**VERİTABANI SİSTEMLERİ ÖZET**

Günümüzde tüm işletmelerin gelişebilmesi ve güvenli bir şekilde verilerini tutmaları için veri tabanına ihtiyaç vardır. İşletmeler ile ilgili verilerin veritabanlarına doğru şekilde kayıt altında alınması, işletme süreçleri ile ilgili temel verilerin raporlanması veya karar destekleri üretme için veri tabanındaki verilerin değerlendirilmesi vb. birçok alanda veritabanı programlama önemli bir yere sahiptir.Verilerin tutulması ve bilgi yönetim sistemleri amaçlı ilk çalışmalar 1960’lı yıllarda başlamıştır. Dönüm noktası, 1970 yılında Edgar Codd ’un ilişkisel veri modelini önermesi olmuştur. Bu aynı zamanda günümüzde yaygın olarak verilerin saklanmasında kullanılan ilişkisel veritabanı yönetim sistemi’nin de temelini oluşturmaktadır. Genel olarak Veritabanı Yönetim Sistemi (VTYS), veri tabanlarını kurmayı, oluşturmayı, kullanıcı haklarını tanımlama, verilere erişim olarak tanımlanabilir.

Farklı üreticilere ait VTYS üzerinde işlemleri yapmak üzere en yaygın olarak ortak kullanılan dil Yapısal Sorgu Dili (SQL)’dir. SQL’in ilk sürümü 1970’lı yıllarda SEQUEL olarak ortaya çıkmış olup 1980’li yıllarda ANSI/ISO standartlarında SQL olarak yeni sürümleri ortaya çıkmıştır.

SQL komutları kullanım amaçlarına göre üç genel kategoriye ayrılır. Bunlar;

Veri Tanımlama Dili (DDL-Data Definition Language)

Veri İşleme Dili (DML Data Manipulation Language)

Veri Kontrol Dili (DCL-Data Control Language)

Eğer işletme süreçlerinde veri isleme için akış kontrolü, döngü vb. gerektiren bir dil ihtiyaç olursa, standart SQL bu işlemlerin yapılması için yeterli olmaz. Bu durum için çözüm yöntemleri veritabanı ile etkileşimi de içermek üzere üç sınıfa toplanabilir:

Gömülü SQL(Embedded SQL)

Veritabanı bağlantısı için yazılım kütüphaneleri kullanma

Veritabanı programlama dili kullanma

İşletmelerin veritabanı sistemlerinde çok farklı kullanıcılar için veriler bulunmaktadır Bazı veriler belirli kullanıcıların ortak erişimine açık iken bazı verilerin ise tüm kullanıcılara açık olmaması gerekir. İşletmelerin veritabanı sistemleri birçok kullanıcı için tasarlanmış ve kullanılmaktadır. İşlemlerin doğru yönetimi için VTYS’de hareket (transaction) yönetimi önemli bir yere sahiptir

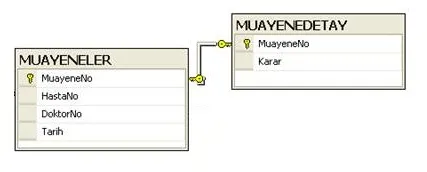
SQL Server 2014 Express yazılımını kurmadan önce birtakım hazırlık ve kontrolleri yapmanız gerekmektedir. Yazılımın çalışabilmesi için bilgisayarınızın sağlaması gereken minimum sistem gereksinimleri sağlayıp sağlamadığı, kontrol ederek kurulum dosyasının indirilmesi gerekmektedir. Yazılımın kurulumu için ilgili dosyayı açıp klasördeki kurulum dosyasının çalıştırılması gerekmektedir.

Kurulum,Evrensel Kurallar , Kurulum Tipi, Lisanslama, Özellik Seçimi, Oluşum Yapılandırması ,Oluşum Yapılandırması , Sunucu Yapılandırma , Veritabanı Motoru Yapılandırması, Kurulum İşlemi ve Tamamlanma.

Kurulum için bu işlemlerin yapılması gerekir.

Veritabanı yönetim sistemi (VTYS), veritabanlarını tanımlama, oluşturma, işleme, yetkilendirme ve bakım için kullanılan programlar bütünüdür.Geleneksel dosya sistemlerinde, veriler ayrı dosyalarda saklanırken, veritabanı sistemi ilişkili verilerin mantıksal ve fiziksel olarak tanımlandığı bir yapı sunar.Geleneksel dosya sistemleri, her alt sistem veya uygulama için ayrı dosyalar kullanır.Veritabanı yönetim sistemleri, tüm verileri tek bir merkezde toplar ve veri tekrarını önler.Veritabanı yönetim sistemlerinde veri güvenliği ve gizliliği sağlanabilirken, geleneksel dosya sistemlerinde bu daha zordur.VTYS mimarisi, istemci/sunucu mimarisi ile modüler bir yapılanmayı içerir ve veritabanının mantıksal yapısını tanımlamak için veri modelleri kullanılır.Üç şema mimarisi, içsel, kavramsal ve dışsal düzeylerde tanımlanan şemaları içerir ve VTYS'nin işlevselliğini sağlar.Mantıksal veri bağımsızlığı, kavramsal şemanın değiştirilmesini gerektirmeden dışsal şemada veya uygulama programlarında değişiklik yapabilme yeteneği olarak tanımlanırken, fiziksel veri bağımsızlığı ise kavramsal şemada yapılan değişikliklerin içsel şemayı etkilemeden yapılabilmesini ifade eder.Bu bağlamda, fiziksel veri bağımsızlığının, fiziksel ayrıntıları içeren veritabanı ve dosya sistemlerinde yaygın olduğu belirtilmiştir.Ancak, kavramsal veri bağımsızlığına ulaşmanın daha zor olduğu vurgulanmıştır, çünkü bu tür bağımsızlık yapısal değişikliklere izin verirken uygulama programlarını etkilememektedir.Metin ayrıca, veritabanı yönetim sistemlerinin (VTYS) farklı düzeylerdeki kullanıcı gruplarına uygun dil ve arayüzler sağlaması gerektiğini belirtmektedir.VTYS'nin tamamlanmasıyla veri tanımlarının yapıldığı ve VTYS kataloğuna derlendiği, bu sürecin veritabanı yaklaşımının temel özelliklerinden biri olduğu ifade edilmiştir.VTYS'nin veri işleme dili olarak bir dil kullandığı ve günümüzde genellikle bu dillerin ayrı olmadığı, SQL gibi geniş kapsamlı birleştirilmiş dillerin kullanıldığı belirtilmiştir.Hiyerarşik veritabanı, 1960'lar ve 1970'lerde yaygın olarak kullanılan en eski veri modelidir.Bu modelde veriler, kök ve dallar biçiminde hiyerarşik bir yapıda düzenlenir.Herhangi bir veriye erişmek için arama işlemi kök düğümden başlayarak alt düğümlere doğru devam eder.Bir dal silindiğinde, bu dala bağlı tüm alt dallar ve veriler de silinir.Bu tür, hiyerarşik veri modelinin sınırlamalarını gidermek için geliştirilmiş ve ağ veri modelini temel alır.Hiyerarşik modelde tek yönlü ilişkiler bulunurken, ağ modelinde düğümler arasında çoklu ilişkiler mümkündür, bu da daha karmaşık veri ilişkilerini yönetmeyi sağlar.Bu veritabanı yapısında, birden çok ilişki biçimi kullanılabilir, ancak günümüzde, ilişkisel veri modeli hemen hemen tüm veritabanı yönetim sistemlerinde tercih edilen bir modeldir.Günümüzde ve gelecekteki pek çok uygulama, sadece metin bazlı verileri değil, aynı zamanda multimedya ögelerini de içeren veritabanı yönetim sistemlerine ihtiyaç duyar.Geleneksel veritabanı sistemleri, grafik ve multimedya ögelerini işlemek için uygun değildir, bu nedenle nesneye yönelik veri modelleri geliştirilmiştir.Bu modeller, multimedya içeriklerini ve farklı kaynaklardan gelen verileri bir araya getiren web uygulamalarını yönetmek için kullanılabilir.Nesneye yönelik veritabanları, ilişkisel veri modellerinden farklı olarak daha karmaşık veri türlerini işleyebilir, ancak işlem hızı açısından ilişkisel veritabanlarından daha yavaş olabilir.Bu nedenle, günümüzde hem ilişkisel hem de nesneye yönelik veri modellerini birleştiren veritabanı yönetim sistemlerinin yaygınlaştığı gözlemlenmektedir.

Bire-Bir Örnek:



Bire-Çok Örnek:



Çoka-Çok Örnek:

