



İSTANBUL OKAN ÜNİVERSİTESİ

MESLEK YÜKSEKOKULU

BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ

BİLGİSAYAR PROGRAMCILIĞI PROGRAMI

VERİ TABANI TASARIMI KAVRAMLARI

GEZİ YÖNETİMİ SİSTEMİ

HAZIRLAYAN

Selçuk Yıldız

19MY03030

DANIŞMAN

ÖĞR.Emrah Sarıçiçek

2020 Kasım

İÇİNDEKİLER

İçindekiler

PROJEM	3
VERİTABANI	
Veri Tabanı Yönetim Sistemleri	4
Veri Tabanı Yönetim Sistemleri Avantajları	4
Veri Tabanı Yönetim Sistemleri Dezavantajları	4
GELİŞME	4
ER DİYAGRAMI	4
NORMALİZASYON	5
TRANSACT-SQL.....	5
DOSYA VE DOSYA.....	5
TABLOLAR.....	6
Tablo Oluşturma	7
Şema	8
VIEW	9
View Kullanımı ve Örneği	9
STORE PROCEDURE	10
Store Procedure Kullanımı ve Örneği	10
Çalışma Süresi	11
SONUÇ	11
KAYNAKÇA	11

GİRİŞ

PROJEM

Projemin amacı bir gezi sitesi için gerekli olacak her şeyi hazırlamak. Örneğin tatil yapacak bir kişinin ihtiyaçlarını her türlü karşılayabilmesi için gereken her şeyi sağlamak. Hiçbir verinin birbirine karışmaması ve düzenli olması çok büyük kolaylık sağlıyor. Size projem de kullandığım her şeyi anlatacağım.

VERİ TABANI

Veritabanı genellikle elektronik olarak bir bilgisayar sisteminde depolanan yapılandırılmış bilgi veya verilerden oluşan düzenli bir koleksiyondur. Veritabanı genellikle bir veritabanı yönetim sistemi (DBMS) ile kontrol edilir. Veri ve DBMS ve aynı zamanda bunlarla ilişkili uygulama yazılımları bir araya getirildiğinde sıklıkla yalnızca veritabanı olarak kısaltılan veritabanı sistemi olarak ifade edilir.[1]

Microsoft SQL Server'da veri tabanı oluşturmak için iki yöntem vardır.

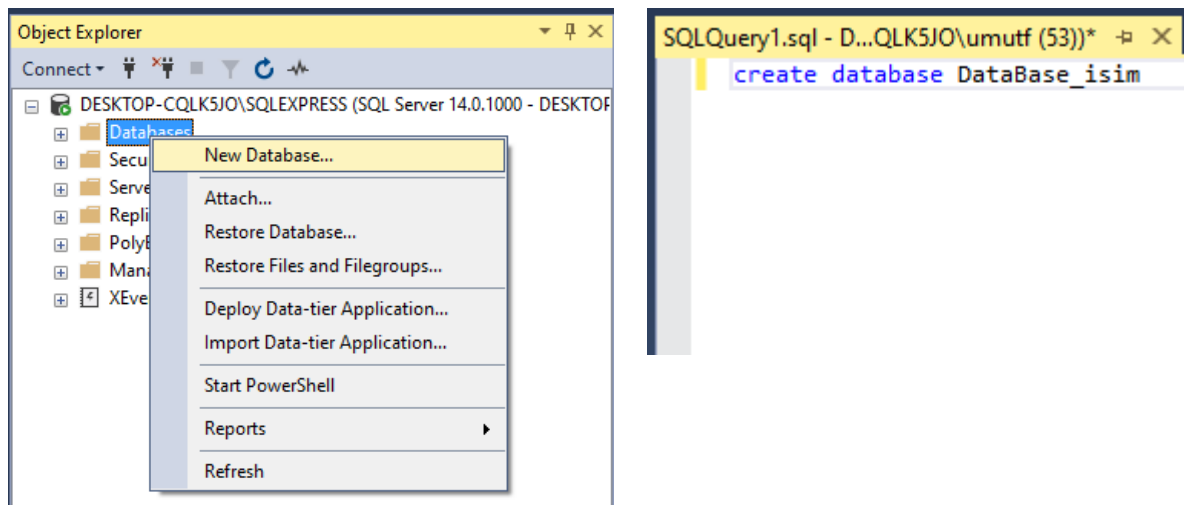
Birinci yol ;

İkinci yol;

Programdan new database

Programda kod ile oluşturabiliriz.

diyerek oluşturabiliriz.



Veri Tabanı Yönetim Sistemleri

Veri tabanı yönetim sistemi(DBMS), veri tabanlarını oluşturma ve yönetme sistem yazılımıdır. Veri tabanı yönetim sistemi kullanıcılara ve programcılara sistematik bir yolla veri oluşturmayı, geri almayı, güncellemeyi ve yönetmeyi sağlar.[2]

Veri Tabanı Yönetim Sistemleri Avanjları

Gereksiz Veri Tekrarının Olmaması. Her veri planlı tekrarlar dışında bir kez yer alır. tüm uygulamalar aynı veriyi kullanır. güncellemelerde bütünlük ve tutarlık sorunları ortadan kalkar.[3]

Veri Tabanı Yönetim Sistemleri Dezavantajları

Veri tabanı sisteminin kurulumu ve bakımı klasik dosya sistemine göre daha maliyetli olabilir.[4]

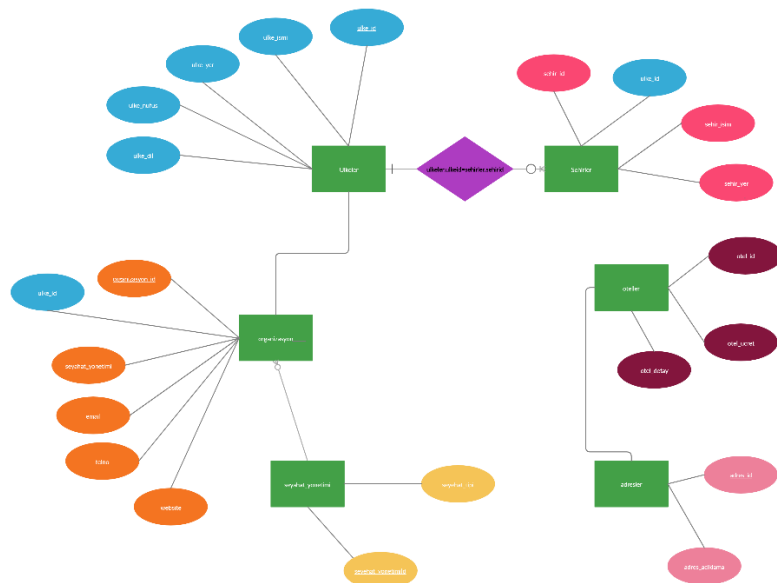
Veri Erişimini İyileştirmek

Veritabanında veriye erişmenin birçok yolu vardır. Bu yollardan en hızlı, en Pratik ve en güvenli olanını tercih etmek **veri erişimini iyileştirmek** olarak ele alınır.

GELİŞME

ER DİYAGRAMI

ER Diyagramları gerçek hayattaki nesneler ve bu nesneler arasındaki bağlantıları, ilişkileri tanımlayan diyagram modelleme tekniğidir.



NORMALİZASYON

Normalizasyonun iki temel amacı vardır. Veri tabanında veri tekrarlarını ortadan kaldırmak ve veri tutarlılığını (doğruluğunu) artırmak.

	otel_id	otel_ad	otel_ucret	otel_detay
1	1	RAMADA	2323	3+2
2	2	Double Three	3323	3+3
3	3	Hilton	5458	6+6
4	4	UMMON	4789	3+2
5	5	SİVİS	7777	6+5

TRANSACT-SQL

SQL, düzeltilmesi veya değiştirilmesi istenen bilgileri açıkça belirtmeye izin veren ve yerine getirilebilecek başlıca işlemleri tanımlamamızı sağlayan bir komut takımıdır. Bu komutların oluşturduğu yapıya T-SQL dili denir. T-SQL ile veri ve sorgulara erişebilir, güncelleyebilir ve ilişkisel veri tabanı sistemi yönetilebilir. T-SQL komutları kullanım amaçlarına göre üç genel kategoriye ayrılır.[5]

DOSYA VE DOSYA GRUPLARI

1.Birincil Veri Dosyaları: .mdf uzantılı bir dosyadır. Her veritabanında 1 adet bulunmak zorundadır. Veritabanının en başta olan nesne ve verileri ile birlikte başka bir dosya kullanmıyorsa tamamını bu dosyada saklar.

2.İkincil Veri Dosyaları: .ndf uzantılıdır. İkincil dosya zorunlu değildir. Ancak tanımlanırsa, dosya grupları yardımı ile veritabanı nesnelerinin farklı disklere dağılımını sağlamak gibi fiziksel seviye veri dağılımını düzenlenme amacıyla kullanılabilir.

3.Transaction Log Dosyaları: .ldf uzantılıdır. Transaction log dosyasına, veritabanının da yapılan işlemler saklanır ve hata kurtarma amaçlı işlemler yapılırken kullanılır. Bir veritabanında en az 1 adet bulunulmalıdır.

Dosya ve Dosya Grupları Oluşturma

```
SQLQuery1.sql - D...QLK5JO\umutf (53))* -p X
ALTER DATABASE databaseisim
ADD FILEGROUP dosyagroup
```

Bu şekilde dosya grup oluşturulur.

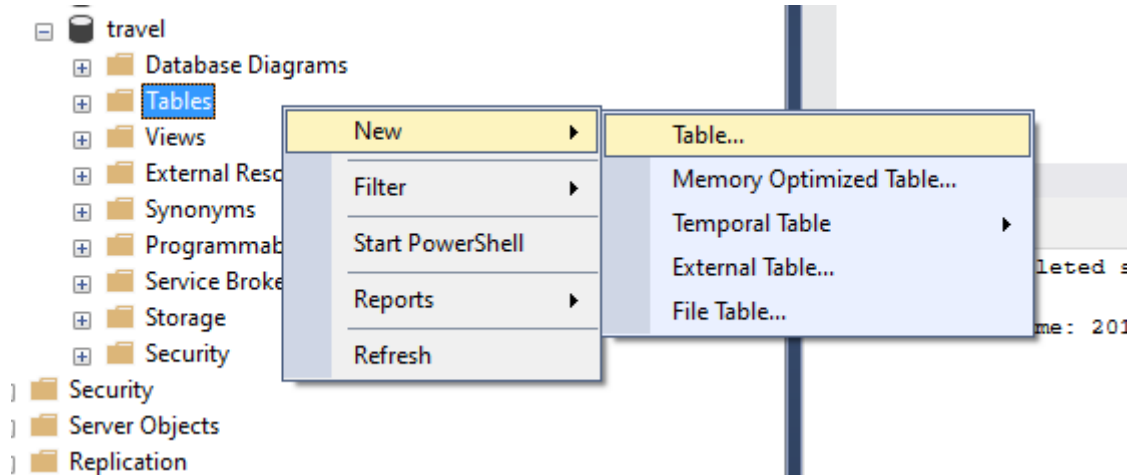
```
SQLQuery1.sql - D...QLK5JO\umutf (53))* -p X
CREATE DATABASE veritabanı_ismi
ON PRIMARY
(NAME = okul_Data,
FILENAME = 'D:\data\dosyaismi.mdf',
SIZE = 2MB,
MAXSIZE = 10MB,
FILEGROWTH = 10%)
LOG ON
(NAME = okul_Log,
FILENAME = 'dosya yolu',
SIZE = 1MB,
MAXSIZE = 5MB,
FILEGROWTH = 10%)
```

Bu şekilde dosya grubuna dosya eklenir.

TABLolar

Microsoft Sql Server'da tablo oluşturma'nın iki yolu vardır. Birinci yolu;

Database altında oluşturulmuş olduğumuz travel veri tabanımızın hemen altında bulunan tables klasörü üzerindeyken sağ tuş yaparak listeden New Table Komutuna tıklıyoruz.



Daha sonra sütunlara girmek istediğiniz başlıkları yazıyoruz ve özelliklerini tanımlıyoruz. İstediğiniz column'u primary key yapıyorsunuz.

	Column Name	Data Type	Allow Nulls
PK	otel_id	int	<input type="checkbox"/>
	otel_ad	varchar(50)	<input checked="" type="checkbox"/>
	otel_ucret	float	<input checked="" type="checkbox"/>
	otel_detay	varchar(20)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Tabloyu oluşturduktan sonra içine bilgi girmek için tablomuzun üstüne tıklayıp Edit Top 200 Rows 'a tıklayarak içine bilgileri giriş yapıyoruz.

	otel_id	otel_ad	otel_ucret	otel_detay
▶	1	RAMADA	2323	3+2
	2	Double Three	3323	3+3
	3	Hilton	5458	6+6
	4	UMMON	4789	3+2
	5	SİVİS	7777	6+5
*	NULL	NULL	NULL	NULL

Adresler Tablom :

	adres_id	adres_aciklama
▶	1	FATİH S N 41., S...
	2	MURADİYE S N ...
	3	OSMAN KAVU...
*	NULL	NULL

Kullanici Tablom :

	kullanici_id	kullanici_ad	kullanici_email	kullanici_isim...	kullanici_tel	kullanici_tc	kullanici_sifre	kullanici_yetki	kullanici_sehir	kullanici_ulke	kullanici_posta...	kullanici_puan
▶	1	umcan	umcan@hotma...	umut furkan can	555555	123456789	1234	True	ankara	türkiye	6	8000
	2	ozi	ozi@hotmail.co...	oğuzhan baykan	666666	987654321	3214	False	new york	amerika	7	5000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Organizasyon Tablom :

	organizasyon_id	ulke_id	seyahat_yonte...	email	telno	website
▶	1	1	3	umutcan@hot...	5354566966	um.com
	2	2	4	ozi@hotmail.co...	555555	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Oteller Tablom :

	otel_id	otel_ad	otel_ucret	otel_detay
▶	1	RAMADA	2323	3+2
	2	Double Three	3323	3+3
	3	Hilton	5458	6+6
	4	UMMON	4789	3+2
	5	SİVİS	7777	6+5
*	NULL	NULL	NULL	NULL

Şehirler Tablom :

	sehir_id	ulke_id	sehir_isim	sehir_yer
▶	1	1	Amerika	New york
	2	2	Türkiye	İstanbul
	3	3	Almanya	Köln
	4	4	İtalya	Paris
*	NULL	NULL	NULL	NULL

Seyahat Tablom :

	seyahat_yonet...	seyahat_tipi
▶	1	3
	2	2
	3	1
*	NULL	NULL

Ülkeler Tablom :

	ulke_id	ulke_ismi	ulke_yer	ulke_nufus	ulke_dil
▶	1	Amerika	Kuzey Amerika	300000	ENG
	2	Türkiye	Asya	200000	TR
	3	Almanya	AVRUPA	500000	DE
	4	İtalya	AVRUPA	500000	IT
*	NULL	NULL	NULL	NULL	NULL

ŞEMA

Schema databasedeki tablo, view ve stored procedure gibi objeleri gruplamaya yarar. Bir objeyi bir şema altında yaratabilirsiniz veya bir şemanın altına transfer edebilirsiniz. Ayrıca şema bazlı yetki de verebilirsiniz. Yani user sadece izin verdiginiz şemanın altındaki objeleri görebilir diğerleri göremez. Normalde obje sayılarının fazlalığı yönetimi ve izinlerini oldukça zorlaştıracaktır. Her bir objeyi tek tek yetkilendirmek yerine şema bazında yetkilendirme yapmak daha uygun olacaktır.


```
CREATE SCHEMA şema_ismi [ AUTHORIZATION kullanıcı_ismi ] [ şema_ögesi [ ... ] ]
```

VIEW

Sql view öğeleri veritabanının altında tutulur. Sql view nedir ve niçin kullanılır sorusunun yanıtı ise; bir yahut birden fazla tabloyu join ile birleştirip, where ile koşul belirttikten sonra içerisinde group by deyiminin kullanılabildiği sanal tabloya view denir. view in amaçlarından bir tanesi ise birden fazla tablodan çekilen verilerin tek tablo içerisinde toplanması ve bu sanal tablo içerisinde gerekli filtreler verilerek sorgu ile raporlama yapılabilmesidir, bunlar sql view avantajları arasında sayılabilir.[6]

```
CREATE VIEW viewAd
AS
SELECT sütunAdları
FROM baseTabloAdi
```

```
select *
from view1
```

View örnekleri :

```
create view vwDil
AS
select ulke_ismi, ulke_dil from ulkeler
INNER JOIN sehirler on ulkeler.ulke_id=sehirler.ulke_id
```

1)

	ulke_ismi	ulke_dil
1	Amerika	ENG
2	Türkiye	TR
3	Almanya	DE
4	İtalya	IT

```
create view vwulke as select ulke_ismi, ulke_nufus, ulke_yer
from ulkeler where ulke_yer='AVRUPA' with check option
select * from vwulke
```

2)

	ulke_ismi	ulke_nufus	ulke_yer
1	Almanya	500000	AVRUPA
2	İtalya	500000	AVRUPA

3)

```
create view vwOtel_ucret AS SELECT otel_ad,otel_ucret
from oteller where otel_ucret>4000

select * from vwOtel_ucret
```

	otel_ad	otel_ucret
1	Hilton	5458
2	UMMON	4789
3	SİVİS	7777

STORE PROCEDURE

Yapısal Yordamlar anlamına gelmektedir. Bu yazıda ben kendilerini SP olarak isimlendireceğim. SP belirli bir işlevi, görevi yerine getirmek için özellikle yapılandırılmış bir veya daha fazla tablo, sp vs. ile ilişki kod parçacıklarıdır. Kısaca Derlenmiş SQL cümlecği diyebiliriz.[7]

Store Procedure örnekleri :

1)

```
create proc ulkeler_proc
as
select ulke_ismi,sehir_isim
from ulkeler U inner join sehirler S
on U.ulke_ismi=S.sehir_isim
go

exec ulkeler_proc
```

	ulke_ismi	sehir_isim
1	Amerika	Amerika
2	Türkiye	Türkiye
3	Almanya	Almanya
4	İtalya	İtalya

2)

	organizasyon_id	ulke_id	seyahat_yontemi	email	telno	website
1	1	1	3	umutcan@hotmail.com	5354566966	um.com
2	2	2	4	ozi@hotmail.com	555555	NULL

```
create proc organizasyon_proc
as
select * from organizasyon
```

```
exec organizasyon_proc
```

ÇALIŞMA SÜRELERİ

Sorgu Hazırlama : 8 saat

Rapor hazırlama :10 saat

Sunum Hazırlama : 3 saat

SONUÇ

Projemde öğrenebildiğim bütün her şeyi kullanmaya çalıştım.

Stabil olarak çalışmasını sağladım.

KAYNAKÇA

[1] <https://www.oracle.com/tr/database/what-is-database.html>

[2] <https://webmaster.kitchen/veritabani-yonetim-sistemi-nedir/>

[3] <https://teknokoliker.com/2012/05/veri-taban-nedir.html>

[4] <https://kamer2.wordpress.com/veritabani-yonetim-sistemlerivtys/>

[5] <http://www.bilgisayarmuhendisleri.com/sayfa.aspx?s=84>

[6] <https://www.furkanpezek.com.tr/2018/04/sql-view-kullanimi/>

[7] <https://medium.com/gokhanyavas/stored-procedure-nedir-76bf84b9a676>