***1.12 Çalışma Soruları***

***1. Geleneksel dosyalama sistemlerinin çalışma şeklini açıklayarak bildiğiniz bir programlama dili yardımıyla txt dosyadan veri okumak ve veri yazmak için bir uygulama geliştiriniz.***

Dosyalar (kütükler yada files) aynı yapıya sahip ve birbirleri ile ilişki içerisinde olan "kayıt"lar (records) topluluğudur. Kayıt ise, yapılan işlemler sırasında ana belleğe yazılan veya ana bellekten okunan veri birimidir. Bir kayıt içindeki ayrı ayrı veri parçalarına da "alan" (field) adı verilir.

   Örneğin bir öğrenci bilgileri dosyasını ele alırsak, öğrencinin adı, soyadı, numarası, aldığı dersler ve bu derslerden aldığı notlar ayrı ayrı birer alandır. Bir öğrenciye ait bütün bu alanlar bir araya gelerek öğrencinin kaydını oluşturmakta ve tüm öğrencilere ait kayıtlar da bir araya gelerek öğrenci bilgileri dosyasını oluşturmaktadır.

   Dosya yapısı için bir başka örnek de nüfus kütükleridir. Nüfus kütükleri; ad, soyad, ana adı, baba adı, doğum tarihi, doğum yeri gibi alanlar içeren kayıtlardan oluşurlar.

   Dosyalar çok büyük miktarlarda veri içerdiklerinden yan belleklerde tutulurlar ve kullanılış amaçlarına göre de değişik şekillerde düzenlenirler. Düzenleniş şekline ve kaydedildiği yan belleğe (disk yada teyp gibi) bağlı olarak bir kütüğe genel olarak iki değişik şekilde erişilir.

**Sıralı Erişimli Dosya Sistemleri**

    Bu tür erişimde bilgilere daha önce belirlenen bir sıra izlenerek birbiri ardı sıra ulaşılır. Sıralı erişim yönteminin en büyük dezavantajı, herhangi bir bilgiye doğrudan ulaşmanın mümkün olmamasıdır. Örneğin 500 kayıt bulunan bir dosyada 453. sırada yer alan bir kayda ulaşmak için bu kayıttan önce yer alan, 452 adet kaydın okunarak atlanması zorunludur. Bu nedenle sıralı erişim yöntemi genellikle aynı yapıdaki işlerin birleştirilerek kümeler halinde yapıldığı (batch processing) uygulamalarda kullanılır.

    Yukarıda örnek olarak sözünü ettiğim öğrenci bilgileri dosyası karne hazırlama uygulamasında sıralı erişim yöntemi kullanılarak oluşturulabilir. Çünkü bu uygulamada öğrenci kayıtlarının öğrenci okul numaralarına göre sıralı olarak bir yan bellekte tutulmaları ve bu kayıtların yan bellekten sırayla okunarak karnelerin hazırlanması sıralı erişim yönteminin verimli olarak kullanılabileceği bir uygulamadır.

**Doğrudan Erişimli Dosya Sistemleri**

    Bu erişim yönteminde, ulaşılmak istenen herhangi bir bilgiye, diğer kayıtlardan bağımsız olarak doğrudan ulaşmak mümkündür. Ancak, bunun için her kaydı diğer kayıtlardan ayıran özel bir tanımlayıcıya gerek duyulur. Bu özel tanımlayıcı "anahtar" (key) olarak adlandırılır.       Böylelikle anahtar bilindiği zaman, diğer kayıtlar incelenmeksizin ve kayıtların kaydediliş sırasıyla ilgilenilmeksizin istenilen kayda doğrudan ulaşmak mümkündür. Yine, öğrenci bilgileri dosyasını ele alacak olursak örneğin öğrenci belgesi uygulamasında 500 öğrenciden 453 numaralı olana ait bilgileri içeren belgeye ulaşmak ve bunu yazıcıdan almak gerektiğinde anahtar alan olarak numarayı kullanmakla doğrudan erişim yöntemi en etkili olan yöntemdir. Çünkü sıralı erişim yönteminde 452 adet öğrenciye ait bilgiler incelendikten sonra 453. öğrencinin bilgilerine ulaşılması zaman almakta ve gereksiz yere 452 adet kayıt incelenmiş olmaktadır.

    Sonuç olarak şunu söyleyebiliriz ki bir dosya içerisindeki tüm bilgilerin değerlendirilip elden geçirilmesi gerektiğinde sıralı erişim yöntemi, pek çok kayıt içerisinde sadece bir kayda ait bilgilerin işlenmesi gerektiği durumlarda ise doğrudan erişim yöntemi etkin olarak kullanılabilmektedir.

    Bir dosyaya erişim yöntemlerini belirleyen etmenlerden biri de dosyanın düzenleniş şeklidir. Erişim yöntemleri ile de birlikte değerlendirilecek şekilde dosyalar düzenlenişlerine göre: sıralı (sequential) dosyalar ve dizinli (indexed) dosyalar olmak üzere ikiye ayrılırlar. Sıralı dosyalar oluşturulurken, veriler birbiri ardı sıra girilir ve işlenirken de yine ancak bu şekilde işlenirler (sıralı erişim). En basit dosya yapısı olmakla beraber oldukça sık kullanılırlar.

Dizinli dosyalar ise, yapı olarak sıralı dosyalardan farklıdır. Dosyanın, kayıtların anahtarlarına göre yan belleğin hangi adresinde bulunduğunu gösteren bir dizini (index) vardır. Bu dizin, yapı olarak bazı kitapların sonlarında yer alan index ile aynıdır. Kullanılış amacı da kitaplardaki indexler gibidir. Anahtarı bilinen bir kaydın bulunduğu yan bellek adresi bu indexten öğrenilerek kayda doğrudan erişilir.

    Dizinli kütüklere doğrudan erişmek mümkün olduğu gibi, dizinde kullanılan anahtarlara göre sıralı erişmek de mümkündür. Örneğimizdeki öğrenci kütüğünü dizinli bir kütük olarak yaratırsak, karne hazırlamak ve basmak istediğimizde sıralı erişim yöntemini, bir öğrenciye ait bilgilere ulaşmak ve örneğin bir öğrenci belgesi basmak istediğimizde ise doğrudan erişim yöntemini kullanabiliriz.

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

void main(void)

{

setlocale(LC\_ALL, "Turkish");

FILE \*fp;

int id1;

if ((fp = fopen("deneme.txt", "w")) == NULL) {

printf("Dosya açılamadı!\n");

exit(1);

}

printf("Bir int değer giriniz: ");

scanf\_s("%d", &id1);

if (fwrite(&id1, sizeof(int), 1, fp) != 1) {

printf("Yazma hatası!\n");

exit(1);

}

fclose(fp);

if ((fp = fopen("deneme.txt", "r")) == NULL) {

printf("Dosya açılamadı!\n");

exit(1);

}

if (fread(&id1, sizeof(int), 1, fp) != 1) {

printf("Okuma hatası!\n");

exit(1);

}

printf("Dosyadan okunan değişken değeri: %d", id1);

fclose(fp);

\_getch();

}

***2. Geleneksel dosyalama yöntemleriyle veritabanı yönetim sistemlerinin benzerlik ve farklılıklarını açıklayınız.***

Geleneksel dosyalama sistemleri veritabanı yönetim sistemleri öncesinde veri depolamak için kullanılan sistemlerdir.

Zamanla geleneksel dosyalama sistemleri artan veri kapasitesi, veri işleme hızı ve kullanım alanı gibi birçok ihtiyaca cevap veremez duruma gelmiştir. Dolayısıyla artık yerini veritabanı yönetim sistemleri yazılımlarına bırakmıştır.

Geleneksel dosyalama sistemine göre verilere erişim hızlı.

–Gereksiz veri tekrarını engeller.

–Her ihtiyaca, büyük veya küçük bütün otomasyon sistemleri günümüz  artık veri tabanı kullanmaktadır.

***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. Veritabanının görevlerini açıklayarak kullanıldığı alanlara örnek veriniz.***

En basit anlamda veritabanı, belirli bir amaca yönelik düzen verilmiş kayıt ve dosyaların tümüdür.

Veritabanı insanlar tarafından sadece bilgisayarlarda kullanıldığı düşünülmektedir. Fakat bu büyük bir yanılgıdır. Hayatımızın büyük bir alanında veritabanı kullanılmaktadır. Ancak en sık kullanıldığı yerler bilgisayarlardır. Veritabanı yazılımını diğer yazılımlardan ayıran en önemli özellik ise, aranan bir bilgiye hızlıca erişimi sağlamasıdır.

Veritabanı yazılımları, özellikle hastanelerde sıkça kullanılır. Çünkü hastenelerde gelen hasta kayıtlarının veritabanı yazılımı olmadan tutulması neredeyse imkansızdır. Ayrıca, birçok dev firma, stok durumu, üretim durumu gibi endüstriyel verilerini, veritabanı yazılımları ile kontrol ederler. Bu örnekler gibi birçok iş alanında veritabanı yazılımları kullanılır.

Havayolları: Rezervasyonlar ve Bilet Satışı  
Bankalar: Bankacılık işlem ve hareketleri  
Üniversiteler: Öğrenci Takip sistemi ve Not girişleri  
Satış: Müşteri ve Ürün bilgileri  
Çevrimiçi Perakendiciliği: Sipariş Kayıtları ve sonuçları  
İnsan Kaynakları: Çalışan Profilleri ve Maaşları  
Üretim: Stoklar ve üretilen ürünlerin takibi

***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 very türüne verilen isimdir, tablo içerisinde birden fazla kullanılır. Satır sütuna ait very gurubudur. Örneğin bir öğrencinin no, ad, soyad bilgilerinin tamamı satırı ifade etmektedir.

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

**Birincil Anahtar (Primary Key)**

Birincil anahtar amacı benzersiz bir tabloda her bir kayıtı ayrı ayrı belirlemektir. Genel olarak, birincil anahtar tek bir sütun üzerinde tanımlıdır ve iki sütun üzerinde tanımlı olması mümkün değildir.

Üç temel kural birincil anahtarları tanımlayan özelliktir:

\*Her kayıtın birincil anahtarında bir değer olması gerekir. O boş olamaz.  
\*Birincil anahtar değerleri benzersiz olmalıdır.  
\*Birincil anahtar değerleri yenilenen olmamalıdır. Eğer yenileme olursa bir kayıt silindiğinde bu hata ve karışıklık yaratabilir, bu yüzden birincil anahtar başka bir kayıt yeniden atanınca yinelenen olmamalıdır.

**Yabancı (İkincil) Anahtar (Forein Key)**

Bir yabancı anahtar bir alanı (veya hedef olarak başvurulan bir tablodan) başka bir tabloda belirli bir anahtar yerine birincil anahtarı gösteren kolonlardır. Bu ilgili tabloları mantıklı bilgilerle ilişkilendiren bir yoludur.

Örneğin, ürün tablosunda  başvuran birincil anahtar alanı üreticiler tablosunda bir yabancı anahtar içerir. Bu şekilde, her ürün – dış kilit noktaları ilişkili bir üreticisi var olur. Unutmayın ki yabancı anahtar genellikle benzersiz değidir ancak başvurulan alanı başvurulan tabloda (birincil anahtar) gösterir.

***7. Veritabanı kullanıcı türleri nelerdir?***

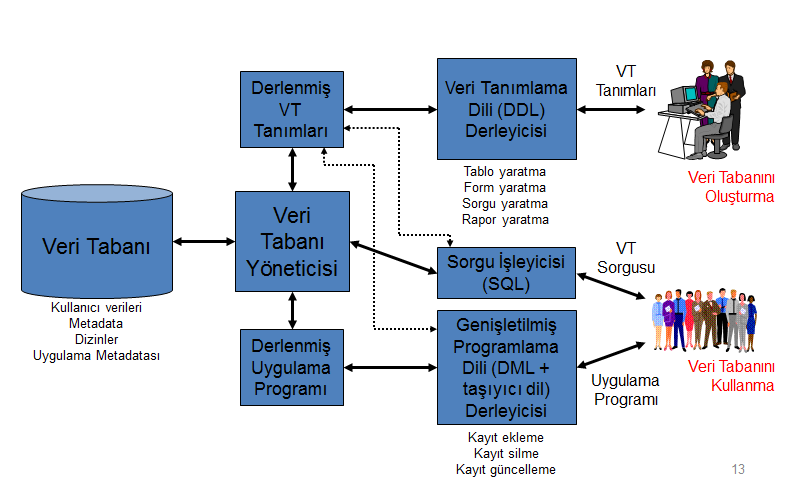
**Veritabanı Yöneticisi (DBA-Database Administrator)**Tasarım,oluşturma ve işletiminden sorumludur. Görevleri;  
– Tasarımı  
– Performans Analizi  
– Erişim Yetkilerini Düzenleme ve Erişim Sağlama  
– Yedekleme ve Geri Yükleme  
– Veri Bütünlüğü Sağlama  
– Sistem Sürekliliği Sağlama

**Uygulama Programcısı**  
Son kullanıcılara yönelik uygulama yazılım geliştirmek. Veri işleme dili ve  
geliştirme.

**Sorgu Dili Kullanıcıları**  
Mevcut sorgular dışındaki diğer sorguları hazırlamak. Veri eklemek,silmek.

**Son Kullanıcılar**  
Yazılımı kullanan

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



***9. Veritabanı ile VTYS’nin farkını açıklayınız.***

Veritabanı birbirleriyle ilişkili olan verilerin birlikte tutulduğu, depolandığı alanlardır. Veri Tabanı Yönetim Sistemleriyse veritabanları oluşturmak, değiştirmek, veri işlemek ve veri çıkarmak için dizayn edilmiş yazılımlardır. Bir veri tabanı yönetim sistemi sayesinde birden fazla veritabanı control edilebilir ve farklı yazılımların sistemle iletişimi sağlanabilir. Yadi VTYS veritabanını kullandığımız onu kontrol eden bir yazılımdır.

***10. İlişkisel veri modelini açıklayınız.***

İlişkisel veri modelinde veriler tablolar içinde saklanır. İlişkiler kurulurken birincil anahtar (Primary key) ve yabancı anahtarlar (Foreign Key) kullanılır. Anahtar alanlar sayesinde indeksleme yapma olanağı sunan ilişkisel veri tabanlarında erişim ve işlemler daha hızlı yapılabilir.

***11. Bir öğrenci bilgi sistemi için kullanılacak veritabanı, tablolar ve tablolar arasındaki ilişkileri 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),

)

