1. DCL 고르기

DCL은 Grant, Revoke, TCL은 Commit, Rollback

2. 테이블의 구조를 생성, 변경, 삭제 -> DDL (Create, Alter, Rename, Drop)

3. TCL 개념 말하고 용어 무엇인지

4. 데이터 언어와 SQL 명령어에 대한 설명

- 비절차적 데이터 조작어(DML)은 사용자가 무슨 데이터를 원하는지만을 명세, 어떻게 접근해야 하는지 -> 절차적 데이터 조작어의 영역. (PL/SQL, SQL Server)

5. DB 시스템 언어의 종류와 명령어 연결하기

6. Oracle

- ALTER TABLE PRODUCT ADD CONSTRAINT PRODUCT\_PK PRIMARY KET(PROD\_ID)가 올바른 기본키 변경구문

- PK 지정시 괄호 안에 ADD CONSTRAINT PRIMARY EKY(PROD\_ID); 불가능

- PK 지정을 하나의 괄호 안에서 지정 가능

CREATE RABLE PRODUCT

( PROD\_ID VARCHAR(10) NOT NULL,

PROD\_NM VARCHAR2(100) NOT NULL,

REG\_DT DATE NOT NULL,

REGR\_NO NUMBER(10),

CONSTRAINT PRODUCT\_PK PRIMARY KEY (PROD\_ID);

7. 컬럼 정보 변경

- ALTER TABLE 기관분류 ALTER COLUMN 분류명 VARCHAR(30) NOT NULL

ALTER TABLE 기관분류 ALTER COLUMN 등록일자 DATE NOT NULL;

- 분류명을 수정할 때 NOT NULL 조건 꼭 넣어야 함, 아니면 NULL로 변경됨

- SQL Server는 괄호를 사용하지 않음 (여러 개여 컬럼 동시 수정하는 구문 지원 X)

8. NULL의 개념

- 모르는 값, 값의 부재, NULL과의 모든 비교는 알 수 없음을 반환

- 공백문자나 숫자 0과 다름!

9. 삭제 옵션

- CASCADE는 해당 컬럼 아니더라도 함께 다 삭제, SET NULL은 없어진 컬럼만 NULL로 처리

10.

- UNIQUE : 중복 불가능, Null 가능

- PK = UNIQUE + NOT NULL

11. 테이블명, 컬럼명 명명 규칙

- 문자로 시작, 알파벳/숫자/\_,$,# 가능

12. 컬럼을 인덱스로 지정

- 디폴트 값 지정 : DEPT\_CODE VARCHAR2(4) DEFAYKT ‘0000’ NOT NULL

- 인덱스 : CREATE INDEX IDX\_EMP\_01 ON EMP (JOIN\_DATE);

- 테이블 생성시 PRIMARY KEY 지정했다면, ALTER TABLE EMP ADD CONSTRAINT EMP\_PK PRIMARY KEY (EMP\_NO);로 또 지정하려고 하면 오류 발생

13. COUNT(\*) == COUNT(PK)

14. 외래키 특징

- 테이블 생성 시 설정할 수 있음

- NULL 값을 가질 수 있음

- 한 테이블에 여러 개 존재할 수 있음

- 외래키 값은 참조 무결성 제약을 받을 수 있음

15. 테이블의 제약 조건

- CHECK : DB에서 데이터의 무결성을 유지하기 위하여 테이블의 특정 컬럼에 설정하는 제약

16. 컬럼 삭제

- DROP 테이블명 DROP COLUMN 컬럼명

17. 삭제 옵션

- CASCADE : 부모 삭제시 자식 같이 삭제됨

- Set Null : 자식의 해당 필드 Null로

- Set Default : 자식의 해당 필드를 디폴트값으로 설정

- Restrict : 자식 테이블에 PK값이 없는 경우만 부모 삭제 허용

- No action : 참조무결성을 위반하는 삭제/수정 액션을 취하지 않음

18. 테이블 이름 변경

RENAME 현재테이블명 TO 새로운테이블명

19. 삽입 액션

- Automatic : M테이블에 PK가 없는 경우 M의 PK를 생성 후 C 입력

- Set Null : C의 외부키를 Null값으로 처리

- Set Default : C의 외부키를 지정된 기본값으로 입력

- Dependent : M 테이블에 PK가 존재할 때만 C 입력 허용

20. 테이블행 삽입 가능 여부?

- 삽입 컬럼을 명시하지 않은 경우, NULL 가능한 컬럼이라도 삽입할 값을 명시해줘야 함

21.

- UPDATE BOARD SET BOARD\_ID = 200 WHERE BOARD\_ID = ‘100’; 가능

- Question) 문자열 형태 넣어야 하는 컬럼에 SET으로 숫자 형태 써도 되나? 묵시적 형변환으로 오류가 생기진 않음

22.

- FK 있는 쪽에 삼지창 존재

- 고객ID를 삭제하면 SET NULL로 해놔서 NULL로 업데이터하려고 DBMS에서 시도하지만, 주문 테이블 고객 ID컬럼의 NOT NULL 제약조건에 의해 실패함.

23. 테이블 삭제 명령어

- TRUNCATE TABLE, DROP TABLE : 로그를 남기지 않음

- DELETE \* FROM STADIUM : 문법 X -> DELETE FROM STADIUM

24. 중복 제거

- SELECT DISTINCT 거주지, 근무지

- SELECT 거주지, 근무지 / GROUP BY 거주지, 근무지;

25. TRUNCATE

- TRUNCATE : 모든 데이터 삭제, 디스크 사용량 초기화, 테이블 스키마 정의 유지

- DELETE : 모든 데이터 삭제, 디스크 사용량 초기화 X

- DROP : 모든 데이터 삭제, 디스크 사용량 초기화, 스키마 정의 삭제

- DELETE TABLE FROM 테이블명 -> 존재 X

26.

|  |  |  |
| --- | --- | --- |
| DROP | TRUNCATE | DELETE |
| DDL | DDL | DML |
| 롤백 불가능 | 롤백 불가능 | 커밋 이전 롤백 가능 |
| Auto 커밋 | Auto 커밋 | 사용자 커밋 |
| 테이블이 사용한 모드 스토리지 release | 최초 테이블 생성시 할당된 스토리지 남기고 release | 데이터를 모두 삭제해도 사용했던 스토리지는 releae되지 않음 |
| 테이블 정의 완전히 삭제 | 최초 생성된 초기 상태 남김 | 데이터만 삭제 |

- TRUNCATE는 UNDO를 위한 데이터를 생성하지 않기 때문에 동일 데이터량 삭제 시 DELETE보다 빠름

27. 트랜잭션의 특성

- 원자성

: 트랜잭션에서 정의된 연산들은 모두 성공적으로 실행되든지 아니면 전혀 실행되지 않은 상태로 남아 있어야 한다.

- 일관성

: 트랜잭션이 실행되기 전의 데이터베이스 내용이 잘못 되어 있지 않다면 트랜잭션 이후에도 내용에 잘못이 있으면 안된다.

- 고립성

: 트랜잭션이 실행되는 도중 다른 트랜잭션의 영향을 받아 잘못된 결과를 만들면 안된다.

- 지속성

: 트랜잭션이 성공적으로 수행되면 그 트랜잭션이 갱신한 데이터베이스의 내용은 영구적으로 저장된다.

28. 트랜잭션 시 발생 문제

- Dirty read : 다른 트랜잭션에 의해 수정되었지만 아직 커밋되지 않은 데이터를 읽는 것

- Non-repeatable read : 한 트랜잭션 내에서 같은 쿼리를 두번 수행했는데, 그 사이에 다른 트랜잭션이 값을 수정/삭제하여 두 쿼리 결과가 다르게 나타나는 현상

- Phantom read : 한 트랜잭션 내에서 같은 쿼리를 두 번 수행, 첫번째 쿼리에서 없던 유령 레코드가 두번째 쿼리에서 나타나는 현상

29. O와 SS에서의 롤백

- 오라클 : SS와 달리 DDL 문장 수행 후 자동으로 COMMIT 수행

30.

- 트랜잭션 : 데이터베이스의 논리적 연산단위로서 밀접히 관련되어 분리될 수 없는 한 개 이상의 데이터베이스 조작

31.

ROLLBACK은 최초의 BEGIN TRANSACTION 시점까지 이후 문장을 모두 되돌림

32. SAVEPOINT

- 저장점 정의시 롤백할 때 트랜잭션에 포함된 전체 작업이 아니라 현 시점에서 SAVEPOINT까지 트랜잭션의 일부만 롤백할 수 있음

- SAVEPOINT ? ROLLBACK TO ?

- SAVE TRANSACTION ? ROLLBACK TRANSACTION ?

33. WHERE

- FROM절 다음에 위치, 컬럼명 (비교연산자) 문자,숫자,표현식 – 비교 컬럼명

34. 비교연산자 우선순위

NOT – AND – OR

35. NULL의 사칙연산

- NULL값이 포함된 사칙연산의 결과는 NULL이다

36. NULL의 조건절

- IS NULL, IS NOT NULL만 가능

37. 오라클과 SS에서의 NULL 처리

1) Question : 서비스 번호 컬럼의 모든 레코드가 ‘001’과 같은 숫자형식으로 입력되어 있어야 오류가 발생하지 않는다? -> 서비스 번호에 ‘AA1’값이 하나라도 있다면 오류 발생.

2) ‘’은 오라클에서 NULL로 입력됨

3) IS NULL 조건으로 조회해야 함.

4) SS에서눈 서비스명=’’으로 조회해야 함.

- ‘’에 대해서 오라클은 NULL로, SS는 ‘’문자 그대로 입력하여야 한다.

38. 날짜 관련 함수

아래는 모두 같은 표현이다.

- CONCAT(A, B) = ‘2014120100’

- (A, B) IN ((‘20141201’, ‘00’))

- A = ‘20141201’ AND B = ‘00’

- A|B = ‘2014120100’ (오라클 : ‘|’, SS : ‘+’)

- Question : WHERE TO\_DATE(‘201501’, ‘YYYYMM’) = SVC\_END\_DATE 이면 2015년 1월 1일 000000시가 된다. 따라서, SVC\_END\_DATE가 해당 시간까지 해당하는 하나만 해당한다.

40. 단일행 함수와 다중행 함수

가) 함수의 입력 행수에 따라 단일행 함수와 다중행 함수로 구분

나) 단일행 함수는 SELECT, WHERE, ORDERBY, UPDATE의 SET절에 사용 가능

다) 1:M 관계의 두 테이블을 조인할 경우, M쪽에서 출력된 행이 하나씩 단일행 함수의 입력값으로 사용되므로 단일행 함수 사용 가능

라) 다중행 함수도 여러 개의 인수가 입력되면 단일 값만을 반환함

- 단일행 함수

- 데이터 타입 변경 가능

- 중첩해서 사용 가능

- 추출되는 각 행마다 작업을 수행

- 각 행마다 하나의 결과를 반환

- 다중행 함수

- 그룹(집계) 함수!

41. REPLACE 함수

- REPLACE(C1, CHR(10), 공백) : C1에서의 CHR(10)의 값을 없애라!

|  |  |  |
| --- | --- | --- |
| LEN(C1) | LEN(R함수) | +1한 값 |
| 3 | 2 | 2 |
| 5 | 3 | 3 |

- 단일행 문자형 함수의 종류

-- SUBSTR/SUBSTRING(문자열, m[, n]) : 문자열 중 m위치에서 n개의 문자 길이에 해당하는 문자를 돌려준다. N이 생략되면 마지막 문자까지.

-- LTRIM(문자열 [, 지정문자]) : SS에서는 지정문자를 사용할 수 없으므로 공백만 제거할 수 있음

42. 날짜 다루기

TO\_DATE(‘2015.01.10 10’, ‘YYYY.MM.DD HH24’) -> 2015.01.10 10시

1/24/(60/10) -> 10분

⨀ DATE 타입에 대한 연산

- DATE + NUMBER(일수), NUMBER/24(시간), NUMBER/24/?(분)

- 매일밤 11시 : TRUNCATE(SYSDATE) + 23/24

43. Searched\_Case\_expression을 simple\_case\_expression으로 변환하는 문제

- case when loc = ‘NY’ THEN ‘EAST == CASE LOC WHEN ‘NY’ THEN’ ‘EAST

44. CASE, ELSE ?

- ISNULL(SUM(CASE WHEN POSITION = ‘FW’ THEN 1 END), 0) FW, …

- NVL로 대체 가능, 이때 ELSE 0으로 줘야 함!

45.

Question) 4번도 맞는 결과 아닌가?

46. NULL 관련 함수

- nvl(a, b), isnull(a, b) : a의 결과값이 null이면 b를 출력한다.

- nvl2(col, a, b) : col이 널이 아니면 a, 널이면 b를 리턴한다.

- nullif(a, b) : a와 b가 같으면 null, 같지 않으면 a를 리턴한다.

- coalesce(a, b) : null이 아닌 최초의 표현식을 나타낸다. 없으면 null 리턴한다.

47.

NULL이 들어간 연산은 NULL이 결과

48.

데이터가 행 3개면 SELECT하면 3개의 결과값이 하나씩 다 나옴

50.

AVG 연산은 NULL을 아예 취급하지 않고 계산

⨀ 집계함수의 종류

- COUNT(\*) : NULL 값을 포함

51.

- SELECT COUNT(GRADE) 는 숫자 결과를 출력

- IN(NULL)은 불가능

52. 서브쿼리

- 서브쿼리는 인라인 뷰(inline view)에서 사용 불가능하고, 이를 활용하기 위해서는 where 절에서 사용되어야 한다.

⨀ GROUP BY절 vs HAVING 절

- GROUP BY절 : SELECT절과는 달리 ALIAS 명을 사용할 수 없음, 집계함수 사용 가능, WHERE가 GB보다 먼저 실행되어 행들을 미리 제거시킴

- HAVING절 : GROUP BY절의 기준 항목이나 소그룹의 집계 함수를 이용한 조건을 표시할 수 있음, GB 절에 의한 소그룹별로 만들어진 집계 데이터 중 , HAVING절에서 제한 조건을 두어 조건을 만족하는 내용만 출력한다. GROUP BY절 뒤에 위치한다.

53. 중첩된 집계 함수의 SELECT

- 중첩된 집계 함수 AVG(COUNT(\*))는 결과값이 1개일 수밖에 없으므로 SELECT 절에서 GROUP BY절에 기술된 메뉴ID와 사용유형코드는 기술될 수 없다.

54.

55. ORDER BY에 CASE구문

- ORDER BY (CASE WHEN ID = 900 THEN 0 ELSE ID END) : ID가 900인 경우는 0으로 만들고, 그게 아니라면 해당 ID값을 그대로 가져간다.

56. SELECT, GROUP BY, HAVING절 포함 규칙

- 오라클 : 행기반 DB이므로 SELECT에 없는 컬럼으로 정렬 가능하다.

- 불가능한 경우

SELECT 지역, 매출금액

FROM (

SELECT 지역, 매출 금액

FROM 지역별매출

)

ORDER BY 년 ASC;

⨀ SELECT 절과 GROUP BY 규칙

- SELECT 절의 그룹함수 제외한 모든 컬럼들은 GROUP BY 절에 사용되어야 한다.

- ORDER BY, HAVING절은 GROUP BY 절과 관련하여 그 기준이 되는 컬럼 or 그룹 함수가 사용 가능

57. ORDER BY절

- ORDER BY 절에 컬럼 대신 Alias 명이나 컬럼 순서를 나타내는 정수를 혼용하여 사용할 수 있다.

- Question) 그러면 먼저 있는 표현식에 대하여 동일할 경우 그때 에서 두번째 표현식에 대해서 정렬하는 경우를 말하는 것인가?

58. ORDER BY절의 CASE구문

59. SELECT 문장의 실행 순서

- FROM – WHERE – GROUP BY – HAVING – SELECT – ORDER BY

60. TOP(N)에서 동일한 값의 경우

- WITH TIES을 ORDER BY 절과 함께 사용

61. JOIN 조건

- 여러 테이블로부터 원하는 데이터를 조회하기 위해서 전체 테이블 개수에서 최소 N-1개의 JOIN 조건 필요함

⨀ JOIN 문장

- WHERE절에 테이블1.컬럼1 = 테이블2.컬럼2;로 JOIN 조건 작성

- INNER JOIN일 시 ON 절에 JOIN조건을 작성

62. JOIN시 WHERE절

- SELECT 절의 컬럼은 영화와 배우 테이블에서 가져옴

- WHERE 절의 영화.영화번호 = 출연.영화번호 형태로 작성

63. JOIN시 DBMS 옵티마이저

- DBMS 옵티마이저는 From 절에 나열된 테이블이 아무리 많아도 항상 2개의 테이블씩 짝을 지어서 Join을 수행한다.

64. LIKE 연산을 이용한 JOIN

- Question) SELECT COUNT(\*)가 되면 개수가 출력되어야 하는 거 아닌가?

65. 순수 관계 연산자

- SELECT, PROJECT, JOIN, DIVIDE

66. INNER JOIN, 원하는 값만 출력

- 고객도 있어야 하고, 추천컨텐츠도 있어야 하니까 INNER JOIN하는 게 맞음

- 맞는 경우1

SELECT C.컨텐츠ID, C.컨텐츠명

FROM 고객 A INNER JOIN 추천컨텐츠 B

ON(A.고객ID = B.고객ID) INNER JOIN 컨텐츠 C

ON(B.컨텐츠ID = C.컨텐츠ID) LEFT OUTER JOIN 비선호컨텐츠 D

ON (B.고객ID = D.고객ID AND B.컨텐츠ID = D.컨텐츠 ID)

WHERE A.고객ID = #custId#

AND B.추천대상일자 = TO\_CHAR(SYSDATE, ‘YYYY.MM.DD’)

AND D.컨텐츠ID IS NULL

🡪 ON 절에 넣기)

A, B, C, D INNER JOIN필요하니까 하고, B기준으로 JOIN해야 하니까 LEFT OUTER JOIN해야 하고, 그럼 비선호컨텐츠의 경우엔 NULL로 채워져 있게 조인이 됨. 그래서 IS NULL인 걸로 뽑아오면 비선호는 제외돼서 출력됨.

🡪 WHERE 절에 넣기)

WHERE B.추천대상일자 = TO\_CHAR(SYSDATE, ‘YYYY.MM.DD’)

AND NOT EXISTS (SELECT X.컨텐츠ID

FROM 비선호컨텐츠 X

WHERE X.고객ID = B.”

AND X.컨텐츠ID=B.”

1) 고객ID 일치하는 경우뿐만 아니라, 컨텐츠ID가 동일한 경우도 함께 AND로 넣어야 한다. 🡪 고객 ID 가지고만 컨텐츠ID를 셀렉을 했는데, 비선호컨텐츠 등록하지 않은 것까지 등록 이력만 있으면 다 제외돼서 출력

2) RIGHT OUTER JOIN 🡪 오른쪽 테이블이 기준이 되고, 오른쪽 비선호컨텐츠 데이터는 모두 출력되는데, 왼쪽 테이블에서 오른쪽과 매핑되는 데이터만 출력됨.

67. INNER JOIN

1)

FROM 제품X, 생산제품Y, 생산라인Z

WHERE X.제품코드 = Y.제품코드

AND Z.라인번호 = Y.라인번호

2) 테이블 사이의 JOIN이 조건이 없는 경우엔 카티시안 곱이 발생

3) Question) FK이 NULL일 수도 있어서 그런건가?

68.

- 고객 <-> 구매 정보 -> 구매이력이 있어야 하므로

INNER JOIN 구매정보 B ON A.고객번호 = B.고객번호

69.

- INNER JOIN은 한번에 3개씩 하면 안됨

70. USING 조건절 이용

- JOIN 칼럼에 대해서 Alias나 테이블 이름과 같은 접두사 붙일 수 없음