

능단평2_재시험



날짜

@2024년 12월 26일

아래 요구사항을 잘 읽고 본인이름.zip파일로 압축하여 카톡으로 전달하세요.



- zip파일에 있어야하는 파일 본인이름_test2.sql, cmd 캡처, 디비버 전체화면(프로젝트, 계정 보이게 캡처)
- 계정명 : DBTEST / 비밀번호 : 1234
- 디비버 프로젝트 명 : 본인이름_test2
- CMD 캡처 파일명 : 본인이름_CMD계정.jpg
- 디비버 캡처 파일명 : 본인이름_디비버계정.jpg

▼ 1. 아래 요구사항에 맞게 테이블을 구성하세요.

1) 요구사항 분석

도서관에서 회원의 정보와 책의 정보가 필요하다

회원 정보 : 회원번호, 이름, 나이, 핸드폰번호, 주소

책의 정보 : 도서번호, 책이름, 장르

- 한명의 회원은 여러권을 책을 대여할 수 있다
- 테이블 명 : WEB_MEMBER, WEB_BOOK (대여 테이블은 생각하지 않는다)
- 제약조건 : PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK

- 회원번호 (Member_ID)와 도서번호 (Book_ID)는 각각의 테이블에서 UNIQUE해야 함.

- 장르 (Genre)는 '인문학', 'IT', '추리', '경영' 중 하나여야 함.

- 회원의 나이, 핸드폰번호, 주소는 NULL일 수 있음.

2) 개념적 설계

3) 논리적 설계

4) 물리적 설계

5) 구현

▼ 2. TBL_TEST_STUDENT 테이블을 생성하고 10명의 학생을 추가한 후 테이블을 조회하세요.

컬럼과 타입은 아래 결과 화면과 동일하며 STU_ID는 PK로 설정되어있고, 모든 컬럼에는 NOT NULL 제약조건이 설정되어있다.

10명의 학생은 아래와 같다.

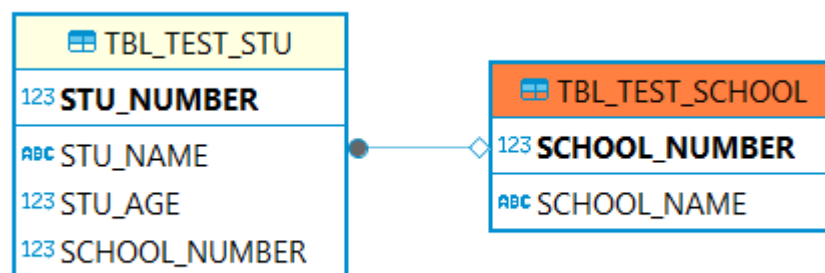
(1, '최현욱', '010-1234-5678', 20, '컴퓨터 공학과')
(2, '고민시', '010-2345-6789', 22, '인공지능학과')
(3, '신예은', '010-3456-7890', 21, '정보보호학과')
(4, '이재욱', '010-4567-8901', 23, '컴퓨터 공학과')
(5, '김소현', '010-5678-9012', 20, '인공지능학과')
(6, '김지원', '010-6789-0123', 24, '정보보호학과')
(7, '이도현', '010-7890-1234', 22, '컴퓨터 공학과')
(8, '여진구', '010-8901-2345', 21, '인공지능학과')
(9, '차은우', '010-9012-3456', 23, '정보보호학과')
(10, '박은빈', '010-0123-4567', 20, '컴퓨터 공학과')

	123 STU_ID ▼	ABC STU_NAME ▼	ABC STU_PHONE ▼	123 STU_AGE ▼	ABC STU_MAJOR ▼
1	1	최현욱	010-1234-5678	20	컴퓨터 공학과
2	2	고민시	010-2345-6789	22	인공지능학과
3	3	신예은	010-3456-7890	21	정보보호학과
4	4	이재욱	010-4567-8901	23	컴퓨터 공학과
5	5	김소현	010-5678-9012	20	인공지능학과
6	6	김지원	010-6789-0123	24	정보보호학과
7	7	이도현	010-7890-1234	22	컴퓨터 공학과
8	8	여진구	010-8901-2345	21	인공지능학과
9	9	차은우	010-9012-3456	23	정보보호학과
10	10	박은빈	010-0123-4567	20	컴퓨터 공학과

▼ 3. 2번에서 만든 테이블에서 아래 요구사항대로 쿼리문을 작성하세요.

- 1) 학과를 중복없이 조회하기(select절, from절만 이용해서 조회할 것)
- 2) 학생수가 4명 이상인 학과 조회하기(반드시 IN LINE VIEW 서브쿼리 이용)
- 3) 학번, 이름, 나이, 자신이 속한 학과의 최대나이를 조회하기(반드시 SCALAR 서브쿼리 이용)
- 4) 학생의 나이가 평균 나이보다 어린(미만인) 학생들만 조회하기(반드시 SUB QUERY 서브쿼리 이용)

▼ 4. TBL_TEST_SCHOOL 테이블과 TBL_TEST_STU 테이블이 아래 엔티티 관계도처럼 관계가 맺어져있다.
테이블을 아래 엔티티 관계도를 보고 동일하게 생성 한 뒤 값을 추가하고 조회하세요.
(단, 제약조건은 PK와 FK만 설정할 것)



[TBL_TEST_SCHOOL
테이블]

 -----(1,'DBMS 고등학교')
 (2,'JAVA 고등학교')
 (3,'PYTHON 고등학교')

	123 SCHOOL_NUMBER ▼	ABC SCHOOL_NAME ▼
1	1	DBMS 고등학교
2	2	JAVA 고등학교
3	3	PYTHON 고등학교

[TBL_TEST_STU 테이블]

 (1,'김철수',17,1)
 (2,'신짱구',17,1)
 (3,'이유리',18,2)
 (4,'김영희',19,2)

	123 STU_NUMBER ▼	ABC STU_NAME ▼	123 STU_AGE ▼	123 SCHOOL_NUMBER ▼
1	1	김철수	17	1 ↗
2	2	신짱구	17	1 ↗
3	3	이유리	18	2 ↗
4	4	김영희	19	2 ↗

▼ 5. 4번에서 만든 테이블에서 join을 활용하여 아래 문제를 해결하세요.

1) 각 학생의 이름과 그들이 소속된 학교 이름을 조회하기

	ABC STU_NAME ▼	ABC SCHOOL_NAME ▼
1	김철수	DBMS 고등학교
2	신짱구	DBMS 고등학교
3	이유리	JAVA 고등학교
4	김영희	JAVA 고등학교

2) 학교 이름과 그 학교에 속한 학생 수를 조회하기(단, 반드시 left join 이용할것)

	ABC SCHOOL_NAME ▼	123 STU_COUNT ▼
1	DBMS 고등학교	2
2	JAVA 고등학교	2
3	PYTHON 고등학교	0

3) 학생의 나이가 평균 나이 이상인 학생의 이름과 그들이 소속된 학교 이름을 조회하기

	ABC STU_NAME ▼	ABC SCHOOL_NAME ▼	123 STU_AGE ▼
1	이유리	JAVA 고등학교	18
2	김영희	JAVA 고등학교	19

▼ 7. rownum을 이용하여 TBL_TEST_STUDENT 테이블에서 나이가 많은 순서 대로(나이 내림차순으로 정렬) 3~5번째 학생들만 조회하세요.