

# Database design guide

## (테이블, 컬럼 정의 방법)

2004년 11월 18일

Sunny Kwak

[sunnykwak@hanmail.net](mailto:sunnykwak@hanmail.net)

[sunnykwak.egloos.com](http://sunnykwak.egloos.com)

## Copyrights

© 2004 Sunny Kwak. All rights reserved.

Other disclaimers

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

## Version

Version 1.0 (draft)

## History

2004년 11월 18일 목요일 First created.

## Table of Contents

---

<b>1. INTRODUCTION .....</b>	<b>4</b>
1.1 OBJECTIVES .....	4
1.2 TARGET AUDIENCE .....	4
<b>2. 데이터베이스 설계 지침 .....</b>	<b>4</b>
2.1 테이블스페이스(TABLESPACE) 생성 규칙.....	4
2.2 테이블 생성 규칙 .....	4
2.2.1 테이블 명칭 부여.....	4
2.2.2 컬럼 명칭 부여 및 컬럼 설계 규칙.....	4
2.2.3 제약 조건 설계 (primary key, foregin key, null or not null) .....	6
<b>3. 오라클 데이터베이스 설계 규칙.....</b>	<b>6</b>
3.1 테이블 설계 규칙 .....	6
3.1.1 인덱스 설계 규칙.....	7

# 1. Introduction

## 1.1 Objectives

데이터베이스 설계 시 참고해야할 정보, 필수적으로 알아야할 지침 등을 정리하였다.

## 1.2 Target Audience

UbiFlow 프로젝트 매니저, 테크니컬 리더

# 2. 데이터베이스 설계 지침

## 2.1 테이블스페이스(tablespace) 생성 규칙

테이블스페이스는 데이터용 테이블 스페이스 공간과 인덱스용 테이블스페이스를 별도로 생성해야 한다.

## 2.2 테이블 생성 규칙

### 2.2.1 테이블 명칭 부여

- I 데이터베이스 제품 종류에 따라서 테이블 명칭의 대소문자가 구분되는 것(사이베이스 등)과 대소문자 구분이 되지 않는 제품(오라클 등)이 있다. 따라서, 데이터베이스 제품 종류에 투명(transparent)하며, 일관성 있는 테이블 설계를 위해 모든 테이블 명칭은 소문자만으로 구성한다.
- I 테이블 명칭은 단위 어플리케이션 명칭(메일, 결재, 게시)과 각 테이블의 기능 혹은 역할을 표현하는 단어(문서, 첨부, 사용자) 등 둘 이상의 단어를 합성하여 명명한다.
- I 테이블 명칭 내에서 각 단어 간의 구분은 '\_'(underline)을 사용하도록 한다.
- I 테이블 명칭의 두번째 이후의 단어는 가능한 표준 영어 명사를 사용하도록 하며, 어절의 길이가 길어 짧게 표현할 필요가 있을 경우 이외에는 가급적 약어 혹은 한글 발음 표기를 사용하지 않도록 한다.
- I 테이블 명칭의 길이는 가급적 최대 20자리를 넘지 않도록 한다

pt_user	: 공통 시스템의 사용자 정보
pt_group	: 공통 시스템의 부서 정보
pt_sign_doc	: 결재 시스템의 결재 문서

### 2.2.2 컬럼 명칭 부여 및 컬럼 설계 규칙

- I 컬럼 명칭은 필히 소문자만으로 구성한다.
- I 컬럼 명칭은 가능한 둘 이상의 어절을 합성하여 구성한다.
- I 컬럼 명칭의 첫번째 어절은 컬럼의 분류 혹은 테이블 명칭의 약자를 나타내도록 한다.

doc(문서), mail(편지), att(첨부), code(코드)
--------------------------------------

- I 컬럼 명칭의 두번째 어절은 컬럼에 저장되는 데이터의 용도를 나타내도록 한다.

id, name, subject, filepath

- I 입력 문자열의 길이가 일정한 경우 가능한 char 타입을 사용하며 varchar2 혹은 varchar 타입은 가급적 사용하지 않도록 한다. (디스크 공간의 효율적 활용 및 insert, update 시 속도 향상과 fragmentation 방지를 위해 varchar 보다는 char 타입이 효율적이다. char 타입의 출력 시 right trim 처리가 필요할 수 있으므로 주의. 단, update가 많이 발생하지 않는 경우에는 varchar를 사용해도 무방하다고 봄.)
- I 날짜 혹은 시간을 표현할 경우 date, datetime 타입을 사용해서는 안되며, char 타입을 사용하도록 한다. 년 월일을 저장할 경우에는 8자리, 시분초까지 저장할 경우에는 14자리를 사용한다.

char(8) : 날짜(YYYYMMDD), char(14) : 날짜, 시간(YYYYMMDDHHMI24SS)

- I varchar2 혹은 varchar 타입을 사용할 경우, 2000 byte 이상의 데이터를 저장할 필요가 있을 경우, long 혹은 text 타입을 사용해야 한다. 또한, 가변 길이 문자열 컬럼을 정의할 경우, 오라클에서는 varchar2 타입을 기타 데이터베이스에서는 varcahr 타입을 사용한다.
- I 입력 문자열의 길이가 짧은 경우 가급적 long, text 타입을 사용하지 말아야 한다. long 혹은 text 타입은 각종 함수(nvl,ltrim), 인덱스(index), like 검색 등을 사용할 수 없다. 또한 하나의 테이블에 둘 이상의 long 타입 컬럼을 만들 수 없다. 단, text 타입은 둘 이상의 컬럼을 허용한다.
- I 컬럼 내에 데이터가 저장될 때, ',', ';' 등의 구분자(delimiter)가 포함되는 경우는 가급적 없도록 설계 시 고려한다. 구분자를 포함한 데이터를 저장하고자 하는 경우에는 하위 테이블(child table)을 별도로 추가한 후 구분자를 이용해 분할 데이터를 하위 테이블에 저장한다.
- I 모든 ID 관련 컬럼은 8~10 byte, 이름 항목은 40 byte, 제목 및 설명은 255 byte, 문서 내용에 관한 내용은 2G byte의 크기를 권장한다.
- I 다음 테이블에 기술하는 항목 명칭들은 관습적으로 사용되는 컬럼 명칭이다. 이외의 사례는 'UbiFlow naming rule' 문서를 참조하라.

[ Table 1 ] 테이블 명칭

한글 명칭	컬럼 명칭	example
사용자	user	user_id, user_name
부서	group	group_id, group_name
기관	org	org_id, org_tel, org_fax
전화	tel	user_tel, group_tel
팩스	fax	user_fax
직위	title	user_title, user_titlename
직책	todo	user_todo, user_todonaame
개수	cnt	doc_attcnt, doc_refcnt, child_cnt
순번	seqno	add_pos_seqno, att_seqno
순서	order	display_order
문서	doc	doc_yearmon, doc_number, doc_type
수신	recv	recv_date

예약	resv	resv_date
참조	ref (reference)	doc_refcnt
문서함	folder	folder_type, folder_id
설명	comment	folder_comment, type_comment
제목	subject	doc_subject
대장 혹은 등록	reg (register)	reg_date, reg_method

### 2.2.3 제약 조건 설계 (primary key, foreign key, null or not null)

- I 기본 키(primary key)의 명칭은 테이블 명칭의 앞에 pk\_를 추가한다.
- I 기본 키는 가급적 인덱스용 테이블스페이스에 생성토록 한다.
- I insert query 시 성능 향상을 도모할 경우에는 가급적 foreign key를 지정하지 말아야 한다. 설계서 상에는 논리적으로 선언해야 하며, 개발 완료 후 운영용 데이터베이스에서는 삭제해도 무방하다. (개발 및 테스트 단계에서 참조 무결성이 검증되었다고 판단한다.)
- I 프로그램 개발 완료 후 각 테이블의 데이터를 추출(select)하는 모든 쿼리를 검사한다. 이중에서 가장 많은 빈도로 호출되는 쿼리를 선택한 후, 쿼리의 응답 속도가 최상이 될 때 까지 인덱스를 조정한다.
- I 최빈도의 쿼리에 대한 인덱스 형성 후, 그보다 낮은 빈도로 접근하는 쿼리에 대한 인덱스를 생성하거나, 두 쿼리의 성능에 균형(balance)을 조절할 수 있도록 첫번째 인덱스를 조정한다.
- I 앞서 하나 혹은 두 인덱스에 의해서 다른 쿼리의 성능이 떨어지는 현상이 발생하지 않는지 확인한다.
- I 가급적 null 값이 허용되고, null 데이터를 많이 포함하는 컬럼은 인덱스 대상에서 제외한다.
- I 데이터 값이 편차가 적은 컬럼 또한 인덱스 대상에서 제외한다.

igt\_log의 userlog\_type, igt\_group의 org\_id

- I 인덱스 생성으로 인한 성능 향상을 얻을 수 있는 경우는 인덱스를 포함하는 테이블이 데이터가 많거나, 향후 행(row)의 수가 급격히 증가할 것이 예상되는 경우이다.
- I 일반적으로 1000 ~ 10000건 이하 행을 지니는 테이블에서는 인덱스를 만들어도 성능 향상 효과가 없거나, 최적화기(optimizer)에 의해서 인덱스가 무시될 수도 있다.
- I 인덱스가 실제 사용되는지 여부를 알기 위해서는 오라클의 경우, plan table을 사용하거나, 싸이베이스 혹은 MS-SQL 서버의 경우에는 set showplan on 명령을 사용하여야 한다.
- I 하나의 테이블에 가급적 3 개 이상의 인덱스를 만들지 않아야 한다. 인덱스가 많을 경우, 추가(insert), 수정(update) 및 삭제(delete) 시 쿼리의 반응 속도가 느려지게 되며, 디스크 사용량이 급증한다.

## 3. 오라클 데이터베이스 설계 규칙

### 3.1 테이블 설계 규칙

- I 모든 테이블 생성 시 테이블스페이스 이름을 명시하며, PCTFREE 10, PCTUSED 80,

PCTINCREASE 0 등 세 가지 설정을 추가한다.

```

/* 환경 설정 정보 */
CREATE TABLE pt_ini
(
    ini_name    varchar2(40)    NOT NULL,    /* 환경 설정 항목 명칭    */
    ini_value   varchar2(255)   NOT NULL,    /* 환경 설정 값          */
    ini_explain varchar2(255)   NULL        /* 환경 설정 항목 설명    */
)
PCTFREE 10
PCTUSED 80
TABLESPACE gw_data
STORAGE ( PCTINCREASE 0 )
/

```

### 3.1.1 인덱스 설계 규칙

- I 인덱스 생성 시 테이블스페이스 이름을 명시하며, STORAGE 옵션을 다음과 같이 설정한다.

```

ALTER TABLE PT_ini ADD CONSTRAINT pk_ini PRIMARY KEY ( ini_name )
USING INDEX TABLESPACE gw_index
STORAGE(INITIAL 32K NEXT 32K MINEXTENTS 1 MAXEXTENTS 121 PCTINCREASE 0)
/

```

- I 인덱스 생성을 위해 storage 크기는 8K \* 4배수로 결정하며, 다음의 표를 참조하여 생성한다.

**[ Table 2 ]** 인덱스 저장 용량 산정

페이지 크기	테이블 속성	적용 테이블 사례
32K	insert rarely, delete rarely	문서 번호 관리(doc_number), 환경 설정(ini), 서버 설정(server), 볼륨 설정(volume), 부서 및 사용자(group, user) 등
128K	insert frequently, delete frequently or insert rarely, but index record size is long	메모(memo), 메일 문서함(mail_doc_folder) 및 첨부(mail_attach), 게시판 문서 관련 테이블(bbs_doc) 등
512K	insert frequently, delete rarely	주소록 관리(addrbook), 결제 경로 관리, 공람 내역 등
2M	insert very frequently, delete rarely	결제 대장, 결제 문서 수발신 기록, 결제 날인 기록 등