

자습서 - 차트 표현식

Qlik Sense®

November 2023

Copyright © 1993-2023 QlikTech International AB. 무단 전재 및 복제를 금합니다.



1 자습서 시작!	4
1.1 학습 내용	4
1.2 이 자습서의 대상	4
1.3 이 자습서의 단원	4
1.4 추가 자료 및 리소스	4
2 시각화에서 표현식 사용	5
2.1 표현식이란	5
2.2 표현식을 사용할 수 있는 경우	5
2.3 표현식은 언제 평가됩니까?	5
3 집계 함수란 무엇입니까?	6
3.1 Sum()을 사용하여 금액 통합	6
3.2 Max()를 사용하여 가장 높은 판매 값 계산	7
3.3 Min()을 사용하여 가장 낮은 판매 값 계산	8
3.4 Count()를 사용하여 엔터티 수 계산	8
Count()와 Count(distinct)의 차이점	9
4 중첩 집계	11
4.1 항상 한 수준의 집계 함수	11
4.2 중첩 집계를 위해 Aggr() 사용	11
4.3 최대 평균 주문 값 계산	12
5 네이키드 필드 참조	15
5.1 항상 표현식에 집계 함수를 사용하십시오.	15
If() 함수를 사용하여 송장 날짜 분할	15
5.2 네이키드 필드 참조 방지	16
If() 함수에서 네이키드 필드 참조 방지	16
6 The importance of Only()	19
6.1 Only()를 사용한 서로 다른 표현식	20
7 실생활의 예	25
7.1 총 수익 백분율 계산	25
7.2 송장 발행 지연	27
7.3 감사합니다.	31

1 자습서 시작!

이 자습서는 Qlik Sense의 차트 표현식을 소개합니다. 표현식은 데이터를 처리하고 시각화에서 볼 수 있는 결과를 생성하는 데 사용되는 함수, 필드 및 수학 연산자의 조합입니다.

차트 표현식은 주로 측정값에 사용됩니다. 또한 제목, 부제, 각주와 더불어 차원에 대한 표현식을 사용하여 더 동적이고 강력한 시각화를 만들 수도 있습니다.

1.1 학습 내용

자습서를 완료하고 나면 시각화에서 표현식을 사용하는 데 익숙해집니다.

1.2 이 자습서의 대상

Qlik Sense 기본 사항에 익숙해야 합니다. 예를 들어 다양한 시트에서 데이터를 로드하고 앱을 만들고 시각화를 만들 수 있어야 합니다.

데이터 로드 편집기에 액세스하고 Qlik Sense Enterprise on Windows에서 데이터를 로드할 수 있어야 합니다.

1.3 이 자습서의 단원

이 자습서의 항목은 임의의 순서로 완료할 수 있습니다. 그러나 이후 항목은 사용자가 이전 항목에 익숙하다고 가정합니다. 스크린샷은 Qlik Sense Enterprise SaaS에서 만든 것입니다. 다른 배포에서 Qlik Sense Enterprise를 사용하는 경우 약간의 차이가 발생할 수 있습니다.

1.4 추가 자료 및 리소스

- [Qlik](#)에서는 보다 자세한 정보를 알아볼 수 있도록 다양한 리소스를 제공합니다.
- [Qlik 온라인 도움말](#)을 사용할 수 있습니다.
- [Qlik Continuous Classroom](#)에서 교육(무료 온라인 과정 포함)이 제공됩니다.
- [Qlik Community](#)에서 토론 포럼, 블로그 등을 찾을 수 있습니다.

2 시각화에서 표현식 사용

Qlik Sense에서 시각화는 차트로 구성되며, 각 차트는 차원과 측정값으로 구성됩니다. 표현식을 사용하여 시각화를 보다 동적이고 복잡하게 만들 수 있습니다.

시각화에는 정보 전달을 돕기 위해 제목, 부제, 각주 및 기타 요소가 포함될 수 있습니다. 시각화를 구성하는 요소는 간단할 수 있습니다. 예를 들어, 차원은 데이터를 나타내는 필드로 구성되며 제목은 텍스트로 구성됩니다.

측정값은 필드를 기준으로 계산됩니다. 예: **Sum(Cost)**를 사용하면 **Cost** 필드의 모든 값이 **Sum** 함수를 사용하여 집계됩니다. 달리 말하자면, **Sum(Cost)**은 표현식입니다.

2.1 표현식이란

표현식은 함수, 필드, 수학 연산자(+ * / =)의 조합입니다. 표현식은 시각화에 표시할 수 있는 결과를 생산하기 위해 앱에 있는 데이터를 처리하는 데 사용됩니다. 기본 계산만 사용하여 간단하게 만들거나, 함수 필드 및 연산자를 사용하여 복잡하게 만들 수도 있습니다. 표현식은 스크립트와 차트 시각화에서 모두 사용됩니다.

모든 측정값은 표현식입니다. 측정값과 표현식의 차이는 표현식의 경우 이름이나 상세 데이터가 없다는 점입니다.

차원, 제목, 부제, 각주와 더불어 차원에 대한 표현식을 사용하여 더 동적이고 강력한 시각화를 만들 수 있습니다. 예를 들어, 이는 시각화의 제목이 정적 텍스트가 아닌, 선택 내용에 따라 결과가 변경되는 표현식에서 생성될 수 있음을 의미합니다.

2.2 표현식을 사용할 수 있는 경우

시각화를 편집할 때 속성 패널에 **fx** 기호가 있는 경우 표현식을 사용할 수 있습니다. **fx**를 클릭하면 표현식을 작성하고 편집할 수 있도록 설계된 식 편집기가 열립니다. 표현식 필드에 표현식을 직접 입력할 수도 있습니다.

표현식은 마스터 항목으로 직접 저장할 수 없습니다. 그러나 마스터 측정값 및 마스터 차원에는 표현식이 포함될 수 있습니다. 측정값 또는 차원에서 표현식을 사용하여 마스터 항목으로 저장하면 측정값 또는 차원의 표현식이 유지됩니다.

2.3 표현식은 언제 평가됩니까?

로드 스크립트에서 스크립트가 실행될 때 표현식이 평가됩니다. 시각화에서 표현식은 해당 표현식에 포함된 필드, 변수 또는 함수가 값 또는 논리 상태를 변경할 때마다 자동으로 평가됩니다. 구문과 사용 가능한 함수 측면에서 스크립트 표현식과 차트 표현식 사이에 몇 가지 차이점이 있습니다.

3 집계 함수란 무엇입니까?

집계 함수는 다대일 함수입니다. 많은 레코드의 값을 입력으로 사용하고 모든 레코드를 요약하는 단일 값으로 축소합니다. Sum(), Count(), Avg(), Min(), 및 Only()는 모두 집계 함수입니다.

Qlik Sense에서 대부분의 수식에는 정확히 한 수준의 집계 함수가 필요합니다. 여기에는 차트 표현식, 텍스트 상자 및 레이블이 포함됩니다. 표현식에 집계 함수를 포함하지 않으면 Qlik Sense가 자동으로 Only() 함수를 지정합니다.

- 집계 함수는 데이터에서 여러 레코드의 일부 속성을 설명하는 단일 값을 반환하는 함수입니다.
- 계산 차원을 제외한 모든 표현식은 집계로 평가됩니다.
- 표현식의 모든 필드 참조는 집계 함수로 래핑해야 합니다.



식 편집기를 사용하여 Qlik Sense에서 표현식을 만들고 변경할 수 있습니다.

3.1 Sum()을 사용하여 금액 통합

Sum()은 집계 데이터 전체에서 필드 또는 표현식을 통해 지정된 값의 합계를 계산합니다.

각 관리자의 총 판매액과 모든 관리자의 총 판매액을 계산해 보겠습니다.

앱 내의 *Which Aggregations?* 시트에서 두 개의 테이블(제목 Sum(), Max(), Min() 및 Count())을 찾습니다. 각 테이블을 사용하여 집계 함수를 만듭니다.

다음과 같이 하십시오.

1. 사용 가능한 Sum(), Max(), Min() 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. Sum(Sales)
5. **적용**을 클릭합니다.

관리자별로 총 판매액을 보여주는 테이블

Sum(), Max (), Min()	
Manager	Sum(Sales)
Totals	\$ 104,852,674.81
Dennis Johnson	\$ 15,945,030.85
Stewart Wind	\$ 15,422,448.79
Carolyn Halmon	\$ 11,363,424.41
John Greg	\$ 9,770,909.24
Samantha Allen	\$ 7,540,947.33
Amanda Honda	\$ 6,436,630.86
Brenda Gibson	\$ 6,215,872.87
Kathy Clinton	\$ 5,154,950.48
Molly McKenzie	\$ 5,079,387.55
John Davis	\$ 4,060,007.10

각 관리자의 판매액과 모든 관리자의 총 판매액을 볼 수 있습니다.



데이터 형식이 올바른지 확인하는 것이 가장 좋습니다. 이 경우 숫자 서식을 화폐로 설정하고 서식 패턴을 \$ #,##0;- \$ #,##0으로 설정합니다.

3.2 Max()를 사용하여 가장 높은 판매 값 계산

Max()는 집계된 데이터의 행별로 가장 높은 값을 찾습니다.

다음과 같이 하십시오.

1. 열 추가를 클릭하고 **측정값**을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. **Max (Sales)**
4. **적용**을 클릭합니다.

관리자별로 총 판매액과 가장 높은 판매액을 보여주는 테이블

Sum(), Max (), Min()		
Manager	Sum(Sales)	Max(Sales)
Totals	\$ 104,852,674.81	\$ 555,376.00
Dennis Johnson	\$ 15,945,030.85	\$ 285,350.40
Stewart Wind	\$ 15,422,448.79	\$ 258,946.70
Carolyn Halmon	\$ 11,363,424.41	\$ 555,376.00
John Greg	\$ 9,770,909.24	\$ 310,156.07
Samantha Allen	\$ 7,540,947.33	\$ 52,469.65
Amanda Honda	\$ 6,436,630.86	\$ 133,568.68
Brenda Gibson	\$ 6,215,872.87	\$ 119,030.00
Kathy Clinton	\$ 5,154,950.48	\$ 47,326.42
Molly McKenzie	\$ 5,079,387.55	\$ 79,134.97
John Davis	\$ 4,060,007.10	\$ 118,210.17

각 관리자의 가장 높은 판매액과 가장 높은 총 판매액을 볼 수 있습니다.


3.3 Min()을 사용하여 가장 낮은 판매 값 계산

Min()은 집계된 데이터에서 행별로 가장 낮은 값을 찾습니다.

다음과 같이 하십시오.

1. 열 추가를 클릭하고 **측정값**을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. *Min (Sales)*
4. **적용**을 클릭합니다.

관리자별로 총 판매액, 가장 높은 판매액 및 가장 낮은 판매액을 보여주는 테이블

Sum(), Max (), Min()				
Manager		Sum(Sales)	Max(Sales)	Min(Sales)
Totals		\$ 104,852,674.81	\$ 555,376.00	-\$ 27,929.88
Dennis Johnson		\$ 15,945,030.85	\$ 285,350.40	-\$ 27,929.88
Stewart Wind		\$ 15,422,448.79	\$ 258,946.70	-\$ 1,687.63
Carolyn Halmon		\$ 11,363,424.41	\$ 555,376.00	-\$ 13,749.60
John Greg		\$ 9,770,909.24	\$ 310,156.07	-\$ 17,883.07
Samantha Allen		\$ 7,540,947.33	\$ 52,469.65	-\$ 1,687.91
Amanda Honda		\$ 6,436,630.86	\$ 133,568.68	-\$ 15,122.77
Brenda Gibson		\$ 6,215,872.87	\$ 119,030.00	-\$ 11,903.00
Kathy Clinton		\$ 5,154,950.48	\$ 47,326.42	-\$ 3,418.90
Molly McKenzie		\$ 5,079,387.55	\$ 79,134.97	-\$ 1,631.49
John Davis		\$ 4,060,007.40	\$ 110,240.47	\$ 12,770.70

각 관리자의 가장 낮은 판매액과 가장 낮은 총 판매액을 볼 수 있습니다.

3.4 Count()를 사용하여 엔터티 수 계산

Count()는 각 차트 차원의 값, 텍스트 및 숫자의 수를 계수하는 데 사용됩니다.

데이터에서 각 관리자는 많은 영업 담당자(*Sales Rep Name*)를 관리합니다. 영업 담당자 수를 계산해 보겠습니다.

다음과 같이 하십시오.

1. 사용 가능한 Count() 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. *Count([Sales Rep Name])*
5. **적용**을 클릭합니다.

영업 담당자 및 총 영업 담당자 수를 보여주는 테이블

Count()	
Sales Rep Name	Count([Sales Rep Name])
Totals	64
Amalia Craig	1
Amanda Honda	1
Carl Lynch	1
Molly McKenzie	1
Sheila Hein	1
Brenda Gibson	1
Dennis Johnson	1
Ken Roberts	1
Robert Kim	1
William Fisher	1
Cary Frank	1
Edward Smith	1
Lee Chin	1
Ronald Milam	1

총 영업 담당자 수가 64임을 알 수 있습니다.

Count()와 Count(distinct)의 차이점

관리자 수를 계산해 보겠습니다.

다음과 같이 하십시오.

1. 테이블에 새 차원을 추가합니다. *Manager*.
단일 관리자가 둘 이상의 영업 담당자를 관리하므로 동일한 관리자 이름이 테이블에 두 번 이상 나타납니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. *Count(Manager)*
5. 표현식을 사용하여 다른 측정값을 추가합니다. *Count(distinct Manager)*
6. **적용**을 클릭합니다.

영업 담당자, 총 영업 담당자 수, 각 영업 담당자를 관리하는 관리자, 잘못된 총 관리자 수 및 올바른 총 관리자 수를 보여주는 테이블.

3 집계 함수란 무엇입니까?

Count()				
Sales Rep Name	Count([Sales Rep Name])	Manager	Count(Manager)	Count(distinct Manager)
Totals	64		64	18
Amalia Craig	1	Amanda Honda	1	1
Amanda Honda	1	Amanda Honda	1	1
Carl Lynch	1	Amanda Honda	1	1
Molly McKenzie	1	Amanda Honda	1	1
Sheila Hein	1	Amanda Honda	1	1
Brenda Gibson	1	Brenda Gibson	1	1
Dennis Johnson	1	Brenda Gibson	1	1
Ken Roberts	1	Brenda Gibson	1	1
Robert Kim	1	Brenda Gibson	1	1
William Fisher	1	Brenda Gibson	1	1
Cary Frank	1	Carolyn Halmon	1	1
Edward Smith	1	Carolyn Halmon	1	1
Lee Chin	1	Carolyn Halmon	1	1
Ronald Milam	1	Carolyn Halmon	1	1

*Count(Manager)*를 식으로 사용하여 열에서 총 관리자 수가 64로 계산된 것을 볼 수 있습니다. 잘못된 계산입니다. *Count(distinct Manager)* 식을 사용하여 총 관리자 수가 18로 올바르게 계산됩니다. 각 관리자는 이름이 목록에 나타나는 횟수에 관계없이 한 번만 계산됩니다.

4 중첩 집계

차트 표현식에 포함되는 모든 필드 이름은 정확히 하나의 집계 함수로 둘러싸야 합니다. 집계를 중첩해야 하는 경우 **Aggr()**을 사용하여 두 번째 집계 수준을 추가할 수 있습니다. **Aggr()**은 집계 함수를 인수로 포함합니다.

4.1 항상 한 수준의 집계 함수

일반적인 앱에는 다음이 포함될 수 있습니다.

- 데이터에 100만 개의 레코드
- 피벗 테이블에 100개의 행
- 게이지 또는 텍스트 상자에 단일 KPI

크기는 다르지만 세 숫자 모두 여전히 모든 데이터를 나타낼 수 있습니다. 숫자는 서로 다른 집계 수준입니다.

집계 함수는 많은 레코드의 값을 입력으로 사용하고 모든 레코드의 요약으로 볼 수 있는 단일 값으로 축소합니다. 한 가지 제한 사항이 있습니다. 집계 함수는 다른 집계 함수 내에서 사용할 수 없습니다. 일반적으로 모든 필드 참조를 하나의 집계 함수로 정확히 래핑해야 합니다.

다음 표현식이 작동합니다.

- **Sum(Sales)**
- **Sum(Sales) / Count(Order Number)**

다음 표현식은 중첩 집계이므로 작동하지 않습니다.

- **Count(Sum(Sales))**

이에 대한 해결책은 **Aggr()** 함수의 형태로 제공됩니다. 해당 이름과 달리 집계 함수가 아닙니다. 수학의 행렬과 같은 "다대다" 함수입니다. N 레코드가 있는 테이블을 M 레코드가 있는 테이블로 변환합니다. 값으로 구성된 배열을 반환합니다. 하나의 측정값과 하나 이상의 차원이 있는 가상 일반표로 간주될 수도 있습니다.



Aggr() 함수를 계산된 차원에서 사용하여 다양한 수준의 중첩 차트 집계를 생성할 수 있습니다.

4.2 중첩 집계를 위해 **Aggr()** 사용

Aggr()은 지정된 하나 또는 여러 차원에서 계산된 표현식에 대한 값의 배열을 반환합니다. 판매량, 고객 및 지역별 최대값을 예로 들 수 있습니다. 고급 집계에서 **Aggr()** 함수는 중첩된 집계에 대한 입력으로 **Aggr()** 함수의 결과로 구성된 배열을 사용하는 다른 집계 함수 안에 포함될 수 있습니다.

이 함수를 사용하면 **Aggr()** 문은 하나 이상의 차원으로 그룹화된 하나의 표현식으로 가상 테이블을 생성합니다. 그런 다음 이 가상 테이블의 결과는 외부 집계 함수가 더 집계할 수 있습니다.

4.3 최대 평균 주문 값 계산

차트 표현식에 간단한 **Aggr()** 문을 사용하겠습니다.

지역 수준에서 전체 척도뿐만 아니라 두 가지 더 복잡한 표현식도 표시하고 싶습니다.

- 각 지역 내 관리자별 가장 큰 평균 주문 값.
- 가장 큰 평균 주문 값을 담당하는 관리자.

표준 표현식 **Sum(Sales)/Count([Order Number])**를 사용하여 각 지역의 평균 주문 값을 쉽게 계산할 수 있습니다.

앱 내의 *Nested Aggregations* 시트에서 제목이 *Aggr() function*인 테이블을 찾습니다.

다음과 같이 하십시오.

1. 사용 가능한 *Aggr()* function 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. *Sum(Sales)/Count([Order Number])*
5. **적용**을 클릭합니다.

지역별 평균 주문 값을 보여주는 테이블.

Aggr() function	
Region	Average order value
Totals	\$ 1,887
Germany	\$ 405
Japan	\$ 604
Nordic	\$ 641
Spain	\$ 577
UK	\$ 1,390
USA	\$ 1,821



데이터 형식이 올바른지 확인하는 것이 가장 좋습니다. 이 경우 각 열에서 계산을 나타내도록 **레이블**을 변경합니다. 통화 값이 있는 열에서 **숫자 서식**을 **화폐**로 변경하고 **서식 패턴**을 \$ #,##0;- \$ #,##0으로 변경합니다.

목표는 각 지역의 가장 큰 평균 주문 값을 검색하는 것입니다. Qlik Sense에서 관리자별로 각 지역의 평균 주문 값을 가져와서 가장 큰 값을 표시하도록 **Aggr()**을 사용해야 합니다. 관리자별로 각 지역의 평균 주문 값을 가져오려면 **Aggr()** 문에 다음 차원을 포함해야 합니다.

Aggr (Sum (Sales) /Count ([Order Number]), Region, Manager)

이 표현식으로 인해 Qlik Sense는 다음과 같은 가상 테이블을 생성합니다.

관리자별로 각 지역의 평균 주문 값을 보여주는 **Aggr()** 함수의 가상 테이블.

Virtual table of Aggr() function

Region	Manager	Average order value
Totals		-
Germany	Micheal Williams	\$ 3,506
Germany	Dennis Johnson	\$ 1,380
Germany	Molly McKenzie	\$ 820
Germany	David Laychak	\$ 624
Germany	John Davis	\$ 456
Germany	Sheila Hein	\$ 445
Germany	Amanda Honda	\$ 443
Germany	John Greg	\$ 436
Germany	Samantha Allen	\$ 404
Germany	Stewart Wind	\$ 393
Germany	William Fisher	\$ 380
Germany	Ken Roberts	\$ 379
Germany	Kathy Clinton	\$ 335
Germany	Odessa Morris	\$ 331

Qlik Sense가 관리자별로 각 지역의 개별 평균 주문 값을 계산할 때 이러한 값 중 가장 큰 값을 찾아야 합니다. **Aggr()** 함수를 **Max()**로 래핑하여 이 작업을 수행합니다.

Max (Aggr (Sum (Sales) /Count ([Order Number]), Manager, Region))

다음과 같이 하십시오.

1. 열 추가를 클릭하고 측정값을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. `Max(Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))`
4. 적용을 클릭합니다.

관리자별로 지역, 평균 주문 값 및 각 지역의 가장 큰 평균 주문 값을 보여주는 테이블.

Aggr() function		
Region	Average order value	Largest average order value
Totals	\$ 1,087	\$ 12,338
Germany	\$ 405	\$ 3,506
Japan	\$ 604	\$ 2,182
Nordic	\$ 641	\$ 2,554
Spain	\$ 577	\$ 1,639
UK	\$ 1,390	\$ 12,338
USA	\$ 1,821	\$ 8,615

지역 수준에서만 모든 관리자의 가장 큰 평균 주문 값을 볼 수 있습니다. 이는 두 가지 복잡한 표현식 중 첫 번째입니다. 다음 요구 사항은 이러한 큰 평균 주문 값을 담당하는 관리자 이름이 해당 값 옆에 표시되도록 하는 것입니다.

이를 위해 이전과 동일한 **Aggr()** 함수를 사용하지만 이번에는 **FirstSortedValue()** 함수를 사용합니다. Qlik Sense에서 관리자를 가져오도록 **FirstSortedValue()** 함수를 사용합니다. 이 함수의 두 번째 부분에 지정된 특정 차원의 경우 다음과 같습니다.

FirstSortedValue (Manager,-Aggr (Sum (Sales) /Count (Order Number), Manager, Region))



표현식에서 작지만 매우 중요한 부분이 있습니다. **Aggr()** 표현식 앞에 빼기 기호를 사용하는 것입니다. **FirstSortedValue()** 함수 내에 데이터 배열의 정렬 순서를 지정할 수 있습니다. 이 경우 빼기 기호를 사용하면 Qlik Sense에서 가장 큰 값부터 가장 작은 값으로 정렬합니다.

다음과 같이 하십시오.

1. 열 추가를 클릭하고 **측정값**을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. `FirstSortedValue(Manager,-Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))`
4. **적용**을 클릭합니다.

지역, 평균 주문 값 및 각 지역의 가장 큰 평균 주문 값과 해당 주문 값을 담당하는 관리자를 보여주는 테이블.

Aggr() function			
Region	Average order value	Largest average order value	Manager
Totals	\$ 1,087	\$ 12,338	Dennis Johnson
Germany	\$ 405	\$ 3,506	Micheal Williams
Japan	\$ 604	\$ 2,182	Brenda Gibson
Nordic	\$ 641	\$ 2,554	Kathy Clinton
Spain	\$ 577	\$ 1,639	Micheal Williams
UK	\$ 1,390	\$ 12,338	Dennis Johnson
USA	\$ 1,821	\$ 8,615	Carolyn Halmon

5 네이키드 필드 참조

필드가 집계 함수에 포함되지 않은 경우 네이키드로 간주됩니다.

네이키드 필드 참조는 여러 값을 포함할 수 있는 배열입니다. 이 경우 Qlik Sense는 사용자가 원하는 값을 모르면 NULL로 평가합니다.

5.1 항상 표현식에 집계 함수를 사용하십시오.

표현식이 제대로 평가되지 않으면 집계 함수가 없을 가능성이 높습니다.

표현식에서 필드 참조는 값의 배열입니다. 예:

Max(Invoice Date)가 단일 값을 보여주는 테이블과 **Invoice Date**가 값의 배열임을 보여주는 테이블.

Max(Invoice Date)	Invoice Date
Max([Invoice Date])	Invoice Date 🔍
▼ 6/26/2014	▲ 1/12/2012
	1/13/2012
	1/18/2012
	1/19/2012
	1/20/2012
	1/21/2012
	1/22/2012
	1/25/2012
	1/26/2012

송장 날짜 필드를 집계 함수에 묶어야 단일 값으로 축소됩니다.

표현식에서 집계 함수를 사용하지 않으면 Qlik Sense는 기본적으로 **Only()** 함수를 사용합니다. 필드 참조가 여러 값을 반환하면 Qlik Sense는 NULL로 해석합니다.

If() 함수를 사용하여 송장 날짜 분할

If() 함수는 조건부 집계에 자주 사용됩니다. 이 함수는 함수 내에 제공된 조건이 True로 평가되는지, False로 평가되는지에 따른 값을 반환합니다.

앱 내의 *Naked field references* 시트에서 제목이 *Using If() on Invoice dates*인 테이블을 찾습니다.

다음과 같이 하십시오.

1. 제목이 Using If() on Invoice dates인 사용 가능한 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. `If([Invoice Date]>= Date(41323), 'After', 'Before')`
5. **적용**을 클릭합니다.

송장 날짜를 참조 날짜로 분할하여 보여주는 테이블.

Using If() on Invoice dates		
Date	Q	if([Invoice Date]>= Date(41323), 'After', 'Before')
Totals		Before
2/10/2013		Before
2/11/2013		Before
2/12/2013		Before
2/13/2013		Before
2/14/2013		Before
2/17/2013		Before
2/18/2013		After
2/19/2013		After
2/20/2013		After
2/21/2013		After
2/24/2013		After
2/25/2013		After

이 표현식은 *Invoice Date*가 참조 날짜(2/18/2013) 이전인지를 테스트하여, 이전이면 'Before'를 반환합니다. 해당 날짜가 참조 날짜(2/18/2013) 이후이거나 같으면, 'After'를 반환합니다. 참조 날짜는 정수 41323으로 표현됩니다.

5.2 네이키드 필드 참조 방지

언뜻보기에 이 표현식은 올바르게 보입니다.

```
If([Invoice Date]>= Date(41323) 'After', 'Before')
```

참조 날짜 이후의 송장 날짜를 평가하여 'After'를 반환해야 하며, 그렇지 않으면 'Before'를 반환해야 합니다. 그러나 *Invoice Date*는 네이키드 필드 참조이고 집계 함수가 없으므로 여러 값을 가진 배열이며 NULL로 평가됩니다. 이전 예에서는 테이블에 *Date* 값마다 하나의 *Invoice Date*만 있었기 때문에 표현식이 올바르게 계산되었습니다.

유사한 표현식이 다른 차원 값에서 계산하는 방법과 발생하는 네이키드 필드 참조 문제를 해결하는 방법을 살펴보겠습니다.

If() 함수에서 네이키드 필드 참조 방지

이전과 유사한 표현식을 사용할 것입니다.

```
If([Invoice Date]>= Date(41323), Sum(Sales))
```

이번에는 함수가 참조 날짜 이후의 판매액을 합산합니다.

앱 내의 *Naked field references* 시트에서 제목이 *Sum(Amount)*인 테이블을 찾습니다.

다음과 같이 하십시오.

1. 사용 가능한 *Sum(Amount)* 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.

4. 다음을 입력합니다. `If([Invoice Date]>= 41323, Sum(Sales))`
5. 적용을 클릭합니다.

연도 및 연간 판매액 합계와 함께 **If()** 함수를 사용한 표현식의 결과를 보여주는 테이블.

Sum(Amount)		
Year	Sum(Sales)	If([Invoice Date]>= Date(41323), Sum(Sales))
Totals	\$ 104,852,675	-
2012	\$ 40,173,302	-
2013	\$ 42,753,991	-
2014	\$ 21,925,382	-



각 표현식 간의 차이를 보여 주기 위해 측정값에 레이블을 그대로 유지합니다. 통화 값이 있는 열에서 숫자 서식을 화폐로 변경하고 서식 패턴을 \$ #,##0;- \$ #,##0으로 변경합니다.

매년 참조 날짜 이후에 나오는 송장 날짜 배열이 있습니다. 표현식에 집계 함수가 없으므로 이는 NULL로 평가됩니다. 올바른 표현식은 **If()** 함수의 첫 번째 매개 변수에서 **Min()** 또는 **Max()**와 같은 집계 함수를 사용해야 합니다.

`If(Max([Invoice Date])>= Date(41323), Sum(Sales))`

다음과 같이 하십시오.

1. 열 추가를 클릭하고 측정값을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. `If([Invoice Date]>= Date(41323), Sum(Sales))`
4. 적용을 클릭합니다.

연도 및 연간 판매액 합계와 함께 **If()** 함수를 사용한 다양한 표현식의 결과를 보여주는 테이블.

Sum(Amount)			
Year	Sum(Sales)	If([Invoice Date]>= Date(41323), Sum(Sales))	If(Max([Invoice Date])>= Date(41323), Sum(Sales))
Totals	\$ 104,852,675	-	\$ 104,852,675
2012	\$ 40,173,302	-	-
2013	\$ 42,753,991	-	\$ 42,753,991
2014	\$ 21,925,382	-	\$ 21,925,382

또는 **If()** 함수는 **Sum()** 함수 내에서 사용할 수 있습니다.

`Sum(If([Invoice Date]>= Date(41323), Sales))`

다음과 같이 하십시오.

1. 열 추가를 클릭하고 측정값을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.

3. 다음을 입력합니다. `Sum(If([Invoice Date]>= Date(41323), Sales))`
4. **적용**을 클릭합니다.

연도 및 연간 판매액 합계와 함께 **if()** 함수를 사용한 다양한 표현식의 결과를 보여주는 테이블.

Sum(Amount)				
Year	Sum(Sales)	If([Invoice Date]>= Date(41323), Sum(Sales))	If(Max([Invoice Date])>= Date(41323), Sum(Sales))	Sum(If([Invoice Date]>= Date(41323), Sales))
Totals	\$ 104,852,675	-	\$ 104,852,675	\$ 58,563,348
2012	\$ 40,173,302	-	-	\$ 0
2013	\$ 42,753,991	-	\$ 42,753,991	\$ 36,637,967
2014	\$ 21,925,382	-	\$ 21,925,382	\$ 21,925,382

두 번째부터 마지막까지의 표현식에서 **if()** 함수는 차원 값마다 한 번 평가되었습니다. 마지막 표현식에서는 원시 데이터의 행마다 한 번 평가됩니다. 함수가 평가되는 방법의 차이로 인해 결과가 달라지지만 둘 다 답을 반환합니다. 첫 번째 표현식은 NULL로 간단히 평가됩니다. 위 그림은 2/18/2013을 참조 날짜로 사용하여 표현식 간의 차이를 보여줍니다.

6 The importance of Only()

Only()는 그룹에 사용 가능한 값이 하나만 있는 경우 값을 반환합니다. 이 값이 집계 결과의 결과입니다. 집계 함수가 지정되지 않은 경우 Qlik Sense에서는 **Only()**를 기본값으로 지정합니다.

차트 차원과 매개 변수 간에 일대일 관계가 있으면 **Only()** 함수는 가능한 값만 반환합니다. 여러 값이 있으면 NULL을 반환합니다. 예를 들어 단가=12인 제품만 검색하는 경우 단가가 12인 제품이 둘 이상 있으면 NULL을 반환합니다.

다음 이미지는 일대일 관계와 일대다 관계의 차이점을 보여줍니다.

Manager Number와 Manager 간에 일대일 관계를 나타내는 테이블

One-to-one relationship	
Manager Number	Manager
104	Amanda Honda
109	Brenda Gibson
111	Carolyn Halmon
118	David Laychak
121	Dennis Johnson
132	John Davis
134	John Greg
144	Kathy Clinton
145	Ken Roberts
157	Micheal Williams
159	Molly McKenzie
160	Odessa Morris
169	Samantha Allen
176	Sheila Hein
179	Stephanie Reagan
181	Stewart Wind
184	Viginia Mountain
185	William Fisher

Sales Rep Name과 Manager의 일대일 관계를 나타내는 테이블.

One-to-many relationship	
Sales Rep Name	Manager
Amalia Craig	Amanda Honda
Amanda Honda	Amanda Honda
Cart Lynch	Amanda Honda
Molly McKenzie	Amanda Honda
Sheila Hein	Amanda Honda
Brenda Gibson	Brenda Gibson
Dennis Johnson	Brenda Gibson
Ken Roberts	Brenda Gibson
Robert Kim	Brenda Gibson
William Fisher	Brenda Gibson
Cary Frank	Carolyn Halmon
Edward Smith	Carolyn Halmon
Lee Chin	Carolyn Halmon
Ronald Milam	Carolyn Halmon
Amelia Fields	David Laychak
Deborah Halmon	David Laychak
Judy Rowlett	David Laychak
Angelen Carter	Dennis Johnson
Dennis Fisher	Dennis Johnson

이 **Only()** 함수는 집계 함수입니다. 많은 레코드를 입력으로 사용하고 **Sum()** 또는 **Count()**와 유사하게 하나의 값만 반환합니다. Qlik Sense는 거의 모든 계산에서 집계를 사용합니다. 차트, 정렬 표현식, 텍스트 상자, 고급 검색 및 계산된 레이블의 표현식은 모두 집계이므로 집계 함수 없이는 계산할 수 없습니다.

그러나 사용자가 명시적인 집계 함수가 없는 표현식을 입력하면 어떻게 됩니까? 예를 들어 정렬 표현식이 *Date*로 설정되어 있는 경우는 어떻게 됩니까? 또는 표현식 `[Product Type]='Beer and Wine'`을 사용하여 맥주와 와인 제품을 구매한 고객을 고급 검색한 경우는 어떻게 됩니까?

여기서 **Only()** 함수가 계산에 영향을 미칩니다. 표현식에 명시적 집계 함수가 없는 경우 Qlik Sense는 **Only()** 함수를 암시적으로 사용합니다. 위의 경우 **Only()** (날짜)가 정렬 표현식으로 사용되고 **Only()** ([제품 유형])='Beer and Wine'을 검색 기준으로 사용합니다.

경우에 따라 새 표현식은 사용자가 예상하지 않은 결과를 반환합니다. 위의 두 예는 *Date* 또는 *Product Type* 중 가능한 값이 하나만 있는 경우에 정상적으로 작동하지만, 둘 이상의 값이 있는 경우에는 모두 작동하지 않습니다.

6.1 Only()를 사용한 서로 다른 표현식

비슷한 표현식으로 4개의 KPI를 만들겠습니다. 이런 방식으로, 네이키드 필드 참조가 있거나 표현식의 다른 위치에 **Only()**가 있는 경우 선택 결과에 어떻게 큰 영향을 미치는지 비교할 수 있습니다.

앱 내의 *Importance of Only()* 시트에서 *Invoice Date* 차원이 있는 필터 창을 찾습니다.

다음과 같이 하십시오.

1. KPI를 만듭니다.
2. 측정값 추가를 클릭합니다. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. `Month([Invoice Date])`
4. 측정값 `Month(Only([Invoice Date]))`, `Month(Max([Invoice Date]))`, `Only(Month([Invoice Date]))`를 사용하여 KPIs를 세 개 더 만듭니다.
5. 적용을 클릭합니다.

네 개의 KPIs와 서로 다르지만 유사한 세 개의 표현식을 보여주는 필터 창입니다.

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Jun	Only(Month([Invoice Date])) —
<div>Q Invoice Date</div> <div> <div>1/12/2012</div> <div>1/13/2012</div> <div>1/18/2012</div> <div>1/19/2012</div> <div>1/20/2012</div> <div>1/21/2012</div> <div>1/22/2012</div> </div>	



각 KPI에서 숫자 서식이 측정값 식으로 설정되었습니다.

네이키드 필드 참조가 있으면 **Only()** 함수가 최하위 수준에 삽입됩니다. 즉, 처음 두 개의 KPI(`Month([Invoice Date])` 및 `Month(Only([Invoice Date]))`)가 동일하게 해석되며 항상 동일한 결과를 제공합니다.

표시된 바와 같이 네 개의 KPI 중 세 개가 NULL을 반환합니다. 반면에 세 번째 KPI(`Month(Max([Invoice Date]))`)는 선택하지 않더라도 값을 반환합니다.

표현식을 작성할 때는 항상 어떤 집계를 사용할 것인지, 또는 여러 값이 있는 경우 어떤 값을 사용할 것인지를 확인해야 합니다. NULL을 사용하여 여러 값을 나타내려면 표현식을 그대로 두십시오. 숫자의 경우 대신 **Sum()**, **Avg()**, **Min()** 또는 **Max()**를 사용할 수 있습니다. 문자열의 경우 **Only()** 또는 **MinString()**을 사용할 수 있습니다.

다음과 같이 하십시오.

1. 시트 편집을 중지합니다.
2. 필터 창에서 1월의 날짜를 선택합니다.
3. ✓을 클릭하여 선택 내용을 확인합니다.

한 날짜를 선택하면 KPI 결과가 변경됩니다.

Month([Invoice Date]) Jan	Month(Only([Invoice Date])) Jan
Month(Max([Invoice Date])) Jan	Only(Month([Invoice Date])) Jan

Q Invoice Date
1/12/2012 ✓
1/13/2012
1/18/2012
1/19/2012
1/20/2012
1/21/2012
1/22/2012

한 날짜를 선택하면 모든 KPI는 올바른 답을 반환합니다. *Month([Invoice Date])*의 표현식과 같이 표현식에 네 이키드 필드 참조가 포함된 경우에도 고유하게 선택하면 적절한 값이 반환될 수 있습니다.

다음과 같이 하십시오.

1. 필터 창에서 1월의 날짜를 추가로 선택합니다.
2. ✓을 클릭하여 선택 내용을 확인합니다.

1월에서 두 날짜를 선택하면 KPI 결과가 변경됩니다.

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Jan	Only(Month([Invoice Date])) Jan
<div> <div>Q Invoice Date</div> <div> <div>1/12/2012 ✓</div> <div>1/13/2012 ✓</div> <div>1/18/2012</div> <div>1/19/2012</div> <div>1/20/2012</div> <div>1/21/2012</div> <div>1/22/2012</div> </div> </div>	

처음 두 KPI는 NULL을 반환하고 다른 두 KPI는 1월의 적절한 값을 반환합니다. 특히 네 번째 KPI는 선택한 두 날짜가 1월의 날짜이기 때문에 올바른 값을 반환합니다.

다음과 같이 하십시오.

1. 필터 창에서 1월이 아닌 날짜를 추가로 선택합니다.
2. ✓을 클릭하여 선택 내용을 확인합니다.

다른 월에서 여러 날짜를 선택하면 KPI 결과가 변경됩니다.

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Feb	Only(Month([Invoice Date])) —

Q Invoice Date	
	1/12/2012 ✓
	1/13/2012 ✓
	2/1/2012 ✓
	1/18/2012
	1/19/2012
	1/20/2012
	1/21/2012

다른 월에서 여러 날짜를 선택하면 세 번째 KPI만 값을 반환합니다. 표현식 `Month(Max([Invoice Date]))`에 따라 선택한 항목에서 가장 큰 월 값을 반환합니다. 네이키드 필드 참조를 포함하는 표현식에는 `Only()`가 자동으로 삽입되므로 최하위 수준이 표현식에 적합하다고 항상 판단할 수는 없습니다. `Only()`를 배치하는 것이 중요합니다.

7 실생활의 예

Qlik Sense에서의 시각화는 데이터에 대한 통찰력을 제공할 수 있습니다. 차트에서 표현식을 사용하면 작업에 특별히 적용되는 결과를 얻을 수 있습니다. Qlik Sense의 기능 범위를 통해 옵션을 쉽게 사용할 수 없는 경우에도 필요에 맞게 표현식을 사용자 지정할 수 있습니다.

7.1 총 수익 백분율 계산

수익을 판매액과 해당 판매액에 대한 비용의 차이로 정의합니다. 매월 수익뿐만 아니라 월별 판매액의 몇 퍼센트가 수익인지 계산합니다.

수익 백분율을 계산하기 위해 다음 표현식을 사용할 수 있습니다.

$(\text{Sum}(\text{Sales}) - \text{Sum}(\text{Cost})) / \text{Sum}(\text{Sales})$

표현식은 더 단순화할 수 있습니다.

$1 - \text{Sum}(\text{Cost}) / \text{Sum}(\text{Sales})$

앱 내의 *Examples from real life* 시트에서 제목이 *Margin*인 테이블을 찾습니다.

다음과 같이 하십시오.

1. 제목이 *Margin*인 사용 가능한 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. $\text{Sum}(\text{Sales})$
5. 표현식을 사용하여 세 개의 측정값 $\text{Sum}(\text{Cost})$, $\text{Sum}(\text{Sales}) - \text{Sum}(\text{Cost})$ 및 $1 - \text{Sum}(\text{Cost})/\text{Sum}(\text{Sales})$ 를 추가합니다.
6. **적용**을 클릭합니다.

판매액 합계, 월별 비용 합계 및 계산된 월별 수익을 금액 및 백분율 형식으로 보여 주는 테이블

Margin					
Month	Q	Sum(Sales)	Sum(Cost)	Calculated Margin	Margin %
Totals		\$ 104,852,675	\$ 61,571,565	\$ 43,281,110	41%
2012-Jan		\$ 1,773,750	\$ 1,122,474	\$ 651,276	37%
2012-Feb		\$ 3,867,568	\$ 2,352,955	\$ 1,514,613	39%
2012-Mar		\$ 3,892,195	\$ 2,339,154	\$ 1,553,041	40%
2012-Apr		\$ 3,660,634	\$ 2,241,036	\$ 1,419,598	39%
2012-May		\$ 3,191,648	\$ 1,961,629	\$ 1,230,019	39%
2012-Jun		\$ 4,259,260	\$ 2,540,976	\$ 1,718,284	40%
2012-Jul		\$ 2,519,873	\$ 1,488,274	\$ 1,031,598	41%
2012-Aug		\$ 3,799,274	\$ 2,312,303	\$ 1,486,971	39%
2012-Sep		\$ 3,739,098	\$ 2,239,469	\$ 1,499,629	40%
2012-Oct		\$ 3,036,456	\$ 1,897,354	\$ 1,139,102	38%
2012-Nov		\$ 3,528,099	\$ 2,193,961	\$ 1,334,138	38%
2012-Dec		\$ 2,905,449	\$ 1,693,359	\$ 1,212,089	42%
2013-Jan		\$ 4,574,043	\$ 2,691,980	\$ 1,882,063	41%
2013-Feb		\$ 3,333,840	\$ 1,925,155	\$ 1,408,685	42%
2013-Mar		\$ 4,266,053	\$ 2,521,409	\$ 1,744,645	41%
2013-Apr		\$ 2,498,576	\$ 1,417,551	\$ 1,081,024	43%
2013-May		\$ 3,533,538	\$ 2,040,086	\$ 1,493,452	42%
2013-Jun		\$ 4,115,434	\$ 2,386,136	\$ 1,729,298	42%
2013-Jul		\$ 2,696,222	\$ 1,515,881	\$ 1,180,341	44%
2013-Aug		\$ 3,792,982	\$ 2,165,853	\$ 1,627,129	43%
2013-Sep		\$ 4,087,106	\$ 2,395,942	\$ 1,691,164	41%
2013-Oct		\$ 2,917,027	\$ 1,699,705	\$ 1,217,322	42%
2013-Nov		\$ 3,647,346	\$ 2,161,120	\$ 1,486,225	41%
2013-Dec		\$ 3,291,823	\$ 1,925,886	\$ 1,365,936	41%
2014-Jan		\$ 4,114,861	\$ 2,363,597	\$ 1,751,264	43%
2014-Feb		\$ 3,198,718	\$ 1,732,256	\$ 1,466,461	46%
2014-Mar		\$ 3,789,271	\$ 2,131,698	\$ 1,657,573	44%
2014-Apr		\$ 3,575,329	\$ 2,035,458	\$ 1,539,871	43%
2014-May		\$ 3,541,237	\$ 2,015,104	\$ 1,526,133	43%
2014-Jun		\$ 3,705,966	\$ 2,063,802	\$ 1,642,164	44%



데이터 형식이 올바른지 확인하는 것이 가장 좋습니다. 이 경우 각 열에서 계산을 나타내도록 레이아웃을 변경합니다. 통화 값이 있는 열에서 숫자 서식을 화폐로 변경하고 서식 패턴을 \$ #,##0;- \$ #,##0으로 변경합니다. 수익 백분율의 숫자 서식을 숫자로 설정하고 서식을 단순 및 12%로 설정합니다.

판매액 및 비용을 기준으로 계산된 월별 수익을 볼 수 있습니다. 또한 판매액의 몇 퍼센트가 수익을 구성하는지 확인할 수도 있습니다.

앱 데이터에는 이미 월별 수익에 대한 데이터가 있습니다. 이를 통해 원래 데이터와 계산을 비교하는 것이 좋습니다.

다음과 같이 하십시오.

1. 열 추가를 클릭하고 **측정값**을 선택합니다.
2. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.

3. 다음을 입력합니다. $Sum(Margin)$
4. 표현식을 사용하여 다른 측정값을 추가합니다. $(Sum(Sales) - Sum(Cost)) - Sum(Margin)$
5. 적용을 클릭합니다.

데이터 집합에서 나오는 월별 수익에 대한 추가 결과 계산된 수익과의 차이가 있는 수익 테이블.

Margin						
Month	Q	Sum(Sales)	Sum(Cost)	Calculated Margin	Margin %	Sum(Margin)
Totals		\$ 104,852,675	\$ 61,571,565	\$ 43,281,110	41%	\$ 43,253,189
2012-Jan		\$ 1,773,750	\$ 1,122,474	\$ 651,276	37%	\$ 651,276
2012-Feb		\$ 3,867,568	\$ 2,352,955	\$ 1,514,613	39%	\$ 1,514,613
2012-Mar		\$ 3,892,195	\$ 2,339,154	\$ 1,553,041	40%	\$ 1,553,041
2012-Apr		\$ 3,660,634	\$ 2,241,036	\$ 1,419,598	39%	\$ 1,419,598
2012-May		\$ 3,191,648	\$ 1,961,629	\$ 1,230,019	39%	\$ 1,230,019
2012-Jun		\$ 4,259,260	\$ 2,540,976	\$ 1,718,284	40%	\$ 1,718,284
2012-Jul		\$ 2,519,873	\$ 1,488,274	\$ 1,031,598	41%	\$ 1,031,598
2012-Aug		\$ 3,799,274	\$ 2,312,303	\$ 1,486,971	39%	\$ 1,486,971
2012-Sep		\$ 3,739,098	\$ 2,239,469	\$ 1,499,629	40%	\$ 1,499,629
2012-Oct		\$ 3,036,456	\$ 1,897,354	\$ 1,139,102	38%	\$ 1,139,102
2012-Nov		\$ 3,528,099	\$ 2,193,961	\$ 1,334,138	38%	\$ 1,334,138
2012-Dec		\$ 2,905,449	\$ 1,693,359	\$ 1,212,089	42%	\$ 1,212,089
2013-Jan		\$ 4,574,043	\$ 2,691,980	\$ 1,882,063	41%	\$ 1,882,063
2013-Feb		\$ 3,333,840	\$ 1,925,155	\$ 1,408,685	42%	\$ 1,408,685
2013-Mar		\$ 4,266,053	\$ 2,521,409	\$ 1,744,645	41%	\$ 1,744,645
2013-Apr		\$ 2,498,576	\$ 1,417,551	\$ 1,081,024	43%	\$ 1,081,024
2013-May		\$ 3,533,538	\$ 2,040,086	\$ 1,493,452	42%	\$ 1,493,452
2013-Jun		\$ 4,115,434	\$ 2,386,136	\$ 1,729,298	42%	\$ 1,729,298
2013-Jul		\$ 2,696,222	\$ 1,515,881	\$ 1,180,341	44%	\$ 1,180,341
2013-Aug		\$ 3,792,982	\$ 2,165,853	\$ 1,627,129	43%	\$ 1,627,129
2013-Sep		\$ 4,087,106	\$ 2,395,942	\$ 1,691,164	41%	\$ 1,691,164
2013-Oct		\$ 2,917,027	\$ 1,699,705	\$ 1,217,322	42%	\$ 1,217,322
2013-Nov		\$ 3,647,346	\$ 2,161,120	\$ 1,486,225	41%	\$ 1,486,225
2013-Dec		\$ 3,291,823	\$ 1,925,886	\$ 1,365,936	41%	\$ 1,365,936
2014-Jan		\$ 4,114,861	\$ 2,363,597	\$ 1,751,264	43%	\$ 1,731,437
2014-Feb		\$ 3,198,718	\$ 1,732,256	\$ 1,466,461	46%	\$ 1,463,099
2014-Mar		\$ 3,789,271	\$ 2,131,698	\$ 1,657,573	44%	\$ 1,657,573
2014-Apr		\$ 3,575,329	\$ 2,035,458	\$ 1,539,871	43%	\$ 1,537,112
2014-May		\$ 3,541,237	\$ 2,015,104	\$ 1,526,133	43%	\$ 1,526,133
2014-Jun		\$ 3,705,966	\$ 2,063,802	\$ 1,642,164	44%	\$ 1,640,192

계산된 수익 열의 일부 값은 데이터에서 직접 나오는 수익 열의 값과 다릅니다. 수익 불일치 열에서 2014년 동안 한 달 안에 발생했음을 명확하게 볼 수 있습니다. 계산된 수익과 데이터 집합에서 나오는 수익의 차이는 작지만, 특정 연도에 발생한 사실에 대해 알고 싶습니다. 해당 연도에 어떤 변경 사항이 있었습니까? 데이터를 조사하고 올바른 질문을 해야 비즈니스에 중요한 사항을 파악할 수 있습니다.

7.2 송장 발행 지연

이 예에서는 송장 생성 날짜와 약속된 생산 제품 배송 날짜를 수집하는 회사를 기반으로 하는 데이터를 사용합니다. 두 날짜가 항상 동일하지는 않습니다. 또한 일부 송장에는 두 개의 약속된 배송 날짜가 있을 수 있습니다. 회사에서 사용되는 송장 발행 시스템에 의해 자동으로 생성되므로 앞의 날짜는 항상 송장 발행 날짜와 동일합니다. 뒤의 날짜는 회사와 고객 간에 배송하기로 합의되어 약속된 배송 날짜입니다.

이러한 날짜를 테이블에 추가해 보겠습니다.

Examples from real life 시트에서 제목이 *Invoicing delays*인 테이블을 찾습니다.

다음과 같이 하십시오.

1. 제목이 Invoicing delays인 사용 가능한 테이블을 선택합니다.
속성 패널이 열립니다.
2. **열 추가**를 클릭하고 **측정값**을 선택합니다.
3. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
4. 다음을 입력합니다. *Only([Invoice Date])*
5. 표현식을 사용하여 다른 측정값을 추가합니다. *Max([Promised Delivery Date])*
6. **적용**을 클릭합니다.

각 송장의 약속된 배송 날짜 및 송장 발행 날짜를 보여 주는 테이블

Invoicing delays		
Invoice Number	Invoice date	Promised delivery date
Totals	-	31 Dec 2014
100001	30 Apr 2013	29 Apr 2013
100002	30 Apr 2013	30 Apr 2013
100005	30 Apr 2013	30 Apr 2013
100006	30 Apr 2013	30 Apr 2013
100007	30 Apr 2013	30 Apr 2013
100008	30 Apr 2013	30 Apr 2013
100009	30 Apr 2013	30 Apr 2013
100010	30 Apr 2013	30 Apr 2013
100011	01 May 2013	01 May 2013
100013	01 May 2013	01 May 2013
100018	02 May 2013	02 May 2013
100021	02 May 2013	02 May 2013
100023	02 May 2013	02 May 2013
100027	03 May 2013	03 May 2013
100028	03 May 2013	03 May 2013
100029	03 May 2013	03 May 2013
100030	03 May 2013	03 May 2013
100034	06 May 2013	06 May 2013
100036	06 May 2013	06 May 2013



데이터 형식이 올바른지 확인하는 것이 가장 좋습니다. 날짜를 표시하는 열에서 숫자 서식을 날짜로 설정하고 서식을 단순 및 17 Feb 2014로 설정합니다.

송장 발행 날짜와 약속된 배송 날짜가 항상 같은 것이 아님을 확인할 수 있습니다. 약속된 배송 날짜가 두 개인 경우 뒤의 날짜를 계산에 사용해야 합니다.

송장 발행 날짜와 약속된 배송 날짜의 차이를 계산해 보겠습니다. 다음 표현식을 사용합니다.

Max ([Promised Delivery Date]) - [Invoice Date]

세 가지 시나리오가 있습니다.

- 두 날짜는 동일하며 표현식의 결과는 0입니다.
- 송장이 발행된 후 제품이 배송되기로 약속되었으며 결과는 양의 정수입니다.

- 제품이 배송되기로 약속된 후 송장이 발행되었으며 결과는 음의 정수입니다.

다음과 같이 하십시오.

1. **열 추가**를 클릭하고 **측정값**을 선택합니다.
2. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. $Max([Promised Delivery Date]) - [Invoice Date]$
4. **적용**을 클릭합니다.

각 송장의 약속된 배송 날짜 및 송장 발행 날짜뿐만 아니라 송장 발행에서 약속된 배송까지의 일수를 보여 주는 테이블

Invoicing delays			
Invoice Number	Invoice date	Promised delivery date	Days from invoicing to delivery
Totals	-	31 Dec 2014	-
307258	21 Jul 2012	22 Feb 2012	-150
108707	30 Jul 2013	29 Apr 2013	-92
109851	09 Aug 2013	14 May 2013	-87
111190	26 Aug 2013	31 May 2013	-87
112112	05 Sep 2013	10 Jun 2013	-87
116817	28 Oct 2013	16 Aug 2013	-73
109998	12 Aug 2013	05 Jun 2013	-68
113609	23 Sep 2013	22 Jul 2013	-63
115559	14 Oct 2013	12 Aug 2013	-63
108081	22 Jul 2013	21 May 2013	-62
109357	05 Aug 2013	05 Jun 2013	-61
310525	26 Aug 2012	26 Jun 2012	-61
315709	25 Oct 2012	25 Aug 2012	-61
329238	27 Dec 2012	27 Oct 2012	-61
103809	03 Jun 2013	08 Apr 2013	-56
112368	09 Sep 2013	16 Jul 2013	-55
118091	11 Nov 2013	18 Sep 2013	-54
112120	05 Sep 2013	15 Jul 2013	-52
112121	05 Sep 2013	18 Jul 2013	-49



송장 발행에서 배송까지의 일수를 나타내는 Days라는 마지막 열을 기준으로 테이블을 정렬합니다.

날짜 사이에는 다양한 차이가 있습니다. 음수 값은 송장 발행이 지연되었음을 나타냅니다. 양수는 송장이 발행된 후 약속된 배송이 완료되었음을 나타냅니다.

약속된 배송 날짜 이후에 발행된 송장 수를 계산해 보겠습니다.

다음과 같이 하십시오.

1. **열 추가**를 클릭하고 **측정값**을 선택합니다.
2. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.

- 다음을 입력합니다. `Count(Distinct If(Aggr(Max([Promised Delivery Date])<[Invoice Date],[Invoice Number]),[Invoice Number]))`
- 적용을 클릭합니다.



또는 `Sum(Aggr(If(Max([Promised Delivery Date])<[Invoice Date]< 0, 1, 0), [Invoice Number]))`를 사용할 수도 있습니다.

송장 발행 지연 테이블에는 지연된 송장 수를 표시하는 추가 열이 있습니다.

Invoicing delays					
Invoice Number	Q	Invoice date	Promised delivery date	Days from invoicing to delivery	Invoice delayed (T/F)
Totals		-	31 Dec 2014	-	3421
307258		21 Jul 2012	22 Feb 2012	-150	1
108707		30 Jul 2013	29 Apr 2013	-92	1
109851		09 Aug 2013	14 May 2013	-87	1
111190		26 Aug 2013	31 May 2013	-87	1
112112		05 Sep 2013	10 Jun 2013	-87	1
116817		28 Oct 2013	16 Aug 2013	-73	1
109998		12 Aug 2013	05 Jun 2013	-68	1
113609		23 Sep 2013	22 Jul 2013	-63	1
115559		14 Oct 2013	12 Aug 2013	-63	1
108081		22 Jul 2013	21 May 2013	-62	1
109357		05 Aug 2013	05 Jun 2013	-61	1
310525		26 Aug 2012	26 Jun 2012	-61	1
315709		25 Oct 2012	25 Aug 2012	-61	1
329238		27 Dec 2012	27 Oct 2012	-61	1
103809		03 Jun 2013	08 Apr 2013	-56	1
112368		09 Sep 2013	16 Jul 2013	-55	1
118091		11 Nov 2013	18 Sep 2013	-54	1
112120		05 Sep 2013	15 Jul 2013	-52	1
112121		05 Sep 2013	18 Jul 2013	-49	1
117460		04 Nov 2013	16 Sep 2013	-40	1

마지막 열은 총 송장 수에 대한 백분율이며 KPI로서 더 큰 의미가 있습니다.

다음과 같이 하십시오.

- KPI를 만듭니다.
- 측정값 추가를 클릭합니다. **fx** 기호를 클릭합니다.
식 편집기를 엽니다.
- 다음을 입력합니다. `Count(Distinct If(Aggr(Max([Promised Delivery Date])<[Invoice Date],[Invoice Number]),[Invoice Number]))/Count([Invoice Number])`
- 적용을 클릭합니다.

지연된 송장의 백분율을 보여 주는 KPI

Percentage of delayed invoices

4%

송장의 평균 지연을 계산해 보겠습니다.

다음과 같이 하십시오.

1. 새 KPI를 만듭니다.
2. **측정값 추가**를 클릭합니다. ***fx*** 기호를 클릭합니다.
식 편집기를 엽니다.
3. 다음을 입력합니다. $Avg(Aggr(If(Max([Promised Delivery Date]) < [Invoice Date]), (Max([Promised Delivery Date]) - [Invoice Date])), [Invoice Number]))$
4. **적용**을 클릭합니다.

송장 발행의 평균 지연을 보여 주는 KPI



7.3 감사합니다.

이제 이 자습서가 끝났습니다. Qlik Sense에서 차트 표현식을 작성하는 데 필요한 기본 지식을 얻으셨기 바랍니다. 웹 사이트를 방문하면 앱에 대한 더욱 많은 영감을 얻을 수 있습니다.