**İrem ak 2. sınıf 1. öğretim 223010710013**

**Veri tabanı**

* Düzenli bilgiler topluluğudur.
* Bilgisayar ortamında saklanan düzenli verilerdir.
* Bilgisayar terminolojisinde, sistematik erişim imkanı olan, yönetilebilir, güncellenebilir taşınabilir, birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir.
* Bilgisayarda sistematik şekilde saklanmış, programlarca işlenebilecek veri yığınıdır.

**Veri Tabanı Yönetim SistemiVTYS**

Veri tabanı tanımlamak, yaratmak, yaşatmak ve veri tabanına denetimli erişim sağlamak için kullanılan yazılım sistemidir.

**DOSYA SİSTEMLERİNİN SAKINCALARI**

Klasik dosya sistemleri kullanılmaya başlandıktan sonra bazı dezavantajları olduğu ortaya çıkmıştır.

**Bunlar şöyle sıralanabilir :**

**Veri tekrarı:** Aynı veri çeşitli dosyalarda birden fazla yer alabilmektedir buda sistemin hantallaşmasına neden olur. Mesela bir stok dosyasında stok numarası verisinin malzeme dosyasında, fatura dosyasında ve ambar girişi dosyasında yer alması gibi.

**Verinin birkaç dosyada güncellemesi:** Veri birden fazla dosyada tekrar edilebildiği için, verinin bir dosyada güncellenip diğerlerinde güncellenmemesi Veri Bütünlüğünün (Data Integrity) bozulmasına neden olabilir. Buna bağlı olarak birbiri ile çelişen raporlar üretilebilir. **Belleğin tekrarlı bilgi nedeniyle israfı:** Aynı verinin birden fazla dosya içinde bulunması nedeniyle kullanılan veri hard diskte fazla yer işgal edecek. Yani hard disk tekrarlı veriler için kullanılmış olacaktır.

**Sadece belirli bir dilin kullanılması** :Verilerin dosya sisteminde saklandığı ortamlar için değişik programlama dillerinden bir tanesi kullanılır. Kullanılan bu programlama dili ise SQL dili gibi esnek değildir.

**Veritabanı yönetim sistemlerinin üstünlükleri**

* Veri tekrarı ve veri tutarsızlığını önler.
* Veri paylaşımına olanak verir.
* Uzmanlık bilgisine ihtiyaç duyulmayacak derecede kullanım kolaylığı sağlar.
* İhtiyaç duyulan veriye, tanımlanmış kullanıcı yetkileri kapsamında kolaylıkla erişilmesini sağlar.
* Veri güvenliği ve gizliliğini güçlü bir biçimde yerine getirir.
* Veriler ve uygulamalarla ilgili standart yapı ve kuralların olması kullanım kolaylığı sağlar.
* Verileri yedekleme ve kurtarma konusunda kolaylık sağlayan programlar barındırır.

**VERİTABANI KULLANICILARI**

* Veritabanı ile herhangi bir şekilde etkileşimde olan kişi ya da kişiler veritabanı kullanıcısı olup aşağıdaki gibi sınıflandırılabilirler:
* Veritabanı Sorumluları
* Veritabanı Yöneticisi
* Veritabanı Tasarımcısı
* Son Kullanıcılar
* Standart Kullanıcılar
* Sıradan ya da Parametrik Kullanıcılar
* Gelişmiş Kullanıcılar
* Bağımsız Kullanıcılar
* Sistem Analistleri ve Uygulama Programcıları
* **Veritabanı Sorumluları:**
  + Veritabanı Yöneticisi: Veritabanına erişim yetkilerini belirler, kullanımını düzenler, güvenliğini sağlar, yazılım ve donanım kaynaklarını yönetir.
  + Veritabanı Tasarımcısı: Verilerin tanımlanması, depolanması ve uygun yapıların seçilmesinden sorumludur. Veritabanı kullanıcılarının ihtiyaçlarını anlayarak tasarım yapar.
* **Son Kullanıcılar:**
  + Standart Son Kullanıcılar: Nadiren erişim yaparlar, gelişmiş sorgu dillerini kullanarak istedikleri bilgilere ulaşırlar.
  + Sıradan Son Kullanıcılar: Temel iş fonksiyonları için düzenli sorgular ve güncellemeler yaparlar. Farklı konum ve görevlerde olabilirler.
  + Gelişmiş Son Kullanıcılar: Karmaşık gereksinimlerini karşılamak için veritabanının özelliklerini kullanırlar.
  + Bağımsız Son Kullanıcılar: Hazır paket programlarını kullanarak kişisel veritabanlarını yönetirler.
* **Sistem Analistleri ve Uygulama Programcıları:**
  + Sistem Analistleri: Son kullanıcıların gereksinimlerini belirler, standart işlemleri ve ayrıntıları saptar.
  + Uygulama Programcıları: Analistlerin belirlediği ayrıntıları program haline getirir, test eder, hataları ayıklar ve belgeleme yaparlar.

**VTYS Mimarisi:**

* VTYS'ler zaman içinde gelişerek tek bir entegre sisteme sahip olmaktan çıkıp istemci/sunucu mimarisine dönüşmüştür.
* İstemci modülü, VTYS'nin kullanıcı iş istasyonlarında veya kişisel bilgisayarlarda çalışan parçasıdır ve genellikle uygulama programlarını ve kullanıcı arayüzlerini içerir.
* Sunucu modülü, veri deposuna erişim ve sorgulama işlemlerini sağlayan fonksiyonları içerir.

**Üç Şema Mimarisi:**

* VTYS, veri tanımlarını (şema) saklamak için kendi kendine tanım yapabilen bir katalog kullanır.
* **İçsel Düzey (Fiziksel Düzey):** Fiziksel depolama yapısını tanımlayan içsel şemayı içerir. Fiziksel şema, veriyi depolama ayrıntılarının tamamını ve veritabanına erişim yollarını tanımlayan fiziksel veri modelini kullanır.
* **Kavramsal Düzey:** Tüm veritabanının yapısını tanımlayan kavramsal şemayı içerir. Kavramsal şema fiziksel depolama ayrıntılarını gizler ve verilerin tipi, ilişkileri, kullanıcı işlemleri ve kısıtlamalar gibi tanımlara odaklanır.
* **Dışsal Düzey (Görünüm Düzey):** Kullanıcı gruplarının ilgilendiği belirli veritabanı bölümlerini tanımlayan dışsal şemaları içerir. Her dışsal şema, bir grup kullanıcının görüntüsünü sağlar. Bu sayede kullanıcılar, diğer kısmı gizlenmiş veritabanı ile etkileşime geçerler.
* Veritabanı sistemleri, kullanıcıların dış şemalarından yapılan istekleri kavramsal şema isteğine ve daha sonra içsel şema isteğine dönüştürmek zorundadır. Bu dönüşümler eşleştirme (mapping) olarak adlandırılır ve farklı şema düzeyleri arasında gerçekleştirilir.

Bu üç şema mimarisi, veritabanı sistemlerinin karmaşıklığını azaltır ve kullanıcıların verileri daha iyi anlamalarına ve yönetmelerine yardımcı olur. İçsel düzey, verilerin fiziksel depolama ayrıntılarını içerirken, dışsal düzey kullanıcıların ilgilendiği belirli veritabanı bölümlerini yansıtır ve kavramsal düzey ise genel veri yapısını tanımlar. Bu ayrım, veritabanı yönetimini daha etkili ve yönetilebilir hale getirir.

**Veri Bağımsızlığı:** Veri bağımsızlığı, veritabanı sistemlerinde üç farklı şema düzeyi arasındaki ilişkileri tanımlar. Bu bağımsızlık üç ana kategoriye ayrılabilir:

* **Mantıksal Veri Bağımsızlık**: Kavramsal şemanın dışsal şemalarda veya uygulama programlarında değiştirilebilmesini ifade eder. Bu, veritabanının genişletilmesi, kısıt değişiklikleri veya veritabanının daraltılması gibi değişiklikleri içerebilir.
* **Fiziksel Veri Bağımsızlık:** Kavramsal şemada bir değişiklik yapılmaksızın içsel şemada değişiklik yapabilme kapasitesini ifade eder. Bu, içsel şemalarda yapılan değişikliklerin dışsal şemalara etki etmeden gerçekleştirilebilmesini sağlar. Örneğin, veritabanının performansını artırmak amacıyla indekslerin eklenmesi gibi içsel şema değişiklikleri buna örnektir.

**Veritabanı Yönetim Sistemlerinin Kullanılan Dilleri**

* **Veri Tanımlama Dili (DDL - Data Definition Language):** Kavramsal ve fiziksel şemaları tanımlamak ve aralarındaki ilişkileri belirlemek için kullanılır. Veritabanı yöneticisi ve tasarımcıları tarafından kullanılır.
* **Depolama Tanımlama Dili (SDL - Storage Definition Language):** İçsel şemanın fiziksel özelliklerini tanımlamak için kullanılır. Depolama yapısının oluşturulması, performans iyileştirmeleri ve güncellemeleri içsel şema değişikliklerine örnek olarak verilebilir.
* **Görünüm Tanımlama Dili (VDL - View Definition Language):** Üç düzeyli mimaride kullanıcı görünümlerini ve bu görünümlerin kavramsal şemadaki eşleştirmelerini belirlemek için kullanılır.
* **Sorgu Dili (SQL - Structured Query Language):** Veri işleme ve sorgulama işlemleri için kullanılır. Sorguları oluşturmak, veri çekmek, eklemek, silmek ve güncellemek için kullanılır.

**Veritabanı Yönetim Sistemlerinin Bileşen Modülleri:** Veritabanı yönetim sistemleri, veritabanı ve kataloğu yönetmek ve veri işleme işlemlerini gerçekleştirmek için çeşitli bileşenlerden oluşur. Bu bileşenler şunları içerir:

* **Veritabanı Yönetici Personeli:** Veritabanı tanımları ve şemaları oluşturur ve değiştirir.
* Veri Tanımlama Dili Derleyicisi: Veri tanımlama dili ile belirtilen şemaları işler ve VTYS kataloğunda saklar.
* **Sorgu İşleyici (Query Processor):** Kullanıcıların veya uygulamaların veritabanına gönderdiği sorguları işler, çözümlemesi yapar ve uygun algoritmayı seçer.
* **Sorgu Eniyileyici (Query Optimizer):** Sorguları optimize eder, işlemleri yeniden düzenler ve performansı artırmak için gerekli indeksleri kullanır.
* **Depolama Yöneticisi:** Disk üzerinde veritabanı ve kataloğun depolanmasını kontrol eder ve veri erişimini optimize eder.

**VERİTABANI TÜRLERİ**

* **Hiyerarşik Veritabanı:** Bu veritabanı türü, verilerin hiyerarşik bir ağaç yapısında düzenlendiği ve ilişkilerin ebeveyn-çocuk ilişkisiyle tanımlandığı bir modeli kullanır. Ancak veri tekrarı ve erişim zorlukları gibi sorunlara yol açabilir.
* **Ağ Veritabanı:** Ağ veritabanı, hiyerarşik modele bir alternatif olarak geliştirilmiştir. Bu modelde, her düğümün birden fazla ebeveyn ve çocuk düğümü olabilir. Veriler daha esnek bir şekilde ilişkilendirilebilir. Ancak karmaşık olabilir ve performans sorunlarına yol açabilir.
* **İlişkisel Veritabanı:** İlişkisel veritabanları, tablolar ve ilişkiler aracılığıyla verilerin düzenlendiği ve sorgulandığı bir modeli kullanır. Bu model, SQL (Structured Query Language) gibi bir sorgu dili kullanır ve günümüzde en yaygın olarak kullanılan veritabanı türüdür. İlişkisel veritabanları, veri tekrarı sorunlarını azaltır ve karmaşıklığı azaltır.
* **Nesneye Yönelik Veritabanı:** Nesneye yönelik veritabanları, çok çeşitli veri tiplerini ve multimedya içeriklerini saklama ve yönetme yeteneği sunar. Bu tür veritabanları, özellikle nesne tabanlı programlamayı destekleyen uygulamalarda kullanılır. İlişkisel veritabanlarından daha karmaşık veri türlerine sahiptirler.