İş Zekası için Dört-Katmanlı Veri Modellemesi Gerçekleştirimi

Blind Author[1,2], Blind Author [1], Blind Author [2], Blind Author [2]

[1]Instution 1

[2] Instution 2

abc@abc, abc@abc, abc@abc, abc@abc

Özet: İş zekası yazılımları kurumların karar verme aşamasında ihtiyaç duydukları bilgiyi ortaya çıkaran ve böylece karar verme sürecine katkı yapan uygulamalardır. Veri modellemesi herhangi bir iş zekası aracında kullanışlılık, performans ve doğruluk bakımından çok önemlidir. Bu çalışmada BI.NET adlı iş zekası aracının kullandığı dört-katmanlı veri modeli ve modeller arası eşlemenin nasıl yapıldığı anlatılmaktadır.

Anahtar Sözcükler: Veri Modellemesi, Sorgulama, İş Zekası, Karar Destek Sistemleri.

Four-layer Data Model Implementation for Business Intelligence

Abstract: Business intelligence software contributes institutions' decision support processes by deriving knowledge, required while making decisions, from data. Data modeling is crucial in any business intelligence tool development as it must enable useful, correct and efficient processing. This work introduces four-layer data model and schema matching between layers that are used in business intelligence tool named BI.NET.

Keywords: Data Modeling, Querying, Business Intelligence, Decision Support Systems.

1. Giriş

Kurumların gelişen dünyada varlıklarını sürdürebilmeleri için zamanlı ve doğru karar almaları gerekmektedir. Bu kararlar faaliyet alanları ile ilgili güncel gelişmeleri takip etmeyi, geçmiş faaliyetlere ait verilerin saklanmasını ve bu verilerden karar almaya yardımcı olacak doğru bilgiyi çıkarmayı gerektirir. Bu gereksinimleri karşılayacak araçlar iş zekası kavramından yola çıkılarak üretilmektedir. İş zekası (*Business Intelligence*-BI) kavramı, doğru karar vermek için doğru zamanda doğru bilgiye doğru erişim olarak tanımlanabilir [1]. Tanımdan da anlaşılacağı üzere iş zekasında asıl amaç doğru karar vermek olduğu için, iş zekası karar destek sistemi (*Decision Support System*-DSS) kapsamında değerlendirilir. Daha keskin bir anlatımla iş zekası veri-güdümlü (*data-driven*) karar destek olarak da ifade edilebilir.

Doğru bilginin elde edilmesi ancak doğru veri ile olabilmektedir. Doğru veriden kasıt, karara yön verecek olan verinin eksiksiz, hatasız, tutarlı ve güncel olmasıdır. Doğru veriye sahip olunsa bile, uygun araç ve gereçlerle işlenmediği takdirde bir işe yaramayacağı kesindir. Doğru erişim bu hususu ifade eder. Yani doğru erişimden anlaşılması gereken, verinin amaca yönelik olarak doğru bir şekilde işlenmesidir (algoritma, yöntem, iş-akış vb.). İş zekası sorgularındaki çeşitlenen ihtiyaçları doğru olarak karşılamak için: karmaşık SQL sorguları, OLAP, veri madenciliği, ah-hoc sorgulama, istatistiksel analiz, raporlama, harici entegrasyon, anahtar performans göstergeleri, dashboard ve grafiksel sunum gibi teknolojiler kullanılmaktadır. Tüm bu amaçlara yönelik olarak etkin saklama ve erişim için veri modellemesi bir iş zekası

yazılımında en önemli etmenlerden biridir. Bu bildiride TUBİTAK destekli geliştirmiş olduğumuz BI.NET adlı iş zekası yazılımının dört-katmanlı veri modeli tanıtılmaktadır.
Bölüm 2'de İş Zekası kavramı ve modelleri, Bölüm 3'te ise BI.NET iş zekası yazılımında kullanılan dört-katmanlı veri modeli anlatılmıştır. Bölüm 4 ise değerlendirme ve sonuç kısmı olarak yer almaktadır.

2. İş Zekası Kavramı ve Modelleri

İş zekası literatürde terim olarak ilk kez 1958 yılında Hans Peter Luhn tarafından "gerçekliklerin birbirleriyle olan ilişkilerinin kullanılarak aksiyon kılavuzu oluşturulması" anlamında kullanılmıştır [2]. Bu tarihten sonra ise iş zekası kavramı karar destek sistemlerinin bir parçası olarak varlığını ve gelişimini sürdürmüştür. 1989'a gelindiğinde ise bugünkü anlamıyla, "gerçekliklerden yola çıkarak karar alma süreçlerine yardımcı olacak metot ve kavramların geliştirilmesi" olarak tanımlanmıştır [3]. İş zekası önceleri karar alma yetkisi olan yönetici konumundaki kişilere hitap ederken, günümüzde karar alma sürecinin daha alt kısmındaki yetkililere de hitap etmektedir. Kişiler yaptıkları işi en etkin yollarla yapabilmek için iş zekası uygulamalarına erişim ihtiyacı duymaktadır. Bilgisayarların ve iş zekası araçlarının yaygınlaşması ile bu imkan sağlanmıştır.
İş zekası doğru zamanda doğru verilere ulaşarak karar verme sürecinde rol almaktır. Karar verme sürecinde, varılan kararların doğruluğu sahip olunan bilginin doğruluğuna doğrudan bağlıdır. Doğru bilgi ise kaynaklık eden verilerin doğru olmasını gerektirir. Verilerin doğru olması, karara yön verecek olan verinin eksiksiz, hatasız, tutarlı ve güncel olması demektir. Karar verme sürecinde doğru bilgiye sahip olmak her zaman yeterli olmayabilir. Doğru bir karar çoğunlukla doğru zamanda verilen bir karardır. Zamanın belirli bir anında doğru bilgilerle verilen bir karar, bir süre geçtikten sonra doğruluğunu yitirebilir. Bunun nedeni verilen karara temel teşkil eden verilerin geçen süre içinde güncelliğini yitirmiş olabilmesidir. Dolayısıyla doğru bir karar vermenin gereksinimi doğru verilere anlık olarak erişip doğru bilgilerin çıkarılmasıdır.

İş zekasına yönelik model ve standartlar da önerilmiştir. *Corporate Information Factory* (CIF) bir kurumdaki operasyonel verilerden bilgi elde edilmesine kadar olan tüm yazılım, donanım ve diğer sistem aktivitelerini tanımlayan bir modeldir [4]. Modele göre, uygulama veritabanlarından (*corporate applications*) veriler öncelikle *staging area* denilen geçici bir yere çekilir ve burada ETL (*Extract, Transform, Load*) işlemine tabi tutularak kurumsal veri ambarına (*Enterprise Data Warehouse*, EDW) yüklenir. Kurumsal veri ambarının özellikleri (i) kurumda bir tane olması, (ii) entegre veriye sahip olması, (iii) tarihi veriye sahip olması, (iv) kalıcı ve detaylı veriye sahip olması olarak sayılabilir. EDW'ye ilave olarak bazı kurumlar Global ODS (*Operational Data Store*) da tutabilirler. ODS fonksiyon olarak EDW ile aynıdır fakat tarihi bilgi içermez ve EDW'ye göre çok daha sık güncellenir. Kurumdaki birimlerin ihtiyaçlarına göre çok sayıda veri kilerleri (*data mart*) hem global ODS hem de EDW'den elde edilir. Veri kilerleri, daha çok bir birimin kullanımına ilişkin özet bilgileri içerir. Veri madenciliği ve DSS uygulamaları (BI dahil) EDW'yi veri kaynağı olarak kullanır.
CIF modelinde çalışan ekipler ve kullanılan araçlar iki kısma ayrılabilir: GDI (*Getting Data In*) ve GIO (*Getting Information Out*) [5]. GDI ekibi ve araçları kurumdaki operasyonel kaynaklardan ETL ile ortak bir veri ambarı oluşturmakla sorumludur. GIO ekibi ve araçları ise veri

ambarından DSS için gerekli bilgileri elde etmekle sorumludur. GDI ekibi kurumun IT ekibinin bir parçasıdır, fakat GIO ekibi kurumun IT ekibi ile iş analistleri (BI aracını kullananlar) arasında bir köprü görevini yerine getirir. Kısaca, GIO ekibi BI araçlarının kullanımını izleme ve analizcilere BI aracını etkin kullanım için destek verme (iş analiz katmanında, örneğin OLAP analizi nedir ve nasıl yapılır gibi) işlerini yaparlar. Buradan hareketle veri modelinin sadece veritabanı uzmanlarına değil aynı zamanda iş zekası uzmanlarına hitap etmesi gerekliliği çıkartılabilir.

Günümüzde iş zekasını destekleyen çok sayıda yazılım geliştirilmiş durumdadır. Bunlar arasında Microsoft BI, Oracle BI Suite Enterprise, IBM Çözümleri (Cognos, SPSS), SAS BI, Qlikview, Teradata, RapidMiner, Eclipse BIRT Project, SpagoBI, JasperSoft, Pentaho ürünleri vardır. *Common Warehouse Metamodel* (CWM) [6] tüm bu araçların veri modellerini ortaklaştırma ve birbiri ile model paylaşımı için önerilmiştir. Bahsedilen ürünlerin birçoğu CWM'i desteklemektedir.

Veri modeli detayları bir sonraki kısımda anlatılan BI.NET iş zekası yazılımı CIF bakış açısıyla CWM temel alınarak geliştirilmiştir.

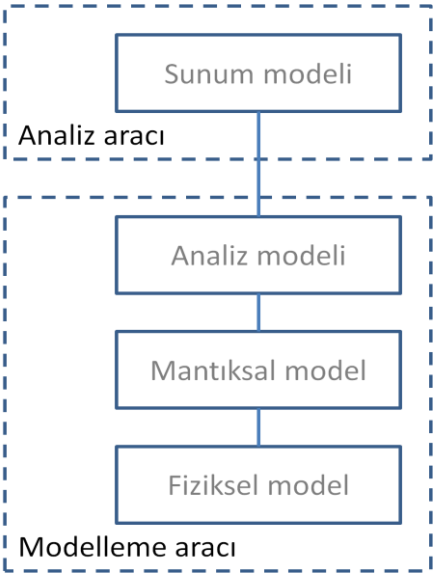
3. Dört-Katmanlı Veri Modellemesi

Veritabanı uzmanı ve iş zekası uzmanının (analist) veritabanına bakış açıları aynı değildir. Veritabanı uzmanı daha çok etkin saklama, güncelleme ve sorgulama açısından veritabanı şemasının normalize olmasını tercih ederken, iş zekası uzmanı analiz modellemesi açısından denormalize görünümü tercih eder. Ayrıca şemadaki bazı tablolar ve öznitelikler analiz için gerekli olmayabilir.

Katmanlı veri modellemesi analistin doğru veri ile başlaması ve doğru çıkarıma varması için gereklidir. Bu yüzden BI.NET'in veri modeli dört-katmanlı yapıdadır (Şekil 1). En alt katman (fiziksel model) çoğunlukla ilişkisel bir veritabanı şemasıdır (tablo, view ve öznitelikler) ve Oracle, MS SQL Server, MySQL ve PostgreSQL gibi VTYS'lerden veritabanı şemaları sorgulanarak elde edilir. Bu modelin analizde ihtiyaç olabilecek şema objelerinin seçimi ve kısmen dönüştürülmesi ile mantıksal model elde edilir. Bu aşamada yapılan işlemler çeşitlidir: denormalizasyon, veri temizleme, filtreleme, bütünleme vb. Mantıksal model fiziksel veritabanı şemasının iş dünyasındaki karşılığı olarak düşünülür. Mantıksal katmanda öncelikle fiziksel alanlardan kopyalama yöntemi ile mantıksal alanlar oluşturulur, bu alanlardan yeni mantıksal alanlar türetilebilir, alanlar kullanıcı dostu olarak yeniden adlandırılabilir ve üzerinde matematiksel sayma ve toplama yapılacak alanlar belirlenir. Analiz modelinde amaç ise, daha ileri bütünlemeler (*aggregation*) yardımıyla birbirinden bağımsız analiz tabloları (kategoriler) elde etmektir. Yani burada her tablo belirli bir bakış açısı (örneğin, muhasebe, finans, insan kaynakları) ile oluşturulur. BI.NET kapsamında kullanıcı yetkilendirmeleri bu üst seviyede yapılır. Bu üç aşama tamamen veri modellemesine yöneliktir ve buraya kadar herhangi bir veri sorgulaması (yani sadece veritabanı şeması kullanılır) yoktur. Bu yüzden bu üç aşama BI.NET içinde Modelleme aracı içinde yapılır (Şekil 2) ve tüm modeller XML formatında saklanır. Modelleme aracının kullanıcıları hem veritabanı hem de iş zekası konularında uzmandırlar.

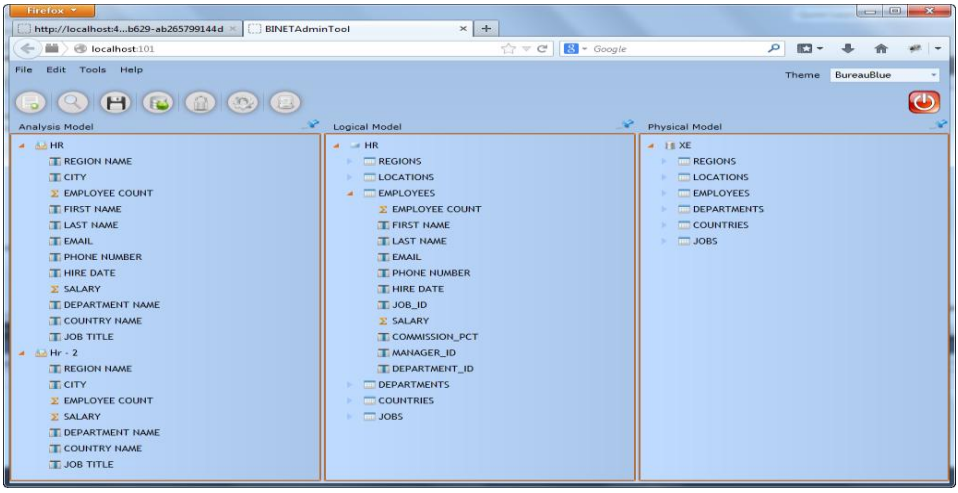
Sunum modeli, tablo olarak saklanmış olan analiz modelinin grafiksel (tablo, pivot, ibreli ölçer (*gauge*), pasta grafik, çubuk grafik, çizgi grafik ve nokta grafik) olarak sunuma hazırlanmasıdır. Sunum modeli sonuçları kullanıcının gösterge panelinde birer nesne olarak gösterilirler. Sunum modelini BI.NET içinde Analiz aracı gerçekleştirir (Şekil 3 ve 4). Analiz

aracının kullanıcıları iş alanı uzmanı olup veritabanı konusunda uzman değildir.
Analiz aracında yapılan tüm analizler için kaynak VTYS'ye bir veya daha fazla SQL sorgusu yapılması gerekir. Bunun için her katman geçişinde şema eşleşmesi ve buna dayalı sorgu yeniden yazması yapılır. Oldukça karmaşık olan bu işlem sorgunun doğru oluşturulması ve doğru bilgiye dayalı karar alma açısından kritiktir.

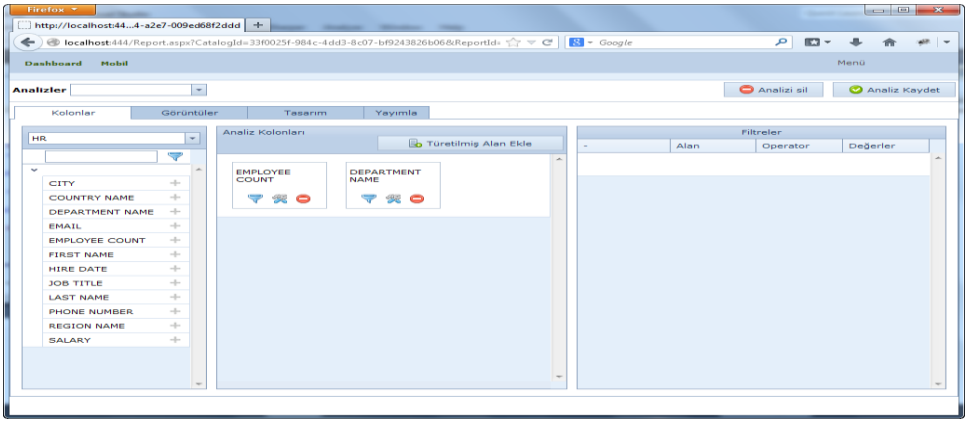


Şekil 1. BI.NET dört-katmanlı veri modellemesi

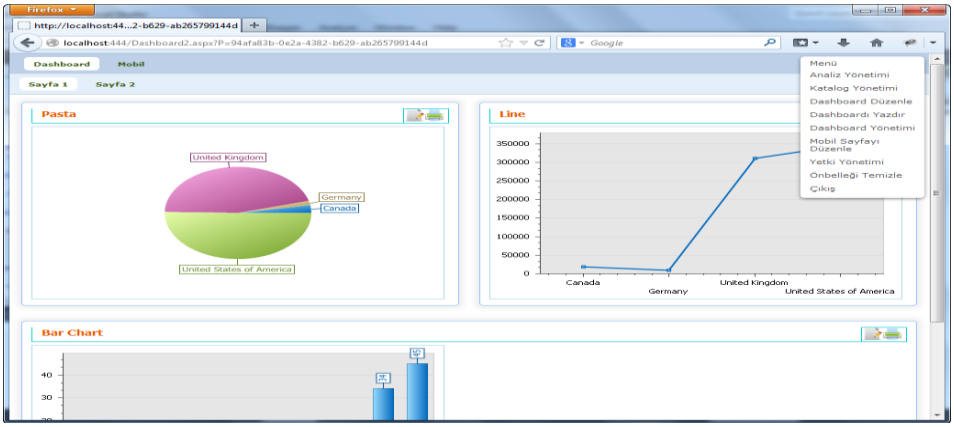
Sunum modelinde seçilen alanlardan uygun SQL sorgusu oluşturulması için sırasıyla seçilen alanların fiziksel öznitelik karşılıklarının, karşılık gelen bu fiziksel özniteliklerin hangi tablolarda yer aldığının, tablolar arasındaki ilişkilere karşılık gelen ve mantıksal modelde tanımlanmış ilişkilerin, hangi öznitelikler üzerinde birleştirme fonksiyonlarının yer aldığının ve hangi özniteliklere filtre atandığının bulunması gerekir.
SQL sorgu oluşturmadan önce belirlenen bu bilgiler ışığında üzerine çalışılan veritabanının bir alt şeması oluşturulur. Bu şemanın oluşturulması için seçilen özniteliklerin varlığı şayet bu öznitelikler birbirine referans bir öznitelikle doğrudan bağlı iki tabloda yer almıyorlarsa yeterli değildir. Bir A tablosundan B tablosuna bir referans, yine B tablosundan bir C tablosuna referans ve hem A hem C tablosundan seçilmiş iki öznitelik olsun. Bu durumda SQL sorgusunun oluşturulması için B tablosuna, B tablosundaki özniteliklerden C'ye referans veren öznitelik ile A'dan referans verilen özniteliklere de ihtiyaç vardır. Dolayısıyla bu gibi tablolar ve alanlar da SQL sorgusunda yer almalıdır.
Seçilmediği halde SQL sorgusunda yer alması gereken öznitelik ve tabloların tespiti en kısa yol algoritmaları ile bulunur. Bir veritabanı tabloları bir çizgenin düğümleri, ilişkileri ise çizgedeki düğümler arasındaki yönlü kenarları gösterecek şekilde düşünülebilir. Bu çizge üzerindeki her hangi iki düğüm arasındaki en kısa yolun bulunmasıyla bu yol üzerinde yer alan düğümlere karşılık gelen tablolar ve bu tablolardaki referans veren ve verilen özniteliklerde sorguya dahil edilir.
Şekil 5'te Oracle Express Edition'da hazır gelen HR veritabanının basit bir şeması görülmektedir. Bu şema kısaca kıtalar, kıtalara bağlı ülkeler, ülkelere bağlı bölgeler, bölgelere bağlı departmanlar ve departmanlara bağlı çalışanlar şeklinde yapılandırılmıştır. Örnek bir gösterim olarak sunum modelinde çalışan maaşının ("Employee" tablosundaki "salary" özniteliği) ve ülke adının ("Country" tablosundaki "countryName" özniteliği) seçilmiş olduğunu varsayalım. Bu iki özniteliğin seçilmesiyle istenilen sonuç esasında ülke bazlı maaş toplamıdır. Bu bilgiye ulaşmak için elle yazılması gereken SQL sorgusu basitçe ve sırasıyla "Employee" tablosunun "Department" tablosuyla beraber join ile sorgulanması, bu sonucun "Location" tablosuyla beraber sorgulanması ve son olarak da "Country" tablosunun sorguya dahil edilmesi şeklinde oluşturulur.



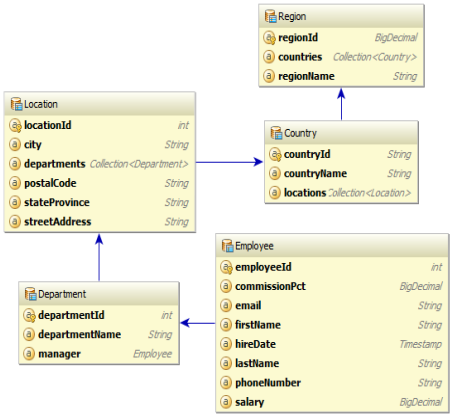
Şekil 2. BI.NET modelleme aracı



Şekil 3. BI.NET analiz aracı (analiz nesnesi tasarımı)



Şekil 4. BI.NET analiz aracı (gösterge paneli)



Şekil 5. Örnek bir HR veritabanı şeması

Sorgu oluşturmada kritik olan konu tüm sorgunun tek bir seferde tüm tablolar birden kullanılarak birden fazla join ile oluşturulmamasıdır. Bu tip durumlarda bütünleştirme fonksiyonu olan alanların değerleri arttırımsal olarak çiftlenmektedir. Bunun önüne geçmek için sorgu oluşturma adım adım olmalıdır. BI.NET sorgu oluşturma algoritması şu şekilde çalışmaktadır:

- Oluşturulan sorgu şemasından en az referans verilen tabloyu bul
- Bulunan tablonun referans verdiği ilk tabloyu bul.
- Bu iki tabloyu birleştiren sorguyu yaz, oluşan sorguyu yeni ve geçici bir tablo olarak birleştirilen iki tablonun yerine şemaya ekle.
- İki tablonun kendileri dışındaki tablolara/tablolardan referans veren/verilen alanları geçici tabloya dahil et.
- Tüm bu adımları tek bir tablo kalana kadar devam ettir.
- Son ve tek kalan tablodan veritabanında çalıştırılacak sorguyu sadece sunum katmanında seçilen alanları içerecek şekilde oluştur.

HR veritabanından "salary" ve "countryName" fiziksel özniteliklerine karşılık gelecek şekilde sunum modelinden alanların seçilmiş olduğu bir örnek seçim için oluşturulan SQL sorgusu Şekil 6'da görülmektedir.

Sayma, toplama gibi bütün sonuçların istendiği durumlarda ilgili alanlara modelleme aracıyla bütünleme fonksiyonlarından atanmalıdır. Bu örnek için "salary" alanına toplam fonksiyonu atanmalıdır. Bazı durumlarda sayılması istenilen değerlere ihtiyaç vardır. Bunun için ise o alanlara say fonksiyonu atanmalıdır ve bu alanlar tablolardaki tekil değerler içeren anahtar alanlar olmalıdır.

Bir SQL sorgusunda filtreler sorgunun "WHERE" ya da "HAVING" kısmının oluşmasını sağlar. Dinamik sorgu oluşturucu filtre atanan alanın taşıdığı bütünleme fonksiyonunun varlığına göre sorgunun "HAVING" ya da "WHERE" kısmını oluşturur.

5. Değerlendirme ve Sonuç

Karar vericiler verdikleri kararların içerik ve zamanlama olarak doğru olmasını amaçlarlar. Bu amaçtan hareketle var olan durumun değerlendirilmesi ve alınacak kararların var olanı nasıl, ne yönde değiştireceğine dair ipucu niteliğinde doğru bilgilere dayanması gerekir. Bu tip bir değerlendirme ve doğru bilgilere erişim için iş zekası araçlarına ihtiyaç vardır.

Veri modellemesi iş zekası araçlarında en kritik öğelerden biridir. Çünkü, farklı roldeki kullanıcıların veri modelleme ihtiyacı farklıdır ve bu modeller arasında şema eşlemesi gerekir. Şema eşlemesi, en üst seviyede seçilen alanların en alt seviyede veritabanı nesnelerine eşlenmesini ve otomatik olarak doğru sorgunun yapılmasını gerektir. Sorgunun doğruluğu kadar hızlı çalışması da önemli olduğundan BI.NET kapsamında çizge tabanlı bir veri yapısı kullanılarak sorgu oluşturulmaktadır.

```
SELECT N23 AS "COUNTRY NAME", N22 AS "SALARY"
FROM(SELECT SUM(N19) AS N22, N20 AS N23
     FROM (SELECT SUM(N16) AS "N19", "N0" AS "N20"
           FROM (SELECT SUM(N13) AS "N16", "N6" AS "N17"
                 FROM (SELECT SUM(N2) AS "N13", "N8" AS "N14"
                FROM (SELECT SUM(SALARY) AS "N2", "DEPARTMENT_ID" AS "N10"
FROM EMPLOYEES
GROUP BY "DEPARTMENT_ID") EMPLOYEES
                       INNER JOIN
(SELECT "DEPARTMENT_ID" AS "N11","LOCATION_ID" AS "N8"
                            FROM DEPARTMENTS
                            GROUP BY "DEPARTMENT_ID", "LOCATION_ID") DEPARTMENTS
                        ON EMPLOYEES.N10 = DEPARTMENTS.N11
                       GROUP BY "N8") N12
                        INNER JOIN
(SELECT "LOCATION_ID" AS "N9", "COUNTRY_ID" AS "N6"
                            FROM LOCATIONS
                            GROUP BY "LOCATION_ID", "COUNTRY_ID") LOCATIONS
                        ON N12.N14 = LOCATIONS.N9
                        GROUP BY "N6") N15
                INNER JOIN
(SELECT "COUNTRY_NAME" AS "N0", "COUNTRY_ID" AS "N7"
                     FROM COUNTRIES
                     GROUP BY "COUNTRY_NAME", "COUNTRY_ID") COUNTRIES
                ON N15.N17 = COUNTRIES.N7
                GROUP BY "N0") N18
        GROUP BY N20) N21
```

Şekil 6. Otomatik Oluşturulmuş Örnek Sorgu

6. Kaynaklar

[1] R. Stackowiak, J. Rayman, R. Greenwald, "Oracle Data Warehousingand Business Intelligence Solutions", **Wiley**, 2007.

[2] H. P. Luhn, "A Business Intelligence System", IBM Journal 2 (4): 314, 1958.

[3] D. J. Power, "A Brief History of Decision Support Systems, version 4.0". **DSSResources.com**, son erişim Aralık 2013.

[4] W. H.Inmon , C. Imhoff , R. Sousa, "Corporate Information Factory", 2001, **Wiley**.

[5] C. Imhoff, "Enterprise Business Intelligence: Leading the Way to Competitive Advantage", **Intelligent Solutions Inc. (White paper)**.

[6] J. Poole, D. Chang, D. Tolbert, D. Mellor, "The Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration". **OMG Press (John Wiley&Sons)**, 2002.