2019-10-28

**연습용 계정생성**

Local\_SYS 다른사용자 우클릭 - 사용자 생성

사용자이름 : TEST

비밀번호 : 1234

기본 테이블스페이스 : USERS

임시 테이블스페이스 : TEMP

부여된 롤 : 모든권한 /

CONNECT는 특히 필수 외부에서 접속가능, 외부서비스 시 필요

시스템권한 : 모든권한

적용

DCL - 제어어, 권한, 공간 등의 시스템 관리

DDL - 정의어, 구조(테이블, 계정 등등) 의 생성, 수정, 삭제, 관계관리

 CREATE -> 생성

 ALTER -> 변경

 DROP -> 제거

**게시판 설계**

Number - 숫자

Varchar2 - 문자열

Date - 날짜

Varchar2 -> 4000byte

공백 -> 1byte

숫자 -> 1byte

영어소문자 -> 1byte

영어대문자 ->2byte

한글 -> 4byte

**기본적인 게시판 테이블의 데이터형식**

테이블정보

컬럼명 / 뜻 / 데이터형식 / 데이터크기

NO / 글번호 / N / PK

TITLE / 제목 / V / 100

CON / 내용 / V / 4000

DT / 작성일 D SYSDATE

WRITER / 작성자 / V / 100

HIT / 조회수 N 0

**데이터 추가**

INSERT INTO 테이블명(컬럼명, … )

VALUES (값, 값, … )

;

해당부분이 없는경우 테이블의 컬럼 순서대로 모든 컬럼을 대상으로 진행됨

지정된 컬럼 개수만큼 타입을 동일하게

**TRANSACTION -> 작업관리**

 COMMIT : 현재까지의 작업을 적용한다.

 ROLLBACK : 마지막 COMMIT시점으로 되돌린다.

Timeline

COMMIT

|INSERT

|

|INSERT

|

|INSERT

|

|ROLLBACK ( 처음 COMMIT으로 돌아감 ) / COMMIT

|

TRANSACTION

INSERT

INSERT

INSERT

**업무할 때**

**토드를 쓸 때는 AUTO COMMIT이 TRUE로 되어있기 때문에**

**AUTO COMMIT 끄고 할것**

**데이터 수정**

UPDATE 테이블명 SET 컬럼명 = 값, …

★WHERE 조건

;

값에 컬럼명이 오는경우 해당 컬럼의 값이 위치하게 됨

NO를 2와 3와으로 바꿔 데이터 삽입 후 커밋

INSERT INTO TEST(NO, TITLE, CON, WRITER)

VALUES(3, 'TEST', 'TEST', 'TESTER')

;

UPDATE TEST SET TITLE = 'AAA'

;

다음과 같이 WHERE을 사용하지 않으면, 원하는 데이터의 수정이 불가

**데이터 삭제 -> ROW삭제**

DELETE FROM 테이블명

WHERE 조건

;

WHERE조건을 명시하지않으면, 데이터가 몽땅~ 삭제된다.

UPDATE TEST SET TITLE = TITLE || NO

WHERE NO = 1

;

DELETE FROM TEST

;

**오라클 시퀀스**

SEQUENCE : 증가되는 숫자를 관리하기 위해 만들어진 기능

NEXTVAL : 다음값

CURRVAL : 현재값

TRANSACTION에 영향을 받지 않음.

시퀀스 우클릭 - 새 시퀀스

이름 : TEST\_SEQ

다음으로 시작 : 1

증분 : 1

최소값 : 1

최대값 : 9 28개

캐시 : 캐시없음

주기 : 지정x

정렬 : 지정x

시퀀스 초기화

시퀀스 우클릭 후 편집 - 다음부터 시작 1로 변경

예 누르고 확인 누르면 다시 1부터 시작가능

INSERT INTO TEST(NO, TITLE, CON, WRITER)

VALUES(TEST\_SEQ.NEXTVAL, 'TEST' || TEST\_SEQ.CURRVAL, 'TEST', 'TESTER')

;

SELECT TO\_CHAR(SYSDATE, 'YYYY') || LPAD('1', 4, '0') AS NO

FROM DUAL

;

SELECT T.NO, T.TITLE, T.CON, T.DT, T.WRITER, T.HIT

FROM ( SELECT NO, TITLE, CON, WRITER, HIT,

CASE WHEN FLOOR(SYSDATE - DT) > 0

THEN FLOOR(SYSDATE - DT) || '일전'

WHEN FLOOR((SYSDATE - DT) \* 24) > 0

THEN FLOOR((SYSDATE - DT) \* 24) || '시간전'

WHEN FLOOR((SYSDATE - DT) \* 24 \* 60) > 0

THEN FLOOR((SYSDATE - DT) \* 24 \* 60)|| '분전'

ELSE '방금전'

END AS DT,

ROW\_NUMBER() OVER(ORDER BY NO DESC) AS RNUM

FROM TEST

WHERE TITLE LIKE '%' || '1' || '%'

) T

WHERE T.RNUM BETWEEN 1 AND 10

;

|  |  |  |
| --- | --- | --- |
| 함수명 | 사용법 |  |
| LPAD(값, 숫자, 값2) | 값1의 글자수가 숫자개수만큼 될때까지 왼쪽에 값2를 추가한다.  EX) LPAD('1', 4' '0') => 0001  1이 4글자가 될때까지 왼쪽에 0으로 채운다. |  |
| RPAD(값, 숫자, 값2) | '' |  |
|  |  |  |

**게시판의 4대요소**

**C**reate -> insert

**R**ead -> select

**U**pdate -> update

**D**elete -> delete

선생님이 내주신 HR문제

-- 부서이동이 1회이상 있었던 사원들의 정보를 출력하시오.

-- 출력컬럼 : 이름, 현재부서명, 과거부서명

-- 추론과정

(SELECT D.DEPARTMENT\_NAME AS P, D.DEPARTMENT\_ID AS A

FROM DEPARTMENTS D INNER JOIN JOB\_HISTORY H

ON D.DEPARTMENT\_ID = H.DEPARTMENT\_ID)

;

SELECT E.LAST\_NAME, D.DEPARTMENT\_NAME AS 현재부서, E.DEPARTMENT\_ID, TO\_CHAR(START\_DATE, 'YYYY-MM-DD')

,RANK() OVER(PARTITION BY LAST\_NAME ORDER BY TO\_CHAR(START\_DATE, 'YYYY-MM-DD')) A

FROM JOB\_HISTORY H

INNER JOIN EMPLOYEES E

ON H.EMPLOYEE\_ID = E.EMPLOYEE\_ID

AND H.DEPARTMENT\_ID != E.DEPARTMENT\_ID INNER JOIN DEPARTMENTS D

ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

;

SELECT E.FIRST\_NAME AS FIRST\_NAME, D1.DEPARTMENT\_NAME, D2.DEPARTMENT\_NAME,

RANK() OVER(PARTITION BY LAST\_NAME ORDER BY TO\_CHAR(START\_DATE, 'YYYY-MM-DD')) A

FROM JOB\_HISTORY H

INNER JOIN EMPLOYEES E

ON H.EMPLOYEE\_ID = E.EMPLOYEE\_ID

AND H.DEPARTMENT\_ID != E.DEPARTMENT\_ID

INNER JOIN DEPARTMENTS D1

ON H.DEPARTMENT\_ID = D1.DEPARTMENT\_ID

INNER JOIN DEPARTMENTS D2

ON E.DEPARTMENT\_ID = D2.DEPARTMENT\_ID

;

-- 정답

SELECT A.이름, A.현재부서, A.이전부서

FROM ( SELECT E.FIRST\_NAME AS 이름, D1.DEPARTMENT\_NAME AS 이전부서, D2.DEPARTMENT\_NAME AS 현재부서,

RANK() OVER(PARTITION BY LAST\_NAME ORDER BY TO\_CHAR(START\_DATE, 'YYYY-MM-DD')) A

FROM JOB\_HISTORY H

INNER JOIN EMPLOYEES E

ON H.EMPLOYEE\_ID = E.EMPLOYEE\_ID

AND H.DEPARTMENT\_ID != E.DEPARTMENT\_ID

INNER JOIN DEPARTMENTS D1

ON H.DEPARTMENT\_ID = D1.DEPARTMENT\_ID

INNER JOIN DEPARTMENTS D2

ON E.DEPARTMENT\_ID = D2.DEPARTMENT\_ID ) A

WHERE A.A = 1

;