**실습 과제 보고서**

|  |  |  |  |
| --- | --- | --- | --- |
| 연도 | 2021년 | 학기 | 1학기 |
| 과목명 | 데이터베이스 프로그래밍 | 분반 | 분반 |
| 실습 번호 | **Lab02** | | |
| 이름 | 문정현 | 학번 | 1815060 |

**[실습]**

|  |  |  |
| --- | --- | --- |
| 1. 실행하여 SQL문과 결과창을 캡쳐하세요.  (1) **show user;** -> 과목번호200011인 수업을 듣는 학생의 학번, 이름, 학년, 전공 을 출력  (2) **show user;**->과목번호200013인 수업을 가르치는 교수님의 교번, 이름, 대학, 전공을 출력 | | |
| (1) |  | |
| (2) |  | |
| 2. 실행하여 SQL문과 결과창을 캡쳐하세요.  (1) 연봉을 1000미만으로 받는 사원이 소속된 부서 중 부서번호가 최소인 부서와 동일한 부서에서 근무하는 사원들의 정보를 출력.  (2) 세일즈맨의 최저급여를 받는 사원보다 많은 급여를 받는 **'CLERK'**인 직원들의 정보 출력  (3) 각각 출력하여 결과창을 캡쳐하고 ALL과 ANY의 차이점을 **설명하세요.**  select \* from emp where sal**<ALL**(SELECT SAL FROM EMP WHERE JOB='SALESMAN');  select \* from emp where sal**>ALL**(SELECT SAL FROM EMP WHERE JOB='SALESMAN');  select \* from emp where sal**<ANY**(SELECT SAL FROM EMP WHERE JOB='SALESMAN');  select \* from emp where sal**>ANY**(SELECT SAL FROM EMP WHERE JOB='SALESMAN'); | | |
| (1) |  | |
| (2) |  | |
| (3) | **(설명) :**  **ANY는 조건을 만족하는 값이 하나라도 있으면 결과를 리턴하고, ALL은 모든 값이 조건을 만족해야 결과를 리턴한다.**  **< ALL 값들 중 최소값 보다 작으면 결과를 리턴**  **> ALL 값들 중 최대값 보다 크면 결과를 리턴**  **< ANY 값들 중 최대값 보다 작으면 결과를 리턴**  **> ANY 값들 중 최소값 보다 크면 결과를 리턴**  **(캡쳐)** | |
| **3.** Integrity Constraints 실습   * 현재 사용자 보여주고, BOOK1 (title : not null, author, pub\_year, publisher)을 갖는 테이블을 생성하고, 테이블 구조를 보인 후, NOT NULL 제약 조건을 위배하는 INSERT 문을 이용하여 레코드 삽입을 시도한 것을 보여주고 조건을 만족하도록 insert문을 고쳐서 삽입해준다. (화면 캡처) * 제약조건을 삭제하지 않고 , BOOK1 테이블의 NOT NULL 조건을 적용시키지 않으려면 어떻게 해야 하는가 ? (명령문만 직접 보고서에 써주세요.) * Book2(book 테이블 수정)의 title을 외래키로 갖는 review테이블 생성하고 무결성 제약조건에 위배되는 레코드 삽입하고 이를 보여준다.(화면 캡쳐) | | |
| (1) |  | |
| (2) | **ALTER TABLE BOOK1 DISABLE CONSTRAINT SYS\_C007093;** | |
| (3) |  | |
| **4.** Mybook 테이블을 생성하고 NULL 에 관한 다음 SQL문 실행해보세요. 결과를 보면서 NULL 에 대한 개념을 정리하며 각각의 문제마다 간단히 설명하세요. | | |
| No | 캡쳐 | **설명(간단히)** |
| (1) |  | **NVL(price,0)은 price열의 값이 NULL인 경우에는 해당 열의 값을 0으로 출력하는 함수이다.** |
| (2) |  | **비교연산자로 비교가 불가능하여 원하는 열의 값이 NULL인 행을 찾을때 IS NULL을 사용해야 한다.** |
| (3) |  | **비교연산자로 비교가 불가능하여 원하는 열의 값이 NULL인 행을 찾을때 비교 연산자(=)는 사용하면 찾을 수 없다.** |
| (4) |  | **NULL+숫자의 결과는 NULL로 나온다.** |
| (5) |  | **해당하는 행이 하나도 없는 경우 SUM(), AVG()의 결과는 NULL, COUNT() 결과는 0이 나온다** |
| (6) |  | **COUNT(컬럼명)을 사용하면 NULL값은 제외하고 COUNT 하고 COUNT(\*)를 사용하면 NULL도 포함하여 전부 COUNT한다.** |
| (7) |  | **테이블에 대한 SUM()과 AVG(**  **)를 수행할 때 NULL값이 있는 행은 계산에서 제외된다.** |