

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**İTÜ KAMPÜSÜ AFET BİLGİ SİSTEMİ, AFET YÖNETİMİ OBJE MODELİNİN
OLUŞTURULMASI VE İLİŞKİLİ SORGULAMALARIN YAPILANDIRILMASI**

YÜKSEK LİSANS TEZİ

Müh. Himmet KARAMAN

(501011720)

Anabilim Dalı: JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ

Programı: JEODEZİ VE FOTOGRAMETRİ MÜHENDİSLİĞİ

MAYIS 2003

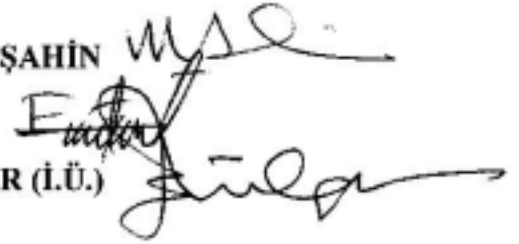
İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**İTÜ KAMPÜSÜ AFET BİLGİ SİSTEMİ, AFET YÖNETİMİ OBJE MODELİNİN
OLUŞTURULMASI VE İLİŞKİLİ SORGULAMALARIN YAPILANDIRILMASI**

**YÜKSEK LİSANS TEZİ
Müh. Himmet KARAMAN
(501011720)**

**Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2003
Tezin Savunulduğu Tarih : 26 Mayıs 2003**

**Tez Danışmanı : Prof.Dr. Muhammed ŞAHİN
Diğer Jüri Üyeleri Doç.Dr. Ergin TARI
Doç.Dr. Enver ÜLGER (İ.Ü.)**



MAYIS 2003

ÖNSÖZ

Bu yüksek lisans tez çalışması boyunca bana vermiş olduğu tavsiyelerinden, desteğinden ve ilgisinden dolayı danışmanım Prof. Dr. Muhammed Şahin'e ve çalışmalarımda bana yol gösteren Doç. Dr. Ergin Tarı ve Doç. Dr. Cengizhan İpbüker'e en içten teşekkürlerimi sunmayı bir borç olarak görüyorum. Çalışmam sırasında benden yardımlarını esirgemeyen Araştırma Görevlisi arkadaşlarım, Hakan Yavaşoğlu, Turan Erden ve Serdar Bilgi'ye özel olarak teşekkür ederim.

Ayrıca, gerek evde gerekse okuldaki çalışmalarımda bana vermiş olduğu tavsiyelerinden ve yol göstermesinden dolayı kuzenim Araştırma Görevlisi Dr. Bahadır Ergün'e ve aileme minnet borçluyum.

Mayıs 2003

Himmet KARAMAN

İÇİNDEKİLER:

KISALTMALAR	vi
TABLO LİSTESİ	vii
ŞEKİL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. Bölüm: GİRİŞ	1
1.1. Giriş ve Çalışmanın Amacı	1
2. Bölüm: TABİS	4
2.1. TABİS'in Amacı	4
2.2. TABİS Obje Kataloğu - TABİS-OK	5
2.2.1. TABİS-OK'nda Obje Kavramı	8
2.2.1.1.Obje Grubu	10
2.2.1.2.Obje Türü	10
2.2.1.3.Parça Obje Kavramı	11
2.2.1.4.Öznitelik Verileri	11
2.2.1.5.Öznitelikler İçin Veri Tipleri	12
2.2.1.6.Öznitelikler İçin Veri Kardinallik Derecesi	13
2.2.2. Afet Yönetimi Obje Alanları Kataloğu (TABİS-AOK)	13
3. Bölüm: VERİTABANI	15
3.1. Veritabanının Tanımı ve Amacı	15
3.2. Veritabanı Türleri	16
3.2.1. İlişkisel (Relational) Veritabanı Modeli	16
3.2.2. Veritabanı Yapısı	18
3.2.2.1.Tablolar (Tables)	19
3.2.2.2.Tetikler (Triggers)	19

3.2.2.3.Saklı Prosedürler (Stored Procedures)	20
3.2.2.4.Görünümler (Views)	20
3.2.3. Normalizasyon Kuralı	20
3.2.4. Veri Bütünlüğü	21
3.2.5. Mantıksal Veriler Arasındaki İlişkiler	21
3.3. SQL	21
3.4. Veritabanı Yazılımları	22
3.4.1. Microsoft Access	23
3.4.2. MySQL	23
3.4.3. IBM DB2	24
3.4.4. Informix	24
3.4.5. Microsoft SQL Server	24
3.4.6. PostgreSQL	24
3.4.7. Oracle	25
3.4.8. Interbase	25
3.4.9. Progress	25
4. Bölüm: POSTGRESQL VERİTABANI	26
4.1. PostgreSQL Nedir	26
4.2. PostgreSQL'in Tarihi	27
4.3. Teknik Açından PostgreSQL	28
4.3.1. Desteklenen Platformlar	28
4.3.2. PostgreSQL Mimarisi	29
4.3.3. PostgreSQL'in Sınırlamaları	30
4.4. PostgreSQL Veri Tipleri	31
4.5. PostgreSQL Kurulumu	34
4.5.1. PostgreSQL'i Başlatma	36
4.6. PostgreSQL Araçları	37
4.6.1. psql	37
4.6.2. PgAdmin II	38
4.6.3. Pgaccess for Windows	39
4.6.4. pgaccess	40
5. Bölüm: UYGULAMA	41
5.1. Afet Yönetimi Obje Modelinin Hazırlanması	41
5.1.1. TABİS-OK Afet Yönetimi Obje Alanlarından Çıkartılan	

Elemanlar	41
5.1.2. TABİS-OK Afet Yönetimi Obje Alanlarına Eklenen Elemanlar	46
5.2. Veritabanı Tasarımı	47
5.3. Linux Üzerinde PostgreSQL ile Veritabanı Oluşturulması	50
5.3.1. Tablo Yaratma	53
5.3.1.1.psql ile Tablo Yaratma	53
5.3.1.2.PgAdmin II ile Tablo Yaratma	54
5.3.1.3.Pgaccess ile Tablo Yaratma	54
5.3.2. Oluşturulan Tablolara Veri Girişi	55
5.3.3. Oluşturulan Veritabanının Test Edilmesi	56
5.3.4. Oluşturulan Veritabanında Sorgulamalar	57
5.3.5. Oluşturulan Veritabanının Ağ Üzerinden Sunulması	60
6. Bölüm: SONUÇLAR VE TARTIŞMA	64
KAYNAKLAR	66
EKLER	67
ÖZGEÇMİŞ	88

KISALTMALAR

GIS	: Geographic Information System
CBS	: Coğrafi Bilgi Sistemi
UAS	: Uzaktan Algılama Sistemi
TABİS	: Türkiye Afet Bilgi Sistemi
GPS	: Global Positioning System
OK	: Obje Kataloğu
TOK	: Temel Topografik-Mekansal Obje Alanları Kataloğu
AOK	: Afet Yönetimi Obje Alanları Kataloğu
SMM	: Sayısal Mekan Modeli
SAFM	: Sayısal Afet Modeli
STH	: Sayısal Topografik Harita
SAH	: Sayısal Afet Yönetimi Haritaları
ATH	: Analog Topografik Haritalar
SQL	: Structured Query Language
DBMS	: Database Management System
RDBMS	: Relational Database Management System
AYOM	: Afet Yönetimi Obje Modeli
GPL	: Global Public Licence (Küresel Herkese Açık Lisans)
ACID	: Analysis Console for Intrusion Databases

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. PostgreSQL Veri Tipleri	32
Tablo 4.2. PostgreSQL İçin RPM Kurulum Paketleri.....	35
Tablo 5.1. TABİS-OK AYOM obje alanları ve obje türleri indeksi.....	42
Tablo A.1. TABİS-OK'na Eklenen öznitelikler ve öznitelik değerleri.....	67

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : TABİS Referans modeli.....	7
Şekil 2.2 : TABİS'te Veri modeli	8
Şekil 2.3 : TABİS'te Parça obje	10
Şekil 2.4 : TABİS'te Parça objelerin topolojik elemanları	11
Şekil 4.1 : Tipik PostgreSQL mimarisi	29
Şekil 4.2 : PostgreSQL sunucusunun başlatılması.....	36
Şekil 4.3 : Initialization işleminden sonra oluşan dosyalar.....	36
Şekil 4.4 : Postgresql.conf dosyasının düzenlenmesi	37
Şekil 4.5 : Psql ile veritabanına bağlantı	38
Şekil 4.6 : PgAdmin II giriş ekranı	39
Şekil 4.7 : Pgaccess for windows ekranı.....	40
Şekil 4.8 : Pgaccess ile veritabanına bağlanma.....	40
Şekil 5.1 : TABİS-OK'nun yapısı	48
Şekil 5.2 : Obje Modelinin Gruplandırılması.....	49
Şekil 5.3 : Oluşturulan veritabanındaki ilişkisel tablolar.....	50
Şekil 5.4 : PostgreSQL'e psql ile ilk bağlantı.....	51
Şekil 5.5 : Psql ile yeni bir kullanıcı ve veritabanı yaratılması.....	52
Şekil 5.6 : Psql ile tablo yaratma.....	53
Şekil 5.7 : PgAdminII ile tablo yaratma.....	54
Şekil 5.8 : Pgaccess ile veritabanına bağlanma.....	55
Şekil 5.9 : Pgaccess ile tablo yaratma.....	55
Şekil 5.10 : SQL komutları ile veritabanına veri girilmesi.....	56
Şekil 5.11 : Pgaccess ile veri girişi.....	56
Şekil 5.12 : Veritabanında bütünlük denetiminin testi.....	57
Şekil 5.13 : SQL ile sorgulama.....	58
Şekil 5.14 : Pgaccess ile sorgulama.....	59
Şekil 5.15 : Pgaccess ile görsel sorgulama.....	59
Şekil 5.16 : Pgaccess ile yapılan sorgulama sonucu.....	60
Şekil 5.17 : PostgreSQL'e uzaktan erişim.....	61
Şekil 5.18 : PostgreSQL'i internet bağlantısına hazırlamak.....	62
Şekil 5.19 : Ağ üzerinde veriyolu.....	62
Şekil 5.20 : PostgreSQL'e ağ üzerinde farklı platformlardan erişim.....	63

İTÜ KAMPÜSÜ AFET BİLGİ SİSTEMİ, AFET YÖNETİMİ OBJE MODELİNİN OLUŞTURULMASI VE İLİŞKİLİ SORGULAMALARIN YAPILANDIRILMASI

ÖZET

İstanbul’da olası bir afet karşısında İTÜ Ayazağa kampüsünde olabilecek can ve mal kaybını azaltmak için “Ulusal UAS-CBS Bazlı Veri Tabanı ve Afet Yönetimi Odaklı Karar Destek Sistemi Standardının Oluşturulması Projesi” kapsamında geliştirilmiş olan, Türkiye Afet Bilgi Sistemi’nin (TABİS) İ.T.Ü. Ayazağa Kampüsünde uygulanabilmesi için gerekli olan Kampüse Yönelik bir Afet Yönetimi Bazlı Obje Modelinin ve Veri Tabanının oluşturulması gerekmektedir.

Bu çalışmada, geliştirilen obje modeli ve veri tabanı ile İ.T.Ü. Ayazağa Kampüsünde kurulacak olan kampüs afet bilgi sistemi için bir altlık ve veri kataloğu oluşturulmuştur. Ayrıca geliştirilen obje modeli, bir Coğrafi Bilgi Sistemi Platformuyla ilişkilendirilip, İ.T.Ü. Ayazağa Kampüsü Afet Bilgi Sistemi yapılandırılabilir.

Bu çalışma ile oluşturulan veritabanı için ilişkisel model seçilmiş ve bu ilişkisel veritabanı modeli PostgreSQL veritabanı yazılımı ile oluşturulmuştur. Veritabanı oluşturulurken TABİS Obje Kataloğundan yararlanılmıştır. Veritabanına girilecek kampüse özel yeni veriler için obje kataloğuna öznitelik verileri eklenmiş, veritabanının kampüse yönelik optimizasyonu içinse kampüste bulunması mümkün olmayan katalog elemanları veritabanına katılmamıştır.

Bu çalışma ile, hazırlanması için bir altyapı ve veri kataloğu oluşturulan, İ.T.Ü. Ayazağa Kampüsü Afet Bilgi Sistemi herhangi bir afet halinde meydana gelebilecek hasarların azaltılması, kayıpların minimuma indirilmesi ve yapılacak olan kurtarma çalışmalarının planlı ve hızlı bir biçimde yürütülmesine yardımcı olacaktır. Çalışmada yapılandırılan sistem, her ne kadar kampüs bazlı ve küçük ölçekli görünse de, obje modelleri ve veritabanı ölçeklenebilir bir yapıda şekillendirildiği için oluşturulan modele, daha büyük ölçekli alanların afet yönetimi için parametrelerin eklenmesi, sistemin daha büyük ölçekli yerlerde uygulanmasını da mümkün kılmış ve gelecekte oluşturulması muhtemel, kent ve ülke afet bilgi sistemleri için de bir örnek ve altlık sağlamıştır.

I.T.U. CAMPUS DISASTER INFORMATION SYSTEM; CONSTITUTION OF THE EMERGENCY MANAGEMENT BASED OBJECT MODEL AND CONSTRUCTION OF THE RELATED QUERIES

SUMMARY

To mitigate the property and life losses at the İ.T.U. Ayazağa Campus, after a probable disaster that may occur in İstanbul, it is strongly necessary to constitute an emergency management based object model and database in order to apply the Turkey Disaster Information System (TABİS) which was developed within the scope of “Constitution of the National RSS-GIS Based Database and Emergency Management Focused Decision Support System Project”.

In this project, an infrastructure and a data catalog have been constituted for the Campus Disaster Information System, which will be established in the İ.T.U. Ayazağa Campus, using the created object model and data base. Furthermore, the object model can be connected to a geographic information system platform, and by this way İ.T.U. Ayazağa Campus Disaster Information System (KABİS) can be constructed.

The relational model has been chosen for the database which was constituted in this study and this model has been realized by using the PostgreSQL database software. TABİS-OK (Turkey Disaster Information System Object Catalog) was used in the creation phase of the database. Attribute data have been added into the object catalog for the special data regarding the Campus. Whereas, for the optimization of the database devoted to the campus, the objects which may not be found or occurred in the campus, have not been added into the database.

In this study, an infrastructure and a data catalog have been established. In this respect, the İ.T.U. Ayazağa KABİS will help to reduce the damages, to minimize the losses and to execute the rescue operations rapidly on a probable disaster.

The system that was constructed in the study seems campus based and small scaled. But, because of the object models and the database that were created in a form that can be scaled, addition of required parameters for larger areas in emergency management will make the system available for the larger areas.

1. GİRİŞ

1.1 Giriş ve Çalışmanın Amacı

Her ne kadar ülkemiz özellikle deprem başta olmak üzere sel, heyelan gibi doğal ya da yangın, terör ve benzeri insan kaynaklı felaketlerle uzun yıllardır karşı karşıya kalmakta olsa da, gerek felaketlerin olduğu bölgelerdeki endüstrileşme derecesi ve nüfus yoğunluğundaki artışın getirdiği ekonomik ve yaşamsal kayıplardaki artış gerekse insanlarımızdaki bilinçlenme dolayısıyla bu felaketler son zamanlarda oldukça fazla dikkat ve endişe çekmeye başlamıştır. Bu nedenle, afetlere ve afet yönetimine yönelik çalışmalarda artış görülmektedir. Bütün bu çalışmalar arasında Coğrafi Bilgi Sistemine (CBS) dayalı afet yönetimi en önde gelenlerden biridir.

Afet bilgi sistemleri, meydana gelebilecek deprem, sel, heyelan, gibi doğal felaketlerin ardından hasar ve kayıpları minimum seviyede tutmak için ya da yangın, gibi insan kaynaklı felaketlerin olabilirliklerini en aza indirebilmek için ve tüm bu felaketlerin meydana gelmesinin ardından yapılacak arama, kurtarma, hasar tespit, onarım ve iyileştirme çalışmalarının en kısa sürede ve en doğru şekilde yapılmasını sağlamak amacını taşır.

İstanbul’da olası bir afet karşısında İTÜ Ayazağa kampüsünde olabilecek can ve mal kaybını azaltmak için “Ulusal UAS-CBS Bazlı Veri Tabanı ve Afet Yönetimi Odaklı Karar Destek Sistemi Standardının Oluşturulması Projesi” kapsamında geliştirilmiş olan, Türkiye Afet Bilgi Sistemi’nin (TABİS) İ.T.Ü. Ayazağa Kampüsünde uygulanabilmesi için gerekli olan Kampüse yönelik bir Afet Yönetimi Bazlı Obje Modelinin ve Veri Tabanının oluşturulması bu çalışmanın amacını oluşturmaktadır.

TABİS; “Uzaktan Algılama Sistemi (UAS), Global Konum Belirleme Sistemi (GPS) ve diğer veri toplama tekniklerini kullanarak özellikle acil durum planlamasında, uygulamasında ve herhangi bir afet durumunda, afet yönetimi ve hasar tahmininde kullanılabilecek, ayrıca normal zamanlarda merkezi ve taşra idaresi (bakanlıklar, valilikler, kaymakamlıklar, belediyeler) için karar destek sistemi olarak yararlanılabilecek Türkiye genelinde uygulamalara temel oluşturan CBS (Coğrafi

Bilgi Sistemi) tabanlı bir bilgi ve yönetim sistemi modeline ilişkin standartların sağlandığı bir çalışmadır.

Kurulan sistem, kullanıcısı olan kurumun güncel, doğru, standart ve tutarlı veriye ulaşmasını sağlayacaktır. Bilgi fazlalığı karmaşasını engelleyen bir anlayışla gerekli bilgilerin paylaşımı ve ilgili bilgilerin bir araya getirilmesi ve bu bilgilerin karşılıklı etkileşim sonuçlarının planlama ve yönetim amaçlı kullanımı gözetmektedir.

Günümüzde hızla güncellenebilen sistematik bilgiye sahip olmak için geliştirilen teknolojiler birçok ülkede kullanılmaya başlanmıştır. Bu teknolojilerden olan ilişkisel veri tabanı ile çeşitli kurumlardan alınan bilgiler kullanışlı bir şekilde güncellenebilecek ve coğrafi bilgi sistemleri de (CBS) bu bilgilerin kullanılması ve farklı analizlerin elde edilmesinde yardımcı olacaktır“. (Şahin ve diğ., 2002)

Bu çalışmada veri tabanı, güvenliği ve veri saklama kapasitesinin büyüklüğü nedeniyle UNIX bazlı LINUX işletim sistemi üzerinde PostgreSQL veri tabanı programında ilişkisel yapıda hazırlanmıştır. Her ne kadar unix bazlı işletim sistemleri bu veritabanının gelecekteki kullanıcıları düşünüldüğünde kullanım ve uyum zorluğu taşıyor gibi görünse de veri tabanına her türdeki işletim sisteminden birçok farklı arayüz kullanılarak ya da kullanıcı profiline uygun yeni bir arayüz geliştirilerek de bağlanmak mümkündür. Böylelikle hem veri güvenliği hem de kullanıcı dostu bir yapıyı bir arada kullanmak mümkündür. Bu çalışmada veri tabanı linux işletim sisteminde hazırlanmış ve sunucu olarak da linux işletim sistemi seçilmiştir.

Geliştirilen obje modeli ve veri tabanı ile İ.T.Ü. Ayazağa Kampüsünde kurulacak olan kampüs afet bilgi sistemi için bir altlık ve veri kataloğu oluşturulmuştur. Ayrıca geliştirilen obje modeli, bir Coğrafi Bilgi Sistemi Platformuyla ilişkilendirilip, İ.T.Ü. Ayazağa Kampüsü Afet Bilgi Sistemi kolaylıkla yapılandırılabilir. Böylece, İstanbulda gerçekleşebilecek herhangi bir afet anında ve afet sonrasında meydana gelebilecek hasarların azaltılması, kayıpların minimuma indirilmesi ve yapılacak olan kurtarma çalışmalarının planlı ve hızlı bir biçimde yürütülmesi sağlanabilir.

Çalışmada yapılandırılan sistem, her ne kadar kampüs bazlı ve küçük ölçekli görünse de, obje modelleri ve veri tabanı ölçeklenebilir bir yapıda şekillendirildiği için oluşturulan sisteme, kayıp tahmini için daha ayrıntılı bir kampüs envanter veritabanı eklemek, risk altındaki altyapı sistemleri için hassasiyet fonksiyonları geliştirmek, kayıplar, sığınak ihtiyaçları için parametreler geliştirip eklemek, CBS (Coğrafi Bilgi

Sistemleri) platformlarının tüm avantajlarını kullanabilen acil durum yöneticileri, bilimsel araştırmacılar ve karar mekanizmalarında çalışanları da içeren geniş çeşitlilikte kullanıcılar için uygun ve internet üzerinden hizmet sunacak bir arayüz geliştirip eklemek de mümkündür. Böylece sistemin daha büyük ölçekli yerlerde uygulanması da mümkün olacak ve gelecekte oluşturulması muhtemel, kent ve ülke afet bilgi sistemleri için de örnek bir çalışma olacak ve önemli bir altlık sağlayacaktır.

2. TABİS

2.1 TABİS'in Amacı

17 Ağustos 1999 ve 12 Kasım 1999 Marmara Depremleri, resmi kayıtlara göre 20000 insanımızın ölümüne, çok sayıda insanımızın yaralanmasına, işsiz ve evsiz kalmasına, 30000'in üzerinde binanın hasar görmesine sebep olmuştur. Özellikle depremler ve diğer doğal afetler ülkemizin büyük bir kısmını halen tehdit etmeye devam etmektedir. Bu nedenle, İçişleri Bakanlığı ile İstanbul Teknik Üniversitesi arasında Mayıs 2001'de imzalan bir anlaşma ve proje paketi ile İTÜ'nün bilgi birikiminin ülke yararına sunulmasına karar verilmiştir (Şahin ve diğ., 2002).

Projenin amacı; Uzaktan Algılama Sistemi (UAS), Global Konum Belirleme Sistemi (GPS) ve diğer veri toplama tekniklerini kullanarak özellikle acil durum planlamasında, uygulamasında ve herhangi bir afet durumunda, afet yönetimi ve hasar tahmininde kullanılabilecek, ayrıca normal zamanlarda merkezi ve taşra idaresi (bakanlıklar, valilikler, kaymakamlıklar, belediyeler) için karar destek sistemi olarak yararlanılabilecek Türkiye genelinde uygulamalara temel oluşturan CBS (Coğrafi Bilgi Sistemi) tabanlı bir bilgi ve yönetim sistemi modeline ilişkin standartların geliştirilmesidir. Geliştirilecek bu coğrafi bilgi sistemi adının ise **Türkiye Afet Bilgi Sistemi-TABİS** olması önerilmiştir (Şahin ve diğ., 2002).

Proje ile Türkiye'nin afet planlaması ve yönetimi konusunda iller arasında uyumlu çalışmayı ve koordinasyonu sağlayabilecek standartlar ortaya konarak bölgesel, çevresel ve yönetsel bir bilgi sistemi modeli oluşturulmaktadır.

Anılan projenin disiplinler arası tasarımılanmasının gerekçesi olarak aşağıdaki hususlar sayılabilir (Şahin ve diğ., 2002):

- Felaketin olduğu bölgedeki nüfus ve yapılaşma yoğunluğuna bağlı olarak kayıpların derecesi artmaktadır. Doğa ya da insani nedenlerden kaynaklanan felaketler endüstrileşme arttıkça daha iç içe geçmektedir.

- Ülkemizde doğa felaketlerinin neden olduğu risklerin tahmini ve sistematik olarak araştırılması için kullanılabilecek bilgi alt yapısı yeterli değildir, amaca yönelik uygun mekanizmalar geliştirilmemiştir.
- Benzer biçimde afet sonuçlarının toplumun ekonomik, sosyal yaşamını nasıl etkileyeceği henüz sağlıklı analiz edilememektedir.
- Afet anında ilgili birimleri yönlendirecek entegre bilgilere ulaşmayı sağlayacak araçlar yoktur.

Böyle bir sistemin temel kullanıcıları: Sivil savunma birimleri, İdari birimler (valilikler, kaymakamlıklar, belediyeler), Hizmet sektörü (bankalar, sigorta şirketleri, inşaat şirketleri), Araştırma kurumları, Kamu yararına çalışan sivil toplum örgütleridir (Şahin ve diğ., 2002).

2.2 TABİS Obje Kataloğu (TABİS-OK)

Türkiye Afet Bilgi Sisteminin kalbi temel mekansal veri tabanıdır. TABİS olarak adlandırılan bu sistemin referans modeli (TABİS Referans Modeli) (Şekil-1.1) iki vektörel bileşenden meydana gelmektedir. Bu bileşenler ;

- Sayısal mekan modeli (**SMM**) ve
- Sayısal afet modelidir (**SAFM**)

olarak adlandırılmışlardır.

Her iki sayısal model de mekanı, objeye (nesne) dayalı bir temel üzerinde tek tek elemanlarına ayırarak oluşturmaktadır. Bu işlem veri tabanı modellemesinde mekanın “atomize” edilmesidir. Her iki sayısal modelin atomize verileri obje katalogu biçiminde hazırlanmıştır (Şahin ve diğ., 2002).

Hazırlanan TABİS-OK, sonuç ürünleri SMM ve SAFM olan iki temel bileşenden oluşmaktadır. Bu iki bileşen de;

- TABİS-Temel Topografik-Mekansal Obje Alanları Kataloğu (TABİS-TOK),
- TABİS-Afet Yönetimi Obje Alanları Kataloğu (TABİS-AOK)

olarak adlandırılmışlardır.

TABİS-TOK'daki amaç, sistemin tesis edileceği bölgedeki mekanda varolan ve topografyanın karakteristik parçaları olan somut objelerin modellenmesidir. Bu amaca paralel olarak TABİS-OK'nun bu bileşeni "Temel Topografik-Mekansal Objeler Alanları" olarak adlandırılmıştır. TABİS-TOK ülke genelinde kendi amaçları için ayrıntılı mekansal bilgi sistemi kurmak isteyen kamu ve özel kurumlar için de kesin bir veri standardı niteliğine sahiptir. TABİS-TOK'nda bulunan obje modelleri, obje tanımları, öznitelik tanımları, özniteliklere ait veri tipleri ve öznitelik değerleri analog topografik bir harita içeriği ile eşleştirilebileceği (örtüştüğü) için afet yönetimi amaçlı TABİS-TOK modeline uygun oluşturulacak bir coğrafi bilgi sistemi, bölge aynı olmak şartı ile birçok farklı hedeflere sahip ve bu farklı hedeflere yönelik kurulacak olan diğer coğrafi bilgi sistemleri ile tamamen uyumlu olarak çalışabilecektir. TABİS-TOK'na göre modellenmiş bir sanal harita "Sayısal Mekan Modeli" (SMM), TABİS-AOK'na göre modellenmiş bir sanal harita ise "Sayısal Afet Modeli" dir (SAFM).

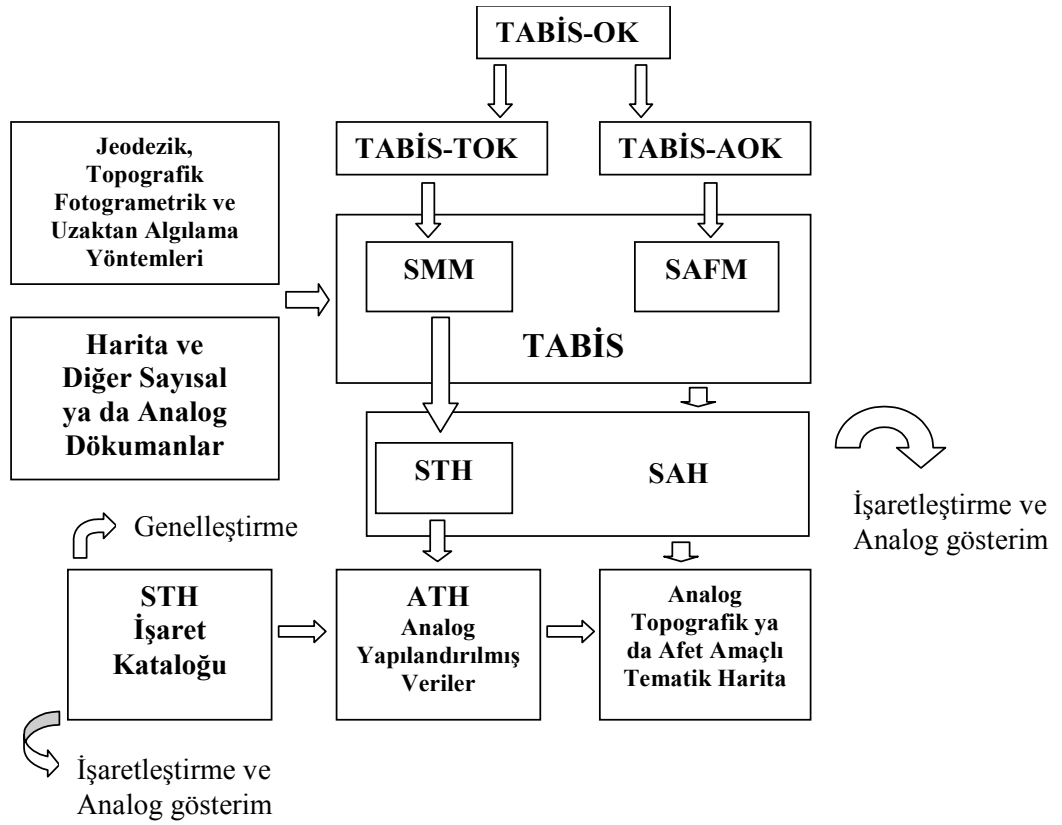
TABİS-TOK'na göre modellenmiş olan SMM'lerinde yapılacak olan veri genelleştirme işlemleri ile "Sayısal Topografik Haritalar" (STH), ve bu STH'lar ile afet analizi ve yönetimi amaçlı verilerin oluşturduğu sayısal modellerin kombinasyonu ile çok sayıda değişik içerikli "Sayısal Afet Yönetimi Haritaları" (SAH) elde edilir. Aynı yöntem kullanılarak, 1:5000 ve 1:25000 ölçek aralığında "Analog Topografik Haritalar" (ATH) ve benzer ya da değişik içerikli, afet yönetimini temel alan analog tematik haritalar da elde edilebilir.

TABİS-OK'dan sağlanacak diğer yararlar aşağıdaki başlıklarda toplanabilir (Şahin ve diğ., 2002);

- Belli standartlarda verilerle modellenmiş temel coğrafi bilgi sistemi altyapısı ile çalışılacağından verilerin kontrolsüzce büyümesi kesinlikle önlenecektir. Aynı verilerin farklı kurumlar tarafından yönetilmesi engellenecektir.
- Disiplinler arası çalışma ile verimlilik artacak, ülke genelinde değişik amaçlar için kurulacak kaliteli coğrafi bilgi sistemlerinin temeli atılmış olacaktır.
- Veri değişimi alanında geliştirilecek diğer standartlar sayesinde coğrafi bilgi sistemi kullanıcıları herhangi diğer bir sistem ile çalışabilmek amacıyla zahmetli veri değişim programları yazmak zorunda kalmayacaklardır.

- Her amaç için aynı derecede verimli kullanılabilir bir CBS amaçlı mekan modellemesi yoktur. Bu nedenle ilgili kurumlar kendi mekansal referanslı olgularını kendileri modelleyerek ve bu temel sistem üzerine uygulayarak entegrasyon sağlayabileceklerdir.
- Analog kartografik ürünler için standart kartografik görselleştirme sağlanabilecektir. Böylece bir kurum tarafından üretilmiş bir sayısal ya da analog ürün herkes tarafından daha kolay anlaşılacaktır.
- İlgili CBS ler için tek bir jeodezik referans sistemi olacağından sistemler arasında geometrik-mekansal entegrasyon sorunsuz işleyecektir.

Aşağıda TABİS-OK'un mekan modelleme yaklaşımı grafiksel olarak TABİS Referans Modeli adıyla gösterilmiştir (Şekil 2.1).



Şekil 2.1 TABİS Referans modeli

OK'nda ana başlıklarıyla aşağıdaki konular açıklanmaktadır.

1. Topografik karakterli somut objeler,

2. Afet yönetimini ilgilendireceği öngörülen uzmanlık alanı objeleri,
3. Obje tanımları ve objeleri daha yakından tanımlamayı amaçlayan öznitelik verileri.

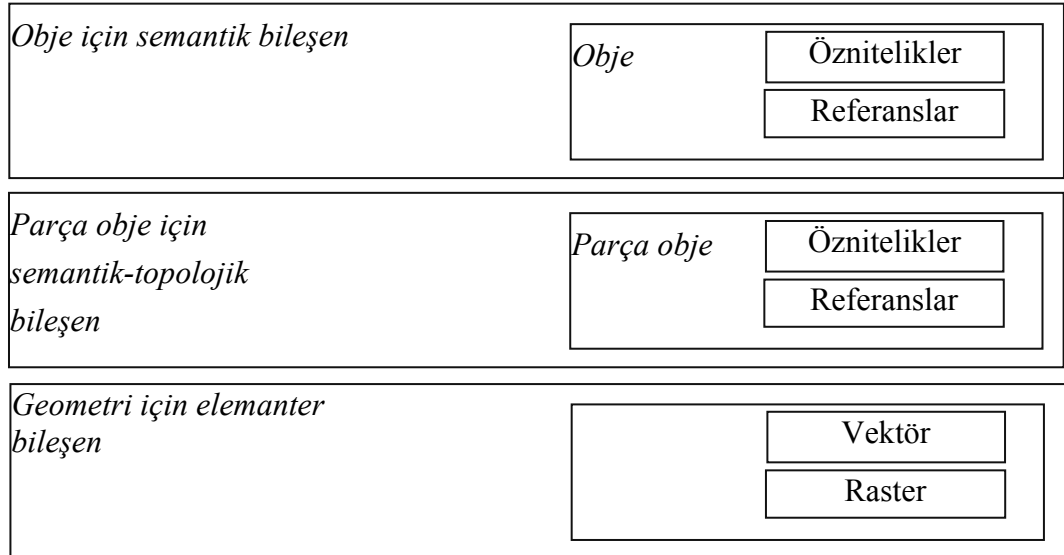
2.2.1. TABİS-OK’nda Obje Kavramı

TABİS-OK’nda “obje”, duyu organları ile izlenebilen ya da değişik aletler kullanılarak varlığı saptanabilen mekanda var olmuş, var olan ya da var olması muhtemel her şeydir. Mekanda var olmak kavramı ile ilgili objelerin mekansal referansının bulunması ya da diğer bir ifade ile geometrik olarak atomize edilebilmesi anlatılmaktadır (Şahin ve diğ., 2002).

Atomize edilmenin ise noktasal, çizgisel ve alansal olarak üç tipi bulunmaktadır.

Mekansal referansı olan bir elemanter objenin aşağıdaki tüm koşulları sağlaması gerekir.

1. Mekanda başka objelerle sınırları belirlenebilen planimetrik geometrisi,
2. Topolojik ilişkileri,
3. Kendine özgü özniteliklerinin olması,
4. Herhangi bir biçimde tanımlanmış olması.



Şekil 2.2 TABİS’te Veri Modeli

TABİS-OK’nda geçen somut, kompleks, ya da parça obje olsun, tüm objeler için semantik ve topolojik bileşenler öznitelikler ve varolsun ya da olmasın referanslardan oluşmaktadır. Geometri için olan elemanter bileşenler ise vektör ve raster verilerden oluşmaktadır (Şekil 2.2).

TABİS-TOK’da yer verilmiş objeler belli mantıksal kurallara göre oluşurlar. Örneğin bir hastane, bir parsel, bir elektrik direği, bir köprü vb. klasik somut objelerdir. TABİS-TOK aynı zamanda “kompleks objeler” kavramı ile de çalışmaktadır. Kompleks obje, sistem modellemesi bakımından mekan ile noktasal, çizgisel ya da alansal olarak referanslandırılmış bulunan somut objelerin (genellikle topolojik olarak birbirlerine komşu ve semantik olarak belli bir birimin parçaları olan) yeni bir ad altında birleştirilmesinden oluşan objelerdir. Buna örnek olarak bir hastane, sağlık ocağı, deney laboratuvarı, veteriner, vb. somut objeler ,“sağlık kompleksi” kompleks objesi altında birleştirilmesi verilebilir.

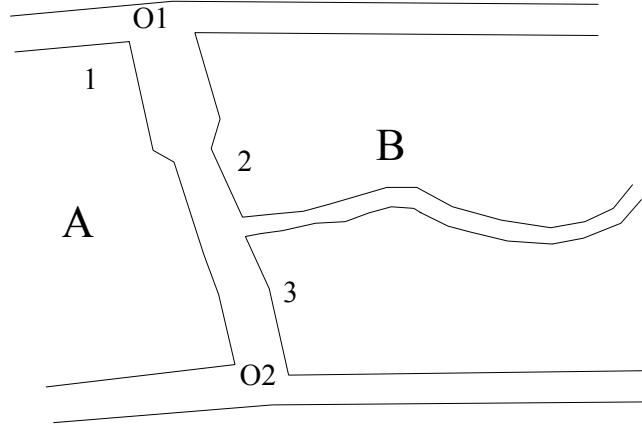
Bir obje bazı kriterlere göre ayrılmış parça objelere de sahip olabilmektedir. Buna göre TABİS-OK’nda geçen herhangi bir obje planimetrik-geometrik boyutlarına bağlı olarak noktasal, çizgisel ya da alansal olabilir. Bir obje bunlardan ikisi ya da üçü kesinlikle olamaz. Objelerin bu özellikleri TABİS-OK’nda “obje tipi” olarak adlandırılmıştır (Şahin ve diğ., 2002).

Örneğin bir O1 noktası ile O2 noktası arasındaki “iki şeritli”, “zeminde” “köy yolu” özellikli A “cadde” objesini ele alalım. Sistemde bu obje “parça objelere” ayrılır (Şekil 2.3).

Bu parçalama:

- Topolojik olarak bu objeye komşu (kesiksiz olarak) aynı temel obje alanındaki başka bir obje ile düğüm noktalarında ve
- Objeye üzerinde belli bir özniteliğin değiştiği noktadan itibaren (örneğin yol genişliğinin değişmesi)

yapılır. Bu örnekte B ise başka bir objedir.



Şekil 2.3 TABİS’te parça obje

2.2.1.1. Obje Grubu

Aynı türden objeleri içine alan bir kavramdır. Demiryolları, karayolları gibi. Obje grubunun ait olduğu tek bir temel topografik-mekansal obje alanı vardır. Bir obje grubu içinde yer alan obje türleri birbirleri ile prensip olarak örtüşmezler. Yalnız bazı obje türleri bazılarını içersinde barındırırlar. Örneğin “A.A.2.06 Stadyum”, “A.A.2.07 Spor Alanı”, “A.A.2.08 Yüzme Havuzu” “A.A.2.01 Spor Kompleksi” nin elemanları olması gibi. Bu durumda ilgili obje türü içinde yer alan daha küçük alanlı obje türlerinin sınırları kesinlikle içinde bulunduğu objenin sınırı dışına taşamaz. En fazla iki sınır genelde kısmen çakışık olur.

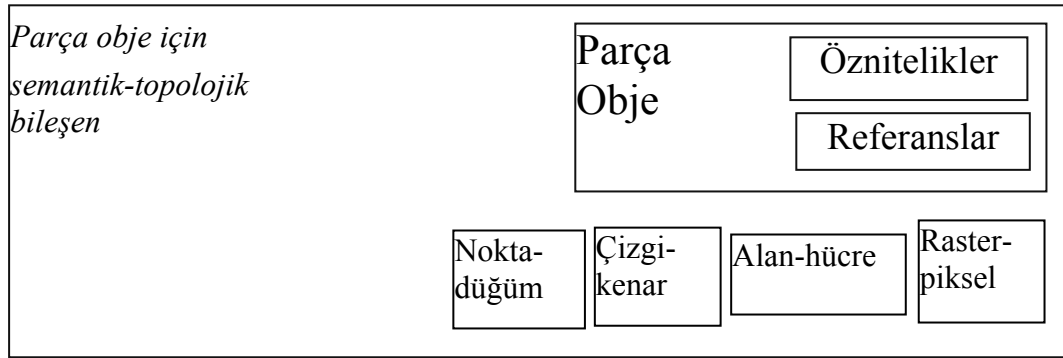
Bir obje alanının obje grupları yoksa (B. Afet Yönetimi Obje Alanları” altında bulunan “B.C. Kritik Tesisler” obje alanı gibi) ilgili obje alanı aynı anda obje grubu olarak değerlendirilir ve iki harfî izleyen kod “0” alınır. Böyle bir kod sistemi ile, ilgili obje alanı altında tek obje grubunun bulunduğu ifade edilmektedir (Şahin ve diğ., 2002).

2.2.1.2. Obje Türü

TABİS-TOK ve TABİS-AOK nın oluşturulmasında aynı obje için tanımlanan kavramdır. “Elektrik Direği”, “Rogar”, “Tarla”, “Anıt Ağaç”, “Deformasyon Hızı”, “Orman Yangını”, “Dini yapılar” vb. gibi. Şüphesiz obje türleri de ait oldukları obje alanı ve obje grubuna bağlı olarak kodlar almışlardır. Bir obje türü ancak tek bir obje grubu altında yer alabilir (Şahin ve diğ., 2002).

2.2.1.3. Parça Obje Kavramı

Parça obje bir objenin alt parçalarıdır. Kendine özgü bir geometrisi ve özniteliği vardır. Bir obje üzerinde özniteliğin ya da topolojik bir düğüm noktasının bulunduğu yerlerde parça objeler başlar ya da biter. Parça objeler mutlaka bir objeye aittirler. Genellikle “obje tipi” çizgisel olan objelerin, parça objeler biçiminde oluşturulması gerekebilmektedir (Şekil 2.3) Şahin ve diğ. (2002). Parça objelerde TABİS-OK’nda yeralan diğer objeler gibi mekansal referanslara sahip olmalıdırlar. Diğer bir ifade ile geometrik olarak atomize edilebilmelidirler. Atomize edilme ise noktasal, çizgisel ve alansal olarak üç şekilde yapılabilir (Şekil 2.4).



Şekil 2.4 TABİS parça obje modelinin topolojik elemanları

2.2.1.4. Öznitelik Verileri

Bir obje ya da obje parçasına ait nitel ya da nicel bilgilerdir. Bir binanın iskan yılı, yapıldığı malzeme, en üst noktasının yüksekliği, kat adedi ya da oturduğu alanın m2 cinsinden değeri gibi (Şahin ve diğ., 2002).

İster TABİS-TOK isterse TABİS-AOK içinde bulunsun bazı obje türlerine ait öznitelik verilerinin alması muhtemel değerleri de verilmiştir. Bu özniteliklerin, sistemin kurulması aşamasında kardinallik dereceleri de dikkate alınarak kullanıcıya sistem tarafından sunulması gerekir. Kardinallik derecelerinin sistem tarafından kontrol altında tutulabilmesi sistemin tutarlılık ilkesi bakımından belirleyici öneme sahiptir. Bir özniteliğin “kardinallik” derecesi, ilgili öznitelik değerlerinin ilgili obje türüne atanmasındaki zorunluluk ve tek ya da çok değerlilik durumunu göstermektedir (Şahin ve diğ., 2002).

“Karakter dizgesi (string)” veri tipindeki öznitelik değerleri için sistemin kurulmasında ve hizmetinde kullanılacak yazılım, karakter sayısını sonsuz sayıda yönetebilmelidir.

Özellikle TABİS-AOK altında yer verilmiş olan “B.B.Afet Riski Alanları” obje alanı, “B.C.Kritik Tesisler” obje alanı ve “B.D.Tarihi ve Kültürel Alanlar” obje alanı altında yer alan obje türlerinin somut elemanlarına ait öznitelikler gerekirse daha da genişletilerek bir veri tabanında ayrı olarak yönetilmelidir. Bunun temel nedeni bir afetin oluş alanının ya da muhtemel bir afet analizinde afetin oluşacağı model bölgenin mekansal referansının değişken olmasıdır. Fakat bu bilgi sistemi afet analizlerinde kolaylıkla plancıların hizmetine TABİS ile beraber sunulabilir yapıda olmalıdır. Diğer ifade ile, veri tabanında yönetilen bu tür verilerin, mekansal olarak modellenen afet ile ilişkisi, değişik analiz çalışmaları bakımından sağlanabilmelidir.

Herbir öznitelik için bir kısaltma tanımlanmıştır. Anlamlı kısaltmalar yalnızca harflerden oluşur ve tek anlamlıdır. Ayrıca herbir obje türüne ait somut elemanların “Adı” verisi öznitelik karakterindedir. Bunların kardinallik derecesi her bir obje türü için yine ayrıca belirtilmiştir. “Adı” özneliği veri tipi bakımından ise “karakter dizgesi-string” türündedir (Şahin ve diğ., 2002).

2.2.1.5. Öznitelikler İçin Veri Tipleri:

Obje türlerine ilişkin öznitelikler aşağıdaki veri tiplerinde olabilir (Şahin ve diğ., 2002):

- Sayı (Number)
- Gerçek sayı (Real)
- Tam sayı (Integer)
- Mantıksal (Logical)(and, oder, or vb.)
- Anahtar (Boolean)(yanlı ya da doğru)
- Karakter dizgesi (String)
- İkili sistem (Binary)
- Dizge (Array)
- Tarih (Date)

- Money (TL, USD vb.)

2.2.1.6. Öznitelikler İçin Veri Kardinallik Derecesi

Bir özniteliğin kardinallik derecesi, ilgili öznitelik değerlerinin ilgili obje türüne atanmasındaki sayısal durumunu göstermektedir. Bu sütunda yer alan “kardinallik” gösterimlerinin açıklaması aşağıda verilmiştir (Şahin ve diğ., 2002).

1:1 İlgili özniteliğin ilgili objeye mutlaka atanması gerekir ve atama ancak tek değerli olabilir. Örnek olarak bir fabrikanın “**DRM Durumu**” öznitelik değerinin “**110 Çalışıyor**” ya da “**111 Hizmet dışı, kapatıldı**” olması gibi.

0:1 İlgili öznitelik ilgili objeye atanabilir ya da atanmayabilir. Örneğin “**A.A.3.10 Bina**” obje türünün “**GEC Binanın Gece Nüfusu**” özniteliği için bu tür bir kardinallik geçerlidir.

0:? İlgili öznitelik ilgili objeye atanmayabilir ya da çok değerli olarak atanabilir. Örneğin “**A.A.3.10 Hastane**” obje türünün “**DPG Deprem Güvenliği**” öznitelik değeri için bu tür bir kardinallik öngörülmüştür.

2.2.2. Afet Yönetimi Obje Alanları Katalogu (TABİS-AOK)

TABİS-AOK kapsamında afet yönetimi için kurulacak bir coğrafi bilgi sisteminde yer alması öngörülen, münferit mekansal referansı olan obje alanları ele alınmıştır. Bu alanlar belli bir uzmanlık gerektiren jeoloji, toprak yapısı, kritik tesisler, risk bölgeleri gibi konu alanlarını içermektedir (Şahin ve diğ., 2002).

Bu çerçevede yer alan temel obje alanları, obje türleri bu obje türlerine ilişkin öznitelik değerleri afet planlamasında temel oluşturan risk analizi için gereken veriler yapılandırılarak düzenlenmiştir. Verilerin yapılandırılmasında şehir plancıları, afet planlaması uzmanları, jeologlar, toprak ilmi ile ilgili uzmanlardan yararlanılmıştır.

Bilindiği gibi modern afet yönetimi afet planlaması ile mümkündür. Afet planlamasının aşamaları olan risk analizi için yöreye özgü olarak CBS de oluşturulan verilerin kullanılması büyük bir önem taşır. Benzer şekilde afet sırasında müdahale aşamasında ve daha sonra iyileştirme aşamasında da çabuk ve doğru karar almak için bu verilere başvurulması söz konusudur. Yukarıda belirtilen aşamalarda kullanılacak veriler temel obje alanlarından sağlanabildiği gibi yöreye özgü olarak bilimsel

yöntemlerle elde edilecek risk analizi ile ilgili diğer verilerin de bu yapıya ilave edilmesi mümkün olacaktır (Şahin ve diğ., 2002).

Bu bölümde sistemden yararlanılarak planlama yapılmasına olanak sağlayacak “B.G Planlar” obje alanı da bulunmaktadır. Burada oluşturulacak objeler ve bu objelerle diğer mekansal objeler arasında ilişki kurularak değişik mekansal analizlerin yapılması mümkün olabilecektir.

“B.F. Çevre Kirliliği” obje alanı üç başlıkta (obje grubu) ele alınmaktadır. Bunlar, su, hava ve gürültü kirliliğidir. Bu obje grubu dahilinde yer alan veriler, genellikle bir noktada yapılan ve ilgili kirliliği karakterize etmek için yaygın olarak kullanılan ölçme değerlerine dayanmaktadır. Bu ölçme değerleri, anılan bağlamda birer öznitelik verisi olarak ele alınmışlardır. Bu bölümde yer alan kavramlar ilgili yönetmelikler temel alınarak tanımlanmışlardır (Şahin ve diğ., 2002).

3. VERİTABANI

3.1 Veritabanının Tanımı ve Amacı

Merriam-Webster sözlüğünde veritabanı; “Özellikle hızlı arama, düzeltme ve ekleme için organize edilmiş, genellikle büyük çaplı olan bir veri koleksiyonudur” olarak tanımlanır. Bir veritabanı yönetim sistemi (DBMS: DataBase Management System), kitaplıklar uygulamalar ve yardımcı programların birleşmesinden oluşur ve verilerin saklanması ve yönetilmesi ile ilgili konulardaki ayrıntılardan veritabanı yöneticilerini kurtarır. Aynı zamanda, kayıtların güncellenmesi ve kayıtlar üzerinde araştırma yapılması da mümkündür (Kaynarca, 1994). Veritabanları, verilerin saklanması ve yönetilmesi için kullanılmalıdır. Küçük bilgiler için metin dosyaları yeterli olacaksa bunun için veritabanı kullanılması gerekmez. Bir kurumun tüm belgeleri üzerinde arama/sorgu yapılacaksa bunun için de veritabanı kullanmanız gerekmeyecektir, bir internet ağı indeksleme programı da bunun için yeterli olabilir. Ancak, bir e-ticaret sitesinde müşterilerin bilgilerinin tutulması söz konusu ise, ya da bir kurumun değişik birimlerine kullanıcı/müşteri/personel giriş/çıkış bilgileri tutulacaksa bunun için verinin büyüklüğüne ve çalışacağınız platforma uygun bir veritabanı sunucusunu kullanmak gerekecektir (Gündüz, 2002b).

Veritabanı yönetim sistemleri zamanla çok daha fazla özellikler kazanmaktadırlar. Veritabanı;

- Belirli bir tarzda organize edilmiş bilgi “koleksiyon”udur.
- En az bir tablodan oluşmak zorundadır.
- Veritabanının içindeki tablolar ise veri alanlarından oluşur (data field).
- Kitaplıklar, uygulamalar ve yardımcı programların birleşmesinden oluşur.

- Verilerin saklanması ve yönetilmesi ile ilgili konulardaki ayrıntılardan veritabanı yöneticilerini kurtarır.
- Kayıtların güncellenmesi ve kayıtlar üzerinde araştırma yapılması da mümkündür.

3.2 Veritabanı Türleri

Günümüzde, dünyaca kabul görmüş beş temel veritabanı türü bulunmaktadır (Başoğlu, 1994). Bunlar;

1. Hiyerarşik Veritabanı (Hierarchical Database)
2. Ağ Yaklaşımı (Network, Codasyl Approach)
3. İlişkisel Veritabanı (Relational Database)
4. Melez Veritabanı (Hybrid Database)
5. Nesnel Veritabanı (Object Oriented Database)

Mevcut temel veritabanı türlerinden dünyaca en çok kabul gören ve kullanılan üçüncü sıradaki ilişkisel veritabanıdır ve aynı zamanda bu çalışmada da kullanılan veritabanı türüdür.

3.2.1 İlişkisel (Relational) Veritabanı Modeli

1970’de Dr. E. F. Codd’un yazdığı “A Relational Model of Data for Large Shared Data Banks” adlı yazı ile ilişkisel olarak adlandırılan yeni bir tür veritabanı tanımlanmış oldu. Bu yazıda, Dr. Codd ilişki mantığını tanıtmış ve tabloların gerçek dünyadaki nesneleri göstermekte, nitelemekte nasıl kullanılabileceğini belirtmiştir. İlişkisel Veritabanının tanımı Dr. Codd’un bu yazısında yer alan ve “Codd’un 12 Kuralı” olarak adlandırılan ilişkisel veritabanına yönelik 13 adet kuralın içinde bulunmaktadır (Perkins, 2001).

İlişkisel veritabanlarını tanımlayan bu önemli kurallar:

- Kural 0 Kurulum: Bir ilişkisel Veritabanı Yönetim Sistemi (RDBMS), veritabanlarını bütünüyle kendi ilişkisel yetenekleri ile yönetebilmelidir.
- Kural 1 Bilgi: İlişkisel bir veritabanındaki tüm bilgiler (tablo ve kolon isimleri de dahil) açıkça tablolarda değerler olarak temsil edilmelidirler.
- Kural 2 Garantili Erişim: Bir ilişkisel veritabanındaki her değere, tablo adı, birincil anahtar değeri (primary key value) ve kolon adı'nın kombinasyonu ile oluşan erişim garantilenmelidir.
- Kural 3 Sistemik Boş Değer Desteği: DBMS, boş değerlerin (bilinmeyen ya da uygulanamayan veri) olağan (default) değerlerinden ayrı ve her alandan (domain) bağımsız davranışları için sistemik bir destek sağlamalıdır.
- Kural 4 Aktif-Online İlişkisel Katalog: Veritabanının ve içeriğinin tanımları mantıksal bir seviyede simgelandığı için veritabanı dili kullanılarak sorgulanabilmelidir.
- Kural 5 Ayrıntılı Veri Altdili: Veri tanımını, manipülasyonunu, güvenilirlik kurallarını, yetkilendirmeyi ve işgörmeyi ayrıntılı olarak destekleyen, en azından bir dilin desteklendiği iyi tanımlı bir syntax'ı olmalıdır.
- Kural 6 Görünüm Yenileme Kuralı: Teorik olarak mümkün olan tüm gösterimler sistem aracılığıyla yenilenmeli ve güncellenmelidir.
- Kural 7 Belirli-Seviye Eklemeleri, Güncellemeler ve Silmeler: DBMS, sadece set-level düzeltmelerini değil, ayrıca set-level eklemeleri, güncellemeleri ve silmelerini de desteklemelidir.
- Kural 8 Fiziksel Veri Bağımsızlığı: Saklama yapılarına fiziksel erişim metodları değiştiğinde uygulama programları ve ad hoc programları mantıksal olarak etkilenmemelidir.
- Kural 9 Mantıksal Veri Bağımsızlığı: Tablo yapılarına değişiklik yapıldığında uygulama programları ve ad hoc programları mantıksal olarak bundan etkilenmemelidir.

- Kural 10 Bütünlük Bağımsızlığı: Veritabanı dili bütünlük kurallarını tanımlamaya uygun ve yeterli olmalıdır. Online katalogda saklanmalı ve atlanılamamalıdır.
- Kural 11 Dağıtım Bağımsızlığı: Veri ilk dağıtıldığında ya da yeniden dağıtıldığında, uygulama programları ve ad hoc istekleri mantıksal olarak etkilenmemelidir.
- Kural 12 Altversiyonsuzluk: Alt seviye dilleri kullanarak, veritabanı dili ile tanımlanmış güvenirlik kurallarının atlanması mümkün olmamalıdır (Perkins, 2001).

3.2.2 Veritabanı Yapısı

Öncelikle, tablolardaki kayıtlar matematiksel açıdan tuple olarak tanımlanırlar. Bir tuple tanımlanmış bir veri tipi olan bileşenlerden oluşan sıralı grup olarak tanımlanır. Tüm tuplelar aynı sayıda ve tipte bileşenlerden oluşur. İlişkisel veritabanlarında bu “küme” ya da tabloya eklenen tüm kayıtlar aynı yapıda olmalıdırlar. Ayrıca tuplelardan oluşan bir tabloda aynı veriler bulunmaz (no duplicate record). Dolayısıyla ilişkisel veritabanlarındaki herhangi bir tabloda birbiriyle tamamen aynı iki kayıt (row or record) bulunamaz. Bu, çok gereksiz bir sınırlama olarak görünebilir. Örnek vermek gerekirse, aynı kullanıcının aynı malı iki kez sipariş etmesi görünürde engellenmiştir. Bu da tabloya bir bileşen eklenerek çözülebilir (Gündüz, 2002b).

Bir kayıttaki her bir bileşen “atomik”, yani bir veri olmalıdır; başka bir kayıt ya da diğer bileşenlerin listesi olamaz. Aynı zamanda, daha önce de belirttiğimiz gibi, tablodaki bileşenlerin veri tipleri de üsttekilerle ve dolayısıyla tablo tanımlarındakilerle aynı olmalıdır. Diğer bir deyişle, veritabanı tarafından desteklenen veri tiplerinden biri olmalıdır.

Birbiriyle eş kayıtları ayırmak için kullanılan bileşenlere anahtar (key) denir. Tablodaki bir kaydı diğer tüm kayıtlardan ayırmak için kullandığımız bileşene, birincil anahtar (primary key) adı verilir. Birincil anahtar, o kaydı tekil (unique) yapar. Tüm ilişkisel veritabanlarında her bir tablo ya da ilişkide mutlaka birincil anahtar olmalıdır (Başaraner, 1997).

İlişkisel veritabanı yapısını belirleyen son bir özellik de bütünlük denetimi'dir (referential integrity). Veritabanında uygulama geliştiren programcılar yazdıkları kodun veritabanının bütünlüğünü bozmamasına dikkat etmelidirler. Uygun bir örnek bir müşteri kaydının silinmesi olabilir. Eğer müşteri tablosundan bir kayıt silinecekse, o kişinin sipariş tablosundaki siparişlerinin de silinmesi gerekir. Aksi takdirde olmayan bir müşteriye ait siparişler veritabanında kalacaktır (Gündüz, 2002b).

İlişkisel veritabanları tablolar (table), tetikler (trigger), saklı prosedürler (stored procedure) ve görünüm (view) içerirler.

3.2.2.1 Tablolar (Tables)

Veritabanı içinde bulunan veri setleri tablolarda organize edilirler. Bir tablo, bir çeşit nesne koleksiyonu içerir. Örneğin bir adresi oluşturanlar, sokak adı, şehir ve posta kodu ve benzeri nesnelerdir.

Verinin her bir parçası değer (value) olarak adlandırılır ve kendi sütununda bulunur. Tüm bir veri seti'ne (adres örneğinde bakarsak; adres, sokak adı, şehir ve posta kodu) kayıt (record) denir ve aynı zamanda satır olarak da adlandırılabilir (Perkins, 2001).

3.2.2.2 Tetikler (Triggers)

Tetikler, veritabanında yeni bir kayıt eklenmesi gibi bazı olayların oluşması ile çalıştırılan kod parçalarıdır. Tetikler, öncelikli olarak veri tabanındaki veri bütünlüğü ve güvenilirliğini korumak amacıyla kullanılırlar. Örneğin OZNITELIK tablonuza gireceğiniz OZNITELIK_ID'nin kesinlikle üç karakterden oluşmasını istiyorsanız OZNITELIK tablonuz için girilecek OZNITELIK_ID'nin kaç karakterden oluştuğunu control eden bir tetik oluşturup koyabilirsiniz. Eğer girilen OZNITELIK_ID üç karakterden küçük ya da büyükse kayıt veritabanına kabul edilmeyecektir (Perkins, 2001).

3.2.2.3 Depolanmış Süreçler (Stored Procedures)

Saklı prosedürler, bir veritabanı programcısının veritabanı içerisine bir veritabanı operasyonu koymasına ve programcıların bu operasyona SQL kullanarak erişimine izinverirler. Örneğin veritabanınız önemli ve bazı kullanıcılarınızın erişmesini istemediğiniz gizli bilgilerin kaydedildiği tablolar içeriyor, bir veritabanı yöneticisinin bu gizli tablolara ait okuma ve yazma haklarını sadece istenilen kullanıcılara tahsis etmesi ve diğer kullanıcılara bu tablodaki bilgilerin ya hiçbirine ya da bazılarına erişimine izin vermesi işlemine saklı prosedür yaratma adı verilir (Perkins, 2001).

3.2.2.4 Görünümler (Views)

Veritabanı terminolojisinde, görünüm, sadece bir tablodan değil birkaç tablonun birleşiminden oluşan bir veri seti olarak tanımlanır (Perkins, 2001).

3.2.3 Normalizasyon Kuralı

Eğer veriler normalize edilirlse, verilerin bağlı olduğu veritabanını anlamak ve kullanmak çok daha fazla kolaylaşacaktır. Veriyi normalize etmenin anlamı, onu etkili bir şekilde organize etmektir. Birkaç seviye normalizasyon katmanı vardır ve bunlardan herbiri daha yüksek derecede bir organizasyon ekler. Veritabanı terminolojisinde her bir organizasyon katmanı Normal Sınıfı olarak adlandırılır. Teknik olarak farklı normal sınıfları:

Birinci Sınıf Normal: Bir tablodaki her değer tek bir değer içermelidir ve tablo tekrarlanan veri grupları içermemelidir. (Veri, veritabanında sadece bir kez görünmelidir.)

İkinci Sınıf Normal: Veri direkt olarak tablonun birincil anahtarının (primary key) başka bir tabloya taşınmasına bağımlı değildir.

Üçüncü Sınıf Normal: Tablodaki diğer sütunlardan ya da veritabanındaki diğer tablolardan elde edilebilecek tüm alanlar (fields) kaldırılmalıdır (Perkins, 2001).

3.2.4 Veri Bütünlüğü

Veritabanımızdaki LITOLOJIK_BIRIMLER_OZNTL tablosuyla, OZNITELIK_ID, OZNITELIK_KODU ve OZNITELIK_DEGERI adlı alanlar OZNITELIKLER_LIT_BR adlı tablodan ayrılmıştır. Bunun nedeni OZNITELIKLER_LIT_BR tablosunda varolan OZNITELIK_ID ve OZNITELIK_KODU adlı alanlara, LITOLOJIK_BIRIMLER_OZNTL tablosunda bulunmayan bir veri girilmeye çalışıldığında veritabanının bu veri ve kayıtlı kabul etmemesini sağlamaktır. Örnekten de anlaşılacağı gibi. Eğer LITOLOJIK_BIRIMLER_OZNTL tablosunda bulunmayan bir veri girilseydi, bu veri hiç bir anlamlı veri sağlamayacaktı. Bunu sağlamanın birkaç yolu vardır. Bunlar, tetikler (triggers) ve kısıtlamalardır (constraints). Veri bütünlüğü veritabanının işlerliği için kesinlikle gözönünde bulundurulması gereken bir etmendir.

3.2.5 Mantıksal Veriler Arasındaki İlişkiler

Farklı mantıksal dosyalar arasındaki ilişki 1:1 (bire-bir), 1:n (bire-çok) ve n:m (çok-çok) olabilir. Fakat tablolar tasarlanırken n:m ilişkiler 1:n ilişkilere indirgenmiş olarak kullanılır. Örneğin bir parselin sahibi varsa 1:1, parselin birden çok sahibi varsa veya bir kişi birden çok parselde sahip ise 1:n ve birden fazla kişinin birden fazla parselde ortak olarak sahiplerse n:m ilişkiden söz edilir (Başoğlu, 1994; Berber, Koçyiğit, 1996).

3.3 SQL

SQL (Structured Query Language), bir veritabanı dilidir. Program geliştiricileri ve Veritabanı kullanıcıları, bir veritabanına veri eklerken, silerken, güncellerken veya sorgularken bu dili kullanırlar. Hem ANSI hem de ISO standardı olmasından dolayı, çoğu veritabanı programı bu dili kullanmaktadır.

SQL'in kullandığımız bir kaç sorgulama temel komutu vardır. Bunların en basitleri:

- SELECT (veri seçerken kullanılır)
- DELETE (veri silerken kullanılır)

- UPDATE (veri güncellerken kullanılır)
- INSERT (veri girerken kullanılır)

SELECT * FROM tablo1 ; // tablo1 adlı tablodaki tüm verileri seçer.

SELECT * FROM tablo1 WHERE adi = 'Himmet'; // tablo1 tablosundan, adi kısmında Himmet yazan tüm kayıtları getirir.

SELECT * FROM tablo1 WHERE adi LIKE '%met'; // tablo1 tablosunda, adının baş kısmı ne olursa olsun, sonu met olanları listeler.

DELETE FROM tablo1; // tablo1 tablosundaki tüm verileri siler.

DELETE FROM tablo1 WHERE adi = 'Himmet'; // Adı alanında Himmet olan tüm kayıtları siler.

UPDATE tablo1 SET adi = 'Ali'; // tablo1 tablosundaki tüm kayıtların adi alanını Ali ile değiştirir.

UPDATE tablo1 SET adi = 'Ali' WHERE adi = 'Himmet'; // tablo1 tablosundaki adi kısmında Himmet yazan tüm kayıtları Ali ile değiştirir.

INSERT INTO tablo1 (adi,soyadi) VALUES ('Himmet','Karaman'); // tablo1 tablosuna, Himmet Karaman satırını ekler ({' Himmet', 'Karaman'})

3.4 Veritabanı Yazılımları

Verileri belli bir veri tabanı modeline göre veri depolama tekniklerine uygun olarak bilgisayar ortamında depolayan ve yöneten yazılımlardır. Bir veritabanı seçmeden önce, bu veritabanı ile ne yapılacağı iyice düşünülüp kararı ona göre verilmelidir. Önemli olan veritabanının işiniz için yeterli olup olmadığıdır.

Bunun için bazı temel soruları cevaplamak yararlı olacaktır (Gündüz, 2002b):

1. Bu veritabanı ile neler yapılacak? Küçük bir şirket çalışanlarının özel bilgileri mi tutulacak, yoksa büyük bir şirketin binlerce müşterilerinin bilgileri mi?

2. Sitenizi günde kaç kişi ziyaret edecek?
3. Aynı anda kaç işlem yapılacaktır?
4. Güvenlik ne ölçüde olacaktır?
5. Verilerinizin güvenliği ne ölçüde olacaktır?

Başlıca veri tabanı yazılımları:

3.4.1 Microsoft Access

Microsoft firmasının Office paketi içinden çıkan Access, paralı veritabanları arasında nispeten ucuz olarak göze çarpar. Küçük ölçekli uygulamalardaki gereksinimleri karşılayabilir. Eğer bir web sitesinde veri miktarı ve aynı anda yapılan işlem sayıları az ise, Access kullanılabilir. Tek bir veri tablosunda 2 GB a kadar veri depolayabilir ve aynı anda 255 bağlantıya izin verilebilir. Access, MS Windows sistemlerinde kullanılamamakta, bu da yaygınlaşmasını engellemektedir. Hareket Kilitleme (Transaction locking) özelliğine sahiptir, ancak tetikleme (trigger) ve saklı prosedür (stored procedure) özelliklerine sahip değildir (Gündüz, 2002b).

3.4.2 MySQL

MySQL Inc. Tarafından kodlanan MySQL, Access ile karşılaştırıldığında daha güvenlidir. Windows'un yanı sıra Linux, OS/2, Solaris, AIX ve birçok işletim sistemini desteklemesi nedeniyle çok yaygındır. Ev kullanıcıları tarafından, kolay kurulumu ve gelen kurulum paketleri nedeniyle sıkça tercih edilmektedir. Tablo başına 8 TB veri depolayabilmektedir. MySQL' in en büyük dezavantajlarından biri ücretsiz olmasından kaynaklanan destek eksikliğidir. Özellikle web uygulamaları için çok hızlıdır. Hareketleri desteklemediği için alabildiğine yalındır ve hareket desteği olmadan gerçekleştirilebilecek internet ağı uygulamaları için çok hızlı bir alternatiftir. Oldukça fazla yazılım desteği bulunmaktadır (Gündüz, 2002b).

3.4.3 IBM DB2

IBM firmasının ürünü olan DB2, Access ve MySQL e göre daha performanslı, ancak küçük işletmelere göre daha yüksek maliyete sahiptir. Windows ve Unix sistemlerinde çalışabilir. “Hareket Kilitleme”, “tetikleme” ve “saklı prosedür” özelliklerine sahiptir (Gündüz, 2002b).

3.4.4 Informix

Ücretli ve güçlü bir veritabanıdır. Orta ölçekli işletmelerin yükünü kaldırabilecek kapasitededir. 1994’deki Postgres kodundan geliştirilmeye başlanmıştır (Gündüz, 2002b).

3.4.5 Microsoft SQL Server

Yine Microsoft firmasının bir ürünü olan Microsoft SQL Server (MSSQL), iyi bir performansa sahiptir. En büyük dezavantajı, sadece Windows üzerinde çalışabilmesidir. Kullanım kolaylığı, güvenilirliği ve işlem gücüyle dikkat çekmektedir. Maliyeti diğer veritabanlarına göre yüksektir. Tablo başına 4 TB veri depolayabilmektedir. “Hareket Kilitleme”, “tetikleme” ve “saklı prosedür” özelliklerine sahiptir (Gündüz, 2002b).

3.4.6 PostgreSQL

PostgreSQL, veritabanları için ilişkisel modeli kullanan ve SQL standart sorgu dilini destekleyen bir veritabanı yönetim sistemidir. PostgreSQL aynı zamanda iyi performans veren, güvenli ve geniş özellikleri olan bir DBMS’tir. Hemen hemen tüm UNIX ya da Unix türevi (Linux, FreeBSD gibi) işletim sistemlerinde çalışır. Ayrıca NT çekirdekli tüm Windows sistemlerde de çalıştırılabilir. Ücretsiz ve açık kodludur. PostgreSQL diğer ticari ya da açık kodlu veritabanlarında bulunan özelliklerin hemen hemen hepsini (ya da daha fazlasını) kapsar (Gündüz, 2002b).

3.4.7 Oracle

Oracle, dünyanın en güçlü ve güvenilir veritabanı olarak gösterilmektedir, ancak çok yüksek maliyeti nedeniyle sadece büyük kurumların tercih edebilecekleri bir veritabanıdır. Windows ve Unix sistemlerinde kullanılabilir. Oracle, sınırsız sayıda tabloları desteklemektedir. Çok büyük yazılım desteği vardır (Gündüz, 2002b).

3.4.8 Interbase

Linux platformu üzerinde çalışan, ücretsiz bir veritabanı sunucusudur. Yaygın değildir. Ücretsizdir (Gündüz, 2002b).

3.4.9 Progress

Linux ve Windows platformu üzerinde çalışan çok kuvvetli bir veritabanı sunucusudur (Gündüz, 2002b).

4. POSTGRESQL VERİTABANI

4.1 PostgreSQL Nedir

PostgreSQL, veritabanları için ilişkisel (relational) modeli kullanan ve SQL standart sorgu dilini destekleyen bir veritabanı yönetim sistemidir. PostgreSQL aynı zamanda iyi performans veren, güvenli ve geniş özellikleri olan bir DBMS'tir. Hemen hemen tüm UNIX ya da Unix türevi (Linux, FreeBSD gibi) işletim sistemlerinde çalışır. Ayrıca NT çekirdekli tüm Windows sistemlerde de çalıştırılabilir. Tabii ki ücretsiz ve açık kodludur. PostgreSQL diğer ticari ya da açık kodlu veritabanlarında bulabileceğiniz özelliklerin hemen hemen hepsini (ya da daha fazlasını) kapsar.

PostgreSQL'in PostgreSQL FAQ'da listelenen özellikleri:

- Hareketler (Transactions)
- Altseçimler (Subselects)
- Görünümler (Views)
- Yabancı Anahtar Bütünlük Denetimi (Foreign Key Referential Integrity)
- Gelişmiş (Karmaşık) Kilitleme (Sophisticated Locking)
- Kullanıcı-tanımlı tipler (User-defined Types)
- Miras (Inheritance)
- Kurallar (Rules)
- Çok-sürümlü koştur zamanlılık denetimi (Multi-version concurrency control)

PostgreSQL en iyi GPL (Global Public Licence) veritabanı sunucusudur (Gündüz, 2002a).

4.2 PostgreSQL'in Tarihi

PostgreSQL'in gemişı 1977'de Kaliforniya'daki Berkeley Üniversitesinde (UCB) yapılan alıřmalara dayanır. UCB'de 1977-1985 yılları arasında Ingres adı verilen ilişkişel veritabanı geliştirildi. Ingres UCB açısından oldukça verimli idi; akademik ve araştırma yapılan kurumlarda UNIX alıřtırılan sistemlerde oldukça başarılı oldu. Ticari pazara hizmet vermek amacıyla Ingres kodu Relational Technologies/Ingres Corporation tarafından satın alındı ve ilk ticari relational veritabanlarından biri oldu (Gündüz, 2002a).

Ingres CA-INGRES II adını aldı ve Computer Associates firmasının bir ürünü oldu. Orijinal UCB kodundan bugünkü modern yapıya herhangi bir kod kaldığını söylemek zordur.

Aynı süreçte, Berkeley'deki ilişkişel veritabanı sunucusu üzerindeki alıřmalar 1986-1994 arasında devam etti ve bu veritabanı Postgres adını aldı. Bu kod ise Illustra tarafından satın alındı ve Informix olarak geliştirilmeye başlandı (Momjian, 2001).

1994'te SQL özellikleri Postgres'e eklendi ve bu veritabanı Postgres95 adını aldı.

1996 yılında Postgres tanınmaya başlandı ve kod geliřtirmesi için e-posta listesi açılmasından sonra bir ok gönüllü Postgres'i geliřtirmek için alıřmaya başladı. Bu aşamadan sonra Postgres son kez adını deęiřtirdi ve adındaki "95" ekinin yerine daha uygun olan SQL konmasına karar verildi. Bunun nedeni Postgres'in artık SQL standartlarını desteklemesiydi. Böylece PostgreSQL doğdu (Gündüz, 2002a).

Bugün, bir takım halinde alıřan geliřtiriciler PostgreSQL'i Perl, Apache ve PHP gibi açık kodlu yazılımlar gibi geliřtirmektedirler. Kullanıcılar kaynak koda erişebilmekte ve açıkların kapanmasına, kodun geliřtirilmesine ve yeni önerilerle PostgreSQL'in geliřtirilmesine yardımcı olmaktadırlar. Resmi PostgreSQL sürümleri www.postgresql.org adlı internet adresinden yayınlanmaktadır (Momjian, 2001; Gündüz, 2002a).

4.3 Teknik Açından PostgreSQL

Bazı teknik özellikler açısından, PostgreSQL şunları sunar:

%100 ACID ve ANSI SQL uyumludur, replikasyon (ticari ve ticari olmayan çözümler) ana veritabanının (master) çok sayıda başka veritabanlarına (slave) çoklanmasını sağlar, ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python ve Ruby için doğal arabirimlere sahiptir, veri bütünlüğü denetimi (referential integrity), kurallar (rules), görünüm (views), tetikler (triggers), sıralamalar (sequences), miras (inheritance), dış ilişkilendirmeler (outer-joins), saklı prosedürler (stored procedures) gibi temel ilişkisel veritabanı özelliklerini barındırır. Kod geliştiriciler için açık API ve Doğal SSL desteği sağlar, UNION, UNION ALL ve EXCEPT sorgularına destek ve doğal kerberos yetkilendirmesi yapılabilir, fonksiyonel ve parçalı (partial) indeksler, prosedürel diller, MD5, SHA1, XML ve diğer işlevsel özellikler yüklenebilir. Diğer SQL-uyumlu sistemlerle paylaşabilmek için taşınabilir SQL yaratan araçlara sahiptir. Özelleştirilmiş, kullanıcı-tanımlı veri tipleri için geliştirilebilir veri tipi sistemi ve hızla gelişen yeni veri tipleri içerir. Daha az SQL-uyumlu ilişkisel veritabanı yönetim sistemlerinden (RDBMS) geçişi kolaylaştıran cross-database uyumluluk fonksiyonları içerir. Birden fazla işlemcide rahatlıkla çalışabilir. Çok kullanıcıli sistemlerde daha hızlı erişim sağlar. Son derece güvenli bir altyapıda çalışır (Linux çekirdeği) ve ODBC desteği sayesinde microsoft ürünleri ile veri aktarımı yapabilir. Ticari bir veritabanı olmadığı için, hızdan ziyade güvenliğe önem verir (Gündüz, 2002a).

4.3.1 Desteklenen Platformlar

PostgreSQL Veritabanı tarafından desteklenen platformlar aşağıda verilmiştir.

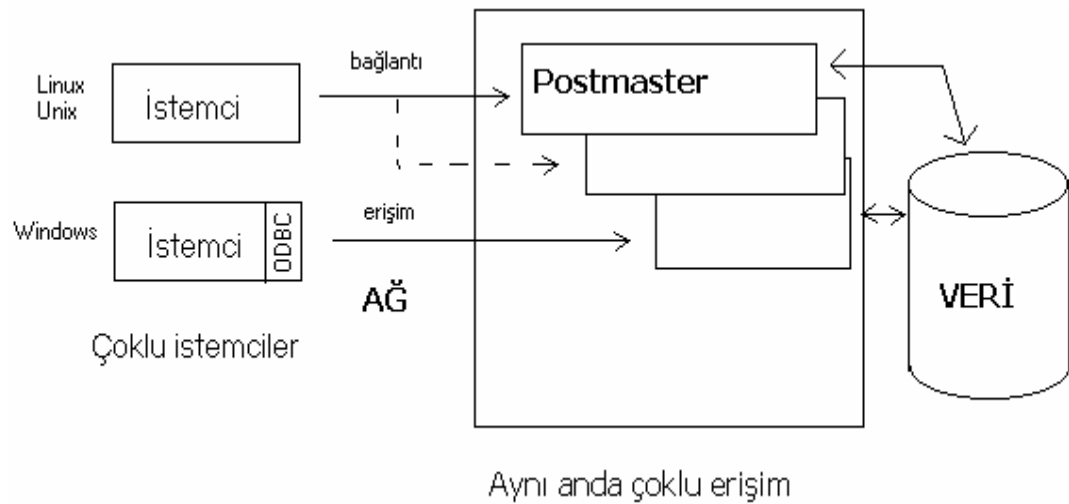
- | | |
|---------------------------------|---------------------|
| - Linux (Kernel 2.0.X ve üzeri) | - SCO OpenServer 5 |
| - AIX 4.3.2 | - SCO Unix Ware 7 |
| - HP – UX 9.0x ve 10.20 | - Solaris 2.5.1-2.7 |
| - FreeBSD 4.X | - Sun OS 4.1.14 |
| - IRIX 6.5.6f | - WinNT/Cygwin |

- MacOS-X Darwin
- NetBSD 1.4, 1.4u
- QNX 4.25
- BSDI 4.0.1
- BeOS 5.0.3

4.3.2 PostgreSQL Mimarisi

PostgreSQL'in gücü, onun mimarisinden gelir. Ticari veritabanı sistemleri ile ortak olarak PostgreSQL sunucu-istemci ortamında kullanılabilir. Bu hem kullanıcılar hem de geliştiriciler açısından oldukça fazla yarar sağlar.

PostgreSQL kurulumunun kalbi veritabanı sunucu işlemidir (process). Postmaster olarak adlandırılır .Tek bir sunucu üzerinde çalışabilir. Veritabanındaki bilgilere erişebilecek programlar sunucu tarafında çalışır. İstemci tarafındaki programlar, sunucu ile aynı makinede olsalar bile veriye direk olarak erişemezler. İşte bu istemci-sunucu ayrımı uygulamaların ayrı ayrı makinelerde çalışmasına izin verir. İstemcileri sunucudan ayırmak için bir ağ kurulabilir, ve istemci uygulamalarını geliştirmek için kullanıcılara uygun bir ortam kullanılabilir. Örnek olarak, veritabanı UNIX ortamında çalıştırılıp ve istemci programları Microsoft Windows'da kodlanabilir.



Şekil 4.1 Tipik PostgreSQL Mimarisi

Şekil 4.1’de tipik olarak dağıtılmış bir PostgreSQL uygulamasının şeması görülebilir. Bir ağ ortamında sunucuya bağlanan bir çok istemci vardır. PostgreSQL için bunun bir TCP/IP ağı, yerel ağ ya da internet olması gerekmektedir (Gündüz, 2002a).

Her bir istemci, bu istemciden gelen erişim isteklerine özel olarak servis yapmak için yeni bir sunucu işlemi yaratan bir ana veritabanı sunucu işlemine (burada postmaster olarak gösterilir) bağlanır.

İstemci programları PostgreSQL’e özel bir mesaj protokolu kullanarak bağlanırlar. İstemciye uygulamanın çalışması için standart bir arayüz sağlayan yazılım kurmak mümkündür. Bu Open DataBase Connection (ODBC) standartıdır. Bu, Access ve Excel gibi Microsoft Office ürünlerini de içeren programların PostgreSQL’i veritabanı olarak kullanmalarına olanak sağlar.

PostgreSQL’in istemci-sunucu mimarisi, iş gücünün bölünmesine yardımcı olur. Büyük miktarda veriyi tutabilecek ve erişebilecek şekilde düzenlenmiş bir sunucu makinesi güvenli bir veri deposu olarak kullanılabilir. Gelişmiş grafiksel uygulamalar istemciler için geliştirilebilir. Alternatif olarak, internet ağı tabanlı uygulamalar da istemci tarafındaki işlemleri görebilir (Gündüz, 2002a).

4.3.3 PostgreSQL’in Sınırlamaları

Bilgileri saklamak için tablolar yaratıp onlara veri ekleyerek veritabanı kullanıldığında, hiç bir platformda sınırsız veri saklama lüksümüz olmadığı ortadadır. Tüm veritabanı sistemleri bir şekilde sınırlandırılmıştır.

PostgreSQL bir veritabanının toplam büyüklüğü için herhangi bir sınır koymaz. Şu anda bilinen 60 GB’lık bir veritabanı vardır. PostgreSQL’in veriyi düzenleme yönteminden dolayı çok fazla tablo içeren veritabanlarında başarımlar gittikçe düşer. PostgreSQL veriyi saklamak için çok sayıda dosya kullanacaktır, ve işletim sistemi tek bir dizinde bu kadar çok dosyayı yönetemezse, başarımlar düşecektir.

PostgreSQL normalde tablo verilerini 8k’lık parçalarda tutar. Bu blokların sayıları 32-bit signed integer kadar sınırlıdır (2 milyarın hemen üstü) ve 16 TeraByte kadar bir tablo büyüklüğü sağlar. Temel blok büyüklüğü PostgreSQL kurulurken 32k ya kadar yükseltilebilir ve bu da teorik olarak 64 TB’lık bir sınır getirir.

Bazı işletim sistemleri dosya büyüklükleri için bir sınır koyarlar. Bu nedenden, PostgreSQL tablo verilerini her biri en fazla 1GB büyüklükte olabilecek çoklu dosyalarda tutar. Büyük tablolar için bu bir çok dosya anlamına gelecek ve daha önce de belirtildiği gibi sistem başarımının düşmesine neden olacaktır. Bu büyüklük işletim sisteminden bağımsızdır (Gündüz, 2002a).

Tablo üzerinde yaratılabilecek indexlerde PostgreSQL tarafından konan herhangi bir limit yoktur. Ancak unutulmaması gereken, oldukça fazla kolon içeren bir tabloda çok fazla index yaratma çalışırsak başarımlarımız gittikçe düşecektir.

PostgreSQL , sürüm 7.1 ve sonrasında bir tablodaki herhangi bir field için 1 GB'lık bir sınır getirmiştir. Pratikte bu limit sunucunun veriyi işleme ve istemciye transfer etmesi için gerekli hafıza miktarından gelir.

PostgreSQL'de tutulabilecek en fazla kolon sayısı, configure edilmiş blok büyüklükleri ve kolon tiplerine bağlıdır. Varsayılan değer olarak blok büyüklüğü olan 8k'da en az 250 kolon saklanabilir, bu sayı eğer fieldlar oldukça basit ise (tamsayı değerleri gibi) 1600'e kadar çıkabilir. Blok büyüklüğünü arttırmak eş zamanlı olarak bu limitleri de arttırır. Bir satırın büyüklüğü için bir sınır yoktur (Gündüz, 2002a).

4.4 PostgreSQL Veri Tipleri

Sayılar, karakter dizgileri ya da zamanlar gibi her tip bilgiyi tutabilen sadece tek bir tip verinin olduğu bir veritabanının kullanımının ne kadar kolay olabileceğini düşünmek oldukça cezbedicidir. Buna rağmen, her ne kadar tek bir veri tipi tablo yaratmayı kolaylaştırıcaksa da, farklı veri tiplerinin olması belirli avantajlar sunar:

Tutarlı sonuçlar; örneğin veri tipi INTEGER olan bir sütundan yapılan bir seçme işleminin sonucu her zaman bir INTEGER olacaktır.

Veri Onayı; örneğin veri tipi INTEGER olan bir sütun girilmeye çalışılacak her DATE değerini reddedecektir.

Yoğun Saklama; tek tipli sütunlar daha yoğun veri saklar.

Performans; tek tipli sütunlar daha çabuk işlem yaparlar.

Bu nedenlerden dolayı ilişkisel veritabanlarındaki her sütun sadece tek bir tip veri tutabilir. Bir sütun içinde farklı veri tipleri olamaz (Momjian, 2001)

PostgreSQL'deki veri tipleri aşağıdaki Tablo 4.1'de verilmiştir.

Tablo 4.1 PostgreSQL Veri Tipleri

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Mantıksal (Logical)	Boolean, bool	bool	Gerçeklik değeri tutar. Değerleri, doğru (TRUE), 't', 'true', 'y', 'yes', 1, ya da yanlış (FALSE), 'f', 'false', 'n', 'no', 0, olarak kabul eder.

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Tamsayı (Exact Number)	Smallint	int2	İki-byte'lık işaretli tamsayıları tutar. - 32768'den +32767'ye kadar olan tamsayılar girilebilir.
	integer, int	int4	Dört-byte'lık işaretli tamsayıları tutar. - 2147483648'den +2147483647'ye kadar olan tamsayılar girilebilir.
		int8	Sekiz-byte'lık işaretlenmiş tamsayıları tutar. Yaklaşık olarak 18 haneli presizyon verir.
	Bit	Bit	Tek bir bit tutar, 0 ya da 1.
	bit varying	varbit	Bir sıra bit tutar. Bir tabloya bu tip bir veri girişi yapabilmek için kullanılması gereken syntax; INSERT INTO ... VALUES(011101::varbit);

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Yaklaşık Sayı (Approximate Number)	Numeric (precision, scale)	Numeric (precision, scale)	Belirlenen presizyonda tamsayıları tutar. Presizyon sınırı yoktur.
	decimal (precision, scale)	decimal (precision, scale)	Olağan halinde presizyon 9, ölçek 0'dır. Sınırı yaklaşık olarak 8000 digittir.
	float (precision)	float4, float8	Verilen presizyondaki değişken sayılardır. İstenen presizyon 7'den küçükse FLOAT4, büyükse FLOAT8 kullanılmalıdır. Sağladığı maksimum presizyon 15 digittir, FLOAT(15).
	Real	Float4	float(precision).
	double precision	Float8	float(15) ile aynı.
		Money	decimal(9,2) ile aynıdır. Kullanımı engellenmiştir.

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Zamansal (Temporal)	Timestamp	timestamp, datetime	M.Ö.4713'den M.S.1465001'e kadar 1 microsecond çözünürlükle zamanı tutar. Formatı: YYYY-AA-GG SS-DD:SS+0X X Greenwich zamanına bağlı olarak elde edilir.
	timestamp with timezone	timestamp with timezone	M.Ö.1903'den M.S.2037'ye kadar 1 microsecond çözünürlükle zamanı tutar.
	Interval	interval, timespan	Yaklaşık olarak +/- 178000000 yıllık aralığı, 1 microsecond çözünürlükle tutar.
	Date	Date	Zamanı M.Ö.4713'den M.S.32767'ye kadar 1 gün çözünürlükle tutar.
	Time	Time	Günün saatini 0'dan 23:59:59.99'a kadar 1 microsecond çözünürlükle tutar.
	time with timezone	Time with timezone	Time ile aynıdır, farkı timezone'un da tutulmasıdır.

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Karakter (Character)	Char	Char	Tek bir karakter tutar.
	char(n)	Char(n)	Daha az karakter girişi boşluklarla doldurulacak şekilde, tam olarak n karakter tutar. Uzunluğu bilinen kısa dizgiler için tavsiye edilir.
	varchar(n), char varying(n)	varchar(n)	Az sayıda karakter girişinde kalan kısmın boşlukla doldurulmadığı, maksimumu n sayıda olmak üzere, değişken sayıda karakterin tutulduğu veri tipidir. Karakter dizgileri için standart seçimdir.
		Text	PostgreSQL'in özel varchar değişkenidir. Karakter sayısında bir maksimum limit belirlemeyi gerektirmez.

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar								
Geometrik (Geometric)		point	(x, y) koordinatları								
		Line	Bir çift nokta (Sınırsız hat)								
		lseg	Bir çift nokta (Sınırlı hat)								
		box	Bir dikdörtgenin ters köşe noktaları								
		path	Kapalı ya da açık bir sıra nokta <table><tr><td>path</td><td>4+32n bytes</td><td>((x1,y1),...)</td><td>Kapalı rota (polygon ile aynı)</td></tr><tr><td>path</td><td>4+32n bytes</td><td>[(x1,y1),...]</td><td>Açık rota</td></tr></table>	path	4+32n bytes	((x1,y1),...)	Kapalı rota (polygon ile aynı)	path	4+32n bytes	[(x1,y1),...]	Açık rota
	path	4+32n bytes	((x1,y1),...)	Kapalı rota (polygon ile aynı)							
	path	4+32n bytes	[(x1,y1),...]	Açık rota							
	polygon	Bir dizi nokta, fiili olarak kapalı bir rota, Fakat PostgreSQL'de farklı ele alınmıştır.									
	circle	Bir daire belirten bir nokta ve bir uzunluk (yarıçap)									

Veri Tipi	SQL Adı	PostgreSQL Adı	Notlar
Çeşitli (Variable)	Serial	[uses an integer]	Standart SQL’de serial bir tabloda her yeni bir satır eklendiğinde artan numeric bir sütundur. PostgreSQL standart SQL syntaxını kabul etmesine rağmen serial tipi ayrı bir tip olarak görmez. PostgreSQL bir değeri bu tipte tutmak için integer veri tipini ve otomatik olarak değeri artış ya da azalışını sağlamak için bir sequence kullanır. Menzili 0’dan +2147483647’ye kadardır.
		Oid	Bir nesne (object) id’sidir. PostgreSQL her satıra gizli bir oid ekler ve 4 byte’lık bir integer saklar, maksimum değeri yaklaşık olarak 4 milyardır.
		Cidr	Y’nin netmask olduğu x.x.x.x/y formunda network adresi saklar. CIDR, classless inter-domain routing’in kısaltmasıdır.
		Inet	host kısmının 0 olabilmesi dışında cidr ile aynıdır.
		macaddr	XX:XX:XX:XX:XX:XX formunda MAC adresidir

4.5 PostgreSQL Kurulumu

PostgreSQL kurulumu UNIX tabanlı işletim sistemlerinde iki şekilde gerçekleştirilebilir. Bunlar;

1. Kaynak Koddan Kurulum
2. RPM’den Kurulum (Binary Installation)

şeklinde. Kaynak koddan (source code) kurulum, RPM’den kurulumla göre daha zahmetli olsa da kurulan programın hangi parçasının hangi dosyada olduğunun bilinmesi ve istenilen parametrelerin değiştirilme şansının olması bu dezavantajı ortadan kaldırır. Kaynak koddan kurulum Unix tabanlı sistemlerde yaygındır ve bu işlemde, kaynak kod derlenerek paket kurulur.

Bu çalışmada PostgreSQL kaynak koddan değil RPM’den kurulmuştur. RPM’den tüm özellikleri ile çalışan bir PostgreSQL veritabanını kurabilmek için Tablo 4.2’deki paketler gerekmektedir.

Paketleri kurarken, aynı revision leveldaki yani aynı sürüme ait paketlerin kullanılması önerilir. Paketleri kurmak için, RPM paket yönetim uygulaması kullanılabilir (Gündüz, 2002a).

Tablo 4.2 PostgreSQL için RPM Kurulum Paketleri	
Postgresql	Ana paket
postgresql-libs	Library dosyaları, v7.1.0 ve sonrasını kurarken önemlidir.
postgresql-devel	Development için gereken dosya ve kitaplıklar.
postgresql-jdbc	PostgreSQL için Java database connectivity
postgresql-odbc	PostgreSQL için Open database connectivity
postgresql-perl	Perl için PostgreSQL arayüzü
postgresql-python	Python için PostgreSQL arayüzü
postgresql-server	Bir sunucuyu yaratmak ve çalıştırmak için gerekli programlar
postgresql-tcl	Tcl için PostgreSQL arayüzü
postgresql-test	PostgreSQL test suite
postgresql-tk	PostgreSQL için Tk kabuğu ve Tk-tabanlı GUI
Postgresql-contrib	PostgreSQL ile dağıtılan contributed source.

Öncelikle postgresql-libs paketini kurulmalıdır:

```
[root@localhost root]#rpm -i postgresql-libs-7.3PGDG.i686.rpm
```

Bu işlemin root iken yapılması gerekmektedir. **GnoRPM** (Gnome ile gelir) ya da **kpackage** (KDE ile gelir) uygulamalarını da rpmleri kurmak için kullanılabilir. Bu komut, paketi açarak içindeki dosyaları gerekli yerlere yerleştirecektir. Ardından postgresql paketi kurulmalıdır. Bu paketler, gereken minimum paketlerdir. Bunların dışındakiler ise kullanım gereksinimlerine göre kurulabilir. Örneğin, PostgreSQL sunucusu kuracaksanız, postgresql-server paketinizi sisteminize kurmanız gerekir (Gündüz, 2002a). Bu çalışmada PostgreSQL'e ait tüm paketler kurulmuş ve kullanılmıştır. Özellikle PostgreSQL sunucusu kurulumda anlatıldığı gibi bir Linux işletim sistemli bilgisayara sunucu olarak kurulmuş, hem windows hem de Linux işletim sistemlerine sahip bilgisayarlara sunucu özelliklerini sağlamıştır.

PostgreSQL kurulumunda bir standart bulunmamaktadır. PostgreSQL kurulumu uygulamalar (applications), yardımcı programlar (utilities) ve veri dizinlerinden (data directories) oluşur. Ana PostgreSQL uygulamaları (postmaster ve postgres) istemcilerden veri erişimini sağlayan servislerin sunucu tarafındaki kodunu içerirler. pg_ctl gibi yardımcı uygulamalar sunucunun aktif olduğu tüm anlardaki ana sunucu işlemlerini (master server processes) kontrol etmekte kullanılır. Data dizini ise PostgreSQL tarafından bir veritabanı için gereken tüm veriyi, kayıtları, tabloları ve sistem parametrelerini tutmak için kullanılır (Gündüz, 2002a).

Tipik bir PostgreSQL kurulumu tüm bu bileşenleri bulundurur. PostgreSQL, genel olarak /usr/local/pgsql dizinine kurulur. Bu dizin, kaynak koddan kurarken varsayılan dizindir. RPM’den kurarken ise /var/lib/pgsql dizini kullanılır. Verimlilik ve yönetim kolaylığı açısından, değişik kategorilerdeki dosyalar değişik yerlere konulabilir. PostgreSQL bize bu esnekliği sağlamaktadır. Örnek olarak, SuSE ve RedHat’ta PostgreSQL uygulamaları /usr/bin, log dosyası /var/log/postgresql, veri ise /var/lib/pgsql/data dizinlerine yerleştirilir. Bu, özellikle yedek alırken işe yarar. Gereksiz dosyaların yedeğinin alınması durumunda kalınmaz (Gündüz, 2002a).

4.5.1 PostgreSQL’i Başlatma

PostgreSQL’i RPM den kurduğunuzda kaynak koddan kurulumu göre çoğu işlemi yapmanıza gerek kalmaz: postgres kullanıcısı, data dizini vb yaratılır ve başlatma betiği oluşturulur. Sadece ntsysv ile sisteminiz her başladığında PostgreSQL’in başlamasını sağlamanız gerekir.

İlk çalıştırma sırasında sıfırdan başlatma işlemi yapılacaktır. Dolayısıyla initdb ile veritabanının başlatılmasına gerek kalmayacaktır.

İki kurulumda da pg_hba.conf dosyasının düzenlenmesi aynı şekildedir.

Öncelikle PostgreSQL’ in initialize işlemi için sunucunun başlatılması gerekir (Şekil 4.2).

```
[root@localhost root]# /etc/rc.d/init.d/postgresql start
Initializing database: [ OK ]
Starting postgresql service: [ OK ]
```

Şekil 4.2 PostgreSQL Sunucusunun Başlatılması

Ardından /var/lib/pgsql/data dizinine geçilir. Az önceki init işleminden sonra bu dizinde oluşacak dosyalar Şekil 4.3’de gösterimiştir.

```
[root@localhost root]# cd /var/lib/pgsql/data/
[root@localhost data]# ls
base global pg_hba.conf pg_ident.conf PG_VERSION pg_xlog
postgresql.conf postmaster.opts postmaster.pid
```

Şekil 4.3 initialization işleminden sonra oluşan dosyalar

Buradaki postgresql.conf dosyası Şekil 4.4’de gösterildiği gibi düzenlenmelidir.

```
tcpip_socket = true
max_connections = 32 # 1-1024
port = 5432
#           Performance
sort_mem = 512
shared_buffers = 2*max_connections # min 16
```

Şekil 4.4 postgresql.conf dosyasının düzenlenmesi

postgresql.conf dosyası Şekil 4.4’de gösterildiği gibi düzenlendikten sonra PostgreSQL veritabanı hazırdır. İlk olarak yapılması gereken PostgreSQL sunucusu olan postmaster’ı çalıştırmaktır. Postmaster’ı çalıştırmak için en kolay yol bir kabuk açıp, root kullanıcısından kurulumda yaratılan postgres kullanıcısına geçmek ve ardından postmaster’ı “postmaster -i” komutu ile başlatmaktır. Bundan sonra PostgreSQL kullanıma hazır olacaktır.

4.6 PostgreSQL Araçları

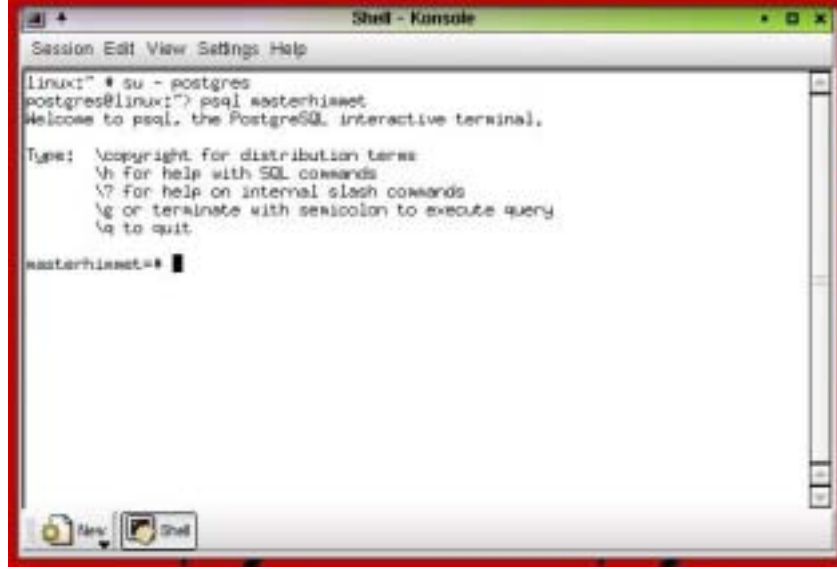
4.6.1 psql

Oracle’daki SQL*PLUS gibi PostgreSQL’de psql adında komut satırı aracı vardır. PostgreSQL veritabanları genellikle bu uygulama tarafından yaratılır ve yönetilir.

psql : psql [seçenekler] [veritabanı_adı [kullanıcı_adı]]

biçiminde çalışır.

Psql bağlanmak istenen veritabanı adı verilerek çalıştırılır. Sunucunun adını, veritabanının dinlediği port numarasını ve bağlantı için geçerli bir kullanıcı adı ve şifresinin bilinmesi gerekmektedir. Basit bir bağlantı aşağıdaki gibidir (Şekil 4.5).

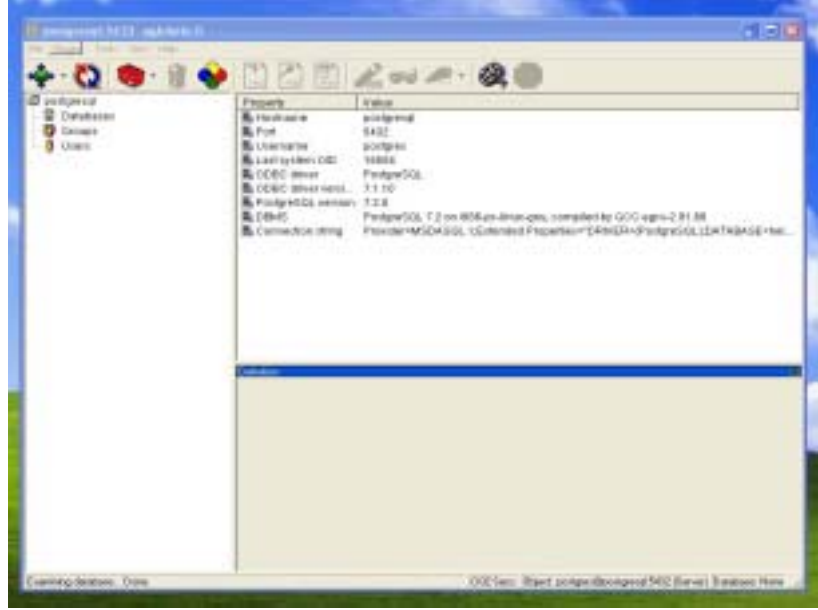


Şekil 4.5 psql ile veritabanına bağlantı

Varsayılan veritabanı, kullanıcı adı, sunucu makine adı ve dinlenen port numarası sırasıyla PGDATABASE, PGUSER, PGHOST ve PGPORT çevre değişkenlerinin ayarlanması ile değiştirilebilir. Bu varsayılan değerler yine sırasıyla psql'e -d, -U, -h ve -p seçeneklerini geçirerek değiştirilebilir (Gündüz, 2002a).

4.6.2 PgAdmin II

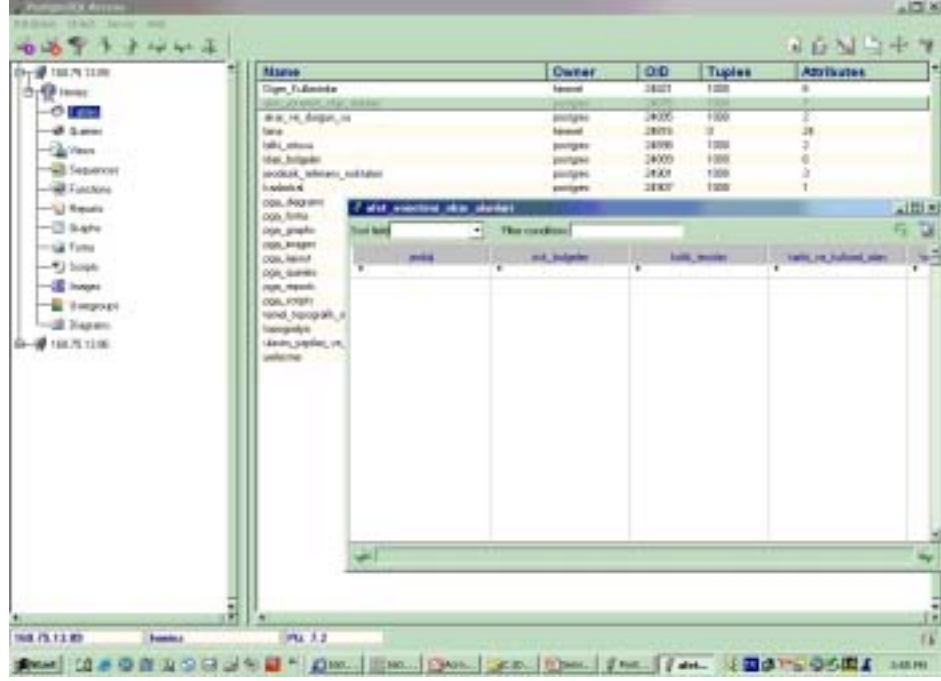
PgAdmin, PostgreSQL için Windows arayüzüdür (Şekil 4.6). Oldukça geniş bir kullanıcı kitlesi bulunmaktadır. Ücretsizdir. PgAdmin, PostgreSQL sunucusu ile iletişim kurabilmek için, <http://odbc.postgresql.org/> adresinden indirilebilecek PostgreSQL ODBC Driver (psqlODBC) kullanır. Tüm PostgreSQL nesne tipleri bu program ile yaratılabilir, drop edilebilir ve düzenlenebilir (Gündüz, 2002a).



Şekil 4.6 PgAdmin II giriş ekranı

4.6.3 Pgaccess for Windows

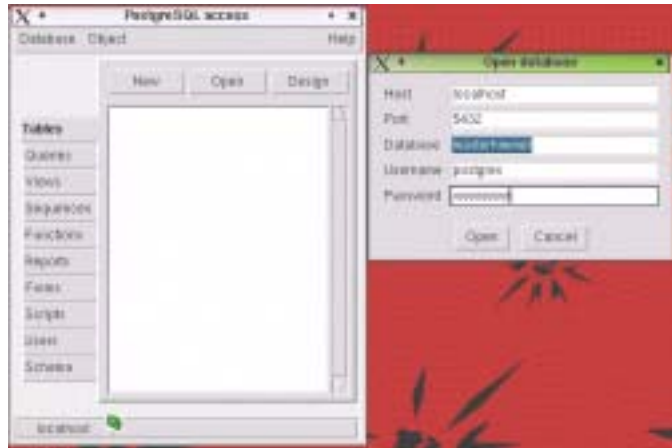
PostgreSQL için Windows arayüzüdür. Tcl/Tk dilleri ile yazılmıştır. Suse Linux 8.0 işletim sistemi ile birlikte gelen pgaccess arayüzünün windowsa uyarlanmış bir versiyonudur. Görsel ve işlevsel açıdan PgAdmin II'ye oldukça benzemektedir (Şekil 4.7). Ağ ve internet üzerinden Unix ya da Windows tabanlı bilgisayarlardaki PostgreSQL sunucularına bağlanıp kullanıcı yetkileri çerçevesince işlem yapılmasına olanak tanır.



Şekil 4.7 pgaccess for Windows ekranı

4.6.4 pgaccess

PostgreSQL için Linux arayüzüdür. Suse Linux 8.0 işletim sistemi ile birlikte gelir. Küçük ve kısıtlı kullanım özelliklerine sahip bir ekranı vardır (Şekil 4.8). Windows için olan sürümü gibi ağ veya internet üzerinden PostgreSQL sunucularına bağlanma imkanı tanır.



Şekil 4.8 pgaccess ile veritabanına bağlanma

Bu çalışmada yazıda tanıtılan tüm PostgreSQL araçları kullanılmıştır.

5. UYGULAMA

5.1 Afet Yönetimi Obje Modelinin Hazırlanması

Bu çalışmanın amaçlarından biri olan Afet Yönetimi Bazlı Obje Modelinin oluşturulması aşaması, “Ulusal UAS-CBS Bazlı Veri Tabanı ve Afet Yönetimi Odaklı Karar Destek Sistemi Standardının Oluşturulması Projesi” kapsamında geliştirilmiş olan, Türkiye Afet Bilgi Sistemi’nin (TABİS) Obje Kataloğu olan TABİS-OK’nın Afet Yönetimi Obje Alanları Temel Obje Alanı temel alınarak gerçekleştirilmiştir (Karaman, 2002).

Afet Yönetimi Obje Modelinin hazırlanması için öncelikle bir kampüste TABİS projesinin oluşturulması için gerekli olan ölçeklendirme işlemi yapılmıştır. Bu ölçeklendirme işlemi, obje kataloğunun afet yönetimine dayalı kısmı olan Afet Yönetimi Obje Alanları Kataloğunda, İ.T.Ü. Ayazağa Kampüsü bazında olan ve olmayan obje alanları, obje grupları ve obje türleri araştırılmıştır. Yapılan araştırma sonucunda mevcut Obje Alanları ve Obje Gruplarından aşağıda liste halinde verilen ekleme ve çıkarmalar yapılmıştır. Bu düzeltmeler sonucunda İ.T.Ü. Ayazağa Kampüsüne yönelik Afet Yönetimi Obje Modeli oluşturulmuştur.

5.1.1 TABİS-OK Afet Yönetimi Obje Alanlarından Çıkartılan Elemanlar

Obje kataloğundan yapılan eksiltmeler sadece bu çalışmada yapılandırılan veritabanının optimizasyonu için yapılmıştır. Böylece oluşturulan veritabanında oluşabilecek boş ve anlamsız veri fazlalığı önlenmiştir. Obje kataloğundan, Obje Alanı, Obje Grubu ve Obje Türü eksiltmeleri Tablo 5.1’deki Tabis Obje Kataloğu Afet Yönetimi Obje Modeli, Obje Alanları ve Obje Türleri İndeksinde Üzerleri çizili olarak gösterilmiştir. Yapılan öznitelik elemeleri metinde belirtilmiştir.

B.B. Risk Bölgeleri Obje Alanı altındaki , B.B.1. Doğal Afet Riski Obje Grubunun, B.B.1.01. Çığ ve B.B.1.17. Yanardağ Patlaması Obje Türleri ve B.B.4. Akut Afet Yeri Obje Grubunun, B.B.4.01. Akut Afet Yeri adlı Obje Türünün, AFT kısaltmalı Afet Türü özniteliğinden 101 Çığ, 118 Köprü Yıkılması ve 119 Akarsu Kıyısı Erozyonu öznitelik değerleri çıkarılmıştır.

Tablo 5.1 Tabis Obje Kataloğu Afet Yönetimi Obje Modeli, Obje Alanları ve Obje Türleri İndeksi

TABİS Obje Kataloğu		Versiyon 3.2: 20/10/2002
Obje Alanları ve Obje Türleri İndeksi		
Kodu:	Adı:	Türü:
B.	Afet Yönetimi Obje Alanları	Temel Obje Alanı
B.A.	Jeoloji	Obje Alanı
B.A.0	Jeolojik Objeler	Obje Grubu
B.A.0.01	Litolojik birimler	Obje Türü
B.A.0.02	Jeofizik	Obje Türü
B.A.0.03	Geoteknik	Obje Türü
B.A.0.04	Sismik aktivite	Obje Türü
B.A.0.05	Geçmiş Deprem Merkezleri	Obje Türü
B.A.0.06	Deformasyon hızı	Obje Türü
B.A.0.07	Fay	Obje Türü
B.A.0.08	Kıvrımlar	Obje Türü
B.B.	Risk Bölgeleri	Obje Alanı
B.B.1	Doğal Afet Riski	Obje Grubu
B.B.1.01	Çığ	Obje Türü
B.B.1.02	Deprem	Obje Türü
B.B.1.03	Dolu	Obje Türü
B.B.1.04	Soğuk Hava, Don	Obje Türü
B.B.1.05	Fırtına, Kasırga	Obje Türü
B.B.1.06	Hortum	Obje Türü
B.B.1.07	Kar	Obje Türü
B.B.1.08	Kaya Yuvarlanması	Obje Türü
B.B.1.09	Kuraklık	Obje Türü
B.B.1.10	Orman Yangını	Obje Türü
B.B.1.11	Sel	Obje Türü
B.B.1.12	Sıcak Hava	Obje Türü
B.B.1.13	Sis	Obje Türü
B.B.1.14	Şiddetli Yağmur	Obje Türü
B.B.1.15	Toprak Kayması	Obje Türü
B.B.1.16	Zararlı Böcek İstilas	Obje Türü
B.B.1.17	Yanardağ Patlaması	Obje Türü
B.B.2	Teknolojik Afet Riski	Obje Grubu
B.B.2.01	Bulaşıcı hastalık	Obje Türü
B.B.2.02	Trafik kazası	Obje Türü
B.B.2.03	Gaz zehirlenmesi	Obje Türü
B.B.2.04	Gıda zehirlenmesi	Obje Türü
B.B.2.05	Hayvan hastalıkları	Obje Türü
B.B.2.06	Bina Yangını	Obje Türü

B.B.3	Politik Afet Riski	Obje Grubu
B.B.3.01	Terörist Saldırı	Obje Türü
B.B.4.	Akut Afet Yeri	Obje Grubu
B.B.4.01	Akut Afet Yeri	Obje Türü
B.C.	Kritik Tesisler	Obje Alanı
B.C.0.01	Hastane	Obje Türü
B.C.0.02	Diğer Sağlık Kurumu	Obje Türü
B.C.0.03	Eczane, Eczane Deposu, Sağlık Malzemesi Deposu	Obje Türü
B.C.0.04	Bayındırlık İl Müdürlüğü	Obje Türü
B.C.0.05	Karayolları Teşkilatı	Obje Türü
B.C.0.06	Devlet Su İşleri Teşkilatı	Obje Türü
B.C.0.07	Köy Hizmetleri Teşkilatı	Obje Türü
B.C.0.08	Demiryolları Teşkilatı	Obje Türü
B.C.0.09	Türkiye Denizcilik İşletmeleri	Obje Türü
B.C.0.10	Hava Limanı	Obje Türü
B.C.0.11	İnşaat Şirketi	Obje Türü
B.C.0.12	İtfaiye Teşkilatı	Obje Türü
B.C.0.13	Emniyet Birimleri	Obje Türü
B.C.0.14	Acil Durum Yönetim Merkezleri	Obje Türü
B.C.0.15	Sığınak	Obje Türü
B.C.0.16	Kızılay	Obje Türü
B.C.0.17	Sivil Toplum Kuruluşu	Obje Türü
B.C.0.18	Basın Yayın Kuruluşu	Obje Türü
B.C.0.19	Orman Teşkilatı	Obje Türü
B.C.0.20	Çevre İl Müdürlüğü	Obje Türü
B.C.0.21	Enerji Dağıtım ve İlgili Kurum	Obje Türü
B.C.0.22	Müftülük	Obje Türü
B.C.0.23	Haberleşme Kurumu	Obje Türü
B.C.0.24	Valilik	Obje Türü
B.C.0.25	Belediye	Obje Türü
B.C.0.26	Sivil Savunma Teşkilatı	Obje Türü
B.C.0.27	Ekmek Fırını	Obje Türü
B.C.0.28	Market	Obje Türü
B.C.0.29	Gıda Deposu	Obje Türü
B.C.0.30	Tehlikeli Madde Depolama Tesisi	Obje Türü
B.C.0.31	Hububat Deposu	Obje Türü
B.C.0.32	İnşaat Malzeme Deposu	Obje Türü
B.C.0.33	Su Deposu	Obje Türü
B.C.0.34	Okul	Obje Türü
B.C.0.35	İbadethane	Obje Türü
B.C.0.36	Garnizon	Obje Türü
B.C.0.37	Banka	Obje Türü
B.C.0.38	Cezaevi	Obje Türü
B.C.0.39	Otogar	Obje Türü
B.C.0.40	Gar, İstasyon	Obje Türü
B.C.0.41	Uçak Terminali	Obje Türü
B.C.0.42	Helikopter Pisti	Obje Türü
B.C.0.43	İş Makinası Deposu	Obje Türü
B.C.0.44	Büyük Mutfak Ünitesi	Obje Türü
B.C.0.45	Sosyal Barınma Tesisi	Obje Türü
B.C.0.46	Klima İstasyonu	Obje Türü
B.C.0.47	Olabilecek Diğer Tesisler	Obje Türü
B.D.	Tarihi ve Kültürel Alanlar	Obje Alanı
B.D.0	Tarihi ve Kültürel Alanlar	Obje Grubu

B.D.0.01	Dini Yapılar	Obje Türü
B.D.0.02	Eğitim Yapıları	Obje Türü
B.D.0.03	Su Yapıları	Obje Türü
B.D.0.04	Ticaret Yapıları	Obje Türü
B.D.0.05	Sanayi Yapıları	Obje Türü
B.D.0.06	Konutlar	Obje Türü
B.D.0.07	Konaklama Yapıları	Obje Türü
B.D.0.08	Kamu Yapıları	Obje Türü
B.D.0.09	Kuleler	Obje Türü
B.D.0.10	Surlar	Obje Türü
B.D.0.11	Heykeller ve Anıtlar	Obje Türü
B.D.0.12	Müzeler	Obje Türü
B.E.	Toprak Yapısı ve Erozyon	Obje Alanı
B.E.0	Toprak Yapısı ve Erozyon	Obje Grubu
B.E.0.01	Arazi Kullanma Kabiliyeti Sınıfları	Obje Türü
B.F.	Çevre Kirliliği	Obje Alanı
B.F.1	Su Kirliliği	Obje Grubu
B.F.1.01	Numune Alma Noktası	Obje Türü
B.F.2	Hava Kirliliği	Obje Grubu
B.F.2.01	Hava Kalitesi Ölçme Noktası	Obje Türü
B.F.3	Gürültü Kirliliği	Obje Grubu
B.F.3.01	Gürültü Ölçme Noktası	Obje Türü
B.G.	Planlar	Obje Alanı
B.G.1	Çevre Düzeni Planı	Obje Grubu
B.G.1.01	Konut	Obje Türü
B.G.1.02	Ticaret	Obje Türü
B.G.1.03	Karışık	Obje Türü
B.G.1.04	Sanayi	Obje Türü
B.G.1.05	Sosyal Donatımlar	Obje Türü
B.G.1.06	Depolama	Obje Türü
B.G.1.07	Rekreasyon	Obje Türü
B.G.1.08	Spor	Obje Türü
B.G.1.09	Stadyum	Obje Türü
B.G.1.10	Fuar	Obje Türü
B.G.1.11	Hayvanat Bahçesi	Obje Türü
B.G.1.12	Botanik Bahçesi	Obje Türü
B.G.1.13	Kamp	Obje Türü
B.G.1.14	Lunapark	Obje Türü
B.G.1.15	Mezarlık	Obje Türü
B.G.1.16	Fidanlık	Obje Türü
B.G.1.17	Ulaşım Tesisleri	Obje Türü
B.G.1.18	Maden Ocağı	Obje Türü
B.G.1.19	Arıtma Tesisi	Obje Türü
B.G.1.20	Rafineri	Obje Türü
B.G.1.21	Tersane	Obje Türü
B.G.1.22	Atık Madde Depolama	Obje Türü
B.G.1.23	Su Üretim Tesisleri	Obje Türü
B.G.1.24	Elektrik Santrali	Obje Türü
B.G.1.25	Tuzla	Obje Türü
B.G.1.26	Hayvan Çiftliği	Obje Türü
B.G.1.27	Orman	Obje Türü
B.G.1.28	Koru	Obje Türü
B.G.1.29	Çalılık	Obje Türü
B.G.1.30	Mera	Obje Türü

B.G.1.31	Tarla	Obje Türü
B.G.1.32	Bahçe	Obje Türü
B.G.1.33	Bataklık	Obje Türü
B.G.1.34	Özel Mahsül Alanı	Obje Türü
B.G.1.35	Bozkır	Obje Türü
B.G.1.36	Bitkisiz Alan	Obje Türü
B.G.1.37	Diğer	Obje Türü
B.G.2	Nazım İmar Planı	Obje Grubu
B.G.2.01	Konut	Obje Türü
B.G.2.02	Ticaret	Obje Türü
B.G.2.03	Karışık	Obje Türü
B.G.2.04	Sanayi	Obje Türü
B.G.2.05	Sosyal Donatılar	Obje Türü
B.G.2.06	Depolama	Obje Türü
B.G.2.07	Rekreasyon	Obje Türü
B.G.2.08	Spor	Obje Türü
B.G.2.09	Stadyum	Obje Türü
B.G.2.10	Fuar	Obje Türü
B.G.2.11	Hayvanat Bahçesi	Obje Türü
B.G.2.12	Botanik Bahçesi	Obje Türü
B.G.2.13	Kamp	Obje Türü
B.G.2.14	Lunapark	Obje Türü
B.G.2.15	Mezarlık	Obje Türü
B.G.2.16	Fidanlık	Obje Türü
B.G.2.17	Ulaşım Tesisleri	Obje Türü
B.G.2.18	Arıtma Tesisi	Obje Türü
B.G.2.19	Tersane	Obje Türü
B.G.2.20	Atık Madde Depolama	Obje Türü
B.G.2.21	Su Üretim Tesisleri	Obje Türü
B.G.2.22	Elektrik Santrali	Obje Türü
B.G.2.23	Orman	Obje Türü
B.G.2.24	Koru	Obje Türü
B.G.2.25	Mera	Obje Türü
B.G.2.26	Tarla	Obje Türü
B.G.2.27	Bahçe	Obje Türü
B.G.2.28	Özel Mahsül Alanı	Obje Türü
B.G.2.29	Diğer	Obje Türü
B.G.3	Uygulama İmar Planı	Obje Grubu
B.G.3.01	Konut	Obje Türü
B.G.3.02	Ticaret	Obje Türü
B.G.3.03	Karışık	Obje Türü
B.G.3.04	Sanayi	Obje Türü
B.G.3.05	Sosyal Donatılar	Obje Türü
B.G.3.06	Depolama	Obje Türü
B.G.3.07	Rekreasyon	Obje Türü
B.G.3.08	Spor	Obje Türü
B.G.3.09	Stadyum	Obje Türü
B.G.3.10	Fuar	Obje Türü
B.G.3.11	Hayvanat Bahçesi	Obje Türü
B.G.3.12	Botanik Bahçesi	Obje Türü
B.G.3.13	Kamp	Obje Türü
B.G.3.14	Lunapark	Obje Türü
B.G.3.15	Mezarlık	Obje Türü
B.G.3.16	Fidanlık	Obje Türü

B.G.3.17	Ulaşım Tesisleri	Obje Türü
B.G.3.18	Arıtma Tesisi	Obje Türü
B.G.3.19	Tersane	Obje Türü
B.G.3.20	Atık Madde Depolama	Obje Türü
B.G.3.21	Elektrik Santrali	Obje Türü
B.G.3.22	Koru	Obje Türü
B.G.3.23	Bahçe	Obje Türü
B.G.3.24	Diğer	Obje Türü

5.1.2 TABİS-OK Afet Yönetimi Obje Alanlarına Eklenen Elemanlar

Obje kataloguna yapılan eklemeler sadece öznitelik ve öznitelik değeri seviyesinde oldukları için obje kataloğunda yapılmış olan kodlamalarda herhangi bir düzeltme ya da ekleme yapılmamıştır. Aşağıdaki metinde yazınızca değiştirilen yani eskisi çıkarılıp yerine yenisi ya da yenileri eklenen öznitelikler belirtilmiştir, eklenen özniteliklere ait öznitelik değerleri Ek.3'te verilmiştir. Ek 3'teki gösterim şekli ise Tablo 5.1'dekine benzemektedir. Çıkarılan özniteliğin üstü çizili gösterilmiş, altına onun yerine ve ayrıca eklenen öznitelikler yerleştirilmiştir. İkinci tür tablo ise birden çok Obje Türüne birden eklenen aynı öznitelikler için oluşturulmuştur. Bu tabloda sadece eklenen öznitelikler gösterilmiş ve ait oldukları Obje Türü Kodları bir önceki sütunda belirtilmiştir.

B.B. Risk Bölgeleri Obje Alanı altındaki, B.B.2. Teknolojik Afet Riski Obje Grubundaki B.B.2.01. Bulaşıcı Hastalık, B.B.2.02. Trafik Kazası, B.B.2.03. Gaz Zehirlenmesi, B.B.2.04. Gıda Zehirlenmesi, B.B.2.06. Bina Yangını Obje Türlerinin özniteliklerinden biri olan OCK kısaltmalı Oluşan Can Kaybı çıkarılarak, yerine OPC Oluşan Personel ve OOC Oluşan Öğrenci Cankayıpları öznitelikleri eklenmiştir.

B.B.2.02. Trafik Kazası, B.B.2.03. Gaz Zehirlenmesi, B.B.2.04. Gıda Zehirlenmesi, B.B.2.06. Bina Yangını Obje Türlerine OYR kısaltmalı teknolojik afetin kampüsteki oluş yeri özniteliği eklenmiştir.

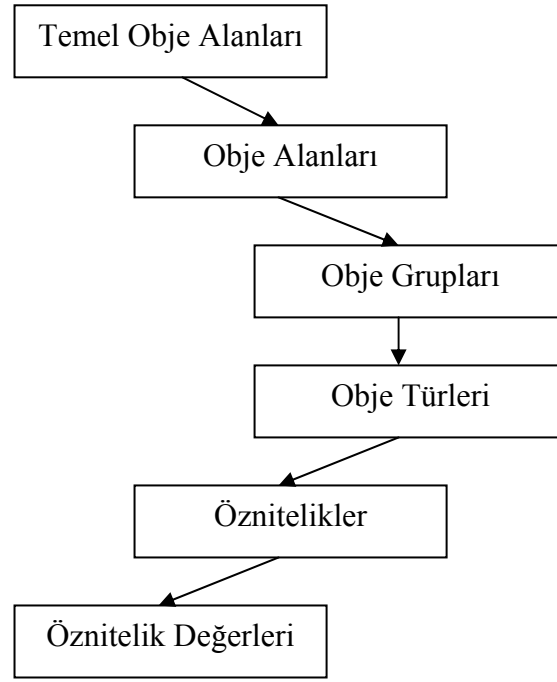
B.B.2.03. Gaz Zehirlenmesi, B.B.2.04. Gıda Zehirlenmesi, B.B.2.06. Bina Yangını Obje Türlerine NDN kısaltmalı teknolojik afetin kampüsteki oluş nedeni özniteliği eklenmiştir.

B.B. Risk Bölgeleri Obje Alanı altındaki, B.B.3. Politik Afet Riski Obje Grubundaki B.B.3.01. Terörist Saldırı Obje Türünün özniteliklerinden biri olan OCK kısaltmalı Oluşan Can Kaybı çıkarılarak, yerine OPC Oluşan Personel ve OOC Oluşan Öğrenci Cankayıpları öznitelikleri ve OYR kısaltmalı terörist saldırının kampüsteki oluş yeri özniteliği eklenmiştir.

5.2 Veritabanı Tasarımı

Veritabanı tasarımı için işlem sırası, tasarlanacak veritabanının nerede kullanılacağına belirlenmesi ile başlar. Çalışmada oluşturulan veritabanı bir Afet Bilgi Sistemi için bir altlık oluşturacağından, içereceği bilgilerin de bu amaca yönelik olması gerekmektedir. Veritabanı tasarım aşamasından önce oluşturulan Afet Yönetimi Obje Modeli (AYOM), oluşturulacak veritabanının modelini, elemanlarını ve ihtiyaçlarını da içermektedir.

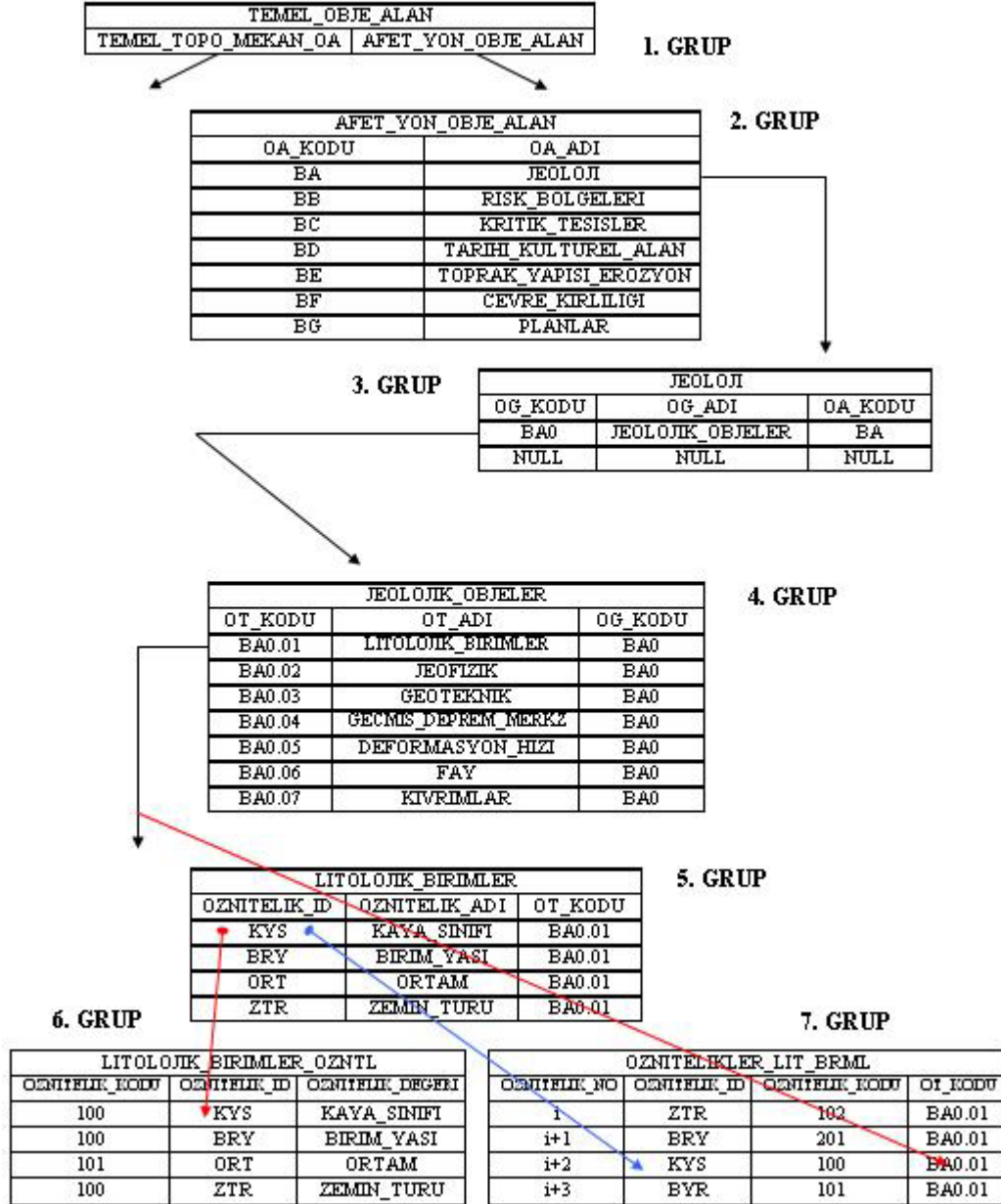
Yapılan veritabanı, yapılandırılan AYOM'a ait olacağı için kaba bir veritabanı yapısı obje modellemesinin tamamlanması ile ortaya çıkmıştır. Oluşturulan AYOM, ana kaynağı olan TABİS-OK gibi hiyerarşik bir yapıda oluştu (Şekil 5.1). Bu yapının doğrudan kullanılması oluşturulacak veritabanının da hiyerarşik bir yapıya sahip olması anlamına geleceğinden veritabanı oluşturulurken, AYOM'da bazı değişikliklere de gidildi. Bu değişiklikler büyük ve modeli etkileyecek değişiklikler değil yalnızca veritabanı sisteminin daha güvenilir, daha kullanışlı ve daha hızlı bir tür olan ilişkisel yaklaşıma uyduracak olan şekilsel değişikliklerdir.



Şekil 5.1 TABİS-OK'nun Yapısı

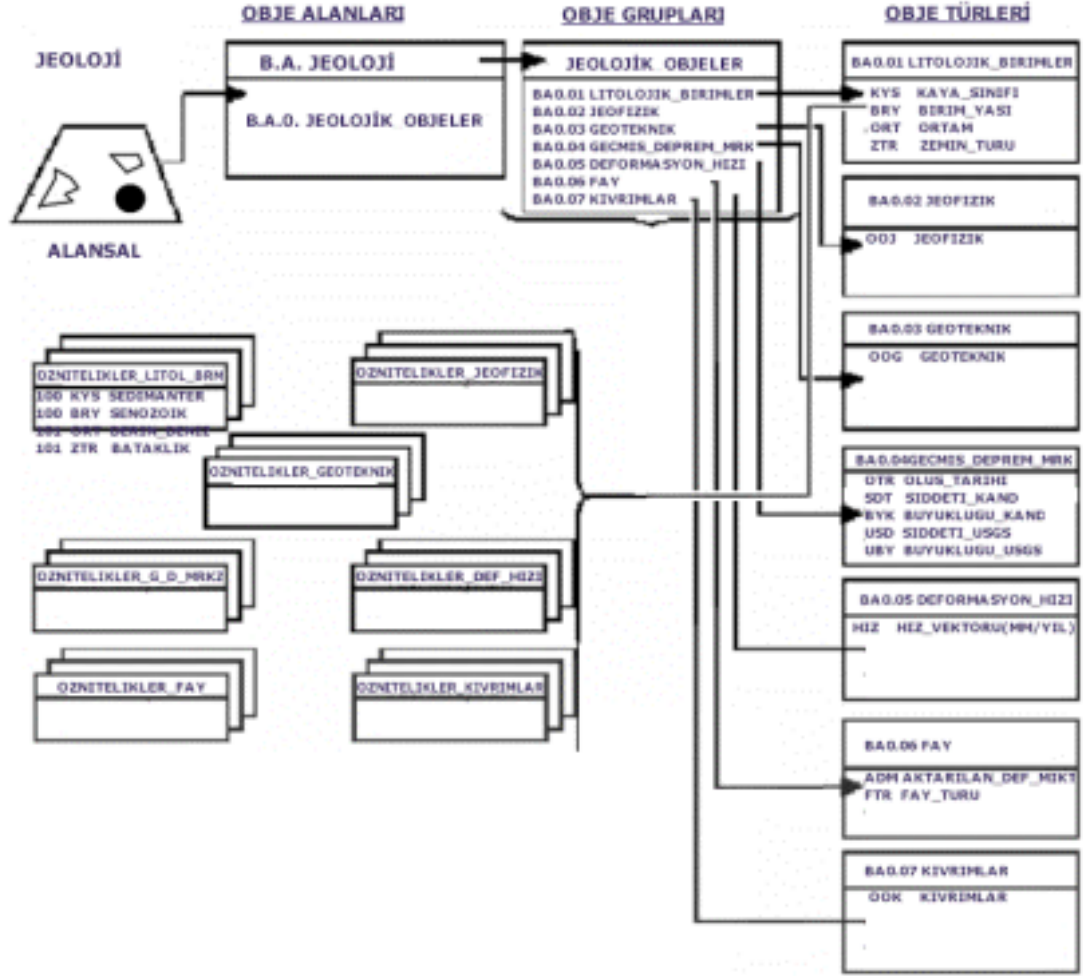
Tasarımın başında Obje Modeli yedi temel gruba ayrıldı (Şekil 5.2). Bu yedi gruptan ilk üçü uzun vadede ne yapısında ne de gruptaki tablolarda barındırdığı verilerde değişiklik yapılması öngörülmeyen ve tablolardaki verilerin de girilerek basit olarak ilişkilendirilen melez yapıda (hem hiyerarşik hem ilişkisel) bir model oluşturdu. Dördüncü grup tablolar ise ilk üç grup ile melez, son üç grup ile ilişkisel modelle bağlantının sağlandığı yine barındırdığı verilerin orta vadede değişmeyeceği öngörülen dolu tablolardan oluşan bir çeşit geçiş grubu oluşturmuş oldu. Dikkat edilmesi gereken nokta oluşturulan bu yapının sadece model için geçerli bir yapı olduğudur. Veritabanı için sadece ilişkisel yapı kullanılmıştır.

Son üç grup tamamen ilişkisel yapıda tasarlanmış ve veri girişi ve güncellemenin sık olacağı bir yapıya uygun olacak şekilde oluşturulmuştur. Temel olarak alınan obje modelinin bu yapısı sadece obje modelinin daha kolay anlaşılabilmesi için oluşturulmuştur. Tasarlanan ve oluşturulan veritabanının yapısı ile bağlantısı yoktur.



Şekil 5.2 Obje Modeli'nin Gruplandırılması

Şekil 5.3'de oluşturulan veritabanının bir obje alanının ve kapsadığı bir obje türü, bu obje türüne ait öznitelikleri ve öznitelik değerleri arasındaki ilişki gösterilmiştir.



Şekil 5.3 Oluşturulan Veritabanındaki İlişkisel Tablolar

5.3 Linux Üzerinde PostgreSQL ile Veritabanı Oluşturulması

PostgreSQL ile veritabanı oluşturmak için Bölüm 4.6. da anlatılan veritabanı araçlarından birinin kullanılması gereklidir. İlk kurulumda, bu araçlardan psql dışında hiçbirisi bilgisayara yüklenmediği için, PostgreSQL üzerinde ilk kez bir veritabanı oluşturmak için psql etkileşimli terminali ile, kurulum sırasında initialize işlemi ile oluşturmuş olduğumuz “template0” ve “template1” geçici veritabanlarından birine bağlanması gerekmektedir.

Güvenlik nedeni ile sunucu işlemleri kesinlikle root olarak yapılmamalıdır. Tüm işlemler postgres kullanıcısı kullanılarak yapılmalıdır. Olası bir sorunda, sistem dışından birisi root erişimi kazanabilir. Bu nedenle, postmaster root olarak çalıştırılmayacaktır.

Postgres kullanıcısı linux sistemlerde root iken useradd komutunu kullanılarak yaratılabilir.

```
[root@localhost root]# useradd postgres
```

Güvenlik açısından root a (ya da id'si 0 olan başka kullanıcı varsa onlara) postgres izninin verilmemesi gerekir. Varsayılan olarak root başlangıçta postgres kullanıcısı değildir ve postmaster'ı başlatamaz. Bu bağlantıyı yapabilmek için root kullanıcısından postgres kullanıcısına geçmek gereklidir (Şekil 5.4).

```
[root@localhost root] su - postgres
[postgres@localhost postgres]
[postgres@localhost postgres]$ /usr/local/pgsql/bin/psql
template1
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \l
      List of databases
Database | Owner
-----+-----
template0 | postgres
template1 | postgres
(2 rows)

template1=# \q
[postgres@localhost postgres]$
```

Şekil 5.4 PostgreSQL'e psql ile ilk bağlantı

psql sadece, bir veritabanına bağlanarak çalıştırabilir. Bu, ilk veritabanını yaratma konusu tavuk-yumurta ikilemini anımsatır. Yaratılan postgres kullanıcısı ilk veritabanını yaratmak için daha önce bahsedilen template1 veritabanı kullanılacaktır. Template1'e bağlandıktan sonra ilk veritabanımızı yaratabilir, ardından psql'i yeniden başlatarak ya da \c parametresi ile yeni yaratılan veritabanına bağlanılabilir.

İlk kez veritabanını postgres kullanıcısı yerine yönetici tarafından tanımlanan bir PostgreSQL kullanıcısıyla yaratmak daha mantıklıdır. Bunu gerçekleştirmek için öncelikle root kullanıcısından Şekil 5.3’de gösterildiği şekilde postgres kullanıcısına geçilmesi gereklidir. Postgres kullanıcısı ile CREATEUSER komutunu kullanarak yeni bir PostgreSQL kullanıcısı yaratılır ve yeni kullanıcıya arzu edilen izinler verilir (Şekil 5.4). Dikkat edilmesi gereken nokta yaratılan kullanıcının aynı zamanda da bilgisayardaki linux işletim sisteminin de bir kullanıcısı olması gerekmektedir. Bu işlemler tamamlandıktan sonra yeni yaratılan kullanıcıya geçilip, ilk veritabanı CREATEDB “veritabanı_adı” komutu ile yaratılabilir.

Veritabanı yaratıldıktan sonra psql etkileşimli terminal ile bu yaratılan veritabanına bağlanılabilir (Şekil 5.5).

```
# su - postgres
[postgres@localhost postgres]CREATEUSER himtez
Shall the new user be able to create databases? (y/n) y
Shall the new user be able to create new users (y/n) n
CREATE USER
[postgres@localhost postgres]su
# su - himtez
[himtez@localhost himtez]CREATEDB masterhim
CREATE DATABASE
[himtez@localhost himtez]psql -d masterhim
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

masterhim=#
```

Şekil 5.5 psql ile PostgreSQL’de yeni bir kullanıcı ve veritabanı yaratılması

Yukarıda anlatıldığı gibi oluşturulan veritabanına bağlanıp yedi gruba ayrılan Afet Yönetimi Obje Modeli içeriğindeki elemanlar tablolar halinde bir veritabanına dönüştürüldü. 1. Grupta sadece bir tek tablo bulunmaktadır. 2. Grupta ise bu sayı ikiye çıkar. 7. Gruba doğru ilerlendikçe Obje Alanlarının, Obje Gruplarının, Obje Türlerinin ve Özniteliklerin içerdiği veri ve bu verilerle ilişkili tablo sayısı artar. Bu akış bir çeşit piramidi anımsatır. Tamamiyle ilişkisel veritabanı kurallarına uygun oluşturulan tabloların ilişkileri 7. Grup tablolarına kadar sadece giriş yapılan ve mevcut verilerin oluşturduğu ve bu verilerin birincil ve yabancı anahtar (foreign key) görevlerini üstlendiği şekilde, 7. Grupta ise ilişkilendirme görevi, girilecek

verilerin tek (unique) olmama olasılığında gözönünde bulundurularak bu tablolarda oluşturulan OZNITELIK_NO adlı SERIAL alana (field) yani sütuna yapılacak olan veri girişinin veritabanı tarafından otomatik olarak yapıldığı ve her yeni girilen kayıt için tek bir değerin tutulduğu alana verilir.

Objekt Modelinin gruplandırılmasını gösteren Şekil 5.2 aynı zamanda da oluşturulan tablolar arasında kurulan ilişkiyi de özetlemektedir. Oluşturulan tablolara ve bu tablolar arasındaki ilişkilere ait örnekler Ek-2’de verilmiştir.

5.3.1 Tablo Yaratma

5.3.1.1 psql ile Tablo Yaratma

Oluşturulan masterhim veritabanında, Şekil 5.6’da verilen sql komutlarını psql komut satırında yazarak yeni tablolar oluşturulabilir. Diğer bir yöntem ise Şekil 5.5’de verilen SQL metinleri bir metin dosyasında yazıp kaydederek ve psql’de “\i dosya_adi” komutu ile dosyayı okutarak da tablolar oluşturmaktır. Şekilde görünen SQL komutları plain text biçimindedir, bunlar kullanılan bir metin programı ile (pico,nano,vi,vb.) istenilen şekilde düzenlenebilir(Gündüz, 2002a).

```
himtez=# create table AFET_YON_OBJE_ALAN
himtez=# (
himtez(#  OA_KODU  varchar(2)  CHECK (length(trim(OA_KODU)) = 2) not null,
himtez(#  OA_ADI   varchar(32)                                not null,
himtez(#  CONSTRAINT          AFET_YON_OBJE_ALAN_pk PRIMARY KEY(OA_KODU)
himtez(# );
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index
'afet_yon_obje_alan_pk' for table 'afet_yon_obje_alan'
CREATE
```

Şekil 5.6 psql ile tablo yaratma

Oluşturulan tabloları silmek ise,

```
himtez=# drop table AFET_YON_OBJE_ALAN
```

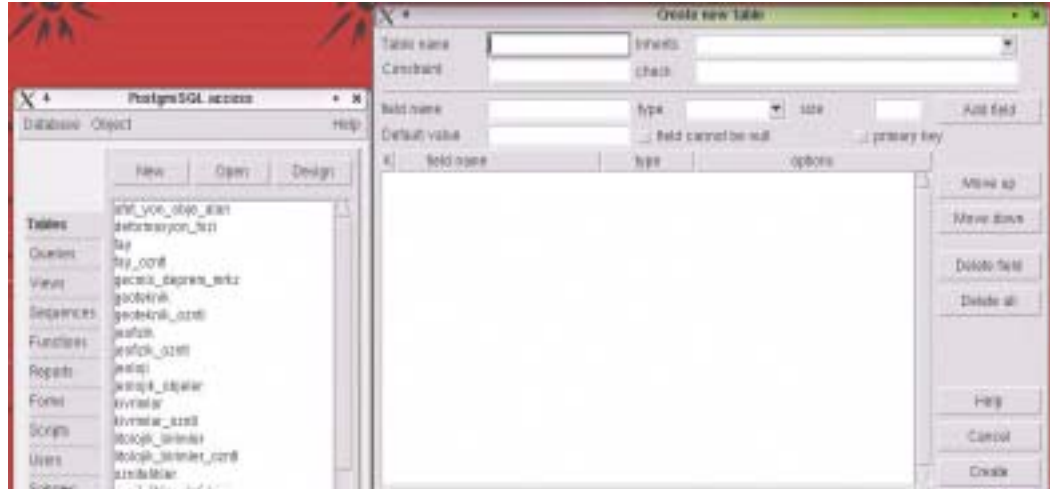
komutuyla gerçekleştirilir.

kullanılmıştır. Pgaccess ile tablo yaratmak için de öncelikle bir veritabanına bağlanması gereklidir (Şekil 5.8). Veritabanına bağladıktan sonra çalışan program tuşlarından yaratılmak istenen veritabanı elemanı seçilip “new” tuşu ile eleman oluşturulur (Şekil 5.9).



Şekil 5.8 Pgaccess ile Veritabanına Bağlanma

Şekil 5.9’da bu çalışmada oluşturulan tablolardan bazıları da görünmektedir.



Şekil 5.9 Pgaccess ile Tablo Yaratma

5.3.2 Oluşturulan Tablolara Veri Girişi

Oluşturulan tablolara veri girişide hem SQL komutları ile hem de pgaccess ve PgAdmin II arayüzleri ile yapılmıştır. Bir ilişkisel veritabanına SQL komutları ile veri girişi Şekil 5.10’ da gösterilen sekilde gerçekleştirilir.

```
insert into oznitelikler_e_d_mrkz (oznitelik_no, otr, sdt, byk, usd,
uby, drn, ock, ot_kodu) values('1999-08-17', 7.4, 7.5, 7.6, 7.7, 15,
17000, 'BA0.04');
insert into oznitelikler_e_d_mrkz (oznitelik_no, otr, sdt, byk, usd,
uby, drn, ock, ot_kodu) values('1999-10-24', 7.2, 7.1, 7.1, 7.0, 9,
9000, 'BA0.04');
```

Şekil 5.10 SQL Komutları ile PostgreSQL Veritabanına Veri Girilmesi

Yukarıda verilen örneğin sözdizimi, “oznitelikler_e_d_mrkz adlı tablodaki, oznitelik_no, otr, sdt, byk, usd, uby, drn, ock, ot_kodu, adlı alanlara sırasıyla, 1999-08-17, 7.4, 7.5, 7.6, 7.7, 15, 17000, BA0.04, değerlerini gir”dir. Bu işlem görüleceği gibi oldukça zahmetli ve dikkat gerektiren bir aşamadır. Veri girişini gerçekleştirmenin daha kolay bir yolu ise arayüzleri kullanmaktır. Çünkü arayüzlerde tablolar tüm alanları ile birlikte görünmekte ve veri girişi daha açık ve geri dönülebilir bir şekilde gerçekleştirilebilmektedir (Şekil 5.11). Hem pgaccess’de hem de PgAdmin II’de veri girişi MS excel’e veri girmekten farksızdır. Girilen değerlerin kabul edilebilirliğinin kontrolü ise kısıtlamalar ve tetikler adı verilen özelliklerle sağlanır. Bu özellikler sayesinde örneğin Şekil 5.10’da oluş tarihi OTR alanına depremin oluş tarihi yerine, oluşan can kaybının yazılması gibi, değersiz, anlamsız sorgulamaları sonuçsuz bırakacak ve veritabanının performansını düşürebilecek verilerin veritabanına sokulması engellenecektir.

oznitelik_no	otr	sdt	byk	usd	uby	drn	ock	ot_kodu
1999-08-17	7.4	7.5	7.6	7.7	15	17000	BA0.04	
1999-10-24	7.2	7.1	7.1	7.0	9	9000	BA0.04	

Şekil 5.11 Pgaccess ile Veri Girişi

5.3.3 Oluşturulan Veritabanının Test Edilmesi

Bölüm 3.2.2. Veritabanı Yapısında açıklanan bütünlük denetimi’nin test edilebilmesi, örnek olarak müşteri kaydı silindiğinde o kişinin sipariş tablosundaki siparişlerinin de otomatik olarak silinmesi işlemini veritabanının kendiliğinden gerçekleştirip gerçekleştirmediğinin kontrolü ile gerçekleşir.

Bu işlem, oluşturulan tüm tablolardaki ilişkili verilede yapıldı. Örnek olarak, FAY, FAY_OZNTL ve OZNITELIKLER_FAY adlı tablolardaki test işlemi şekil 5.12’de

gösterilmiştir. Örnekte, FAY adlı tablodaki FTR (Fay Türü) kaydı silinmiş, ve bu kayıtlarla ilişkili diğer tablolarda da FTR ile ilgili kısımların silinip silinmediği kontrol edilmiştir. FAY_OZNTL ve OZNITELIKLER_FAY adlı tablolardaki ilişkili kayıtların da silindiği şekil 5.12.a’da gösterilmiştir.

a

b

Şekil 5.12 Veritabanında Bütünlük Denetiminin Testi

5.3.4 Oluşturulan Veritabanında Sorgulamalar

Veritabanlarının oluşturulma amacı veri setlerini organize bir şekilde saklayıp ilişkilendirerek tablolar halinde tutmak ve gerektiğinde bu verilerin herbirine veya hepsine ya da belirli bir amaca yönelik olarak birleştirilerek oluşturulacak çeşitli kombinasyonlardaki setlere ulaşılmasını sağlamaktır. Eğer bu saklanan veriler birdaha kullanılmayacak ve geri çağrılmayacak olsalardı, bu kadar çok modelle organize edilmeleriyle ve ilişkilendirilmeleri ile uğraşılmazdı.

Veriler veritabanlarında, mutlaka birgün yararlanılmak için tutulurlar ve bu yüzden düzenli olarak modellenip organize edilirler. İşte verilerin ihtiyaç duyulduğu anda çağrılıp kullanıma hazır hale getirilmesine ya da verilerin bazı işlemlerden ve aşamalardan geçirilip de çağrılmasına sorgulama denir.

Veritabanlarında sorgulamalar, belirli kısıtlamalarla, kurallarla ve ilişkilerle yapılandırılmış verilerin ya da veri setlerinin bu belirlenmiş özellikler yardımıyla istenilen şekilde ve formatta ekrana, kağıda, bir işleme ya da bir programa çağırılması şeklinde gerçekleşir.

İlişkisel veritabanlarında sorgu yapısı, belirli bir ya da birkaç ilişkisel tablodan, belirli verilerin ya da bu verilerin matematiksel veya metinsel fonksiyonlarından elde edilecek yeni verilerin ya da bu verilerden bazılarının istenilen sıra veya şekilde, sorgucu tarafından belirlenecek bir gösterim platformunda kullanıma sunulmasından oluşur. Bu yapı Şekil 5.12’de de gösterildiği gibi SQL sorgulama dilinde oluşturulur.

```
linux:~ # su - postgres
postgres@linux:~> psql masterhimmet
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

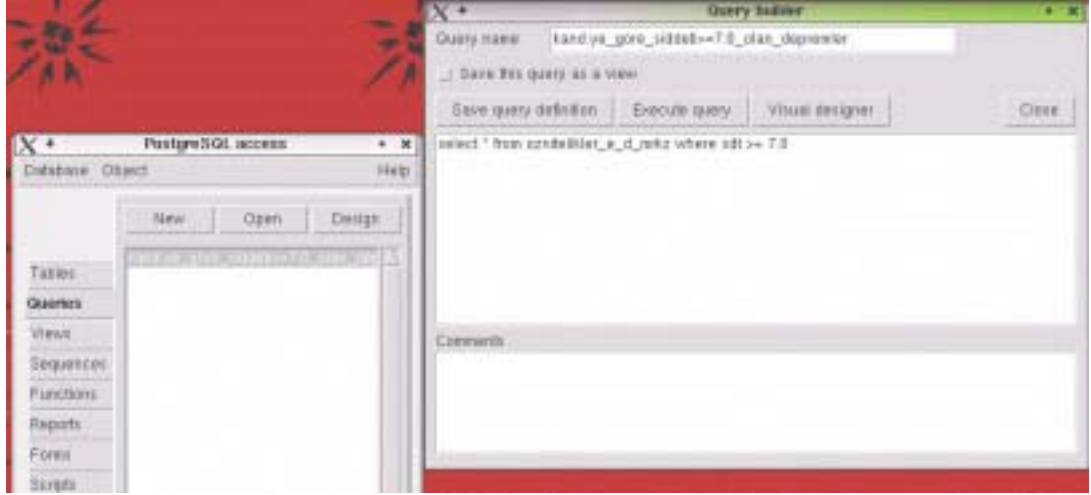
masterhimmet=# select * from pga_queries ;
               queryname               | querytype |
querycommand                        | querytables | querylinks | queryresults
| querycomments
-----+-----+-----+-----
+-----+-----+-----+-----
kand.ye_gore_siddeti>=7.0_olan_depremler | S          | select * from
oznitelikler_e_d_mrkz where sdt >= 7.0 |            |
(1 row)

masterhimmet=# select * from oznitelikler_e_d_mrkz where sdt >= 7.0
masterhimmet-# ;
 oznitelik_no |   otr   | sdt | byk | usd | uby | drn | ock | ot_kodu
-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | 1999-08-17 | 7.4 | 7.5 | 7.6 | 7.7 | 15 | 17000 | BA0.04
2 | 1999-10-24 | 7.2 | 7.1 | 7.2 | 7.3 | 13 | 9000 | BA0.04
(2 rows)
masterhimmet=#
```

Şekil 5.13 SQL ile Sorgulama

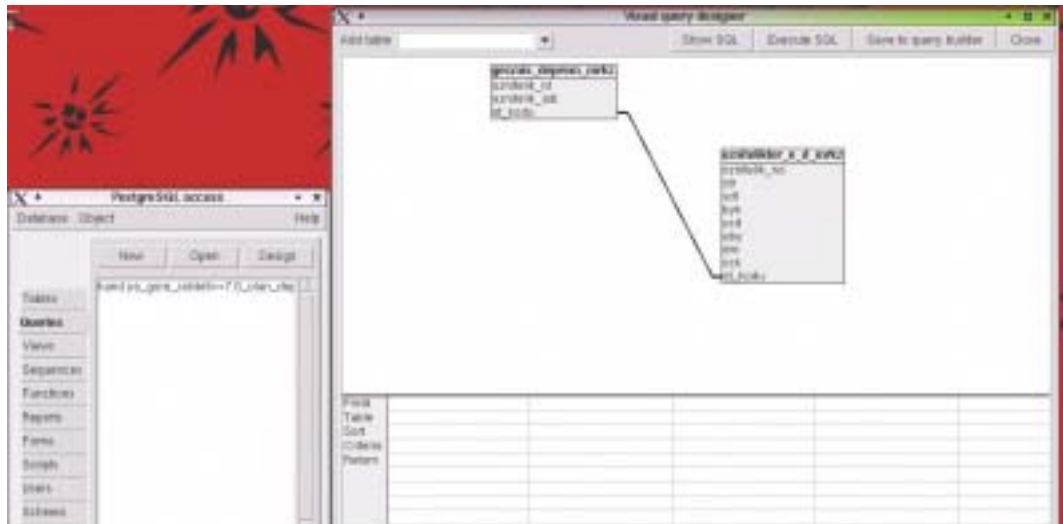
Sorgulama işleminde ister arayüz kullanılsın ister etkileşimli terminal kullanılsın, elde hazırlanmış sorgular yoksa, sorgu mutlaka SQL ile yazılmalıdır. Veritabanı oluşturulurken sistemden istenecek sorgular belirlenmişse, bunlar veritabanı hizmete girmeden önce hazırlanıp, işlemeye hazır sorgular olarak kaydedilebilir. Sorguları sistem kullanımdayken de hazırlayıp eklemek mümkündür. Bu sorguların sonuçları da görünüm (view) olarak veritabanında saklanıp ihtiyaç anında sorgu yazmaya ya seçmeye gerek kalmadan doğrudan ekrana getirilebilir.

Bu çalışmada elde edilmek istenen sonuçlardan biri de ilişkisel sorgulamaların yapılandırılabilmesidir. Yapılan çalışma sonucunda oluşturulan veritabanında yapılan sorgulamalar istenilen sonucu vermiştir. Şekil 5.13 ve Şekil 5.15'te yapılandırılan sorgulamalar ve sonuçları verilmiştir.

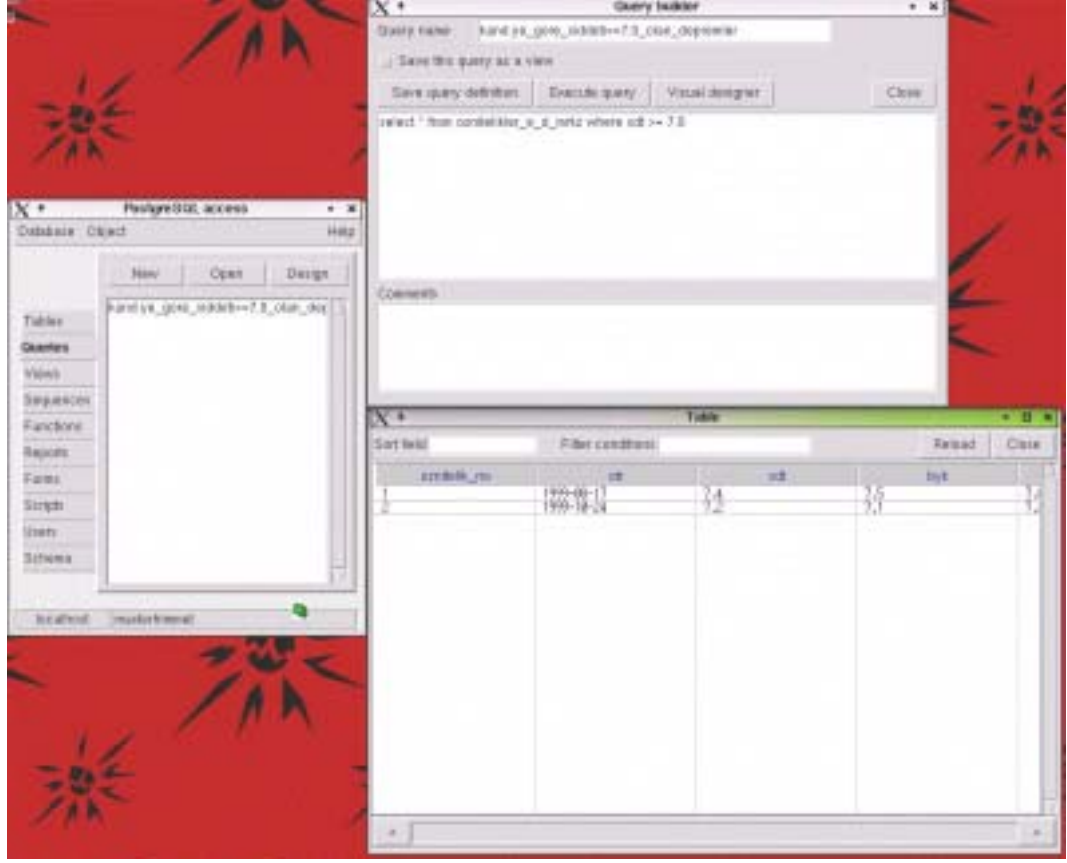


Şekil 5.14 Pgaccess ile Sorgulama

Pgaccess arayüzü ile görsel olarak da sorgu hazırlamak ve hazırlanılan bu sorguların SQL dilindeki karşılıklarını da görmek mümkündür (Şekil 5.14). İster görsel olarak ister SQL dilini kullanarak yapılsın, her iki tipteki sorguların da PostgreSQL'de farklı sonuç vermesi beklenemez.



Şekil 5.15 Pgaccess ile Görsel Sorgulama



Şekil 5.16 Pgaccess ile Yapılan Sorgulama Sonucu

Şekil 5.12’de de Şekil 5.15’de de yapılan sorgu ve bu sorgunun sonucu aynıdır. Buradan da anlaşılacağı gibi oluşturulan veritabanı fiziksel olarak kullanılan programdan, işletim sisteminden ve bilgisayardan bağımsızdır.

5.4 Oluşturulan Veritabanının Ağ Üzerinden Sunulması

Bir veritabanına bağlanmak için iki yol vardır. Bunlardan birincisi, veritabanı ismini psql komutundan hemen sonra yazmak, ikincisi ise -d “veritabanı_adı” parametresini psql komutunun ardından yazmaktır. PostgreSQL ilk kurulduğunda geçici iki boş veritabanından başka bir veritabanı içermediğinden ve bu iki veritabanının yeni veritabanları yaratabilmek için PostgreSQL’e bağlanılmasını sağladığından bahsedilmişti. Bunlardan biri template1 dir. Bu veritabanına yönetim için bağlanılabilir.

Ağ üzerinden bağlanabilmeyi kontrol edebilmek için, ağ içindeki PostgreSQL kurulu başka bir makineden kullanılan makineye bağlanması gerekir. Bu bağlantıyı da yine psql üzerinden psql parametresini, IP adresi ya da host adı için -h parametresi ve sistem veritabanlarından birinin belirtmesi için -d parametresini ekleyerek (henüz gerçek bir veritabanı yaratılmadığı varsayılırsa) komut satırına yazılırsa (Şekil 5.16);

```
remote$ psql -h 160.75.13.89 -d template1
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=#\q
remote$
```

Şekil 5.17 PostgreSQL’e Uzaktan Erişim

PostgreSQL sunucusuna bağlanılmış olur.

Eğer bu bağlantıda bir sorun çıkarsa bu sorun üç farklı nedenden meydana gelmiş olabilir;

1. PostgreSQL’in configuration dosyasındaki (postgresql.conf) TCP-IP bağlantı seçeneği false olarak kalmıştır (Şekil 5.4). Bu seçenek “true” olarak değiştirilmelidir.
2. PostgreSQL’in configuration dosyasındaki (postgresql.conf) bağlantı portu seçeneği iki bilgisayarda da farklıdır (Şekil 5.4). Bu seçenek tüm PostgreSQL sürümlerinde ve araçlarında yaygın olarak 5432 olarak gelir. Her iki bilgisayarda da port numarasının aynı olması gerekmektedir.
3. PostgreSQL sunucusu postmaster ya başlatılmamıştır ya da internet bağlantılarını kabul edecek şekilde başlatılmamıştır. Bu durumu kontrol etmek için:

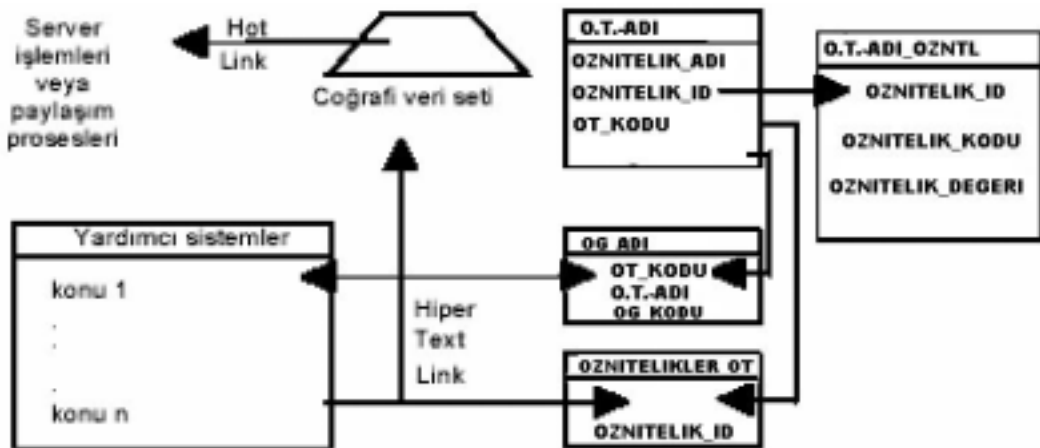
“Pg_ctl status” komutu ile postmaster sunucusunun çalışıp çalışmadığı ve eğer çalışıyorsa internet bağlantılarını kabul edip etmediği kontrol edilebilir. Varsayılan değer olarak PostgreSQL internet bağlantılarını kabul etmez.

İnternet bağlantısını sağlamak için önce postmaster'ı durdurmak sonra da internet bağlantılarını kabul edecek şekilde başlatmak gerekir. Bu işlem Şekil 5.17'de gösterilmiştir.

```
linux:~ # su - postgres
postgres@linux:=> pg_ctl status
pg_ctl: postmaster is running (pid: 1931)
-D /var/lib/pgsql/data
postgres@linux:=> pg_ctl stop
postgresql is shutting down .....
postgresql is succesfully shutdown
postgres@linux:=>postmaster -i
    DEBUG: database system was shut down at 2001-10-29 13:23:35
    EET
    DEBUG: CheckPoint record at (0, 1522064)
    DEBUG: Redo record at (0, 1522064); Undo record at (0, 0);
    Shutdown TRUE
    DEBUG: NextTransactionId: 615; NextOid: 18720
    DEBUG: database system is in production state
```

Şekil 5.18 PostgreSQL'i İnternet Bağlantısına Hazırlamak

Bu işlemlerden sonra artık PostgreSQL internet ve ağ üzerinden sunuma hazırdır. Network üzerinden PostgreSQL'e PgAdminII ve PgAccess for Windows ile de bağlanılabilir. Bağlantıyı sağlamak için yapılması gereken işlemler aynıdır. Şekil 5.19'de Suse Linux 8.0 işletim sisteminde kurulu PostgreSQL'e, network üzerinden, üzerinde PostgreSQL kurulu olmayan bir Windows 2000 işletim sistemine sahip bir bilgisayardan yapılan bağlantı gösterilmiştir. Windows 2000 işletim sistemine sahip bilgisayarda sadece PgAccess for Windows arayüzü kuruludur. Şekil 5.18 Network üzerindeki veri yolunu sembolize etmektedir.



Şekil 5.19 Ağ Üzerinde Veriyolu

6. SONUÇLAR VE TARTIŞMA

Bilgi sistemlerine yönelik veritabanı tasarımı Jeodezi ve Fotogrametri alanında eski bir konu değildir. Özellikle veritabanı teknolojisindeki gelişmeler sayesinde coğrafi bilgi sistemlerine dayalı veritabanı modelleri daha kullanışlı olacak şekilde özel amaçlara yönelik olarak şekillendirilebilmektedir. Bu özel amaçlardan biri ve Türkiye için en önemlisi afet yönetimine dayalı veritabanı modellemesidir. Yapılan çalışmada da günümüzün en yaygın ve günümüz şartlarına en uygun model olan ilişkisel veritabanı modeli seçilmiş ve kullanılmıştır. Çalışma sonucunda elde edilen sorgulamalar tatmin edici ve tutarlı çıkmış böylece veritabanı modelinin yapılandırılan Afet Yönetimi Obje Modeline uyumluluğu da görülmüştür.

Her ne kadar kullanılan model istenilen sonucu verse de yapılandırılan sistemin amacı da gözönünde bulundurulduğunda, sistemin sadece günümüzde kullanılması değil uzun bir vadede kullanılması düşünülmektedir. Bu amaçlar da dikkate alındığında, yapılandırılan modelin bu aşamada bırakılması sürekli gelişen dünyamızda atıl bir seviyede kalıp güncelliğini yitirmesine ve doğacak yeni ihtiyaçları karşılayamamasına neden olabilir. Her ne kadar oluşturulan ilişkisel veritabanı yıllardır dünyanın birçok yerinde başarıyla kullanılıyor olsa da, oluşan yeni teknolojilere de sırt çevirmemek ve sistemi ilerleyen teknolojik gelişmelerin hızına ayak uydurur bir seviyede tutmak gerekmektedir.

Günümüzde çok sık olmasa da özellikle programcılığa ve bilgi sistemlerine yönelik olarak kullanılan nesnel ya da nesneye yönelik (object-oriented) model ilk olarak gözönünde bulundurulması gereken gelişmelerden biridir. Deneme aşamasında ve kullanımda olan bilgi sistemlerine yönelik nesnel model veritabanları incelenmeli ve gerekirse bu modelle ilgili uygulamalar da yapıp sistemin ihtiyaçlarını karşılama seviyeleri karşılaştırılmalıdır.

Gelişme ve geliştirme olarak dikkate alınması gereken bir diğer konu ise, kullanılan PostgreSQL yazılımında bulunan ve kullanıcılara oldukça büyük kolaylıklar sağlayan arayüz ve fonksiyon geliştirip yazılımı kullanıcıya özelleştirme yeteneğidir.

Sistemin kullanıcılarının profillerine yönelik olarak PostgreSQL yazılımı istenildiği gibi modifiye edilebilmektedir. Aynı zamanda kullanıcı arayüzleri, internet üzerinden sorgulama yapılmasını sağlayacak Java, Perl, Python, C, Tcl, vb. programlama dilleri destekli bir internet sitesi de geliştirilmesi gereken önemli detaylardır. Geliştirilebilecek olan bu arayüzler kullanıcıya özel olarak da çalıştırılabilir. Örneğin İ.T.Ü. Ayazağa Kampüsünde sisteme veri girecek kullanıcıların kullanım tercihlerine göre geliştirilecek arayüzler veritabanında tanımlı bir şifre ve kullanıcı adına bağlı olarak çalıştırılabilir. Böylece sistemin maksimum etkinlikle kullanımı sağlanabilir.

İstanbul Teknik Üniversitesi'nde mevcut personel ve öğrenci veritabanları sistem ile ilişkilendirilip, kullanıcı grupları ve bu gruplara özel sorgulama izinleri atanarak ve bu sorgulamaların yapılacağı JDBC destekli bir sorgulama sitesi kurularak öğrenci ve personeline tanınan haklar çerçevesinde sistemden yararlanması sağlanabilir.

Ülkemizin coğrafi konumu itibarı ile her an bir afete maruz kalabileceğimiz oldukça açık olarak gözükmemektedir. Bu olası afetlerden de minimum kayıpla çıkabilmek için oluşturulması gereken afet bilgi sistemlerinin önemide net bir şekilde ortadadır. Tüm bunlar gözönünde bulundurulduğunda afet bilgi sistemlerine veri ve bu verileri sorgulama yeteneği sağlayan veritabanlarının da gelişmelere ve bilgi sistemlerinin de veritabanı sistemlerindeki gelişmelere ayak uydurması gerekliliği ortadadır.

KAYNAKLAR

- Başaraner, A.M.**, 1997. Turistik Amaçlı Coğrafi Bilgi Sistemi Oluşturulması, *Yüksek Lisans Tezi*, Y.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Başoğlu, N.**, 1994. veri Tabanları, Boğaziçi Üniversitesi Bilgisayar Programcılığı Sertifika Programı Ders Notları
- Berber, M., Koçyiğit, D.**, 1996. Veri Yapıları ve Veri Tabanları, Kartografya’da Otomasyon Dersi Seminer Çalışması (Prof. Dr. Mehmet Selçuk), Y.T.Ü.
- Gündüz, D., (a)**, 2002. PostgreSQL Veritabanı Sunucusu Kurulumu ve Çalıştırılması, *Linux Kullanıcıları Derneği Seminerleri*.
- Gündüz, D., (b)**, 2002. Veritabanına Giriş, *Linux Kullanıcıları Derneği Seminerleri, İstanbul*.
- Karaman, H. ve diğerleri.**, 2002. GIS Standards of Turkey based on Emergency Management, *International Symposium on Geographic Information Systems*, İstanbul, Turkey, September 23-26, 2002.
- Kaynarca, A.**, 1994. A Geodetic Database Design Architecture, *Master of Science*, Boğaziçi University, Institute for Graduate Studies in Earthquake Research, İstanbul.
- Momjian, Bruce**, *PostgreSQL: Introduction and Concepts*, Boston, MA: Addison-Wesley, 2001.
- Perkins, Jeff**, *PostgreSQL*, Indianapolis, Ind. : Premier Press, c2001.
- Sahin, M., Uçar, D., Baykal, O., Türkoğlu, H., Tarı, E., İpbüker, C., Musaoğlu, N., Göksel, Ç., Coşkun, M.Z., Kaya, Ş., Yiğiter, R., Erden, T., Karaman, H., Yavaşoğlu, H., Bilgi, S., Üstün, B.**, 2002. TABİS: Ulusal UAS-CBS Bazlı, Afet Yönetimi Odaklı CBS Standartlarının Oluşturulması Projesi, İ.T.Ü., İstanbul.

<http://www.m-w.com>

<http://www.postgresql.org>

EKLER

EK 1: Veritabanı İçinde Yaratılan Tabloların SQL Syntaxları

```
create table AFET_YON_OBJE_ALAN
(
    OA_KODU   varchar(2)  CHECK (length(trim(OA_KODU)) = 2) not null,
    OA_ADI    varchar(32)                                not null,
    CONSTRAINT AFET_YON_OBJE_ALAN_pk PRIMARY KEY(OA_KODU)
);
```

```
create table JEOLOJI
(
    OG_KODU   varchar(3)  CHECK (length(trim(OG_KODU)) = 3) not null,
    OG_ADI    varchar(32)                                not null,
    OA_KODU   varchar(2) REFERENCES AFET_YON_OBJE_ALAN ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT JEOLOJI_pk PRIMARY KEY(OG_KODU)
);
```

```
create table JEOLOJIK_OBJELER
(
    OT_KODU   varchar(6)  CHECK (length(trim(OT_KODU)) = 6) not null,
    OT_ADI    varchar(32)                                not null,
    OG_KODU   varchar(3)  REFERENCES JEOLOJI ON UPDATE CASCADE ON
DELETE SET NULL,
    CONSTRAINT JEOLOJIK_OBJELER_pk PRIMARY KEY(OT_KODU)
);
```

```
create table LITOLOJIK_BIRIMLER
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32)                                not null,
    OT_KODU   varchar(6) REFERENCES JEOLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT LITOLOJIK_BIRIMLER_pk PRIMARY KEY(OZNITELIK_ID)
);
```

```
create table LITOLOJIK_BIRIMLER_OZNTL
(
    OZNITELIK_KODU INTEGER CHECK (OZNITELIK_KODU >= 100) not null,
    OZNITELIK_ID   CHAR(3) REFERENCES LITOLOJIK_BIRIMLER ON UPDATE
CASCADE ON DELETE SET NULL,
    OZNITELIK_DEGERI VARCHAR(28)                                not null,
    CONSTRAINT LITOLOJIK_BIRIMLER_OZNTL_pk PRIMARY
KEY(OZNITELIK_DEGERI)
);
```

```

create table OZNITELIKLER
(
    OZNITELIK_NO SERIAL not null,
    OZNITELIK_ID CHAR(3) REFERENCES LITOLOJIK_BIRIMLER ON UPDATE
CASCADE ON DELETE SET NULL,
    OZNITELIK_KODU INTEGER CHECK (OZNITELIK_KODU >= 100) not null,
    OT_KODU varchar(6) REFERENCES JEOLLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT OZNITELIKLER_pk PRIMARY KEY(OZNITELIK_NO)
);

create table JEOFIZIK
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32) not null,
    OT_KODU varchar(6) REFERENCES JEOLLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT JEOFIZIK_pk PRIMARY KEY(OZNITELIK_ID)
);

create table JEOFIZIK_OZNTL
(
    OZNITELIK_NO SERIAL NOT NULL,
    OZNITELIK_KODU INTEGER DEFAULT 999 CHECK (OZNITELIK_KODU >= 100)
not null,
    OZNITELIK_ID CHAR(3) DEFAULT '000' REFERENCES JEOFIZIK ON
UPDATE CASCADE ON DELETE SET NULL,
    OZNITELIK_DEGERI VARCHAR(128) not null,
    VERI_GIRIS_TARIHI TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT JEOFIZIK_OZNTL_pk PRIMARY KEY(OZNITELIK_NO)
);

create table GEOTEKNIK
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32) not null,
    OT_KODU varchar(6) REFERENCES JEOLLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT GEOTEKNIK_pk PRIMARY KEY(OZNITELIK_ID)
);

create table GEOTEKNIK_OZNTL
(
    OZNITELIK_NO SERIAL NOT NULL,
    OZNITELIK_KODU INTEGER DEFAULT 999 CHECK (OZNITELIK_KODU >= 100)
not null,
    OZNITELIK_ID CHAR(3) DEFAULT '000' REFERENCES GEOTEKNIK ON
UPDATE CASCADE ON DELETE SET NULL,
    OZNITELIK_DEGERI VARCHAR(128) not null,
    VERI_GIRIS_TARIHI TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT GEOTEKNIK_OZNTL_pk PRIMARY KEY(OZNITELIK_NO)
);

```

```

create table GECMIS_DEPREM_MRKZ
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32) not null,
    OT_KODU varchar(6) DEFAULT 'BA0.04' REFERENCES JEOLLOJIK_OBJELER
ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT GECMIS_DEPREM_MRKZ_pk PRIMARY KEY(OZNITELIK_ID)
);

```

```

create table OZNITELIKLER_E_D_MRKZ
(
    OZNITELIK_NO SERIAL not null,
    OTR DATE CHECK (OTR BETWEEN '1800-01-01' AND CURRENT_DATE) NOT
NULL,
    SDT FLOAT,
    BYK FLOAT,
    USD FLOAT,
    UBY FLOAT,
    DRN INTEGER,
    OCK INTEGER NOT NULL,
    OT_KODU varchar(6) DEFAULT 'BA0.04' REFERENCES JEOLLOJIK_OBJELER
ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT OZNITELIKLER_E_D_MRKZ_pk PRIMARY KEY(OZNITELIK_NO)
);

```

```

create table DEFORMASYON_HIZI
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32) not null,
    OT_KODU varchar(6) REFERENCES JEOLLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT DEFORMASYON_HIZI_pk PRIMARY KEY(OZNITELIK_ID)
);

```

```

create table OZNITELIKLER_DEF_HIZI
(
    OZNITELIK_NO SERIAL not null,
    HIZ INTEGER not null,
    OT_KODU varchar(6) DEFAULT 'BA0.05' REFERENCES
JEOLLOJIK_OBJELER ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT OZNITELIKLER_DEF_HIZI_pk PRIMARY KEY(OZNITELIK_NO)
);

```

```

create table FAY
(
    OZNITELIK_ID CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI VARCHAR(32) not null,
    OT_KODU varchar(6) REFERENCES JEOLLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT FAY_pk PRIMARY KEY(OZNITELIK_ID)
);

```

```

create table FAY_OZNTL
(
    OZNITELIK_KODU  INTEGER  CHECK (OZNITELIK_KODU >= 100) not null,
    OZNITELIK_ID    CHAR(3) REFERENCES FAY ON UPDATE CASCADE ON
DELETE SET NULL,
    OZNITELIK_DEGERI  VARCHAR(28)                                not null,
    CONSTRAINT       FAY_OZNTL_pk PRIMARY KEY(OZNITELIK_DEGERI)
);

create table OZNITELIKLER_FAY
(
    OZNITELIK_NO      SERIAL                                not null,
    ADM               FLOAT                                NOT NULL,
    OZNITELIK_ID      CHAR(3) CHECK (OZNITELIK_ID = 'FTR') DEFAULT
'FTR' REFERENCES FAY ON UPDATE CASCADE ON DELETE SET NULL,
    OZNITELIK_KODU    INTEGER  CHECK (OZNITELIK_KODU >= 100) not null,
    OT_KODU           varchar(6) DEFAULT 'BA0.06' REFERENCES
JEOLOJIK_OBJELER ON UPDATE CASCADE ON DELETE SET NULL,
    CONSTRAINT        OZNITELIKLER_FAY_pk PRIMARY KEY(OZNITELIK_NO)
);

create table KIVRIMLAR
(
    OZNITELIK_ID      CHAR(3) CHECK (length(trim(OZNITELIK_ID)) = 3) not
null,
    OZNITELIK_ADI     VARCHAR(32)                                not null,
    OT_KODU           varchar(6) REFERENCES JEOLOJIK_OBJELER ON UPDATE
CASCADE ON DELETE SET NULL,
    CONSTRAINT        KIVRIMLAR_pk PRIMARY KEY(OZNITELIK_ID)
);

create table KIVRIMLAR_OZNTL
(
    OZNITELIK_NO      SERIAL                                NOT NULL,
    OZNITELIK_KODU    INTEGER DEFAULT 900 CHECK (OZNITELIK_KODU = 900)
not null,
    OZNITELIK_ID      CHAR(3) DEFAULT 'OOK' CHECK (OZNITELIK_ID = 'OOK')
REFERENCES KIVRIMLAR ON UPDATE CASCADE ON DELETE SET NULL,
    OZNITELIK_DEGERI  VARCHAR(128)                                not null,
    VERI_GIRIS_TARIHI  TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT        KIVRIMLAR_OZNTL_pk PRIMARY KEY(OZNITELIK_NO)
);

```

EK 2: Veritabanı İçin PostgreSQL ile Oluşturulan Tablolardan Örnekler (Çıktılar psql İnteraktif Terminalden Her Türden Bir Örnek Alınmıştır)

List of relations		
Name	Type	Owner
afet_yon_obje_alan	table	postgres
deformasyon_hizi	table	postgres
fay	table	postgres
fay_ozntl	table	postgres
gecmis_deprem_mrkz	table	postgres
geoteknik	table	postgres
geoteknik_oznt_oznitelik_no_seq	sequence	postgres
geoteknik_ozntl	table	postgres
jeofizik	table	postgres
jeofizik_ozntl	table	postgres
jeofizik_ozntl_oznitelik_no_seq	sequence	postgres
jeoloji	table	postgres
jeolojik_objeler	table	postgres
kivrimlar	table	postgres
kivrimlar_oznt_oznitelik_no_seq	sequence	postgres
kivrimlar_ozntl	table	postgres
litolojik_birimler	table	postgres
litolojik_birimler_ozntl	table	postgres
oznitelikler	table	postgres
oznitelikler_d_oznitelik_no_seq	sequence	postgres
oznitelikler_def_hizi	table	postgres
oznitelikler_e_d_mrkz	table	postgres
oznitelikler_e_oznitelik_no_seq	sequence	postgres
oznitelikler_f_oznitelik_no_seq	sequence	postgres
oznitelikler_fay	table	postgres
oznitelikler_oznitelik_no_seq	sequence	postgres
pga_forms	table	postgres
pga_layout	table	postgres
pga_queries	table	postgres
pga_reports	table	postgres
pga_schema	table	postgres
pga_scripts	table	postgres
(32 rows)		

```
postgres@linux:~> psql masterhimmet
Welcome to psql, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
masterhimmet=# select * from afet_yon_obje_alan
masterhimmet=# ;
 oa_kodu | oa_adi
-----+-----
 BA      | JEOLJJI
 BB      | RISK_BOLGELERI
 BC      | KRITIK_TESISLER
 BD      | TARIHI_VE_KULTUREL_ALAN
 BE      | TOPRAK_YAPISI_VE_EROZYON
 BF      | CEVRE_KIRLILIGI
 BG      | PLANLAR
(7 rows)
```

```

masterhimmet=# select * from jeoloji;
og_kodu |      og_adi      | oa_kodu
-----+-----+-----
BA0      | JEOLOJIK_OBJELER | BA
(1 row)

```

```

masterhimmet=# select * from jeolojik_objeler;
ot_kodu |      ot_adi      | og_kodu
-----+-----+-----
BA0.01  | LITOLOJIK_BIRIMLER | BA0
BA0.02  | JEOFIZIK          | BA0
BA0.03  | GEOTEKNIK          | BA0
BA0.04  | GECMIS_DEPREM_MERKEZLERI | BA0
BA0.05  | DEFORMASYON_HIZI  | BA0
BA0.06  | FAY                | BA0
BA0.07  | KIVRIMLAR          | BA0
(7 rows)

```

```

masterhimmet=# select * from litolojik_birimler;
oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
KYS           | KAYA_SINIFI   | BA0.01
BRY           | BIRIM_YASI    | BA0.01
ORT           | ORTAM         | BA0.01
ZTR           | ZEMIN_TURU    | BA0.01
(4 rows)

```

```

masterhimmet=# select * from jeofizik;
oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
00J          | JEOFIZIK      | BA0.02
(1 row)

```

```

masterhimmet=# select * from geoteknik;
oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
00G          | GEOTEKNIK     | BA0.03
(1 row)

```

```

masterhimmet=# select * from gecmis_deprem_mrkz ;
oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
OTR          | OLUS_TARIHI   | BA0.04
SDT          | KANDILLI_SIDDETI | BA0.04
BYK          | KANDILLI_BUYUKLUGU | BA0.04
USD          | USGS_SIDDETI  | BA0.04
UBY          | USGS_BUYUKLUGU | BA0.04
DRN          | DERINLIGI     | BA0.04
OCK          | OLUSAN_CAN_KAYBI | BA0.04
(7 rows)

```

```

masterhimmet=# select * from deformasyon_hizi ;
oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
HIZ          | HIZ_VEKTORU(MM/YIL) | BA0.05
(1 row)

```

```

masterhimmet=# select * from fay;
  oznitelik_id |          oznitelik_adi          | ot_kodu
-----+-----+-----
  ADM          | AKTARILAN_DEFORMASYON_MIKTARI | BA0.06
  FTR          | FAY_TURU                       | BA0.06
(2 rows)

```

```

masterhimmet=# select * from kivrimlar;
  oznitelik_id | oznitelik_adi | ot_kodu
-----+-----+-----
  OOK          | KIVRIMLAR     | BA0.07
(1 row)

```

```

masterhimmet=# select * from litolojik_birimler_ozntl ;
  oznitelik_kodu | oznitelik_id |          oznitelik_degeri
-----+-----+-----
          100 | KYS          | SEDIMANTER
          101 | KYS          | ALUVYON
          102 | KYS          | ESKI_ALUVYON
          103 | KYS          | YAMAC_MOLOZU
          104 | KYS          | KUMUL
          105 | KYS          | PLAJ_KUMU
          106 | KYS          | CAKIL
          107 | KYS          | KUMTASI
          108 | KYS          | SILTTASI
          109 | KYS          | CAMURTASI
          110 | KYS          | SEYL
          111 | KYS          | KONGLOMERA
          112 | KYS          | BRES
          113 | KYS          | KIRECTASI
          114 | KYS          | MARN
          115 | KYS          | DOLOMIT
          116 | KYS          | TRAVERTEN
          117 | KYS          | TUF
          118 | KYS          | EVAPORIT
          119 | KYS          | CORT
          200 | KYS          | METAMORFIK
          201 | KYS          | FILLAT
          202 | KYS          | SIST
          203 | KYS          | GNAYS
          204 | KYS          | MERMER
          205 | KYS          | AMFIBOLIT
          206 | KYS          | KATAKLASTIK
          207 | KYS          | PIROKLASTIK
          208 | KYS          | MILONIT
          209 | KYS          | METAPLUTONIK
          210 | KYS          | METAVOLKANIK
          211 | KYS          | METASEDIMANTER
          212 | KYS          | METAKONGLOMERA
          213 | KYS          | KUARSIT
          214 | KYS          | FELS
          215 | KYS          | HORNFELS
          300 | KYS          | VOLKANIK
          301 | KYS          | BAZALT
          302 | KYS          | GABRO
          303 | KYS          | DIYORIT
          304 | KYS          | ANDEZIT
          305 | KYS          | GRANIT
          306 | KYS          | RIYOLIT
          307 | KYS          | GRANUDIYORIT
          308 | KYS          | KUARS
          309 | KYS          | MONZONIT

```

310		KYS		DASIT
311		KYS		RIYODASIT
312		KYS		SIYENIT
100		BRY		SENOZOIK
101		BRY		KUVATERNER
102		BRY		TERSIYER
200		BRY		MESOZOIK
201		BRY		KRETASE
202		BRY		JURA
203		BRY		TRİYAS
300		BRY		PALEOZOIK
301		BRY		PERMIYEN
302		BRY		KARBONIFER
303		BRY		DEVONIYEN
304		BRY		SILURYEN
305		BRY		ORDOVİSYEN
306		BRY		KAMBRIYEN
100		ORT		Sedimanter
101		ORT		DERİN_DENİZ
102		ORT		SG_DENİZ
103		ORT		PLATFORM
104		ORT		KARASAL
105		ORT		FLUVYAL
106		ORT		GOL
107		ORT		RESİFAL
200		ORT		Metamorfik
201		ORT		YESİLSİST
202		ORT		MAVİSİST
203		ORT		YUKSEK_BASINÇ_ALCAK_BASINÇ
300		ORT		Volkanik
301		ORT		PLUTONİK
302		ORT		EXTRUSİVE
303		ORT		INTRUSİVE
304		ORT		OKYANUSAL_KABUK
101		ZTR		Göl
102		ZTR		BATAKLIK
103		ZTR		ALUVYAL_DUZLUK
104		ZTR		TASKIN_OVASI
105		ZTR		DELTA
106		ZTR		PLAJ

(86 rows)

lines 1-90/90 (END)

```

masterhimmet=# select * from jeofizik_ozntl ;
  oznitelik_no | oznitelik_kodu | oznitelik_id |          oznitelik_degeri          |          veri_giris_tarihi
-----+-----+-----+-----+-----
      1 |          999 | 000          | JEOFIZIK_INCELEMESI_YAPILMAMIS | 2003-04-27 03:44:55.298228-07
(1 row)

```

```

masterhimmet=# select * from geoteknik_ozntl ;
  oznitelik_no | oznitelik_kodu | oznitelik_id |          oznitelik_degeri          |          veri_giris_tarihi
-----+-----+-----+-----+-----
      1 |          999 | 000          | GEOTEKNIK_BAKIMDAN_YETERSIZ | 2003-04-27 04:04:43.580506-07
(1 row)

```

```

masterhimmet=# select * from fay_ozntl ;
  oznitelik_kodu | oznitelik_id | oznitelik_degeri
-----+-----+-----
      100 | FTR          | DOGRULTU_ATIMLI_FAY
      101 | FTR          | SAG_YONLU_FAY
      102 | FTR          | SOL_YONLU_FAY
      200 | FTR          | NORMAL_FAY
      300 | FTR          | TERS_FAY
      400 | FTR          | BINDIRMELI_FAY
(6 rows)

```

```

masterhimmet=# select * from kivrimlar_ozntl ;
  oznitelik_no | oznitelik_kodu | oznitelik_id |          oznitelik_degeri          |          veri_giris_tarihi
-----+-----+-----+-----+-----
      2 |          900 | 00K          | FAYDAKI_0.2MMLIK_DEFORMARYONUN_SONUCU | 2003-04-27 05:04:43.467817-07
      3 |          900 | 00K          | FDS | 2003-04-27 05:04:51.219578-07
(2 rows)

```

```

masterhimmet=# select * from oznitelikler;
  oznitelik_no | oznitelik_id | oznitelik_kodu | ot_kodu
-----+-----+-----+-----
      1 | BRY          |          101 | BA0.01
      2 | ORT          |          302 | BA0.01
(2 rows)

```

```

masterhimmet=# select * from oznitelikler_e_d_mrkz ;
oznitelik_no |   otr   | sdt | byk | usd | uby | drn | ock | ot_kodu
-----+-----+-----+-----+-----+-----+-----+-----+-----
          1 | 1999-08-17 | 7.4 | 7.5 | 7.6 | 7.7 | 15 | 17000 | BA0.04
          2 | 1999-10-24 | 7.2 | 7.1 | 7.2 | 7.3 | 13 | 9000 | BA0.04

```

(2 rows)

```

masterhimmet=# select * from oznitelikler_def_hizi ;
oznitelik_no | hiz | ot_kodu
-----+-----+-----

```

```

          1 |    2 | BA0.05
          2 |    1 | BA0.05
          3 |    0 | BA0.05

```

(3 rows)

```

masterhimmet=# select * from oznitelikler_fay ;
oznitelik_no | adm | oznitelik_id | oznitelik_kodu | ot_kodu
-----+-----+-----+-----+-----
          1 | 0.2 | FTR          |                | 100 | BA0.06
          2 | 0.08 | FTR          |                | 400 | BA0.06
          3 | 0.1 | FTR          |                | 102 | BA0.06
          4 | 0.3 | FTR          |                | 200 | BA0.06

```

(4 rows)

```

masterhimmet=#

```

EK 3: TABİS Obje Kataloğuna Eklenen Öznitelikler ve Öznitelik Değerleri

TABİS Obje Kataloğu		Versiyon 3.2: 20/10/2002		
B.Afet Yönetimi Obje Alanları:		Obje Grubu:		
Kodu:	Adı:	Kodu:	Adı:	
B.B.	Risk Bölgeleri	B.B.2.	Teknolojik Afet Riski	
Obje Türü:				
Kodu:	Adı:			
B.B.2.01	Bulaşıcı Hastalık			
Tanım:				
Mekansal Referans Türü: Alansal				
Adı:				
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)			
ADI	Adı (Resmi Adı)			
Öznitelikleri:				
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi
OÇK		Oluşan Can Kaybı	0:1	integer
		Salgında oluşan can kaybı yazılacaktır.		
<i>OPC</i>		<i>Oluşan Personel Can Kaybı</i>	<i>0:1</i>	<i>integer</i>
		Salgında oluşan personel can kaybı yazılacaktır.		
<i>OOC</i>		<i>Oluşan Öğrenci Can Kaybı</i>	<i>0:1</i>	<i>integer</i>
		Salgında oluşan öğrenci can kaybı yazılacaktır.		

TABİS Obje Kataloğu		Versiyon 3.2: 20/10/2002		
B.Afet Yönetimi Obje Alanları:		Obje Grubu:		
Kodu:	Adı:	Kodu:	Adı:	
B.B.	Risk Bölgeleri	B.B.2.	Teknolojik Afet Riski	
Obje Türü:				
Kodu:	Adı:			
B.B.2.02	Trafik Kazası			
Tanım:				
Mekansal Referans Türü: Noktasal				
Adı:				
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)			
ADI	Adı (Resmi Adı)			
Öznitelikleri:				
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi
OCK		Oluşan Can Kaybı	1:1	integer
		Kazada oluşan can kaybı yazılacaktır.		
OPC		Oluşan Personel Can Kaybı	0:1	integer
		Kazada oluşan personel can kaybı yazılacaktır.		
OOC		Oluşan Öğrenci Can Kaybı	0:1	integer
		Kazada oluşan öğrenci can kaybı yazılacaktır.		
OYR		Oluş Yeri	1:1	varchar
		Kazanın kampüsün neresinde olduğu yazılacaktır.		

TABİS Obje Kataloğu		Versiyon 3.2: 20/10/2002		
B.Afet Yönetimi Obje Alanları:		Obje Grubu:		
Kodu:	Adı:	Kodu:	Adı:	
B.B.	Risk Bölgeleri	B.B.2.	Teknolojik Afet Riski	
Obje Türü:				
Kodu:	Adı:			
B.B.2.03	Gaz Zehirlenmesi			
Tanım:				
Mekansal Referans Türü: Alansal				
Adı:				
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)			
ADI	Adı (Resmi Adı)			
Öznitelikleri:				
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi
OCK		Oluşan Can Kaybı	0:1	integer
		Zehirlenmede oluşan can kaybı yazılacaktır.		
OPC		<i>Oluşan Personel Can Kaybı</i>	<i>0:1</i>	<i>integer</i>
		<i>Zehirlenmede oluşan personel can kaybı yazılacaktır.</i>		
OOC		<i>Oluşan Öğrenci Can Kaybı</i>	<i>0:1</i>	<i>integer</i>
		<i>Zehirlenmede oluşan öğrenci can kaybı yazılacaktır.</i>		
OYR		<i>Oluş Yeri</i>	<i>1:1</i>	<i>varchar</i>
		<i>Zehirlenmenin kampüsün neresinde olduğu yazılacaktır.</i>		
NDN		<i>Nedeni</i>	<i>1:1</i>	<i>varchar</i>
		<i>Gaz zehirlenmesinin nedeni yazılacaktır.</i>		

TABİS Obje Kataloğu			Versiyon 3.2: 20/10/2002	
B.Afet Yönetimi Obje Alanları:			Obje Grubu:	
Kodu:	Adı:		Kodu:	Adı:
B.B.	Risk Bölgeleri		B.B.2.	Teknolojik Afet Riski
Obje Türü:				
Kodu:	Adı:			
B.B.2.04	Gıda Zehirlenmesi			
Tanım:				
Mekansal Referans Türü: Alansal				
Adı:				
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)			
ADI	Adı (Resmi Adı)			
Öznitelikleri:				
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi
OÇK		Oluşan Can Kaybı	0:1	integer
		Zehirlenmede oluşan can kaybı yazılacaktır.		
OPC		Oluşan Personel Can Kaybı	0:1	integer
		Zehirlenmede oluşan personel can kaybı yazılacaktır.		
OOC		Oluşan Öğrenci Can Kaybı	0:1	integer
		Zehirlenmede oluşan öğrenci can kaybı yazılacaktır.		
OTR		Oluş Yeri	1:1	varchar
		Zehirlenmenin kampüsün neresinde olduğu yazılacaktır.		
NDN		Nedeni	1:1	varchar
		Gaz zehirlenmesinin nedeni yazılacaktır.		

TABİS Obje Kataloğu			Versiyon 3.2: 20/10/2002		
B.Afet Yönetimi Obje Alanları:			Obje Grubu:		
Kodu:	Adı:		Kodu:	Adı:	
B.B.	Risk Bölgeleri		B.B.2.	Teknolojik Afet Riski	
Obje Türü:					
Kodu:	Adı:				
B.B.2.06	Bina Yangını				
Tanım:					
Mekansal Referans Türü: Noktasal					
Adı:					
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)				
ADI	Adı (Resmi Adı)				
Öznitelikleri:					
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi	
OÇK		Oluşan Can Kaybı	0:1	integer	
		Yangında oluşan can kaybı yazılacaktır.			
<i>OPC</i>		<i>Oluşan Personel Can Kaybı</i>	<i>0:1</i>	<i>integer</i>	
		Yangında oluşan personel can kaybı yazılacaktır.			
<i>OOC</i>		<i>Oluşan Öğrenci Can Kaybı</i>	<i>0:1</i>	<i>integer</i>	
		Yangında oluşan öğrenci can kaybı yazılacaktır.			
<i>OYR</i>		<i>Oluş Yeri</i>	<i>1:1</i>	<i>varchar</i>	
		Yangının kampüsün neresinde olduğu yazılacaktır.			
<i>NDN</i>		<i>Nedeni</i>	<i>1:1</i>	<i>varchar</i>	
		Yangının nedeni yazılacaktır.			

TABİS Obje Kataloğu			Versiyon 3.2: 20/10/2002	
B.Afet Yönetimi Obje Alanları:			Obje Grubu:	
Kodu:	Adı:		Kodu:	Adı:
B.B.	Risk Bölgeleri		B.B.3.	Politik Afet Riski
Obje Türü:				
Kodu:	Adı:			
B.B.3.01	Terörist Saldırı			
Tanım:				
Mekansal Referans Türü: Alansal				
Adı:				
Kısaltması	Adı (Veri Tipi: string, Kardinallik: 0:1)			
ADI	Adı (Resmi Adı)			
Öznitelikleri:				
Kısaltması	Kodu	Adı	Kardinallik	Veri Tipi
OÇK		Oluşan Can Kaybı	1:1	integer
		Saldırıda oluşan can kaybı yazılacaktır.		
OPC		<i>Oluşan Personel Can Kaybı</i>	0:1	integer
		Saldırıda oluşan personel can kaybı yazılacaktır.		
OOC		<i>Oluşan Öğrenci Can Kaybı</i>	0:1	integer
		Saldırıda oluşan öğrenci can kaybı yazılacaktır.		
OYR		<i>Oluş Yeri</i>	1:1	varchar
		Saldırının kampüsün neresinde olduğu yazılacaktır.		

TABİS Obje Kataloğu		Versiyon 3.2: 20/10/2002		
Eklenen Özniteliğin Ait Olduğu:				
B.Afet Yönetimi Obje Alanları:		Obje Grubu:		
Kodu:	Adı:	Kodu:	Adı:	
B.C.	Kritik Tesisler	B.C.0.	Kritik Tesisler	
B.G.	Planlar	B.G.3.	Uygulama İmar Planı	
Obje Türü:				
Kodu:	Adı:			
B.C.0.02	Diğer Sağlık Kurumu			
B.C.0.14	Acil durum Yönetim Merkezleri			
B.C.0.17	Sivil Toplum Kuruluşu			
B.C.0.23	Haberleşme Kurumu			
B.C.0.26	Sivil Savunma			
B.C.0.32	İnşaat Malzeme Deposu			
B.C.0.34	Okul			
B.C.0.37	Banka			
B.C.0.44	Büyük Mutfak			
B.C.0.45	Sosyal Barınma Tesis			
B.G.3.01	Konut			
B.G.3.02	Ticaret			
B.G.3.03	Karışık			
B.G.3.05	Sosyal Donatımlar			
B.G.3.06	Depolama			
B.G.3.07	Rekreasyon			
B.G.3.09	Stadyum			
B.G.3.10	Fuar			
B.G.3.17	Ulaşım Tesis			
B.G.3.22	Atık Madde Depolama			
B.G.3.24	Elektrik Santrali			
Eklenen Ortak Öznitelikleri:				
Kısaltması	O.Türü Kodu	Adı	Kardinallik	Veri Tipi
YKP	B.C.0.02	Yatak Kapasitesi	1:1	İnteger
		Sağlık Kurumunun Yatak Kapasitesi Yazılacaktır.		
SYP	B.C.0.15	Sığınak Yapısı	1:1	Varchar
		Sığınakın Yapısı Yazılacaktır.		
KST	B.C.0.16	Koruma Sağladığı Tehlikeler	1:1	Varchar
		Sığınakın hangi tür tehlikelere karşı koruma sağlayabildiği yazılacaktır.		

YEA	B.C.0.18	Yayın Erişim Alanı	1:1	İnteger
		Yapılan yayının alınma mesafesi Km biriminde yazılacaktır.		
ASH	B.C.0.23	Sağlayabileceği Haberleşme	1:1	Varchar
		Haberleşme kurumunun afet anında sağlayabileceği yayın türü yazılacaktır.		
ASK	B.C.0.26	Sağlayabileceği Katkılar	1:1	Varchar
		Sivil savunma kurumunun afet yerinde katılabileceği görevler yazılacaktır.		
YYK	B.C.0.28	Yiyecek Kapasitesi	1:1	İnteger
		Markette bulunan yiyecek kapasitesi yazılacaktır.		
ICK	B.C.0.28	İçecek Kapasitesi	1:1	İnteger
		Markette bulunan içecek kapasitesi yazılacaktır.		
SUK	B.C.0.28	Sağlık Ürünü Kapasitesi	1:1	İnteger
		Markette bulunan sağlık ürünü kapasitesi yazılacaktır.		
EUK	B.C.0.28	Enerji Ürünü Kapasitesi	1:1	İnteger
		Markette bulunan pil vb. enerji ürünü kapasitesi yazılacaktır.		
YKS	B.C.0.29	Yararlanabilecek Kişi Sayısı	1:1	İnteger
		Gıda deposundaki gıda maddelerinden yararlanabilecek kişi sayısı yazılacaktır.		
ZGM	B.C.0.30	Zararlı Etki Mesafesi	1:1	İnteger
		Afet sırasında ve sonrasında depodan yayılacak tehlikeli maddenin etki mesafesi Km biriminde yazılacaktır.		
ZGT	B.C.0.30	Zararlı Etki Türü	1:1	Varchar
		Afet sırasında ve sonrasında depodan yayılacak tehlikeli maddenin ne türde bir etki yapabileceği yazılacaktır.		
ISK	B.C.0.33	İçerdiği Su Miktarı	1:1	İnteger
		Depoda bulunan su miktarı litre biriminde yazılacaktır.		
BKS	B.C.0.45	Barınan Kişi Sayısı	1:1	İnteger
		Tesiste barınan kişi sayısı yazılacaktır.		

CCN	B.G.3.01	Çocuk Nüfusu	1:1	İnteger
		Konutta ikamet eden çocuk nüfusu yazılacaktır.		
YSN	B.G.3.01	Yaşlı Nüfusu	1:1	İnteger
		Konutta ikamet eden yaşlı sayısı yazılacaktır.		
ERN	B.G.3.01	Erişkin Nüfusu	1:1	İnteger
		Konutta ikamet eden yetişkin sayısı yazılacaktır.		
YKS	B.G.3.05	Yararlanan Kişi Sayısı	1:1	İnteger
		Donatımlardan yararlanan toplam kişi sayısı yazılacaktır.		
SYK	B.G.3.09	Seyirci Kapasitesi	1:1	İnteger
		Tesisin toplam seyirci kapasitesi yazılacaktır.		
ZYK	B.G.3.10	Ziyaretçi Kapasitesi	1:1	İnteger
		Tesisin toplam ziyaretçi kapasitesi yazılacaktır.		
SEM	B.G.3.24	Sağladığı Enerji Miktarı	1:1	İnteger
		Santralin ürettiği enerji miktarı Kwatt/Saat biriminde yazılacaktır.		
CPS	B.G.3.02	Çalışan Personel Sayısı	1:1	İnteger
	B.G.3.03 B.G.3.05 B.G.3.06 B.G.3.07 B.G.3.09 B.G.3.10 B.G.3.17 B.G.3.22 B.G.3.24	Kurumda ya da tesiste çalışan personel sayısı yazılacaktır.		

AGS	B.C.0.14	Acil Durum Görevlisi Sayısı	1:1	Integer
	B.C.0.17 B.C.0.23 B.C.0.26 B.C.0.32 B.C.0.34 B.C.0.37 B.G.3.01 B.G.3.02 B.G.3.03 B.G.3.05 B.G.3.06 B.G.3.07 B.G.3.09 B.G.3.10 B.G.3.17 B.G.3.22 B.G.3.24	Tesiste ya da kurumda acil durum anında görevleri belirtilen kişi sayısı yazılacaktır.		
ADG	B.C.0.14	Acil Durum Görevi	1:1	Varchar
	B.C.0.17 B.C.0.23 B.C.0.26 B.C.0.32 B.C.0.34 B.C.0.37 B.G.3.01 B.G.3.02 B.G.3.03 B.G.3.05 B.G.3.06 B.G.3.07 B.G.3.09 B.G.3.10 B.G.3.17 B.G.3.22 B.G.3.24	Acil durum anında tesiste ya da kurumda görevlilere hangi acil durum görevleri verildiği yazılacaktır.		
	Kodu			
	100	Arama Kurtarma		
	101	Lojistik		
	102	İlkyardım		
	999	Diğer		

ÖZGEÇMİŞ

Himmet KARAMAN, 1979 yılında Kütahya’da doğdu. Lise öğrenimini Ankara Ayrancı Süper Lisesinde tamamladıktan sonra, 1997 yılında İ.T.Ü. İnşaat Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümünde lisans eğitimine başladı. 2001 yılında eğitimini tamamladıktan sonra aynı yıl İ.T.Ü. İnşaat Fakültesi Jeodezi ve Fotogrametri Mühendisliği Bölümü, Ölçme Tekniği Anabilim Dalında Araştırma Görevlisi olarak göreve başladı. Aynı yıl, İ.T.Ü. Fen Bilimleri Enstitüsü, Jeodezi ve Fotogrametri Anabilim Dalında yüksek lisans eğitimine başladı. Halen bu öğrenimine devam etmektedir.