**1.12. ÇALIŞMA SORULARI**

**1**. Geleneksel dosyalama sistemlerinin çalışma şeklini açıklayarak bildiğiniz bir programlama dili yardımıyla txt dosyadan veri okuma ve yazma için bir uygulama geliştirin.

#include <stdio.h>  
#include <conio.h>  
int main()  
{  
int a = 25, b = 50; /\*Dosya okuma ve yazma c uygulama örneği.\*/  
FILE \*fptr,\*fptr1;  
fptr=fopen("okuma.txt","w");  
fprintf ( fptr, "%d %d\n", a, b ) ;  
fptr1 = fopen ("okuma.txt", "r" );  
fscanf ( fptr1, "%d %d", &a, &b);  
printf ("%d %d\n", a, b ) ;  
getch();  
}

Geleneksel dosyalama sistemleri; alfabetik dosyalama, numaralı dosyalama, kronolojik dosyalama ve karma dosyalama sistemleri kullanılır.

**Alfabetik Dosyalama Sistemi:** Alfabetik dosyalama sistemi, adından da anlaşılacağı üzere, gerek dosyalar içindeki kayıt ve belgelerin, gerekse dosyaların, A’dan Z’ye gruplandırılarak sıraya konulmasını gerektirir. İki harf arasına rehberler konularak dosyalara erişim kolaylaştırılır. Kavalye ya da seperatör de denilen bu rehberler, üzerinde harflerin yazılı olduğu karton ya da plastikten yapılmış, genellikle renkli, dosyaya tutturulabilecek şekilde tasarlanmış araçlardır.

**Numaralı Dosyalama Sistemi:** Dosyalama sisteminde, alfabetik sistemde olduğu gibi harfler değil rakamlar kullanılır. Dosyalar arasına yine alfabetik sistemde olduğu gibi rehberler konur. Numaralı dosyalama sistemi, sıra numaralı (ardışık) sistem ve ondalık (desimal) sistem olarak iki şekilde kullanılır. Bunların yanı sıra ortadaki asal sayıya ve sondaki asal sayıya göre numaralı dosyalama sistemleri de tasarlanabilir.

**Kronolojik dosyalama:** Kronolojik dosyalamada kayıt ve belgeler gün, ay, yıl, yüzyıl gibi tarih sırasına dizilerek dosyalanır. Özellikle tarihlerin önemli olduğu bankaların kredi servisleri, mahkemeler, askerlik şubeleri gibi kurum ve kuruluşlarda kullanılan bu sistem, yapılan işlemlerin özellikle de borç-alacak ilişkilerinin tarih sırasına göre izlenmesini kolaylaştırır. Kronolojik sistem amacına hizmet ettiğinde çok kullanışlı olmasına rağmen, uygulama alanı oldukça dardır. Genellikle karma sistemler içinde kullanılır.

**Karma dosyalama:** Karma dosyalama sistemi, diğer sistemler tek başlarına etkili ve yeterli olmadıkları zaman tercih edilmektedir. Bu sistemde ana grupların ve alt grupların tasarlanmasında birbiriyle uygun ve sistemin gereksinimlerini karşılayan dosyalama sistemlerinden iki ya da üç tanesi bir arada kullanılır. Örneğin, ana grup tarih sırasına göre düzenlenirken alt gruplar alfabetik esasa ya da coğrafik bölgelere göre düzenlenebilir. Ana grup coğrafik bölge esasına göre; alt gruplar da numara ya da tarih sırasına göre gruplandırılabilir. Ana grup konuya göre, alt gruplar da numara sırasına, tarih sırasına ya da coğrafik bölgelere göre gruplandırılabilir. Son olarak; ana grup alfabetik sırayla, alt grup da ondalık numaraya göre düzenlenebilir.

**2.** Geleneksel dosya sistemleri ile veritabanı yönetim sistemlerinin benzerlik ve farklılıklarını açıklayınız.

**Geleneksel Yaklaşımın (Dosya - İşlem Sistemi) Sakıncaları**

• Veri tekrarı ve veri tutarsızlığı

• Verinin paylaşılamaması

• Uygulamalardaki her yeni gereksinimin ve değişikliğin yalnız uzman kişiler tarafından karşılanabilmesi

• Veriye erişim ve istenen veriyi elde etme güçlükleri

• Karmaşık veri saklama yapıları ve erişim yöntemlerini bilme zorunluluğu

• Bütünlük (integrity) sorunları

• Güvenlik, gizlilik sorunları

• Tasarım farklılıkları, standart eksikliği

• Yedekleme, yeniden başlatma, onarma gibi işletim sorunları

**Veri Tabanı Yaklaşımının Yararları**

• Ortak verilerin tekrarının önlenmesi; verilerin merkezi denetiminin ve tutarlılığının sağlanması

• Veri paylaşımının sağlanması

• Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı mimarilerle kullanıcılardan gizlenmesi

• Her kullanıcıya yalnız ilgilendiği verilerin, alışık olduğu kolay, anlaşılır yapılarda sunulması

• Sunulan çözümleme, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması.

• Veri bütünlüğü için gerekli olanakların sağlanması, mekanizmaların kurulması

• Güvenlik ve gizliliğin istenilen düzeyde sağlanması

• Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi

**3.** VTYS’nin geleneksel sisteme göre üstün özelliklerini açıklayınız.

**Veri Tekrarı (“Data Redundancy”) :** Aynı verinin sürekli tekrarlanmasını önler.Aynı tablo içerisinde, farklı bilgisayarlarda vb..; Veri Tekrarı (“Data Redundancy”) azaltılır ya da yok edilir.

**Veri Tutarlılığı (“Data Consistency”):**Aynı verinin değişik yerlerde birkaç kopyasının bulunması “bakım” zorluğu getirir: bir yerde güncellenen bir adres bilgisi başka yerde güncellenmeden kalabilir ve bu durum veri tutarsızlığına (“Data Inconsistency”) yol açar.

**Veri Paylaşımı / Eşzamanlılık (“Concurrency”):**Veri tabanı yönetim sistemi (VTYS) kullanılmadığı durumlarda veriye sıralı erişim yapılır. Yanı birden çok kullanıcı aynı anda aynı veriye erişemez. Bir VTYS’de ise aynı veritabanlarına saniyede yüzlerce, binlerce erişim yapılabilir.

**Veri Bütünlüğü (“Data Integrity”):**Bir tablodan bir öğrenci kaydı silinirse, öğrenci var olduğu diğer tüm tablolardan silinmelidir.

**Veri Güvenliği (“Data Security”):**Verinin isteyerek ya da yanlış kullanım sonucu bozulmasını önlemek için çok sıkı mekanizmalar mevcuttur. Veri tabanına girmek için kullanıcı adı ve şifreyle korumanın yanı sıra kişiler sadece kendilerini ilgilendiren tabloları ya da tablo içinde belirli kolonları görebilirler.

**Veri Bağımsızlığı (“Data Independence”):**Programcı, kullandığı verilerin yapısı ve organizasyonu ile ilgilenmek durumunda değildir.

**4.** Veritabanın görevini açıklayarak kullanıldığı alanlara örnek veriniz.

Bilgisayar ortamında veri saklama ve erişiminde geçmişten günümüze değişik yöntemler ve yaklaşımlar kullanılmıştır. Bunlardan Geleneksel Yaklaşım (Dosya - İşlem Sistemi) verileri ayrı ayrı dosyalarda gruplamaya dayanır. Verileri saklamak için programlama dillerinde kullanılan sıralı (Sequential) ve rastgele (Random) dosyalama sistemleri gibi. Birbiriyle ilgili olan ve aynı gruba dahil olan veriler bir dosyada, bir başka gruba dahil olan veriler de başka bir dosyada tutulurdu. Geleneksel Yaklaşımın birçok sakıncası vardır ve bu sakıncaların beraberinde getirdiği sorunların üstesinden gelebilmek için de Veri Tabanı Yaklaşımı zamanla Geleneksel Yaklaşımın yerini almıştır. Günümüzde veriler artık Veri Tabanı Yaklaşımı ilkesine göre VTYS’lerde tutulmakta ve işlenmektedir.

Günümüzde, bir çok alandaki veri işlemlerinde pek çok Veri Tabanı Yönetim Sistemleri programları yaygın olarak kullanılmaktadır. Birbirinden farklı isimler adı altında anılan bu programlar için bir çok nesne birbiri ile aynı temel işlevi yerine getirmekte olup, yaklaşık olarak aynı teorilere dayanarak çalışırlar. Veri tabanı, bir kuruluşun uygulama programlarının kullandığı operasyonel verilerin bütünüdür. Veritabanı Yönetim Sistemleri, verilerin fiziksel hafızadaki durumlarını, kullanıcıların erişimlerini düzenleyen sistemlerdir. İlişkisel VTYS’ler günümüzde yaygın olarak kullanılmaktadır.

**5.** Tablo, satır ve sütun kavramlarını açıklayınız.

* Tablo;
  + Veritabanı içerisinde verilerin tutulduğu kısımdır.
  + Satır ve sütunlardan oluşmaktadır.
  + Tablo içerisinde her satır bir kaydı ifade ederken , her sütun her bir kayıta ait bir özelliği ifade eder aynı zamanda her bir sütuna field yada alan denir.
* Satır ve Sütun;
  + Tabloyu oluşturan temel kavram.
  + Sütun tablo içerisinde tutulan her bir veri türüne verilen isimdir, tablo içerisinde birden fazla kullanılır.
  + Satır sütuna ait veri gurubudur. Örneğin bir öğrencinin no,ad,soyad bilgilerinin tamamı satırı ifade etmektedir.

**6.** Birincil anahtar ve yabancıl anahtar kavramlarını ve farklılıklarını açıklayınız.

Primary Key benzersiz bir tablodaki her satırı tanımlamak için kullanılır.Birincil anahtar, bir tablo üzerinde bir veya birden çok alan ile oluşabilir. Birden fazla alanı birincil anahtar olarak oluşturulduğunda buna Composite key (birleşik anahtar) denir.Birincil anahtar kısıtlayıcısınıı neredeyse her tabloda kullanırız. Bu bize hız kazandırır. Primary Key olan alanlar Null değerler olamaz ve bu bölüme aynı değerler girilemez. Genel de otomatik olarak artan değerler için geçerlidir.Otomatik artan değerler Identity komutu ile sağlanır. Identity komutundan sonra işlemin kaçtan başayacağı ve kaçar kaçar artacağı belirtilir. Identity(1,3) 1'den başlayacağını ve 3'er 3'er artacağını gösterir bize.

Foreign Key bir tabloya girilebilecek değerleri başka bir tablonun bir belli alanında yer alabilecek veri grubu ile sınırlandırmaya ve en önemlisi de ilişkilendirmeye yarar. Bir tabloya foreign key eklediğimizde referans edilen ve referans eden iki tablo arasında bir ilişki kurulur. Bir tablo için foreign key ayarlanmışsa bu tabloya girilecek her kaydın referans edilen tablodaki kayıtla eşleşmesi gerekir. Eğer foreign key olan kolon null değerler alabiliyorsa eşleşmeye bakılmaz.

**7.** Veritabanı kullanıcı türleri nelerdir?

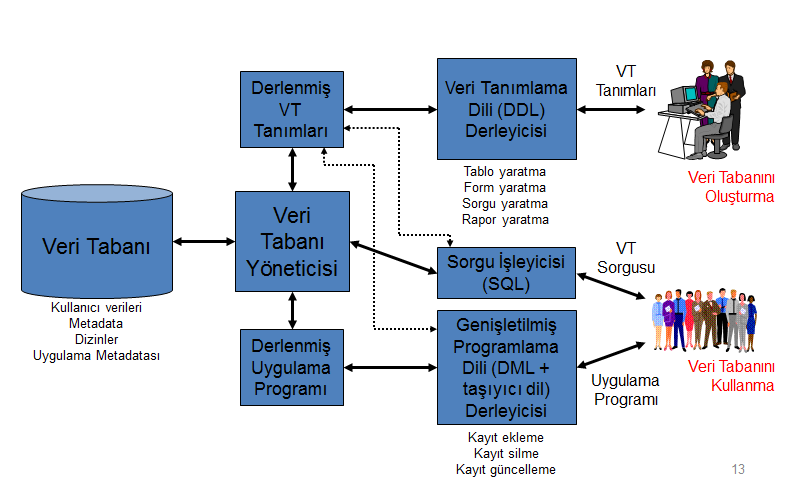
Kullanıcı Türleri

Bir veritabanının kullanıcılarını belirtirken, onları 3 genel sınıf halinde tanımlayabiliriz:

**Uygulama programcısı :** COBOL, C++ gibi bazı yüksek seviyeli dillerde programlar yazmakla sorumludur.

**Son-kullanıcı :** Sorgu dili ile veritabanına erişen kişidir.

**Veritabanı yöneticisi (DBA) :** Veritabanı üzerindeki tüm işlemleri kontrol eder.

**8.** Örnek bir veritabanı için kullanıcılar belirleyerek yetkilendirmelerini şematik olarak gösteriniz.

**9.** Veritabanı ile VTYS’ nin farkları nelerdir?

**Veri tabanları** birbirleriyle ilişkili bilgilerin depolandığı alanlardır. Bilgi artışıyla birlikte bilgisayarda bilgi depolama ve bilgiye erişim konularında yeni yöntemlere ihtiyaç duyulmuştur. Veri tabanları; büyük miktardaki bilgileri depolamada geleneksel yöntem olan ‘’dosya-işlem sistemine’’ alternatif olarak geliştirilmiştir.

**Veri tabanı yönetim sistemi** (VTYS, İngilizce: Database Management System, kısaca DBMS), veri tabanlarını tanımlamak, yaratmak, kullanmak, değiştirmek ve veri tabanı sistemleri ile ilgili her türlü işletimsel gereksinimleri karşılamak için tasarlanmış sistem ve yazılımdır.

**10.** İlişkisel veri modelini açıklayınız.

En geniş kullanım alanı olan modeldir. Bu modeli temel alan bazı VTYS'ler şunlardır:

Oracle, Sybase, DB2, SQL Server, Access v.s. İlişkisel modele alternatif model olarak, nesneye-yönelik model ise yaygınlaşmaya devam etmektedir. Nesneye-yönelik model için bazı örnek sistemler şunlardır: Object Store, Ontos, Versant, O2, Itasca. İlişkisel model ile nesneye-yönelik modelin bazı kavramlarının kaynaşmasından ortaya çıkan nesne-ilişkisel modelini kullanan ürünler ise son yıllarda birçok endütriyel VT şirketince piyasaya sürülmüştür. Bunlara örnek sistemler: UniSQL, Informix Universal Server, Oracle , DB2, SQL-Server, Sybase vs.

**11.** Bir öğrenci bilgi sistemi kullanılacak veritabanı, tablolar ve tablolar arasındaki ilişkiyi belirleyiniz.

create database ogrenciIsleri

use ogrenciIsleri

create table bolum(

bolumId int primary key,

bolumAdi varchar(30),

bolumAciklama text,

bolumEposta varchar(30),

)

create table ogrenci(

OgrenciNo varchar(11),

OgrenciAdi varchar(30),

OgrenciSoyadi varchar(30),

OgrenciDogumYeri varchar(30),

bolumId int CONSTRAINT fk\_bolumId foreign key references bolum(bolumId),

)

alter table ogretmen alter column OgretmenId int not null

alter table ogretmen add constraint OgretmenId primary key (OgretmenId ASC)

alter table ogrenci alter column OgrenciNo int not null

alter table ogrenci add constraint OgrenciNo primary key (OgrenciNo ASC)

create table ders(

DersId varchar(11) primary Key,

DersAdi varchar(30),

DersKodu varchar(10),

DersKredi varchar(2),

bolumId int CONSTRAINT fk\_ders\_bolumId foreign key references bolum(bolumId),

)

create table Ogrenci\_Ders(

OgrenciNo int constraint FK\_Ogrenci\_Ders\_OgrenciNo foreign key references ogrenci(OgrenciNo),

DersId varchar(11) constraint FK\_Ogrenci\_Ders\_DersId foreign key references ders(DersId),

puan varchar(5)

)

create table Ogretmen\_Ders(

OgretmenId int constraint FK\_Ogretmen\_Ders\_OgretmenId foreign key references ogretmen(OgretmenId),

DersId varchar(11) constraint FK\_Ogretmen\_Ders\_DersId foreign key references ders(DersId),

)

