

**JEODEZİK VERİ TABANI TASARIMI
ve WEB TABANLI YÖNETİMİ**

YÜKSEK LİSANS TEZİ
Mih. Özgür ÖZASLAN
(501001753)

Tezin Enstitüye Verildiği Tarih: 5 Mayıs 2003
Tezin Savunulduğu Tarih: 30 Mayıs 2003

Tez Danışmanı : Doç. Dr. Rahmi Nurhan ÇELİK (İ. T. Ü)

Diğer Jüri Üyeleri : Prof. Dr. Rasi m DENİZ (İ. T. Ü)

Prof. Dr. Onur GÜRKAN (B. Ü)

MAYIS 2003

ÖNSÖZ

Bilginin güç olduğu günümüz dünyasında, bu güç kadar bu güce ulaşım hızı ve gücün düzenliliği de büyük önem taşımaktadır. Bu çalışmada Jeodezi ve Fotogrametri Mühendisliği alanında, jeodezik ölçme yöntemleriyle elde edilen bilgilerde bu hız ve etkin kullanımın nasıl sağlanabileceği anlatılmaya ve örnek bir uygulamayla gösterilmeye çalışılmıştır.

Çalışmalarımın her aşamasında engin bilgilerinden yararlandığım yaratıcı fikirleriyle çalışmayı güzelleştiren ve bana her zaman destek olan danışman hocam sayın Doç. Dr. Rahmi Nurhan ÇELİK'e sonsuz teşekkürlerimi sunarım

Çalışmalarım sırasında mesleki bilgi ve deneyimlerinden faydalandığım dostluğundan destek aldığı arkadaşlarımla ve meslektaşlarımla görevlileri Yük. Müh. Bülent ÖZÖNER EROL ve Yük. Müh. Serdar EROL başta olmak üzere tüm Jeodezi Anabilim Dalı ailesine teşekkürlerimi sunarım

Şu anda içerisinde yer aldığı fiziksel ve teknik altyapısından yararlandığı İTÜ İnşaat Fakültesi Bilgi İşlem Merkezi'ne ve hayatımın her aşamasında desteklerinden benden esirgeyeyen sevgili aileme teşekkür ederim

Yapılan çalışmamın mesleğimize katkı vermesi ve bu alanda daha sonradan çalışacak olan meslektaşlarıma yol göstermesi dileklerime...

Mayıs 2003

Özgür ÖZASLAN

İÇİNDEKİLER

| | |
|---|-------------|
| KISALTMALAR | v |
| TABLO LİSTESİ | vi |
| ŞEKİL LİSTESİ | vii |
| ÖZET | viii |
| SUMMARY | x |
| 1. GİRİŞ | 1 |
| 2. VERİ TABANLARI ve YAPISAL SORGULAMA DİLİ | 4 |
| 2.1. Veri Tabanı ve Temel Kavramlar | 4 |
| 2.2. Veri Tabanı Tasarımı | 7 |
| 2.3. Veri Modelleri ve Veri Yapıları | 9 |
| 2.3.1. Kavramsal Veri Modelleri | 9 |
| 2.3.2. Fiziksel Veri Modelleri | 12 |
| 2.4. Yapısal Sorgulama Dili (SQL) | 13 |
| 3. VERİ TABANI, WEB PROGRAMLAMA DİLİ ve İLİŞKİLERİ | 17 |
| 3.1. MySQL Veri Tabanı | 17 |
| 3.2. PHP Web Programlama Dili | 18 |
| 3.3. PHP ve MySQL İlişkisi | 20 |
| 4. JEODEZİK VERİ TABANI | 23 |
| 4.1. Jeodezik Veri Tabanı Gereksinimi | 23 |
| 4.2. Jeodezik Veri Tabanı İçin Genel Tablolar | 25 |
| 4.2.1. Proje Tablosu | 26 |
| 4.2.2. İstasyon Tablosu | 26 |
| 4.2.3. Kurum Tabloları | 27 |
| 4.2.4. Personel Tablosu | 28 |
| 4.2.5. Alet Tablosu | 28 |
| 4.2.6. Meteoroloji Tablosu | 29 |
| 4.3. Yatay Doğrultü Ölçmeleri Tabloları | 30 |
| 4.4. Uzunluk Ölçmeleri Tabloları | 31 |
| 4.5. Düşey Açı Ölçmeleri Tabloları | 33 |
| 4.6. Yükseklik Ölçmeleri Tabloları | 34 |
| 4.7. GPS Ölçmeleri Tabloları | 34 |

| | |
|---|-----------|
| 5. ÖRNEK GPS VERİ TABANI ve WEB YÖNETİM ARAYÜZÜ TASARIMI | 39 |
| 5.1. Çalışmaya Genel Bakış | 39 |
| 5.2. GPS Veri Tabanı Tasarımı | 40 |
| 5.3. Web Yönetim Arayüzü Tasarımı | 46 |
| 6. SONUÇLAR VE ÖNERİLER | 50 |
| KAYNAKLAR | 53 |
| ÖZGEÇMİŞ | 55 |

KISALTMALAR

| | |
|--------------------|-------------------------------------|
| CBS | : Coğrafi Bilgi Sistemi |
| JeoVT- VL.0 | : Jeodezik Veri Tabanı Versiyon 1.0 |
| GIS | : Geographic Information Systems |
| PHP | : Personal Home Page Tool |
| SQL | : Structured Query Language |
| VTYS | : Veri Tabanı Yönetim Sistemi |
| VTS | : Veri Tabanı Sistemi |
| ODBC | : Open Database Connectivity |
| CPU | : Central Processing Unit |
| HTML | : Hyper Text Markup Language |
| ASP | : Active Server Pages |
| JSP | : Java Server Pages |
| CFS | : Coldfusion |

TABLO LİSTESİ

| <u>No</u> | | <u>Sayfa</u> |
|--------------------|--|--------------|
| Tablo 2.1. | Mantıksal operatörler..... | 15 |
| Tablo 2.2. | Karşılaştırma Operatörleri..... | 15 |
| Tablo 4.1. | Proje bilgilerinin tutulacağı tablo (proje)..... | 26 |
| Tablo 4.2. | İstasyon bilgilerinin tutulacağı tablo (istasyon)..... | 27 |
| Tablo 4.3. | Kurum bilgilerinin tutulacağı tablo (kurum)..... | 28 |
| Tablo 4.4. | Kuruma ait birim bilgilerinin tutulacağı tablo (birim)..... | 28 |
| Tablo 4.5. | Proje-kurum ilişkilerini sağlayacak ara tablo (proje_kurum)..... | 28 |
| Tablo 4.6. | Personel bilgilerinin tutulacağı tablo (personel)..... | 29 |
| Tablo 4.7. | Alet bilgilerinin tutulacağı tablo (alet)..... | 29 |
| Tablo 4.8. | Meteorolojik bilgilerinin tutulacağı tablo (meteoroloji)..... | 30 |
| Tablo 4.9. | Yatay doğrultü ölçmeleri ölçme tablosu (yatay_dog_olcu)..... | 30 |
| Tablo 4.10. | Yatay doğrultü ölçmeleri ölçme tablosu (yatay_dog_olcu)..... | 31 |
| Tablo 4.11. | Yatay doğrultü ölçmeleri ölçme alet ilişkilerini sağlayacak ara tablo (yataydogolcu_alet)..... | 31 |
| Tablo 4.12. | Uzunluk ölçmeleri ölçme tablosu (uzunluk_olcu)..... | 32 |
| Tablo 4.13. | Uzunluk ölçmeleri ölçme tablosu (uzunluk_olcu)..... | 32 |
| Tablo 4.14. | Uzunluk ölçmeleri ölçme alet ilişkilerini sağlayacak ara tablo (uzunlukolcu_alet)..... | 32 |
| Tablo 4.15. | Düşey açı ölçmeleri ölçme tablosu (dusey_aci_olcu)..... | 33 |
| Tablo 4.16. | Düşey açı ölçmeleri ölçme tablosu (dusey_aci_olcu)..... | 34 |
| Tablo 4.17. | Düşey açı ölçmeleri ölçme alet ilişkilerini sağlayacak ara tablo (duseyaci_olcu_alet)..... | 34 |
| Tablo 4.18. | Yükseklik ölçmeleri ölçme tablosu (yuk_olcu_olcu)..... | 35 |
| Tablo 4.19. | Yükseklik ölçmeleri ölçme tablosu (yuk_olcu_olcu)..... | 35 |
| Tablo 4.20. | GPS ölçmeleri ölçme tablosu (gps_olcu)..... | 36 |
| Tablo 4.21. | GPS ölçmeleri ölçme tablosu (gps_olcu)..... | 37 |
| Tablo 4.22. | GPS ölçmeleri değerlendirme tablosu (gps_degerlendirme)..... | 37 |
| Tablo 4.23. | GPS ölçmeleri ölçme alet ilişkilerini sağlayacak ara tablo (gps_olcu_alet)..... | 37 |
| Tablo 4.24. | GPS ölçmeleri ölçme personel ilişkilerini sağlayacak ara tablo (gps_olcu_personel)..... | 37 |
| Tablo 4.25. | GPS ölçmeleri değerlendirme-personel ilişkilerini sağlayacak ara tablo (gps_degerlendirme_personel)..... | 38 |
| Tablo 5.1. | Verilerin Niteliklerine Göre Gruplandırılması..... | 41 |

ŞEKİL LİSTESİ

| | <u>Sayfa No</u> |
|---|-----------------|
| Şekil 2.1 : Veri Tabanı Sistemi..... | 5 |
| Şekil 2.2 : Veri Tabanı Tasarımı..... | 8 |
| Şekil 2.3 : Üç Şe nâ M nârisi. | 9 |
| Şekil 5.1 : “Qdesi gner” veri tabanı tasarımı yazılı m ekran görüntüsü..... | 40 |
| Şekil 5.2 : Kavramsal Veri Modeli..... | 43 |
| Şekil 5.3 : Fiziksel Veri Modeli..... | 45 |
| Şekil 5.4 : phpMyAdmin MySQL Veri tabanı yöneti marayüzü..... | 47 |
| Şekil 5.5 : İstasyon için kayıt giriř formu örneđi..... | 48 |

JEODEZİK VERİ TABANI TASARIM ve WEB TABANLI YÖNETİM

ÖZET

Verilerin organizasyonu, sunumu ve kullanılması bakımından önemli bir görev üstlenen Coğrafi Bilgi Sistemleri (CBS)'nin temel bileşeni olan veri kadar, verilerin tutulma biçimleri de CBS'nin sürdürülebilirliği ve etkin kullanımı açısından büyük önem taşımaktadır. CBS ile iç içe olan mesleğimizde veri tabanlarının ve bu veri tabanlarında verilerin tutulma biçimlerinin önemi her geçen gün daha da iyi anlaşılmaktadır.

Bu tezde yapılan çalışma doğrudan CBS ile ilgili olmakla birlikte, CBS'nin temel altlığı olan haritanın üretim ve güncelleştirme aşamalarında kullanılan jeodezik ölçümlerin ilişkisel veri tabanı ortamında nasıl tutulabileceğini, bu veri tabanının yönetimi için web tabanlı bir arayüzün nasıl hazırlanabileceğini anlatmaktadır. Diğer taraftan bu çalışmada anlatılan veri modelleri ve veri yapıları, CBS yazılımlarında veya CBS yazılımları ile kullanılabilen veri tabanı yazılımlarında da kullanıldıklarından CBS'nin veri tabanı bileşeninin anlaşılmasında faydalı olacaklardır.

Bu çalışmada jeodezik veri tabanı kavramsal olarak ilişkisel veri modeli kullanılarak tasarlanmış ve sonuçta ortaya çıkan fiziksel model MySQL veri tabanı yazılımına aktarılmıştır. MySQL'e aktarılan veri tabanı yapısının yönetimi için PHP web programlama dili kullanılarak kullanıcı arayüzleri hazırlanmış, veri tabanı web'e açılabilir hale getirilmiştir. Hazırlanan veri tabanı yapısı ve PHP kullanıcı arayüzünden oluşan yazılım paketi JeoVT- VI.0 (**Jeodezik Veri Tabanı Versiyon 1.0**) olarak isimlendirilmiştir.

Hazırlanan bu yazılım sayesinde, jeodezik ölçme yöntemlerine ilişkin verilere gerektiğinde fazla sayıda ilgili kullanıcının doğru, güvenilir, hızlı şekilde ulaşılabilmesi sağlanmış ve bu verilerin saklanması, sürdürülebilirliği daha kolay hale getirilmiştir. Yapılan bu çalışma ile güncel veri tabanı tasarım teknikleri kullanılarak mesleğimizdeki jeodezik ölçme verilerinin tutulması ve bu veri

tabanının web programlama dilleri kullanılarak web'e açılması sağlanmıştır. Bu çalışma, ülkemizde mesleğini bu konuda yapılan ilk uygulamalarından biri olma özelliğini taşımaktadır.

A GEODETIC DATABASE DESIGN and WEB BASED MANAGEMENT

SUMMARY

Database design architecture, which is as important as data, is very important for effective usage and sustainability of Geographic Information Systems (GIS) and it is also a fundamental component of GIS. Understanding of the importance of databases and database design architecture for GIS, increases day by day in our profession.

The study in this thesis is not directly connected with GIS but organizing geodetic measurement data, that is used for creating and updating the base map for GIS, in relational database architecture and creating web based data management interface for managing this data are mentioned. On the other hand, data models and data structures, which are explained in this study, are useful because these models and structures are also used in GIS software or GIS related database software.

In this study, a relational geodetic database was modeled by using relational data model in conceptual level and finally the physical data model, which was modeled by using conceptual data model, was transferred to MySQL database software. A web based data management interface for managing the data, which was transferred to MySQL database software, was created by using PHP web scripting language. The database architecture and the PHP interface are named as JeoVT- V1.0 (**Jeodezik Veri Tabanı Versi on 1.0**) software package.

With this software package, authorized multi-users can access geodetic measurement data in a correct, reliable, economical and fast way when they need. Also archiving and preserving of this data is made easy with this software package. By means of this software, collecting the geodetic measurement data in a database and launching it at the web using web scripting languages might be one of the any first applications in our profession in Turkey.

1. GİRİŞ

Bilgi kadar bilginin saklanması fikri de oldukça eskidir. Çok eski tarihlerde bile insanların bilgileri, saklamak amacıyla kil tabletlere kazıdıkları bilinmektedir. Daha sonraları bilgiler kitaplarda tutulmaya başlanmış ve 20. yüzyılın ortalarında bilgisayarın kullanılmaya başlanmasıyla bilgisayar üzerinde tutulan veri tabanlarının ve veri tabanı yazılımlarının kullanılmasına geçilmiştir. Verilerin tutulma ve sunulma biçimleri zamanla değişmiş olsa da, veri tabanlarının kullanılma nedeni dün de bugün de aynıdır. İnsanların sınırlı hatırlama ve akılda tutma kapasiteleri nedeniyle bilgileri bir veri tabanında tutmak kaçınılmazdır. Ayrıca bilginin özel veya kamuya sunumu yani bilginin paylaşım ve bu bilgilere hızlı, doğru, eksiksiz erişimde veri tabanlarıyla mümkün olmaktadır. [1]

Günümüzde veri tabanı tasarımı ve veri tabanı yönetimi alanları ayrı birer uzmanlık alanı haline gelmiştir. Veri tabanı teknolojisi, bilgisayar donanımı ve yazılımlarının gelişmesine paralel bir seyirle her geçen gün yeni özellikler kazanarak ilerlenmektedir. Bu günün veri tabanı teknolojisi kullanıcıya özel uygulamalar için esnek veri organizasyonu, daha ucuz, daha hızlı veri depolama ortamı ve araçları sağlamaktadır.

Veri tabanı sistemlerinin tasarımı yapısı oldukça karmaşıktır ve uzmanlık gerektirir. Veri tabanı tasarımı için geçmişte hiyerarşik ve ağ veri modelleri kullanılırken günümüzde artık yaygın olarak ilişkisel veri modeli kullanılmaktadır. E. F. Codd'un 1970'de yazmış olduğu "A Relational Model of Data for Large Shared Data Banks" makalesi ile ilişkisel veri yapılarında büyük bir ilerleme kaydedilmiş; hiyerarşik veri tabanı yapılarından, ilişkisel veri tabanı yapılarına doğru bir geçiş yaşanmıştır [2].

Jeodezi alanında veri tabanı teknolojileri ile en çok karşı karşıya gelme Coğrafi Bilgi Sistemlerinin (CBS) ortaya çıkma sürecinden başlayarak, bugüne kadar olan gelişim süreci içerisinde olmuştur. Jeodezi ve Fotogrametri Mühendisliği alanında veri tabanlarının ve bu veri tabanlarında verilerin tutulma biçimlerinin önemi, CBS ile olan yakın ilişkiler sonucunda ve CBS'nin gelişimine paralel olarak her geçen gün daha da iyi anlaşılmıştır.

CBS ile birlikte CBS'nin dışında kalan ve doğrudan mesleği mizi ilgilendiren bazı veri tabanları, bu veri tabanlarının tasarımı CBS'nin popülaritesi yanında biraz ihmal edilmiştir. Bu çalışmada bu bakımdan su altında kalmış bu çalışmaları, su yüzüne çıkarılmaktadır. Bu çalışma doğrudan Jeodezi ve Fotogrametri Mühendislerinin kullanımına yönelik olarak hazırlanmış, jeodezi de kullanılan farklı ölçme yöntemleri ele alınarak, bu yöntemler sonucunda elde edilen ölçme ve değerlendirme sonuçlarının ilişkisel bir veri yapısı içerisinde nasıl tutulabileceği anlatılmıştır. Günümüzde uygulanan farklı jeodezik ölçme tekniklerinin gerektirdiği, zamana bağımlı ve homojen olmayan veri gruplarının saklanacağı bir veri tabanının tasarısı geliştirilmeye çalışılmıştır. Diğer taraftan özet bölümünde de değinildiği gibi, bu çalışmada anlatılan veri modelleri ve veri yapıları, CBS yazılımlarında veya CBS yazılımları ile kullanılabilen veri tabanı yazılımlarında da kullanıldıklarından CBS'nin veri tabanı bileşeninin anlaşılmasında faydalı olacaklardır.

Çalışmanın bir diğer boyutu da hazırlanan veri tabanının çok fazla kullanıcı tarafından kullanılabilecek web tabanlı bir yönetim arayüzü ile desteklenmiş olmasıdır. Bilgiye ilgili kişilerin, ilgili oldukları ve izin verildiği düzeyde, istedikleri yerden ulaşabilmelerini sağlamak amacıyla böyle bir yol izlenmiştir.

Tezin ikinci bölümünde veri tabanları ilgili temel kavramlar verildikten sonra, veri tabanlarını kullanmanın getirdiği avantajlar, veri modelleri, veri yapıları, veritabanı tasarımı ve yapısal sorgulama dili (SQL) anlatılmaya çalışılmıştır.

Üçüncü bölümde uygulamada kullanılan MySQL veri tabanı yazılımı ve PHP web programlama dili hakkında kısa bilgiler verilmiş, PHP ve MySQL'in birbirleriyle olan ilişkisi gösterilmek istenmiştir.

Dördüncü bölümde ise jeodezik veri tabanına neden gereksinim duyulduğu, jeodezik bir veri tabanında yer alabilecek bilgilerin nelere olabileceği anlatılarak, ölçme yöntemlerine göre bu bilgiler sınıflandırılmış ve bu sınıflandırma sonucunda elde alınan ölçme yöntemleri için ilişkisel veri tabanı tablo yapıları oluşturulmuştur.

Tezin beşinci bölümünde, dördüncü bölümde listelenen tablo yapılarının ne şekilde oluşturulduğunun daha net anlaşılması için GPS ölçme yöntemi ele alınarak veri tabanı tasarımı ve veri girişi arayüzü tasarımı konuları biraz daha ayrıntılı olarak anlatılmıştır.

Son olarak tez bu çalış ma sonrası nda elde edilen sonuçlar ve öneriler bölümü ile son bul maktadır.

2 VERİ TABANLARI ve YAPISAL SORGULAMA DİLİ

2.1 Veri Tabanı ve Temel Kavramlar

Veri tabanı, Türk Dil Kurumu sözlüğünde “Bilgisayar kullanımında çözüme erişmek için işlenebilir duruma getirilmiş bilgi ortamı” olarak tanımlanmaktadır. Bilgi dünyasında ise bu tanımlama, küçük farklılıklar göstermekle beraber, “Büyük boyuttaki verilerin gerektiğinde hızlı bir şekilde sorgulanabilmelerini ve kalıcılığını sağlamak amacıyla bilgisayar ortamında oluşturulmuş bir tür koleksiyondur.” şeklinde [3].

Aşağıda veri tabanı ile ilişkili temel bazı kavram ve tanımlara yer verilmiştir. Bu kavram ve tanımlardan sonra veri tabanının ne olduğu daha net olarak anlaşılacaktır.

Bir veri tabanı ;

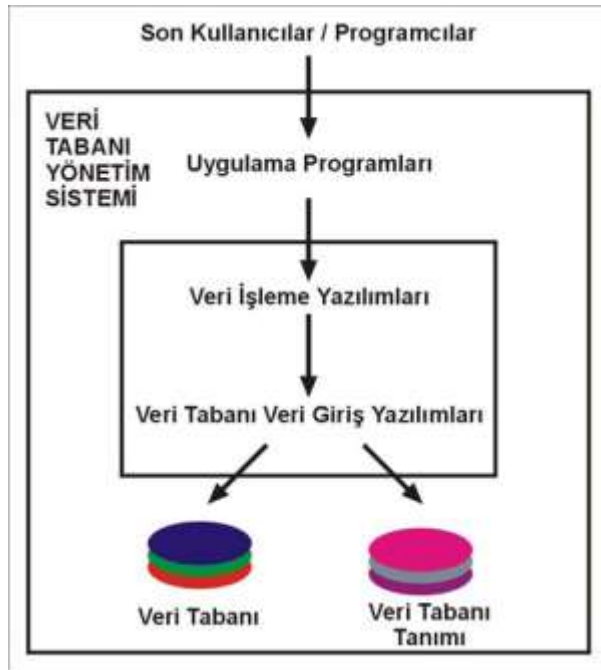
- Şemalar,
- Tablolar,
- Kolonlar (Alanlar),
- Birincil anahtarlar (Primary keys, <pk>) ve yabancı anahtarlar (foreign keys, <fk>),

gibi bölümlerden oluşur.

Veri tabanı içindeki tabloların yerleştirme ve biçimlenmesine şema denir. Kolonlar (columns), veri tabanında bilgi içeren sütunlardır. Her kolon bir bilgi içerir ve içerdikleri bilgilere göre sınıflandırılır. Kolonlar bir araya gelerek tabloları (table) oluştururlar. Tablodaki her bir satır ise ayrı bir veri grubunu gösterir ve kayıt (record, tuple) olarak adlandırılır. Birincil anahtar (<pk>) tablodaki her bir satırın tekliliğini tanımlar. Bilgilere hızlı ulaşmayı ve tablolar arasında bağlantı kurmayı sağlar. Birincil anahtar olarak genellikle ID kolonları seçilir. Bır ya da daha çok kolon aynı anda birincil anahtar olarak seçilebilir, bu amaca göre değişir. Bir tablodaki birincil anahtarın diğer bir tabloda eşleştiği kolona yabancı anahtar (fk) denir. Ayrıca yabancı anahtar kayıtlı olmayan bilgiyi girme izi engeller. [4]

“Bir Veri Tabanı Yönetim Sistemi (VTYS), veri tabanı kurma ve

yaşatma amacıyla geliştirilmiş programlar bütünüdür. Veri tabanı ve Veri Tabanı Yönetim Sistemi yazılımları birlikte, bir Veri Tabanı Sistemi (VTS) oluşturur (Şekil 2.1). Veri tabanının kurulması, veri tabanında yer alacak veri tiplerinin belirlenmesini, tanımlanmasını ve verinin depolanmasını kapsar. Verinin güncelleştirilmesi, sorgulanması ve istenen biçimde "bilgi" olarak kullanıcıya sunulması ise veri tabanının yaşatılması sürecinde yer alan işlemlerdir. Bir veri tabanı yönetim sisteminin temel işlevi program-veri bağımsızlığını sağlamak, yani veri yönetimini uygulamalardan ayıraktır.” [5] Veri Tabanı Yönetim Sistemleri, zamanla çok daha fazla özellikler kazanmaktadır.



Şekil 2.1 Veri Tabanı Sistemi (VTS)

“Program-veri bağımsızlığı gereksinimi, veri tabanlarını doğuran en önemli nedendir. Her bir programın kendi verisini ayrıca organize etmesi yerine, veri sadece veri tabanında, farklı programların kullanımına olanak sağlayacak ortak bir tanımlı altında tutulur. Bu durumda uygulama programlarının, gereksinim duydukları veri veri tabanından çekebilme ve veri tabanına veri koyabilmeleri için veri tabanının yapısını bilmeleri gerekir.” [5]

“Veri tabanının tanımlı sistemin veri sözlüğünde bulunur. Veri sözlüğü, veri dosyalarının yapısı, veri elemanlarının tipi ve depolama formatı ile veri üzerindeki kısıtlayıcılar gibi bilgiler içerir. Veri sözlüğünde yer alan bilgiler meta-veri olarak adlandırılır. Bununla birlikte, veri tabanı sistemleri ile güdülen amaç, veri yönetimini

uygulamalardan ayrılmak istediği için, veri tabanına veri girme ve veri tabanından veri çekme ile veri işleme işlevleri birbirinden ayrılmıştır. Diğer bir ifadeyle verinin fiziksel yapısı ve veriye ulaşım yöntemi, veri işleme programlarından gizlenir. Dolayısıyla çeşitli uygulama programları, veri tabanına doğrudan ulaşmak yerine, bu işlemi veri tabanı yönetim sisteminin sağladığı yazılımlar üzerinden gerçekleştirir. O nedenle, uygulama programlarının veri tabanının yapısı hakkında bilinmesi gerekenler önemli ölçüde azaltılmıştır.” [5]

“Veri yönetiminin uygulamalardan ayrılması ve böylece veri paylaşımının, yani farklı programların aynı veriyi kullanabilmelerini sağlanması için, bir veri tabanı yönetim sisteminin veri tabanına eşzamanlı giriş, veri fazlasının kontrolü, veri bütünlüğü ve veri tabanının güvenliğinin sağlanması gibi fonksiyonları yerine getirmesi gerekir. Ayrıca bir veri tabanı yönetim sisteminin farklı yüzeylerde kullanıcılar için ara yüzler sunması beklenir.” [5]

“Veri tabanına eşzamanlı giriş, farklı uygulamaların aynı anda veri tabanına girebilmesi ve veri tabanında değişiklikler yapabilmesi açısından son derece önemlidir. Ancak farklı kullanıcıların aynı anda aynı kayda girmek istemeleri sorun yaratmaktadır. Eşzamanlı girişin kontrolü için değişik teknikler kullanılmaktadır. Belli bir dereceye kadar veri fazlalığı verimlilik için gerekli olabilir. Ancak kontrolsüz veri fazlalığı güncelleştirme açısından problem olacak ve veri tabanının bütünlüğünü sürekli tehdit edecektir. Veri bütünlüğü dendiğinde veri tabanındaki verinin doğruluğu ve tutarlılığı anlaşılır.” [5]

Veritabanları, yukarıda da belirtildiği gibi, verilerin saklanması ve yönetilmesi için kullanılmaktadır. Küçük bilgiler için kitaplar veya bilgisayarda kullanılan metin dosyaları yeterli olacaksa, bunun için veri tabanı kullanılması gerekmez. Veri sayısı ve bu verilerin boyutu arttığında ve buna ek olarak hızlı ulaşım aynı anda çoklu ulaşım kolay güncellenebilirlik ve kişiye göre erişim hakkı gibi konular da söz konusu olduğunda veritabanlarını kullanmak kaçınılmazdır. [4] Veritabanlarının başlıca kullanım nedenlerini maddeler halinde sıralayacak olursak aşağıdaki gibi gösterilebilir:

- VTYS ile birlikte verilerin saklanması ve yönetilmesi ile ilgili konulardaki ayrıntılardan veri tabanı yöneticilerini kurtarması,
- Program veri bağımsızlığının sağlanması,

- Kayıt ekleme, güncelleme, sorgulama kolaylığı,
- İstenilen yapıda çıktı alabilme kolaylığı,
- Yedekleme kolaylığı,
- Verileri paylaşım kolaylığı,
- Verilere ulaşabilen kişi sayısının arttırılabilirliği,
- Verilere ulaşımın kişilere ve gruplara göre sınırlandırılabilirliği,
- Yukarıda belirtilen bütün işlemlerin diğer veri saklama yöntemleri ele alındığında oldukça hızlı yapılabilirliği dir [4].

2.2 Veri Tabanı Tasarımı

“Bir veri tabanı tasarımı, veri tabanının konusunu oluşturan gerçeğin, veri tabanının oluşturuluş gereksinimi ve beklentileri çerçevesinde soyutlanarak veri tabanına aktarılmasını kapsar. Oluşturulacak bir veri tabanı farklı kullanıcı gereksinimlerini karşılayabilmelidir.” [5]

Veri tabanı tasarımında ilk adım olan veri tabanı kullanıcı gereksinimlerinin belirlenmesidir. Söz konusu gereksinimler, veri tabanında yer alacak veri gruplarını, gruplardaki verilerin tiplerini ve verinin fiziksel olarak depolanması için kullanılacak olan veri yapılarını belirler. [5]

“Gerçeğin veri tabanındaki sayısal temsili, onun belli bir perspektiften bir modeli olup, bir veri tabanı sisteminde gerek kullanıcılar ve gerekse bilgisayar tarafından anlaşılabilir bir tarzda tanımlanması gerekir. Böyle bir tanımlama, veri tabanı literatüründe şema olarak adlandırılır. Veri tabanındaki veri, sürekli olarak işleme tabi olduğundan değişkendir. Yeni kayıtlar girilirken, eski kayıtlar silinebilir ve hatta mevcut kayıtlar değiştirilebilir. Belli bir andaki veri tabanı, bir veri tabanı örneğidir ve belirli bir şemaya uyar.” [5]

Kullanıcı ve bilgisayar düzeyleri sırasıyla "kavramsal" ve "fiziksel" düzeyler, bu düzeylerdeki şemalar da kavramsal şema ve iç şema olarak anılırlar. Kavramsal ve fiziksel düzeylerdeki şemalar, farklı anlayış mekanizmalarına hitap ettiklerinden, kullanılacak veri modelleri de farklı olacaktır. Her iki düzeyde kullanılmak üzere, çeşitli veri modelleri geliştirilmiştir. [5]

“Geleneksel veri tabanı tasarımı, kavramsal düzeyden fiziksel düzeye doğrudur (Şekil 2.2). Kavramsal tasarımda, gereksinimlere göre kavramsal şema belirlenir.

Kavramsal şema tanımlamada, kavramsal ya da mantıksal veri modelleri kullanılabilir. Kavramsal şema, ortalama veri tabanı kullanıcısı için, veri tabanının yapısını genel olarak tanımlar. Kullanıcıların veri tabanının yapısını anlamalarına ve böylece uygulamalarını modellemelerini sağlar. Kavramsal şema, fiziksel depolama yapılarının ayrıntılarına girmeden, varlıklar, veri tipleri, varlıklar arasındaki ilişki tipleri ve kısıtlayıcılar üzerinde yoğunlaşır. Bu bakımdan kavramsal şema, yüksek düzeyli bir tanımlamadır. Diğer bir ifadeyle, kavramsal şema yazılım ve donanımdan bağımsızdır.” [5]



Şekil 2.2 Veri Tabanı tasarımı

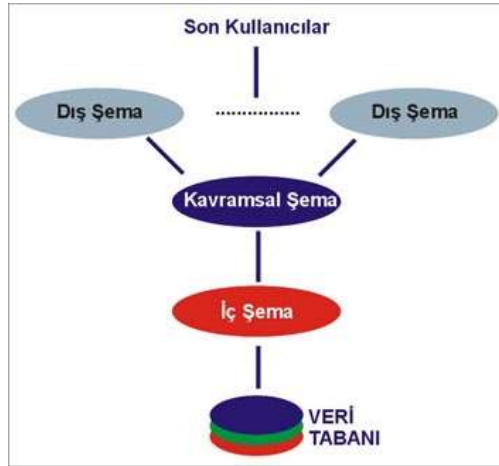
“Kavramsal veri modelleri oldukça yüksek düzeyli olduklarından, kavramsal bir veri modelinde tanımlı bir şema genellikle doğrudan gerçekleştirilemez. Bu nedenle, geleneksel veri tabanı tasarımı kavramsal tasarımdan sonraki adımla çoğunlukla gerçekleştirimi için kullanılacak bir veri tabanı yönetim sisteminin seçimidir. Öeyandan, bugün piyasadaki veri tabanı yönetim sistemlerinin çoğu mantıksal bir veri modeli kullanmaktadır. O nedenle mantıksal veri modelleri, gerçekleştirim veri modelleri olarak da bilinirler. O halde, kavramsal veri modelindeki kavramsal şema, veri tabanı yönetim sisteminin veri modelinde yeniden tanımlanmalıdır. Buradaki işlem iki veri modeli arasında bir dönüşüm olup, bazen mantıksal veri tabanı tasarımı olarak anılır.” [5]

“Fiziksel tasarıma aşmasında, verinin en yüksek verim için veri tabanında fiziksel olarak nasıl organize edilmesi gerektiği belirlenir. Sonuç, iç şemadır. İç şema depolama yapılarının, kayıt formlarının, kayıt alanlarının, veri tabanına giriş yol ve

yöntemleri ile veri tabanının fiziksel gerçekleştirimini ilgilendiren diğer bütün detayları tanımlar. İç şema tanımlamada, genellikle veri yapıları olarak bilinen, fiziksel veri modelleri kullanılır. İç şema yazılım ve donanıma bağlıdır.” [5]

“Her bir dış şema, veri tabanını belirli bir kullanıcı grubunun gereksinim duyduğu biçimde, ilgisiz detayları adeta gizleyerek tanımlar. Dış şema mekanizması, değişik kullanıcı uygulamalarının modellenmesinde, büyük bir esneklik getirmektedir.” [5]

“Bir veri tabanı sistemindeki bu dış şema, kavramsal şema ve iç şema yapısı, Amerikan Ulusal Standartları Enstitüsü (ANSI) nin ‘Üç Şema Mimarisi’ olarak bilinir (Şekil 2.3). Buna göre, dış şema ile kavramsal ve kavramsal şema ile iç şema arasında sürekli, bir dönüşüm vardır. Dış şema tanımlamada kullanılan veri modelinin, kavramsal şemanın tanımlandığı veri modelinden farklı olması durumunda, dış şemadan kavramsal şemaya geçiş, söz konusu veri modellerinin aynı olması durumunda göre daha zor olacaktır.” [5]



Şekil 2.3 Üç Şema Mimarisi

2.3 Veri Modelleri ve Veri Yapıları

2.3.1 Kavramsal Veri Modelleri

“Bir veri modeli gerçeğin belirli bir perspektifte soyutlanarak algılanmasının yapılandırılması ve bu algılanmanın tek anlamı olarak tanımlanması için birtakım kavram ve kurallar sağlar.” [5]

Kavramsal veri modelleri, yüksek düzeyli veri modelleri olarak bilinirler. Gerçeğin yüksek bir düzeyde, herhangi bir yazılım ve donanımdan bağımsız olarak

tanımlanması için bir dizi kavram ve kurallar içerirler. Bugüne kadar en yaygın olarak kullanılan kavramsal veri modeli oluşturma yaklaşımı varlık-ilişki modelidir. Varlık-ilişki modelinin temel kavramları "varlık", "öznitelik" ve "ilişki"dir. Varlıkların birbirleriyle olan ilişkilerinin ifade edilebilmesi, gerçeğin algıya en yakın biçimde tanımlanabilmesi açısından son derece önemlidir. Bir "ilişki tipi" ilişkinin ismi ve özellikleriyle tanımlanır. İki varlık tipi arasında bir ilişki, iki varlık arasında (bire-bir), tek bir varlıkla birden çok varlık arasında (bire-çok), ya da karşılıklı iki den çok varlık arasında (çoka-çok) olabilir. [5]

Hiyerarşik Veri Modeli : “Gerçeği hiyerarşik bir yapıda görür. Bu hiyerarşik karakterdeki gerçeğin temsili için oldukça doğrudan ve doğal bir yol olmakla birlikte, hiyerarşik olmayan yapıların temsiliinde problem yaratır. Hiyerarşik yapı genellikle ağaç yapı ile karakterize edilir; Kök, gövde, dal yerleri, dallar, ve yapraklardan bir yapı. Yabancı literatürde değişik adlarla anılan dal yerleri ve yapraklar bu çalışmada sırasıyla iç-düğüm ve uç-düğüm olarak anılacaktır.” [5]

“Her iç-düğüm (dallanma noktası) ait birden çok uç-düğüm bulunmakla birlikte, herhangi bir uç-düğüm yalnızca bir iç-düğümüne ait olabilir. Bu nedenle hiyerarşik model ile bire-çok ilişkiler doğrudan temsil edilebilirken, çoka-çok ilişkiler sorun olmaktadır.” [5]

“Sonuç olarak hiyerarşik veri modeli, hiyerarşik yapıdaki veri için oldukça etkili ve gerçekleştirim açısından da hızlı olmasına karşın, hiyerarşik olmayan ya da hiyerarşik ilişkiler yanında, karmaşık ilişkiler de içeren veri için uygun değildir. O nedenle, hiyerarşik model fiziksel düzeyde daha geniş bir uygulama alanı bulmuştur. Hiyerarşik veri yapıları bugün, verinin fiziksel organizasyonunda yaygın olarak kullanılmaktadır.” [5]

Ağ Veri Modeli : “Ağ modelinin temel kavramları ‘kayıt tipi’, ‘kayıt veri elemanı’, ve ‘set’dir. Verinin bir isim verilebilen en küçük birimi bir veri elemanıdır. Kayıtlar birbiriyle ilişkili veri elemanlarından oluşur. Kayıt tipi belirli özellikteki kayıtlar için bir kalıptır. Bir kayıt tipi, kayıt ismi, kaydı oluşturan veri elemanlarının ismi ve formatı ile tanımlıdır. Matematiksel set tanımindan farklı olarak, birbirine benzer olmayabilen kayıtlar, bir set oluşturmak üzere birbirine bağlanabilir. Bu bakımdan ağ modeli, hiyerarşik modele göre daha yüksek bir esneklik sunmaktadır.

Ağ modelinin hiyerarşik modelden en önemli farkı, uç-düğüm pozisyonundaki

verinin iç-düğüne işaret edebilmesidir. Böylelikle ağ modelinde 1:n ilişkiler yanında m:n ilişkiler de modellenabilir. Bu da veri tekrarı önemi ölçüde azaltır.” [5]

İlişkisel Veri Modeli : Hiyerarşik ve ağ modellerinin yukarıda söz edilen sorunları, yeni bir model arayışını getirmiş ve ilişkisel veri modeli önerilmiştir. E. F. Codd’un 1970’de yazmış olduğu “A Relational Model of Data for Large Shared Data Banks” makalesi ile ilişkisel veri yapılarında büyük bir ilerleme kaydedilmiştir [2]. Veri tabanı tasarımı ve yönetimi alanında bir devrim niteliğindeki yazıda, Codd ilişki mantığını tanıtmış, tabloların gerçek dünyadaki nesnelere göstermekte ve bunların niteliklerini tutmakta nasıl kullanılabileceğini belirtmiştir [3].

“İlişkisel veri modelinin temel kavram ilişkidir. İlişkiler yardımıyla, veri içerisindeki ilişkiler modellenir. Dolayısıyla, ilişkisel bir veri tabanı, çeşitli ilişki örneklerinden oluşur. Kavramsal olarak ilişkiler, satır ve sütunlardan oluşan iki boyutlu ‘tablo’larla karakterize edilir. Genellikle veri tabanında her tablo için bir dosya bulunur. Tablonun her satırını birbiriyle ilişkili verilerin bir topluluğudur. Sütunlarda ise nitelikler bulunur.” [5]

“Bir tablo içerisinde satırların sırası önemi değildir ancak, iki satır birbirinin aynı olamaz. Bir tablodaki sütun, yani nitelik sayısı sabittir. Bir tablonun her satırını, bir ya da birkaç sütun değeri vasıtasıyla tek anlamlı olarak tanımlanabilir. Söz konusu sütun değeri(ler)i anahtar olarak adlandırılır. Anahtarların diğer önemi işlevi farklı tablolar arasında bağlantıyı sağlamaktır. Bu yolla, kullanıcının veri tabanına yönelttiği bir sorgunun cevaplanması için gerekli veri farklı tablolardan alınabilir.” [5]

“Diğer modellerden farklı olarak, ilişkisel modelde veri içerisindeki ilişkilerin tümü veri tabanı oluşturulurken tanımlanmak zorunda değildir. Bir temel ilişkiler oluşturulduktan sonra, istendiğinde bunlardan yeni ilişkiler türetilebilir. İlişkisel modelin en önemli üstünlüklerinden biri budur. Çünkü farklı kullanıcıların zamanına göre değişebilen istek ve gereksinimleri veri içerisindeki farklı ilişkilere yönelik olmaktadır. Bu bakımdan ilişkisel model, anlık sorgularla başa çıkma konusunda oldukça güçlüdür.” [5]

İlişkisel veri tabanında uygulamaya geliştiren programcılar, yazdıkları kodun veri tabanının bütünlüğünü bozmasına 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 veri tabanında bulunacaktır. [3]

“Gerçekten de ilişkisel modelin popülaritesi çıkışından itibaren hızla artmış, diğer modellere üstünlüğü kabul edilmiş ve bugün MSSQL, ORACLE, INGRES gibi ilişkisel sistemler çok yaygın olarak kullanılmaktadır. Ancak modelin orijinal olarak karışık olmayan veriler için geliştirilmiş olması, onu konumsal veri türünden karışık veri içeren uygulamalar için yetersiz kılmaktadır. Bu nedenle, ilişkisel model için çok çeşitli genişletmeler önerilmiş ve bu genişletmeleri yapısında bulunduran pek çok ilişkisel veri tabanı yönetim sistemi geliştirilmiştir.” [5]

2.3.2 Fiziksel Veri Modelleri

“Fiziksel veri tabanı tasarımı, verinin veri tabanında fiziksel olarak depolanması için kullanılacak dosya yapıları ile veri tabanına giriş yol ve yöntemleri belirler. Fiziksel tasarım sonunda elde edilen iç şema seçilen fiziksel modele göre dosya, kayıt ve kayıt alanlarının nasıl organize edildiğini ve birbiriyle nasıl ilişkili olduğunu ve veriye giriş yollarını gösterir.” [5]

“Genellikle bir veri tabanı yönetim sistemi dosya organizasyonu ve veri erişimi için bir dizi seçenek sunar. Bunlar, çeşitli indeksleme tipleri, birbiriyle ilişkili kayıtların göstergeçler yardımıyla birbiriyle bağlanması, çeşitli adresleme teknikleri, birbiriyle ilişkili kayıtların disk üzerinde gruplanmasıdır. Uygun veri yapılarının seçiminde amaç iyi performans ve bilgisayar kaynaklarının en iyi şekilde kullanımıdır. Bunun için, yanıt zamanı, depolama alan gereksinimi ve işlem yoğunluğu gibi ölçütlere göre bir değerlendirme yapılır. Yanıt zamanı, bir işlemin aldığı zamandır. Depolama alan gereksinimi, belli bir dosya yapısına göre dosyalar ve giriş yolları için gerekli disk alanıdır. İşlem yoğunluğu ise, dakikada gerçekleştirilebilen ortalama işlem sayısıdır.” [5]

“Fiziksel veri modelleri çoğunlukla veri yapıları olarak anılır. Dosya yapısı dendiğinde ise bir veri yapısının gerçekleştirimi anlaşılır. Buradaki ayrımlıdır çünkü verilen bir veri yapısı değişik dosya yapıları kullanılarak gerçekleştirilebilir. Örnek olarak bir liste veri yapısı, sıralı erişimi olarak ya da göstergeçler kullanılarak gerçekleştirilebilir. Bu kısımda, yaygın olarak kullanılan genel amaçlı veri yapıları olarak diziler, listeler, yığınlar, kuyruklar ve ağaçlardır.” [5] Bunların ayrıntısına burada girilmeyecektir.

2.4 Yapısal Sorgulama Dili (SQL)

Yapısal Sorgulama Dili (SQL), veri tabanı uygulamalarında, veri tanımlama, veri tabanının bütünlüğünün kontrolü, veri tabanlarına erişimin kontrolü, veri tabanlarının sorgulanması ve güncellenmesi amaçları için gerekli komutlara sahip olan bir alt dildir. At dil denilmesini nedeni, bir bilgisayar dilinin sahip olması gereken tüm komutlara sahip olmayışından dolayıdır. SQL İngilizce’deki ‘Structured Query Language’ sözcüklerini baş harflerinden oluşur. [4]

SQL’ni sahip olduğu komutlar, çevrim(döngü) oluşturan komutlarla, if, then, else yada go to gibi kontrol ve dallanma komutlarıdır. Fakat SQL’ni diğer dillerle birlikte kullanılması mümkün olduğundan, gerekiyorsa SQL komutları diğer dillerin çevrimya da kontrol komutları içerisinde kullanılabilir. [4]

Bu dilin detayları hakkında bilgi sahibi olunarak veri tabanını performanslı ve hızlı şekilde kullanılabilmek mümkün olur. Çoğu zaman orta derecede SQL kullanılan veritabanlarında performans kaybı ve gereksiz sorgularla karşılaşılır. Veri tabanından sonucu tek sorguda çekmekle iki sorguda çekmek arasında oldukça büyük performans kaybı olacaktır. Aşağıda SQL komut grupları kullanımı hakkında bilgiler verilmiştir. [4]

Yeni bir veri tabanı oluşturmak için,

CREATE DATABASE veri tabanı ismi;

komutu kullanılır.

Varolan bir veri tabanını kaldırmak için,

DROP DATABASE veri tabanı ismi;

komutu kullanılır.

Tabloların oluşturulması için,

CREATE TABLE tablo adı (kolon tanımları);

komutu kullanılır. Kolonlar tanımlanırken veri tipleri ile birlikte tanımlanırlar. Tablolar ile ilişkili veri tiplerinden başka açıklanması gereken bir diğer husus da bazı alanların yanında görülen NOT NULL ifadesinin anlamına gelmektedir. Bu ifade söz konusu alan ile ilişkili olarak mutlaka veri yüklenmesi gerektiğini, ilgili alanın boş bırakılmayacağını ifade eder.

Tablolara veri yüklenmesi,

INSERT INTO tablo ismi **VALUES** (veriler);

komutu ile gerçekleştirilir.

Tablodan satır silmek için,

DELETE tablo ismi **WHERE** koşul;

komutu kullanılır.

Tablo satırlarındaki verilerde değişiklik yapma, güncelleme (update),

UPDATE tablo ismi

SET kolon adı =değer

WHERE koşullar;

komutu ile gerçekleştirilir.

Mevcut tabloya kolon eklemek için,

ALTER TABLE tablo adı **ADD** (eklenecek kolonlar);

komutu kullanılır.

Mevcut tablonun kolonlarında değişiklik yapmak için,

ALTER TABLE tablo ismi **MODIFY** (kolon adı kolon tipi);

komutu kullanılır.

Mevcut tablodan kolon silmek için,

ALTER TABLE tablo ismi **DROP** (kolon adı);

komutu kullanılır.

Mevcut tablonun adını değiştirmek için,

ALTER TABLE tablo ismi **RENAME TABLE** isim;

komutu kullanılır.

Mevcut tabloyu tüümüyle silmek için,

DROP TABLE tablo adı;

komutu kullanılır.

Temel SQL sorgularınları,

SELECT istenilen kolonlar

FROM istenilen kolonların bulunduğu tablo ismi

WHERE koşullar;

komutu ile gerçekleştirilir.

Birden çok koşula dayalı sorgulamalarda NOT, AND ve OR mantıksal operatörleri kullanılarak karmaşık yada bileşik koşullu listelemeleri gerçekleştirmek mümkün olmaktadır.

Tablo 2.1: Mantıksal operatörler

| Operatör | Anlam |
|----------|--|
| AND | Her iki bileşenin de doğru olduğu durumlarda TRUE değeri döndürülür. |
| OR | Bileşenlerin en az birinin doğru olduğu durumlarda TRUE değeri döndürülür. |
| NOT | Verilen koşulun FALSE olma durumunda TRUE değerini döndürür. |

Mantıksal operatörlerin öncelik sırası NOT, AND ve OR şeklindedir. Bu öncelik sırası parantezler kullanılarak değiştirilebilir.

Ayrıca karşılaştırma operatörleri kullanılarak kolonlar, birbirleri ile yada standart bir takım değerler ile karşılaştırılabilirler. Karşılaştırma operatörleri ve işlevleri aşağıdaki tabloda verilmiştir.

Tablo 2.2: Karşılaştırma operatörleri

| Operatör | İşlevi |
|----------|----------------|
| = | Eşittir |
| > | Büyüktür |
| >= | Büyük eşittir |
| < | Küçüktür |
| <= | Küçük eşittir |
| <> | Eşit değildir. |

Kayıtların belirli bir koşula göre gruplandırılması,

SELECT istenilen kolon adları

FROM kolonlarının istenildiği tablo adı

GROUP BY gruplanmak istenilen kolon adı

ORDER BY kolon belirterek içeriisindeki veriye göre gruplama (tek kolon)

HAVING gruplama koşulu;

komutu ile gerçekleştirilir.

COMMIT (Kesi nleřtirme): Yapılan iřlem in tüm tabloyu kalıcı bir řekilde etkilemesi için kullanılan komuttur.

ROLLBACK (Geri alma): Yapılan deęiřiklięin iptal edilerek tablonun bir önceki haline döndürülmesi için kullanılan komuttur.

SQL tümeleri yazılırken her komut için aynı olan ve ařaęıda belirtilen genel yazma kořullarına dikkat edilmelidir.

- SQL tümeleri aksi belirtilmedikçe büyük küçük harf ayrımlı deęildir.
- SQL tümeleri bir veya birden fazla satırdan oluşabilir.
- Anahtar kelimeler satır boyunca bölünmelidir.
- Her SQL tümesi ";" ile bitmelidir. [4]

3. VERİ TABANI, WEB PROGRAMLAMA DİLİ ve İLİŞKİLERİ

3.1. MySQL Veri Tabanı

Bildiği gibi bilgisayar dünyasında kullanılan bir çok veri tabanı program ve sunucusu mevcuttur. Bunların çoğu kullanışlı olmasına rağmen çok pahalı paket programlardır. MySQL'in en büyük özelliği bedava olmasıdır. UNIX ve OS/2 platformları için ücretsiz dağıtım paketiyle birlikte Windows platformları için 30 günlük deneme sürümü sonunda lisans alınmak üzere sunulmaktadır. MySQL'in diğer en büyük özelliği ise veri tabanı pazarındaki en büyük rakiplerinden daha iyi, hızlı ve kullanışlı olmasıdır. MySQL halen daha geliştirilmekte olmasına rağmen mevcut haliyle zengin ve çok kullanılan fonksiyonlar sunmaktadır. MySQL, çok-kanallı (multi-threaded), çok kullanıcı (multi-user), hızlı ve sağlam (robust) bir veri tabanı yönetim sistemidir. Kaynak kodu açık olan MySQL'in pek çok platform için derlenmiş sürümleri hazır, ODBC (Open DataBase Connectivity) sürücülere de mevcuttur. MySQL, tuttuğu tablo ve çok kullanıcı bir sistemlerde söz konusu olan erişim hakları sorununu etkin bir biçimde çözümlenmektedir. Negatif bir yön olarak MySQL'de hareket kilitlenmesi (transaction locking) desteklenmesi, hareket kilitleme güvenliğini sağlama işini programcının üzerine bırakmaktadır. [6]

MySQL Inc. tarafından hazırlanmış bir veri tabanı yazılımıdır. MySQL'in ortaya çıkış nedeni, yaratıldığı yerde çok büyük bir veri tabanını işleyebilecek bir SQL sunucusuna ihtiyaç duyulmasıdır. Geliştiricileri, 500'den fazlası 7 Milyon kayıt içeren 10,000 tablodan oluşan kendi veri tabanlarını (100 gigabyte civarında veri) MySQL'de tutmuşlardır. MySQL, teknik tabiriyle sunucuda daemon olarak çalışır ve arzu eden programa, bildireceği (ve erişim hakkı bulunan) veri tabanı dosyasından veri çekerek sunar. [6]

MySQL'in belli başlı özellikleri :

- Sistemde birden fazla işlemci (CPU) var ise bunları kullanabilmesi,
- Değişik işletim sistemlerinde çalışması,
- Değişik sütun (veri) tiplerini desteklemesi,

- Aynı sorgulama iinde deęişik veri tabanlarındaki tabloları birleřtirme,
- ODBC desteęini bulunması,
- Büyük veri tabanlarını işleyebilme özellięi,
- C ve C++ dillerinde yazılmış olması,
- Bütün verilerin ISO 8859-1 Latin 1 formatında kayıt ediliyor olmasıdır. [6]

3.2 PHP Web Programlama Dili

“İnternet ilk oluşturduğu dönemlerde ağ üzerinde bilgilerin sunumu için HTML (**H**yper **T**ext **M**arkup **L**anguage) dili oluşturulmuştur. HTML saf metin dosyasıdır. Bu metin dosyasının içeriğine görüntülenmek istenen bilgiler uygun komutlarıyla yerleştirilir. Bu metin dosyasında dilin tanımladığı özel etiketler (tag) kullanılır. Sunucu da duran bu dosya isteemi tarafından çağrıldığında olduğu gibi isteemi bilgisayara gönderilir. HTML burada kullanıcının tarayıcısı (browser) tarafından yorumlanarak görüntülenir.” [4]

“Zamanla HTML bir çok noktada yetersiz kalmaya başlamıştır. Bu istekleri karşılamak için kendi başlarına sayfa sunamayan ancak HTML'in içine gömülerek ona özellikler katan diller (script) geliştirilmiştir. Bunların çalışma yöntemleri genel olarak iki farklı şekilde incelenebilir.” [4]

“**İsteemi taraflı scripting dilleri:** Oluşturulan kodun aynen gönderilip kullanıcının tarayıcısında yorumlanması yöntemiyle çalışır. En büyük örneęi JavaScript'dir. Bir çok sayfada karşımıza çıkan kayan yazılar, Windows tarzı menüler bu dille oluşturulmaktadır. Ancak kodu çalıştıran kullanıcının tarayıcısı olduğundan bu eklentiler farklı tarayıcılarının farklı sürümlerinde beklenmedik sonuçlar verebilirler.” [4]

“**Sunucu taraflı scripting dilleri:** Bu dillerde sayfa çağrıldığı anda sunucu tarafından yorumlanır. Kullanıcıya ise bunun sonucunda oluşan HTML kodu gönderilir. Ancak Javascript'deki gibi görsellik değil, sayfanızın çağrılma anında dinamik olarak oluşturuluyor olmasıdır. ASP (**A**ctive **S**erver **P**ages), PERL- **C**GI, JSP (**J**ava **S**erver **P**ages), CFS (**C**oldfusion), PHP (**P**ersonal **H**ome **P**age) en çok kullanılanlarıdır.” [4]

PHP adını "Personal Home Page Tool" kelimelerinin baş harflerinden almıştır. PHP, özellikle dinamik web sayfaları yaratmak amacıyla geliştirilmiş, www sunucu tarafında çalışan bir programlama dilidir.

"Kendisi de C ile yazılan PHP, tıpatıpatı C'nin yapısının Web için uyarlanmış halidir. Yani C C++ bilen biri için PHP öğrenmek en fazla bir saat sürecektir. En büyük özelliği ise HTML dosyasına normal bir etiketler gibi eklenebilir. Bunun için <? ve ?> etiketlerini kullanır. Sayfa çağrıldığı anda dosya önce PHP yorumlayıcısına gider. Burada <? ?> etiketleri arasındaki PHP kodu işlenir ve oluşturulan HTML dosyası istemciye gönderilir. Yani kod o anda bir HTML çıktısı üretir. Ziyaretçinin bütün görebileceği bu çıktıdır. PHP kodunu görmez. Dinamik sayfa kavramı da buradan gelmektedir. Ziyaretçinin kullandığı tarayıcıya, günün saatine, bağlanılan ülkeye ve akla gelebilecek her türlü değişkene bağlı olarak farklı bir sayfa yaratılıp gönderilebilir. Portal tarzı web sitelerinin oluşturulmasında tercih edilebilecek bir web programlama dilidir." [4]

"PHP, Rasmus Lerdorf'un ilk başta sadece kendi sitesinde kullanmak üzere yaptığı, sayaç, ziyaretçi defteri gibi birkaç araçtan oluşuyorken, zamanla başkaları tarafından da kullanılmaya başlanmıştır. Bunun üzerine PHP'yi geliştirip bir form yorumlayıcısı ekleyen Lerdorf, PHP/FI Sürüm 2'yi çıkarmıştır. Bundan sonra artık PHP, kişisel bir proje olmaktan çıkarak, kullanıcıları tarafından açık kod (open source) olarak geliştirilen başlı başına bir web programlama dilidir. PHP3 ise, birçok yeni özellik ve tıpatıpatı yeniden yazılmış olarak karşımıza çıkmış, PHP4'e geldiğinde ise güçlü ZEND scripting motoru kullanmaya başlanmıştır. PHP 1995 yılında ilk kez kullanılmaya başlandığından bu yana epey yol almıştır. Öncelikle linux için yazılmış diğer pek çok uygulamaya gibi C ile yazılmış ve kodunun herkese açık olması oldukça hızlı bir şekilde gelişmesini sağlamıştır." [4]

Web programlamada ve dinamik web sayfaları üretmede elimizde bir sürü seçenek varken neden PHP kullanıyoruz sorusunun cevabı aşağıdaki maddelerle açıklanabilir.

- Programlama dilinin basitliği,
- Veri tabanlarıyla mükemmel iletişim ve bağlantıda kullanılan komutların kolaylığı,
- Kullanılan platformdan bağımsızlık (95/98/NT/2000/XP, Unix ve Linux türevleri üzerinde çalışabilirliği),

- Kaynak kodunun açık olması,
- HTML kodu arasında çok kolay gömülebilme özelliği,
- Nesne-yönelimli (object oriented) programlama yapılabilişi
- Hız, performans, güvenilirlik
- Programları hızlı geliştirebilmek için hazır kütüphaneleri kullanabilme avantajı,
- PHP'nin öğrenilişi veya karşılaşılan sorunların çözülmesi aşamasında soru sorabileceğiniz, fikir alabileceğiniz tartışma listelerini (e-posta , forum) ve elektronik kitapların çok kolay bulunabilmesi
- Web'de çalışmak üzere tasarlanmış olması nın getirdiği diğer avantajlardır [4].

3.3 PHP ve MySQL İlişkisi

PHP birçok veri tabanı yazılım ile kullanım desteklemektedir. PHP tarafından desteklenen veri tabanları şöyle sıralanabilir:

| | | | |
|------------|---------------|----------|----------|
| Adabas D | Interbase | Solid | msQL |
| Sybase | Empress | MySQL | Velocis |
| FileProf | Oracle | Unix DBM | Informix |
| PostgreSQL | Microsoft SQL | | |

“PHP, yaygın kullanılan her türlü veri tabanı kullanıcılarına veri tabanına özel PHP komutlarıyla kullanım imkanı sağlasa da PHP kullanım kılavuzu incelendiğinde veri tabanları arasında en çok PHP komutunun MySQL, daha sonra Oracle için olduğu görülecektir. PHP ile adı en çok geçen veri tabanı MySQL'dir. MySQL komutlarından birçoğu özel durumlarda gerekli durken 10 kadar komut her türlü MySQL işleminde daha yaygın olarak kullanılırlar.” [4]

“Bir PHP dosyasında MySQL veri tabanı kullanılmak istendiğinde mutlaka veri tabanı sunucusuna bağlanmalı ve bir veri tabanı seçilmelidir. Genellikle bir PHP dosyasında bir veri tabanına bağlantı yapılacağı için veri tabanı sunucusuna bağlandıktan sonra bir veri tabanı ismi verilerek tüm diğer MySQL işlemleri o veri tabanı sunucusu ve o veri tabanı üzerinde yapılabilir. Eğer istenirse birden çok veri tabanı sunucusundaki birden çok veri tabanına da bağlantı yapılabilir.” [4]

“MySQL'de oluşabilecek hataları kullanıcının PHP içinde fark edebilmesi için PHP'nin özel komutları vardır. Bu komutlar sayesinde veri tabanı sorgulamasında

oluşan sonuçlar hata numarasıyla birlikte fark edilebilir. Hata gösterge komutları kullanıldığında PHP, MySQL'e dair hiçbir hata mesajı vermeden işleme devam eder.” [4]

PHP’de kullanılabilen MySQL komutları aşağıdaki gibidir [4].

| | |
|----------------------------------|---|
| <code>mysql_connect</code> | : Sunucuya veri tabanı bağlantısı açar. |
| <code>mysql_close</code> | : Belirtilen MySQL bağlantısını kapatır. |
| <code>mysql_create_db</code> | : MySQL’de veri tabanı açar. |
| <code>mysql_drop_db</code> | : Sunucudan veri tabanı siler. |
| <code>mysql_data_seek</code> | : Sonuç satırında belirtilen sıraya geçer. |
| <code>mysql_db_query</code> | : MySQL’e sorgu gönderir. |
| <code>mysql_affected_rows</code> | : Bir önceki işlemlerde etkilenen satır sayısıdır. |
| <code>mysql_errno</code> | : Bir önceki işlemlerdeki MySQL hata numarasını verir. |
| <code>mysql_error</code> | : Bir önceki işlemlerdeki MySQL hata mesajını verir. |
| <code>mysql_fetch_array</code> | : Sonuçları dizi değişkeni olarak alır. |
| <code>mysql_fetch_field</code> | : Sonuç tablosundaki alan adı objektif olarak alır. |
| <code>mysql_fetch_lengths</code> | : Sonuç tablosundaki dizi değişkeninin uzunluğunu alır. |
| <code>mysql_fetch_object</code> | : Sonuç satırını objektif olarak alır. |
| <code>mysql_fetch_row</code> | : Sonuç tablosundan dizi değişkeni alır. |
| <code>mysql_field_name</code> | : Sonuç tablosundaki sonucun tablodaki alan adı verir. |
| <code>mysql_field_seek</code> | : Sonuç tablosunda sıra indeksini belirtilen yere götürür. |
| <code>mysql_field_table</code> | : Alan adı verilen sonucun tablo adı verir. |
| <code>mysql_field_type</code> | : Sonuçtaki alanın hangi tip olduğunu belirtir. |
| <code>mysql_field_flags</code> | : Sonuçtaki alanın hangi tür ekstra parametrelerle tanımlandığını belirtir. |
| <code>mysql_field_len</code> | : Sonuçtaki alanın veri tabanındaki uzunluğunu verir. |
| <code>mysql_free_result</code> | : Sonuçları için atanan hafızayı boşaltır. |
| <code>mysql_insert_id</code> | : Bir önceki veri yerleştirilmeden oluşan otomatik veri değerini verir. |
| <code>mysql_list_fields</code> | : Sonuçtaki tüm tablo alanlarını listeler. |
| <code>mysql_list_dbs</code> | : Sunucudaki tüm veri tabanlarını listeler. |
| <code>mysql_list_tables</code> | : Veri tabanındaki tüm tabloları listeler. |
| <code>mysql_num_fields</code> | : Sonuçtaki alan sayısını verir. |
| <code>mysql_num_rows</code> | : Sonuçtaki satır sayısını verir. |

mysql_pconnect : Sunucuya kalıcı bir bağlantı tanımlar.
mysql_query : Veri tabanına sorgu gönderir.
mysql_result : Sorgudan dönen sonuçları alır.
mysql_select_db : Sunucudan veri tabanı seçer.
mysql_tablename : Verilen alanın ait olduğu tablo adını verir.

4 JEODEZİK VERİ TABANI

4.1 Jeodezik Veri Tabanı Gereksinimi

Jeodezik Veritabanlarının kullanım nedenleri de, geçtiğimiz bölümlerde veritabanlarının kullanım nedenlerinde değinilenlerle aynıdır. Değişen, tutulması istenen veriler ve bu verilerin ne amaçla kullanılacağıdır. Jeodezik veri tabanı oluşturulmasında tutulmak istenen arazide yapılan jeodezik ölçmeler ve bu ölçmeler sonucunda elde edilen değerlendirme sonuçlarıdır. Amaçlanan ise, bu ölçmelerin ve değerlendirme sonuçlarının gerektiğinde yeniden kullanılabilirliği için güvenli bir şekilde saklanması ve yine gerektiğinde bu ölçmelere ve değerlendirme sonuçlarına ilgili kişilerin hızlı olarak erişebilirliğidir. Bilgisayar ortamında oluşturulmuş bir veri tabanı sayesinde bilgisayar ve veri tabanı teknolojilerinin tüm avantajları kullanılabilir.

Jeodezik veri tabanında nelerin tutulacağı sorusu; jeodezik ölçmelerde nelerin ölçüldüğü, bu ölçmeler sonrasında ne tür büyüklükler, sonuçlar elde edildiği göz önüne alınarak cevaplanabilir. Jeodezik ölçmelerde arazide fiziksel veya geometrik birçok büyüklük ölçülebilir. Bu büyüklüklerin ne olduğuna gelince de en çok ölçülen büyüklükler analog veya elektronik teodolitlerle ölçülen yatay doğrultular veya açılarıdır. En çok ölçülen diğer ikinci büyüklük ise yer kontrol noktaları arasındaki mesafelerdir. Çelik şerit metreler, invar teller veya elektronik uzaklık ölçerler ile ölçülen uzaysal mesafeler yardımıyla yatay mesafeler bulunurlar. Diğer bir ölçme türü ise astronomik ölçmelerdir. Astronomik ölçmelerde yer kontrol noktalarından belli doğal gök cisimlerine, belli zamanlarda yapılan yatay ve düşey açı ölçmeleri yardımıyla konum belirleme mümkündür. Ölçülen diğer bir büyüklük ise yükseklik farklarıdır. Nivelman ile yapılan geometrik nivelman veya klasik elektronik teodolitlerle yapılan trigonometrik nivelman ile yatay kontrol noktaları arasındaki yükseklik farkları belirlenebilir. Veri tabanında tutulabilecek başka bir jeodezik ölçme türü ise uydulara yapılan gözlemler ile konum belirlemeye yönelik ölçmelerdir. Günümüzün en çok kullanılan modern ölçme tekniklerinden birisi ise “Global Positioning System (GPS)” adı verilen yapay uydularla konum belirleme tekniğidir.

21+3 yedek GPS uydusu yardımıyla mutlak veya göreceli olarak yerel nokta konularını belirlemek mümkün olmaktadır. Yer in gerçek şeklini belirlemeye yönelik çalışmalarda ve yer in gravite alanının belirlenmesinde bize en çok yardımcı olan fiziksel büyüklük ise gravite değeri dir ve gravi metri k ölçmeler de bir jeodezik veri tabanında mutlaka bulunması gereken ölçmelerdir. Diğer taraftan sıcaklık, nem vb. gibi fiziksel büyüklükler de bazı ölçmelerin değerlendirilmesi açısından ölçülmesi gereken büyüklüklerdir. [1]

Tüm bu yukarıda sözü edilen ölçmeler genel de yer kontrol noktaları arasında veya üzerinde yapılan temel ölçmelerdir. Tabii bu yöntemlerin büyük bir kısmı detay ölçmelerinde de kullanılmaktadırlar. Detay ölçmelerinin de veri tabanında uygun yapılarda tutulması mümkündür. Fakat genel de düşünülen yer kontrol noktalarında yapılan ölçmelerin ve bu ölçmeler sonucunda elde edilen değerlendirme sonuçlarının veri tabanında tutulmasıdır. Detay ölçmelerinin her birinin ayrı ayrı veri tabanında tutulması değil de, örneğin bir kontrol noktasından yapılan detay ölçmelerinin tamamının tek bir dosya vb. şeklinde tutulması şeklinde detay ölçmeleri de veri tabanında tutulabilirler.

Günümüzün modern ölçme aletlerinin birçoğunda kayıt üniteleri bulunmaktadır. Veya bu ölçme aletlerini düzüstü veya avuçiçi bilgisayarlara bağlayarak kayıt yapma kapasitelerini artırmak da mümkündür. Fakat bu üniteler daha çok arazide yapılan ölçme ve dolayısıyla kayıt süresini artırmaya yönelik birimlerdir. Bu yüzden veri tabanı niteliğinde bir yapıya göre tasarlanmamışlardır. Ayrıca değerlendirme sonuçlarının da saklanması ve gerektiğinde yeniden kullanılması söz konusu olduğunda bir veri tabanı kullanılması kaçınılmazdır. [1]

Diğer taraftan bir ölçme aletinin kayıt ünitesini veri tabanı niteliğinde bir depolama aracı olarak kullanmak mümkün de olsa tek veya birkaç tip ölçmeye yönelik bir veri tabanı söz konusu olacaktır. Örneğin bir GPS alıcısı kayıt ünitesi en fazla GPS ölçmelerine yönelik bir veri tabanı niteliği taşıyabilecektir.

Veri tabanı kullanımının bir başka avantajı da standartlaşmadır. Arazide tutulan kayıt, ölçme çizelgeleri, krokiler ve röperler belirli standartlarda da olsa, yazıcı veya çizici kişiye göre farklılıklar gösterebilmektedir. Bu belgeler dosyalanıp saklandıklarında belli bilgiler sadece krokiyi tutan veya ölçmeleri yazan kişi tarafından anlaşılabilir şekilde olabilir. Bu da bir sakıncadır. Veya kullanılan ölçme aletinin kayıt ünitesi,

ölçme aletini üreten ilgili firmanın standartlarına uygun bir depolama formatına sahip olabilir. Tüm ölçmeler için nokta veya proje bazlı ortak bir veri tabanı beraberinde standartlaşmayı da getirecektir. Ölçme ve değerlendirme bilgilerinin ilgili herkes tarafından anlaşılabilirliği sağlanmış olacaktır.

Ölçme ve değerlendirme sonuçlarını bir veri tabanında tutmak daha önce değinildiği gibi depolama kapasitesinin yüksek olmasını sağlayacak, veriye erişimi hızlandıracak çok kullanıcının aynı anda ve istenildiğinde uzaktan erişimine olanak tanıyacaktır. Jeodezik ölçme ve değerlendirme sonuçlarına özel olarak bu avantajları maddeler halinde aşağıdaki gibi sıralanabilir.

- Aynı bölgede değişik zamanlarda aynı ya da farklı amaçlar için yapılmış ölçme ve değerlendirme sonuçlarına hızlı ve eksiksiz erişim sağlanması; bu sayede istatistiksel analizlerin ve yorumların yapılabilmesi,
- Ölçme ve değerlendirme sonuçlarının saklanmasıyla tüm veriler için standart sağlanması,
- İlgili kişilerin, çok kullanıcılar olarak ve izin verildiği ölçüde verilere ulaşımının sağlanması,
- Geliştirilen arayüzlerle ölçmelerin ve değerlendirme sonuçlarının veri tabanının web'e açılabilmesi,
- Yalnızca bir ölçme yöntemi ne, aletine veya ilgili aletin standardına göre değil, tüm yöntemlere göre ve belli bir standartta veri tutmaya olanak sağlanması,
- Kayıt ünitelerinden farklı olarak, sadece ölçmelerin değil değerlendirme sonuçlarının da veri tabanında tutulması,
- İlgili kurumun tüm ölçme ve değerlendirme verilerinin tek bir veri tabanında tutulmasına olanak sağlanmasıdır.

4.2 Jeodezik Veri Tabanı İçin Genel Tablolar

Tüm jeodezik veri tabanı için bazı tablolar bütün ölçme yöntemleri için ortak olarak kullanılmaktadır. Aslında tasarımı açısından tablonun genel bir tablo olması veya bir ölçme yöntemine özgü olması önemli bir farklılık yaratmamaktadır. Bu tabloların ölçme yöntemlerine özgü tablolarla bağlantıları ilişkisel veri tabanı yapısı kullanılarak aynı tablo içerisinde veya ara tablolar ile çözülmektedir. Bu tablolar *proje, istasyon, kurum personel, det* ve *meteoroloji* tablolarıdır. Genel tablo aynı m konunun ve veri tabanının anlaşılabilirliğini artırması nedeniyle yapılmıştır.

Her men her men her tabloda *notlar* veri alanı bulunmaktadır. Bu alan tablolarda unutulmuş olmasında muhtemel bilgilerin girilebilmesinin yanı sıra, veri tabanını kullanan kişiye fazladan istediği veriyi girme serbestisi de sağlanmaktadır.

4.2.1. Proje Tablosu

Proje, belirli bir bölgede belirli bir zaman aralığı içerisinde yapılmış olan tüm jeodezik ölçümleri ve bu ölçümler sonucunda elde edilmiş değerlendirme sonuçlarını içeren bir kavram olarak tanımlanabilir. Veri tabanındaki verilere nokta ve proje bazında ulaşım önemli olduğundan ve yapılacak her jeodezik ölçüm bir proje ile ilişkilendirileceğinden proje tablosu oldukça önemlidir (Tablo 4.1). Projeyi tanımlamaya yetecek yeterli bilgileri en iyi şekilde içermelidir. Bu nedenle gerekirse *notlar* alanı etkin bir şekilde kullanılmalıdır. Tabloda verilen bilgilere ek olarak proje bölgesinde kalılabilecek yer bilgisi, toplam proje maliyeti gibi bilgiler gibi çok genel bilgiler de yer alabilir.

Tablo 4.1: Proje bilgilerinin tutulacağı tablo (*proje*)

| Alanlar | Veri Türü | Açıklama |
|------------------|--------------|--|
| proje_id | bigint(20) | <pk> Oluşturulan her yeni proje için tek anlamlı olan id |
| proje_adi | varchar(250) | Projeyi tanımlayacak bir isim |
| İl | varchar(50) | Projeyi gerçekleştirildiği il. |
| İlçe | varchar(50) | Projeyi gerçekleştirildiği ilçe. |
| Bölge | varchar(50) | Projeyi gerçekleştirildiği bölge. |
| baslangic_tarihi | date | Tarih formatında (yyyy-aa-gg) projeyi başladığı tarih |
| bitis_tarihi | date | Tarih formatında (yyyy-aa-gg) projeyi bittiği tarih |
| Notlar | text | Projeye ilişkin genel notlar vb. |

4.2.2. İstasyon Tablosu

istasyon_id istasyon tablosunun birincil anahtarıdır ve bu tablo herhangi bir ölçüm bilgisi içermektedir. *istasyon_no* alanının değil de, ayrıca eklenen *istasyon_id* alanının birincil anahtar olarak seçilmesini nedeni aynı noktanın değişik projelerde, değişik yöntemlerle hesaplanmış koordinatlarının da tutulmak istenebileceği düşüncesi nedeniyledir. *hesaplanma_yontemi* alanında koordinatların nasıl ve hangi projede üretildiği belirtilebilir. Noktanın herhangi bir referans sistemindeki 3 boyutlu koordinatlarını, tesis tarihi, tesis türü, ulaşım bilgisi vb. gibi bilgiler yanında bazı hesaplamalarda kullanılabilecek gravite ve topoğrafik düzeltme gibi bilgileri de

istasyon tablosu içermektedir. Bir noktanın değişik projelerde hesaplanmış koordinatlarınının da tutulacağı ve bunlardan sadece bir tanesinin kesinin koordinatı olarak kabul edileceği düşüncesiyle *kesin* alanı konulmuştur. Bu alanın 1 olması durumunda koordinatın kesin olduğu anlaşılacaktır. Tüm ölçümler istasyon bazlı olacağından en çok ilişki kurulan tablo istasyon tablosu olacaktır (Tablo 4.2).

Tablo 4.2: İstasyon bilgilerinin tutulacağı tablo (*istasyon*)

| Alanlar | Veri Türü | Açıklama |
|---------------------|---------------|---|
| istasyon_id | bigint(20) | <pk> İstasyonu tanımlayan tek anahtarı id |
| istasyon_no | varchar(20) | İstasyon numarası. |
| istasyon_adi | varchar(50) | Varsa istasyonun adı. |
| X | decimal(20,8) | İstasyonun 1. eksen koordinatı. |
| Y | decimal(20,8) | İstasyonun 2. eksen koordinatı. |
| Z | decimal(20,8) | İstasyonun 3. eksen koordinatı. |
| Kesin | tinyint(1) | Koordinatın kesin olup, olmadığı. |
| Datum | varchar(50) | Koordinatın ait olduğu referans sistemi. |
| Gravite | varchar(50) | İstasyonun gravite değeri. |
| topografik_duzeltme | varchar(50) | İstasyondaki topografik düzeltme değeri. |
| tesis_tarihi | date | İstasyonun tesis tarihi (yyyy-aa-gg). |
| tesis_turu | varchar(50) | İstasyonun zemin işareti türü |
| pafta_adi | varchar(50) | İstasyonun içinde bulunduğu 1:100000'lik pafta bilgisi. |
| ulasim_bilgisi | varchar(250) | İstasyona ulaşım bilgisi metni. |
| Kr oki | varchar(50) | İstasyona ulaşım kr okisi. |
| Roper | varchar(50) | İstasyon röper kr okisi. |
| hesaplanma_yontemi | varchar(100) | Koordinatların hangi yöntemle hesaplandığı. |
| Notlar | text | İstasyona ilişkin genel notlar vb. |

4.2.3 Kurum Tabloları

Yapılan projenin ilişkili olduğu kurum veya kurumları da veri tabanında tutmak gerekli görülmüştür (Tablo 4.3). Gerektiğinde ilgili kuruma ulaşılacak için adres bilgileri ve o kurumda proje ile ilgili olan kişi bilgisi kurum tablosunda yer almaktadır. İlgili kurumun bünyesindeki birimlerin tutulması ve gerektiğinde yeni bir birim eklenebileceği için ayrıca *birim* tablosu oluşturulmuş ve kurum birim ilişkilerini sağlamak için kurum tablosunun birincil anahtarı *kurum_id* bu tabloya yabancı anahtar olarak gelmiştir (Tablo 4.4). Proje ile ilişkili kurumların birden fazla olabileceği düşünülerek bu ilişkiyi sağlamak için (bir projeye birden fazla kurum)

proje_kurum ara tablosu oluşturulmuş ve bu tabloya *proje_id* ve *kurum_id* alanları yabancı anahtarlar olarak gelmişlerdir (Tablo 4.5).

Tablo 4.3: Kurum bilgilerini tutulacağı tablo (*kurum*)

| Alanlar | Veri Türü | Açıklama |
|--------------|--------------|--|
| kurum_id | int(11) | <pk> Kurumu tanımlayıcı tek anlamı id |
| kurum_adi | varchar(100) | Kurumun adı. |
| kurum_adres | varchar(250) | Kurumun adres bilgisi. |
| kurum_tel | varchar(15) | Kurumun telefon bilgisi. |
| kurum_faks | varchar(15) | Kurumun faks bilgisi. |
| kurum_eposta | varchar(100) | Kurumun e-posta bilgisi. |
| ilgili_kisi | varchar(100) | Kurumda proje ile ilgili irtibat kurulacak kişi/ler bilgisi. |
| Notlar | text | Kuruma ilişkin genel notlar vb. |

Tablo 4.4: Kuruma ait birim bilgilerini tutulacağı tablo (*birim*)

| Alanlar | Veri Türü | Açıklama |
|-----------|--------------|--|
| birim_id | int(11) | <pk> Birimi tanımlayıcı tek anlamı id |
| kurum_id | int(11) | <fk> Birimin bağlı olduğu kurumu gösteren yabancı anahtar. |
| birim_adi | varchar(100) | Birim adı. |

Tablo 4.5: Proje-kurum ilişkilerini sağlayacak ara tablo (*proje_kurum*)

| Alanlar | Veri Türü | Açıklama |
|----------|------------|---------------------------------------|
| proje_id | bigint(20) | <fk> Kurum ile ilişkili olan proje id |
| kurum_id | int(11) | <fk> Proje ile ilişkili olan kurum id |

4.2.4 Personel Tablosu

Veri tabanını kullanacak olan kurumun, hem personel bilgilerini tutması hem de yapılan jeodezik ölçmelerin hangisinde hangi personelin yazıcı, gözlemci vb. şeklinde görevli olduğunu tutmak amacıyla *personel* tablosu oluşturulmuştur (Tablo 4.6). *per_id* alanı birincil anahtardır ve bu alanda personelin sicil numarası tek anlamı olacağından birincil anahtar olarak kullanılabilir.

4.2.5 Alet Tablosu

Yapılan ölçmelerde kullanılan aletlerin neler olduğu ve mevcut aletlerin bilgilerini tutmak amacıyla *alet* tablosu oluşturulmuştur (Tablo 4.7). Alet seri numarası, alet ismi ve alet türü bilgilerini içermektedir. Bu tablo daha ayrıntılı bilgiler içerecek şekilde istenilirse genişletilebilir. Örneğin aletin ölçme doğruluğu, varsa sıfır noktası

eki vb gibi teknik özellikleri için de ayrı alanlar açılabilir. Şimdilik bu tablo bu şekilde tasarlanmıştır. İstenirse bu tarz bilgilere bu tablonun *notlar* alanında değinilebilir.

Tablo 4 6: Personel bilgilerinin tutulacağı tablo (*personel*)

| Alanlar | Veri Türü | Açıklama |
|------------|--------------|---|
| per_id | bigint(20) | <pk> Personel sicil numarası |
| kurum_id | int(11) | <fk> Personelin bağlı olduğu kurum id |
| biri_m_id | int(11) | <fk> Personelin bağlı olduğu kurumdaki biri mid |
| per_adi | varchar(50) | Personelin adı. |
| per_soyadi | varchar(50) | Personelin soyadı. |
| per_adres | varchar(250) | Personelin adresi. |
| per_eposta | varchar(100) | Personelin e-posta adresi. |
| per_tel | varchar(15) | Personelin telefon numarası. |
| per_faks | varchar(15) | Personelin faks numarası. |
| per_ceptel | varchar(15) | Personelin cep telefonu numarası. |
| notlar | text | Personele ilişkin genel notlar vb. |

Tablo 4 7: Alet bilgilerinin tutulacağı tablo (*alet*)

| Alanlar | Veri Türü | Açıklama |
|---------------|-------------|---|
| alet_id | int(11) | <pk> Aleti temsil eden tek anlamlı id numarası. |
| alet_seri_no | varchar(50) | Aletin seri numarası. |
| alet_ismi | varchar(50) | Aleti tanımlayıcı isim |
| uretici_firma | varchar(50) | Aleti üreten firmanın ismi. |
| alet_turu | varchar(50) | Alet türü |
| notlar | text | Alete ilişkin genelle notlar vb. |

4.2.6 Meteoroloji Tablosu

Meteoroloji tablosu da herhangi bir istasyonda yapılan meteorolojik ölçmeleri veri tabanında tutmak için tasarlanmış genel tablolardan bir tanesidir (Tablo 4.8). Bu tablonun herhangi bir birincil anahtarı yoktur. *istasyon_id* bu tabloya yabancı anahtar olarak gelmektedir. Herhangi bir ölçme yöntemi için bir istasyonda meteorolojik bilgi gerektiğinde tarih ve istasyon numarası alanları kullanılarak istenilen bilgilere ulaşılabılır. İstenilen zaman aralığında o istasyonda bir meteorolojik ölçme yapılmamışsa, bu sorgulama sonucunda ortaya çıkacaktır. [1]

Tablo 4 8: Meteorolojik bilgilerin tutulacağı tablo (*meteoroloji*)

| Alanlar | Veri Türü | Açıklama |
|----------------|--------------|---|
| istasyon_id | biçim(20) | <fk> Ölçmelerin ait olduğu istasyon. |
| tarih | deciml(20,8) | Julyen takvimi olarak ölçmenin yapıldığı tarih ve saat. |
| basinc | deciml(4,2) | Milibar olarak hava basıncı. |
| kuru_sicaklik | deciml(4,2) | °C olarak havanın kuru sıcaklığı. |
| islak_sicaklik | deciml(4,2) | °C olarak havanın ıslak sıcaklığı. |
| nem | deciml(4,2) | % olarak havanın nem oranı. |
| notlar | text | Meteorolojik genel notlar vb. |

4.3 Yatay Doğrultu Ölçmeleri Tabloları

Yatay doğrultu ölçmeleri oturma ve ölçme tabloları olmak üzere iki ana tablo ile birlikte aletlerle ilişkiyi sağlayacak olan *yataydogoturmaalet* ara tablosundan oluşmaktadır (Sırasıyla Tablo 4.9, Tablo 4.10, Tablo 4.11). [1]

Tablo 4 9: Yatay doğrultu ölçmeleri oturma tablosu (*yataydogoturma*)

| Alanlar | Veri Türü | Açıklama |
|-------------------|--------------|--|
| yataydogoturma_id | biçim(20) | <pk> Yatay doğrultu oturma temsil eden tek anlamı id. |
| proje_id | biçim(20) | <fk> Oturma ait olduğu proje id. |
| baslangic_tarihi | deciml(20,8) | Julyen takviminde ölçmeye başlama tarihi ve saati. |
| bitis_tarihi | deciml(20,8) | Julyen takviminde ölçmenin bitiş tarihi ve saati. |
| gozlemci_id | biçim(20) | <fk> Ölçmede gözlemci olarak görev yapan personelin id numarası. |
| yazici_id | biçim(20) | <fk> Ölçmede yazıcı olarak görev yapan personelin id numarası. |
| kayit_cizelgesi | varchar(50) | Tutulmuş kayıt çizelgesi. |
| alet_yuksekligi | deciml(5,5) | İstasyondaki alet yüksekliği. |
| notlar | text | Oturuma ilişkin genel notlar vb. |

Yatay doğrultu ölçmelerinde oturma aynı istasyondan diğer hedef istasyonlara gözlemci tarafından yapılan ölçmeleri içerir. Yatay doğrultu ölçmelerinde ölçüler istasyon ve oturma bazlı olarak tutulmaktadır. Oturma boyunca durulan nokta ve alet yüksekliği değişmez. Ancak bir oturumda hedef noktalarına birden fazla ölçme (seri) yapmak mümkündür. Bu durumda oturma_id değişmeyecek fakat *yataydogolcu_id* ve *seri_sayisi*, *hedef_id* de sabit kalmak üzere seriler için değişecektir. Diğer taraftan yatay doğrultu ölçmelerinde dürbünün her iki konumunda da ölçme yapmak mümkün olduğundan ölçme tablosunda pozisyon alanı konulmuştur. Ölçme tarihini julyen tarihi olarak girileceği düşünülmüştür. Böylece

tarih içerisinde saat bilgisini de içerecektir. *kayit_cizelgesi* alanı ise arazide ölçme planını gösteren bir kroki vb tutulması durumunda, taranarak bir klasörde resim formatında tutulması, dosya adı ve dizin bilgisinin de bu alanda tutulması amacıyla konulmuştur.

Tablo 4 10 : Yatay doğrultu ölçmeleri ölçme tablosu (*yatay_dog_olcu*)

| Alanlar | Veri Türü | Açıklama |
|---------------------|--------------|--|
| yatay_dog_olcu_id | bigint(20) | <pk> İlgili oturumda yapılan bir ölçmeyi temsil eden tek anlamı id |
| yatay_dog_oturum_id | bigint(20) | <fk> Ölçmenin ait olduğu oturum id |
| istasyon_id | bigint(20) | <fk> Ölçmenin yapıldığı istasyon id |
| hedef_id | bigint(20) | <fk> Ölçmede gözlemlenen hedef id |
| pozisyon | varchar(10) | Dürbünün konusu. |
| seri_sayisi | int(2) | Doğrultu ölçmesinin seri sayısı. |
| yatay_dogrultu | decimal(4,6) | Öçülen yatay doğrultu değeri. |
| notlar | text | Öçmeye ilişkin genel notlar vb. |

Hangi oturumda, hangi istasyonda, hangi aletlerin kullanıldığı bilgisini tutmak amacıyla da *yataydogoturum_alet* ara tablosu oluşturulmuştur. Bu tablonun birincil anahtarı yoktur. *yatay_dog_oturum_id*, *istasyon_id* ve *alet_id* bu tabloya yabancı anahtarlar olarak gelmektedirler.

Tablo 4 11 : Yatay doğrultu ölçmeleri oturum alet ilişkilerini sağlayacak ara tablo (*yataydogoturum_alet*)

| Alanlar | Veri Türü | Açıklama |
|---------------------|------------|--|
| yatay_dog_oturum_id | bigint(20) | <fk> İstasyon ve kullanılan aletler ile ilişkili olan yatay doğrultu oturum id |
| istasyon_id | bigint(20) | <fk> Yatay doğrultu oturumu ve kullanılan aletlerle ilişkili olan istasyon id |
| alet_id | int(11) | <fk> İlgili istasyonda veya hedefte, ilgili oturumda kullanılan alet id |

4.4 Uzunluk Ölçmeleri Tabloları

Uzunluk ölçmeleri tabloları da iki ana ve bir ara tablo olmak üzere 3 tablodan oluşmaktadır (Sırasıyla Tablo 4.12, Tablo 4.13, Tablo 4.14). [1]

Uzunluk ölçmelerinde de bir istasyondan birden fazla istasyona gözlem yaparak uzunluk ölçmesi yapmak mümkün olduğundan oturum mantığı uzunluk ölçmelerinde de değişmeyecektir. Uzunluk ölçmeleri de istasyon ve oturum bazlı olarak tutulacak, istasyon noktasında alet ayarları değişmeyecek, yani bir oturum için sabit olacaktır.

Tablo 4 12 : Uzunluk ölçmeleri durumu tablosu (*uzunluk_oturum*)

| Alanlar | Veri Türü | Açıklama |
|-------------------|---------------|---|
| uzunluk_oturum_id | bigint(20) | <pk> Uzunluk ölçme durumunu temsil eden tek anlı id |
| proje_id | bigint(20) | <fk> Durumun ait olduğu proje id |
| baslangic_tarihi | decimal(20,8) | Julyen takviminde ölçmeye başlama tarihi ve saati. |
| bitis_tarihi | decimal(20,8) | Julyen takviminde ölçmenin bitiş tarihi ve saati. |
| gozlemi_id | bigint(20) | <fk> Ölçmede gözlemi olarak görev yapan personelin id numarası. |
| yazici_id | bigint(20) | <fk> Ölçmede yazıcı olarak görev yapan personelin id numarası. |
| kayit_cizelgesi | varchar(50) | Tutulmuş kayıt çizelgesi. |
| alet_yuksekligi | decimal(5,5) | İstasyondaki alet yüksekliği. |
| yansina_katsayisi | decimal(5,5) | Yansıma katsayısı değeri. |
| notlar | text | Duruma ilişkin genel notlar vb. |

Tablo 4 13 : Uzunluk ölçmeleri ölçme tablosu (*uzunluk_olcu*)

| Alanlar | Veri Türü | Açıklama |
|-------------------|---------------|--|
| uzunluk_olcu_id | bigint(20) | <pk> İlgili durumda yapılan bir ölçüyü temsil eden tek anlı id |
| uzunluk_oturum_id | bigint(20) | <fk> Ölçmenin ait olduğu durumu id |
| istasyon_id | bigint(20) | <fk> Ölçmenin yapıldığı istasyon id |
| hedef_id | bigint(20) | <fk> Gözlenen hedef id |
| hedef_yuksekligi | decimal(5,5) | Gözlenen istasyondaki işaret yüksekliği. |
| seri_sayisi | int(2) | Seri sayısı. |
| egik_uzunluk | decimal(10,6) | İstasyon ve hedef arasındaki eğik uzunluk. |
| notlar | text | Ölçmeye ilişkin genel notlar vb. |

Tablo 4 14 : Uzunluk ölçmeleri durumalet ilişkilerini sağlayacak ara tablo (*uzunluk_oturum_alet*)

| Alanlar | Veri Türü | Açıklama |
|-------------------|------------|---|
| uzunluk_oturum_id | bigint(20) | <fk> İstasyon ve kullanılan aletler ile ilişkili olan uzunluk ölçmeleri durumu id |
| istasyon_id | bigint(20) | <fk> Uzunluk ölçmesi durumu ve aletlerle ilişkili olan istasyon id |
| alet_id | int(11) | <fk> İlgili durumda, ilgili istasyonda veya hedefte kullanılan alet id |

Aynı hedefe birden fazla ölçme yapmak uzunluk ölçmelerinde de mümkün olacağından ölçme tablosunda *seri_sayisi* alanı konulmuştur. Uzunluk ölçmelerinde de *kayit_cizelgesi* alanı, yatay doğrultu ölçmelerindeki ne benzer olarak ölçme planı hazırlanması halinde dosya bilgisini tutmak amacıyla konulmuştur. Uzunluk

ölçmelerinde de istasyon ve hedef noktalarında kullanılan aletlerin uzunluk ölçmeleri oturumları ile ilişkilerini sağlamak için *uzunlukoturum_al et* ara tablosu oluşturulmuştur.

4.5 Düşey Açı Ölçmeleri Tabloları

Düşey açı ölçmeleri de daha önceki ölçme yöntemlerindeki ne benzer olarak 3 tablodan oluşmaktadır. Oturum ve ölçme tabloları olmak üzere iki tablo ile tutulan ölçme değerlerinin yanı sıra *duseyacioturum_al et* tablosu ile istasyonlarda kullanılan alet bilgileri tutulmaktadır (Sırasıyla Tablo 4.15, Tablo 4.16, Tablo 4.17). [1]

Düşey açı ölçmelerinde de oturum aynı istasyondan diğer hedef istasyonlara gözlemci tarafından yapılan, istasyon noktasının ve bu noktadaki alet ayarlarının değişmediği ölçmeleri içerir. Ancak bir oturumda hedef noktalarına birden fazla ölçme (seri) yapmak düşey açı ölçmelerinde de mümkündür. Bu durumda *oturum_id* değişmeyecek fakat *öçme_id* ve *seri_sayisi*, hedef_id de sabit kalacak üzere seriler için değişecektir.

Tablo 4.15: Düşey açı ölçmeleri oturum tablosu (*dusey_aci_oturum*)

| Alanlar | Veri Türü | Açıklama |
|---------------------|---------------|--|
| dusey_aci_oturum_id | bigint(20) | <pk> Düşey açı ölçme oturumunu temsil eden tek anlamı id. |
| proje_id | bigint(20) | <fk> Oturumun ait olduğu proje id |
| baslangic_tarihi | decimal(20,8) | Julyen takvimi nde ölçmeye başlama tarihi ve saati. |
| bitis_tarihi | decimal(20,8) | Julyen takvimi nde ölçmenin bitiş tarihi ve saati. |
| gozlemci_id | bigint(20) | <fk> Ölçmede gözlemci olarak görev yapan personelin id numarası. |
| yazici_id | bigint(20) | <fk> Ölçmede yazıcı olarak görev yapan personelin id numarası. |
| kayit_cizelgesi | varchar(50) | Tutulmuş kayıt çizelgesi. |
| alet_yuksekligi | decimal(5,5) | İstasyondaki alet yüksekliği. |
| notlar | text | Oturuma ilişkin notlar genel vb. |

Ölçme tarihini julyen tarihi olarak girileceği düşünülmüştür. Böylece tarih içerisinde saat bilgisini de içerecektir. Ölçme planı hazırlanması durumunda dosyasının bir dizinde ve dosya dizinin ad bilgisinin veri tabanında tutulması amacıyla burada da oturum tablosunda *kayit_cizelgesi* alanı açılmıştır.

Hangi oturumda, hangi istasyonda, hangi aletlerin kullanıldığı bilgisini tutmak amacıyla da *duseyacioturum_al et* ara tablosu oluşturulmuştur. Bu tablonun da diğer

ara tablolar da olduđu gibi birincil anahtar ı yoktur. *dusey_aci_oturum_id*, *istasyon_id* ve *alet_id* bu tabloya yabancı anahtarlar olarak gelmektedirler.

Tablo 4 16 : Düşey açı ölçmeleri ölçme tablosu (*dusey_aci_olcu*)

| Alanlar | Veri Türü | Açıklama |
|----------------------------|--------------|--|
| <i>dusey_aci_olcu_id</i> | biçim(20) | <pk> İlgili oturumda yapılan bir ölçüyü temsil eden tek anlık id |
| <i>dusey_aci_oturum_id</i> | biçim(20) | <fk> Ölçmenin ait olduğu oturum id |
| <i>istasyon_id</i> | biçim(20) | <fk> Ölçmenin yapıldığı istasyon id |
| <i>hedef_id</i> | biçim(20) | <fk> Gözlenen hedef id |
| <i>hedef_yuksekligi</i> | decimal(5,5) | Hedef yüksekliği. |
| <i>seri_sayisi</i> | im(2) | Seri sayısı. |
| <i>dusey_aci</i> | decimal(4,6) | Öçülen düşey açı değeri. |
| <i>notlar</i> | text | Öçmeye ilişkin genel notlar vb. |

Tablo 4 17 : Düşey açı ölçmeleri oturmalet ilişkilerini sağlayacak ara tablo (*duseyaci_oturum_alet*)

| Alanlar | Veri Türü | Açıklama |
|----------------------------|-----------|---|
| <i>dusey_aci_oturum_id</i> | biçim(20) | <fk> İstasyon ve kullanılan aletler ile ilişkili olan oturum id |
| <i>istasyon_id</i> | biçim(20) | <fk> Oturmalet ilişkili olan istasyon id |
| <i>alet_id</i> | im(11) | <fk> İlgili oturumda, ilgili istasyonda veya hedefte kullanılan alet id |

4.6 Yükseklik Ölçmeleri

Yükseklik ölçmelerine ilişkin veriler oturum ve ölçme tabloları olmak üzere 2 farklı tabloda tutulmaktadır. Bunlar *yuk_olc_oturum* ve *yuk_olc_olcu* tablolarıdır (Sırasıyla Tablo 4 18 ve Tablo 4 19). [1]

Yükseklik ölçmelerinde oturum iki istasyon noktası arasında yapılan gidiş-dönüş yükseklik farkı ölçmeleri sırasında yapılan tüm ölçmeleri içermektedir. Ölçme sırasında bir ileri, bir geri olmak üzere 2 mira kullanılacağı düşünülerek veri tabanı oluşturulmuştur. Gidiş ve dönüş yükseklik farkları, yaklaşık toplam nielman yolu kurulan alet sayısı gibi bilgiler de veri tabanında tutulmaktadır. Oması durumunda kayıt çizelgesi için alan, yükseklik ölçmeleri oturmalet tablosunda da ayrılmıştır.

4.7 GPS Ölçmeleri

Günümüzün modern ölçme tekniklerinden olan GPS ölçmelerinin ve değerlendirme sonuçlarının veri tabanında tutulması amacıyla 6 tablodan oluşan bir yapı tasarlanmıştır.

Tablo 4 18 : Yükseklik ölçmeleri oturum tablosu (*yuk_olc_oturum*)

| Alanlar | Veri Türü | Açıklama |
|---------------------|---------------|--|
| yuk_olc_oturum_id | bigint(20) | <pk> Yükseklik ölçme oturumunu temsil eden tek anlamlı id |
| proje_id | bigint(20) | <fk> Oturumun ait olduğu proje id |
| baslangic_ist_id | bigint(20) | <fk> Başlangıç istasyon id |
| bitis_ist_id | bigint(20) | <fk> Bitiş istasyon id |
| gozlemci_id | bigint(20) | <fk> Ölçmede gözlemci olarak görev yapan personelin id numarası. |
| yazici_id | bigint(20) | <fk> Ölçmede yazıcı olarak görev yapan personelin id numarası. |
| alet_id | int(11) | <fk> Kullanılan alet id |
| mira1_id | int(11) | <fk> Kullanılan mira id |
| mira2_id | int(11) | <fk> Kullanılan mira id |
| gidis_yuk_farki | decimal(4,6) | Gidiş yükseklik farkı. |
| donus_yuk_farki | decimal(4,6) | Dönüş yükseklik farkı. |
| kurulan_alat_sayisi | int(2) | Toplam kurulan alet sayısı. |
| topniv_yolu | decimal(10,6) | Toplam nivelman yolu |
| baslangic_tarihi | decimal(20,8) | Julyen takviminde ölçmeye başlama tarihi ve saati. |
| bitis_tarihi | decimal(20,8) | Julyen takviminde ölçmenin bitiş tarihi ve saati. |
| kayit_cizelgesi | varchar(50) | Tutulan kayıt çizelgesi. |
| notlar | text | Oturuma ilişkin genel notlar. |

Tablo 4 19 : Yükseklik ölçmeleri ölçme tablosu (*yuk_olc_dcu*)

| Alanlar | Veri Türü | Açıklama |
|-------------------|--------------|--|
| yuk_olc_dcu_id | bigint(20) | <pk> İlgili oturumda yapılan bir ölçüyü temsil eden tek anlamlı id |
| yuk_olc_oturum_id | bigint(20) | <fk> Ölçmenin ait olduğu oturum id |
| geri_nok_id | bigint(20) | Geri nokta numarası. |
| geri_mira_no | int(11) | <fk> Kullanılan geri mira id |
| geri_orta_okuma1 | decimal(2,6) | Geri orta okuma 1. |
| geri_orta_okuma2 | decimal(2,6) | Geri orta okuma 1. |
| geri_ust_okuma | decimal(2,6) | Geri üst okuma. |
| geri_alt_okuma | decimal(2,6) | Geri alt okuma. |
| ileri_nok_id | bigint(20) | İleri nokta numarası. |
| ileri_mira_no | int(11) | <fk> Kullanılan ileri mira id |
| ileri_orta_okuma1 | decimal(2,6) | İleri orta okuma 1. |
| ileri_orta_okuma2 | decimal(2,6) | İleri orta okuma 2. |
| ileri_ust_okuma | decimal(2,6) | İleri üst okuma. |
| ileri_alt_okuma | decimal(2,6) | İleri alt okuma. |
| notlar | text | Ölçmeye ilişkin genel notlar. |

Örüntü tablosunda (Tablo 4.20), proje bilgisi ile birlikte, oturumun yapıldığı zaman aralığı julyen zamanı olarak girilmektedir. Ayrıca kayıt çizelgesini saklamak amacıyla dosya dizin bilgisini içerecek bir alan da oturum tablosunda mevcuttur. Diğer taraftan projede kullanılan eğitim açısı ve ölçme aralığı bilgilerinin istendiğinde oturum bazlı olarak girilebilmesini sağlamak amacıyla oturum tablosunda bu bilgilere de yer verilmiştir.

Ölçmeler sırasında istasyonlarda kullanılan aletlerin bilgilerini tutmak amacıyla *gps_oturum_alat* ara tablosu oluşturulmuştur (Tablo 4.23).

Ayrıca GPS ölçmelerinde gözlemci, yazıcı gibi bir personel kullanımı söz konusu olduğundan istasyonlarda görev alan kişi veya kişileri tutmak amacıyla *gps_oturum_personel* ara tablosu oluşturulmuştur. Bu yapı bir istasyonda birden fazla personel bulundurmaya karşılayacak yapıdadır (Tablo 4.24).

Her istasyon için gözlem (observation), yönlendirme (navigation) ve meteoroloji (meteorology) dosyaları bir dizin yapısı içerisinde proje ve nokta bazlı olarak tutulmakta ve bu dosyaların ad bilgileri ise veri tabanında tutulmaktadır (Tablo 4.21). İstendiğinde bilinen dizin yapısı ve dosya adı kullanarak bu bilgilere ulaşmak mümkün olmaktadır.

Diğer taraftan değerlendirme sonrasında elde edilen dengeleme ve baz çözümlerinin dosyaları da ölçme dosyalarında olduğu gibi bir dizin yapısı içerisinde dosya olarak tutulmakta, dosya adları ise veri tabanında yer almaktadır (Tablo 4.22). Değerlendirmede görev alan personel bilgisini tutmak amacıyla da *gps_degerlendirme_personel* ara tablosu oluşturulmuştur (Tablo 4.25).

Tablo 4.20: GPS ölçmeleri oturum tablosu (*gps_oturum*)

| Alanlar | Veri Türü | Açıklama |
|------------------|---------------|--|
| gps_oturum_id | bigint(20) | <pk> GPS oturumun temsil eden tek anahtarı id |
| proje_id | bigint(20) | <fk> Oturumun ait olduğu proje id |
| gps_oturum_adi | varchar(50) | Oturum ismi. |
| baslangic_tarihi | decimal(20,8) | Julyen takviminde ölçmeye başlama tarihi ve saati. |
| bitis_tarihi | decimal(20,8) | Julyen takviminde ölçmenin bitiş tarihi ve saati. |
| kayit_cizelgesi | varchar(50) | Tutulmuş kayıt çizelgesi. |
| veri_araligi | int(2) | Süre olarak veri kayıt aralığı. |
| yukseklik_acisi | int(2) | Alınan eğim açısı. |
| notlar | text | Oturuma ilişkin genel notlar vb. |

Tablo 4 21 : GPS ölçmeleri ölçme tablosu (*gps_olcu*)

| Alanlar | Veri Türü | Açıklama |
|----------------------|--------------|--|
| gps_olcu_id | bigint(20) | <pk> GPS ölçmesini temsil eden tek anahtarı id |
| gps_degerlendirme_id | bigint(20) | <fk> Ölçmenin hangi değerlendirme de kullanıldığı nı gösteren değerlendirme id |
| gps_oturum_id | bigint(20) | <fk> Ölçmenin ait olduğu GPS oturum id |
| istasyon_id | bigint(20) | <fk> Ölçenin ait olduğu istasyon |
| alet_yuksekligi | decimal(5,5) | Alet yüksekliği. |
| olcu | varchar(50) | Gözlem dosyası. |
| navi_gasyon | varchar(50) | Yönlendirme dosyası. |
| meteoroloji | varchar(50) | Meteoroloji dosyası. |
| notlar | text | GPS ölçmesine ilişkin genel notlar vb. |

Tablo 4 22 : GPS ölçmeleri değerlendirme tablosu (*gps_degerlendirme*)

| Alanlar | Veri Türü | Açıklama |
|----------------------|------------|--|
| gps_degerlendirme_id | bigint(20) | <pk> Değerlendirme |
| dengeleme_sonucari | int(2) | İlgili ölçmeler sonucunda elde edilen dengeleme sonuçları. |
| baz_cozumleri | int(2) | İlgili ölçmeler sonucunda elde edilen baz çözümleri. |
| notlar | text | Değerlendirme sonuçlarına ilişkin genel notlar vb. |

Tablo 4 23 : GPS ölçmeleri oturum alet ilişkilerini sağlayacak ara tablo (*gps_oturum_alet*)

| Alanlar | Veri Türü | Açıklama |
|---------------|------------|---|
| gps_oturum_id | bigint(20) | <fk> İstasyon ve kullanılan aletler ile ilişkili olan GPS oturum id |
| istasyon_id | bigint(20) | <fk> İlgili oturumda ilgili aletin kurulduğu istasyon id |
| alet_id | int(11) | <fk> İlgili oturum ve istasyonda kullanılan alet id |

Tablo 4 24 : GPS ölçmeleri oturum personel ilişkilerini sağlayacak ara tablo (*gps_oturum_personel*)

| Alanlar | Veri Türü | Açıklama |
|---------------|------------|--|
| gps_oturum_id | bigint(20) | <fk> Personelin görev yaptığı oturum id |
| istasyon_id | bigint(20) | <fk> Oturumda personelin hangi istasyonda bulunduğu. |
| per_id | int(11) | <fk> Belli bir oturumda, belli bir istasyonda görev alan personel id |

Tablo 4 25 : GPS ölç neleri deęerlendir ne-personel iliřkilerini saęlayacak ara tablo
(*gpsdegerlendir ne_personel*)

| Al anlar | Veri Türü | Açı k l a ma |
|------------------------|------------------|---|
| per_id | int(11) | <fk> İlgili deęerlendir nede görev alan personel id |
| gps _degerlendir ne_id | bi gi nt(20) | <fk> Perseonelin görev aldı ğı deęerlendir ne id |

5. ÖRNEK GPS VERİ TABANI ve WEB YÖNETİM ARAYÜZÜ TASARIMI

5.1 Çalışmaya Genel Bakış

Bir önceki bölümde jeodezik bir veri tabanında hangi ölçülerin ve ölçü ile ilişkili hangi bilgilerin (proje, alet, personel, değerlendirme vb. gibi) tutulabileceği anlatılmış ve veri tabanı tablo yapıları oluşturulmuştur. Bu yapılar anlatılırken bazı ara tablolar bahsedilmiştir. Aslında bu ara tabloların görevi veri tutmak değil veri tutan tablolar arasındaki ilişkileri sağlamaktır. Tüm bu veri tabanı yapısının kurulabilmesi için tutulması istenen veriler seçilmeli, gruplanmalı ve aralarındaki ilişkiler belirlenmelidir. Bu ilişkilerin belirlenmesi kavramsal olarak veri tabanı modelinin oluşturulması anlamına gelmektedir. Daha sonra bu ilişkisel yapıyı sağlamak için ara tablolar oluşturulması gerekebilir. Ara tablolarla beraber oluşan veri tabanının son haline fiziksel model adı verilir. Diğer bir deyişle fiziksel model veri girilmeye hazır hale gelmiş tablo yapılarını ve tablolar arası ilişkileri içerir.

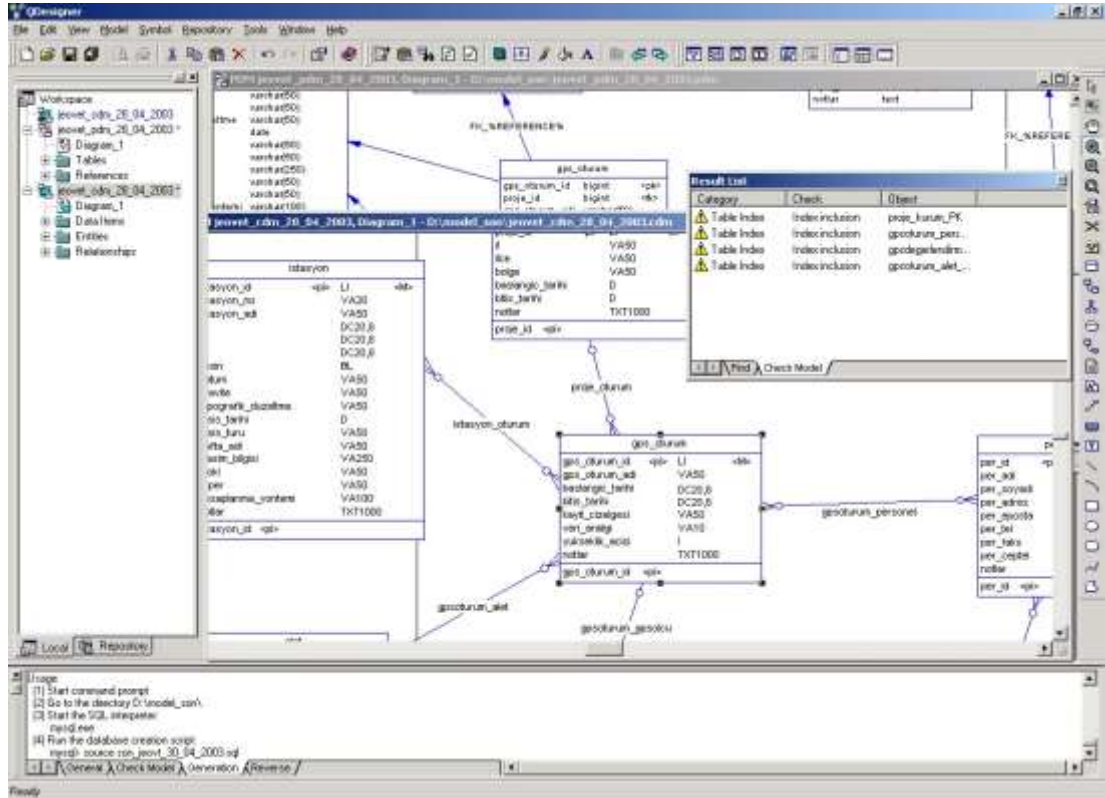
Bir önceki bölümde hazır olarak verilen tabloların nasıl oluşturulduğunun daha açık ve net olarak anlaşılabilmesi için bu bölümde tasarımı aşamaları GPS veri tabanı için anlatılacaktır. Veri tabanının sürdürülebilirliğinin ve etkin kullanımının sağlanması açısından kavramsal ve sonuçta fiziksel olarak sağlam bir yapıda tasarlanması oldukça önemlidir. Günümüzün modern ölçme tekniklerinden birisi olan GPS ölçme tekniği kullanılarak elde edilmiş bilgi, ölçme ve değerlendirmelerin gerektiğinde tekrar kullanılabilirleri amacıyla ilişkisel bir veri tabanı içerisinde nokta ve proje bazlı olarak nasıl tutulabileceği; bu nitelikte bir veri tabanının kavramsal ve fiziksel olarak nasıl modellenebileceği gösterilecektir.

GPS veri tabanı tasarım aşamaları anlaşıldığında diğer ölçme grupları için bu tasarımın nasıl yapıldığı da anlaşılacaktır. Hazırlanan fiziksel model herhangi bir veri tabanı yazılımında kullanılabilir niteliktedir. Zaten tasarım veri tabanı yazılımına göre farklılık göstermeyen bir aşamadır. Bu çalışmanın sonrasında veri tabanı MySQL veri tabanı yazılımında oluşturulmuştur.

Hazırlanan veri tabanı her ne kadar veri girilmeye hazır bir yapı da olsa tüm kullanıcılar veri tabanı yapıları ve kullanım konusunda fazla bilgili veya tecrübeli olmayacaklarından, arayüzlerle veri girişi, sorgulama ve analizler kolay hale getirilmiştir. Bazı veri tabanı yazılımların bu amaca yönelik olarak değişik form arayüzleri geliştirme programları mevcuttur. Bu çalışmada yapılan GPS veri tabanı uygulaması için bu tarz bir program kullanılmamış ve arayüzler PHP web programlama dili kullanılarak oluşturulmuştur.

5.2 GPS Veri Tabanı Tasarımı

Tasarlanan GPS veri tabanında bir projede gerçekleştirilen GPS gözlemlerine ilişkin bilgilerin gerektiğinde daha sonra kullanılabilmesi için, nokta ve proje bazlı olarak tutulması öngörülmüştür. Tasarım için gerekli ön çalışma önce kağıt üzerinde yapıldıktan sonra, “Qdesigner” veri tabanı tasarım programı kullanılarak kavramsal ve fiziksel model tasarlanmıştır. “Qdesigner” yazılımı tasarım ekran görüntüsü Şekil 5.1’de verilmiştir.



Şekil 5.1 “Qdesigner” veri tabanı tasarımı yazılım ekran görüntüsü

Veri tabanının kavramsal ve fiziksel olarak sağlıklı bir şekilde tasarımı ve kullanılabilir duruma gelmesi aşağıdaki aşamalarda gerçekleşmektedir.

- Tutulması istenen verilerin belirlenmesi,
- Verilerin niteliklerine göre gruplandırılması,
- “Kavramsal Veri Modeli”nin oluşturulması,
- “Fiziksel Veri Modeli”nin oluşturulması,
- “Fiziksel Veri Modeli”nin düzeltilmesi,
- Veri tabanının oluşturulmasıdır [7].

Tablo 5.1: Verilerin Niteliklerine Göre Gruplandırılması

| Proje | Kurum | Personel | İstasyon |
|---|--|--|---|
| İl İlçe Bölge Başlangıç Tarihi Bitiş Tarihi | Adı Adres Telefon Faks E-posta İlgili Kişi Kurumdaki Birimler | Sicil no Kurum Kurumdaki Birim Adı Soyadı Adres E-posta Telefon Faks | No Adı X Y Z Datum Gravite Topoğrafik Düzeltilme Tesis Tarihi Tesis Türü Pafta Adı Ulaşım Bilgisi Kırık Röper Hesaplama Yöntemi Diğer Projelerde Hesaplanmış Koordinatları ve Hesap Yöntemi |
| Alat | GPS Cihazı | GPS Değerlendirme | |
| Seri No Adı Üretici Firma Türü | Başlangıç Zamanı Bitiş Zamanı Alat Yüksekliği Kayıt Çözünürlüğü Veri aralığı Yükseklik Açısı Ölçü Dosyası Navigasyon Dosyası Meteoroloji Dosyası | Dengeleme Sonuçları Baz Çözümleri | |

Tutulması İstenen Verilerin Belirlenmesi : Bir projede yapılan GPS ölçmelerine ilişkin hangi bilgilerin ilerideki ölçme ve değerlendirmelerde kullanılmak üzere yeterli olacağı göz önüne alınarak, tutulacak veriler belirlenmiştir. Bu verilerin neler olduğu bir sonraki aşamada boyutlandırılmış (gruplandırılmış) olarak Tablo 5.1’de gösterilmektedir. [7]

Verilerin Niteliklerine Göre Gruplandırılması : Gruplandırma tablo yapılarının belirlenmesinde kolaylık sağlayan bir aşamadır. Yapılan bu gruplandırma; tablo yapılarını hemen hemen belirlemekle birlikte, aynı gruptaki verilerin mutlaka aynı tabloda olacağı gibi bir durum söz konusu değildir. Örneğin gerekirse kurum grubu bir bilgisi de içermesine rağmen ilerideki aşamalarda kavramsal model tasarımı da görüleceği gibi ayrı bir tablo olarak tasarlanmıştır. [7]

Kavramsal Veri Modelinin Oluşturulması : Verilerin gruplandırılması yapıldıktan sonra, ilgili veri gruplarının tablolarda nasıl tutulacağı ve bu tablolar arası ilişkilerin nasıl olacağı belirlenmiştir. Kağıt üzerinde yapılan bir ön çalışmadan sonra “Qdesi gner” yazılı mında tasarımı devamedilmiştir. Kavramsal modelde tablo yapısı belirlenirken veri tabanının en etkin nasıl kullanılabileceği ve veri tabanında veri tekrarı yapılmadan bilgilerin nasıl tutulabileceği göz önüne alınmıştır. “Qdesi gner” yazılı mında oluşturulan kavramsal model, tablolar, veri tipleri ve “Qdesi gner” yazılı mını tasarımı mekranı görüntüsü Şekil 5.2’de gösterilmiştir. [7]

Belirlenen tablolar ve tablolar arası ilişkilerin ne şekilde olabileceği göz önüne alınarak, tablolar arası ilişkilerin yapısı belirlenmiştir. Belirlenen ilişki yapı aşağıdaki gibidir. [7]

Proje-Kurum : Çok-Çok (Bir projede birden fazla kurumun ortak çalışabileceği, bir kurumun da birden fazla proje yapabileceği göz önüne alınmıştır.)

Kurum Birim : Bir-Çok (Kuruma yeni birimler eklenebileceği dâsılığ göz önüne alınarak, kurum boyutunda yer alan birim ayrı bir tabloda gösterilmiştir. Bir kurumda birden fazla birim olacağı göz önüne alınarak ilişki bir-çok olarak kurulmuştur.)

Birim Personel : Bir-Çok (Bir birimde birden fazla personel olabileceği ve her personelin sadece bir birime ait olabileceği göz önüne alınmıştır.)

İstasyon- GPS Oturum : Çok-Çok (Bir istasyonda birden fazla oturum yapılabileceği ve bir oturumda birden fazla istasyon kullanılacağı göz önüne alınarak ilişki çok-çok olarak belirlenmiştir.)

GPS Oturum Alet : Çok-Çok (Bir oturumda birden fazla alet kullanılacağı ve bir aletin de birden fazla oturumda kullanılacağı göz önüne alınarak ilişki çok-çok olarak belirlenmiştir.)

GPS Oturum GPS Ölçü : Bir-Çok (Bir oturumda birden fazla istasyona ait ölçme dosyaları bulunacağından ve bir ölçme dosyası sadece bir oturuma ait olabileceğinden ilişki bir-çok olarak alınmıştır.)

GPS Oturum Alet : Çok-Çok (Bir oturumda birden çok alet kullanılabileceği ve bir aletin de birden fazla oturumda kullanılabileceği göz önüne alınarak ilişki çok-çok olarak belirlenmiştir.)

GPS Oturum Personel : Çok-Çok (Bir oturumda birden çok personelin görev alacağı ve bir personelin de birden fazla oturumda görev yapacağı göz önüne alınarak ilişki çok-çok olarak alınmıştır.)

GPS Değerlendirme- GPS Ölçme : Bir-Çok (Bir değerlendirme de birden çok ölçme dosyası kullanılacağı ve her ölçmenin bir değerlendirme ye ait olacağı göz önüne alınarak ilişki bir-çok olarak belirlenmiştir.)

GPS Değerlendirme- Personel : Çok-Çok (Bir değerlendirme de birden çok personelin görev alacağı ve bir personelin de birden fazla değerlendirme de yer alacağı göz önüne alınarak ilişki çok-çok olarak alınmıştır.)

Fiziksel Veri Modelinin Oluşturulması : Kavramsal model “Qdesigner” yazılım ile oluşturulduğundan, fiziksel model otomatik olarak yazılım tarafından oluşturulmuştur. Fiziksel model ve “Qdesigner” yazılımındaki ekran görüntüsü Şekil 5.3’te gösterilmektedir. Çok-çok ilişkilerin sağlanması için ara tablolar oluşturulmuş ve bu şekilde görülmektedir. Örneğin istasyon proje çok-çok ilişkisini sağlamak için *istasyon_proje* tablosu otomatik olarak oluşturulmuş ve *istasyon_id* ile *proje_id* alanları bu tabloya yabancı anahtar olarak eklenmiştir. [7]

Fiziksel Veri Modelinin Düzeltmesi : Fiziksel model veri tabanı yazılımına aktarılmadan önce gözden geçirilerek düzeltmeler yapılmıştır. Alet tablosuna girilen bilgiler, istasyon ile ilişkilendirildiğinden ve bu tablodaki bilgilerin istasyon ile

olan ilgisinin gerekli olacağı düşüncesiyle fiziksel modelde oluşan *gpsoturum_alat* tablosu istasyon tablosuyla ilişkilendirilmiş; bu tabloya *istasyon_id* yabancı anahtar olarak eklenmiştir. Yapılan GPS ölçmelerinin hangi istasyona ait olacağı bilgisinin girilebilmesi için *gps_olcu* tablosu istasyon tablosu ile ilişkilendirilmiş ve *gps_olcu* tablosuna *istasyon_id* yabancı anahtar olarak eklenmiştir. Ourlarda görev alan personelin hangi istasyonda görev aldığı bilgisini tutabilmek amacıyla da *gps_oturum* ve *personel* tablolarını ilişkilendiren *gpsoturum_personel* tablosu, *istasyon* tablosu ile ilişkilendirilerek *istasyon_id* bilgisinin bu ara tabloya yabancı anahtar olarak eklenmesi sağlanmıştır. *gps_oturum* ve *istasyon* ilişkisi *gpsoturum_alat* tablosunda tutulduğundan, fiziksel modelde kendiliğinden oluşan *gpsoturum_istasyon* ihtiyaç olmadığına karar verilmiş ve bu tablo silinerek iptal edilmiştir.

Veri Tabanının Oluşturulması : Son durumunu alan fiziksel veri tabanı “Qdesigner“ yazılım ile SQL komut dosyası haline dönüştürülmüştür. SQL komutlarını içeren bu dosya yardımıyla veri tabanı, istenilen veri tabanı yazılımında oluşturulabilir. Her veri tabanı yazılımının SQL komut dosyalarını işleyebilme özelliği vardır. Aksi takdirde veri tabanı tek tek SQL komutları ile, bir SQL komut arayüzü ile veya kullanacağınız veri tabanına özel bir arayüz kullanılarak tasarımı tarafından ayrıca oluşturulmak zorunda kalacaktır.

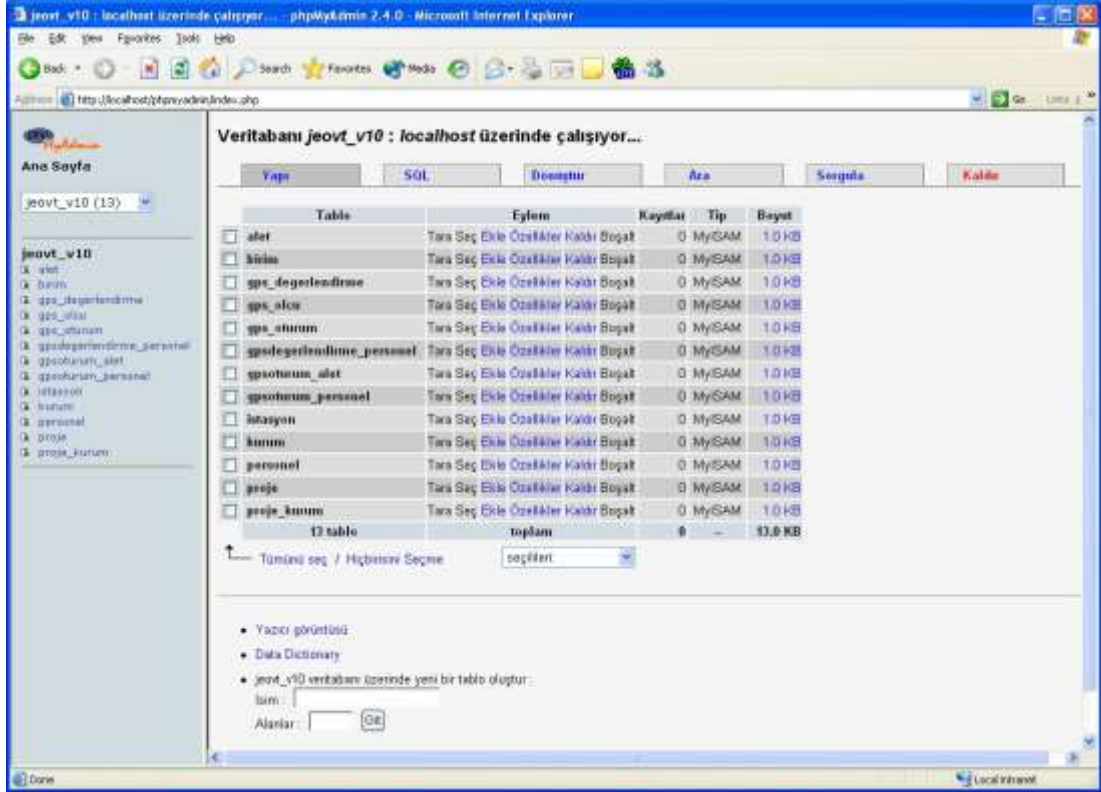
Tasarlanan ilişkisel veri yapısı MySQL3.23 versiyonu için SQL komut dosyası haline dönüştürülmüş ve PHP ile yazılmış bir MySQL yönetim arayüzü olan “php My Admin” yazılım yardımıyla MySQL veri tabanına aktarılmıştır. “php My Admin” arayüzü ve bu arayüzde tabloların görünümü Şekil 5.4’te gösterilmektedir.

5.3 Web Yönetim Arayüzü Tasarımı

Hazırlanan GPS veri tabanına kayıt girişi, güncellenmesi, silinmesi ve veri tabanında arama, sorguların yapılabilmesi gibi işlemler için arayüzler PHP web programlama dili kullanılarak oluşturulmuştur.

Bu bölümün başında da değinildiği gibi arayüz hazırlanmasının en önemli nedenlerinden bir tanesi veri tabanı kullanıcısının işini kolaylaştırmaktır. Veri tabanını kullanacak olan ilgili kişilerin veri tabanı yapısını tam olarak bilmeleri mümkün olmadığından veya bu bazı kullanıcılar için bu mümkün olsa da veri

tabanını kullanabilecek kişi sayısını artırmak, sıradan kullanıcıların da veri tabanının kullanılabilirliğini sağlamak için kullanıcı dostu (userfriendly) arayüzler hazırlanmalıdır.



Şekil 5.4 phpMyAdmin MySQL Veri tabanı yöneti arayüzü

Diğer taraftan hazırlanacak bir arayüz ile kullanıcılara istenilen yetkilerin verilmesi, istenilen arayüzleri kullanılabilirliği ve bazılarının yasaklanması sağlanabilir. Her ne kadar veri tabanı yetkilendirme özelliklerini kullanarak kullanıcı, bilgisayar (host), veri tabanı ve tablo bazında yetkilendirme yapılabilse de bu bir arayüz ile PHP kodu kullanılarak yapıldığında veri alanı bazında dahi bir kısıtlama yapılabilir. Siz arayüzde ne hazırladıysanız kullanıcı sadece onu görür. Ayrıca kullanıcı doğrudan veri tabanını kullanmadığından daha yüksek bir güvenlik sağlanmış olur. PHP arayüz MySQL veri tabanı ile kullanıcı arasında bir tampon bölge olarak düşünülebilir. Böylece iki aşamalı bir güvenlik sağlanmış olur.

PHP arayüzü kullanılmasının en büyük avantajlarından bir tanesi de hazırlanacak veri tabanının web'e açılabilir olmasıdır. Bu çalışmada PHP seçilmesinin esas nedeni veri tabanının web'e ve izin verilen kullanıcıların uzaktan erişimine açılabilirliğidir.

Bu çalışmada PHP'nin oturum (session) ve çerez (cookie) özellikleri kullanılarak kullanıcının oturum açması sonrasında kullanabileceği bir web arayüzü oluşturulmuştur. Kayıt girişi yapılacak olan sayfalar Şekil 5.5'te de gösterildiği gibi web sayfalarında sık sık görülmeye alıştığımız form yapısındadır.

The screenshot shows a web browser window with the title 'İstasyon İşlemleri - Microsoft Internet Explorer'. The address bar shows 'http://localhost/veri/istasyon.php?no=address'. The main content area displays a registration form with the following fields and elements:

- Ad:** Text input field.
- Soyad:** Text input field.
- Adres:** Text input field.
- Telefon:** Text input field.
- E-posta:** Text input field.
- Parola:** Text input field.
- Parola Tekrar:** Text input field.
- Gözetim:** A dropdown menu with 'Gözetim' selected.
- Gözetim:** A button.
- Gözetim:** A button.

Şekil 5.5 İstasyon için kayıt giriş formu örneği

Kraki, röper, kayıt çizelgesi, ölçü dosyası, navigasyon dosyası, meteoroloji dosyası, dengeleme sonuçları ve bazı çözümleri gibi dosyaların veri tabanında değil de bir klasör yapısı içerisinde tutulması uygun görülmüştür. Böylece veri tabanının fazla büyümesi de engellenmiş olacaktır. Bu klasörün güvenliği açısından uygun olan web alanı dışında tutulmasıdır. Aksi halde istenmeyen kullanıcılar da bu dosyalara erişebilirler. Bu dosyaların web alanında tutulması düşünüldüğünde ise klasör şifreli hale getirilerek güvenliği sağlanmalıdır. Konfigürasyon dosyası olarak kullanılan *conf.php* dosyasında bu klasörün nerde tutulacağı belirtilmelidir (*conf.php* dosyası bu bilgilerin yanında MySQL'e bağlantı için kullanılacak kullanıcı adı, kullanıcı şifresi, veri tabanı ismi bilgilerini de tutmaktadır).

Veri klasörü içerisinde istasyon ve proje adlı iki alt klasör içermektedir. Yeni bir istasyon eklendiğinde istasyon dizini altında *istasyon_no* bilgisi ile bir alt dizin

açılmakta ve kroki, röper dosyaları bu klasöre atılarak dosya isimleri veri tabanında saklanmaktadır. Yeni bir proje oluşturulduğunda ise proje isminin ilk 8 karakteri alınarak proje dizini altında bir alt dizin oluşturulmakta, bu dizinin altında ise GPS ölçülerini ve değerlendirme sonuçlarını tutmak amacıyla bir alt dizin oluşturulmaktadır. Yine bu alt dizin bu bilgileri tutmak amacıyla oturum ve değerlendirme adları altında iki alt dizin içermektedir. Bu alt dizinlerde ilgili dosyalar *gps_oturum_id* ve *gps_degerlendirme_id* bilgileri ile açılan klasörlerde tutulmaktadır.

Hazırlanan veri tabanı yapısı ve PHP kullanıcı arayüzünden oluşan yazılım paketi JeoVT- V1.0 (**Jeodezik Veri Tabanı Versiyon 1.0**) olarak isimlendirilmiştir. Yazılımın kullanıcı arayüzü, veritabanı yapısı güncelleştirilmesi devam edecek ve şu an mevcut olan ölçme türleri de yazılıma eklenecektir. Yazılım hakkındaki en güncel bilgilere <http://jeovt.ins.itu.edu.tr> veya <http://jeovt.ifbi.mitu.edu.tr> bağlantılarından ulaşılabilir.

6 SONUÇLAR ve ÖNERİLER

21. yüzyılda insanlığın elindeki en önemli silahlardan birisinin de bilgi olduğu unutulmamalıdır. Bilginin etkin bir silah olarak kullanılabilmesi ise ona doğru ve hızlı olarak erişilebilmesine bağlıdır. İşlenmeyen bir bilgiye ise hızlı olarak ulaşılması oldukça zordur. Bu bakımdan bilgi, bir altın madeniye benzetilebilir. Bu madende ne kadar altın olursa olsun, işlenmediği sürece gerçek değerine ulaşamayacaktır. Altınlar gerçek değerlerine; birileri onları bulup, çıkarıp, ayrıştırıp, insanların kullanımına sunduklarında ulaşmış olacaklardır. Günümüzde ise büyük boyuttaki bilgiyi, bu niteliklerde saklayabilmek veri tabanlarıyla mümkün olabilmektedir.

Bu çalışmada jeodezik bir veri tabanında hangi ölçülerin ve ölçme ile ilişkili hangi bilgilerin (proje, alet, personel, değerlendirme vb.) tutulabileceği anlatılmış ve veri tabanı tablo yapıları oluşturulmuştur. Gravite ölçmelerine, astronomik ölçmelere ve detay ölçmelerine bu çalışmada yer verilmiştir. Bu ölçme yöntemlerinin de jeodezik veri tabanı içerisine dahil edilmeleri gerekmektedir. Tasarım aşamasında bu durum düşünülmüş, yapılarında bu ölçmeleri ve geliştirildiklerinde yeni ölçme tekniklerini içine alabilecek nitelikte tasarlanmıştır. Tez kapsamında ilişkisel veri yapısı tasarım bölümünde de değinildiği gibi, ilişkisel veri tabanlarının en büyük özelliklerinden biri, yeni veri gruplarının mevcut yapıya çok kolay şekilde eklenebilmesidir.

İlişkisel veri modeli çıktığı günden bugüne oldukça fazla kullanılmış ve diğer modellere üstünlüğünü kabul ettirmiştir. Verinin tutulmasında, güncellenmesinde ve yeni veri gruplarının eklenmesinde oldukça etkin kullanılabilen bu veri modelinin başarısı, bu çalışmada da ortaya konmuştur.

Proje, istasyon, kurum personel, alet, meteoroloji genel tabloları ve GPS gözlemleri için PHP web programlama dili kullanılarak kayıt giriş, güncelleme ve sorgu arayüzleri oluşturulmuştur. Diğer ölçme yöntemleri için veri giriş arayüzleri yapılması planlanmış olup çalışmalar devam etmektedir. Zaten veri yapısı ve veri

giriş arayüzleri sürekli gelişmeye açıktırlar ve bu çalışmaların son bulması gibi bir durumsöz konusu değildir.

Hazırlanan arayüzler, veri tabanı yöneticisi niteliğinde bir kullanıcıya göre oluşturulmuştur. Bu kullanıcı, veriler üzerinde her türlü değişikliği yapma hakkına sahiptir. Kısıtlı kullanıcı için yeni kullanıcı arayüzleri hazırlanması da gereklidir ve buna yönelik çalışmalar da devam edecektir.

Hazırlanan yazılımın başka bir eksikliği de kullanıcıların yaptıkları işlemlere ait kayıt (log) dosyaları tutmasıdır. Kayıt tutma özelliğinin eklenmesi kullanıcıların yaptığı işlemleri takip etmek ve denetlemek için oldukça önemlidir. Diğer taraftan kullanıcı arayüzleri her ne kadar basit ve anlaşılır tasarlanmaya çalışılmış olsa da yazılım detaylı olarak anlatan bir elektronik el kitabı (manual) ve yardım dosyası (help) yazılmasına eklenmelidir. İlerleyen çalışmalarda bu özellikler de yazılıma dahil edilecektir.

Veri tabanının tasarımı sırasında, en azından fiziksel veri tabanı tasarımı, bir veri tabanı tasarımı programı kullanılması faydalıdır. Bu programlar; modeli kontrol etme, kaba hataları ortaya çıkarabilme ve sonrasında modeli SQL komut dosyalarına dönüştürerek tasarımı fiziksel modelin oluşturulması sırasında gereksiz vakit kaybından kurtarma özellikleri nedeniyle tercih edilmektedirler.

Bu çalışma bilginin paylaşımı ve sürdürülebilirliğini sağlamakta önemli bir görev üstlenen veri tabanlarının tasarımı bu sürdürülebilirlik ve etkin kullanımda ne kadar önemli olduğunu ortaya koymaktadır. Bu nedenle veri tabanı tasarımı sağlıklı bir şekilde yapılmalı ve bu tasarım sırasında tasarım konusunda uzman kişiler ile ilgili veri tabanını isteyen disiplin tarafından ortak bir çalışma yürütülmelidir. İlişkisel yapı, en başta sağlıklı kurulmalıdır. Aksi takdirde veri tabanı tasarımı yapılacak hatalar ileride büyük sorunlara neden olabilir ve veri tabanını etkin bir şekilde kullanımı dunsuz yönde etkileyebilir.

Veri tabanına ulaşabilecek ilgili insan sayısının arttırılabilmesi ve veri tabanının web'e açılabilmesi için kullanıcı arayüzü hazırlama programı olarak seçilen PHP web programlama dilinin oldukça başarılı olduğu görülmüştür. Aynı zamanda PHP programının iyi bir veri tabanı yazılımı olan ve her geçen gün yeni özellikler katılarak geliştirilen, açık kaynak kodlu MySQL ile olan mükemmel uyumu, MySQL

veri tabanı işlemleri için birçok komut içerir, bu ikilinin seçiminin doğru bir karar olduğunu göstermiştir.

JeoVT- VI. 0 yazılımının geliştirilmesi ne İTÜ İnşaat Fakültesi Jeodezi Anabilim Dalı bünyesinde devam edilecektir. Yazılım hakkında en güncel bilgiler ve son sürümleri için JeoVT resmi sitesi olan <http://jeovt.ins.itu.edu.tr> bağlantısından veya <http://jeovt.ifbim.itu.edu.tr> ve <http://www.jeovt.cjb.net> yansılardan bilgi alınabilir.

KAYNAKLAR

- [1] **Kaynarca, A**, 1994. A Geodetic Database Design Architecture, *MSc Thesis*, Boğaziçi University, İstanbul.
- [2] **Codd, E E**, 1970. Relational Model of Data for Large Shared Data Banks, pp. 377-387, *Communications of the ACM* Vol. 13, No. 6
- [3] **Gündüz, D**, 2002. Veritabanlarına Giriş Seminer Notları, *IV. Akademik Bilişim Konferansı*, Konya
- [4] **Özalan, Ö**, 2002. Web Üzerinden Kütüphane Veritabanı Otomasyonu, *Lisans Bitirme Ödevi*, Mühendislik Bilimleri Bölümü, İTÜ Fen Edebiyat Fakültesi, İstanbul.
- [5] **T C Başbakanlık** 2000. Uusal Bilgi Sistemi, pp. B7- B19, Mayıs 2000, Ankara.
- [6] **Doğan E.** 2003. Kişisel Web Sitesi, <http://erkdogan.netteyim.net/>
- [7] **Özalan, Ö, Çelik, R N**, 2003. Veritabanı Tasarımı ve Örnek GPS Veritabanı, 9 *Türkiye Harita Bilimsel ve Teknik Kurultayı*, Bilkent Hotel, Ankara, 31 Mart-4 Nisan, s. 615-626.
- [8] **Şanlı, İ.**, 1995. GPS Veritabanı Tasarımı ve Marmara GPS Projesine Uygulanması, *Yüksek Lisans Tezi*, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [9] **Çelik, R N, Ölüde mir M T**, 1993. Standart Veri Formatı Geliştirme ve Yararları, Prof. Dr. H Wolf Jeodezi Sempozyumu, İstanbul, 3-5 Kasım s. 491-499.
- [10] **Gorman, M M**, 1991. Database Management Systems, Butterworth Heinemann Weekly, Massachusetts.
- [11] **Wederhold, G**, 1983. Database Design, McGraw Hill, New York
- [12] **Gündüz D.** 2002. Kişisel Web Sitesi, <http://www.gunduz.org>
- [13] **Anonim.** 2003. MySQL Resmî Web Sitesi, <http://www.mysql.com>

- [14] **Anonim**. 2003. PHP Res m i W e b S i t e s i , [http:// www.php.net](http://www.php.net)
- [15] **Anonim**. 2002. T ü r k i y e P H P K u l l a n ı c ı l a r ı G r u b u R e s m i W e b S i t e s i ,
[http:// www.mysql.com](http://www.mysql.com)
- [16] **Anonim**. 2002. T ü r k i y e L i n u x K u l l a n ı c ı l a r ı D e r n e ğ i R e s m i W e b S i t e s i ,
[http:// www.linux.org.tr](http://www.linux.org.tr)

ÖZGEÇMİŞ



1978 yılında Anıyşa' da doğdu. İlkokul u Ordu Güzel ordu İlkokul unda, ortaokul u İspir Lisesi' nde (Erzurum) ta n a n l a d ı k t a n s o n r a , l i s e e ğ i t i m i n e A n k a r a A n a d o l u T a p u v e K a d a s t r o M e s l e k L i s e s i ' n d e y a t ı l ı o l a r a k d e v a m e t t i . 1996 y ı l ı n d a l i s e e ğ i t i m i n i d e t a n a n l a d ı v e a y n ı y ı l İ T Ü İ n Ő a a t F a k ũ l t e s i J e o d e z i v e F o t o g r a m e t r i M i h e n d i s l i ğ i B ũ l ũ m ũ n d e l i s a n s e ğ i t i m i n e b a Ő l a d ı . 1997 y ı l ı n d a İ T Ü F e n E d e b i y a t F a k ũ l t e s i M a t e m a t i k M i h e n d i s l i ğ i B ũ l ũ m ũ i l e c i f t l i s a n s (C A P) p r o g r a m n a v e 1998 y ı l ı n d a i s e İ T Ü İ n Ő a a t F a k ũ l t e s i B l g i İ Ő l e m M e r k e z i ' n d e Ő ğ r e n c i a s i s t a n o l a r a k c a l ı Ő n a y a b a Ő l a d ı . 2000 y ı l ı n d a J e o d e z i v e F o t o g r a m e t r i M i h e n d i s l i ğ i B ũ l ũ m ũ n d e k i l i s a n s e ğ i t i m i n i t a n a n l a y a n Ő z g ũ r Ő Z A S L A N , a y n ı y ı l İ T Ü F e n B l i m h e r i E n s t i t ũ s ũ J e o d e z i v e F o t o g r a m e t r i M i h e n d i s l i ğ i p r o g r a m n d a y ũ k s e k l i s a n s e ğ i t i m i n e b a Ő l a d ı . 2001 y ı l ı Ő u b a t a y ı n d a İ T Ü İ n Ő a a t F a k ũ l t e s i J e o d e z i A n a b i l i m D a l ı a r a Ő t ı r m a g ũ r e v l i s i v e İ n Ő a a t F a k ũ l t e s i B l g i İ Ő l e m M e r k e z i (i f b i m) s o r u m l u y a r d ı m c ı s ı v e s i s t e m y ũ n e t i c i s i o l a r a k a t a n d ı . 2003 y ı l ı Ő u b a t a y ı n d a M a t e m a t i k M i h e n d i s l i ğ i B ũ l ũ m ũ n d e k i e ğ i t i m i n i d e t a n a n l a y a r a k M a t e m a t i k M i h e n d i s i ũ n v a n ı n ı a l a n Ő z g ũ r Ő z a s l a n , h a l e n J e o d e z i A n a b i l i m D a l ı v e İ n Ő a a t F a k ũ l t e s i B l g i İ Ő l e m M e r k e z i ' n d e k i g ũ r e v l e r i n e d e v a m e t m e k t e d i r .

İ T ũ İ n Ő a a t F a k ũ l t e s i B l g i İ Ő l e m M e r k e z i ' n d e k i g ũ r e v l e r i s ı r a s ı n d a s i s t e m y ũ n e t i m i , w e b t a s a r ı m v e r i t a b a n ı t a s a r ı m v e y ũ n e t i m i , a ğ y ũ n e t i m i k o n u l a r ı n d a s t a j v e c a l ı Ő n a l a r y a p m ı Ő t ı r . J e o d e z i v e F o t o g r a m e t r i M i h e n d i s l i ğ i l i s a n s b i t i r m e c a l ı Ő n a s ı “ B l g i s a y a r O r t a m n d a 3 B o y u t l u Ő e h i r M ũ d e l l e r i n i n O u Ő t u r u l n a s ı ” , M a t e m a t i k M i h e n d i s l i ğ i B ũ l ũ m ũ n d e k i b i t i r m e c a l ı Ő n a s ı “ P H P – M y S Q L K u l l a n a r a k W e b ũ z e r i n d e n K ũ t ũ p h a n e V e r i t a b a n ı O o n a s y o n u ” ũ z e r i n e d i r . C a l ı Ő n a l a r ı n a j e o d e z i k v e r i t a b a n l a r ı v e w e b u y g u l a n a l a r ı k o n u s u n d a d e v a m e t m e k t e d i r . H a r i t a v e K a d a s t r o M i h e n d i s l e r i O d a s ı İ s t a n b u l Ő u b e s i v e İ T ũ J e o d e z i v e F o t o g r a m e t r i K u l ũ b ũ ũ y e s i d i r .