용자에게 부여된 접근 권한과 풀
- 무결성 제약조건에 대한 정보
- 칼럼별로 지정된 기본 값

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

데이터 딕셔너리의 종류

- 데이터 딕셔너리는 읽기 전용 뷰로 구성되어 데이터베이스 관리자나 사용자에게 데이터 딕셔너리에 저장된 정보의 조회를 할 수 있음
- 데이터 딕셔너리 뷰는 용도에 따라 USER, ALL, DBA의 3가지 접두어를 가지고 있음

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

데이터 딕셔너리의 종류

- USER 접두어를 가진 데이터 딕셔너리 뷰는 해당 객체를 소유한 사용자만 접근 가능
- ALL 접두어를 가진 데이터 딕셔너리 뷰는 소유자나 소유자로부터 접근 권한을 부여받은 사용자만 접근 가능
- DBA 접두어를 가진 데이터 딕셔너리 뷰는 데이터베이스 관리자만 접근 가능
- 데이터베이스 관리자는 모든 종류의 데이터 딕셔너리 뷰에 접근 가능

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ USER_데이터 딕셔너리 뷰

USER_데이터 딕셔너리 뷰

- 일반 사용자와 관련된 뷰
- 사용자는 USER_데이터 딕셔너리 뷰를 이용하여 자신이 생성한 테이블, 인덱스, 뷰, 동의어 등과 같은 객체나 해당 사용자에게 할당된 권한정보를 조회할 수 있음

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ USER_데이터 딕셔너리 뷰



The screenshot shows the Oracle SQL Developer interface. A query window contains the SQL statement: `SELECT table_name FROM user_tables;`. The results window displays a list of tables:

TABLE_NAME
1 학부
2 교과목
3 전공
4 교수
5 수강
6 교수인사발령

user_tables는 사용자가 소유한 테이블에 대한 정보를 조회하는 데이터 딕셔너리 뷰

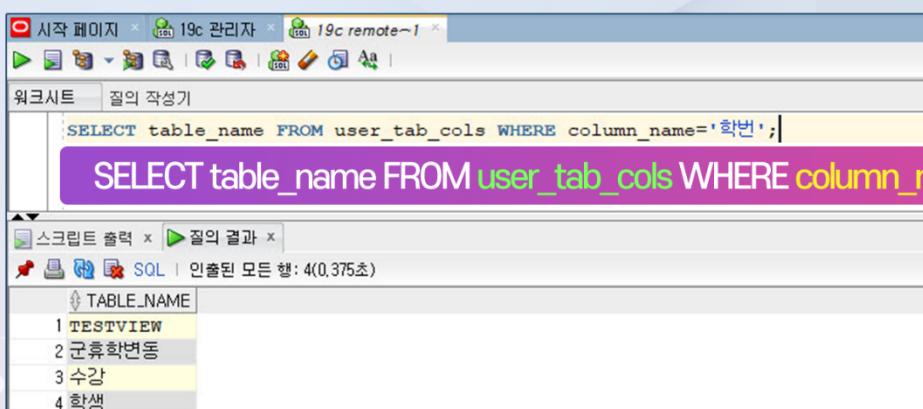
예1) USER_데이터 딕셔너리 뷰 조회

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ USER_데이터 딕셔너리 뷰



The screenshot shows the Oracle SQL Developer interface. A query window contains the SQL statement: `SELECT table_name FROM user_tab_cols WHERE column_name='학번';`. The results window displays a list of tables:

TABLE_NAME
1 TESTVIEW
2 군필학번동
3 수강
4 학생

SELECT table_name FROM user_tab_cols WHERE column_name='학번';

예2) “학번”이라는 칼럼명을 갖고 있는 테이블 찾기

테이블 관리

시퀀스 및 테이블 관리 테이블 관리

데이터 딕셔너리

+ ALL_데이터 딕셔너리 뷰

ALL_데이터 딕셔너리 뷰

- 전체 사용자와 관련된 뷰
- ALL_데이터 딕셔너리 뷰는 해당 객체의 소유자를 확인할 수 있는 OWNER 칼럼이 있음
- 사용자는 ALL_데이터 딕셔너리 뷰를 이용하여 해당 사용자가 접근할 수 있는 모든 객체에 대한 정보를 조회할 수 있음

시퀀스 및 테이블 관리 테이블 관리

데이터 딕셔너리

+ ALL_데이터 딕셔너리 뷰

```
시작 페이지 × 19c 관리자 × 19c remote~1 ×
작업 공간: 질의 작성기
작업 공간: 질의 결과 ×
작업 공간: 스크립트 출력 ×
작업 공간: 질의 결과 ×
작업 공간: SQL | 150개의 행이 인출됨(0.97초)

SELECT owner, table_name FROM all_tables;
```

OWNER	TABLE_NAME
97 STUDY01	교과목6
98 STUDY01	교과목7
99 STUDY01	학생
100 SYS	FINALHIST\$
101 SYS	MODELGTTRAW\$
102 SYS	WRI\$_ADV_ASA_RECO_DATA

all_tables는 자기 소유 또는 권한을 부여받은 테이블에 대한 정보를 조회할 수 있는 데이터 딕셔너리 뷰

ALL_데이터 딕셔너리 뷰 조회

테이블 관리

시퀀스 및 테이블 관리 | 테이블 관리

데이터 딕셔너리

+ DBA_데이터 딕셔너리 뷰

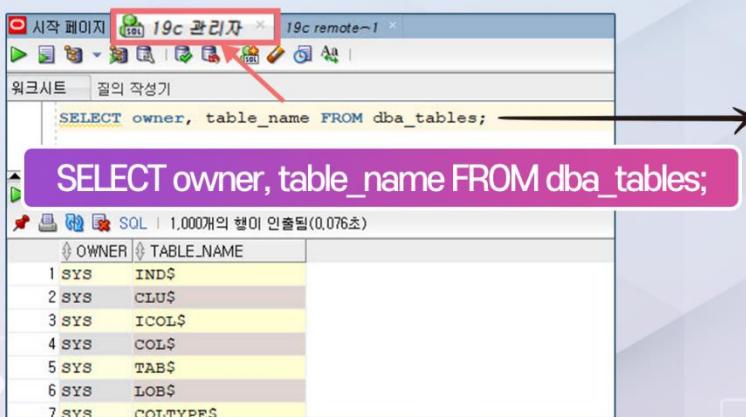
DBA_데이터 딕셔너리 뷰

- 데이터베이스 시스템 관리와 관련된 뷰
- DBA_데이터 딕셔너리 뷰는 **DBA**나 **SELECT ANY TABLE** 시스템 권한을 가진 사용자만 접근할 수 있음
- 주로 데이터베이스 관리자가 사용자의 접근 권한이나 데이터베이스 자원을 관리하기 위한 목적으로 사용

시퀀스 및 테이블 관리 | 테이블 관리

데이터 딕셔너리

+ DBA_데이터 딕셔너리 뷰



```
시작 페이지 19c 관리자 19c remote~1
작업 공간 편집기 워크시트 질의 작성기
질의 작성기
SELECT owner, table_name FROM dba_tables;
```

SQL | 1,000개의 행이 인출됨(0.076초)

OWNER	TABLE_NAME
1 SYS	IND\$
2 SYS	CLUS\$
3 SYS	ICOL\$
4 SYS	COL\$
5 SYS	TAB\$
6 SYS	LOB\$
7 SYS	COLTYPES\$

dba_tables는 모든 사용자가 소유한 테이블에 대한 정보를 조회 할 수 있는 데이터 딕셔너리 뷰

예) DBA_데이터 딕셔너리 뷰 조회

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

1 USER_TABLES

2 USER_OBJECTS

3 USER_CATALOG



시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

1 USER_TABLES : 테이블이 저장된 테이블 스페이스명, 데이터가 저장된 물리적 공간, 그리고 블록 파라미터 정보를 저장

```
SELECT table_name, tablespace_name FROM user_tables
WHERE table_name LIKE '교과%';
```

TABLE_NAME	TABLESPACE_NAME
1 교과목	STUDY
2 교과목2	STUDY
3 교과목3	STUDY
4 교과목4	STUDY
5 교과목5	STUDY
6 교과목6	STUDY

```
SELECT table_name, tablespace_name
FROM user_tables
WHERE table_name LIKE '교과%';
```

예) 사용자가 생성한 테이블에서 테이블명이 “교과”로 시작하는 테이블명과 테이블 스페이스명을 조회

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

2

USER_OBJECTS : 사용자가 생성한 테이블 정보와 함께 인덱스, 시퀀스, 동의어, 뷰 등과 같은 객체에 대한 이름, 종류, 생성 날짜 등의 정보를 저장

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_name LIKE '교과%' AND object_type='TABLE';
```

OBJECT_NAME	OBJECT_TYPE	CREATED
1 교과목	TABLE	21/12/03
2 교과목2	TABLE	22/02/03
3 교과목3	TABLE	22/02/03
4 교과목4	TABLE	22/02/03
5 교과목5	TABLE	22/02/03
6 교과목6	TABLE	22/02/03

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_name LIKE '교과%' AND object_type='TABLE';
```

예1) 사용자 객체에서 객체명이 “교과”로 시작하고, 객체의 종류가 테이블인 것만 조회하는데, 객체명, 객체 종류, 생성 날짜를 출력

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

2

USER_OBJECTS : 사용자가 생성한 테이블 정보와 함께 인덱스, 시퀀스, 동의어, 뷰 등과 같은 객체에 대한 이름, 종류, 생성 날짜 등의 정보를 저장

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_type='VIEW';
```

OBJECT_NAME	OBJECT_TYPE	CREATED
TESTVIEW	VIEW	22/01/12

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_type='VIEW';
```

예2) 사용자 객체에서 객체의 종류가 뷰인 것만 조회하는데, 객체명, 객체 종류, 생성 날짜를 출력

테이블 관리

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

2

USER_OBJECTS : 사용자가 생성한 테이블 정보와 함께 인덱스, 시퀀스, 동의어, 뷰 등과 같은 객체에 대한 이름, 종류, 생성 날짜 등의 정보를 저장

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_type='INDEX';
```

OBJECT_NAME	OBJECT_TYPE	CREATED
전공_PK	INDEX	21/12/03
2학부_PK	INDEX	21/12/03
교과목_PK	INDEX	21/12/03
교수_PK	INDEX	21/12/03
5학점_PK	INDEX	21/12/03
6_IDX_ST_SST	INDEX	22/01/12

```
SELECT object_name, object_type, created FROM user_objects
WHERE object_type='INDEX';
```

예3) 사용자 객체에서 객체의 종류가 인덱스인 것만 조회하는데,
객체명, 객체 종류, 생성 날짜를 출력

시퀀스 및 테이블 관리

테이블 관리

데이터 딕셔너리

+ 사용자 테이블 정보 조회

3

USER_CATALOG : 사용자 소유로 생성된 모든 객체명과 객체 종류에 대한 정보를 저장하며, TABLE_NAME 칼럼은 테이블명이 아니라 객체명이고, TABLE_TYPE 칼럼은 객체의 종류

```
SELECT * FROM user_catalog;
```

TABLE_NAME	TABLE_TYPE
SEQ_MYNO	SEQUENCE
TESTVIEW	VIEW
BIN\$0erPKUuIV6jgUwEAAH+PFA==\$\$0	TABLE
BIN\$0erPKUuKV6jgUwEAAH+PFA==\$\$0	TABLE
BIN\$0f1+h111aA3gUwEAAH8t7Q==\$\$0	TABLE
BIN\$0f1+h11LaA3gUwEAAH8t7Q==\$\$0	TABLE

```
SELECT * FROM user_catalog;
```

실습하기



시퀀스 및 테이블 관리

실습하기

- 수강 테이블의 구조만 복사하여 "수강_구조" 테이블을 생성하기 바랍니다.
- 수강 테이블의 학번, 수강년도, 수강학기, 교과목코드, 취득 점수 컬럼만 복사하여 "수강_구조일부" 테이블을 생성하기 바랍니다.
- 수강 테이블의 학번, 교과목코드, 취득 점수 컬럼만 복사하고, 데이터는 2022년 1학기 데이터만 복사하여 "수강_구조_데이터일부" 테이블을 생성하기 바랍니다.
- 수강 테이블의 구조를 그대로 복사하고, 데이터는 취득 점수가 80점 이상인 학생들의 데이터만 복사하여 "수강_취득점수" 테이블을 생성하기 바랍니다.

- + Oracle Database 19c 버전 사용
- + 최신 버전인 Oracle Database 21c는 현재 cloud에서만 사용 가능하며, 지원 기간은 2년

테이블 관리



실습단계
테이블 관리
수강 테이블의 구조만 복사하기
테이블의 일부 구조와 모든 데이터 복사하기
테이블의 일부 구조와 데이터 일부 복사하기
테이블의 모든 구조와 일부 데이터 복사하기
SEQ_수강_NO1 시퀀스 생성
"수강" 테이블 구조만 복사하여 "수강_장학생" 테이블 생성
"수강_장학생" 테이블에 "장학생순번" 컬럼 추가
"수강_장학생" 테이블의 컬럼 순서 변경(오라클 12이상 가능)
취득점수가 90점 이상인 학생들을 "수강_장학생" 테이블에 입력