

Decision Cube 와 쿼리 리포트의 활용

앞장에서도 간단히 언급한 바 있지만, 델파이 3 부터 기본 컴포넌트 세트에 채용된 Decision Cube 와 쿼리 리포트는 뛰어난 성능과 다양한 기능으로 데이터베이스 어플리케이션의 기능을 한단계 상승시키는 역할을 하고 있다.

Decision Cube 는 의사결정 도구의 결정판이다. 이 도구를 사용하면 드릴-다운(Drilled down)기법을 이용하여 각종 데이터를 조회하여 볼 수 있다. 그리고, 쿼리 리포트는 처음 시작한 쉐어웨어 VCL 컴포넌트로 출발하였지만, 그 성능과 편리함을 인정 받아 기존의 리포트 스피어를 밀어내고 델파이의 기본 출력 컴포넌트로서의 입지를 확실히 하였다.

Decision Cube 의 활용

Decision Cube 를 사용하는 방법은 아주 간단하지만, 활용되는 예가 드물기 때문에 여기에서는 간단한 예제를 이용하여 그 사용법을 알아보도록 한다.

Decision Cube 에 관여하는 컴포넌트는 모두 6 개로 이들의 역할은 다음과 같다.

컴포넌트	설 명
TDecisionCube	Decision Cube 의 가장 중심이 되는 컴포넌트로서, TDecisionQuery 에서 만들어진 DataSet 을 Decision 패밀리에 연결시켜 준다.
TDecisionQuery	Decision 패밀리에서 사용할 수 있도록 쿼리 DataSet 을 연결시켜 준다.
TDecisionSource	Decision Cube 에서 연결된 DataSet 을 Decision 패밀리가 사용할 수 있는 DataSource 의 역할을 수행한다.
TDecisionPivot	Decision Cube 에서 중심점이나 소트의 중심이 되는 부분을 버튼 형식이나 라디오 버튼 형식으로 나타내어, 사용자의 선택을 Decision 에 반영한다.
TDecisionGrid	Decision Cube 의 결정된 데이터를 그리드 형태로 제시한다.
TDecisionGraph	Decision Cube 의 결정된 데이터를 그래프 형태로 제시한다.

1. Decision Cube 의 구성 방법

먼저 Decision cube 를 이용하여 데이터를 보기 위해서는 TDecisionQuery 컴포넌트를 이용하여 데이터 세트를 만들어 내도록 한다. 폼에 TDecisionQuery 컴포넌트를 하나 올려 놓은 다음 기본적인 DatabaseName 프로퍼티를 DBDEMOS 로 선택하고, 오브젝트 인스펙터에서 SQL 프로퍼티 에디터를 이용하여 SQL 문장을 다음과 같이 설정한다.

```
SELECT PaymentMethod, ShipVIA, Terms, ShipDate, COUNT(AmountPaid), SUM(AmountPaid),
```

```

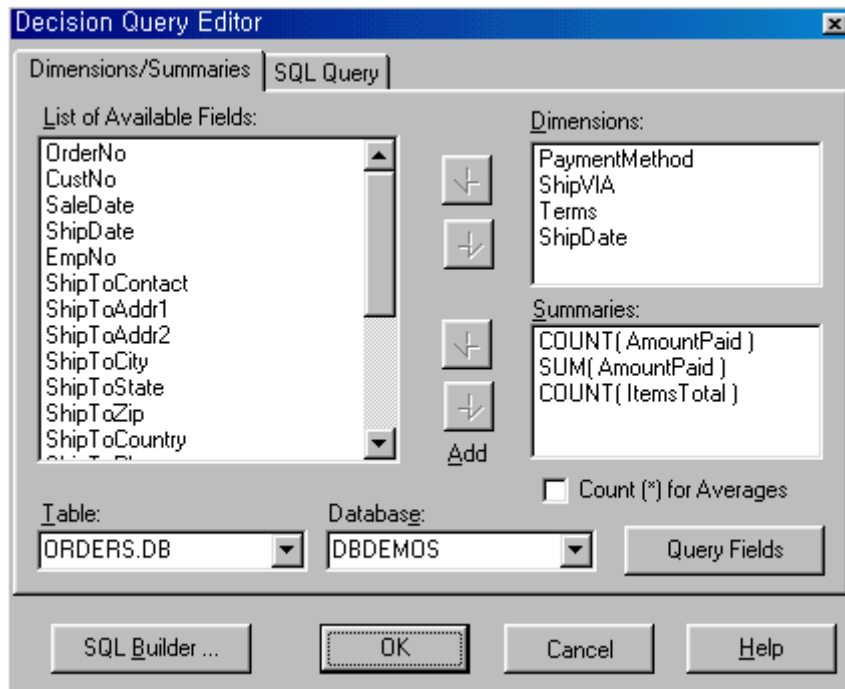
COUNT(ItemsTotal)
FROM "ORDERS.DB" ORDERS
GROUP BY PaymentMethod, ShipVIA, Terms, ShipDate

```

이 쿼리의 내용은 orders.db 파일에 있는 여러 주문 아이템 들에 대한 지불 방법, 선적 방법과 기간, 그리고 선적일과 현재의 지불 상태, 아이템의 수 등에 대한 내용을 지불 방법과 기간 및 선적일, 선적 방법 등의 내용을 참고로 하여 그룹화한 데이터 세트를 뽑아내게 된다. 이제 이 컴포넌트의 Active 프로퍼티를 True 로 하면 해당되는 데이터 세트가 만들어진다. 참고로, Decision Query 의 자세한 내용이 보고 싶으면 TDecisionQuery 컴포넌트를 더블 클릭하면 다음의 폼이 나타날 것이다.

이 화면을 보면 테이블의 내용과 화면에 보이기 위한 필드, 합계 형식으로 나타나는 필드들이 나열되며, 어떤 앨리어스에서 어떤 테이블을 사용하였는지 알 수 있다.

또한, 여기에서 직접 사용할 Dimension 과 Summary 필드를 결정하여 추가할 수도 있으며, 이 내용들이 SQL 프로퍼티에 자동으로 반영된다.



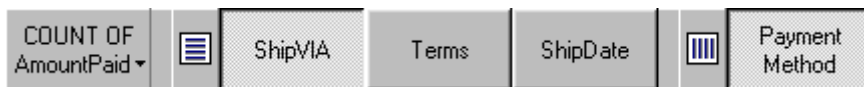
마찬가지로 SQL Query 탭을 선택해보면 앞에서 넣었던 SQL 문장이 나열되어 있으며, 여기서 바로 SQL 문장을 수정할 수도 있다. 이때 SQL 문장을 만드는 것보다 액세스 등에서 사용했던 방법과 비슷하게 비주얼한 환경에서 쿼리를 자동으로 생성하고자 한다면, SQL Builder 버튼을 클릭하여 쿼리를 생성할 수도 있다.

이제는 이렇게 만들어진 데이터 세트를 TDecisionSource 컴포넌트를 통하여 다른 Decision 패밀리에 연결하도록 한다. 이 방법은 기본적으로 TTable 과 TDataSource 를

이용하여 데이터 컨트롤에 연결하던 기존의 방식과 동일하므로 그다지 어렵지 않게 사용할 수 있을 것이다. 먼저 TDecisionCube 와 TDecisionSource 컴포넌트를 폼에 추가하고 TDecisionCube 컴포넌트의 DataSet 프로퍼티를 DecisionQuery1, TDecisionSource 컴포넌트의 DecisionCube 프로퍼티를 DecisionCube1 으로 설정한다.

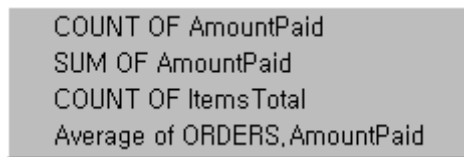
이렇게 연결된 TDecisionSource 컴포넌트를 통하여 TDecisionPivot 과 TDecisionGrid 컴포넌트를 연결하도록 하자. 이때에 Pivot 은 DecisionGrid 에 제시되는 데이터를 조절하는 역할을 한다.

아마도 Pivot 의 내용이 다음과 같이 나타날 것이다.



이 내용을 잘 살펴보면, 앞의 Decision Query Editor 대화 상자에서 Dimensions 와 Summaries 에 나열된 내용으로 이루어져 있다는 것을 알 수 있을 것이다.

Dimensions 에 선택된 필드는 Pivot 에 버튼의 형태로 나타나서 그룹을 동적으로 지정할 수 있게 되며, Summaries 에 지정된 조건들은 다음과 같이 또 다른 형태의 선택을 할 수 있도록 주어진다.



여기에서, 마지막에 나열된 Average of ORDERS.AmountPaid 는 Count 와 SUM 을 사용하여 자연스럽게 자동으로 추가된 내용이다.

마찬가지로 우측의 화면을 보면, ShipVIA 와 ShipDate 등의 버튼에서 해당되는 데이터를 전체 내용을 다 볼수도 있고, 드릴-인(Drill In)하여 조절하여 볼 수도 있다.

여기에서 주의해서 살펴보아야 하는 것은 창살처럼 보이는 비트맵 버튼 들이다. 앞 그림에서 ShipVIA 버튼 좌측에 있는 버튼가 같이 가로로 창살이 있는 비트맵 버튼은 가로열에 영향을 주는 필드를 이 버튼에서 선정한다는 의미이다. 반대로 세로로 창살이 있는 비트맵 버튼은 세로열에 영향을 주는 필드를 선택하는 것이다.

이러한 필드를 조절할 때에는 해당되는 버튼 위에서 마우스의 오른쪽 버튼을 눌러보면 나타나는 팝업 메뉴를 이용하는데, 이때에 Move to Column Area 를 메뉴를 선택하면 Row Area 로 이동하게 되며, 반대로 Row 에서는 Column 으로 이동하는 메뉴가 나타나게 되며 이렇게 동적으로 pivot 의 이동이 가능하다.

그리고, 해당되는 필드의 Drilled In 메뉴를 선택하면, 해당되는 필드의 내용들 중에서 공통적인 부분들만 뽑아서 선택을 할 수 있도록 도와준다. 다음 화면은 ShipDate 버튼에서

Drilled In 메뉴를 선택할 때 나타난다.

Open Dimension All Values
1988
1989
1992
1993
1994
1995

여기에서 Open Dimension 메뉴를 선택하면 처음의 해당 데이터가 모두 DecisionGrid 의 필드에 영향을 끼치며 나타나게 된다. 또한, All Values 메뉴를 선택하면 해당되는 데이터를 전부 나타낸다. 이와 같이 사용자는 마우스의 왼쪽 버튼과 오른쪽 버튼을 번갈아 사용하면, 해당되는 데이터를 자유자재로 볼 수 있다.

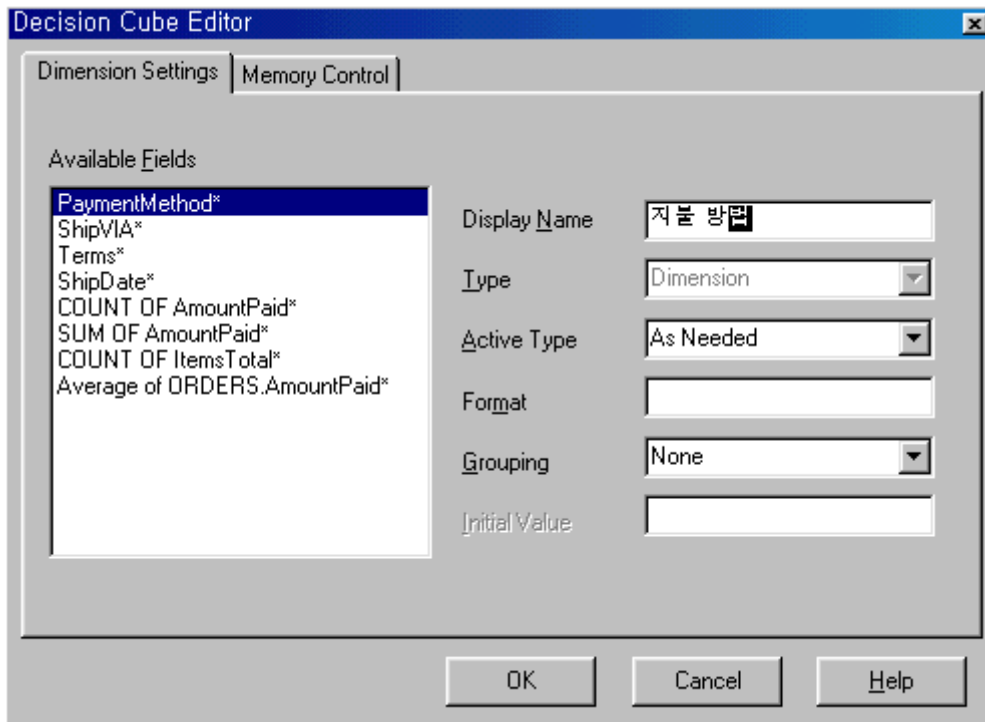
즉, Open Dimension 메뉴를 이용하면 DecisionGrid 에 해당되는 Row 와 Column 에 필드에 해당되는 그룹 형태로 나타나며, Drilled In 메뉴를 선택하면 DecisionGrid 에 나타나는 데이터에 필터를 사용한 것처럼 데이터를 걸러낼 수 있다.

TDecisionGrid 컴포넌트에서 특이한 점은 그리드의 기능에서 특별하게 확장된 부분은 없고, 다만 그리드 내에서 Dimension 의 subtotal(노란색 +, - 그룹)부분을 토글할 수 있어서, 해당되는 내용의 중간합계를 자유자재로 볼 수 있다는 점이 다르다.

마지막으로, 이렇게 표로 나타낼 수 있는 데이터를 그래프로 동적으로 표현하기 위해서는 TDecisionGraph 컴포넌트를 이용한다.

이와 같이 Decision Cube 를 이용하면 복잡한 데이터를 동적으로 사용자가 걸러서 볼 수 있는 훌륭한 인터페이스를 제공할 수 있다. 실제 OA 응용 프로그램을 작성할 때에 사용자로 하여금 다양한 관점에서 데이터를 바라다 볼 수 있는 인터페이스를 제공해 준다.

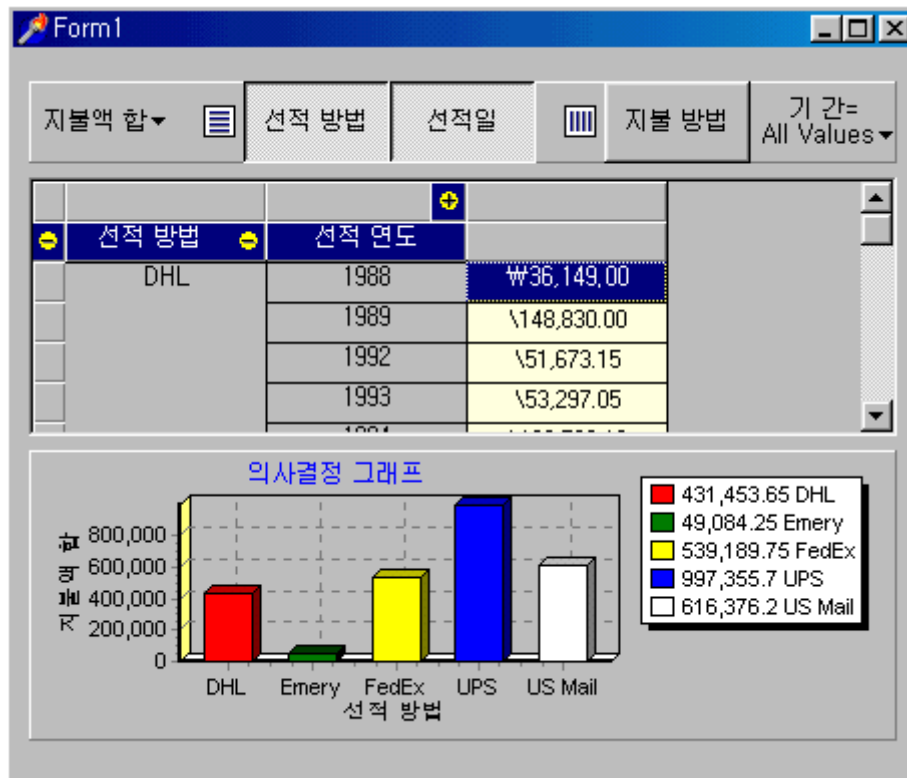
참고로, 필드 이름과 다르게 보기 좋은 이름을 사용하려면 TDecisionCube 컴포넌트의 DimensionMap 프로퍼티의 프로퍼티 에디터를 이용하여 편집하면 된다. 다음 그림은 이를 편집하는 대화 상자이다.



항간에는 이 Decision Cube 의 기능이 느리다고 하는 VB 사용자가 있지만, 실제 VB 에서 지원되는 것은 크리스탈 리포트에서 지원이 되는 부분이며, 사용자가 크리스탈 리포트를 사용해야 한다는 불편함이 있고, 또한 VB 프로그램에서 크리스탈 리포트에게 이러한 값들을 전달하여야 하는 불편함이 있다.

하지만 델파이를 이용하면 간편하게 프로그램에 포함시킬 수 있으며, 이러한 프로그램도 사용자가 확장하거나 조절할 수 있게 프로그래밍할 수 있는 매우 뛰어난 컴포넌트이다.

이 예제 프로그램의 실행 화면은 다음과 같다.



퀵 리포트(Quick Report)의 활용

퀵 리포트는 native VCL 이기 때문에 추가적인 라이브러리나 ocx 파일을 제공하지 않아도 되고, 그 리포팅 기능 자체도 상당히 강력하다. 이를 이용하여 간단한 한 페이지 문서부터 다양한 형태의 그래프를 출력하는 복잡한 리포트까지 만들어낼 수 있다.

퀵 리포트를 이용하여 출력 폼을 디자인하고, 이를 출력하는 코드는 다음과 같이 간단하다.

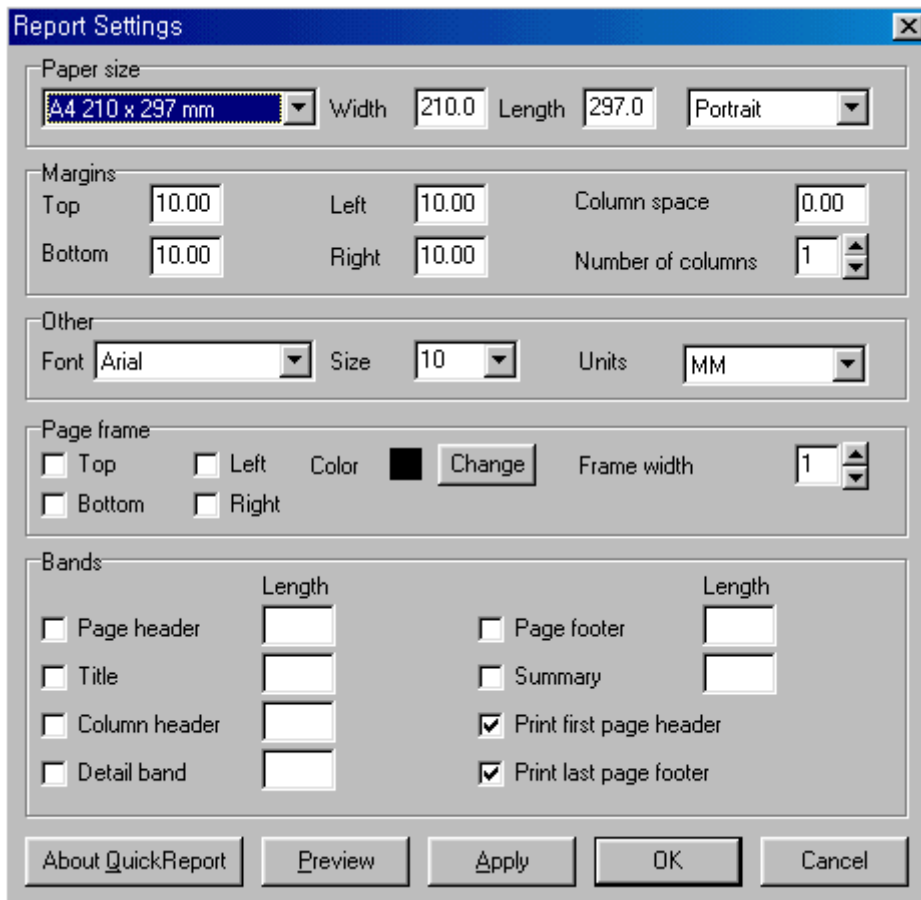
```
폼.QuickRep.Print;
```

```
폼.QuickRep.Preview; //미리보기
```

그러면, 퀵 리포트의 주요 컴포넌트의 기능과 사용 방법에 대해서 간단히 알아보도록 하자.

1. TQuickRep 컴포넌트

이 컴포넌트는 퀵 리포트의 가장 기본적인 컴포넌트로 델파이의 폼을 리포트로 전환하는 역할을 한다. 오른쪽 버튼을 클릭하고 팝업 메뉴에서 Report Setting 을 선택하면 다음과 같이 리포트 용지에 대한 여러가지 기본 설정을 할 수 있다.



참고로 콰 리포트는 도트 프린터를 사용하거나 A4 이외의 종이를 사용할 때에는 오동작하는 경우가 많기 때문에, 일단 해당 기종에서 테스트를 해서 확인을 하고 출력이 정상적으로 나올 때 사용하는 것이 좋다.

콰 리포트에서 자주 사용되는 프로퍼티는 다음과 같다.

프로퍼티	내 용
PageCount	출력되는 전체 페이지수
PageNumber	현재 출력되는 페이지번호
RecordNo	사용중인 레코드 번호
PageFrame	페이지의 여백
PaperSize	용지의 크기
ReportTitle	리포트의 제목
Orientation	출력방향 poPortait(세로), poLandscape(가로)
ShowProgress	출력진행상태 Progreee 표시유무
PaperWidth	용지의 폭
PaperLength	용지의 길이

Columns	컬럼의 수
---------	-------

이외에 Band 의 주요 프로퍼티는 다음과 같다.

프로퍼티	내 용
HasColumnHeader	각페이지에 Column 부분을 사용한다
HasDetail	테이블의 정보를 모두 출력
HasPageFooter	아래 출력내용 설정
HasPageHeader	위 출력내용 설정
HasSummery	리포트 끝에 한번만 출력
HasTime	보고서 시작부에 한번만 출력

2. TQRBand 컴포넌트

TQRBand 컴포넌트는 킷 리포트에서 가장 많이 사용되는 밴드로 BendType 프로퍼티의 값을 조절함으로써 출력되는 곳의 위치를 지정할 수 있도록 한다.

주요 프로퍼티의 내용은 다음과 같다.

프로퍼티	내용
ForceNewColumn	출력 전에 새로운 열을 만든다
ForceNewPage	출력 전에 새로운 페이지를 만든다
LinkBand	여러 개의 밴드를 페이지에 출력 시에 일정한 여백을 갖도록 하고, 공간이 충분하면 링크된 밴드가 이어서 출력되게 한다. 단, 공간이 부족하면 새로운 라인에 출력한다.
HasChild	True 시에 자동으로 Child 가 나타난다.
Frame	테두리의 사각형의 선을 그릴 것인지 결정한다.

BandType 프로퍼티는 다음과 같은 값을 가질 수 있다.

값	내 용
rbChild	다른 밴드가 연결되어 있으면 연결된 밴드를 먼저 출력하고 나중에 출력
rbColumnHeader	각 페이지의 윗 부분에 나오게 한다
rbDetail	테이블 정보가 모두 출력
rbGroupFooter	테이블 정보 맨 끝에 한번 출력
rbGroupHeader	테이블 정보 맨 처음 한번 출력

rbOverlay	페이지 왼쪽 상단부터 출력
rbPageFooter	페이지 아래 출력
rbPageHeader	페이지 위에 출력
rbSubDetail	master/detail 부분의 detail 부분에 사용
rbSummary	맨 마지막에 한 번 합계 낼 경우에 사용
rbTitle	시작부분에 한 번 수행

3. TQRSysData 컴포넌트

리포트 출력 시에 시간, 날짜 등의 내용을 출력하는 컴포넌트이다. Data 프로퍼티를 조절하면 다양한 속성을 이용할 수 있다. Data 프로퍼티에 사용할 수 있는 값에는 다음과 같은 것들이 있다.

값	내 용
qrsDate	시스템 날짜
qrsDateTime	시스템 날짜 + 시간 출력
qrsDetailNo	현재의 레코드 번호
qrsPageNumber	현재 출력 페이지 번호
qrsDetailCount	출력될 총 레코드 수
qrsReportTitle	리포트의 제목으로 TQuickRep 컴포넌트의 ReportTitle 프로퍼티 사용
qrsTime	출력 시에 시간 출력

Text 에 들어 있는 내용을 Data 에서 정의된 내용을 출력하기 전에 먼저 찍고, Data 프로퍼티에서 설정된 내용을 출력한다.

4. TQRGroup 컴포넌트

리포트에서 마스터/디테일 형태의 리포트를 만드는 경우에 사용한다. 이 컴포넌트에 보면 FooterBand, HasChild, Master 등의 프로퍼티가 있는 것을 알 수 있을 것이다. 이때에 Expression 의 프로퍼티를 이용하면, 원하는 DataSet 의 필드에서 특정 함수를 거쳐서 값을 산출할 수 있다.

5. TQRDB, TQRLabel 컴포넌트

TQRLabel 은 출력되는 쿼리 리포트에 원하는 내용을 입력할 수 있는 라벨이라고 생각하면 된다. TQRDB 는 DBLabel 과 거의 동일하다, 해당 밴드 위에 올려놓고 DataSet 의 Fields

를 연결하면 해당 내용을 보여준다.

6. TQRPreview 컴포넌트

퀵 리포트는 기본적인 Preview 화면을 지원한다, 하지만 영문으로 나오고 모양이 그리 이쁘지 않다. 이 컴포넌트를 이용하면 이러한 Preview 화면을 조정할 수 있다. 만들어진 Preview 화면을 사용하려면 퀵 리포트에 등록하여 사용할 수 있다.

이 밖에도 많은 수의 퀵 리포트 컴포넌트가 제공되고 있다. 퀵 리포트는 비교적 강력하면서도 효율적으로 사용할 수 있는 우수한 컴포넌트 세트이기 때문에, 유용성이 매우 높다. 하지만, 비교적 버그가 많은 편이어서 한번 그 버그를 경험한 개발자 들이 등을 돌리는 경우도 매우 많은 편이다.

그렇지만, 그 효율성과 장점 역시 탁월하므로 앞에서도 언급했듯이 간단한 종류의 리포트를 작성할 때에는 이를 사용하는 것이 여러모로 유리하다.

정 리 (Summary)

이번 장에서는 델파이에서 사용되는 우수한 컴포넌트 세트인 Decision Cube 와 퀵 리포트에 대해서 알아보았다. 이들은 비록 어플리케이션을 작성하는데 있어서 핵심적인 부분을 차지하지는 않는다. 그렇지만, 보기 좋은 떡이 먹기도 좋다는 말이 있듯이 데이터베이스 어플리케이션을 개발할 때에도 출력 화면과 인쇄물을 어떻게 생성해낼 수 있는지 여부는 가장 중요한 어플리케이션의 평가 요소 중의 하나이다.

그러므로, 이들을 효과적으로 사용하면 데이터베이스 어플리케이션의 완성도를 높일 수 있으므로 사용법을 꾸준히 익혀두는 것이 바람직하다.