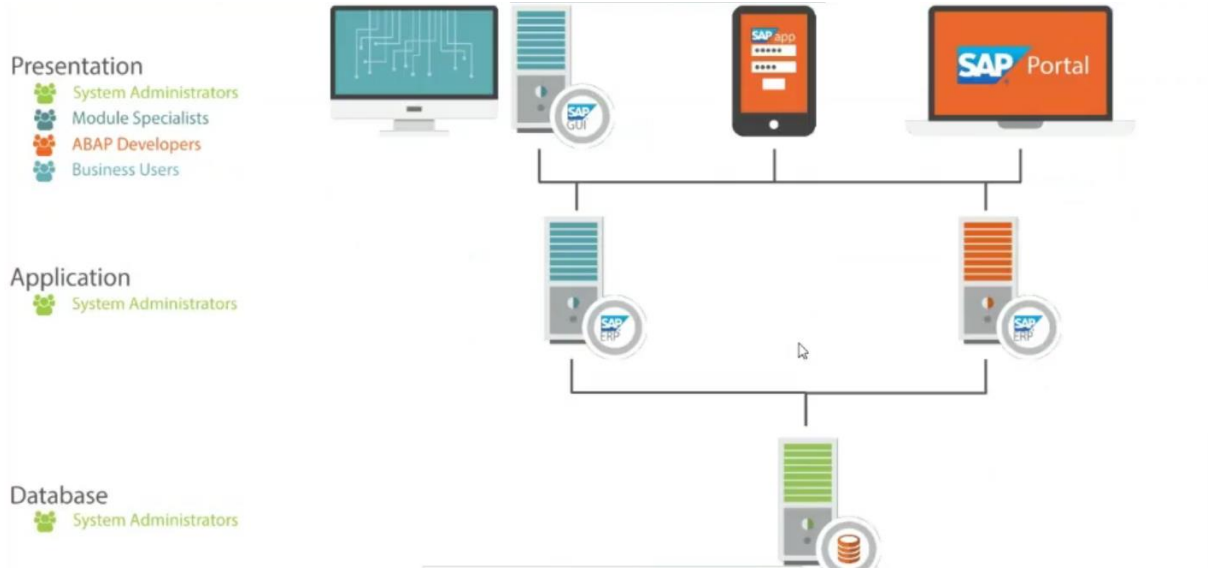
SAP

SAP nedir: SAP 1972 yılına (“System Analysis and Program Development”) adıyla Almanya’da kurulmuş Avrupa’nın en büyük yazılım şirketidir.

SAP nelerden oluşuyor?

Ama bu unsurlardan en önemlisi SAP ERP’dir.

SAP architecture:



Burada görüldüğü gibi SAP architecture 3 kısımdan oluşuyor.

1- **Presentation:** Sunum katmanındaki mimarinin en üstünde, yalnızca bir bilgisayarda, akıllı telefonda, tablette ve hatta dizüstü bilgisayarda çalışabilen farklı istemciler bulunur.
   Bu cihazlarda çalışan istemciler, SAP web tabanlı SAP portalı ,SAP GUI uygulaması olabilir.
   En yaygın katmandır, sistem yöneticileri, modül uzmanları, ABAP geliştiricileri iş kullanıcıları gibi çoğu kullanıcının çalıştığı yer burasıdır.
   a. System Administrators: Sistem Yöneticisi, geliştirilen kod hazır olunca bir katman diğer katmana yönlendiriyor.

b. Modul Specialists: SAP modüllerin uzmanları, şirket içerisindeki iş süreçlerine göre bu modüllerinin uyarlama parametrelerin düzenlendiği yerlerdir(iş kurallarının düzenlendiği yerlerdir).
(aşağıda detaylıca bilgi var modül uzmanı hakkında).
c. Abap Developers: Abap kodu yazan kişidir.
d. Business Users:

2- **Application**: SAP ERP'nin gerçekten kurulduğu ve uygulama programlarımızın fiilen yürütüldüğü yerdir.
Uygulama sunucuları, hem veritabanı sunucusuyla hem de sunum katmanındaki istemcilerle iletişim kurmaktan sorumludur.
Uygulama katmanında orta bölümde bir veya birden fazla uygulama sunucumuz olabilir. Sistem Yöneticisi bu katmanda çalışabilir ve uygulama sunucusuna erişebilir.

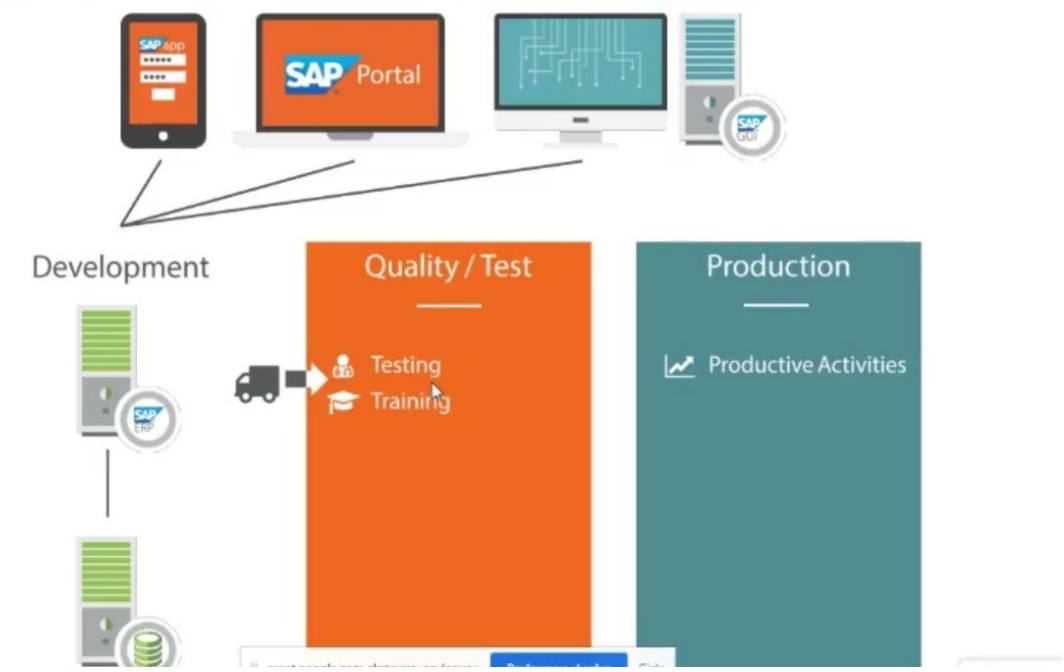
3- **Database**: finans, kontrol, İK ve hatta depolama gibi çok çeşitli ticari faaliyetlerden gelen verileri depolamayı amaçlayan bir veritabanımız var.
Bu veritabanı genel bir MS SQL, Oracle veya hatta yeni nesil veritabanı teknolojisi, SAP'nin bellek içi çözümü SAP HANA olabilir. Sistem Yöneticisi bu katmanda çalışabilir ve veri tabanı sunucusuna erişebilir.

Niçin bu katmanlara ihtiyaç duyulmuş:

1- Flexibility: esneklik sağlamak yani eğer veritabanı değişecekse sadece veritabanı değiştirerek bu işlemi gerçekleştirebiliyoruz.
2- Security: request ile biz bu katmanlar arasında değişikliği aktarıyoruz. Yani mesela development ile application arasında bir request alırken şunu kontrol ediliyor: geliştirici bu geliştirmeyi neden yaptığı, bu geliştirmenin içinde ne olduğu ve nerelere etkilediği etiketlenir.
3- Performance: aynı zamanda farklı programlar farklı katmanda çalışabiliyor.

Not: çalışmalar SAP GUI'den başlıyor yani şipariş geldiğinde

SAP Landscape:



Burada da Sap Landscape 3 aşamadan oluşuyor:

1- **Development:**
   - iş kullanıcısının ihtiyaçlarını karşılayan, mevcut işlevleri değiştiren veya ABAP programlama dilinin yardımıyla yenilerini geliştiren ABAP geliştiricileri,
   - oluşturulmuş standart işlevleri etkinleştiren veya sistemin davranışını iş gereksinimlerine göre özelleştiren modül uzmanları,
   - sistem performansı veya depolama alanı gibi ve gerektiğinde müdahale eden, sistem yöneticileri tarafından kullanılır.

   Geliştiriciler, danışmanlar veya yöneticiler geliştirme sistemindeki işlerini bitirir bitirmez, bir geliştirme paketi veya taşıma talebi olarak sistem yöneticisi tarafından kalite veya test sistemine aktarılır (kopyalanır.)

2- **Quality/ Test:** Bu sistemde herhangi bir geliştirme yapılmaz, test ve eğitim amaçlıdır.
   - Anahtar iş kullanıcısı testlerin başarılı olduğunu söylediğinde sistem yöneticisi, test edilen işlevleri veya özellikleri içeren taşıma paketini yayınlar, canlı sisteme aktarılır (kopyalanır).

3- **Production:** Genellikle sistemin durumunu izleyen ve gerektiğinde müdahale eden sistem yöneticileri ile finansman, depolama, İK, veya diğer faaliyetlerin yürütüldüğü sistem.

Buradaki avantaj :

Abap geliştiricisi, modül uzmanı, iş kullanıcılarının çok farklı sorumlulukları var ve aynı anda tamamen farklı faaliyetler yürütüyorlar; bu sistem ortamı sayesinde birbirlerinin süreçlerine zarar vermeden birlikte uyum içinde çalışabilirler.

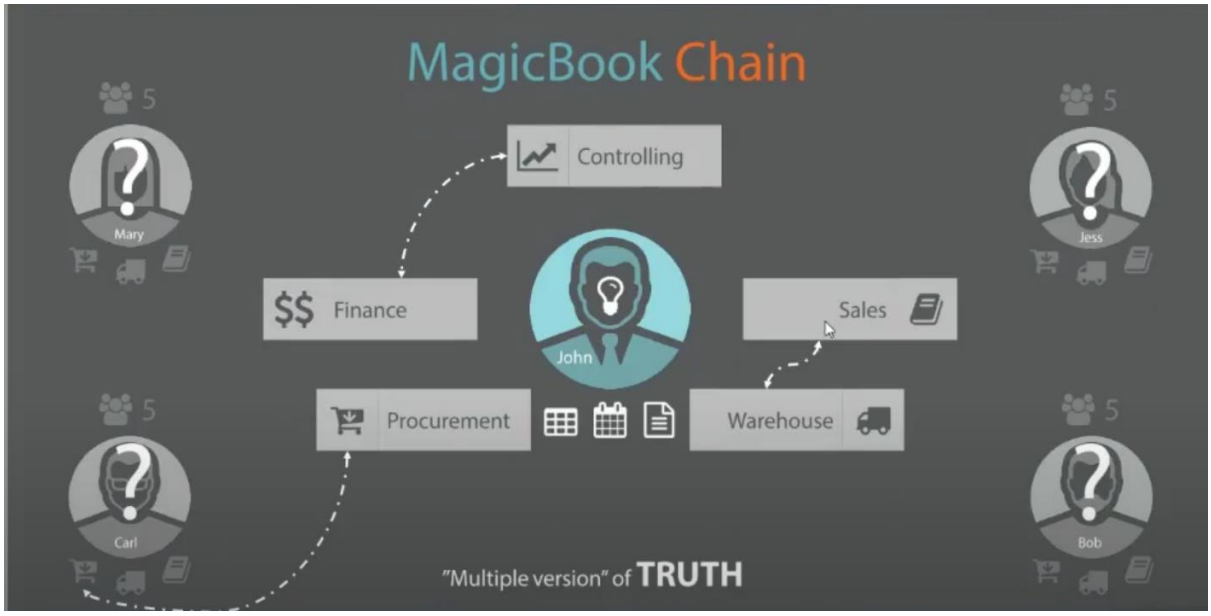
Kullanıcıların birbirlerini engellemeden aynı anda üretken iş faaliyetlerini geliştirmelerini, test etmelerini, öğrenmelerini ve yürütmelerini sağlamak.Şuanda ERP'e gelelim.

ERP nedir ve buna neden ihtiyacımız var?

ERP(Enterprise Resource Planning): iş yerinde olan farklı iş birimlerinin birlikte sorunsuz çalışabilmesi için kendi aralarında yaptıkları çalışmalarla tek bir sistem üzerinden bu işlerin sağlanabilmesi için kullanılıyordur.

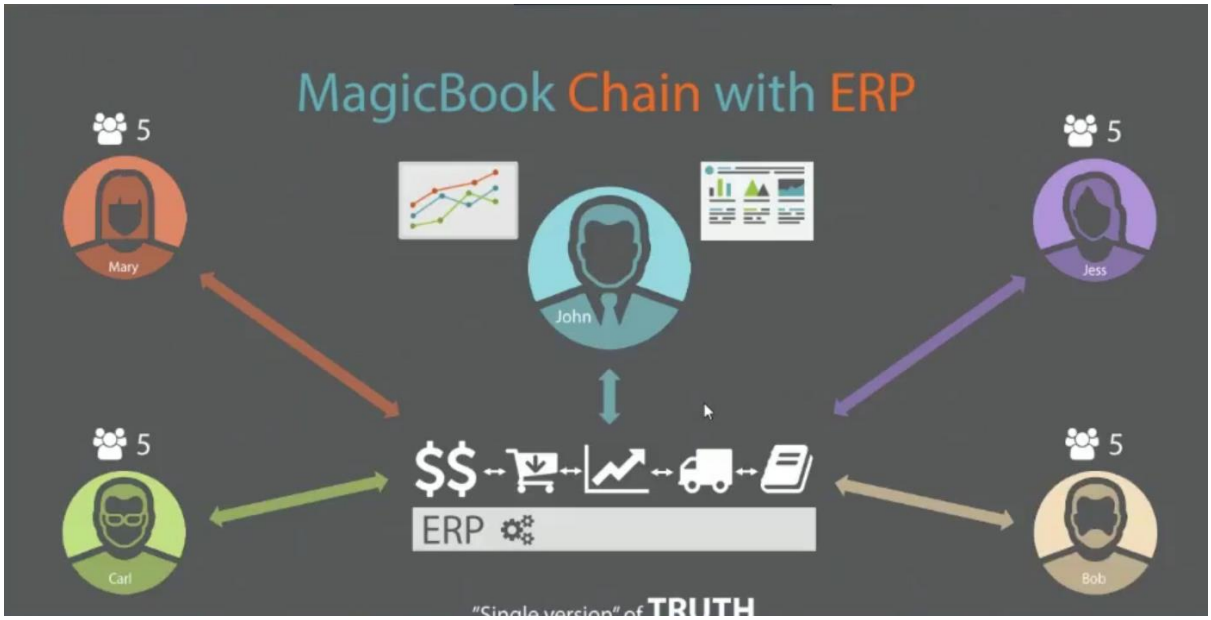
ERP'siz bir sistem nasıl çalışıyor?

Örnek üzerinden anlatmak: John arkadaşımız kendine küçük local bir kitap şirket açtı ve şirketin bütün birimleri (kitabın satışı, alışı, depolaması, maliyeti, materyelleri, karı..) kendi başına idare ediyordu.



Zamanla John şirketi büyüdü. Büyünce de farklı şehirlerde ofisler açtı ve her bir ofisine ait (kitabın satışı, alışı, depolaması, maliyeti, materyelleri, karı..) birimleri oldu. Bu durumda şöyle bir sorun çıktı. Her ofisin birimleri ayrı bir yerde olduğu için diğer ofislerle entegrasyon olamıyordu ve böylece şirketin verileri yanlış çıkıyordu. Yanlış veriler, yanlış sonuçlara yol açıyordur.

ERP'li bir sistem nasıl çalışıyor?



Burada ise John şirketi büyünce ERP'li sistem kullandı. Şuanda John birimleri ve diğer ofislerin birimleri merkezi bir sistemde olduğu için (kitabın satışı, alışı, depolaması, maliyeti, materyelleri, karı..) birimleri takip edebiliyor ve doğru veriler alıyor. Doğru veriler, doğru sonuçlara yönlendiriyordur.

Kısacası:

ERP sistemleri, bir şirketteki kuruluşların finans, İK, tesis bakımı, kalite yönetimi, depolama veya daha fazlası gibi çok çeşitli ticari faaliyetlerden verilerini tek ve merkezi bir veritabanında toplamasına, yönetmesine ve paylaşmasına yardımcı olmayı amaçlamaktadır.

Entegrasyon vardır, yalnızca bir tek doğru sağlar.

Merkezi ve entegre veri tabanı ve işlemsel veriler sayesinde güçlü ve zayıf yönlerimizi keşfetmek için süreçlerimizi takip edip ölçebiliyoruz.

Kaliteli verileri toplayarak, kontrol ederek, izleyerek, ölçerek ve temel süreçleri otomatikleştirerek işimizin kalitesini ve verimliliğini artırabilir ve böylece işimizi daha iyi yürütebilir ve pazar lideri olabiliriz.

SAP ERP'nin birden fazla modeli vardır.

SAP MM(Material Managment): mal hareketlerinin yapıldığı, satışalma süreçlerinin yürütüldüğü ve fatura süreçlerinin yürütüldüğü modüldür.

SAP PP(Production Planning): farklı lokasyonlarda ürün üretilmesi ve ürünlere ait ihtiyaçların planlanması işlemlerinin yürütüldüğü modüldür.

SAP SD(Sales and Distribution): sipariş, sevkiyat, nakliye ve faturalama süreçlerini kapsamaktadır.

SAP AA(Asset Accounting): duran varlıkların yönetilmesi, kontrol edilmesi, duran varlıklara ait amortismanlarının hesaplanması ve rapolanması için kullanılır.

SAP BPC(Business Planning and Consolidation): satış, üretim, satınalma, maliyetlendirme, genel giderler, yatırımlar, finansman, mali tablo planlama süreçlerinin bütçeleme işlemlerini tek bir platformda gerçekleşmesini ve değişen makroekonomik varsayımları, satış öngörülerini, farklı tedarik alternatiflerini dikkate alarak bütçe simelasyonlarını yapmak amacıyla kullanılır.

SAP CO-PA(Controlling - Profitability Analysis(Karlılık Analizi)): ör: kanal müşteri bazında karlılık analizi için gerekli işlemlerin ve ilgili raporlamanın yapıldığı modüldür.

SAP APO-DP(Demand Planning): ürünlerin ihtiyaçlarını tahmin etme modüldür.

SAP FI(Financial Accounting): şirketin mali mevzuatlarına uygun olarak bütün muhasebe hareketlerinin toplandığı, ve bununla ilgili gerekli kontrollerin yapılarak raporlamaların (şirkete ve resmi kurumlara) yapıldığı modüldür.

SAP TM(Transportation Managment): ürünlerin bir noktadan diğer noktaya fiziksel olarak taşınması ile ilgili tüm aktiviteleri kapsamaktadır.

SAP MII(Manufacturing Intelligence, Integration(Üretim Zekası, Entegrasyon)): üretim sahasında veri toplamayı ve raporlamayı sağlayan modüldür.

SAP SRM(Supplier Relationship Management(Tedarikçi İlişkileri Yönetimi إدارة العلاقة مع الموردين)): Burada satınalma talebi oluşturma, teklif toplama, satınalma siparişi oluşturma ve yönetimi, tedarikçi bilgi yöntemi.. göre kullanılmaktadır.

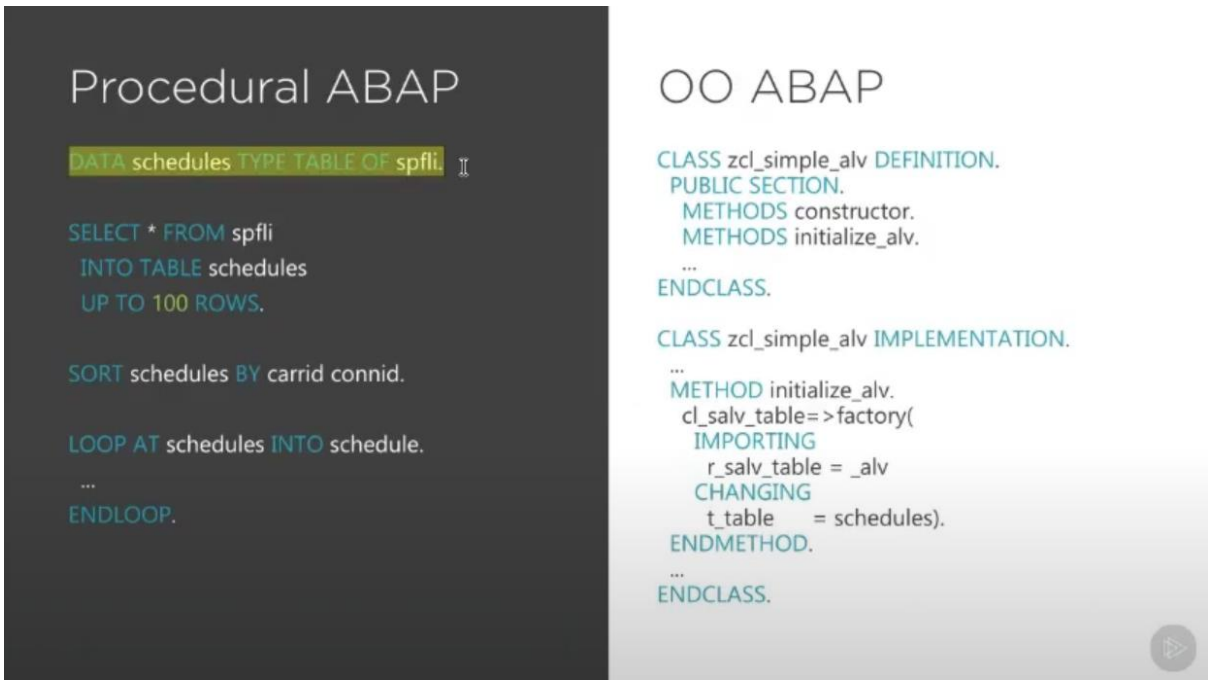
SAP WM(Warehouse Managment): mal girişinden çıkışına kadar depo içerisindeki tüm mal hareketlerinin yönetilmesini, gözlenmesini ve denetlemesini sağlayan modüldür.

SAP ERP Core Application: modüller şirkette genelde 3'e ayrılır.

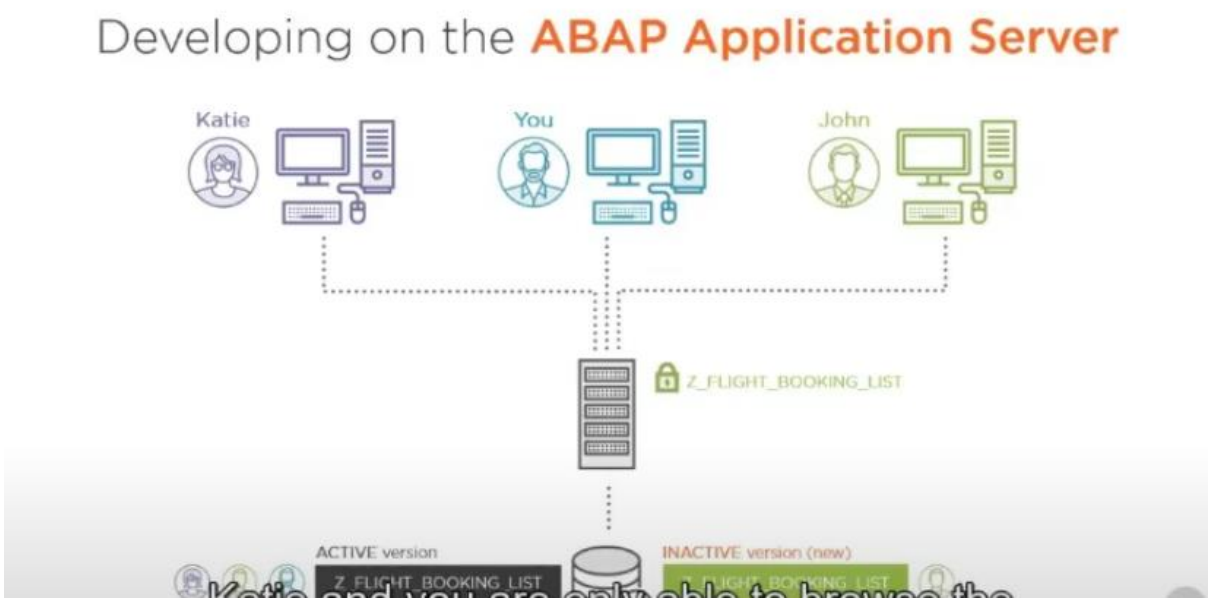
1- Logistics: financal sürecinde input sağlayan ve daha çok operasyonal süreci takip eden modüllerdir.
2- Finance: bu modüllerde en temeli finans (FI) yani finansal hesapların tutulduğu modül, bu yönde de CO(Management Accounting) dediğimiz daha çok maliyet hesapların yapılımı ve yönetimi takip ettiği modüldür.
3- Human Capital Management: burada ise personal yönetimi ile ilgilenen modüldür.

Şuanda ABAP ne olduğunu tanımlama zamanı geldi.

ABAP(Advanced Business Application Programming): SAP şirketi tarafından geliştirilen üst seviye bir nesne tabanlı programlama dilidir.



ABAP hem Procedural hem de OOP destekleyen bir programlama dilidir.



ABAP'in işlemi şöyle oluyor:

ABAP developer ABAP Applicaqtion Server üzerine yazdığı kod database'e kaydediliyor.

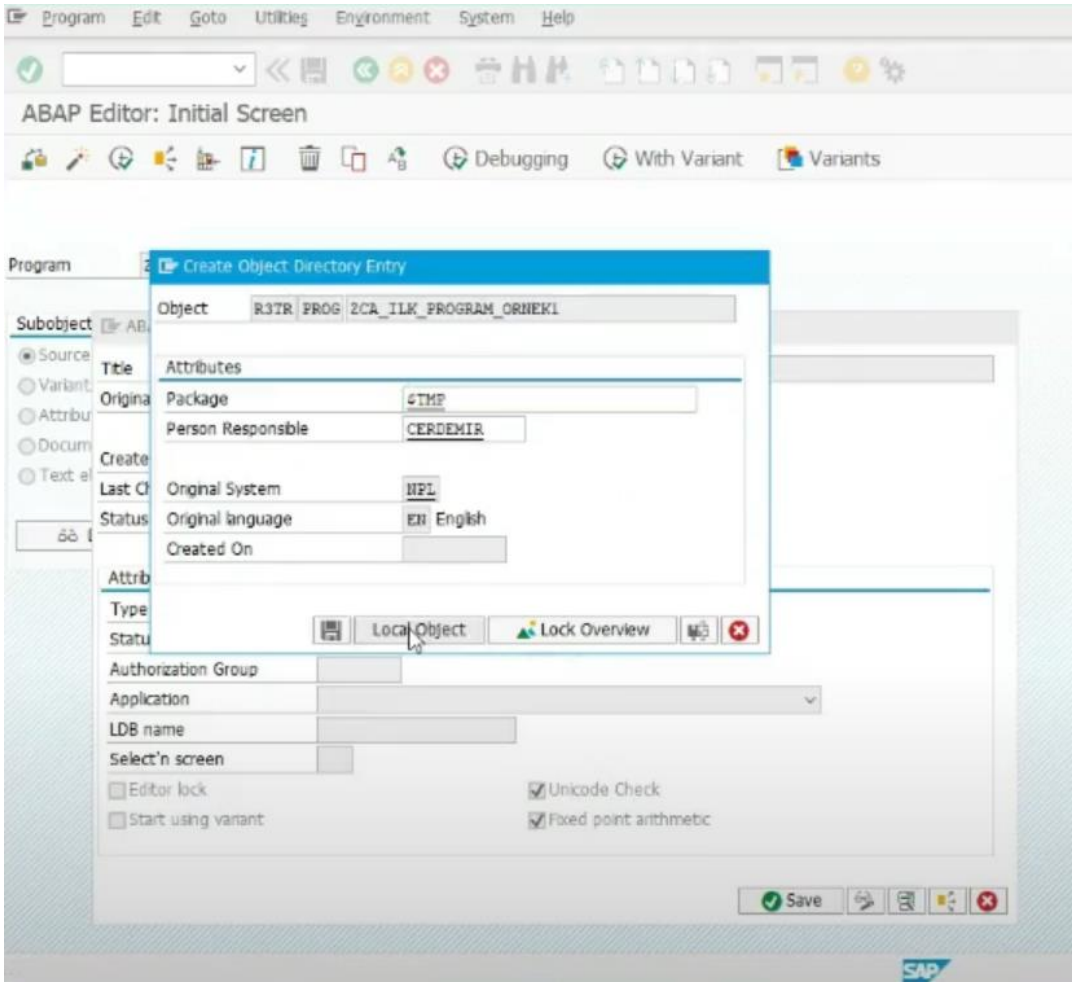
Z_FLIGHT_BOOKING_LIST adıyla bir program yazmaya başladık. Programımız bellekte oluyor ve "save" tuşuna basınca program, merkezi veri tabanına aktarılıyor ve (INACTIVE) version olarak oluyor.

Şuanda ona “compile” yapıyoruz ve programın syntax hatası yoksa executable file’i oluşturuluyor ve (ACTIVE)çalıştırılabilir hale geliyor.

Kodu INACTIVE haldeyken diğer developer onu göremiyorlar.

Kod üzerine değişiklik birden fazla developer yapamaz aynı anda.

Not: $TMP = Local Object



ABAP (Syntax)yazım kuralları:

- Kodlar boşluk ile ayırılıyorlar.
- Her komut nokta ile bitiyor.
- Büyük küçük harfe duyarlı değil.
- Veri türlerini, veri nesnelerini, sınıfları, yöntemleri veya prosedürleri adlandırmak için 30 karektere kadar sınırı var.
- Yorumlama tüm satır ise (*) ile yapılır.
- Yorumlama satrın bir kısmını ise (“) ile yapılır.
- String tek tırnak ile belirtilebilir.
- Structure için (-) kullanılır.
- Object için ise (->) kullanılır.

- (/) alt satra geçmek için kullanılır. Ör: write / ‘lorem’.
- (skip) komutu boş bir bırakmak için kullanılır.
- (new-line) komutu alt satra inmek için de kullanılır.
- (uline) komutu uzun bir çizgi çekmek için kullanılır.

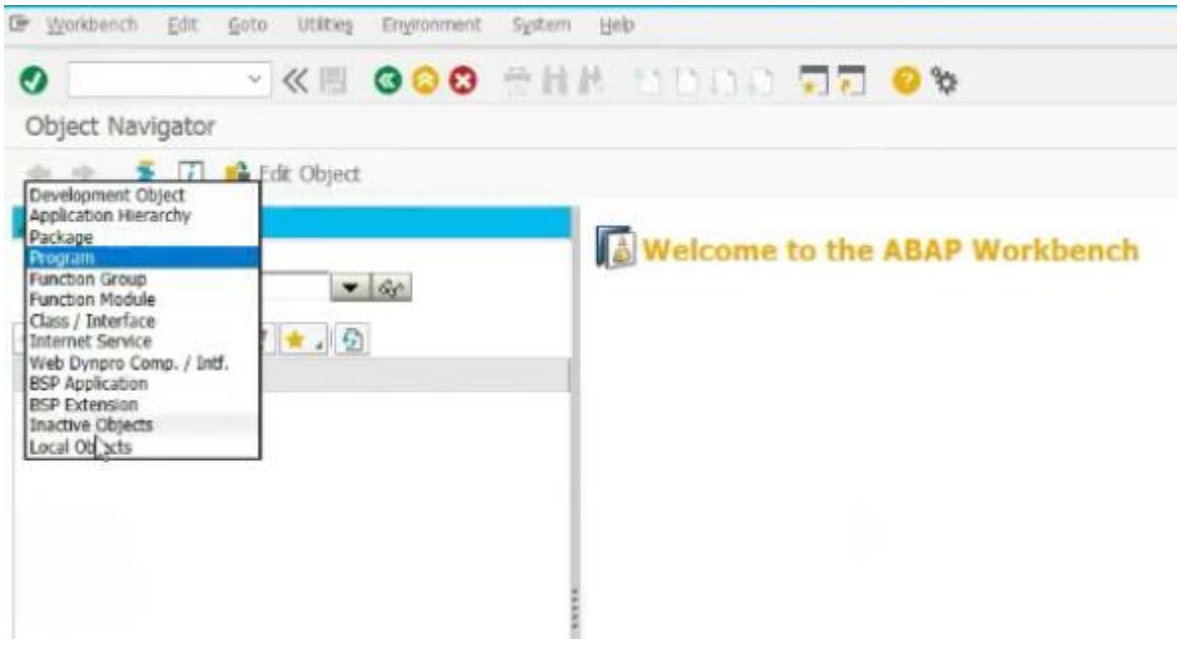
SAP GUI’de oturum kısa yollarını :

1. Manage Cursor, “CTRL” + “/”
2. Open new session, “CTRL” + “+”
3. Run Transaction in new session, /oSe80, /oVA23
4. Run Transaction in the same session, /nse80, /oVA05
5. Close current session /i
6. Close all session (with confirm) /nend
7. Close all session /nex

ABAP/4 Workbench araçları:

Abap workbench’i oluşturan araçların hepsine se80 (object navigatör) kodu ile ulaşılabilir.

Yani aşağıdaki komutların yaptığı iş bu ekran ile yapılalilir:



1- Abap Editör : Abap kod editörü. (Se38)
2- Screen Painter : Kullanıcı ekran oluşturucu (se51)
3- Abap Dictionary : Veri türü sözlüğü (se11)
4- Function Builder : Fonksiyon modülü editörü (se37)
5- Class Builder : Sınıf oluşturucusu (se24)
6- Gateway service Builder : Odata Web hizmeti (SEGW)
7- Sistemde şuanda çalılşanları göstermek. (al08)
8- Work processes of as instance. (sm50)

9- Started as instance. (se51)
10- Sistemde birden fazla kullancı olduğunda kullanılır. Tablolar lock mı değil mi bakmak için. (sm12)
11- Sistem performansı j-hakkında bilgi almak için. (st03)
12- Bir program yazdığımızda ve onu transaction code ile bağlamak için ..(se09).
13- Maintain Transaction (se93). Burada transaction codu yazarak hemen programı çalıştırabiliyoruz.
Programı transaction koda nasıl bağlanabileceğini ile ilgili açıklama Technical Skills Eğitim 3, SAP ABAP4 videonun 1:54 dakikasında.

ABAP objects standart veri türleri:

- D: Tarih (Date) : 8 karakter uzunluğunda, formatı YYYYDDMM yazıyoruz, örneğin 20210101, 2021 yılının Ocak ayının ilk gününü gösteren tarih türü veri nesnesi
- T : Zaman (Time) : 6 karakter uzunluğunda, formatı HHMMSS, örneğin 163245, saat 16, dakika 32 ve saniye 45 zaman türü veri nesnesi.
- I : Tamsayı (Integer) : Tamsayı değişken tanımlamak için kullanılır, sabit 4 bit uzunluğundadır. Örneğin -123 negatif tamsayı veri nesnesi.
- F: Ondalıklı (Float) : Ondalıklı değişken tanımlamak için kullanılır, sabit 8 bit uzunluğundadır. örneğin 32.4
- String : Değişken özelliğe sahip karakter dizisi , örneğin 'Abap Objects'.

- Xstring : değişken uzunluğa sahip Hexadecimal string tanımlamak için kullanılır. Örneğin '0X12AC'.
- C: Karakter : (Character) Veri nesnesinin uzunluğu programcı tarafından belirlenmesi gerekir. Örneğin 'ABC', karakter türüne sahip uzunluğu üç karakter veri nesnesi
- N: Numerik karakter : (Numerical Character) Veri nesnesinin uzunluğu programcı tarafından belirlenmesi gerekir. Örneğin , '06135' uzunluğu beş karakter veri nesnesi.
- P: Ondalıklı sayı (packaged Number) : Veri nesnesinin uzunluğunun ve virgülden sonra kaç basamağın bulunduğunun programcı tarafından tanımlanması gerekir. Örneğin '456.89' uzunluğu üç, virgülden sonra 2 basamağa sahip veri nesnesi.

ABAP'ta lokal veri tipleri, global veri tipleri ve sistem alanları:

- **Lokal Veri Türleri :** Abap standrt veri türleri tanımlanarak, bir abap programı içerisinde programcı tarafından tanımlanırlar. Sadece tanımlandıkları Abap programı içerisinde değişken tanımlamaları için kullanılırlar. Diğer Abap programlarından referans olarak kullanılamazlar.
Lokal veri tipleri bir ABAP programı içerisinde TYPES anahtar sözcüğü ile tanımlanır.
Örnek :

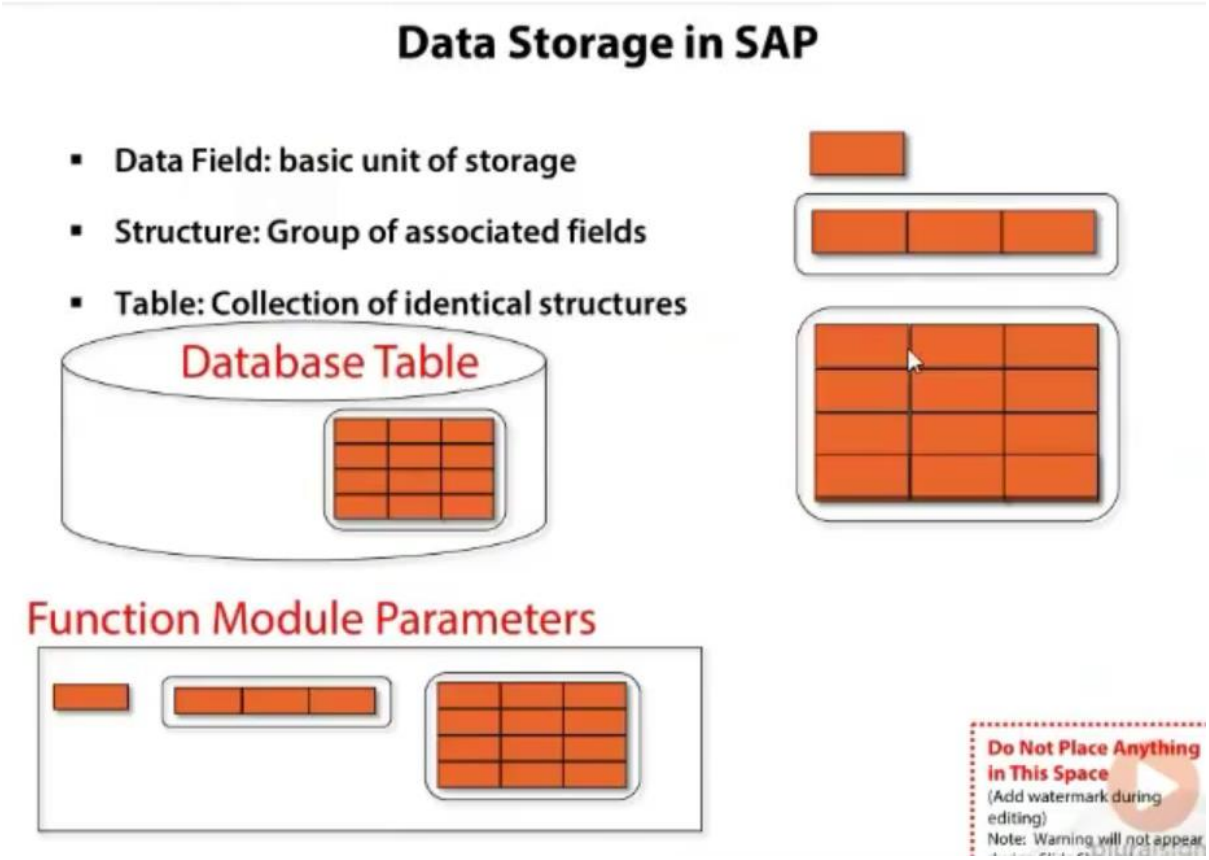
```
TYPES   : gty_faiz_oranı    TYPE p LENGTH 2 decimals 1.
```

DATA : gv_faiz_oranı TYPE gty_faiz_oranı value ‘4.5’.

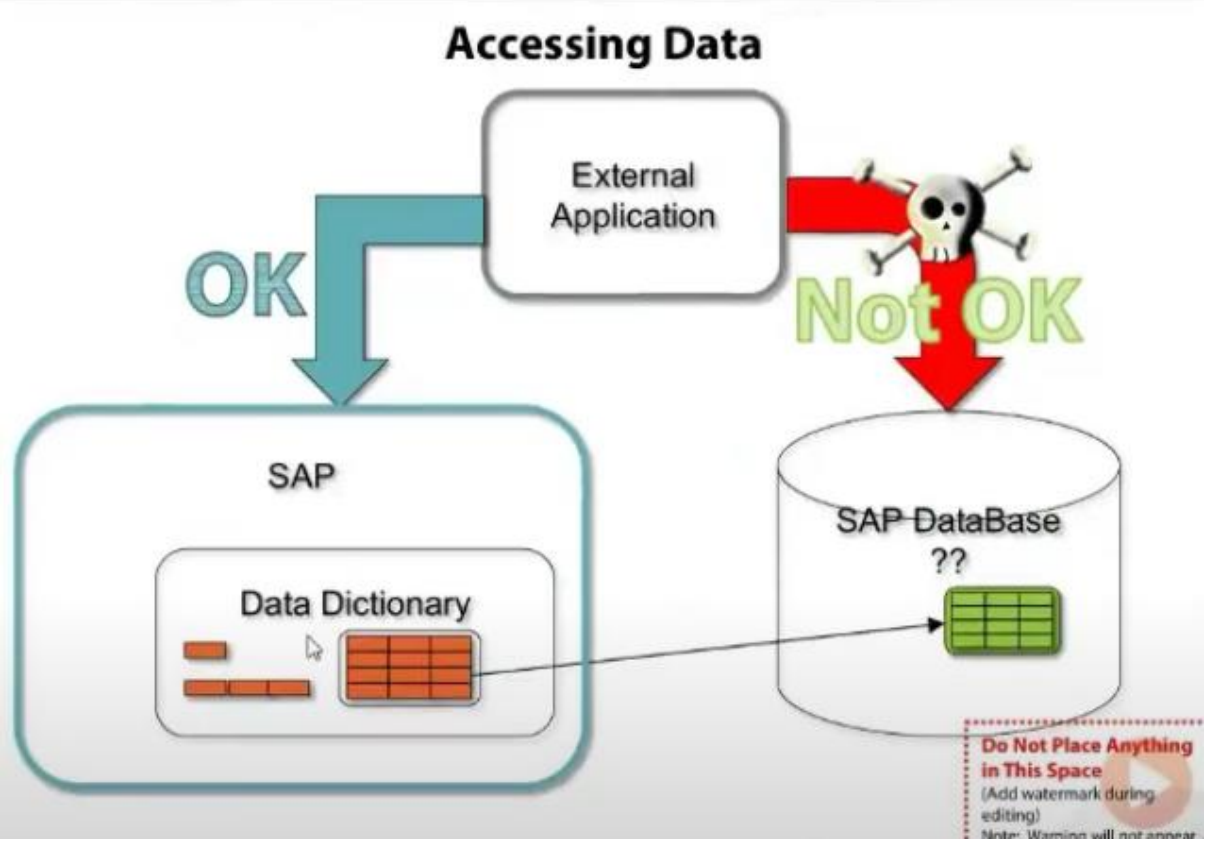
- **GLOBAL Veri Tipleri :** ABAP DDIC’te veri sözlüğü tanaımlanan ve SAP sistemi içinde bütün ABAP programları tarafından kullanılacak veri Türlerine Global veri türleri denir.

  ABAP DDIC’te bir veri elementi (Data Element), yapı (structure), İçsel tablo (internal tablo).

  Sınıf (Class) sistemde tanımlanan global sınıflar, Veri türü Havuzu (type-pool) .



Not: SAP direct database's erişimi yok. Bunun için data dictionary vasıtasıyla erişilir.



Sistem alanları:

- Sy-uname : kullanıcı adı
- Sy-langu : kullanıcının seçtiği sistem dili
- Sy-datum : günün tarihi
- Sy-subrc : Yapılan işlemin başarılı olup olmadığı
- Sy-index : Do while döngülerinde kaçıncı defa döngünün işlendiği
- Sy-repid : program ismi
- Sy-uzeit : sistem zamanı
- Sy-tcode : programın işlem (transaksiyon) kodu
- Sy-tabix : içsel tablo döngülerindehangi tablo satırının işlendiği

Ex: gv_date = sy-datum. => sistemin tarihini bize verir.

ABAP veri sözlüğü (ABAP dictionary) hakkında kısa bilgi:

Sap sisteminde merkezi olarak veri türleri tanımlamasına, veri modelleri ve veri tabanı tablolarının tanımlarının yapılmasına yarayan bir araçtır.
ABAP veri sözlüğü kısaca ABAP DDIC olarak adlandırılır.

ABAP DDIC'te veri türleri, veri tabanı tablosu tanımları , birleşik tablo tanımları (views) , kilit nesneleri (locking objects) bulunabilir.

ABAP DDIC'te veri türlerinin tanımlanması:

- Veri birimleri (data elements) : karakter numerik tanımlanan veri türleri
- Yapısal veri Türleri (Structures) :
- İçcsel tablo vceri türleri : (Table Types) . Tablo biçimindeki veri türleri .

SE11 işlem kodu kullanarak ABAP DDIC açılır.

Kod kısmı ile ilgili bazı notlar:

- DATA: gv_number1 TYPE i,
  gv_date1 TYPE d.
  gv_date1 = '20210909'. => YYYYMMDD formatıyla yazıyoruz ama sonuç DDMMYYYY şekilnde yazılıyor.
  Global value olduğunda adlandırmayı "gv" ile başlatılır.
  Eğer aynı komutla birdan fazla işlem yapmak istiyorsak, komuttan sonra ":" kullanılır buradaki DATA: komutu gibi.
- CONSTANTS gv_kdv_orani TYPE p LENTGH 5 DECIMALS 2 VALUES '46478.08'.
  Constant değişkene tanımlamakla birlikte değer atanır bu şekilde.
  Length 5 ve decimals 2 olduğu için en fazla 7 rakamlı bir sayı olur.
- Form alt_yordan
  DATA lv_number1 TYPE i.
  Endform
  Ve bunu PERFORM alt_yordan ile çağırılır.
  Local değişken tanımlama örneği.
- Eğer Sales and Distribution modülü için bir şey yazıyorsam zsd_ ile başlar.
  Eğer rapor ise: zsd_r_
  Eğer fonksiyon ise zsd_f_
  Eğer class ise zsd_cl_
- PARAMERTERS name(20) TYPE c OBLIGATORY.
  Programa parametreyi bu şekilde verilir.
- gv_char = 'abc'.
  Move 'abc' to gv_char.
  Atamayı iki şekilde yapılabilir.
- Debug işlemi ya (BREAK-POINT) komutu ile yapılır ya da kodun sol tarafına bir kırmızı nokta koyarak yapılır.
- Posta kodu veya 0 ile başlayan numaralar için n(numeric) tipi kullanılır.
- DATA v_name TYPE C LENGTH 10. Veya DATA v_name (10) TYPE C.
  İkisi doğrudur.
- Char c defualt uzunluğu 1'dir.
- Packeged p defualt uzunluğu 8'dir.
- gv_tarih = '20210911'.
  gv_gun = gv_tarih +6(2). => gv_gun = 11.

Burada “gv_tarih”ten 6 karekterden sonra 2 tane karekter alacağımı söylüyorum.

- CONCATENATE gv_gun ‘/’ gv_ay ‘/’ gv_yıl INTO gv_tarih.
  Verilenleri birleştirip gv_tarih’e atıyor.
  Gv_tarih = gv_gun && ‘/’ && gv_ay && ‘/’ && gv_yıl.
  Böyle de birleştirme işlemi yapılabilir.
- PARAMETERS en fazla 8 karekter olabiliyor.

Dersimizin Kaynakları:

1- Cemil Aksel hocanın videoları ve ders anlatımı
2- Aytül Korkut hocanın kitabı
3- Şükrü Bakıroğlu hocanın kitabı
4- SAP kendi sertefika programlarında kullandığı kitaplar.

Betül ALBAYRAK tarafınfan hazırlandı, SAP Eğitiminin 1. ve 2. Haftaların Özetidir.