DB naming rule

2020년 11월 30일 월요일

오전 9:42

## １. Database 관련 Naming Rule

### 가. Database Schema Name

#### *1) 규칙*

█ Database Profile 이름을 의미함

█ DB Alias 이름과 동일하게 함

█ 영문 소문자로만 혹은 대문자로만 작성함

* **대소문자를 섞지 않는다. 예약어들과 구분 짓기 위함.**
* **ORM에서 자동으로 테이블을 생성하면 소문자로 출력된다.**

█ Database Short Name의 길이는 최대 10자리를 넘을 수 없음

█ Database Short Name은 각 Site의 Unique한 Name을 사용함

#### *2) 표기 방식*

<Database Short Name>  
예) TOURDB, ETKP, TKS…

### 나. Table Name

#### *1) 규칙*

█ 테이블명은 대문자로만 혹은 소문자로만 사용함

█ 시스템 구분 코드와 모듈구분코드로 업무 영역을 구분함

█ 의미 있는 테이블명은 3단어까지 사용할 수 있음

█ 단어와 단어 사이는 ‘\_’로 구성함

█ 각 단어는 최대 10자리까지 사용함

█ 구분명은 Table의 특성을 나타냄

█ 예로는 master, detail, control, summary, trigger, history 등이 있음

#### *2) 표기 방식*

<시스템 구분(대메뉴)> + ‘\_’ + <의미 있는 테이블명>

예) 사용자 테이블: ACT\_USERS

### 다. Column Name

#### *1) 규칙*

█ 물리명은 영문 대문자를 이용함. 논리명을 사용자가 알 수 있는 정도에서 명사 및 명사형동사를 사용함

█ Column에 대한 자리 수는 총 12자리로 하며, 제한은 없음

단, 사용하는 Database의 특성에 따라 제한될 수 있음

█ Word와 Word 사이에는 ‘\_’로 구분함

█ 한 Word는 10자리를 넘을 수 없음

█ 모든 Column은 Dictionary List에 등록된 약어사전 및 자료사전을 기초로 작성함

█ Dictionary List에 등록되지 않은 약어는 책임자의 동의 하에 등록함

█ Column Name은 약어의 조합으로 구성

█ 컬럼명에 컬럼을 대표하는 접미사를 사용하여 컬럼명의 성격을 나타냄.

#### *2) 표기방식*

<의미있는 컬럼명> 혹은 <의미있는 컬럼명> + ‘\_’ + 접미사  
종종 자주 사용하는 접미사는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| 접미사 | 내용 | 설명 |
| \_CD | CODE | 주로 코드 테이블의 코드, 각종 코드에 사용된다. 숫자나 문자로 이루어진 코드에 해당되며, 숫자나 문자의 각 부분이 의미가 있는 경우에 코드를 사용한다. 대부분 PK에 해당한다. 예) 대분류 코드 CTGRY\_CD, 시도코드 SIDO\_CD, 사용자 그룹 코드 USER\_GROUP\_CD 등 |
| \_NM | NAME | 코드에 대한 명칭에 주로 사용된다. 논리명이 이름, 명칭인 경우에 해당된다. 예) 사용자이름 USER\_NM, 자원명 RES\_NM, 중분류 코드명 DVSN\_NM, 메뉴명 MENU\_NM |
| \_NO | NUMBER | 숫자로만 이루어진 경우, 주로 논리명이 번호인 경우에 사용. 예) 주민등록번호 JUMIN\_NO, 조문번호 JO\_NO, 게시물번호 BOARD\_NO |
| \_SQ | SEQUENCE | 오라클의 Sequence, MSSQL의 Identity의 경우에 사용한다. 숫자 일련번호로 PK를 설정할 경우 SQ를 사용한다. MSSQL의 Identity의 경우 주로 \_ID를 사용하는 경우가 많은데, 사용자 아이디 – USER\_ID의 ID와 의미가 틀려 SQ를 사용한다. 예) 작업번호 WORK\_SQ, 이력번호 HISTORY\_SQ |
| \_ID | ID | 주로 사용자 아이디의 경우에 사용한다. 예) 사용자아이디 USER\_ID, 등록자아이디 REG\_ID |
| \_DT | DATE | 날짜의 경우 사용한다. DT는 날짜 타입이 DATE형인 경우에만 사용한다. 보통 날짜의 경우 CHAR(8)형으로 20050718식으로 저장을 많이 한다. 이런 경우에는 \_YMD를 사용한다. 예) 삭제일자 DEL\_DT, 변경일자 CHG\_DT |
| \_YMD | YYYYMMDD | 날짜의 경우 사용한다. 날짜 타입이 CHAR 인경우 사용한다. 년월일인 경우 \_YMD를 사용하고, 년월형식으로 CHAR(6)로 저장될 경우 \_YM을 사용한다. 년도, 월, 일자 인경우에는 YEAR, MONTH, DAY등의 컬럼명을 사용한다. |
| \_GB | 구분 | 구분값을 나타낼 때 사용한다. CD는 주로 코드테이블을 별도로 사용할 때 적당하고, 테이블 없이 코드상에서 구별할 때 사용한다. 가령 사용자구분 필드가 있을 때 일반사용자, 내부사용자가 있다면 별도의 사용자 그룹테이블로 분리하여 사용할 경우 GROUP\_CD가 필드명이 되지만, 코드상에서 일반(G), 내부(I)로 사용하기로 결정했다면 GROUP\_GB 필드명을 사용하면 된다. 예) 통계구분 STAT\_GB |
| \_ST | STATE | 상태값이다. 주로 CHAR(1) 형식을 사용한다. 예) 사용자 상태 USER\_ST |
| \_FL | FLAG | 플레그값이다. 종종 삭제하지 않는 테이블에 삭제플레그를 많이 사용된다. 값은 0/1 이나 Y/N를 많이 사용한다. 예) 삭제여부 DEL\_FL, 요청여부 REQ\_FL |
| \_ORD | ORDER | 순서를 나타낼 때 사용한다. 예) 컬럼순서 COLUMN\_ORD |
| \_CNT | COUNT | 예) 조회수 VIEW\_CNT |
| \_AMT | AMOUNT | 예) 재고량 STOCK\_AMT |
| \_SUM | SUM | 예) 분기합계 QTR\_SUM, 년도합계 YEAR\_SUM |

#### *3) 순서규칙*

█ 기본적으로 관계형 모델에서 열(Column)의 순서는 의미가 없음. 그러나, 물리적인 형태로 생성되어 관리될 때에는 보다 효율적인 저장공간의 관리를 위해 다음 순서에 따라 우선순위를 결정함

█ Primary Key가 우선함

█ Primary Key내에서는 Index 의미에 따라 순서를 결정함

█ Not Null Columns이 우선함

█ Not Null Columns 내에서는 Foreign Key, Attributes 순서로 함

█ Null Columns 내에서는 다음의 규칙에 따라 순서를 결정함

█ Fixed Length Columns이 우선함(Date,Number,Char순)

█ Smaller Length Column이 우선함

### 라. Index Name

#### *1) 규칙*

█ 해당하는 테이블명 뒤에 ‘\_IDX’를 붙여 index임을 명확히 함

█ 대문자를 사용함

█ 일련번호는 01 ~ 99까지 사용할 수 있음

█ MSSQL의 경우 클러스터드 인덱스와 넌 클러스터드 인덱스를 구분하여 작성함. 클러스터드 인덱스 \_IDXC를 사용하며, 넌 클러스터드 인덱스는 일반 인덱스 명 룰을 따름.

█ 테이블에 인덱스가 하나만 존재할 경우 일련번호를 사용하지 않아도 됨.

#### *2) 표기 방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + IDX{<일련번호>}

예) Table명 ‘I01\_MASTER’의 Index : I01\_MASTER\_IDX01

### 마. Primary Key Name

#### *1) 규칙*

█ 영문 대문자로 작성함

█ 해당하는 테이블명의 맨 뒤에 ‘\_PK’라는 구분을 사용함

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + PK

예) Table 명 ‘AC\_USERS’의 Primary Key : AC\_USERS\_PK

### 바. Foreign Key Name

#### *1) 규칙*

█ 영문 대문자로 작성함

█ 해당하는 테이블명의 맨 뒤에 ‘\_FK’라는 구분을 사용함

█ 일반적으로 테이블명과 컬럼명까지 사용하나, OBJECT의 명칭이 길어져서 테이블명을 기준으로 작성함.

█ 일련번호: 1 ~ 9

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + FK{<일련번호>}

예) Table 명 ‘I01\_MASTER’의 Foreign Key : I01\_MASTER\_FK1

### 사. Stored Procedure Name

#### *1) 규칙*

█ 길이는 큰 제한이 없으나 오라클의 OBJECT NAME 길이 제한은 있음.

█ 해당하는 테이블명의 맨 뒤에 ‘\_SP’라는 구분을 사용함

█ 기능명은 복수개 사용이 가능하면 3개의 단어를 넘지 않도록 함

█ 기능을 나타내는 명칭이 하나일 경우 일련번호를 생략해도 됨.

█ 단어간에는 ‘\_’로 구분함

█ 업무 룰에 해당되지 않는, 혹은 특정 테이블에 해당되지 않는 DBMS 전반적인 프로시저의 경우, 시스템 프로시저로 작성하는 경우에는 시스템구분 과 테이블명을 생략하고 간단히 작성할 수 있다. 예) 스키마 스크립트 GENERATION – GENERATE\_SP

█ 오라클의 경우 패키지 내부의 프로시저의 경우 패키지 명칭에 시스템구분을 사용하므로, 프로시저나 함수명에 시스템구분 코드를 넣지 않는다. 또한 기능에 따른 일련번호를 사용하지 않고 OOP의 기능인 Method Overloading 의 기능을 사용하여 작성한다. 또한 명칭은 Camel 표기법을 사용하여 작성한다. 예) 사용자를 가져오는 경우 getUsers()

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + <기능명>{<일련번호>} + ‘\_’ + SP

예) ‘I01\_MASTER’ 테이블에서 데이타 입력에 대한 Procedure

: I01\_MASTER\_INS01\_SP

|  |  |  |
| --- | --- | --- |
| 기능명 | 명칭 | 설명 |
| INS | INSERT | 단일 테이블의 단순 INSERT 작업인 경우, 사용자 테이블에 데이터 입력 프로시저의 경우 업무룰이 복잡하여 여러 테이블에 걸쳐 삽입 작업이 된다면(서버측 트랜잭션이 구현된다면) INS를 사용하지 않고, REG를 사용한다. |
| UDT | UPDATE | 단일 테이블의 단순 UPDATE 작업의 경우 |
| DEL | DELETE | 단일 테이블의 단순 삭제인 경우 |
| LST | LIST | SELECT문을 사용하여 조회하는 경우 |
| REG | REGISTER | 등록작업 – 트랜잭션을 사용하여 여러 테이블에 입력 작업이 이루어질 때 |
| MOD | MODIFY | 수정작업 – 트랜잭션을 사용하여 여러 테이블에 수정 작업이 이루어질 때 |
| REM | REMOVE | 삭제작업 – 트랜잭션을 사용하여 여러 테이블에 삭제 작업이 이루어 질 때 |

### 아. Function Name

#### *1) 규칙*

█ 길이는 제한이 없으며 영문 대문자를 사용함

█ 해당하는 테이블명의 맨 뒤에 ‘\_FC’라는 구분을 사용함을 원칙으로 하나, 함수명이 길어서 사용상 불편할 경우, 특정 시스템에 국한하지 않고, 항상 사용하는 라이브러리 같은 함수의 경우 구분가능한 Short Name을 사용해도 무방하다.

█ 단어 간에는 ‘\_’로 구분함

█ 시스템 함수로 작성한 경우에는 접미사를 사용하지 않고, 간략한 함수이름을 사용한다. 예) INSTR, LEASTR(@x bigint, @y bigint) 등

█ 오라클의 경우 패키지 내부의 함수의 경우에는 프로시저의 해당 규칙에 따른다. 즉 시스템구분 코드와 접미사를 사용하지 않고, Camel 표기법으로 간략하게 작성한다.

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <기능명> + ‘\_’ + FC

예) ‘I01\_MASTER’ 테이블에서 주소명를 가져오기 위한 Function

: I01\_GET\_ADDRESSNAME\_FC(p\_AddressCode IN Char) 내지는

: getAddressName(p\_AddressCode IN Char)

### 자. Table Trigger Name

#### *1) 규칙*

█ 영문 대문자로 작성함

█ 일련번호는 01 ~ 99까지 사용 가능함

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + <Timing><Trigger Event><일련번호> + ‘\_’ + TG

® Timing : B(Before), A(After)

® Trigger Event : I(Insert), D(Delete), U(Update)

예) ‘I01\_MASTER’ 테이블에서 데이타 입력 후에 실행되는 Trigger

: I01\_MASTER\_AU01\_TG

### 차. View Name

#### *1) 규칙*

█ 길이는 제한이 없으며, 영문 대문자로 작성함

█ 해당하는 테이블명의 맨 뒤에 ‘\_VW’라는 구분을 사용함

█ 일련번호는 01 ~ 99까지 사용할 수 있음

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명><일련번호> + ‘\_’ + VW

예) AC\_ADMINL\_USER\_VW

### 카. Sequence Name <오라클의 경우에만 해당>

#### *1) 규칙*

█ 길이는 제한이 없으며 영문 대문자를 사용함

█ 해당하는 테이블명의 맨 뒤에 ‘\_SQ’라는 구분을 사용함

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 테이블명> + ‘\_’ + SQ

예) ‘I01\_MASTER’ 테이블의 Sequence : I01\_MASTER\_SQ

### 타. Package Name<오라클의 경우에만 해당>

#### *1) 규칙*

█ 길이는 제한이 없으며 영문 대문자를 사용함

█ 해당하는 테이블명의 맨 뒤에 ‘\_PKG’라는 구분을 사용함

#### *2) 표기방식*

<시스템 구분> + ‘\_’ + <의미있는 패키지명> + ‘\_’ + PKG

예) 검색엔진에서 사용하는 자원에 관련된 패키지 : SCH\_\_PKG

### 파. Check 제약조건

#### *1) 규칙*

█ 길이는 제한이 없으며 영문 대문자를 사용함

█ 기존의 명칭 룰에 해당하는 접미사를 사용하지 않고, 예외적으로 접두어 CK\_를 사용한다. 일반적으로 CHECK와 DEFAULT 제약조건은 특정 테이블에 한정시켜서 작성하기 보다는 시스템 전반에 걸쳐서 사용이 가능하므로 예외규정을 둔다.

#### *2) 표기방식*

CK + ‘\_’ + <의미있는 CHECK명>

예) 이메일 체크 : CK\_EMAIL

예) 성별 체크 : CK\_SEX

### 하. Default 제약조건

#### *1) 규칙*

█ 길이는 제한이 없으며 영문 대문자를 사용함

█ 기존의 명칭룰에 해당하는 접미사를 사용하지 않고, 예외적으로 접두어 DF\_를 사용한다. 일반적으로 CHECK와 DEFAULT 제약조건은 특정 테이블에 한정시켜서 작성하기 보다는 시스템 전반에 걸쳐서 사용이 가능하므로 예외규정을 둔다

#### *2) 표기방식*

DF + ‘\_’ + <의미있는 DEFAULT명>

예) Null String Default – DF\_NULLSTR  
예) 0(Zero) Default – DF\_ZERO