

## 핵심정리

RANK 함수는 ORDER BY를 포함한 QUERY 문에서 특정 항목(칼럼)에 대한 순위를 구하는 함수이며 동일한 값에 대해서는 동일한 순위를 부여하게 된다.

RANK(): 순위

1  
2  
2  
4

DENSE\_RANK():

1  
2  
2  
3

ROW\_NUMBER():

112

다음 중 윈도우 함수(Window Function, Analytic Function)에 대한 설명으로 가장 부적절한 것은?

- ① Partition과 Group By 구문은 의미적으로 유사하다.
- ② Partition 구문이 없으면 전체 집합을 하나의 Partition으로 정의한 것과 동일하다.
- ③ 윈도우 함수 처리로 인해 결과 건수가 줄어든다.
- ④ 윈도우 함수 적용 범위는 Partition을 넘을 수 없다.

113

다음 중 아래와 같은 테이블에서 SQL을 실행할 때 결과로 가장 적절한 것은?

아 래

[테이블 : 고객]

고객번호(PK)	고객명
001	홍길동
002	이순신
003	강감찬
004	이상화
005	이규혁

[테이블 : 월별매출]

월(PK)	고객번호(PK)	매출액
201301	001	200
201301	002	300
201301	003	250
201301	004	300
201301	005	250
201302	001	150
201302	002	150
201302	004	200
201302	005	100
201303	002	100
201303	003	100
201303	004	200
201303	005	350

[SQL]

```
SELECT 고객번호, 고객명, 매출액
      ,RANK() OVER(ORDER BY 매출액 DESC) AS 순위
FROM
    (
        SELECT A.고객번호,
               MAX(A.고객명) AS 고객명,
               SUM(B.매출액) AS 매출액
        FROM 고객 A INNER JOIN 월별매출 B
        ON (A.고객번호 = B.고객번호)
        GROUP BY A.고객번호
    )
ORDER BY RNK;
```

1  
2  
3  
4  
5

①

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	1
002	이순신	550	3
001	홍길동	350	4
003	강감찬	350	4

②

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	2
002	이순신	550	3
001	홍길동	350	4
003	강감찬	350	5

③

고객번호	고객명	매출액	순위
005	이규혁	700	1
004	이상화	700	1
002	이순신	550	2
001	홍길동	350	3
003	강감찬	350	3

④

고객번호	고객명	매출액	순위
003	강감찬	350	1
001	홍길동	350	1
002	이순신	550	2
004	이상화	700	3
005	이규혁	700	3

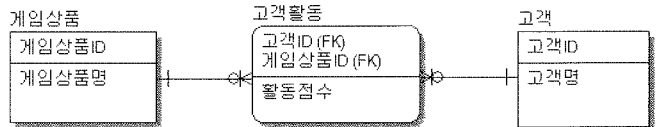
## 핵심정리

DENSE\_RANK 함수는 RANK 함수와 흡사하나, 동일한 순위를 하나의 간수로 취급하는 것이 다른 점이다.

114

아래 데이터 모델에서 활동점수가 높은 고객을 게임상품ID별로 10등까지 선별하여 이벤트를 진행하려고 한다. 다음 SQL 중 가장 적절한 것은? (단, 활동점수가 동일한 고객은 동일등수로 한다. 아래 결과 예제 참조)

아 래



[결과 예제]

게임상품ID	고객ID	활동점수	등수(순위)
001	121	150	1
001	111	150	1
001	234	110	3
001	212	100	4
001	455	100	4
001	182	90	6
001	199	80	7
001	109	70	8
001	876	70	8
001	232	40	10
001	901	40	10
.....	.....	.....	.....

① SELECT 게임상품ID, 고객ID, 활동점수, 순위  
 FROM (SELECT DENSE\_RANK() OVER(ORDER BY 활동점수 DESC) AS 순위  
 ,고객ID, 게임상품ID, 활동점수  
 FROM 고객활동)  
 WHERE 순위 <= 10;

② SELECT 게임상품ID, 고객ID, 활동점수, 순위  
 FROM (SELECT DENSE\_RANK() OVER(PARTITION BY 게임상품ID  
 ORDER BY 활동점수 DESC) AS 순위  
 ,고객ID, 게임상품ID, 활동점수  
 FROM 고객활동)  
 WHERE 순위 <= 10;

③ SELECT 게임상품ID, 고객ID, 활동점수, 순위  
 FROM (SELECT RANK() OVER(ORDER BY 활동점수 DESC) AS 순위  
 ,고객ID, 게임상품ID, 활동점수  
 FROM 고객활동)  
 WHERE 순위 <= 10;

④ SELECT 게임상품ID, 고객ID, 활동점수, 순위  
 FROM (SELECT RANK() OVER(PARTITION BY 게임상품ID  
 ORDER BY 활동점수 DESC) AS 순위  
 ,고객ID, 게임상품ID, 활동점수  
 FROM 고객활동)

## 핵심정리

ROW\_NUMBER 함수는 RANK나 DENSE\_RANK 함수가 동일한 값에 대해서는 동일한 순위를 부여하는데 반해, 동일한 값이라도 고유한 순위를 부여한다.

115

다음 중 추천내역 테이블에서 아래와 같은 SQL을 수행하였을 때의 결과로 가장 적절한 것은?

아 래

[테이블 : 추천내역]

추천경로	추천인	피추천인	추천점수
SNS	나한일	강감찬	75
SNS	이순신	강감찬	80
이벤트응모	홍길동	강감찬	88
이벤트응모	저절로	이순신	78
홈페이지	저절로	이대로	93
홈페이지	홍두깨	심청이	98

[SQL]

```
SELECT 추천경로, 추천인, 피추천인, 추천점수
FROM (SELECT 추천경로, 추천인, 피추천인, 추천점수
      ,ROW_NUMBER() OVER(PARTITION BY 추천경로
      ORDER BY 추천점수 DESC) AS RNUM
      FROM 추천내역)
WHERE RNUM = 1;
```

①

추천경로	추천인	피추천인	추천점수
SNS	나한일	강감찬	75
SNS	이순신	강감찬	80
이벤트응모	홍길동	강감찬	88
이벤트응모	저절로	이순신	78
홈페이지	저절로	이대로	93
홈페이지	홍두깨	심청이	98

②

추천경로	추천인	피추천인	추천점수
홈페이지	홍두깨	심청이	98

③

추천경로	추천인	피추천인	추천점수
SNS	이순신	강감찬	80
이벤트응모	홍길동	강감찬	88
홈페이지	홍두깨	심청이	98

④

추천경로	추천인	피추천인	추천점수
SNS	나한일	강감찬	75
이벤트응모	저절로	이순신	78
홈페이지	저절로	이대로	93

116

다음 중 아래의 SQL에 대한 설명으로 가장 적절한 것은?

아래

[SQL]

```

SELECT  상품분류코드
        ,AVG(상품가격) AS 상품가격
        ,COUNT(*) OVER(ORDER BY AVG(상품가격)
                        RANGE BETWEEN 10000 PRECEDING
                        AND 10000 FOLLOWING) AS 유사개수
FROM    상품
GROUP BY 상품분류코드;

```

-10000 ~ +10000

- ① WINDOW FUNCTION을 GROUP BY 절과 함께 사용하였으므로 위의 SQL은 오류가 발생한다.
- ② WINDOW FUNCTION의 ORDER BY절에 AVG 집계 함수를 사용하였으므로 위의 SQL은 오류가 발생한다.
- ③ 유사개수 컬럼은 상품분류코드별 평균상품가격을 서로 비교하여 -10000 ~ +10000 사이에 존재하는 상품분류코드의 개수를 구한 것이다.
- ④ 유사개수 컬럼은 상품전체의 평균상품가격을 서로 비교하여 -10000 ~ +10000 사이에 존재하는 상품의 개수를 구한 것이다.

117

다음 중 [사원] 테이블에 대하여 아래와 같은 SQL을 수행하였을 때 예상되는 결과로 가장 적절한 것은?

아래

[테이블 : 사원]

사원ID	부서ID	사원명	연봉
001	100	홍길동	2500
002	100	강감찬	3000
003	200	김유신	4500
004	200	김선달	3000
005	200	유학생	2500
006	300	변사포	4500
007	300	박문수	3000

[SQL]

```
SELECT Y.사원ID, Y.부서ID, Y.사원명, Y.연봉
FROM (SELECT 사원ID, MAX(연봉) OVER(PARTITION BY
      부서ID) AS 최고연봉
      FROM 사원) X, 사원 Y
WHERE X.사원ID = Y.사원ID
AND X.최고연봉 = Y.연봉
```

①	사원ID	부서ID	사원명	연봉
	002	100	강감찬	3000
	003	200	김유신	4500
	006	300	변사포	4500

②	사원ID	부서ID	사원명	연봉
	001	100	홍길동	2500
	005	200	유학생	2500
	007	300	박문수	3000

③	사원ID	부서ID	사원명	연봉
	003	200	김유신	4500
	006	300	변사포	4500

④	사원ID	부서ID	사원명	연봉
	004	200	김유신	4500



### 핵심정리

LAG 함수를 이용해  
파티션별 윈도우에서  
이전 몇 번째 행의 값을  
가져올 수 있다.  
이후 몇 번째 행의 값을  
가져오는 것은 LEAD  
함수이며, SQL  
Server에서는 지원하지  
않는 함수이다.

118

다음 중 아래 SQL의 실행 결과로 가장 적절한 것은?

아래

```
CREATE TABLE TBL
(ID VARCHAR2(10),
 START_VAL NUMBER,
 END_VAL NUMBER)
TBL
```

ID	START_VAL	END_VAL
A	10	14
A	14	15
A	15	15
A	15	18
A	20	25
A	25	

```
SELECT ID, START_VAL, END_VAL
FROM (
SELECT ID, START_VAL, NVL(END_VAL, 99) END_VAL,
(CASE WHEN START_VAL = LAG(END_VAL) OVER (PARTITION BY ID
ORDER BY START_VAL, NVL(END_VAL, 99)) THEN 1
ELSE 0
END) FLAG1,
(CASE WHEN END_VAL = LEAD(START_VAL) OVER (PARTITION BY ID
ORDER BY START_VAL, NVL(END_VAL, 99)) THEN 1
ELSE 0
END) FLAG2
FROM TBL)
WHERE FLAG1 = 0
OR FLAG2 = 0
```

①

ID	START_VAL	END_VAL
A	10	14
A	15	18
A	20	25
A	25	99

②

ID	START_VAL	END_VAL
A	15	15

③

ID	START_VAL	END_VAL
A	10	14
A	15	15
A	20	99

④

ID	START_VAL	END_VAL
A	10	18
A	20	99



119

아래 설명 중 ㉠, ㉡에 해당하는 내용을 작성하시오.

GRANT ✓ TO  
REVOKE ✓ FROM

GRANT → REVOKE

아 래

DBMS에 생성된 USER와 다양한 권한들 사이에서 중개 역할을 할 수 있도록 DBMS에서는 ROLE을 제공한다. 이러한 ROLE을 DBMS USER에게 부여하기 위해서는 ㉠ 명령을 사용하며, ROLE을 회수하기 위해서는 ㉡ 명령을 사용한다.

120

다음 중 B\_User가 아래의 작업을 수행할 수 있도록 권한을 부여하는 DCL로 가장 적절한 것은?

아 래

UPDATE A\_User.TB\_A  
SET col1='AAA'  
WHERE col2=3

- ① GRANT SELECT, UPDATE TO B\_User;
- ② REVOKE SELECT ON A\_User.TB\_A FROM B\_User;
- ③ DENY UPDATE ON A\_User.TB\_A TO B\_User;
- ④ GRANT SELECT, UPDATE ON A\_User.TB\_A TO B\_User;

121

아래의 ㉠에 들어갈 내용을 쓰시오.

ROLE

아 래

DBMS 사용자를 생성하면 기본적으로 많은 권한을 부여해야 한다. 많은 DBMS에서는 DBMS 관리자가 사용자별로 권한을 관리해야 하는 부담과 복잡함을 줄이기 위하여 다양한 권한을 그룹으로 묶어 관리할 수 있도록 사용자와 권한 사이에서 중개 역할을 수행하는 ㉠을 제공한다.

## 핵심정리

### PL/SQL의 특징

- PL/SQL은 Block 구조로 되어있어 각 기능별로 모듈화가 가능하다.
- 변수, 상수 등을 선언하여 SQL 문장 간 값을 교환한다.
- IF, LOOP 등의 절차형 언어를 사용하여 절차적인 프로그램이 가능하도록 한다.
- DBMS 정의 예러나 사용자 정의 예러를 정의하여 사용할 수 있다.
- PL/SQL은 Oracle에 내장되어 있으므로 Oracle과 PL/SQL을 지원하는 어떤 서버로도 프로그램을 옮길 수 있다.
- PL/SQL은 응용 프로그램의 성능을 향상시킨다.
- PL/SQL은 여러 SQL 문장을 Block으로 묶고 한 번에 Block 전부를 서버로 보내기 때문에 통신량을 줄일 수 있다.

122

사용자 Lee가 릴레이션 R을 생성한 후, 아래와 같은 권한부여 SQL문들을 실행하였다. 그 이후에 기능이 실행 가능한 SQL을 2개 고르시오.  
(단, A, B의 데이터 타입은 정수형이다)

Kim: S, I, D  
Park: S, I, D

아래

Lee: GRANT SELECT, INSERT, DELETE ON R TO Kim WITH GRANT OPTION;  
Kim: GRANT SELECT, INSERT, DELETE ON R TO Park;  
Lee: REVOKE DELETE ON R FROM Kim;  
Lee: REVOKE INSERT ON R FROM Kim CASCADE;

- ① Park: SELECT \* FROM R WHERE A = 400;
- ② Park: INSERT INTO R VALUES(400, 600);
- ③ Park: DELETE FROM R WHERE B = 800;
- ④ Kim : INSERT INTO R VALUES(500, 600);

123

다음 중 PL/SQL에 대한 설명으로 가장 부적절한 것은?

- ① 변수와 상수 등을 사용하여 일반 SQL 문장을 실행할 때 WHERE절의 조건 등으로 대입할 수 있다.
- ② Procedure, User Defined Function, Trigger 객체를 PL/SQL로 작성할 수 있다.
- ③ PL/SQL로 작성된 Procedure, User Defined Function은 전체가 하나의 트랜잭션으로 처리되어야 한다.
- ④ Procedure 내부에 작성된 절차적 코드는 PL/SQL엔진이 처리하고 일반적인 SQL 문장은 SQL실행기가 처리한다.

## 핵심정리

### 저장 모듈(Stored Module)

SQL 문장을 데이터 베이스 서버에 저장하여 사용자와 애플리케이션 사이에서 공유할 수 있도록 만든 일종의 SQL 컴포넌트 프로그램이며, 독립적으로 실행되거나 다른 프로그램으로부터 실행될 수 있는 완전한 실행 프로그램이다. Oracle의 저장 모듈에는 Procedure, User Defined Function, Trigger가 있다.

124

아래는 임시부서(TMP\_DEPT) 테이블로부터 부서(DEPT) 테이블에 데이터를 입력하는 PL/SQL 이다. 부서 테이블에 데이터를 입력하기 전에 부서 테이블의 모든 데이터를 ROLLBACK이 불가능 하도록 삭제 하려고 한다. 다음 중 ㉠에 들어갈 내용으로 옳은 것은?

아 래

[PL/SQL]

```
create or replace procedure insert_dept authid current_user
as
begin
    ㉠
    INSERT /*+ APPEND */ INTO DEPT (DEPTNO, DNAME,
    LOC)
    SELECT DEPTNO, DNAME, LOC
    FROM   TMP_DEPT;
    commit;
end;
/
```

- ☒ ㉠ TRUNCATE TABLE DEPT; 즉시
- ☒ ㉡ DELETE FROM DEPT;
- ☒ ㉢ execute immediate 'TRUNCATE TABLE DEPT';
- ☒ ㉣ execute 'TRUNCATE TABLE DEPT';

125

다음 중 절차형 SQL 모듈에 대한 설명으로 가장 부적절한 것은?

- ☒ ㉠ 저장형 프로시저는 SQL을 로직과 함께 데이터베이스 내에 저장해 놓은 명령문의 집합을 의미한다.
- ☒ ㉡ 저장형 함수(사용자 정의 함수)는 단독적으로 실행되기 보다는 다른 SQL문을 통하여 호출되고 그 결과를 리턴하는 SQL의 보조적인 역할을 한다.
- ☒ ㉢ 트리거는 특정한 테이블에 INSERT, UPDATE, DELETE와 같은 DML 문이 수행되었을 때 데이터베이스에서 자동으로 동작하도록 작성된 프로그램이다.
- ☒ ㉣ 데이터의 무결성과 일관성을 위해서 사용자 정의 함수를 사용한다.

→ 제약조건      독립적성

126 다음 중 Trigger에 대한 설명으로 가장 부적절한 것은?

- ☒ ① Trigger는 데이터베이스에 의해서 자동으로 호출되고 수행된다.
- ☒ ② Trigger는 특정 테이블에 대해서 INSERT, UPDATE, DELETE 문이 수행되었을 때 호출되도록 정의할 수 있다.
- ☒ ③ Trigger는 TCL을 이용하여 트랜잭션을 제어할 수 있다.
- ☒ ④ Trigger는 데이터베이스에 로그인하는 작업에도 정의할 수 있다.

127 다음 중 특정한 테이블에 INSERT, UPDATE, DELETE와 같은 DML문이 수행 되었을 때, 데이터베이스에서 자동으로 동작하도록 작성된 저장 프로그램으로 가장 적절한 것은? (단, 사용자가 직접 호출하여 사용하는 것이 아니고 데이터 베이스에 서 자동적으로 수행하게 된다.)

- ☒ ① PROCEDURE
- ☒ ② USER DEFINED FUNCTION
- ☒ ③ PACKAGE
- ☒ ④ TRIGGER

#### 프로시저와 트리거의 차이점

프로시저	트리거
CREATE Procedure 문법사용	CREATE Trigger 문법사용
EXECUTE 명령어로 실행	생성 후 자동으로 실행
COMMIT, ROLLBACK 실행 가능	COMMIT, ROLLBACK 실행 안됨