

# 이벤트 프로시저

엑세스 이벤트 프로시저 문제는 엑셀에 비해서 내용이 다양하기 때문에 출제율이 높은 문제를 위주로 학습하는 것이 좋습니다.

## 01 조회 프로시저

Me 개체	<ul style="list-style-type: none"> <li>현재 폼에서 현재 폼에 대한 작업을 수행합니다.</li> <li>생략해서 사용할 수 있지만, 생략하면 자동입력이 불가능합니다.</li> </ul>
Filter, FilterOn	<ul style="list-style-type: none"> <li>조건에 맞는 레코드를 필터링합니다.</li> <li>Me.Filter = "필드명 = " &amp; 컨트롤명 &amp; "" Me.FilterOn = True</li> </ul>
RecordSource	<ul style="list-style-type: none"> <li>레코드 원본을 지정하여 조회합니다.</li> <li>조건문은 반드시 Select문으로 작성합니다.</li> <li>Me.RecordSource = "Select 필드명 From 테이블/쿼리 Where 조건문"</li> </ul> <p>※ 일반적으로 모든 레코드 조회시 Select절에 *을 표시함</p>
RecordSetClone, Bookmark, FindFirst	<ul style="list-style-type: none"> <li>레코드 원본을 복제하여 찾고자 하는 특정 레코드를 조회합니다.</li> <li>Me.RecordsetClone.FindFirst "필드명 = " &amp; 컨트롤명 &amp; "" Me.Bookmark = Me.RecordsetClone.Bookmark</li> </ul>
OrderBy, OrderByOn	<ul style="list-style-type: none"> <li>정렬을 수행합니다.</li> <li>Asc : 오름차순(Ascending), Desc : 내림차순(Descending)</li> <li>Me.OrderBy = "컨트롤명 Asc" 혹은 "컨트롤명 Desc" Me.OrderByOn = True</li> </ul> <p>※ Asc(오름차순)는 생략 가능하나 Desc(내림차순)는 생략 불가능</p>

## 02 이벤트 프로시저 조건식 연결

문자일 때	"필드명 = " & 컨트롤명 & ""
숫자일 때	"필드명 = " & 컨트롤명
날짜일 때	"필드명 = # " & 컨트롤명 & "#"
일부 일치	"필드명 Like '*' & 컨트롤명 & '*'"

### 03 출력 프로시저

DoCmd 개체	<ul style="list-style-type: none"> <li>현재 폼에서 외부 개체에 대한 작업을 수행합니다. 생략해서 사용할 수 없습니다.</li> </ul>
OpenForm	<ul style="list-style-type: none"> <li>현재 폼에서 외부 폼을 실행합니다.</li> <li>DoCmd.OpenForm “폼 이름”, 보기형식, 필터 이름, 조건문 <ul style="list-style-type: none"> <li>※ 보기형식은 '폼 보기'의 경우 'AcNormal'이며, 기본값이므로 생략이 가능합니다.</li> <li>※ 필터 이름은 적는 경우가 거의 없으므로 생략을 위해 콤마(.)를 찍고 조건문 작성합니다.</li> </ul> </li> </ul>
OpenReport	<ul style="list-style-type: none"> <li>현재 폼에서 외부 보고서를 실행합니다.</li> <li>DoCmd.OpenReport “보고서 이름”, 보기형식, 필터 이름, 조건문 <ul style="list-style-type: none"> <li>※ 보기형식은 '인쇄 미리보기'의 경우 AcViewPreview이며, 생략은 불가능합니다.</li> <li>※ 필터 이름은 적는 경우가 거의 없으므로 생략을 위해 콤마(.)를 찍고 조건문 작성합니다.</li> </ul> </li> </ul>
Close	<ul style="list-style-type: none"> <li>현재 폼이나 보고서를 닫습니다.</li> <li>DoCmd.Close 개체 유형, "개체명", 저장 여부 <ul style="list-style-type: none"> <li>※ 개체 유형은 폼 닫기 문제가 대부분이므로 'acForm'이 일반적입니다.</li> <li>※ 저장여부는 acSaveYes(무조건 저장하며 닫기), acSaveNo(무조건 저장하지 않기), acSavePrompt(저장 여부를 묻기)로 나뉘어져 있습니다. 이 때, acSavePrompt의 경우 생략이 가능하고, 생략할 경우에는 'DoCmd.Close'만 작성해도 좋습니다. 단, acSaveYes, acSaveNo의 경우는 총 3개의 인수를 모두 작성해야 합니다.</li> </ul> </li> </ul>

## 04 메시지 상자(이벤트 프로시저)

버튼이 1개일 때	<ul style="list-style-type: none"> <li>변수 지정, 괄호 없이 MsgBox 함수 사용</li> </ul>  <p><b>MsgBox “메시지 내용”, 버튼 종류, “창 제목”</b></p> <p>※ 버튼 종류의 ‘확인’ 버튼은 명령문으로 ‘vbOKOnly’이지만 생략해서 사용 창 제목에서 ‘Microsoft Access’는 생략(엑셀에서도 동일 적용)</p>
버튼이 2개 이상일 때	<ul style="list-style-type: none"> <li>변수를 지정하고, 괄호 열고 MsgBox 함수 사용</li> <li>· ‘예’ 버튼에 기능 구현 : IF문 사용 → If a=vbYes Then ~ End if</li> </ul>  <p><b>a = MsgBox (“메시지 내용”, 버튼 종류, “창 제목”)</b></p> <p>If a = vbYes Then DoCmd.Close End If</p> <p>※ 버튼 종류 : vbYesNo(예/아니오 버튼), vbYesNoCancel(예/아니오/취소 버튼), vbOKCancel(확인/취소 버튼) 기본 버튼 선택(포커스) : vbDefaultBotton1(첫째 버튼 선택), vbDefaultBotton1(둘째 버튼 선택) 그림 삽입 : vbCritical (금지 'X'), vbInformation (정보 'i'), vbQuestion (질의 '?') vbExclamation (경고 '!')</p>

## 06 기타 프로시저

GoToRecord	<ul style="list-style-type: none"> <li>DoCmd.GotoRecord 개체 유형, 개체 이름, 이동 위치</li> <li>특정 레코드로 이동한다.</li> <li>개체 유형은 폼 작업에서는 acDataForm이 일반적이다.</li> <li>이동 위치는 acFirst(첫번째), acLast(마지막), acNext(다음), acPrevious(이전), acGoTo(특정 레코드), acNewRec(새 레코드)</li> </ul>
SetFocus	<ul style="list-style-type: none"> <li>(컨트롤명).SetFocus</li> <li>특정 컨트롤로 포커스를 위치시킵니다.</li> </ul>
Requery	<ul style="list-style-type: none"> <li>Me.Requery (혹은 DoCmd.Requery)</li> <li>폼의 데이터를 다시 표시합니다.(새로고침)</li> </ul>

ReCalc	<ul style="list-style-type: none"> <li>• <b>Me.ReCalc (혹은 DoCmd.ReCalc)</b></li> <li>• 재계산을 수행합니다.</li> </ul>	
Call	<ul style="list-style-type: none"> <li>• <b>Call (프로시저명)</b></li> <li>• 다른 프로시저 불러옵니다.</li> </ul>	
ControlTipText	<ul style="list-style-type: none"> <li>• <b>(컨트롤명).ControlTipText = 내용</b></li> <li>• 마우스 포인터를 잠시 머물고 있으면 풍선 도움말을 표시합니다.</li> </ul>	
TabStop	<ul style="list-style-type: none"> <li>• <b>(컨트롤명).TabStop = True / False</b></li> <li>• 탭정지 기능을 수행합니다. (예 / 아니오)</li> </ul>	
IF문과 SELECT CASE문	<ul style="list-style-type: none"> <li>• If (조건식1) Then     (결과식1)</li> <li>    Elseif (조건식2) Then         (결과식2)</li> <li>    Elseif (조건식3) Then         (결과식3)</li> <li>    Else         (결과식4)</li> <li>End If</li> </ul>	<ul style="list-style-type: none"> <li>• Select Case (공통조건식)</li> <li>    Case is (조건1)         (결과식1)</li> <li>    Case is (조건2)         (결과식2)</li> <li>    Case Else         (결과식3)</li> <li>End Select</li> </ul>

## 06 기타 프로시저

**SQL문** : 데이터베이스에 접근할 수 있는 질의 언어. 테이블이나 쿼리 의 데이터를 검색하고 수정, 삭제, 삽입하는 기능을 가지고 있습니다. VBA에서 SQL문 실행시, **DoCmd.RunSQL**로 실행합니다.

검색	<ul style="list-style-type: none"> <li>• <b>SELECT 필드명 FROM 테이블/쿼리명 WHERE 조건문</b></li> <li>• 검색 기능을 가지고 있고, 주로 RecordSource나 Not in에서 활용</li> </ul>
삭제	<ul style="list-style-type: none"> <li>• <b>DELETE 필드명 FROM 테이블/쿼리명 WHERE 조건문</b></li> <li>• 테이블이나 쿼리에서 조건에 맞도록 레코드 전체를 삭제합니다.</li> </ul>
갱신	<ul style="list-style-type: none"> <li>• <b>UPDATE 테이블/쿼리명 SET 변경내용 WHERE 조건문</b></li> <li>• 테이블이나 쿼리에서 조건에 따라 데이터 값을 갱신(수정)합니다.</li> </ul>
삽입	<ul style="list-style-type: none"> <li>• <b>INSERT INTO 테이블(필드1, 필드2, ...) VALUES (값1, 값2, ...)</b></li> <li>• 폼 내부의 컨트롤 값(값1, 값2, ...)을 특정 테이블 내의 필드(필드1, 필드2, ...)로 순차적으로 삽입합니다.</li> </ul>