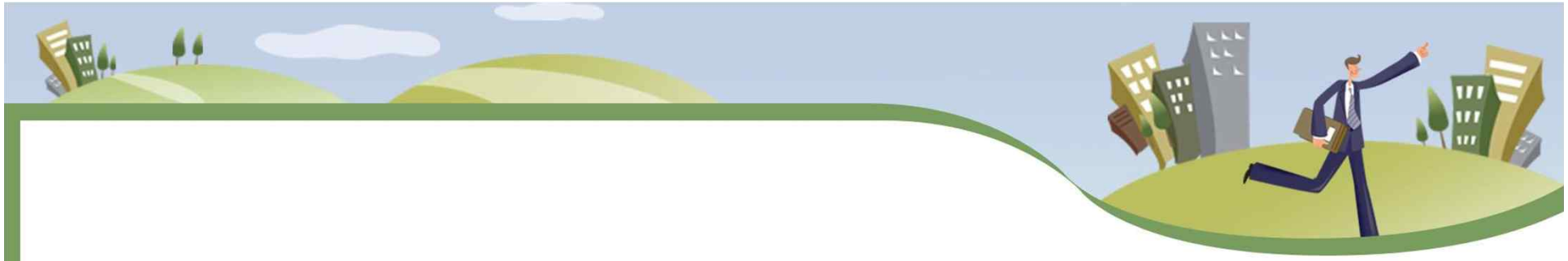
11. 오브젝트 관리



학습 목표

학습을 마친 후 여러분은 …

- 기본적인 데이터베이스 객체에 대해 설명할 수 있다.

- 테이블(Table), 뷰(View), 시퀀스(Sequence), 인덱스(Index), 동의어(Synonym)을 생성, 변경 및 삭제할 수 있다.

오브젝트 종류

- 테이블: 데이터를 저장한다.
- 인덱스:질의의 효율성을 높인다.
- 뷰 : 하나 이상의 테이블에 있는 데이터의 부분집합이다.
- 시퀀스: 기본 키 값을 생성한다.
- 시노님: 객체에 다른 이름을 제공한다.

오브젝트 관련 DDL

- 데이터베이스의 오브제트 구조를 생성, 변경, 삭제하는 명령

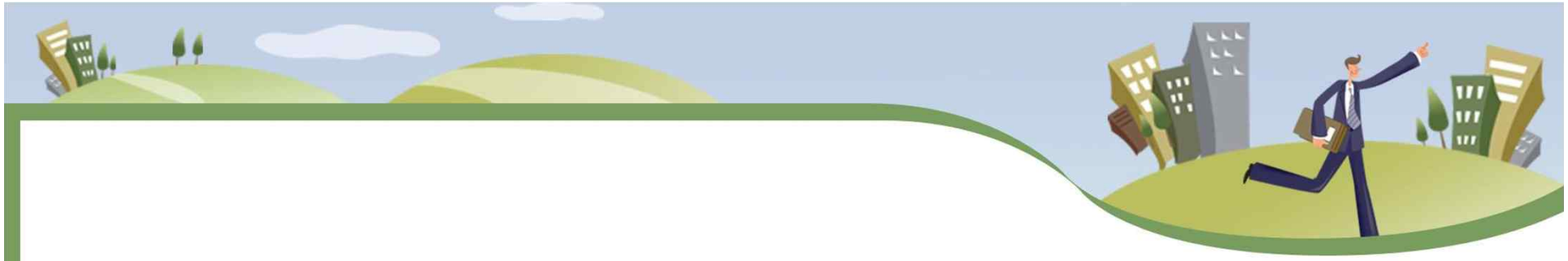
종 류	설 명
CREATE	데이터베이스 내의 모든 객체를 생성할 때 사용
ALTER	이미 생성된 객체의 구조를 변경할 때 사용
DROP	생성되어 있는 객체를 삭제할 때 사용
RENAME	이미 생성한 객체의 이름을 변경할 때 사용
COMMENT	객체이름, 컬럼에 대한 설명을 데이터베이스 내에 저장
TRUNCATE	테이블에 저장되어 있는 모든 행을 삭제할 때 사용

테이블

- 기본적인 데이터 저장 단위이다.
- 사용자의 접근 가능한 모든 데이터를 보유한다
- 레코드와 컬럼으로 구성된다.
- 테이블은 시스템내에서 독립적으로 사용되길 원하는 엔티티를 표현할 수 있다.
- 테이블은 두 엔티티간의 관계를 표현할 수 있다.

create table 구문

```
CREATE TABLE [schema.]table_name
 ( column datatype  [DEFAULT  ...],
   [, column datatype  . . .]
 )
  [TABLESPACE  tablespace ]
  [ PCTFREE  integer ]
  [ PCTUSED  integer ]
  [ INITRANS  integer ]
  [ MAXTRANS  integer ]
  [ STORAGE  storage-clause ]
  [ LOGGING | NOLOGGING ]
  [ CACHE | NOCACHE ] ;
```

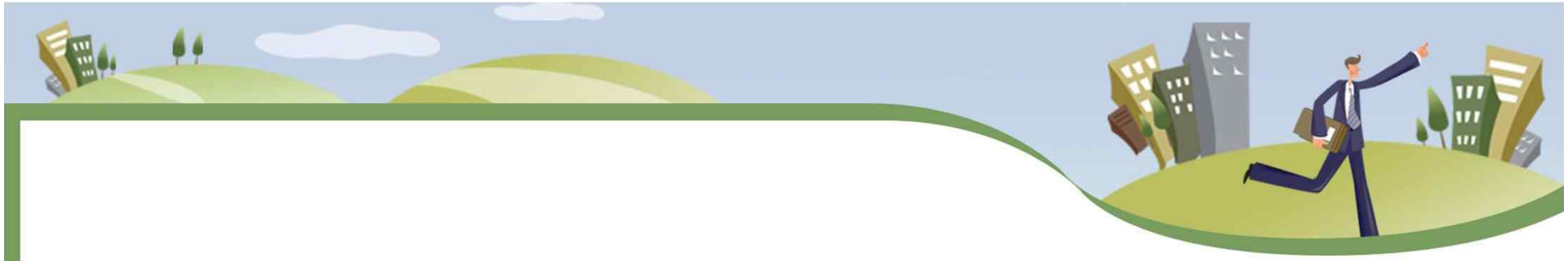


테이블 이름 지정 규칙

- 문자로 시작해야 한다.
- 1자부터 30자까지 가능한다.
- A-Z, a-z, 0-9, _, $, #만 포함해야 한다.
- 동일한 사용자가 소유한 다른 객체의 이름과 중복되지 않아야 한다.
- Oracle Server의 예약어가 아니어야 한다.
- 대소문자를 구분하지 않는다.

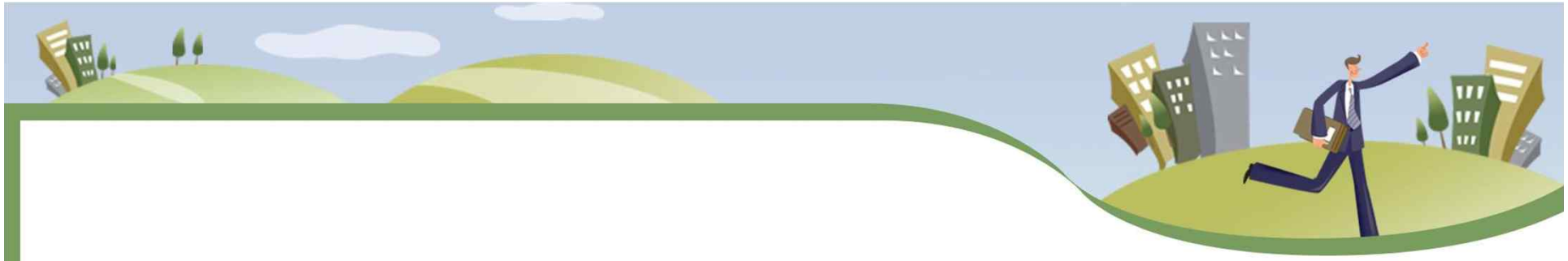
데이터 유형

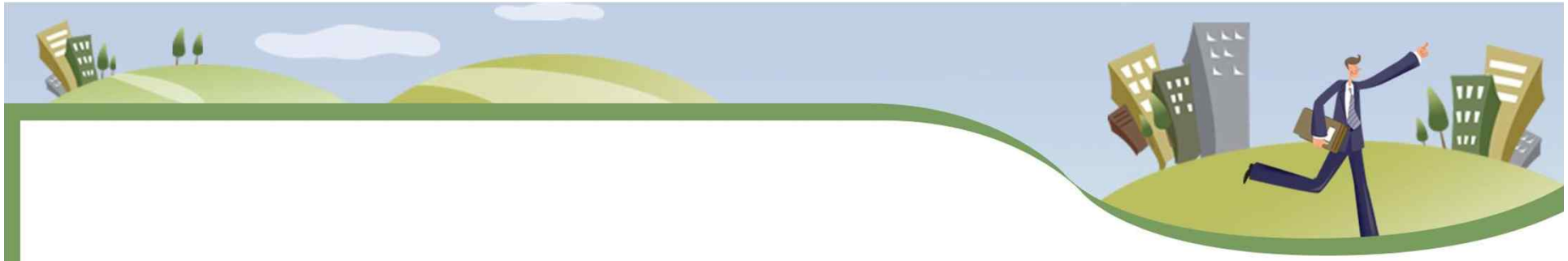
데이터 유형	설 명
VARCHAR2(size)	가변길이 문자데이터
CHAR(size)	고정길이 문자데이터
NUMBER(p,s)	가변길이 숫자데이터
DATE	날짜 및 시간값
LONG	최대 **2GB**의 가변 길이 문자데이터
CLOB	최대 **4GB**의 문자데이터
ROW 및 **LONG ROW**	원시 이진데이터
BLOB	최대 **4GB**의 이진데이터
BFILE	외부 파일에 저장된 이진데이터 **(**최대 **4GB)**
ROWID	테이블에서 행의 고유주소를 나타내는 **64**진수



데이터 유형

데이터 유형	설 명
TIMESTAMP	소수점 이하의 초까지 포함하는 날짜 **TIMESTAMP** **TIMESTAMP WITH TIME ZONE** **TIMESTAMP WITH LOCAL TIME**
INTERVAL YEAR TO MONTH	연수 및 개월 수로 기간을 저장
INTERVAL DAY TO SECOND	날짜 수, 시간 수, 분 수, 초 수로 기간을 저장





create table 구문

- CREATE TABLE 문과 AS *subquery* 옵션을 결합하여 테이블을 생성하고 행을 삽입한다.

```
CREATE  TABLE  table
              [(column, column  . . .)]
AS  subquery ;
```

- 지정한 열 수를 서브 쿼리와 일치 시킨다.
- 열 이름 및 기본 값을 사용하여 열을 정의 한다.

테이블 수정

새로운 열 추가

```
ALTER TABLE  table
ADD   (column datatype [DEFAULT  expr]
      [, . . .] ) ;
```

기존 열의 수정

```
ALTER  TABLE  table
MODIFY (column datatype [DEFAULT expr]
        [, . . .] ) ;
```

테이블 수정

열 삭제

```
ALTER TABLE   table
DROP  (column) ;
```

UNUSED 컬럼 지정

```
ALTER  TABLE  table
 SET      UNUSED  COLUMN  column ;
```

UNUSED 컬럼 삭제

```
ALTER  TABLE  table
DROP   UNUSED  COLUMNS ;
```

테이블 절단

- TRUNCATE TABLE 문
  - 테이블에서 모든 행을 제거한다
  - 해당 테이블이 사용하는 저장 공간을 해제 한다.

```
TRUNCATE TABLE table ;
```

- TRUNCATE 를 사용한 행 제거 작업은 롤백할 수 없다.

테이블 삭제

- 테이블의 모든 데이터 및 구조를 삭제한다.
- 보류중인 트랜잭션을 모두 커밋한다.
- 인덱스를 모두 삭제한다.
- DROP TABLE 문은 롤백할 수 없다.

```
DROP TABLE table ;
```

테이블 이름 변경 / 주석 추가

- 테이블, 뷰, 시퀀스, 시노님의 이름을 변경하려면 RENAME 명령을 사용한다.

```
RENAME old_name TO new_name;
```

- COMMENT 문을 사용하여 테이블 또는 열에 주석을 추가할 수 있다.

```
COMMENT ON TABLE table|COLUMN table.column
IS 'text';
```

제약조건

- 다음 유형의 제약 조건을 사용할 수 있다
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

- 제약조건을 사용하여 다음을 수행한다.
  - 테이블에 행이 삽입,갱신,삭제될 때마다 제약 조건을 만족해야 작업이 수행된다.
  - 다른 테이블에 종속된 테이블의 삭제를 방지한다.

제약조건

```
CREATE  TABLE  table
       ( column   datatype  [DEFAULT  expr]
       [column_constraint ],

                          );
```

```
CREATE  TABLE  emp
       empno     NUMBER(4)  CONSTRAINT  empno_pk
                            PRIMARY KEY,
         deptno   NUMBER(2)  NOT NULL );
```

NOT NULL

- 해당 열에 널 값을 허용하지 않는다

EMPNO	ENAME	...	COMM	DEPTNO
7839	KING			10
7698	BLAKE			30
7782	CLARK			10
7566	JONES			20

NOT NULL 제약조건
(이 열의 모든 행은 널 값을 가질 수 없다)

NOT NULL 제약조건 없음
(이 열은 널 값을 가질 수 있다)

UNIQUE

DEPT

DEPTNO	DNAME	LOC_CODE
10	ACCOUNTING	A1
20	REARCH	B1
30	SALES	C1
40	OPERATION	A1

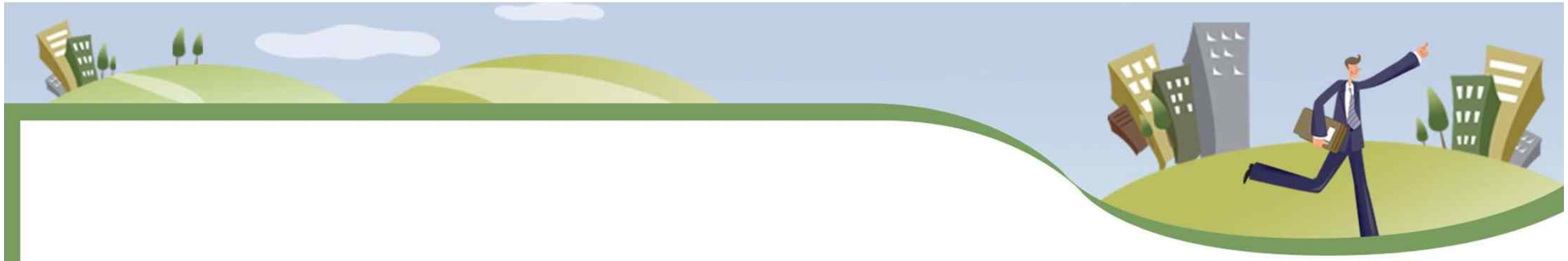
UNIQUE 제약조건

입력되지 않음

50	REARCH	C1
50		C1

(DNAME에 RESEARCH 있음)

입력됨



PRIMARY KEY

➤ 테이블 레벨 또는 열 레벨로 정의한다.

```
CREATE  TABLE  emp
         empno      NUMBER(4) CONSTRAINT empno_pk PRIMARY KEY,
         deptno     NUMBER(2)   ,
         );
```

```
CREATE  TABLE  emp
         empno      NUMBER(4)   ,
         deptno     NUMBER(2)   ,
        CONSTRAINT empno_pk  PRIMARY KEY (empno)
         );
```

FOREIGN KEY

- 동일한 테이블이나 다른 테이블에 있는 기본 키 또는 고유 키 열과의 참조 관계를 설정한다.

```
CREATE  TABLE  emp
        empno     NUMBER(4)  PRIMARY KEY,
        deptno    NUMBER(2)            CONSTRAINT deptno_fk
                                REFERENCES dept(deptno) ,
        job       VARCHAR2(10)
        );
```

```
CREATE  TABLE  emp
        empno     NUMBER(4)   PRIMARY KEY,
        deptno    NUMBER(2)   ,
        job       VARCHAR2(10),
                    CONSTRAINT deptno_fk  FOREIGN KEY( deptno)
                    REFERENCES dept(deptno)
        );
```

CHECK

- 각 행이 만족해야 하는 조건을 정의한다.

```
CREATE   TABLE   emp
       empno     NUMBER(4)    PRIMARY KEY,
        deptno    NUMBER(2)    CONSTRAINT deptno_ck
                   CHECK (deptno BETWEEN 10 AND 77)
                      ,..
       );
```

제약 조건 추가 / 삭제

제약 조건 추가

```
ALTER  TABLE  table
ADD    [CONSTRAINT  constraint] type  (column) ;
```

제약 조건 삭제

```
ALTER  TABLE  table
DROP   CONSTRAINT  constraint  [CASCADE] ;
```

view

뷰(View)란?

- 테이블 또는 다른 뷰를 기반으로 하는 논리 테이블이다.
- 자체적으로 데이터를 포함하지는 않는다
- 뷰를 통해 테이블의 데이터를 보거나 변경이 가능하다.

뷰(View) 사용목적

- 데이터 액세스를 제한하기 위해
- 복잡한 질의를 쉽게 작성하기 위해
- 데이터 독립성을 제공하기 위해
- 동일한 데이터로부터 다양한 결과를 얻기 위해

create view 구문

```
CREATE [OR REPLACE] [FORCE|NOFORCE]
      VIEW view [(alias [, alias] …) ]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint] ]
[WITH READ ONLY [CONSTRAINT constraint] ] ;
```

view 생성

- 부서 10번 사원의 정보를 포함하는 뷰 생성

```
CREATE   VIEW      emp_v10
AS SELECT                    empno , ename , job
FROM                         emp
WHERE                        deptno =10;
```

- DESCRIBE 명령을 사용하여 뷰의 구조를 표시한다

```
DESCRIBE     emp_v10
```

- DROP VIEW 명령으로 VIEW를 제거한다.

```
DROP     VIEW      emp_v10  ;
```

시퀀스

시퀀스란?

- 고유번호를 자동으로 생성한다.
- 공유 가능한 객체이다.
- 일반적으로 기본키값을 생성하는데 사용된다.
- 응용 프로그램 코드를 대체한다.
- 시퀀스값을 메모리에 캐시하면 액세스 효율이 높아진다

create sequence 구문

```
CREATE SEQUENCE sequence
        [INCREMENT BY n]
        [START WITH n]
        [{MAXVALUE n | NOMAXVALUE}]
        [{MINVALUE n | NOMINVALUE}]
        [{CYCLE | NOCYCLE}]
        [{CACHE n | NOCACHE}];
```

NEXTVAL 및 CURRVAL 의사 열

- NEXTVAL은 사용 가능한 다음 시퀀스값을 반환하며, 참조될때마다(서로 다른 사용자일지라도) 고유한 값을 반환한다.
- CURRVAL은 현재 시퀀스 값을 반환한다.
- CURRVAL이 값을 포함하려면 먼저 해당 시퀀스에 대해 NEXTVAL이 수행되어야 한다.

시퀀스 수정

```
ALTER   SEQUENCE  sequence
        [INCREMENT  BY  n]
        [{MAXVALUE  n | NOMAXVALUE}]
        [{MINVALUE  n | NOMINVALUE}]
        [{CYCLE  | NOCYCLE}]
        [{CACHE  n | NOCACHE}] ;
```

시퀀스 제거

```
DROP    SEQUENCE  sequence ;
```

인덱스

인덱스란?

- 스키마 객체이다.
- Oracle Server에서 포인터를 사용하여 행의 검색속도를 높이기 위해 사용한다.
- 데이터 위치를 빠르게 찾는 신속한 경로 액세스 방법을 사용하여 디스크 I/O를 줄여준다.
- 인덱스는 테이블과 독립되어 존재한다.
- Oracle Server에 의해 사용되며, 자동으로 유지 관리된다.

인덱스 생성 구문

- 하나 이상의 열에 대한 인덱스를 생성한다.

```
CREATE  INDEX  index
ON  table  (column[, column] . . .) ;
```

- EMP 테이블의 ENAME에 대한 질의 속도를 향상 시킨다.

```
CREATE  INDEX  emp_ename_idx
ON                    emp(ename) ;
```

- 인덱스 제거

```
DROP  INDEX  emp_ename_idx;
```

시노님

- 시노님(Synonym)이란?
  - 다른 사용자가 소유한 테이블을 쉽게 참조한다.
  - 긴 객체 이름을 짧게 만듭니다.

- 시노님 생성

```
CREATE  SYNONYM  synonym
FOR   object ;
```

- 시노님 삭제

```
DROP  SYNONYM  synonym ;
```