

제 1 절 관계형 데이터베이스 개요

1. 데이터베이스
2. SQL
3. TABLE
4. ERD

제 2 절 DDL

1. 데이터 유형

- CHAR(s) : 고정 길이 문자열
- VARCHAR(s) : 가변 길이 문자열
- NUMBER(n,m) : 정수부분 n 자리 소수부분 m자리인 숫자(Oracle)
- DATE

2. CREATE TABLE

- Constraint
 - primary key : 대표키, 1개만가능, unique key + not null
 - unique key : TB에 정의된 고유 key, NULL 가능함
 - not null
 - check : TRUE, FALSE로 입력 범위값 제약함.
 - foreign key : TB간의 관계를 정의하기 위해 PK를 다른 TB의 FK 로 복사
 - referential integrity 지정가능 : FK 로 참조중인 자식이 있을 경우 함부로 삭제하면 referential integrity 가 깨짐. 자식부터 삭제해야함.
 - CASCADE, RESTRICT, SET NULL 등.

```
CREATE TABLE TEAM (  
    TEAM_ID CHAR(3) NOT NULL,  
    REGION_NAME VARCHAR(8) NOT NULL,  
    ...  
    STADIUM_ID CHAR(3) NOT NULL,  
    ...  
    CONSTRAINT TEAM_PK PRIMARY KEY (TEAM_ID);  
    CONSTRAINT TEAM_FK FOREIGN KEY (STADIUM_ID) REFERENCES  
    STADIUM(STADIUM_ID)  
);
```

- CTAS(CREATE TABLE AS Select ...) 로 TB 생성시 copy 한 TB는 NOT NULL을 제외한 CONSTRAINT 다 풀린다.

3. ALTER TABLE

- COLUMN ADD, DROP, MODIFY, RENAME 에는 ALTER 사용
 - ALTER TABLE (TB_name) DROP COLUMN (col_name)
- DROP CONSTRAINT 에도 ALTER 사용
 - ALTER TABLE (TB_name) DROP CONSTRAINT (constraint_name)

4. RENAME TABLE

- RENAME (target_name) to (new_name)

5. DROP TABLE

- DROP TABLE (tb_name) [CASCADE CONSTARINT]

6. TRUNCATE TABLE

- TRUNCATE TABLE (tb_name)

// 참고 DROP, TRUNCATE, DELETE FROM (tb_name) 의 차이

- DROP : TB 강 버림
- TRUNCATE : TB 초기화 (빠다구만 남김. AUTO COMMIT)
- DELETE : TB 데이터 삭제, RB 가능, 당연히 느림

제 3 절 DML

1. INSERT

- INSERT INTO (tb_name) (COLUMN_LIST) VALUES (VALUE_LIST);
- INSERT INTO (tb_name) VALUES (VALUE_LIST); // 순서대로 넣어야함.

2. UPDATE

- UPDATE (tb_name) SET (target_col) = (target_value); // 해당 col의 내용 다 삭제.
- 특정 record 집으려면 where 써야지

3. DELETE

- DELETE FROM (tb_name);

4. SELECT

- SELECT [DISTINCT] (cols) FROM (tb_name); // DISTINCT 는 중복생략

5. 산술연산자와 합성연산자

- NUMBER 나 DATE 같은 자료형일 경우 SELECT 안에 연산자 사용 가능
 - SELECT PLAYER_NAME, HEIGHT - WEIDHT "키-몸무게" FROM PLAYER;
- 합성연산자 (CONCATENATION)
 - SELECT PLAYER_NMAE || '선수, ' || HEIGHT || ' cm, ' FROM PLAYER;

제 4 절 TCL

1. 트랜잭션 개요

- Transaction 은 DB 의 논리적 연산 단위
- ACID 속성
 - Atomicity : all or nothing
 - Consistency : Transaction 실행 이전 이후 같아야 함. (문제 없으면 실행후에도 문제 없어야함)
 - Integrity : Transaction 중 다른 Transaction 에 영향 받지 않아야 함.(LOCKED)
 - Durability : Transaction 성공 후 영구저장

2. COMMIT

- Transaction 완료
- DDL 은 AUTOCOMMIT 됨

3. ROLLBACK

- COMMIT 이전 변경사항 취소

4. SAVEPOINT

- SAVEPOINT (sv_name);
- ROLLBACK TO (sv_name);
- SAVEPOINT 까지만 저장/ 롤백

제 5 절 WHERE

1. WHERE 조건절 개요

23456. 연산자의 종류

- 비교 연산자 : =, >, <, <=, >=
- SQL 연산자
 - BETWEEN a AND b
 - IN (list)
 - LIKE : '_' (암거나 1개) / '%' (쪽다)
 - IS NULL // NULL 값 찾는방법 <-> IS NOT NULL
- 논리 연산자: AND, OR, NOT
- 부정 비교 연산자
 - !=, ^=, <>(ASNI/ISO표준), col 앞에 not 붙이기 등

7. ROWNUM, TOP 사용

- ROWNUM : ORACLE
- TOP : SQL Server
 - TIES 있으면 같은 값이 있으면 추가적으로 행이 더 나오게 함.

제 6 절 함수

12345. 내장함수(문자형, 숫자형, 날짜형, 변환형)

- 문자형
 - CONCAT (oracle || 나, sql server + 와 같음)
 - SUBSTR(문자열, m,[n]) : m위치에서 n 개의 문자 되돌려줌
 - LTRIM('xxxfrwrex', 'x') : 왼쪽 x 다제거
- 숫자형
 - ABS : 절대값
 - MOD(a,b) : a % b
 - CEIL/CEILING : 숫자보다 크거나 같은 최소 정수 (양수 소숫점 올림)
 - FLOOR : 숫자보다 같거나 작은 최대 정수(양수 소숫점 버림)
 - ROUND(num, [m]) : m자리수까지 나오게 반올림
 - TRUNC(num, [m]) : 숫자 m자리수부터 그냥 잘라서버림 기본은 그냥 소숫점 다버림
- 날짜형
 - SYSDATE / GETDATE() (SQL server)
- 변환형
 - TO_CHAR, TO_NUMBER, TO_DATE (SQL Server : CAST, CONVERT): 명시적변환 (Explicit)
 - Implicit 하게 변환 해주기도 함. 숫자랑 문자같은거

6. CASE 표현

- CASE WHEN (Condition) THEN (...)

7. NULL 관련 함수

- WHERE 에도, IS NULL / IS NOT NULL 있음 유의.(NULL 사용한 비교시 사용)
- 단일행 NULL 함수 (산술연산 등 하다가 null 만날경우 어떻게할지..)
 - NVL(exep1, exep2) / ISNULL(exep1, exep2) : exep1이 NULL 이면 exep2 로 출력함
 - NULLIF(exep1, exep2) : 1,2 가 같으면 NULL 같이 않으면 1을 리턴 (2랑 같아지면 재깎때 사용)
 - COALESCE(exep1, exep2, ...) : 임의의 개수 표현식에서 NULL 이 아닌 최초의 표현식 나타냄

제 7 절 GROUP BY, HAVING 절

1. Aggregate Function

- 여러 행들의 그룹들이 모여서 그룹당 단 하나의 결과를 돌려주는 함수이다.
- MAX, MIN, COUNT, AVG, SUM
 - COUNT(*) 이면 NULL 도 포함. 그냥 표현식이면 NULL 제외

2. GROUP BY

- 소그룹별 통계를 얻는데 사용

3. HAVING

- 그룹을 나타내는 결과 집합에 적용됨. aggregate function 사용 가능함.

4. CASE 표현을 활용한 월별 데이터 집계

5. 집계 함수와 NULL 처리

- 집계함수는 알아서 null 재끼므로 따로 0 으로 바꿔주고 이상한 짓 안해도 된다.

제 8 절 ORDER BY 절

1. ORDER BY 정렬

- SQL문장으로 조회된 데이터들의 다양한 목적에 맞게 특정 칼럼을 기준으로 정렬
- 가장 마지막 단계에서 수행!!!!
- ORDER BY A, B, C 가능함. 이때 ABC 각각 오름 내림 차순 정할 수 있음
 - ORDER BY A DESC, B, C DESC; // 이런식

2. SELECT 문장 실행순서

- FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY
- SELECT 절에 없는 column 도 order by 에서 사용 가능하다.
- SUBQuery 안에 ORDER BY 까지 쓸 수 있다.

3. TOP N Query

- ROWNUM / TOP()
 - TOP 은 TIES 와 함께 사용 시 같은 값이 나오면 그만큼 하나 더 출력함.

제 9 절 JOIN

1. JOIN 개요

2. EQUI JOIN

3. NON EQUI JOIN

4. 3개 이상 TB JOIN

알아서 잘 하자