과제 4

20164269 이현호

1. 소감

이번 과제는 재미있었습니다.

앱이 비정상적으로 종료되는 일이 잦았습니다.

보고서를 작성하는 지금 당장 생각나는 원인은 다음의 2가지 입니다.

(1) 삭제된 데이터에 접근하려고 하는 경우

(2) SQL문에 공백 없이 from절과 tablename이 한 단어로 입력된 경우 (ex : fromtablename)

삭제된 레코드를 읽거나 갱신할 수 없도록 직접 처리해 주어야 했습니다.

책에 나와있는 Cursor 객체를 이용해 레코드에 접근하는 방법론은 해당 명령어가 실행되는 시점의 남아있는 레코드의 숫자를 가지고 순회하기 때문에 삭제 연산을 수행한 후(PRIMARY KEY의 중간이 비어있는 경우)에는 정상적으로 작동하지 않습니다. 따라서 ‘조회’와 ‘갱신’기능은 새롭게 구상해야 했습니다.

“2. 각 버튼 별 수행 기능 및 구현방법” 에서 자세히 다루겠습니다.

2. 각 버튼 별 수행 기능 및 구현방법

(1) ‘메모 리스트 조회’ – 데이터베이스가 null인 경우 “데이터베이스 없음”을 출력하고, null이 아닌 경우 Cursor객체를 이용해 for문에서 메모 리스트(메모 ID, 제목)를 출력합니다.

(2) ‘생성’ - 데이터베이스가 null인 경우 “데이터베이스 없음”을 출력하고 새로 생성합니다.

입력공간이 사라진 상태(GONE)인 경우 VISIBLE로 변경합니다.

입력공간이 보이는 상태(VISIBLE)인 경우 테이블에 데이터를 삽입합니다.

(INVISIBLE이 아닌 GONE으로 설정하는 이유는 입력 공간 아래 ScrollView를 크게 보여주기 위함입니다.)

∴ 하나의 버튼이 두 가지 기능을 수행합니다. 입력공간이 없는 경우 입력공간을 보여주고, 보이는 경우 테이블에 데이터를 삽입하고 입력공간을 보이지 않게 합니다.

(3) ‘읽기’ – ‘메모 리스트 조회’에서 메모 ID와 제목을 조회한 후 사용하는 것을 의도했습니다.

‘삭제’, ‘날림’ 버튼 위의 editText에 숫자 입력 후 ‘읽기’ 버튼을 누르면 해당 ID의 메모의 제목, 내용이 출력됩니다.

(4) – ‘갱신’ – editText에 숫자 입력 후 ‘갱신’ 버튼을 누르면 2번 ‘생성’버튼에서 나타났던 입력공간이 보이고, 다시‘갱신’ 버튼을 누르면 메모 ID의 제목과 내용은 갱신됩니다.

(5) ‘날림’ – 데이터베이스를 삭제합니다. 테스트하기 위해 넣은 기능으로, 추가적인 설명은 생략하겠습니다.

아래는 ‘읽기’ 버튼에서 호출하는 executeQuery함수입니다.

==========================================================================

*//메모 내용 보여줌*public void executeQuery(String target) {  
 println("executeQuery 호출됨.");  
  
 Cursor cursor = database.rawQuery("select \* from " + *tableName* + " where " + "\_id" + " = ?", new String[]{target});  
 */\*if(cursor.getCount() == 1) { //cursor.getCount가 1이면 \_id가 일치하는 행(Row)이 있다는 뜻이다. \_id는 기본키이므로 >=를 사용할 필요가 없다.  
 (Cursor는 null을 리턴하지 않는다. 따라서 cursor != null 은 의미없는 비교식이다.)  
 cursor.moveToFirst() : Cursor를\_id가 일치하는 행 중 가장 첫 번째 행으로 이동시킨다. 일치하는 행이 있는 경우 true, 없는 경우 false 리턴  
 해당 식을 if문에 넣어 cursor를 첫 번째 행으로 이동시킴과 동시에 \_id가 일치하는 행이 있는지 검사할 수 있다. 즉 한 줄을 아낄 수 있다.\*/* if(cursor.moveToFirst()) {  
 int id = cursor.getInt(0);  
 String title = cursor.getString(1);  
 String content = cursor.getString(2);  
  
 println("메모 ID #" + id + " | " + "제목 : " + title + "\n내용 : " + content);  
 }  
 else {  
 println("메모 ID #" + target + "없음");  
 }  
 cursor.close();  
}

==========================================================================

책에 나와있는 Cursor 객체를 이용해 레코드에 접근하지 않았습니다.

Cursor cursor = database.rawQuery("select \* from " + *tableName* + " where " + "\_id" + " = ?", new String[]{target});

위 식은 editText에서 입력받은 target 값과 테이블의 \_id값을 비교해 cursor에 리턴합니다.

그리고 일치하는 레코드가 없는 경우 쿼리가 실행되지 않도록 해야 합니다. (그렇지 않으면 비정상 종료됩니다.) Cursor 클래스에 정의된 moveToFirst()메소드는 일치하는 레코드들 중 첫 번째 레코드로 이동하고, boolean값을 반환(일치하는 레코드가 있는 경우 true, 없는 경우 false)합니다. if문안에 cursor.moveToFirst() 식을 작성하면 일치하는 레코드가 있는 경우, 일치하는 레코드들 중 첫 번째 레코드로 이동하고, 없는 경우 실행하지 않고 else문으로 빠질 것입니다.

3. 실행 화면

|  |  |
| --- | --- |
| [그림 1] 첫 실행 | 1. 실행 후 화면입니다. |
| [그림 2] '메모 리스트 조회' | 2. ‘메모 리스트 조회’ 버튼 클릭 시 생성된 데이터베이스가 없어 “데이터베이스 없음”을 출력합니다. |
| [그림 3] '생성' | 3 '생성' 버튼을 클릭하면 데이터를 입력할 수 있도록 editText를 보여줍니다.  onClick실행 시  WritingArea.setVisibility(View.VISIBLE); |
| [그림 4] 다시 '생성' 버튼 클릭 | 4 다시 '생성' 버튼을 클릭하면 테이블에 메모가 추가되고 editText가 사라집니다.  WritingArea.setVisibility(View.INVISIBLE); |
| [그림 5] 메모 5개 입력 후 | 5. 샘플 메모 5개 입력 후의 화면입니다. |
| [그림 6] '메모 리스트 조회' | 6. ‘리스트 조회’ 버튼 클릭시 이전에 입력한 샘플 메모 5개의 ID와 제목이 출력됩니다. |
| [그림 7] 읽을 대상 입력 | 7. editText에 읽을 메모 ID 입력 |
| [그림 8] 읽기 결과 | 8. 읽기 결과 |
| [그림 9] 3번 메모 '삭제' 후 '메모 리스트 조회' | 9. 3번(악동뮤지션) - 어떻게 이별까지 사랑하겠어, 널 사랑하는 거지) '삭제' 버튼 클릭 후 '메모 리스트 조회' 클릭 |
| [그림 10] '갱신' | 10. 갱신 - 해당 메모 없음 출력(예외처리) |
| [그림 11] '삭제' | 11. 삭제 - 해당 메모 없음 출력(예외처리) |
| [그림 12] '갱신' | 12. 2번(방탄소년단 – Dynamite)갱신 |
| [그림 13] '읽기' | 13. 2번(AKMU (악동뮤지션) – HAPPENING) 메모 읽기 |
| [그림 14] '메모 리스트 조회' | 14. 메모 리스트가 갱신되었음을 확인 |