편의상 컬럼명으로 만 사용 했지만 ‘문자열’ ‘문자’ ‘숫자’ 로 넣어도 가능

Date 형식 확인

select \* from SYS.NLS\_SESSION\_PARAMETERS where parameter = 'NLS\_DATE\_FORMAT';

Date 형식 변경

alter session set nls\_date\_format='YYYY-MM-DD HH24:MI:SS'; // 원하는 형태로

Select distinct [컬럼명] from [테이블명] // 중복값 제거

is null // null인 값 || is not null // null이 아닌 값

nvl(컬럼명, 바꿀값) // 컬럼명에 값이 null이면 바꿀값으로 치환 한다 ex) nvl(null, 0)

order by 컬럼명 [ asc || desc ] // asc 올림차순(작은 값부터) , desc 내림차순(큰 값부터)

initcap(컬럼명) // 단어의 첫글자 대문자로 바꿈

lower(컬럼명) // 모든 문자를 소문자로 변경

upper(컬럼명) // 모든 문자를 대문자로 변경

length(컬럼명) // 문자의 개수 (영어든 한글이든 숫자든 한 글자 당 1 )

concat(컬럼명, 컬럼명) // 문자 결합 [ 컬럼명 || 컬러명 ] 도 가능

substr(컬럼명, 몇번째 자리부터, 몇글자를 추출) // 부분 문자열 추출 ex) substr(‘ABC’, 2, 1) -> ‘B’

lpad(컬럼명, 문자의길이, 넣을 문자) // ex) lpad(‘AB’, 5, ‘\*’) -> ‘\*\*\*AB’

// 컬럼명이 정해진 문자의 길이보다 적을 경우 왼쪽방향으로 넣을 문자로 채우겠다.

rpad(컬럼명, 문자의길이, 넣을 문자) // ex) rpad(‘AB’, 5, ‘\*’ -> ‘AB\*\*\*’

rtrim(컬럼명, 지울문자) // 컬럼명 오른쪽에 지울문자가 있다면 지워라

ltrim(컬럼명, 지울문자) // 컬럼명 왼쪽에 지울문자가 있다면 지워라

ex) rtrim(‘MILLEEEEER’, ‘ER’) -> ‘MILL’ // ltrim(‘MILLLLLLLER’) -> ‘ER’

replace(컬럼명, 찾을내용, 치환할내용) // 컬럼명에 찾을내용 이 있다면 치환할 내용으로 변경

ex) replace(‘ABCDE’ , ‘D’, ‘와우’) -> ‘AB와우DE’

round(컬럼명, 자리) // 숫자를 (0=정수자리) 자리 까지 반올림 하겠다.

trunc(컬럼명, 자리) // 숫자를 (0=정수자리) 까지 만 보여주겠다.

mod(컬럼명1, 컬럼명2) // (컬럼명1 / 컬럼명2) 한 나머지를 보여주겠다.

sysdate // 현재 날자 시간

months\_between(날짜1, 날짜2) // 날짜1과 날짜2 의 일수 차이 계산

to\_number(숫자형문자) // 문자를 숫자로 변경 하지만 안써도 자동형변환이 됨

to\_char(숫자) // 숫자를 문자로 변경해줌 ex) 123 -> ‘123’

다른 활용법

to\_char(숫자, 채울형식)

ex) to\_char(12345, ‘9999999999’) -> ‘ 12345’ 공백 5개 9는 공백으로

ex) to\_char(12345, ‘0000000000’) -> ‘0000012345’

ex) to\_char(12345, ’9,999,999,999’) -> ‘ 12,345’

ex) to\_char(12345, ‘$9,999,999,999’) -> ‘ $12,345’

to\_date(날짜형문자 , 날짜) // 날짜형문자를 날짜데이터로 하지만 자동형변환 가능

decode(컬럼명, 지정한값, 출력값, 지정한값2, 출력값2, else값)

// 컬럼명이 지정한값n과 같다면 출력값n으로 출력

count() 개수, sum() 합, avg() 평균, max() 최대값, min() 최소값

컬럼명 in(값1, 값2, 값3) // 컬럼명 = 값1 or 컬럼명 = 값2 or 컬럼명 = 값3

컬럼명 not in(값1, 값2, 값3) // 컬럼명 != 값1 and 컬럼명 != 값2 and 컬럼명 != 값3

in(null) -> null 무시 // not in(null) -> null로 출력

1=1 항상 참 // 1=2 항상 거짓

제약 정보 확인하기

select \* from user\_constraints where table\_name = '테이블명(대문자로)';

테이블추가

create table 테이블명 ( 컬럼명 타입 );

create table (view) 테이블명 as select 컬럼명 from ~~~ // 테이블 복사 생성(view는 가상테이블)

수정

alter table 테이블명 add constraint ~~~ // 기본키 외래키 유니크 등 제약 조건 걸기

alter table 테이블명 add 컬럼명 타입 // 타입변경

alter table 테이블명 drop column 컬럼명 // 컬럼 삭제

alter table 테이블명 rename column “컬럼명” to 새컬럼명 // 컬럼명 변경

alter table 테이블명 modifiy(컬럼명 not null) // 컬럼에 not null 추가

제약조건 방법

->이쪽은 외래키일때만

Constraint OO\_테이블명\_컬럼명 제약조건(컬럼명) references 참조할테이블명(참조할컬럼명)

OO 은 제약조건에 맞는 pk(primary key), fk(foreign key), uk(unique) 를 적는다

(컬럼명)은 (컴퓨터가) 알 수 있을 때는 적지않아도 된다

Create 시

default 변경할값 // 값을 null로 입력 됬을 시 변경할값으로 변경

generated always as (가상컬럼) virtual

가상컬럼에 더하기 빼기 나누기 case문 을 넣어서 스스로 입력되게 할 수 있다.

DB의 case문 ex)

GENERATED ALWAYS as (

CASE WHEN substr(p\_date,5,2) in ('01','02','03') THEN 1

WHEN substr(p\_date,5,2) in ('04','05','06') THEN 2

WHEN substr(p\_date,5,2) in ('07','08','09') THEN 3

ELSE 4

END

) VIRTUAL