

PL/SQL

OLACLE



목차

I. 데이터베이스 이해

II. 데이터베이스를 구성하는 객체 이해

III. SQL 기본

IV. SQL 함수

V. 그룹 쿼리와 집합 연산자

VI. 조인(Join)과 서브쿼리(SubQuery)

VII.PL/SQL

목차

II. 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

- 1) 테이블 (Table)
- 2) 뷰 (View)
- 3) 인덱스 (Index)
- 4) 시노님 (Synonym)
- 5) 시퀀스 (Sequence)

2. 데이터베이스 SQL

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

- 데이터베이스 객체란?

✓ 데이터베이스 내에 존재하는 논리적인 저장 구조. (테이블, 뷰, 인덱스, 시노님, 시퀀스 등)

데이터베이스 객체	설 명
테이블(table)	데이터를 담고 있는 객체
뷰(view)	하나 이상의 테이블을 연결해 마치 테이블인 것처럼 사용하는 객체
인덱스(index)	테이블에 있는 데이터를 빠르게 찾기 위한 객체
시노님(synonym)	데이터베이스 객체에 대한 별칭을 부여한 객체
시퀀스(sequence)	일련번호 채번을 할 때 사용되는 객체
함수(function)	특정 연산을 하고 값을 반환하는 객체
프로시저(procedure)	함수와 비슷하지만 값을 반환하지 않는 객체
패키지 (package)	용도에 맞게 함수나 프로시저를 하나로 묶어 놓은 객체

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- ✓ 데이터를 넣고 수정하고 삭제하는, 즉 데이터를 담고 있는 객체가 테이블이다.
- ✓ DBMS 상에서 가장 기본적인 객체로 로우(행)와 컬럼(열)으로 구성된 2차원 형태(표)의 객체.

항목	내용
특징	<ul style="list-style-type: none">▪ 개체를 구분하는 고유 이름을 가진다.▪ 행(row, record)과 열(column, field)들로 구성▪ 행과 열을 순서를 가지지 않는다.▪ 하나의 행은 하나의 테이블에 속한다
제약 사항	<ul style="list-style-type: none">▪ 한 데이터베이스 내에서의 테이블 이름은 고유▪ 한 테이블 내에서의 컬럼 이름들은 고유▪ 한 테이블 내에서 열들을 고유▪ 같은 열을 두번 나타낼 수는 없다

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 테이블 생성

CREATE TABLE : 모든 상응하는 컬럼과 자신의 데이터타입을 갖는 새로운 테이블을 만든다.

```
CREATE TABLE <테이블 명>
( 필드명1 TYPE [NOT NULL | NULL],
  필드명2 TYPE [NOT NULL | NULL],
  .
  .
  Constraint 인덱스키명 Primary Key (필드명1 [,필드명2])
  Constraint 외부키명 Foreign Key (필드명2)
    References 외부테이블명(외부필드명)
)
```


II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 테이블 생성

예 제 1)

```
CREATE TABLE ex2_1 (  
    COLUMN1 CHAR(10) ,  
    COLUMN2 VARCHAR2(10) ,  
    COLUMN3 NVARCHAR2(10) ,  
    COLUMN4 NUMBER  
);
```

- 데이터 타입

- ✓ 데이터 타입 (DataType) 이란 컬럼이 저장 되는 데이터 유형.
- ✓ 종류 : 문자 데이터 타입, 숫자 데이터 타입, 날짜 데이터 타입, LOB (Large Object) 데이터 타입, Null

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 데이터 타입

- ✓ 데이터 타입 (DataType) 이란 컬럼이 저장 되는 데이터 유형.
- ✓ 종류 : 문자 데이터 타입, 숫자 데이터 타입, 날짜 데이터 타입, LOB (Large Object) 데이터 타입, Null

- 문자 데이터 타입

데이터타입	설명
CHAR(크기[BYTE, CHAR])	고정길이 문자, 최대 2000byte, 디폴트 값은 1byte
VARCHAR2(크기[BYTE, CHAR])	가변길이 문자, 최대 4000byte, 디폴트 값은 1byte
LONG	최대 2GB 크기의 가변길이 문자형, 잘 사용하지 않음

- ✓ LONG 형은 숫자가 아닌 문자 데이터 타입으로, 오라클에서는 long타입을 더는 사용하지 않도록 권고.
(CLOB, BLOB으로 대체 가능하며, 하위 버전의 호환성을 이유로 11g에서 지원)

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 데이터 타입

- ✓ 데이터 타입 (DataType) 이란 컬럼이 저장 되는 데이터 유형.
- ✓ 종류 : 문자 데이터 타입, 숫자 데이터 타입, 날짜 데이터 타입, LOB (Large Object) 데이터 타입, Null

- 숫자 데이터 타입

데이터타입	설명
NUMBER[(p,[s])]	가변숫자, p(1~31, 디폴트 값은 38)와 s(-84~127, 디폴트값은 0) 는 십진수 기준, 최대 22byte
FLOAT[(p)]	NUMBER의 하위 타입, p는 1~128, 디폴드 값은 128, 이진수 기준, 최대 22byte
BINARY_FLOAT	32비트 부동소수점 수, 최대 4byte
BINARY_DOUBLE	64비트 부동소수점 수, 최대 8byte

- ✓ 숫자 타입은 4가지가 있지만 NUMBER 형만 사용할 때가 많다. 다른 DBMS 는 INTEGER 와 같은 정수형, DECIMAL 과 같은 실수형을 제공한다. 오라클도 이런 타입으로 컬럼을 생성 할 수 있지만 내부적으로는 모두 NUMBER 형으로 변환되어 생성된다.

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 데이터 타입

- ✓ 데이터 타입 (DataType) 이란 컬럼이 저장 되는 데이터 유형.
- ✓ 종류 : 문자 데이터 타입, 숫자 데이터 타입, 날짜 데이터 타입, LOB (Large Object) 데이터 타입, Null

- 날짜 데이터 타입

데이터타입	설명
DATE	BC 4712년 1월 1일부터 9999년 12월 31일, 연,월,일,시,분,초까지 입력 가능하다.
TIMESTAMP[(fractional_seconds_precision)]	연도, 월, 일, 시, 분, 초는 물론 밀리초까지 입력 가능하다 fractional_seconds_precision 은 0~9 까지 입력 할 수 있고 디폴드 값은 6이다.

- ✓ 가장 일반적이고 많이 사용하는 날짜 데이터 타입은 DATE이며 연,월,일과 시,분,초 까지 입력가능.
- ✓ TIMESTAMP는 초 단위를 더 세부적으로 입력 할 수 있다.

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- LOB 데이터 타입

데이터 타입	설명
CLOB	문자형 대용량 객체, 고정길이와 가변길이 문자 집합 지원, 최대 크기는(4GB-1) * (데이터베이스 블록 사이즈)
NCLOB	유니코드(다국어 지원)를 포함한 문자형 대용량 객체, 최대 크기는(4GB-1) * (데이터베이스 블록 사이즈)
BLOB	이진형 대용량 객체, 최대 크기는 (4GB-1) * (데이터베이스 블록 사이즈)
BFILE	대용량 이진 파일에 대한 로케이터(위치, 이름) 저장, 최대 크기는 4GB

- ✓ 일반적인 정형 데이터(문자, 숫자, 날짜 등 구조화된 데이터) 에 비해 텍스트, 그래픽, 이미지, 사운드 등 비정형 데이터는 그 크기가 매우 큰데 , 이런 데이터를 저장하는데 LOB 타입 사용.
- ✓ 문자형 대용량 데이터는 CLOB, 그래픽,이미지,동영상 등 데이터는 BLOB, NCLOB 사용
- ✓ BFILE는 실제 이진 파일이 아닌 데이터베이스 외부에 있는 파일에 대한 로케이터를 저장(수정할 수 없고 읽기만 가능)

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 제약조건 (Constraints)

- ✓ 컬럼에 대한 속성 형태로 정의하지만 엄연히 오라클 데이터베이스 객체 중 하나이며 데이터 무결성을 보장하기 위한 용도로 사용된다.

제약조건이란 ?

제약조건이란 테이블에 부적절한 자료가 입력되는 것을 방지하기 위해서 여러 가지 규칙을 적용해 놓는 거라 생각하면 됩니다. 간단하게 테이블 안에서 데이터의 성격을 정의하는 것이 바로 제약조건 입니다.

- 모든 CONSTRAINT는 데이터 사전(DICTIONARY)에 저장 됩니다.
- 의미 있는 이름을 부여했다면 CONSTRAINT를 쉽게 참조할 수 있습니다.
- 표준 객체 명명법을 따르는 것이 좋습니다.
- 제약조건은 테이블을 생성할 당시에 지정할 수도 있고, 테이블 생성 후 구조변경(ALTER)명령어를 통해서도 추가가 가능합니다.

제약조건	설명
NOT NULL(NN)	열은 NULL값을 포함할 수 없습니다.
UNIQUE KEY(UK)	테이블의 모든 행을 유일하게 하는 값을 가진 열(NULL을 허용)
PRIMARY KEY(PK)	유일하게 테이블의 각행을 식별 (NOT NULL과 UNIQUE조건을 만족)
FOREIGN KEY(FK)	열과 참조된 열 사이의 외래키 관계를 적용하고 설정합니다.
CHECK(CK)	참이어야 하는 조건을 지정함(대부분 업무 규칙을 설정)

II. 장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table. 제약조건

- NULL

- ✓ ‘값이 없음’ 을 의미하며 테이블을 생성할 때 컬럼 속성에 기술한다.
- ✓ 디폴드 값이 NULL 이므로 별도로 지정하지 않으면 해당 컬럼은 NULL을 허용한다.
- ✓ NULL을 허용하지 않으려면 NOT NULL 구문을 명시해야 한다.
- ✓ SQL 문이나 PL/SQL 코드를 작성할 때 개발자가 가장 하기 쉬운 실수 중에 하나가 NULL 처리.
- ✓ DBMS 종류에 따라 NULL 처리가 조금씩 달라 DBMS 사용하는 개발자라면 혼동하기 쉽다.
[ex) NVL (ORACLE), ISNULL (MSSQL), IFNULL (MYSQL)].

- NOT NULL

- ✓ 컬럼을 정의할 때 NOT NULL 제약조건을 명시하면 해당 컬럼에는 반드시 데이터를 입력해야 한다.
- ✓ 제약 조건은 컬럼 속성처럼 보이지만 하나의 데이터베이스 객체이므로 고유의 이름이 있다.

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table.제약조건

- UNIQUE

- ✓ 컬럼에 들어가는 값이 유일해야 한다는 의미, 즉 중복 값을 허용하지 않는다.
- ✓ UNIQUE 제약 조건 역시 데이터 베이스 객체이므로, 컬럼 정의 부분에 UNIQUE를 명시하거나 CONSTRAINTS 키워드를 사용해 정의 할 수 있다.

UNIQUE

컬럼명 데이터타입 UNIQUE

또는

CONSTRAINTS 제약조건명 UNIQUE(컬럼명, ...)

II. 장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table. 제약조건

- 기본키 Primary key

- ✓ 보통 테이블에 키를 생성했다 라고 할 때, 이는 기본키를 의미한다. (PK)
- ✓ 기본키는 UNIQUE 와 NOT NULL 속성을 동시에 가진 제약조건 (유일함을 보장)
- ✓ 테이블 당 1 개의 기본키만 생성할 수 있다.

Primary Key

컬럼명 데이터타입 PRIMARY KEY

또는

CONSTRAINTS 제약조건명 PRIMARY KEY (컬럼명, ...)

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table.제약조건

- 외래키 Foreign Key

- ✓ 테이블 간의 참조 데이터 무결성(**referential integrity**) 을 위한 제약조건.
- ✓ 테이블을 연결시킨다.
- ✓ 반드시 참조하는 테이블이 먼저 생성되어야 하며, 참조키가 참조 테이블의 기본키로 만들어져 있어야 한다.
- ✓ 외래키에 사용할 수 있는 컬럼 개수는 최대 32개.
- ✓ 여러 컬럼을 외래키로 만들려면, 참조하는 컬럼과 외래키 컬럼의 순서와 개수는 같아야 한다.

Foreign Key

CONSTRAINT 외래키명 FOREIGN KEY (컬럼명, ...)

REFERENCES 참조 테이블 (참조 테이블 컬럼명, ...)

II. 장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table. 제약조건

- 외래키 Foreign Key

DEPARTMENT(DEPTNO, DEPTNAME, FLOOR)

EMPLOYEE(EMPNO, EMPNAME, TITLE, DNO, SALARY)

DEPARTMENT

DEPTNO	DEPTNAME	FLOOR
1	영업	8
2	기획	10
3	개발	9

EMPLOYEE

EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2000000
3426	박영권	과장	3	2500000
3011	이수민	부장	1	3000000
1003	조민희	대리	1	2000000
3427	최종철	사원	3	1500000

✓ 시스템 개발 초창기에는 외래키를 생성하지 않는다 (입력 삭제 문제)

II. 장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table. 제약조건

- CHECK

- ✓ 컬럼에 입력되는 데이터를 체크해 특정 조건에 맞는 데이터만 입력 받고 그렇지 않으면 오류를 리턴

CHECK

CONSTRAINT 체크명 CHECK (체크조건)

-DEFAULT

- ✓ 제약조건에는 포함되지 않지만, 컬럼 속성 중 하나
- ✓ 레코드를 입력할 때 해당 필드 값을 전달하지 않으면, 자동으로 설정된 기본값을 저장.

DEFAULT

컬럼명 DATE DEFAULT SYSDATE

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 테이블 삭제

- ✓ 해당 테이블은 물론 모든 데이터가 삭제 된다.
- ✓ CASCADE CONSTRAINTS를 붙이면 삭제할 테이블의 기본키와 UNIQUE 키를 참조하는 제약조건도 삭제됨.

DROP

```
DROP TABLE [스키마.]테이블명 [CASCADE CONSTRAINTS]
```

-테이블 변경

- ✓ 기존의 컬럼의 타입변경, 컬럼 추가, 삭제 등의 수정을 할 수 있다.

ALTER

```
ALTER TABLE [스키마.]테이블명 RENAME COLUMN 변경전컬럼명 TO 변경후컬럼명;
```

II.장 데이터베이스를 구성하는 객체 이해

1. 데이터베이스 객체

1) 테이블 Table

- 테이블 복사

- ✓ 해당 테이블은 물론 모든 데이터가 삭제 된다.
- ✓ CASCADE CONSTRAINTS를 붙이면 삭제할 테이블의 기본키와 UNIQUE 키를 참조하는 제약조건도 삭제됨.

테이블 복사

```
CREATE TABLE [스키마.]테이블 AS  
SELECT 컬럼1, 컬럼2  
FROM 복사할 테이블명;
```


감사합니다!