Stajımın ilk gününde Ataşehir ilçesinde bulunan Intellica İş Zekâsı Yazılım Danışmanlık ‘de bulunarak; burada yer alan Intellica Eğitim kampı ekip arkadaşları ile tanışmadan sonra; Şirketin CEO’su Okan KARADUMAN geniş çapta oryantasyon ile şirket hakkında bilgilendirme, çalışma alanları, partnerlikleri ve çalışma alanları hakkında tanıtımlar yapıldı.

C:\Users\BH\Desktop\ODI VIDEOS\cropped-Logo-Intellica-copy.png

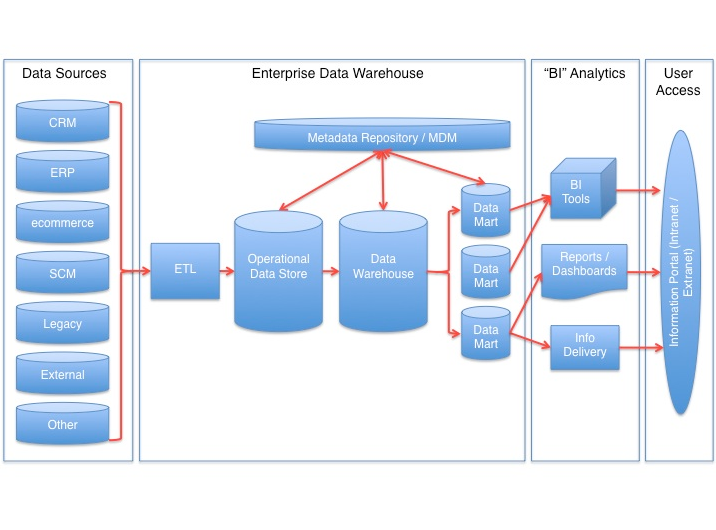
Hakkımızda

Intellica, 10'dan fazla ülkede faaliyet gösteren büyük veri, iş zekâsı danışmanlığı ve yazılım geliştirme çözümleri sunmaktadır. Intellica, ayrıca iş zekâsı, büyük veri ve karmaşık olay işleme konusunda uzmanlaşmış uzman bir şirkettir. Dr. Kemal Unaltuna ve F.Can Alhas tarafından kurulmuştur ve 2005 yılında iş zekâsı danışmanlık şirketi olarak başlamıştır. 2008 yılında da çeşitli sektör ve büyüklükteki proje uygulamalarından elde edilen deneyimlere dayalı çözümler geliştirmek amacıyla kurulmuştur.

Intellica'nın vizyonu, müşterilerine uçtan uca iş çözümleri sunmaktır. Etkili bir İş Zekâsı, CRM ve CEP çözümü oluşturmak, bir işletme için kullanılabilirliği ve değeri sağlamak için önemli miktarda çaba gerektirir.

Intellica'nın müşterileri; İş Bankası, Garanti Bankası, Zagrebacka Bank (Unicredit Group), Finansbank, Türkiye Ekonomi Bankası, ING Bank, Kuveyt Türk, Bank Asya, Türkiye Finans, Albaraka (bankacılık); Turkcell, AVEA, Vodafone, Türk Telekom, Bakcell, Superonline, Digiturk (telekomünikasyon ve yayıncılık), AvivaSa, Anadolu Sigorta, Groupama Sigorta, Groupama Emeklilk, DASK, Ray Sigorta (sigorta), SGK (halk), POAS (petrol), Boyner (perakende), Markafoni, E-bebek, Bilyoner (E-iş).

Stajımın bu iki gününde gününde Intellica kurucularından *Hagop Gobelian* tarafından veri modelleme hakkında ve bu alandaki tübitak ve teydeb projeleri hakkında bilgi verildi. Sonrasında iş zekası (BI) veri madenciliği temelleri ETL sürecinden itibaren DataMart aşamasına kadar ayrıntılı şekilde anlatıldı.



Business Intelligence (BI) terimi, işletme bilgilerinin toplanması, entegrasyonu, analizi ve sunumu için teknolojiler, uygulamalar ve uygulamalar anlamına gelir.

BI, kuruluşun stratejik, taktik ve operasyonel iş kararları üzerinde doğrudan bir etkiye sahiptir. BI, varsayımlar ve içgüdüsel hislerden ziyade tarihsel verileri kullanarak gerçeğe dayalı karar vermeyi destekler.

Veri odaklı kuruluşlar, bir veri havuzundaki geçmiş ve gerçek zamanlı verilere erişmek için sorgular gerçekleştirmek, özelleştirilmiş raporlar oluşturmak ve gelecekteki eğilimleri tahmin etmek için çeşitli BI araçları kullanırlar. Bu araçlar, eğitimli veri bilimcileri tarafından gerçekleştirilen gelişmiş analitiklerin yanı sıra, makine öğrenme algoritmaları tarafından özerk bir şekilde üretilen görüşleri içermektedir.

BI uygulamaları için veri depoları şunları içerir: data warehouse, source database, operasyonel data stores ve data martları.

Data source: Birçok uygulama alanı, etkili ve zamanında karar vermek için yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış verilerin kullanılmasını gerektirir.

Tüm bu veriler iki tür kaynaktan elde edilebilir: iç ve dış.

Dahili veri kaynağı, bir organizasyon içindeki operasyonel sistemler tarafından yakalanan ve tutulan verileri ifade eder (CRM, ERP, vb.).

Operasyonel sistemler sadece bir organizasyonun günlük iş operasyonlarını desteklemek için kullanılan güncel verileri içerir.

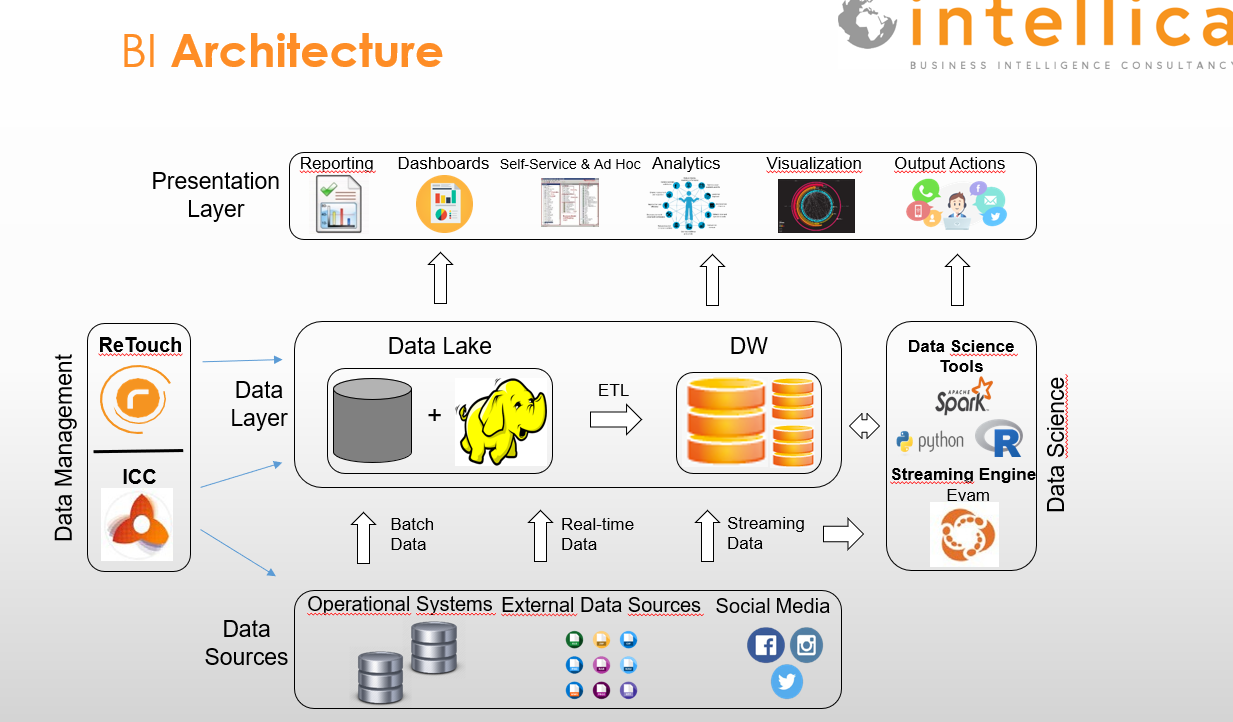
Dış veri kaynağı, kuruluş dışından gelenleri ifade eder. Bu tür veriler, iş ortakları, sosyal medya ve pazar araştırması kuruluşları gibi dış kaynaklardan toplanabilir

.

Data Warehouse : Veri ambarı olarak da adlandırabileceğimiz Data Warehouse, BI sistemlerinde kullanılan tutarlı verilerin tutulduğu merkezileştirilmiş ambarlardır. Bu kısa tanımla birlikte aslında veri ambarları, veri analizi, veri madenciliği ve raporlama işlemlerinin her biri için verinin farklı kopyalarını da tutarlar. Ayrıca sorgu performansını artırmak için bir çok durumda Denomalize işlemine tabi tutulmaları gerekmektedir.

Data Mart: Data Mart’ları Veri Ambarının bir parçası olarak tanımlayabiliriz. Örnek vermek gerekirse, bir şirket içerisinde bulunan Pazarlama ve Satış departmanlarına ait analizleri Veri Ambarının tamamından değilde sadece Pazarlama yada sadece Satış departmanlarına ait bölümlerinde çekip işlemeye Data Mart adını verebiliriz.Kısaca ETL aşamasında DW aşamasında kadar geçen organizasyonun özetti şeklindedir.

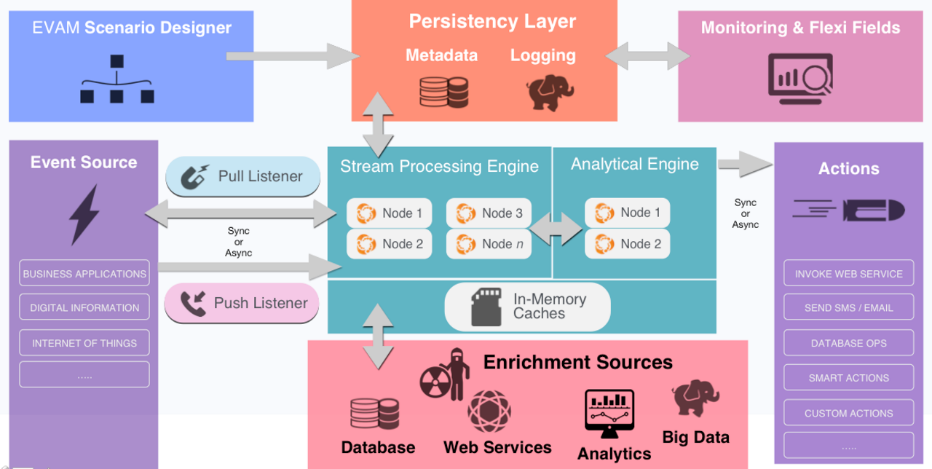
ETL: Data sourcestan verileri kayıpsız bir şekilde ETL toolları(ODI vb.) alma aşamasıdır. Bu aşama ile ilgili ilerleyen günlerde ayrıntılı şekilde Oracle Data Integrator toolunun çalışma prensiplerini göreceğiz.



Modelleme alanında Kullanılan ve Intellica tarafından geliştirişlen ITDM (Intellica Telecomunication Data Model ) gösterilerek temel modelleme prensipleri kazandırıldı.

Evam ile ilgili olan eğitimde bize Tolga Can Erdem, Intellicanın geliştirdiği EVAM ( Event Action Manager ) yazılımı hakkından bilgi verdi.

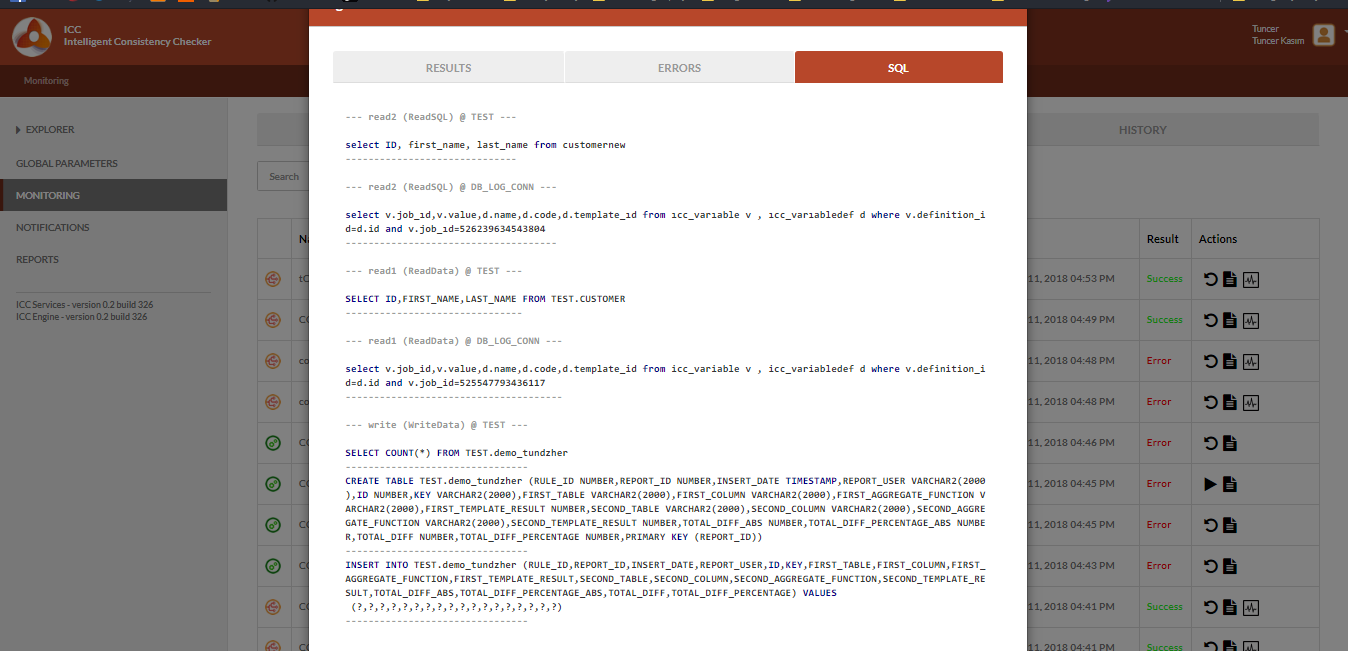
EVAM basit şekilde bahsedecek olursak C.E.P yani complex event prosessing yapan, birçok kaynaktan alınan veriyi işlemek amacıyla kullanılan bir yazılım.



Eğitimin ikinci kısmında birlikte EVAM yazılımını birlikte kurulumunu yapıp üzerinde Oracle 11g database ile işlemler yaptık.

Aynı zamanda yine Intellicanın geliştirdiği ICC toolunu tanıdık. ICC farklı kaynak sistemlere bağlanıp aynı verilerin doğruluğu gözlemlenen bir java tabanlı yazılım.

ICC aynı zamanda hata uyarıları dizayn edebildiğimiz bir uygulama olup ETL aşamasında tabloların doğruluğunu kontrol için yardımcı oluyor.Aynı zamanda ICC ile kontrol jobları oluşturup verileri kontrolünü sağlıyoruz.



Eğitimin bugününde evamcep.atlassian.net üzerinden çalışmamızda kullanılacak olduğumuz

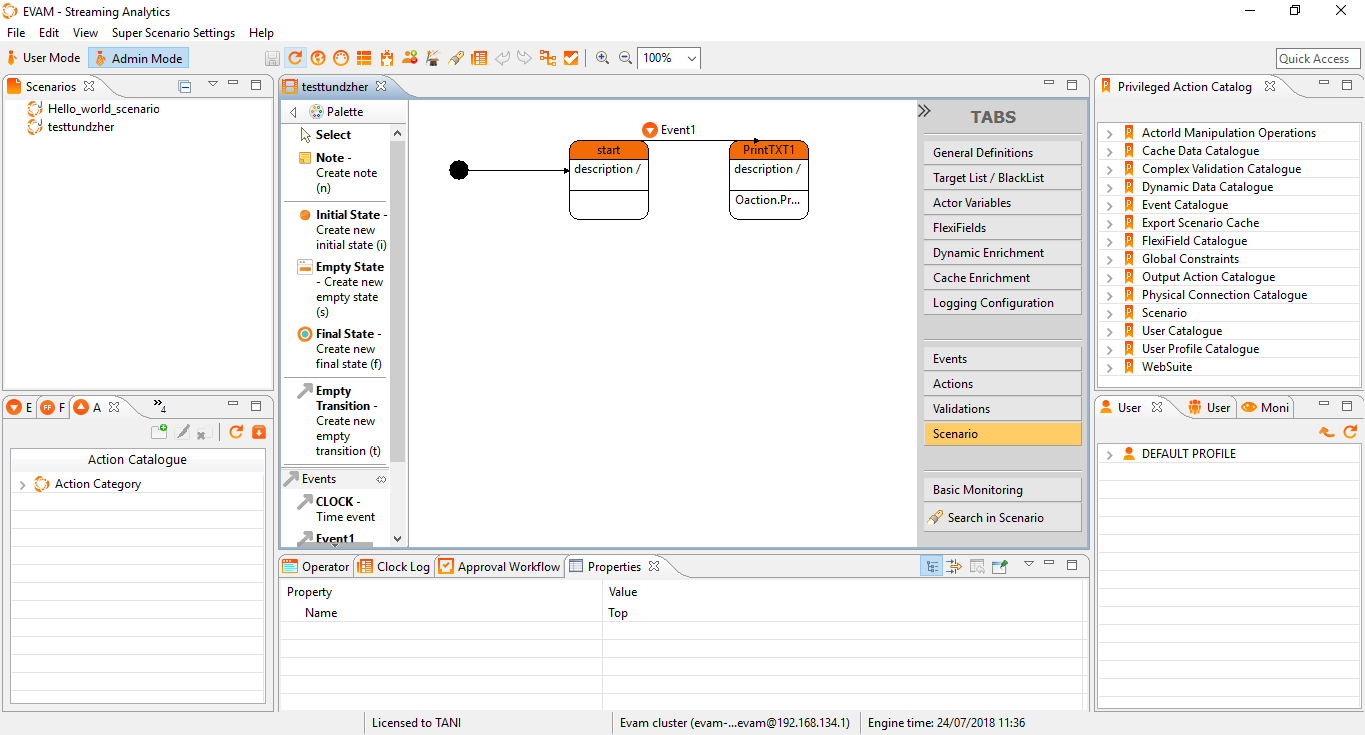
* Evam Engine
* Evam web süite
* Listener
* Snaketail

Programlarını indirip EvamEngine’i java 1.8 idk sı ile birlikte local olarak kurulum yaparak config klasörü altında Oracle 11g database’ine bağlantı ayarlarını kurup EvamEngin Shell üzerinden çalıştırıyoruz.

Bu işlemlere paralel olarak Web üzerinden erişim için evam web süite kurulumu gerçekleştirildi.

Aynı zamanda yapılan işlemleri gözlemlemek için Log dosyalarını anlık görüntüleme için Snaketail programı kullanılarak Log dosyAlarına bağlantısı yapıldı.

Uygulama aşamasında Bin klasörü içind e Shell açıp ./evam.bat start 0 deyip evam engini ayağa kaldırıyoruz. Sonrasında Evam Designer i çalıştırıp Evam içerisinde 1 tane event tanımlıyoruz sonrasında action tanımlayıp adını PrintTXT olarak girip print komutunu seçiyoruz. Son olarak bu işlemleri drag and drop yöntemi ile evam üzerinde bir job oluşturup çalıştırıyoruz. Sonuç olarak EVAM makinesi üzerinde çalıştırmış olduğumuz işlemin sonucunu Snaketail üzerinden izliyoruz.



Eğitimimizin bugününde Mustafa Erol ( Intellica Solutions Architect) tarafından SQL eğitimi verildi. Sql sorgu temelleri öğretilerek sql developer’da örnek HR veritabanı ile çalışmalar yapıldı*.*

**SQL ( STRUCTURED QUERY LANGUAGE )**

SQL kelimesinin açılımı Structured Query Language’dir. Yani “Yapısal Sorgulama Dili”. Veritabanındaki veriler sıradan dosyalar gibi değildir. Belli bir düzene ve yapıya göre saklanır. Bu nedenle bu verilere yapısal veriler denir. SQL, veritabanındaki verileri okumak, güncellemek, yeni veri eklemek ve silmek gibi işlemleri yapan bir sorgulama dilidir. Satırlarca kodun yaptığı işlemler SQL ile tek bir sorguda yapılabilir.

SQL, (“Structured Query Language”) verileri yönetmek ve tasarlamak için kullanılan bir veritabanı yönetim sistemidir. SQL herhangi bir veri tabanı ortamında kullanılan bir alt dildir. SQL ile yalnızca veri tabanı üzerinde işlem yapılabilir. SQL'e özgü cümleler kullanarak veri tabanına kayıt eklenebilir, olan kayıtlar değiştirilebilir, silinebilir ve bu kayıtlardan listeler oluşturulabilir. SQL yapısal bir sorgulama dilidir.

SQL temel olarak şu ifadelerle kullanılır:

**SELECT – FROM – WHERE - ORDER BY - GROUP BY – HAVING – UPDATE – DELETE – INSERT**

Oracle Database 11g XE edition kurulduktan sonra buna paralel olarak Oracle SQL developer kurulumu tamamladık. Sonrasında SQL Developer programının kurulumunu yaptıktan sonra eğitimcilerimiz ile birlikte çeşitli sorgulamalar yaparak SQL’in temel kullanımını anlamaya çalıştık. Sonrasında programlama dilinde bazı sorguların kullanımlarının araştırmalarını yapmaya çalıştık.

**SQL JOIN TİPLERİ**

**Inner join:** Birinci tablodaki tüm kayıtları çeker ve bu kayıtlar ile eşleşen 2.tablodaki kayıtlarda listelenir. Dönen kayıt sayısı 1.tablodaki kayıt sayısıdır.

**Left outer joın:** Soldaki tablodaki tüm kayıtlar alınır, sağdaki tabloda eşleşen kayıtlar aynen yazılır eşleşmeyenler “null” döner. Kayıt sayısı sol tablodaki kayıt sayısı kadardır.

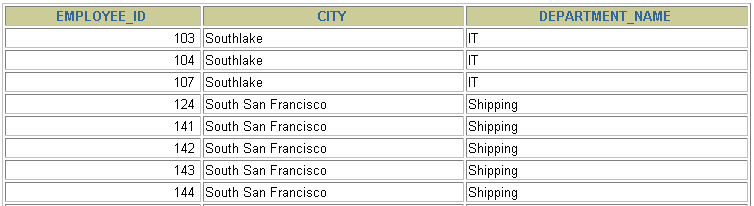
**Right outer joın:** Yukarıdaki ile tam tersi işlemi yapar. Kayıt sayısı sağ tablodaki kayıt sayısıdır.

**Full outer join:** Left ve right join birleşimidir. Ancak her iki tabloda da “null” değerleri

görülebilmektedir.

**Self join:** Bir tablonun kendisiyle birleştirilme işlemidir. Bazen oluşturduğumuz tablodaki bir bilgi yine aynı tablo içindeki başka bir bilgiye referans olabilir.

Eğitimimiz bugününde yine SQL Developer HR veritabında, join fonksiyonlarıyla ilgili örneklere devam edildi ve tarih fonksiyonları ile çalışmalar yapıldı.

**

Örnek:

SELECT employee\_id, city, department\_name

FROM employees e

INNER JOIN departments d

ON d.department\_id = e.department\_id

**SQLTarih Fonksiyonları**

Oracle veritabanı, tarihleri dahili bir sayısal biçimde saklar: yüzyıl, yıl, ay, gün, saat, dakika, saniye. Varsayılan tarih görüntüleme formatı DD-MON-RR'dir.

SYSDATE: Tarih ve Saati döndüren bir fonksiyondur.

MONTHS\_BETWEEN: iki tarih arasındaki ay sayısı. MONTHS\_BETWEEN ('01-SEP-95','11-JAN-94') 🡪 19.6774194

ADD\_MONTHS: tarihe bir ay ekler. ADD\_MONTHS ('11-JAN-94',6) 🡪 11-JUL-94

NEXT\_DAY: Belirtilen tarihin ertesi gününü getirir. NEXT\_DAY('01-SEP-95','FRIDAY')🡪08-SEP-95

LAST\_DAY: ayın son gününü getirir. LAST\_DAY('01-FEB-95')🡪 28-FEB-95

ROUND: tarihi yuvarlar (ay, yıl). ROUND(SYSDATE ,'YEAR')🡪01-JAN-96

TO\_CHAR, TO\_NUMBER , TO\_DATE : Tarihleri istediğimiz formata çeviren fonksiyonları

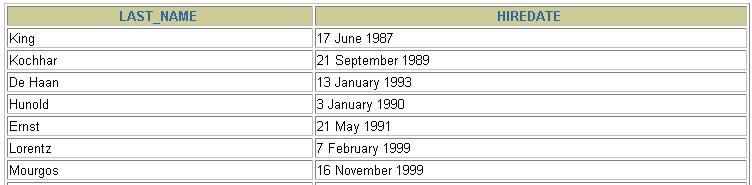
Örnek:

**SELECT last\_name,**

**TO\_CHAR(hire\_date, 'DD Month YYYY')**

**AS HIREDATE**

**FROM employees;**



ekran çıktısı

Eğitimimiz bugününde yine SQL Developer HR veritabında, Group fonksiyonlarıyla ile ilgili çalışmalar yapıldı.

**SQL GROUP FONKSİYONLARI**

Grup fonksiyonları, grup başına bir sonuç vermek için satır kümelerinde çalışır.

* AVG, COUNT,MAX,MIN,SUM

COUNT: (DISTINCT expr), expr öğesinin boş olmayan farklı değerlerinin sayısını döndürür.

AVG: Belli bir gruptaki ortalama değeri döndürür.

MAX,MIN: Belli bir gruptaki en yüksek ve en düşük değerleri döndürür.

SUM: Belli bir gruptaki tüm değerlerin toplamını döndürür.

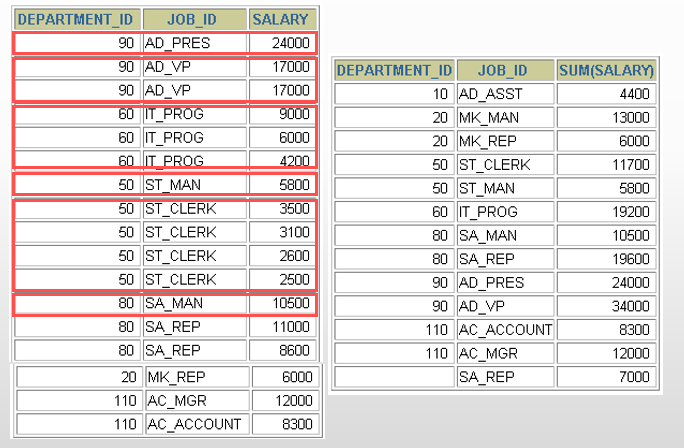
GROUP BY kullanarak bir tablodaki satırları daha küçük gruplara bölünür.

Örnek: Her job için departmanlar gruplanarak her bir departmadaki toplam maaş hesaplayan sorgu.

**SELECT department\_id dept\_id, job\_id, SUM(salary)**

**FROM employees**

**GROUP BY department\_id, job\_id ;**



HAVING kullanılarak gruplanan birimlerden istenilen şartlar istenir

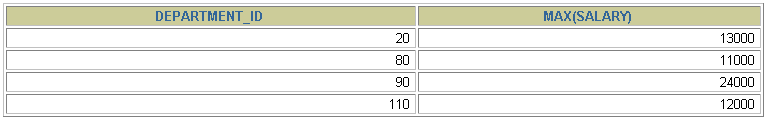
ÖRNEK: maaşı 10000den fazla olan departmanları gösteren sorgu.

**SELECT department\_id, MAX(salary)**

**FROM employees**

**GROUP BY department\_id**

**HAVING MAX(salary)>10000 ;**



Eğitimimiz bugününde bir database’de mevcut tablolar üzerinde işlemler yaptık. Ve Subquery ifadelerini öğrendik.

**Data Manipulation Language**

Bir tabloya yeni satırlar ekleme, tablodaki mevcut satırları değiştirme, mevcut satırları tablodan kaldırma işlemlerinin yapıldığı, sql yapılarıdır.

INSERT: Tabloya yeni satır eklemede kullandığımız ifade.

UPDATE: Tablodaki satırları değiştirmede kullandığımız ifade.

WHERE: Bu ifade kullanılarak belirli satır veya satırlar değiştirilir

DELETE: Varolan satırlar değiştirilir.

**Subquery:** Group by fonksiyonlarında olduğu gibi belli sütunları gruplar, fakat büyük ve karmaşık tablolarda subquery kullanılır.

**Subquery** , ana sorgudan önce bir kez yürütülür.

**Subquery** sonucu ana sorgu tarafından kullanılır

**11000**

**SELECT last\_name**

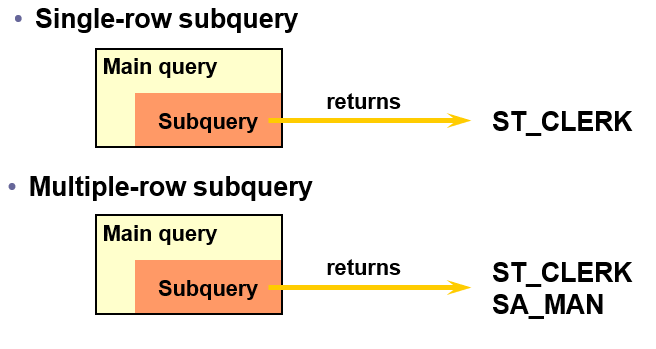
**FROM employees**

**WHERE salary >**

**(SELECT salary**

**FROM employees**

**WHERE last\_name = 'Abel');**

Subqueries kullanımı:

Subqueriesleri parantez içine alın.

Subqueries karşılaştırma koşulunun sağ tarafına yerleştirin.

single-row subquerieslere sahip single-row operatörleri kullanın

ve multiple-row Subquerieslara multiple-row operatörleri kullanın

SINGLE-ROW SUBQUERY ÇALIŞTIRMA:

**SELECT last\_name, job\_id, salary**

**ST\_CLERK**

**FROM employees**

**WHERE job\_id =**

**(SELECT job\_i**  **d**

**FROM employees**

**2600**

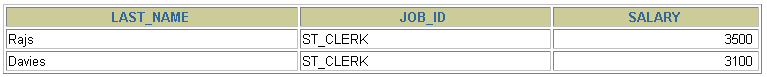
**WHERE employee\_id = 141)**

**AND salary >**

**(SELECT salary**

**FROM employees**

**WHERE employee\_id = 143);**

****

**DATABASE NESNELERİ:**

**Tablo**: Temel depolama birimi; satır ve sütunlardan oluşur

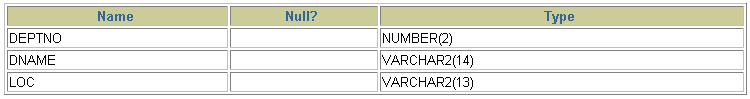
**View**: Mantıksal olarak bir veya daha fazla tablodaki veri alt kümelerini temsil eder

**Sequence:** Sayısal değer üreteci

**Index:** Bazı sorguların performansını artırır

**Synonym:** Nesnelere alternatif isimler verir

CREATE TABLE Statement:

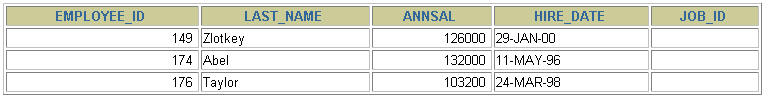
**CREATE TABLE dept DESCRIBE dept;  
 (deptno NUMBER(2),**

**dname VARCHAR2(14),**

**loc VARCHAR2(13));**

**Table created.**

ALTER TABLE Statement:

**ALTER TABLE dept80 DESCRIBE dept;**

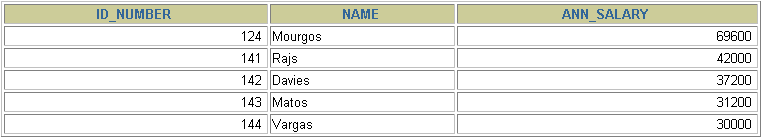
**ADD (job\_id VARCHAR2(9));**

**Table altered.**

Yeni kolon (job\_id), son kolonun yanına geldi

CREATE VIEW Statement:

**CREATE VIEW salvu50**

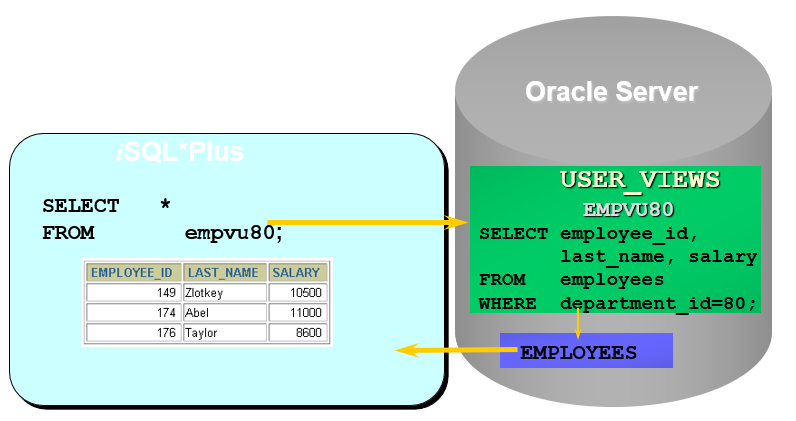
** AS SELECT employee\_id ID\_NUMBER, last\_name NAME,**

**salary\*12 ANN\_SALARY**

**FROM employees**

**WHERE department\_id = 50;**

**View created.**



Querying a View Removing a View

**DROP VIEW *view*;**

**DROP VIEW empvu80;**

**View dropped.c**

Eğitimimizde bu iki gününde farklı sistemler arasında yüksek hızda veri taşımak için kullanılan ELT(Extract, Load ve Tranform) mimarisine sahip oracle tool’u olan Oracla Data Integrator(ODI) 11g programını yükledik. Ve birden çok kaynak sistemden kendi veri ambarımızı oluşturabilmak için ODI kullanımı detaylı bir şekilde Intellica Solution Architect Eren Denli tarafından anlatıldı.

**ODI,** veri kaynakları çok sayıda, temizlenmiş ve tablolardan çekmeli, daha sonra hedeflere yüklenecek veriler alımını sağlar. Bu yeni set edilmiş takılabilir bileşenleri veya herhangi heterojen kaynaklara uzaktan erişimi sağlar.Yani onların tanımlarıyla bağlantılar kurar.kendileri offline olsa bile ; onlarla ilgili işlemler yapabilir.Sürükle bırak mantığı ile ama yine de bir kod tabanı üzerinden yapar.Süreç akışı ve veri haritalama kullanarak,ELT işleme benzer veri entegrasyonu ODI ile gerçekleştirilebilir. Veri birden fazla kaynaktan, çeşitli dönüşüm süreçleri gönderilerek  ve ayıklanan bir nihai hedefe yüklenerek oluşturmak istediğimiz tablo kriterlerine uyacak kadar esnek bir yapı ile ODI de(daha doğrusu; yine Oracle’ın kendisinde) işlenebilir.. Dönüşümler kaynak sistemleri veya platformu üzerinden, hedef ortamda özel kodu dahil bilgi modülleri tarafından tanımlanarak oluşabilir.

Oracle Data Integrator, yüksek performanslı hareket ve veri dönüşümü düzenine heterojen sistemler toplu işlem, gerçek zamanlı, senkron ve asenkron modları; dramatik bir yenilikçi, tasarım yaklaşımı ve modüler ile kullanıcı verimliliğini artırır. Oracle Data Integrator, grafik modulleri ve yazılım maddeleri ile buna izin verir

Kullanım Alanları:

• Ters mühendislik uygulama modelleri.

• Veri tutarlılığı kontrolu.

• Arabirimleri uygulamaları arasında Tasarım, test, işletmek ve korumak

• Check veri arayüzleri tarafından, hata izolasyonu ve / veya geri dönüşüm ile işlenmiş olarak  akar.

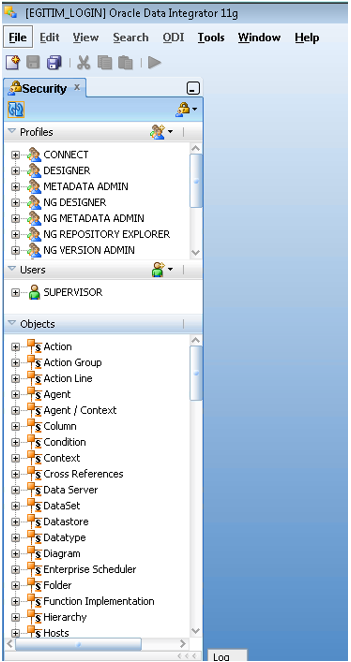
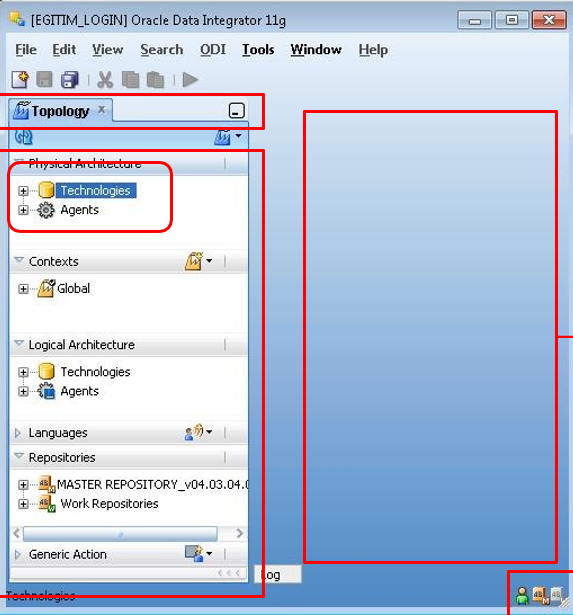
• eksik veri girişi tanımlamak.

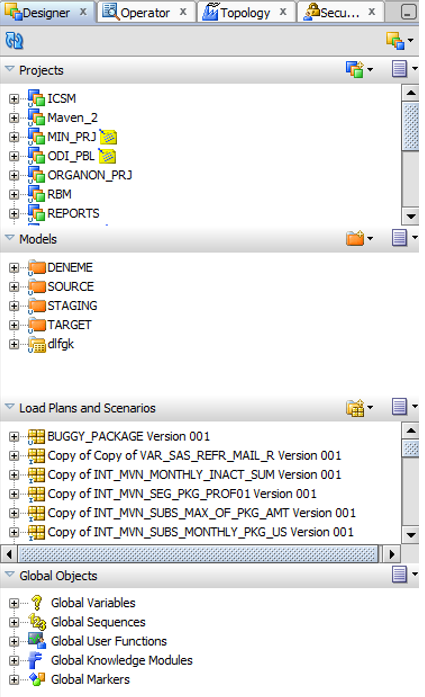
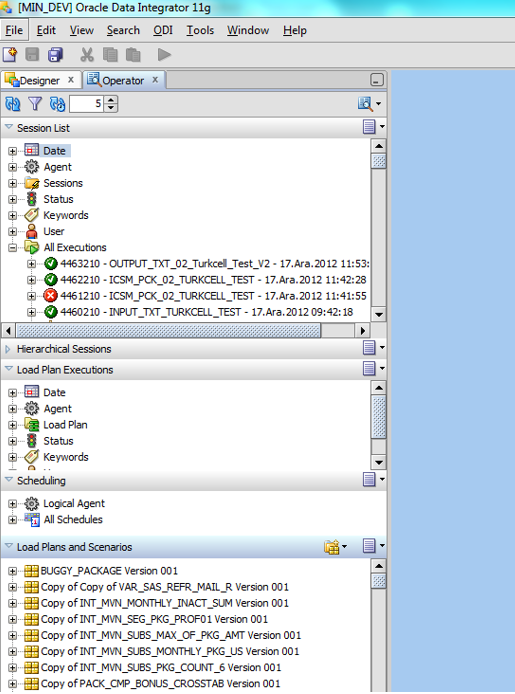
Oracle Data Integrator modülleri çoğu bilgisayar uzmanları için (arabirim  yöneticileri, programcılar, veri yöneticileri, vb) Tasarlanmıştır.

4 tane  grafik Modülü vardır.   Designer,  operator,  Topoloji Yöneticisi, ve güvenlik yöneticisi. Bu moduller    Java Virtual Machine 1.5(J2SE)  , grafik platformunda kurulabilir.windows uygulamaları, Linux , HP-UX, Solarıs, AIX, ve Mac OS diğerlerinin ortasında dahil edilebilir.

Bu Topolojilerin grafiksel pozisyonlarına şekilden genel olarak ulaşabiliriz.

**Designer**data  dönüştürümü ve  data  güvenliği için deklaratif kuralları  tanımlar  Tüm Proje geliştirmeleri  bu modulde alan götürür.  Bu alan   database ve uygulama  metadata  getirtilir ve tanımlanır. Designer modul    üretim senaryoları  oluşturmak  için kurallar ve metadata kullanır. Bu  Metadata yönetimi ve geliştirimi için  işin özündeki ve önemi ciddi seviyede olan bir moduldur.

**Operator** üretimi denetler ve yönetir. Üretim operatoru  için design edilmiştir ve uygulama logaritmalarını hatalı sayımlar ile gösterir

**Topology manager**  altyapı sisteminin  fiziksel ve mantıksal mimarisini tanımlar.Serverlar, Şemalar, ve araçlar bu modulde  genellikle altyapı sisteminin veya proje  yöneticileri kayıtlıdır.

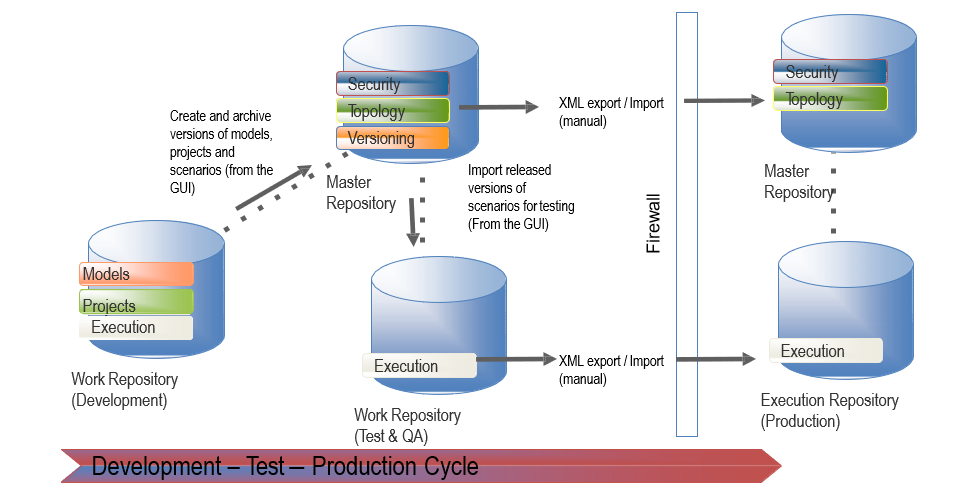
**Security Manager**  kullanıcı profilleri ve onların özel erişimlerini yönetir.Bu modul genellikle  güvenlik yöneticileri tarafından kullanılır

Oracla Data Integrator(ODI) 11g programındaki master-work repositorylerinin kullanımı hakkında örnekler yaptık ve, oluşturduğumuz veriable ve senaryoları tek bir package’da topladık.

**Data model,** data server'da store edilmiş tablo veri yapısı grubuna karşılık gelmektedir. ODI topology sekmesinde tanımlanmış mantıksal bir şemaya dayanmaktadır ve sadece metadata bilgisini içermektedir.

**ODI'da 2 tip repository mevcuttur.  
Master repository:** Kurum yada kuruluşlar hakkında bilgilerinin tutulduğu  repository'dir. Şirketin IT kaynakları, güvenlik bildirimleri, veri modeli ve versiyonlama bilgileri tutulur.

**Work Repository:** Data model ve project'ler üzerinde bulunana bilgileri içeren bir veri yapısıdır. Repository ilişkisel bir veritabanı üzerinde depolanır. Bir master repository üzerinde birden fazla work repository oluşturulabilir ama bir work repository birden fazla master repository ile ilişkili olamaz.



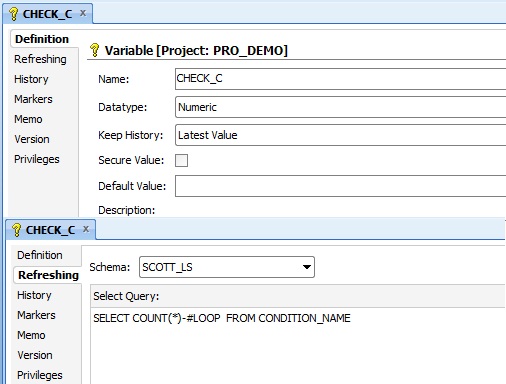
Repository'i oracle veritabanı düşündüğümüzde aslında bir şemadır. Bu şema altında ODI repository'leri kurulduğunda kendisinin kayıt atacağı tabloları oluşturur.

**Global ve project olmak üzere 2 tip variable vardır.**

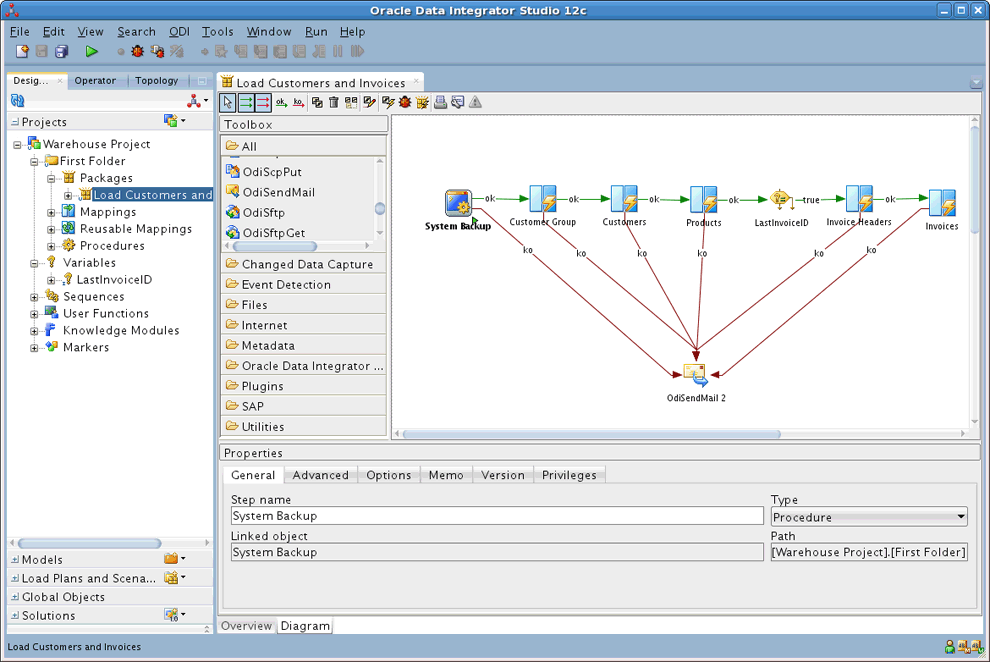
**Variable**, ODI'da tek bir değeri depolayan bir object'dir. Bu değer string, number ve date olabilir. Bu variable, değeri  bir mantıksal şema üzerinde çalıştırılan sorgu sonucuna göre update olabilir. Örnek olarak database'den mevcut tarihi ve saati alabilir. Bir variable global olarak veya project içinde oluşturulabilir. Global variable tüm proje'lerde kullanılabilir ama project variable sadece oluşturulduğu projede kullanılabilir.

Variable'lar tüm oracle data integrator expression'larında kullanılabilir.

* Mapping
* Filters
* Joins
* Constraint

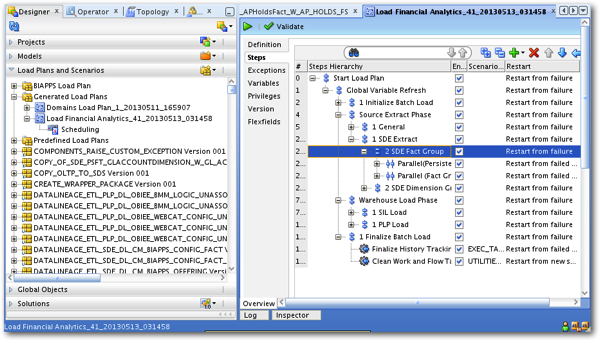
veriable oluşturma ekranı.

**Package**, ODI'daki en büyük execution unit'tir yani bir package ile birden fazla işi tek seferde execute edilebilir Örnek olarak birden fazla mapping'i tek bir package'te aynı anda execut etme.



Bir package, mapping, variable veya procedure component'i bittiği zaman bir **scenario**'da derlenir.  Scenario, production için execution etme birimidir. scenario schedule edilebilir.

**Load plan**,  oluşturduğun scenario'ların execution'larını sırayla, paralel veya koşula bağlı olarak bir çok scenario'yu execute etmek veya çalıştırmak için bir process'tir



**E$**: ODI tarafından oluşturulan geçici bir hata tablosudur. Check Knowledge Module(CKM), tarafında oluşturulur. CKM'e takılan kayıtlar bu tabloya insert edilir.

**J$**: Tüm değişikliklerin kaydedildiği tablodur. Değişilik tipi(insert,update,delete) ile beraber değiştirilen kayıtları tutar Journalizing Knowledge Module tarafında oluşturulur.

**I$**: Integration Knowledge Module(IKM) tarafında oluturulur. LKM,veriyi C$ tablosuna yükledikten sonra IKM, I$ tablosuna yükler.

**C$**:Loading Knowledge Module(LKM) tarafında staging area'da oluşturulur. Source'dan yüklenecek verileri geçici olarak tutar.

.

**Knowledge Module tipleri**:

**Reverse Enginering Knowledge Module (RKM):**Bu modül, Source database'den tablo ve diğer objelerin metada bilgilerini okumak için kullanılır.

**Loading Knowledge Module (LKM):** Source database'deki verileri,  staging area'ya yüklemek için kullanılır.

**Integration(Load) Knowlege Module (IKM):** Staging area'daki verileri target tablolara yüklemek için kullanılır.

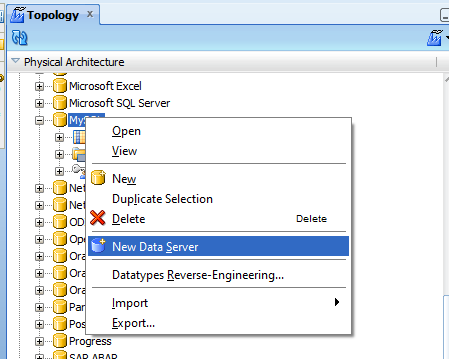
**Check Knowledge Module (CKM):** Veri tutarlılığını kontrol etmek için kullanılır.

**Journalizing Knowledge Module (JKM):**Değişikliği takip etmek için source database'den  veri değişikliklerinin(update, delete,insert) günlüğünü oluşturmak için kullanılır.

**ODI’da Physical Schema- Logical Schema-Context zincirini, datanın ODI’a bağlanma sürecini adım adım işledik.**

**Physical Schema- Logical Schema-Context**

Herhangi bir fiziksel bağlantıyı ‘Physical Architecture ‘ olarak tanımlamak için bir Data Server oluşturmanız gerekir. Physical Architecture'de istediğiniz teknolojiye tıklayın ve New Data Servera tıklayın.



Bundan sonra, örnek ayrıntılarını vermeniz gerekir. Herhangi bir Veritabanı örneğinde, ana bilgisayar, kullanıcı / şifre ve JDBC ayrıntıları sağlayın

Örnek olarak Oracle veritabanı olması durumunda JDBC Detayları girilir.

Herhangi bir dosya durumunda. Sadece ham dosyaları arabirimden seçeceğiniz klasör yolunu sağlayın.

Şimdi Data Server ekledikten sonra, bir sonraki adım data servera Physical Schema eklemektir.

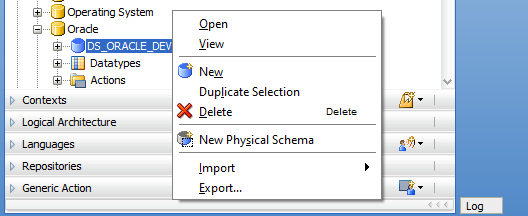
**Physical Schema**, veri kaynağına gerçek bağlantıyı gösterir (herhangi bir veritabanı örneği veya herhangi bir dosya sistemi).

ODI Physical Schema bir çift şemaya karşılık gelir:

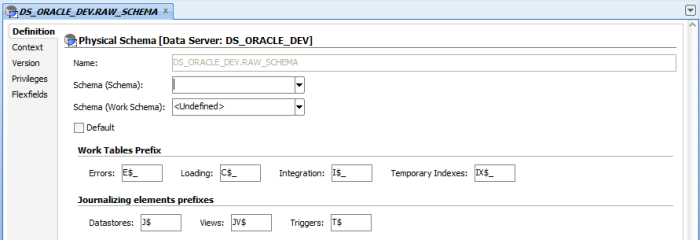
Main Şema (data) - ODI'nin arayüzler için kaynak ve hedef nesneleri aradığı şey.

Work Schema - Genel olarak ODI tarafından evreleme amacıyla kullanılır, burada ODI kaynaklar ve hedeflerle ilişkili geçici veri nesneleri oluşturur. Örnek C $, I $, E $ tabloları. Ayrı iş şeması oluşturmak ve kullanmak her zaman tercih edilir.

Herhangi bir fiziksel şema tanımlamak için oluşturduğunuz Veri Sunucusuna gidin ve ardından yeni " New Physical Schema " yı tıklayın.

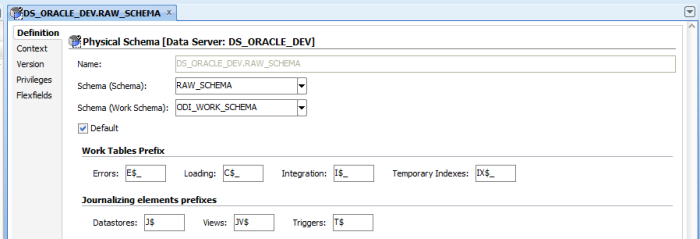


Bundan sonra açılır menü kullanarak Schema Name & Work Schema adını vermeniz gerekir. Seçilen DATA SERVER örneğinde mevcut tüm şema gösterecektir.



RAW\_SCHEMA - kaynak / hedef tablolarımın bulunduğu şema.

ODI\_WORK\_SCHEMA - ODI evreleme amacıyla oluşturuldu.

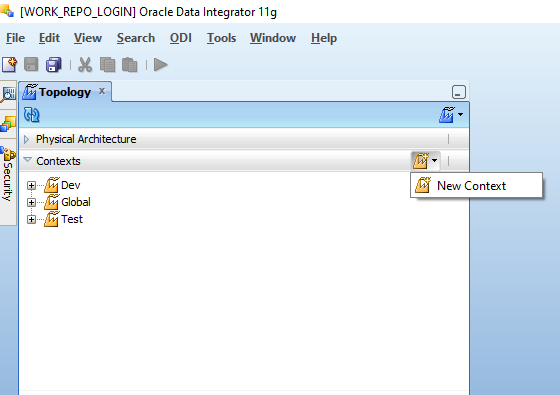


Şimdi **Physical Schema** ekledikten sonra, bir sonraki adım **Logical Schema** ve **Context** oluşturmaktır.

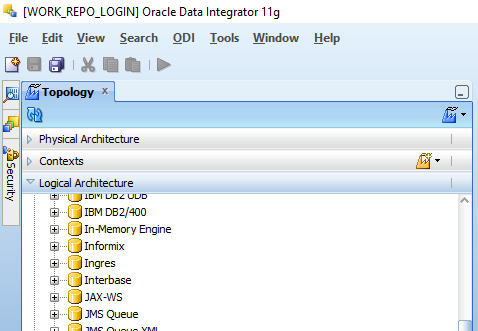
**Logical Schema**, bu kaynak / hedef nesnesiyle ilişkili mantıksal adı temsil eder.

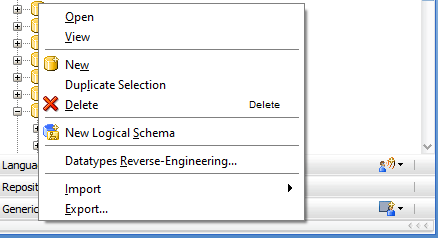
Bir **Logical Schema**, **Context** birlikte çoklu fiziksel şema ile ilişkilendirilebilir, yani bir **Logical Schema**, farklı **Context** kullanarak farklı **Physical Schema** ile ilişkilendirilebilir.

Herhangi bir yeni **Context** eklemek için, **Context** Bölümüne gidin ve ardından yeni **New** **Context** tıklayın.

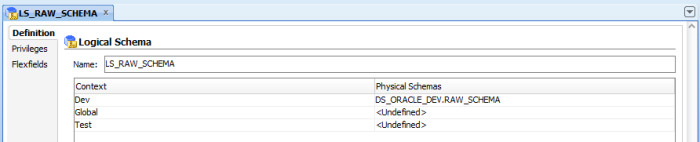


**Logical Schema** Logical Architecture define içerisinde tanımlamak. İstediğiniz teknolojiye tıklayın ve New Logical Schema tıklayın.





Bundan sonra **Logical Schema** adı ve **Context** ile **Physical Schema** eşlemesi sağlamanız gerekir.



Dolayısıyla, Dev Contextinde **Logical Schema** LS\_RAW\_SCHEMA, **Physical Schema** DS\_ORACLE\_DEV.RAW\_SCHEMA'yı işaret ediyor.

**Logical Schema** LS\_RAW\_SCHEMA, Test Contextindefarklı **Physical Schema** DS\_ORACLE\_TEST.RAW\_SCHEMA'yı işaret ediyor olabilir.

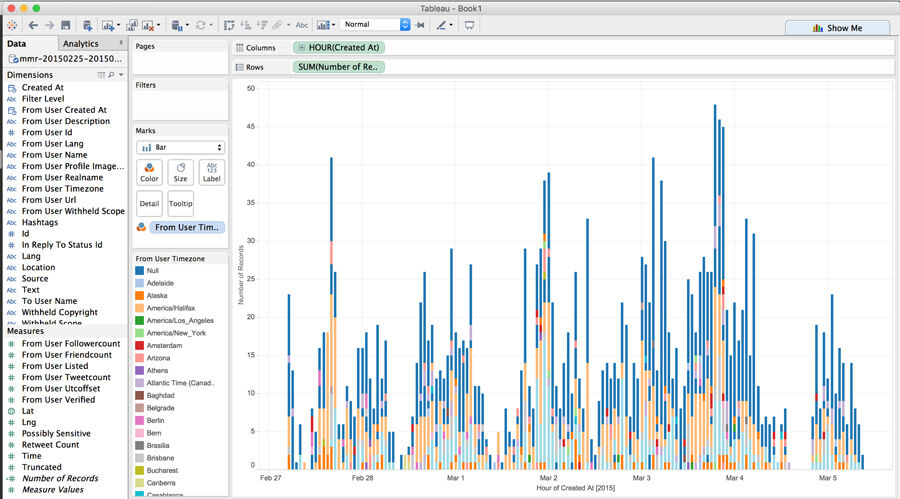
Eğitimimizin bugününde Halil Baysal / Consultant *tarafından şirketlerin büyük datalarının raporlanması, analizi için kullandıkları iş zekası uygulaması* **Oracle**, [Business Intelligence Foundation Suite](http://www.oracle.com/technetwork/middleware/bi-foundation/overview/index.html) *gösterildi ve iş zekası analitiği ayrıntılı bir şekilde işlendi.*

, İŞ ZEKASI VE ANALİTİĞİ:

Operasyonel işlemler, satış kanalları, pazar değişimleri, tüketici beklentileri ve geri bildirimleri, rakiplerin tutumları için sayısız farklı noktadan sayısız ve sürekli bir veri akışı gerçekleşiyor. Gerçekleşen bu veri akışı içerisinde anında değerlendirmeler yapabilmek ve bu değerlendirmelerin sonucu olarak tepki verebilmek için saniyelerle yarışılmakta. Genellikle farklı yazılımlar üzerinden elde edilen bu çıktıların derlenmesi, yönetimi ve analizi sürecinde daha pratik ve entegre çözüm arayışları söz konusu. “**Big Data**” olarak ifade ettiğimiz bu veri yığınının hızlı ve etkili bir şekilde uygun sorular etrafında geçerli cevaplar oluşturulabilmesi şirketlerin bilgi/analiz ve raporlama birimlerine yoğun bir iş yükü anlamına gelmekte. Elbette verinin anlamlandırılması aşamasında da soruların doğru sorulması ve duruma göre yeni sorularla verilerin test edilmesi de oluşturulacak değerlendirme ve strateji planlamaları için kritik bir öneme sahip. İşte bu noktada verilerin anlamlandırılarak bilgiye dönüştürülebilmesi esası yatıyor

İhtiyaç duyulan veriye erişmeyi ve bu veriyi bilgiye dönüştürmesi sağlayan iş zekası uygulamalarına başta finans olmak üzere bir çok sektörde ve her ölçekteki firma tarafından ihtiyaç duyulmaktadır.

Verimli bir iş yönetiminin karşılığı olarak rekabet ortamında öne geçmek ve avantajları erken fark etmek için kullanılan en önemli ve yaygın olan **Oracle**, [Business Intelligence Foundation Suite](http://www.oracle.com/technetwork/middleware/bi-foundation/overview/index.html) ile veri depolama sürecinden analiz aşamasına kadar tüm ihtiyaçlara çözüm üretme hedefiyle hareket ediyor. Yer alan diğer uygulamalar ise platform, kullanım kabiliyeti, özelleştirilebilme gibi değerlendirmeler içerisinde liderliğe sahipler.



### **Data Visualization in OBIEE 12c**

Data visualization, verilerin resimsel veya grafiksel bir formatta sunulması ve kolayca göz ardı edilebilecek karmaşık bir problemi önemser ve grafik, desen ve tasarım kullanarak işleri kolaylaştırır. Gerçek zamanlı verileri zengin raporlara ve görselleştirmelere dönüştürmede faydalıdır. Etkili veri görselleştirme, bilgilendirici, verimli, çekici ve bazı durumlarda etkileşimli ve öngörücü olmalıdır.

Görsel olarak sunulan analizleri görmemizi sağlar; bu da zor kavramları kavramamıza / yeni modelleri kolayca tanımlamamıza yardımcı olur.

Data visualization faydaları:

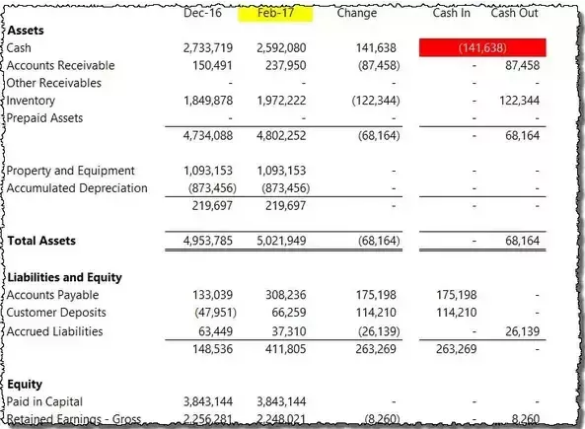
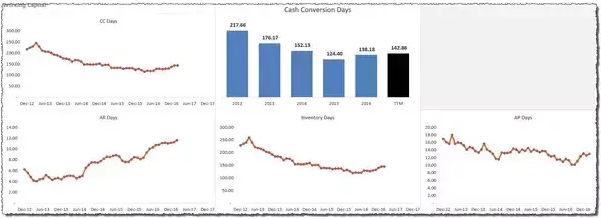
1. Bilgileri yeni ve daha yapıcı yollarla özümseyin.

2. Operasyonel ve ticari faaliyetler arasındaki ilişkileri ve kalıpları görselleştirmek

3. Yükselen trendleri tanımlayın ve daha hızlı hareket edin

4. Verileri yönetin ve doğrudan etkileşimde bulunun

5. Yeni bir iş dilini destekleyin



Geleneksel raporlamayı gösteren Şekil1deki tabloyu ve her çizgiyi Şekil2deki gibi görselleştirme yeteneğini düşünün. Basit bir görselleştirme, sıkıcı verilerin canlanmasına neden olabilir:

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  |

|  |
| --- |
|  |