SQL Nedir?
SQL,Structured Query Language ifadesinin kısaltmasıdır.IBM tarafından 1970 yılında database sistemlerinin yönetimi için oluşturuldu.Bir programlama dili **değildir.** Database'mizdeki nesneleri ve verilerinizi yönetmek için kullanırız.SQL ifadeleri alt kümelere ayrılarak gruplandırılmıştır.

- **DDL İşlemleri:**Data Definition Language ifadesinin kısaltmasıdır.Verinin tanımlanmasına ait işlemleri yapar.Bu işlemlerde data manipülasyonu değil dataların tutulduğu DB,Şema,Tablo,View,Partition gibi DB elemanlarının tanımlanma işlemlerini yapılır.Bu gruba giren SQL ifadeleri *CREATE,ALTER,DROP,TRUNCATE,COMMENT,RENAME*'dir.
- **DML İşlemleri:**Data Manipulation Language ifadesinin kısaltmasıdır.Verinin manipülasyonu ile ilgili işlemleri yapar.Bu gruba giren SQL ifadeleri S*ELECT,INSERT,UPDATE,DELETE*'dir.
- **DCL İşlemleri:**Data Control Language ifadesinin kısaltmasıdır.Bu komutlarla herhangi bir DB nesnesine yetki tanımlaması yapılıp geri alınabilir.Bir tabloya SELECT yetkiniz olabilirken INSERT yetkiniz olmayabilir bu işlemler ilgili komutlarla yönetilmektedir.Bu gruba giren ifadeler *GRANT,REVOKE*'dir.
- **TCL İşlemleri:**Transaction Control Language ifadesinin kısaltmasıdır.Kullandığınız DB'de autocommit özelliği açık değilse yaptığınız DML işlemleri bir nevi askıda durur ve sizin son bir onay vermenizi ya da hata varsa geri almanızı bekler.Bu komutlarla onay verip veya geri alabilirsiniz işlemlerinizi.Bu gruba giren ifadeler *COMMIT,ROLLBACK*'dir.

Bu bilgiler ışığında SQL'in görevleri şunlardır:

- Nesne oluşturma,değiştirme,silme
- Veri sorgulama
- Tabloya satır ekleme,güncelleme,silme
- Yetki tanımlamalarını yapma
- Veri tutarlılığını sağlama

Peki bu SQL nerede çalışır bu çalıştığı yeri tanıyalım.

Database Nedir?

Verilerinizi saklamak ve rahatça yönetmek için kullanılan bir yazılımdır.Bir çok farklı database modeli vardır fakat günümüzde en çok kullanılan model olan RDBMS'tir.(Relational Database Management System).Genellikle bu yazılımlar her zaman çalışır vaziyette olan SERVER olarak nitelendirdiğimiz güçlü bilgisayarlar üzerinde kurulu olurlar.Bir SERVER'da birden çok Database'i kurulu olabilir.

Database'lere örnek olarak:Oracle,Postgresql,SQL Server,Mysql vb verilebilir.

Sık kullanılan Database nesnelerini tanıyalım.

Tablo Nedir?
Database'de verileri tutan nesnelerdir.

Databaseler ilgili verileri yarattığınız tabloların satır ve sütunlarında saklar.

*Satır(Row Or Record):*Yatay olarak gördüğümüz kendine özgü veri kümesi bir alt veya üst satırla hiç bir ilişkisi yoktur kendine ait verileri tutar.

*Sütunlar(Column Or Field):*Dikey olarak gördüğümüz satıra ait bilgileri özelleştirilmiş alanlarda saklamamıza yardımcı olur.

External Table Nedir?
Bir Flat File'ı DB'inizde tablo gibi işleminize olanak sağlar.İki tip driver yardımıyla Flat File'lar tablolara dönüştürülür bunlar ORACLE_LOADER ve ORACLE_DATAPUMP'dır. Tabloyu yaratırken kullandığımız file'lar CSV,TXT,XLSX vb ise ORACLE_LOADER,binary formatlı ise ORACLE_DATAPUMP seçilmelidir.
Oluştururken tanımlamalarını yaptığımız parametrelerden bahsedelim.

TYPE: Oracle_Loader veya Oracle_DataPump
DEAULT DIRECTORY: Bir path değil bir directory'dir.İnput ve Output dosyalarının nerede tutulacağına karar verir.
ACCESS PARAMETERS: Verilere nasıl ulaşacağımızı tanımlamızı sağlar.
LOCATION: Hangi file olduğunu belirttiğimiz kısım.
REJECT LIMIT: Bunu 0 tutarsanız verdiğiniz ACCESS PARAMETERS'lara veya bir üstte tabloyu CREATE ederken belirttiğiniz sütunların formatlarına uymayan bir kayıt geldiğinde tablonuz hata döndürecektir.UNLIMITED derseniz dosya daki tüm veriler hata alsada tabloyu kullandığınız SQL ifadeleri çalışmaya devam eder.

Daha detay bilgiler için :
https://docs.oracle.com/cd/B19306_01/server.102/b14215/et_concepts.htm

View Nedir?
Bir tablo gibi çağırabildiğiniz fakat fiziksel olarak veri tutmayan nesnelerdir.Kullanım şekli olarak bir tablodan farkı yoktur.SQL scriptinizin döndürdüğü verileri bir çok yerde kullanacaksanız ideal bir çözümdür.View kullanılırken dikkat edilmesi gereken en önemli konu View'i yaratmak için kullandığınız SQL scriptinizdeki tabloların verisi değiştiğinde View'inizde döndürdüğü veriler otomatik olarak değişecektir.
Bir view kullandığınızda bu View'e her eriştiğinizde içerisindeki SELECT ifadesini her seferinde çalıştırıyorsunuz demektir ve bu sorgunuz kötü performans gösteren bir sorguysa veya çok büyük veri seti döndürüyor ise kesinlikle kullanmamanız gereken bir yöntemdir.Bu amaçla Materialized View'ler yaratılmıştır veya bir tabloya ilgili data setini basmanız performans açısından çok daha kullanışlı olacaktır.

Materalized View Nedir?

View ile benzer bir kullanımı vardır.İçerisinde SELECT ifadesini tutar ve bunun döndürdüğü data setini kullanıcıya gösterir.Fakat bu SELECT ifadesini fiziksel olarak bir tabloda tutar ve sizin belirlediğiniz bir çalışma aralığına göre datayı bu tabloda günceller.Dolayısıyla View'e her sorgu attığınızda çalışmaz bu da ciddi performans kazancı sağlar.Dezavantaj olarak ise en güncel data değil Materalized View'in en son çalıştığı dataya ulaşmış olursunuz.DWH'da kullanılan bir yapıdır.

Index Nedir?
İndexler tablodaki verilere hızlı bir şekilde ulaşmanıza olanak sağlar.Tablolar veya viewler için yaratılabilir.İki tip index vardır.İndexleme işlemi yaptığımızda bu sütunlara ait veriler için bir işaretleyici oluştururuz ve bu işaretleyici sayesinde verilere hızlıca ulaşırız.

- **Clustered İndex:**Sıralanmış indexdir.Bir tabloda Primary Key olarak tanımlanan genelde Id olarak ifade edilen sütun için otomatik Clustered Index oluşur.Bir tabloda bir tane PK olanı olacağında sadece bir tane de Clustered Index olabilir.İlgili sütun Auto Increment olacağından tüm verilerin sırasını biliyor ve ilgili veriyi bulup hızlıca getiriyor olarak düşünebiliriz.
- **Non-Clustered Index:**Sıralanmamış Indexdir.Birden çok olabilir.İşlem sonucunda verinin nerede olduğuna karar verdiği dizinleri disk üzerinde ilgili tablo dışında bir yerde tutar.Bu da disk maliyetlerimizi arttıracaktır.Non-Clustered Indexler Clustered indexlere göre daha yavaş çalışırlar.

Select hızımızı arttırırken Indexli tablolarda Update,Delete ve Insert işlemlerinin daha yavaş süreceğini unutmamalıyız.

Hemen hemen her zaman aynı sütunları kullanıyorsak WHERE ve JOIN ifadelerinde bu sütunları indexlemek mantıklı olabilir.Çok fazla farklı sütun kullanılıyorsa her biri için indexlemek aksine daha da kötü performans sergiletebilir.Bu yüzden bir veya birden çok sütunu indexleyeceksek analizini iyi yapmalıyız.Ayrıca çok az tekil veri barındıran sütunlara bu işlemi yapmak mantıksızdır.Elimizde bir sütun olduğunu düşünelim sadece 0 ve 1 tutan buna index atılmamalıdır index atılacak sütun veya sütunların tekil hali tablonuzun satır sayısının en az %30'una yakın olmalıdır.

DML islemleri sonucunda Indexler cok hızlı bir şekilde eskiyebilir bu yüzden sık olarak rebuild edilmelilerdir.

Bu seviyede bilmemize kesinlikle gerek yok bence fakat daha da detay okumak isteyenler aşağıdaki linki inceleyebilir.

https://stackoverflow.com/questions/1108/how-does-database-indexing-work

B-Tree,Leaf,Nodes Nedir?
Mantıksal olarak bilmemiz de yarar olan bir terim.Rutin işlerimizde neredeyse karşımıza çıkmaz.

İndexlediğimiz verilerin tutulma biçimi ile ilgilidir.Nodes ve Leaflerden oluşan ağacı andıran bir yapıdır.En üst node(root)'un belirlendiği değerden küçük değerler için Sol node'a

büyükler için Sağ node gider.Ve bu işlem ilgili keyin ait olduğu node'a kadar devam eder.Dolayısıyla tüm bir tablonun blocklarını okumaktansa sadece ilgili node'a ait blockları okur.Bu da ciddi bir performans artışı sağlar SELECT için.UPDATE,DELETE ve INSERT için yavaş olma sebebide budur çünkü sadece tabloya bu işlemler yapılmaz aynı zamanda Index içinde bu işlemlerin yapılması gerekir.

Kendine ait bir child node'u olmayan node'lara Leaf denir.

Daha fazla detay için aşağıdaki kaynakları ziyaret edebilirsiniz
https://www.andrew.cmu.edu/course/15-121/lectures/Trees/trees.html
https://www.javatpoint.com/b-tree

Synonym Nedir?
Synonym herhangi bir DB'deki başka bir nesneye ulaşmak için yaratılan nesnelerdir.Bu bir tablo olabileceği gibi view,function,prosedürde da olabilir.Bir tablo için oluşturulan Synonymde SELECT,UPDATE,DELETE,INSERT işlemleri yapılabilir referansındaki tablo da bu değişiklikleri gerçekleştirecektir.Viewler gibi fiziksel olarak diskde yer kaplamazlar.

Neden kullanırız? N tane DB'de aynı tabloyu kullanmak istediğinizi düşünelim.Bu durumda bu tabloda bir değişiklik yapmak istersek N tane DB'de gidip bu değişikliği yapmamız gerekmektedir.Fakat bir DB'de tablo olarak yaratıp diğer N-1 DB için Synonym yaratırsak sadece 1 DB değişikliği yaparız ve tüm DB'ler güncellenmiş olur.Bu hem maliyet kazancı hem data tutarlılığını garanti eder.

Ayrıca var olan bir tablonuzun adını değiştirmek bu tabloyu kullanan tüm uygulamalarda değişiklik yapmanızı gerektirmektedir.Bunu yapmak yerine Synonym kullanabilirsiniz.

Sequence Nedir?
Bir tablo için PK gibi çalışan unique ve auto increment birden çok sequence oluşturulabilir.Her kullanıcının işlemi için bir değer atanır Commit edilmemişse dahi ilgili kayıtlar için değerler atanmıştır Rollback yapıldığında dahi bu işlem geri dönmez.Bu yüzden hiç bir işlem için duplice kayıt oluşmadığı garanti edilmiş olur.Ascending veya Descending olarak increment olacak şekilde yaratılabilir aynı zaman da increment miktarı ve max,min değerler atanabilir.
Tablo'dan bağımsız bir nesnedir. DWH'da çok kullanılan bir method değildir.

Detay isteyenler için:
https://www.sqlshack.com/difference-between-identity-sequence-in-sql-server/

Partition Nedir?
Partition bir tabloyu istediğimiz sütuna göre tablolara böler.Bir tablo gibi davrandığından sorgularınızda bir tablonun herhangi bir partitionını kullanabilirsiniz.Veriyi yönetmeyi kolaylaştırdığı gibi performansı ciddi bir şekilde attırır.Bir çok Partition yapısı vardır en çok kullanılan methodların açıklamalarını aşağıda paylaşacağım..Tablonuzda yarattığınız partitionların gruplandırma kuralı dışında bir veri insert etmeye çalışırsanız hata alırsınız.Bu yüzden önce ilgili verinin girebileceği bir partition yaratıp sonra insert veya update etmeniz gerekmektedir.2 tip kullanımı vardır 1.cisi direk tabloyu çağırırsınız ve WHERE veya JOIN

lerde partitionlamanızı sağlayan sütunu kısıtlarsınız ve bunun sonucunda ilgili partition(lar)a gider veya 2.olarak direk ilgili partitionları FROM veya JOIN de refere ederiz.

- **Range:** Tablonuzdaki verileri ve yeni gelecek verileri belli aralıklara göre bölmenizi sağlayan yapıdır.Bu aralık bir tarih,sayı veya metin olabilir.DWH da özellikle Fact tablo dediğimiz tablolarda çok sık tarihe göre ayrılmış partitionlar görebilirsiniz.
- **List:** Tablonuzdaki verileri belirli değerlere göre bölebilirsiniz.Örnek vermek gerekirse bir sütununuz var ve A'dan Z'ye kadar değerler alabilir ve bu değerlere sahip binlerce kaydınız varsa A partitionına sadece A değerine sahip kayıtlar gelecek B partitiona sadece B değerine sahip kayıtlar girecektir.Bir List partitiona ait birden çok değerde tanımlayabilirsiniz.Örneğin Partition 1'a A,B,C değerleri girecektir diyebilirsiniz.
- **Hash:**Range ve List tiplerinde net bir kural çizebiliyorduk Hash genellikle herhangi net bir kırılım olmayan durumlarda tablonuzu istediğiniz x parçaya bölmenize sağlar.

Daha fazla detay için aşağıdaki linki ziyaret edebilirsiniz.

https://docs.oracle.com/cd/E18283_01/server.112/e16541/part_admin001.htm

Constraint Nedir?
Bir constraint tablonuzdaki veri tutarlığını sağlamak için kullanılan kısıtlayıcılardır.Bir tabloda NULL değer içermemesi gereken bir sütununuz var ise constraint koyarak bunu garantileyebilirsiniz veya 3 sütun üzeirnden uniqueliği sağladığınız bir durumda bu 3 sütunu içeren bir unique constraint oluşturabilirsiniz.

En çok kullanılan constraintler aşağıdaki gibidir:

- **NOT NULL:**Sütuna NULL veri gelmesini kısıtlar.
- **UNIQUE:**Bir veya birden çok sütun için duplice veri gelmesini engeller.
- **PRIMARY KEY:**NOT NULL ve UNIQUE constraintlerinin birleşimidir.
- **FOREIGN KEY:**Bir sütuna insert veya update atacaksanız refere ettiği tabloda bu değerin olmasını zorunlu kılar.
- **CHECK:**Herhangi spesifik bir koşula göre kısıt uygular.5'ten büyük veya 100 ile 200 arasında gibi.

Detaylar için:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/clauses002.htm

Tablespace Nedir?
Fiziksel olarak yer tutan index,tablo,materilazed view'ler belirtmiş olduğumuz tablespacelerde tutulur.Tablespace'lar datafile'ların genel adını oluşturan mantıksal bir kavramdır.Fiziksel olarak datayı tutmazlar datayı tutan aslında datafile'lardır.Her bir tablespace birden çok sayıda datafile'da tutulabilir.Burda en çok karşımıza çıkacak sorun tablespace'in dolu olmasıdır.Bu aslında ilgili datafile'larda yer kalmadı demektir bu durumda ya var olan datafile'ların boyutu arttırılır ya da yeni datafile'lar eklenir.Bir diğer karşımıza çıkabileceği konu yeni bir tablespace yaratılması konusudur.Burda da yine ilgili datafile'lar yaratılır bu file'ları tutacak mantıksal kavrama isim verilir.Bundan sonra yaratacağınız bir

tabloda iligli tablespace'i seçerseniz artık bu yeni file'larda verileriniz tutulacaktır.Tablespace'lerinizde yeteri kadar yer olmazsa DML işlemlerinizde hata alırsınız önemli bir konudur

Genelde DBA'lerin işidir fakat bazen ETL süreçlerimizde bu hatayı alabiliriz DBA'lerin gözünden kaçmış olabilir bu tip durumlarda doğru tanı koyabilmek için bilinmesinde fayda vardır.

Sistem tablolarından veya bazı DB connection için kullanılan IDE'lerde bunları rahatça analiz edebilirsiniz.

Daha fazla bilgi için :

https://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm

Function Nedir?
Bir fonksiyon aslında belli bir amacı gerçekleştirmek için kullanılan bir SQL komutudur.Her seferinde kod bloğunu tekrar yazmaktansa belli bir isimle DB'de saklayıp çağırılmasına yarar. Herhangi bir sayıda girdiği değerine sahip olabilir sonucunda fonksiyonun amacına göre tek bir değer veya bir tablo döndürebilir.İki tür fonksyion vardır:

- **System Defined Functions:**Bu fonksiyonlar DB kurulumda veya sonradan eklenen modüllere erişilen DB'ye özgür fonksiyonlardır.Hemen hemen her DB'de farklı isimler ile aynı işi yapan fonksiyonlara rastlayabilirsiniz.(Oracle için: https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions001.htm)
- **User Defined Functions:**Kullanıcılar tarafından oluşturulan fonksiyonlardır.System fonksiyonlarının talebinize yetersiz kaldığı durumlarda yazarız.

PL/SQL Nedir?
SQL'in genişletilmiş halidir diyebiliriz.SQL'in yetersiz kaldığı durumlarda PL/SQL'e geçiş yapmamız gerekebilir.Bir kod bloğunuz var ve bunun tablonuzdaki farklı değerler için çalışmasını istiyorsanız döngülerle bu kod bloğunuzu o değerler için defalarca çalıştırabiliriz.Veya herhangi bir koşul gerçekleşirse bir kod bloğu gerçekleşmezse başka bir kod bloğu çalıştırmak istersek PL/SQL'e başvurabiliriz.SQL PL/SQL çağıramazken PL/SQL tüm SQL komutlarını çağırabilir.
Bir PL/SQL kodu genellikle şu blokları içerir.

- **DECLARE(Zorunlu Değil):**Bu kısım kullanacağımız değişkenleri ve veri tiplerini belirttiğimiz kısımdır.Cursor kullanıyorsak memory'de cursor için yer ayırttığımız kısımdır
- **BEGIN:**Beginden END bloğuna kadar istediğiniz kod bloğunu buraya yazarız.
- **EXCEPTION(Zorunlu Değil):**Hataları yakalayıp ilgili hataları alırsa ne yapmak istediğinize karar verdiğiniz kısımdır.
- **END**

Öğrenmek isteyenler için kaynak1:https://www.oracletutorial.com/plsql-tutorial/
kaynak2:https://www.oracle.com/tr/database/technologies/appdev/plsql.html

Stored Procedure(SP) Nedir?

Stored procedure'ler functionlar gibi belli parametreler alır bir kod bloğu çalıştırır ve bir değer döner veya dönmez.SP'ler performanslıdır bir kez çalıştılar mı memoryde planları oluşur ve bu planları tekrar kullanırlar.
Peki aynı kod bloğunu hem stored procedure hem functionla çalıştırabiliyorsak ne zaman function ne zaman stored procedure kullanmalıyız.DWH'da da sıkça kullanılır

- Functionlarda bir RETURN zorunludur procedure'da bu zorunluluk yoktur.
- Çalıştırma mantıkları bambaşkadır.SP'ler PL/SQL ile çağrılır yani BEGIN exec stored_procedure; END; şeklinde iken functionlar direk SELECT schema.function() FROM table şeklinde SQL komutu içinde kullanılır.
- Bir SP'yi function içinde kullanamazsınız fakat bir function'ı SP içinde kullabilirsiniz.

Package Nedir?
Bir package değişkenleri,cursorleri ve alt program bloklarını tutmak için yaratılan bir objedir.DB'de saklanır ve birden çok kullanıcı ve uygulama tarafından kullanılabilir.
spec ve body'den oluşur.spec değişkenleri tipleri cursorleri belirttiğiniz global tanımlamalar yapılan kısımdır.Body ise kod bloklarınızı yazdığınız kısımdır.
Genellikle aynı amaca hizmet eden kod bloklarını bir arada tutmak için yaratılır.Bir cursor tanımlarsanız bu tüm kod bloklarında kullanabilirsiniz ve tek seferlik memory ayırırsınız.
Bu kod bloklarını SP olarak düşünebilirsiniz.Ayrıca paketin bir alt programını çağırdığınız tüm paket için plan oluşur ve diğer alt programlar için bir daha memoryde yer açmaz.DWH'da da sıkça kullanılır.

Daha fazla bilgi için
: https://docs.oracle.com/cd/B19306_01/appdev.102/b14261/packages.htm

Trigger Nedir?

Aslında bir SP'dir.DB'de verdiğiniz koşullar sağlandığında otomatik çalışır.3 farklı durum için de trigger yaratılabilir.

- **DML Triggers:**Bir tabloya INSERT,UPDATE,DELETE işlemi yapıldığında çalışmasını istediğiniz triggerlar.
- **DDL Triggers:** CREATE,DROP,ALTER gibi işlemler yapıldığında çalışmasını sitediğiniz triggerlar.
- **DB Operation Triggers:**LOGON,LOGOFF,SERVERERROR gibi DB operasyonlarında çalışmasını istediğimiz triggerlar.

Bir trigger için tetikleyici işlemden önce veya sonra için çağrılabilir yeni ve eski değerleri çağırabilir.Bir tablodaki belli bir kısım sütuna update geldiğinde eski değerleri kaybetmek istemediğiniz bir durumda bu veriler için bir trigger yaratıp başka bir tabloda tutabiliriz.
Daha fazla bilgi için:
https://docs.oracle.com/database/121/TDDDG/tdddg_triggers.htm#TDDDG52000

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver15

DBLink Nedir?
Bir databeseden başka bir databese erişim sağlayan bir DB nesnesidir.Bu nesneyi yarattıktan sonra bunun aracılığıyla diğer DB'nin tablo ve view'lerine erişebilirsiniz.İlgili DB'de yaratma yetkileri bağlanmak istediğiniz remote DB'de session yetkinizin olması gerekmektedir.
Daha fazla bilgi için :
https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5005.htm

FK,PK Nedir?

- **Primary Key:**Bir veya birden çok sütunlardan oluşan bir satırı diğerlerinden ayırmaya yarayan tekilliği garanti edilmiş sütunlardır.Bir tabloda sadece bir PK alanı bulunabilir.
- **Foreign Key:**Bir veya birden çok sütundan oluşan ve başka bir tablonun primary keyi olan sütunlardır.Eklediğiniz bir satırdaki foreign keyin değeri primary keyi olan tablodada olmalıdır.Bunun sayesinde veri bütünlüğü sağalanıp tablolar arasında ilişki kurulabilir.Bir tabloda birden çok FK alanı olabilir.

Meta Data Nedir?
Meta Data data hakkındaki data şeklinde ifade edilmektedir.Bir DB'de user-defined fonksiyonlar,şemalar,tablo,view adları,sütunları,tipleri,kuralları,boyutu,ilişkileri gibi bilgileri içerir.Bunlara sistem tabloları içerisinden ulaşabiliriz.

System Table Nedir?
Çok fazla işimizin düştüğü ve işimizi kolaylaştıran DB'nin kendi tablolardır.Bir tablonun sütunlarına ihtiyacınız varsa veya hangi tabloda ne tip indexleriniz var öğrenmek istiyorsanız bu tablolara başvururuz.Bu örnekler çoğaltılabilir fakat bunlar hakkında şunu bilmemiz yeterli olacaktır DB'deki yaratılan herşey bir sistem tablosunda tutulmaktadır.Çok fazla sistem tablosu var fakat en çok ihtiyaç duyduğumuz bir kaçını yazıp detay için link bırakacağım.

- **ALL_TABLES:**DB'deki tüm tablolara ait bilgileri döndürür.Size'ı,Adı,Şeması,Tablespace'i,Satır sayısı vb..
- **ALL_TAB_COLUMNS:**Bir tabloya ait tüm sütunları veri tiplerini,uzunluklarını,Nullable durumunu vb sütunu tanımlayan tüm bilgileri içerir.
- **ALL_TAB_PARTITIONS:**Hangi tablonun kaç parititonu var isimleri,max min değerleri vb bilgileir içerir.

https://docs.oracle.com/database/timesten-18.1/TTSYS/systemtables.htm#TTSYS720
https://www.techonthenet.com/oracle/sys_tables/index.php

Explain Plan Nedir?
Bir sql komutu çalıştırdığınızda DB'niz optimizer'ı sorgunuz için bir çok plan içinden en uygun planı bulmaya çalışır ve genelde bu planı yürütür.

Bir explain plan hangi tabloya önce gidecek,hangi tabloya nasıl erişecek,join varsa nasıl bir join seçeceği,filtrelemeler aggregationlar vb barındırır.
Bu planlar aracılığıyla nerede ne kadar süre kaybettiğinizi bulup komutlarınızı güncelleyebilrisiniz.Başka bir yazıda nasıl iyi SQL yazılırı konuşacağız fakat burda da belirtmekte fayda var fazla adımlardan her zaman kaçının olabildiğince ufak data setlerini çağırmaya çalışın.
Çünkü sorgunuz geç çalışmasının başlıca sebeplerinden biri FULL SCAN etmenizdir tabloları işte Explain planla bunları tespit edebiliriz.

Explain Plan okumak ve adımları başka bir yazının konusu olacak genel tanım için yeterli sanırım bu açıklama.

Daha fazlası için :https://docs.oracle.com/cd/B10501_01/server.920/a96533/ex_plan.htm
https://docs.oracle.com/cd/B10501_01/em.920/a86647/toc.htm

OLTP Nedir?
Online Transaction Processing'in kısaltmasıdır.DWH'ımız için "Source" olan DB'lerdir.Birincil önceliği dataların güvenle eklenip,güncellenmesi ve son kullanıcıya bilgilerinin gösterilmesidir.Bir e-ticaret sitesinden alışveriş yaptığınızı düşünün burdaki sepetiniz,ödemeleriniz,siparişleriniz,üye bilgileriniz her biri bir tabloda tutulmakta bir biriyle konuşmaktadır.Her verdiğiniz sipariş için bir insert atılır tabloya veya güncellemelerinizde update atılır.Bu DB'lerdeki Select'lerde genelde çok ufak bir kitle döndürürler.Bilgilerim sekmesine girdiğinizde gördüğünüz tüm bilgiler o anda DB'den dönen select ifadesinin sonuçlarıdır.Mimari olarak normalize yapıda olurlar bu en önemli farkıdır OLAP'tan(Kimball model için).

Normalizasyon Nedir(3NF)?
Data tekrarını azaltmak,insert,update ve deletelerde ortaya çıkabilecek tutarsızlıklardan kaçınmak ve büyük tabloları daha ufak tablolara bölmek için geliştirilen databese dizayn tekniğidir.Bir çok formu vardır fakat genelde 3rd Normal Form'a kadar geliştirilir.
Her bir adım bir önceki adımların üzerine yeni kurallar gelmesi ile devam eder.Dolayısıyla 1NF olmadan 2NF olmaz.Kimball method uyguluyorsanız çok işiniz düşmez.

Daha fazla bilgi için :https://www.guru99.com/database-normalization.html

OLAP Nedir?
Online Analytical Processing'in kısaltmasıdır.DWH bir OLAP sistemdir.Birincil amacı son kullanıcıya bilgileri göstermesidir fakat bunu yaparken çok büyük data setlerini kullanabilir.Burdaki insert ve update(tavsiye edilmez) OLTP gibi transaction transaction değil bulk'tır.Data güncelliği besleyen ETL'in sıklığı kadardır.Burdaki datalar denormalize olarak tutulurlar OLTP'den en büyük farkı budur.(Kimball model için)

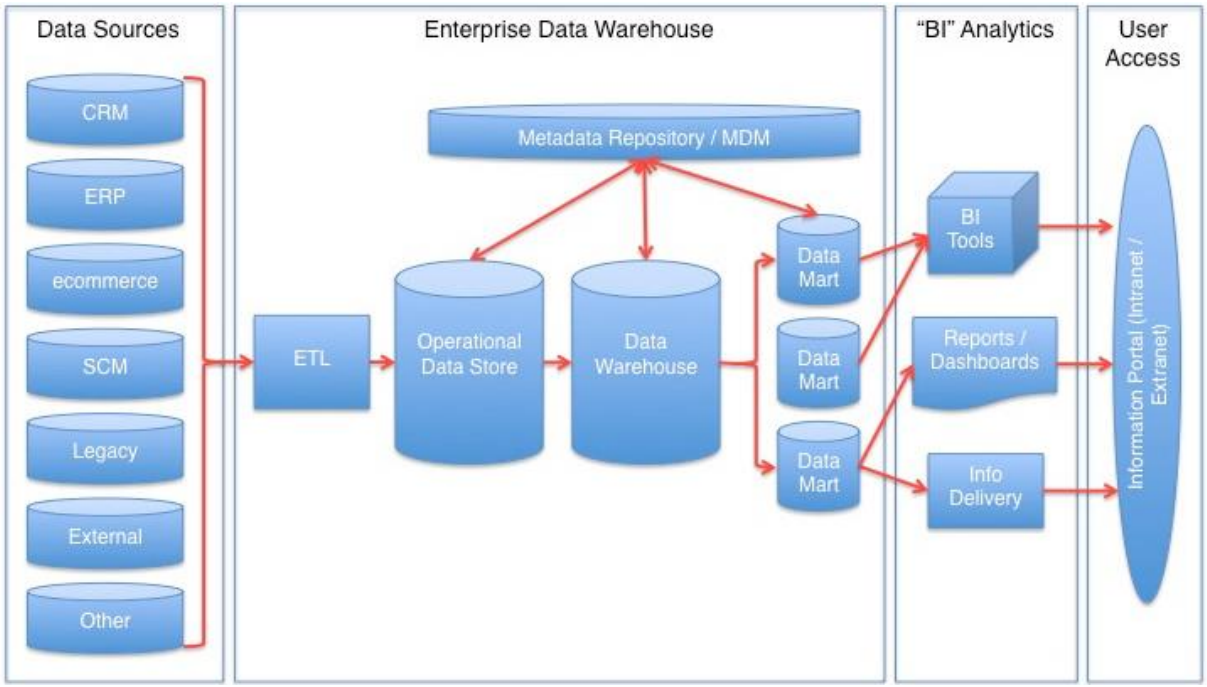
DWH Nedir?
Birden çok kaynakdan gelen dataların toplanması için kullanılan DB'dir.Genelde BI araçlarımız bu DB üzerine kuruludur ve bu şekilde son kullanıcıya rapor ve analiz imkanı tanırız.Karar destek sistemleri olarakda adlandırıldığı olur(DSS).DWH'lar genellikle t-1 olarak ifade edilen bir gün öncesine kadar güncellerdir.Bazı tablolar veya Data Martlar(DM'ler) real time veya near real time olabilir.
DWH'a neden ihtiyaç vardır:

1. Bir çok farklı kaynağı tek bir yerde toplayabilmek
2. Geçmişi korumak(SCD veya snapshotlar)
3. Büyük veri setini çağıran sorgular için farklı mimari ihtiyacı
4. Tek bir DB kaynağınız olsa bile OLTP sistemiminizi yormamak

3 tip DWH vardır:

1. Enterprise Data Warehouse(EDW):Tüm bilgilerin yer aldığı tek bir DWH'dır.
2. Operational Data Store(ODS):Yapısı OLTP'ye daha yakındır fakat OLTP source'a 1-1 benzemek zorunda değildir veri üzerinde değişiklikler yapılabilir.T-1 veya real time olabilir.Genellikle satır satır analize ihtiyaç olduğu durumlarda kullanılır.
3. Data Mart:DWH'ın sadece bir konuya özgü ufak halidir.

ODS ve DM'ler DWH içinde de olabilir ayrı ayrı DB'lerde de olabilir.



Resim kaynağı:https://commons.wikimedia.org/wiki/File:Datawarehouse_reference_architecture.jpg

Tüm DWH yapıları bu şekildedir demek doğru olmaz genelde Data Sources->ETL->DWH->BI Tools adımları zorunlu diğerleri opsiyonel demek daha doğru olabilir.

ODS Nedir?
Bir çok farklı kaynaktan 1-1 veya biraz değiştirilmiş dataları tutan bir yapıdır.DWH dışı bir DB olabileceği gibi DWH içinde de olabilir.Genellikle satır bazlı raporlamalar için kullanılır.Genellikle daha ufak data setleriyle uğraşır DWH'a kıyasla.Örneğin bir kullancının kartında her zamanın ki 20 katı bir harcamalık kayıt yakalamak istersek ODS işimizi fazlasıyla kolaylaştırır.Ayrıca bir çok farklı kaynağın birleştiği tek bir DB için diğer IT operasyonlarıda yönetilebilir.

Data Mart Nedir?
DWH'ın tek konuya odaklanmış halidir.(Satış,Finans vb).3 Çeşit DM vardır:

1. Bağımlı DM'ler:DWH'da tüm süreç işledikten sonra DM'ye bilgi çıkılır.O yüzden DM,DWH'a bağımlıdır.
2. Bağımsız DM'ler:DWH'dan tamamen ayrı bir süreç dolayısıyla yönetilmesi çok ve ayağa kaldırılması çok kolay.Fakat sonrasında her DM için ETL süreçleri kendi mantıkları daha da zor hale gelebilir.
3. Hibrit:Bazı tabloların DWH'dan bazılarının direk sourcedan geldiği DM'ler.

Bağımlı DM'ler yönetilebilirler olarak en mantıklısı durmaktadır.
DM'ler farklı bir DB'ye çıkabileceği gibi DWH içerisinde DM mantığıyla da tutulabiliyor.

Fact Nedir?
Fact tablolar Kimball methodunun ana tablolarıdır.İçerisinde yalnızca Foreign Key'ler ve Mesaure'lar tutar.Foreign Keyler Dimension table'lara gitmemiz için gereken bağlantı sütunları iken Measure'lar ölçümlenen herşeydir.Bu tablolar Star Schema ve Snowflake'in tam ortasında yer alırlar. Bu tablolarda genelde PK alanı yoktur.Fakat bir PK olacaksa bu tablodaki tüm FK'lardır.
Bu tablolar dikine büyürler ve gerçekten çok büyük veri tutarlar.Bu yüzden bu tabloların partitionlı olması çok önemlidir.Burdaki partition factinizin tipine ve iş sürecinize göre değişiklik gösterebilir.Factlerinizde FK'larınızı doldurken kontrol adımları koyup koşullarınıza uymayan kayıtlar için -1,-2 gibi id'ler basabilir dimension tablolarınıza bu eklediğiniz -1,-2 kayıtları için kayıt atıp son kullancıya bir sorun olduğunu gösterebilirsiniz.
3 tip fact vardır:

1. **Transaction Fact Table:**Bu fact yapıları geçmişe dönük update'ler gelebiliyorsa belirli bir tarihten itibaren sonrasının silinip tekrar yüklendiği olmuyorsa o gün yeni gelen dataların eklenerek büyüyen tablolardır.Bu tablolar için iş biriminin raporlama ihtiyacına kalmakla beraber aylık veya günlük parititonlar oluşturabilirsiniz.Genelde bu factlerde hatalı bir delete işlemi için ETL süreci tekrar başlatılıp veri kaybı yaşamadan süreciniz devam edebilir
2. **Snapshot Fact Table:**Günlük,haftalık,aylık olarak satışçı,müşterinin o anki durumunun bir tabloda saklandığı fact yapılarıdır.Genelde günlük olurlar geriye dönük hesaplaması ciddi zorlukları olan veya bulunamayan measure'lar içerirler.O yüzden bu factlerde genelde oluşturduğunuz tarihten öncesi için data yüklemesi yapamazsınız.Delete işlemleride çok sorun oluşturur backup'ınız yoksa çünkü bulamazsınız satışçı veya müşterinin 2 gün önceki halini.Bu factler için genelde günlük partitionlar uygundur.
3. **Accumulated Fact Sheet**:Bir transaction başlangıçta facte işleniyor bittiğinde bu kayıt update ediliyor.Ben hiç denk gelmedim açıkçası deneyimime ait yorumlarım yok maalesef.

Fact tabloları parititonlı yaptığımız durumlarda ETL sürecimize partitionı yoksa ekle adımı eklediğimizden emin olmalıyız yoksa ETL sürecimiz hata alacaktır.

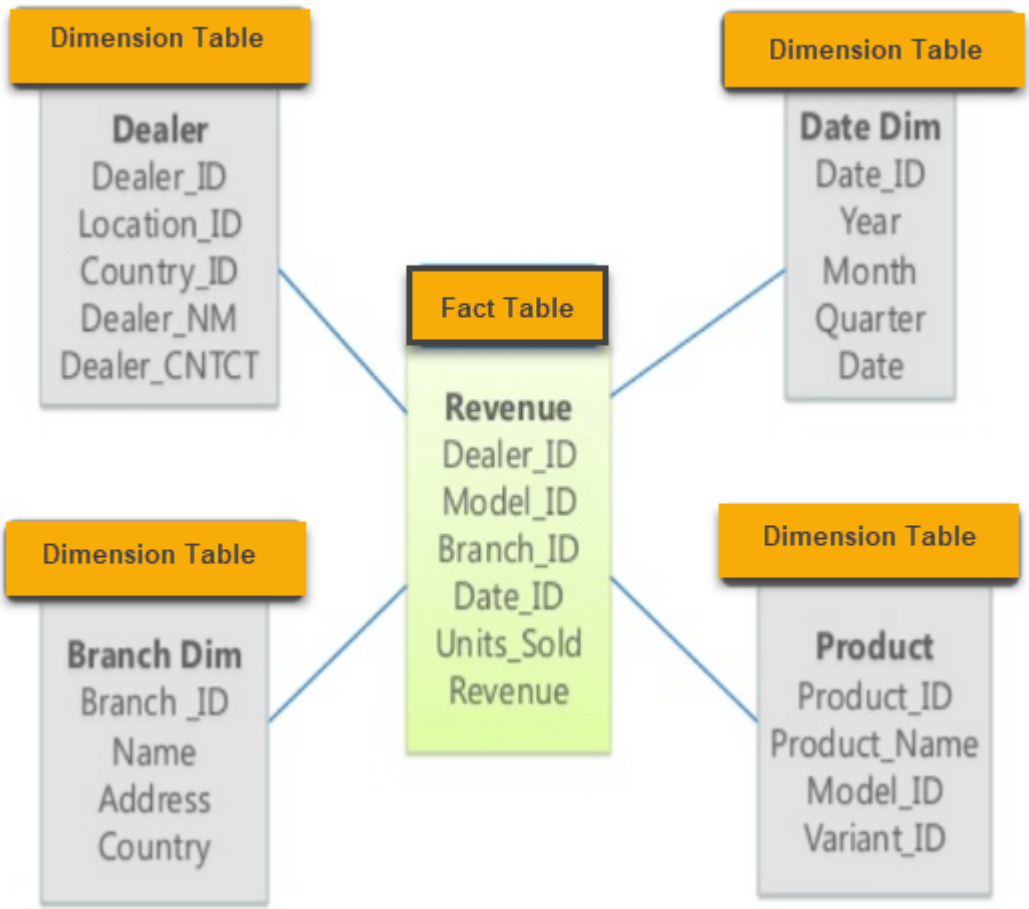
Dimension Table Nedir?
Dimension tabloları açıklama tutan tablolar olarak düşünebiliriz.Fact'imizde bulunan FK'lar bu tabloların PK'larıdır bunlar aracılığıyla join kurup factimizdeki hesaplamaların hangi il,hangi müşteri veya hangi satışçı ait olduğunu bulabiliriz.

Factinizde yer alan CustomerID FK alanı için 1 yazıyorsa bunun hiç bir anlamı yoktur.Fakat Dim_Customer tablonuzda bu 1'in karşısında Hüseyin yazdığında veya bu customerin ili İstanbul yazdığında kimin veya hangi ilin ne kadarlık sipariş ettiğini anlamlandırdınız demektir.

Dimension tablolar PK,FK ve Açıklama sütunları barındırabilir.Her Dimension tablonuza -1,-2 gibi farklı hata durumlarınız için kayıt atmanızda fayda vardır.

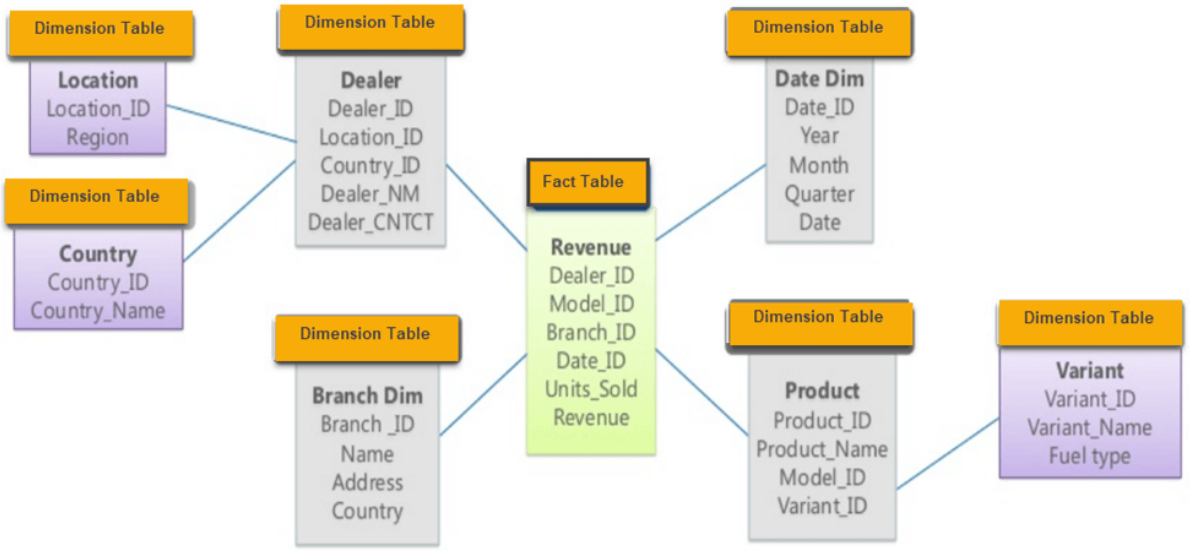
Star Schema Nedir?

Ortada bir factin olduğu ve FK'lar aracılığıyla Dimension table'lara gittiği yapıdır.Genelde tüm fact'ler Dim_Date tablosuyla joinlenir.Fact'inizdeki Raporlama Tarihi(İsim değişebilir)'de bir FK'dır ve Dim_Date'deki Date sütunuyla joinlenir.Bir dimension birden çok factle joinlenebilir ve star schema oluşturabilir burda bir sınır yoktur.En önemli kural Fact Tabloyu başka bir fact tablo ile joinlemeyin.Hatta BI Toolunuzda farklı iki factte measure'ları aynı klasör ile son kullanıcıya sunmayın.1-1 aynı FK'lara sahip olmayan factler datanız çoklatacaktır.İki factli yapıların BI'da Full Join kuracağını unutmayın sorgunuzda var fakat görselinizde yoksa bile 0 değerli dimensionlar gösterirseniz.BI toollarında dimension alanları muhakkak Dimension Table'lardan getirin bunlar FK'lar olsa bile.Factten sadece measure'ları alın.



Resim Kaynak :https://www.guru99.com/star-snowflake-data-warehousing.html

Snowflake Nedir?
Star Schemadan tek farkı Dimension table'lardan başka Dimension table'lara gittiğimiz durumlar olabilmekte mantıksal olarak bir farkı yoktur.
Örneğin Size Ülke bazında gelirler lazım Factinizde Location_Id bulunur.Dim_Location tablosuna gittiğinizde City_Id gibi bir sütun görürsünüz ve burdan Dim_City tablosuna gidersiniz.Star Schema ve Snowflake BI Toollarındaki geliştirmelerde daha net gözükür fakat DWH'ı böyle modelleminiz gerek elbette.Bazı durumlarda Snowflakeden Star Schemaya dönmek isteyebilirsiniz bu kısım önemli :))Snowflake olan bir yapıda önce bir dimension sonra öbürüne sonra öbürüne şeklinde bir tablo yerine birden çok tabloyla ilişki kurarsınız fakat bu ara dimension tablolar gerçekten çok büyük olabilir işte bu durumda performans sorununa yol açabilir.Bunun yerine fact tablonuza nihai dimension tablonuzun FK'sını basıp direk buna bağlanabilir ve süreyi kısaltabilrsiniz.



Resim Kaynak :https://www.guru99.com/star-snowflake-data-warehousing.html

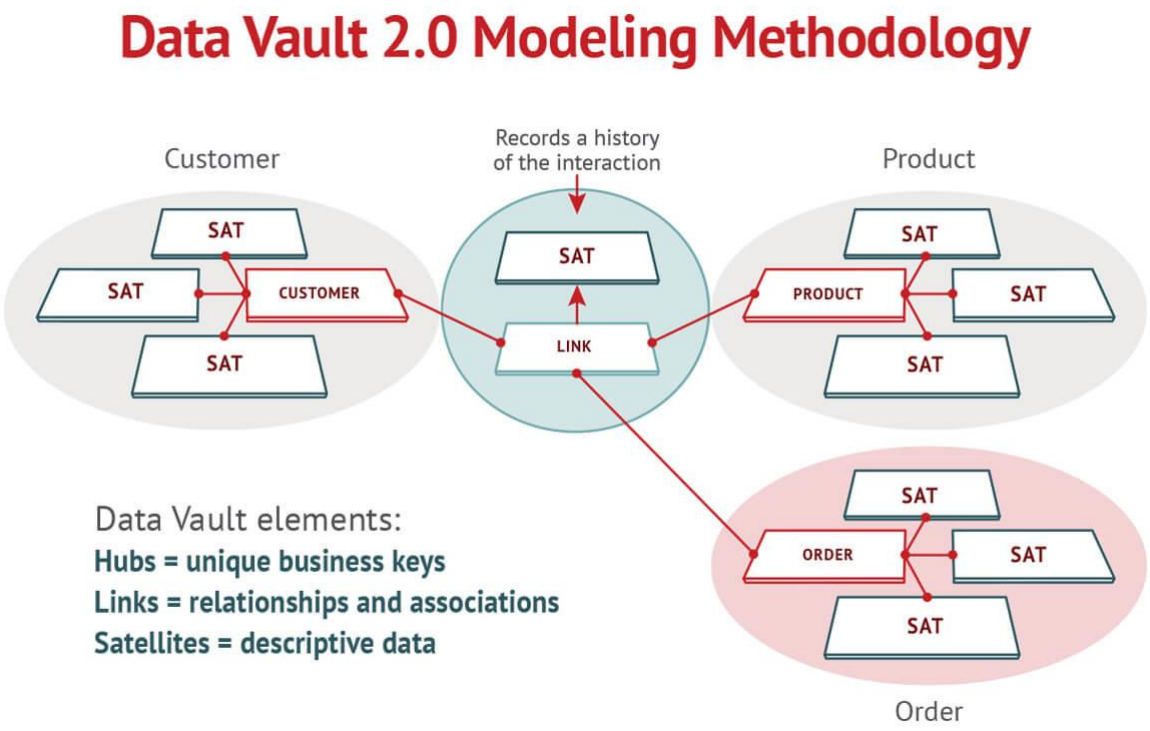
Inmon Method Nedir?(3NF)
Inmon DWH mimarisi için normalize yapıyı önermiştir.Dolayısıyla veri tekrarı çok azdır.Normalizasyonu yukarda belirtmiştim zaten.DWH olarak iyi bir çözüm olmasına rağmen kurması ve geliştirmesi daha fazla zaman almaktadır.Genelde Kimballa method tercih edilmesinin sebebide budur.Fakat İnmon methodunda da daha fazla esneklik vardır raporu çekecek kişi için.Ayrıca ilişkilerin fazlalığından dolayı bir biriyle konuşabilen daha fazla veri setleri elde edebilirsiniz.

Kimball Method Nedir?
İnmonın aksine denormalize bir yapı önermektedir.Fact ve Dimensionların ilişikilerini Star schema veya snowflake olarak isimlendirilmiş methodlarla yapar.
Geliştirilmesi daha hızlı olduğu ve daha az karmaşık olduğu için tercih edilir.Fakat çok fazla değişikliğin olduğu yapılarda çok uygun değildir.

Data Vault Nedir?
İnmon ve Kimball'a methodlarına göre daha yeni bir DWH mimarisi methoddur.Star schema ve 3NF birleştirmeyi öngören bir methottur.
3 farklı tipte tablodan oluşur:

- **Hub:**Hublar ilgili tablonun Id'lerini,bunların yükleme zamanını ve hangi kaynaktan gelen tutan tablolardır.
- **Link:**Bu tablolar birbiriyle konuşacak Hubların ID'lerini ve yükleme zamanlarını ve kaynağını tutarlar.
- **Satellite:**Bu tablolar asıl veriyi tutan tablolardır.Bir Id'ye ait tüm veriler bu tablolarda durur.Bir Satellite tablonun sadece bir tane Hub tablosu olmalıdır.

Satellite tablolar Hublara,Hublar Linkler aracılığıyla birbirine bağlanır.



Resim kaynağı:https://www.wherescape.com/solutions/project-types/data-vault-automation/
Çok fazla kaynak ve örnek yok webde fakat bu linke göz atabilirsiniz:https://kentgraziano.files.wordpress.com/2012/02/introduction-to-data-vault-modeling.pdf

İnmon,Kimball ve Data Vault kıyaslamalı güzel bir araştırma yazsının linkini bırakıyorum avantaj ve dezavantajlarına bakabilirsiniz

https://www.researchgate.net/publication/312486486_Comparative_study_of_data_warehouses_modeling_approaches_Inmon_Kimball_and_Data_Vault/link/5f19972892851cd5fa3f5976/download

ETL Nedir?
Extract-Transform-Load'un kısaltmasıdır.Bir veri kaynağındandan başka bir veri kaynağına data taşıma işleminin adıdır.Genelde bir veya birden çok source DB veya Flat File'dan DWH'a veri aktarımı için kullanılır fakat tam tersi olduğu durumlarda olabilir.Extract adımında

kaynakdan veri alınır,Transform adımında data temizliği,parçalanması veya aggregate işlemi yapılır,Load adımında ise dönüştürülen veri yüklenir.Bu ETL süreci için bir çok farklı firma ürün çıkarmıştır bunlardan en çok kullanılanları şöyledir:SSIS,İnformatica,SSIS,Data Stage,Talend,Data Services vb..Tüm bu toollar aynı şeyi yaparlar isimlendirme farkları vardır sadece birini iyi anladığınızda diğerlerini öğrenmeniz çok kısa sürer ve genelde 1 firma sadece 1 tanesini kullanır hepsini öğrenmek zorunda değilsiniz.

ETL aracı kullanırken olabilidiğince tool'a ait componentleri kullanmaya özen göstermeliyiz bu yönetilebilirliği arttıracaktır.Çoğu zaman yazılan SQL'i basıp geçmek size ciddi zaman kazandırır ama ETL sürecinde bir şey değiştirmek istediğinizde SQL'i tekrar analiz etmeniz gerekir.Ayrıca ilişkileri kopardığınız için hangi tablo veya sütun nerede kullanılmış bilgilerini kaybederseniz ve ciddi bir reengineering sürecine girerseniz işin içinden çıkamayabilirsiniz.

PL/SQL ile de ETL sürecinin bir kısmını veya tamamını yönetenler var fakat tavsiye etmiyorum kodlar bir noktadan sonra çok fazla karışabiliyor.

ELT Nedir?
ELT olarak data aktarım yöntemini seçersek Extract-Load-Transform olarak süreç değişir.Datayı DWH'a yükledikten sonra istemiş olduğunuz transform işlemlerini DWH üzerinde yapar.
Piyasada en çok kullanılan ELT toolları şöyledir:ODI,Talend,Informatica vb.ELT genelde büyük veriler ve cloud sistemler için daha idealdir.Ve data yükleme süresi gerçekten çok kısadır.

ETL ve ELT arasındaki farkları merak edenler için :https://www.guru99.com/etl-vs-elt.html

SCD Nedir?
Slowly Changing Dimensions ifadesinin kısaltmasıdır.Bazen Dimension tablolardaki veriler değişebilir ve bu değişim bizim için anlamlı ise bunu kaybetmek istemeyiz işte bu durumda Dimension tablolarımızı SCD yapılarına dönüştürürüz.Bir çok farklı versiyonu olmakla beraber en çok kullanılanı Type 2 'dir.Örneğin Dim_Customer diye bir dimension tablonuz var ve medeni durumu değişiyor ve bu sizin raporlamalarınızda önemli bir alan bu durumda Type 2 aynı müşteri için yeni bir kayıt yarat fakat bu her iki kaydada başlangıç ve bitiş tarihleri ve hangisinin güncel olduğunu rahatça yakalamak için de bir flag koy der.Bu durumda herhangi bir tarihte o müşterinin medeni durumu nedir rahatlıkla bulabiliriz.

Tüm tiplere bakmak isteyenler için: https://www.datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.html

Surrogate Key Nedir?
Normalde kaynak tablonuzda var olmayan tabloya dair hiç bir bilgi anlatmayan sıralı bir sayı dizisidir.Kaynak tablonuzun PK'sı yada Natural Key olarak ifade edilen (TC alanı gibi) alanlar olamaz.Tüm Dimensionlarda olmasını ve factleri böyle bağlamak gerektiğini söylüyor Kimball fakat gerçekte bunu yapan çok fazla yok genelde SCD yapılarında zorunluluktan uygulanıyor.SCD tablolarında neden bir zorunluluk derseniz üstte bahsettiğim gibi genelde type 2 tercih ediliyor ve type 2 yaptığınız aynı PK için ikinci bir kayıt oluştuyorsunuz tabloda fakat bu durumda ilgili kaydı ifade edecek bir tekil anahtarınız olmuyor bu yüzden SCD tablolarında genelde bir de SK alanı neredeyse zorunludur.

CDC Nedir?
Change Data Capture ifadesinin kısaltmasıdır.ETL süreçlerinde çok önemli bir yeri vardır.Çünkü ETL sürecinizde bir jobınız var ve 100 milyonluk bir tabloyu aktarıyor her gün 100 milyon kaydı aktarmakla uğraşmak yerine sadece ETL jobımızın son çalışmadan sonra değişen verileri(1 milyon olsun) almak çok daha kısa sürecektir.Olabildiğince tüm ETL joblarını böyle yapmak ETL sürecinizi inanılmaz hızlandırır.ETL sürecinin hızı çok önemlidir çünkü gece ETL süreciniz hata alabilir veya sourceda değişiklik yapılabilir ve güncel datalar üzerinden raporlama istenebilir bu durumda sürecinizin ne kadar kısa ise o kadar rahat hareket edersiniz.ETL süresi kısaltmada bu yöntem elbette tek başına yeterli değil diğerlerini başka bir yazıda uzunca yazağım.
CDC yapabilmek için elinizde bir tarih veya sayısal bir alan olmalı DWH'ıma yüklediğim son kayıt budur diyebilmek için.Ve bu alandan büyük verileri sadece çektiğinizde daha ufak bir data seti elde edersiniz.Burdaki önemli nokta şu PK alanınızı alamazsınız.Çünkü CDC içersinde insert kadar update de barındırır.Son yüklediğiniz PK 100 olsun 50 tane insert 30 tane update geldi bu durumda siz sadece 50 tane insert olanı yakalayabilirsiniz çünkü 100den büyükleri alacaksınız.Fakat aslında siz 30 update'i almanız lazım o yüzden update veya insert geldiğinde tablonuzdaki değeri güncelleyen bir alan olmalı.50 inserti yüklemek kolay çünkü zaten DWH'da yok fakat updateler için DWH'da var olan kayıtları silip yüklemeniz gerekir.

BI Nedir?
Business Intelligence ifadesinin kısaltmasıdır.Amaç ilgili toollarla anlamlı analizler,dashboardlar hazırlamak ve çıkarımlar yapmaktır.DWH'ın üzerine kuruludur aslında tüm ETL ve DWH işlerimizin görünmeye başladığı kısım budur.Yaptığımız DWH görsele dönüşür.Ayrıca son kullanıcılara(iş birimleri) self servis imkanı da sağlamamıza yarar ve herkes istediği kırılım ve filtrelerde analizlerini yapabilir.Kullanıcının her isteği aslında ilgili tablolara bir sql isteği olarak gelir.Her BI toolu kendine özgü farklılıklar ve yetenekler barındırsa da genel amaç dashboardlar-self servis yapılarıdır ve herkesin yetkisi dahilinde bilgiye ulaşmasıdır.

En çok kullanılan BI Toolları şöyledir:Power BI,Tableau,OBIEE,SAP Business Objects,Qlik vb.

Row Level Security Nedir?
RLS,BI dünyasının en önemli konularından biridir.Veri setlerinin kullanıcı bazlı kısıtlanması demektir.BI toolunuzda bir dashboard hazırladınız 10 farklı şirketin gelir kalemleri var içinde diyelim.Kullanıcının sadece kendi firmasını görmesini istersiniz genelde veya bölge müdürünün kendisine bağlı 3 şirketi ceo'nun 10 şirketi de görmesini istersiniz işte bu yapılan kullanıcı bazlı yetkilendirme RLS olarak adlandırılır.

Data Mining Nedir?
Büyük veri setlerinde şirketin karlılığını arttıracak anormallikler,farkında olunmayan ilişkiler ve korelasyonların tespiti için yapılan işlemler denebilir.Genel istatistik bilgileriyle,makine öğrenmesi veya yapay zekayla verilerinizi analiz edebilir çıktılar yakalayabilirsiniz.DWH dünyasında DWH'cıların direk ilgi alanı olmamakla beraber data mining yapan ekiplere veri sağladığımız için aslında ilişkimiz bulunmaktadır.Sosyal medya veya web sitesi ziyaretlerinizde karşınızda çıkan reklamlarda kişiselleştirilmiş reklamlar olup bir mining sonucunda size o reklam sunuluyor veya netflix de bir bir dizi spotifyda bir sanatçıda.

Basit bir excelde de çıkardığınız bulgular basit düzeyde bir mining sayılabilir fakat piyasada en çok kullanılan ve daha profesyonel düzeyde kullanılan toollar ve diller şöyle:SPSS,SAS,R,Pyhton,Knime,Rapidminer vb.

Flat File Nedir?
Tek bir tabloluk DB'ler olarak düşünebiliriz.Her bir satır bir kaydı ifade etmektedir.CSV n satırlı 1 sütunlu veri var gibi gözükebilir bunlar separator'u(virgül,tab,space vb.) aracılığıla parçayalıp gerçek sütun sayısına ulaşabiliriz.Yönetimi ve taşınması kolay olduğundan ve herkes tarafından kullanıldığından tercih edilen yapılardır.DWH'da çok fazla işimiz düşer çünkü bir iş biriminden data alıyorsanız anlık değişebilen hedefler gibi genelde sizinle CSV veya Excel paylaşırlar.Bunları ETL süreciyle DWH'a işleminiz gerekir.Excel,CSV,TXT,XML vb tüm dosyalara genelde Flat File deriz.

FTP Nedir?
File Transfer Protocol'ün kısaltmasıdır.
DWH'da sık sık işimiz düşer.
İki farklı bilgisayar arasında dosya transferi yapmanıza izin veren bir yapıdır.Bazen FTP'de bizim için source veya target olabilir.
SFTP olarak güvenlikli versiyonu daha çok tercih edilmektekdir.Aynı şey olmakla beraber SFTP daha güvenlikli bir yapıdır.Size bir kullanıcı ve şifre verdikten sonra ilgili SFTP'ye bağlanabilirsiniz.Burda da yetkilendirmeler mevcuttur her klasör ve dosya için okuma,yazma ve execute etme yetkisi verilebilir ihityaca göre.Bunlar r,w,x olarak ifade edildiği gibi rakamlarla da edilebilir.
Filezilla,Winscp gibi araçlarla bağlanılabilir.

Daha fazlası için : https://www.hostknox.com/tutorials/ftp/file-permissions

Cron Nedir?
Cron Unix işletim sisteminde belirli aralıkla komutlarınızı otomatik yürütmek için yazılan bir zamanlayıcıdır.Windowsdaki Task Scheduler'a benzer.
Bazı ETL toolların da geliştirme işlemi yaptıktan sonra bu ETL job'ımızın her gece veya her saat başı çalışmasını isteriz.İşte bu durumlarda sh uzantılı dosyamız için bir cron yazarız ve unix makinemizde istenilen aralıkta bu job çalışmaya başlayacaktır.Zamanlayıcısı bir tık farklı merak edenler için link bırakıyorum.

https://www.freeformatter.com/cron-expression-generator-quartz.html