



# Arkeolojik Buluntu Envanter Sistemi

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Gülbahar BOZDAĞ

**Y220240182**

Proje Danışmanı: Prof. Dr. Femin Yalçın KÜÇÜKBAYRAK

Haziran 2024

# Arkeolojik Envanter Buluntu Sistemi

## ÖZ

Arkeolojik Buluntu Envanteri Sistemi, arkeologların ve arařtırmacıların bulgularını kolayca kaydetmelerine, buluntu envanterinin güncel, kayıt altına alınmasına ve referans olarak erişilebilir hale getirilmesine olanak sağlar.

Ayrıca arařtırmacıların bulguları analiz etmeleri ve arkeolojik bozulmaları daha iyi anlamaları için araçlar sağlar.

Bu sistem arkeologların buluntu ve kazılarını korumak, sürdürmek ve analiz etmek için kullandıkları bir veri tabanı sistemidir.

Veri tabanı içerisinde; Bulma verilerinin tutulduğu tablolar vardır.

Kullanıcıların veri tabanına erişmesi için bir Kullanıcı Arayüzü bulunmaktadır. Bu sistem web tabanlı bir uygulamadır.

Raporlama ve Analiz Aracı özellikleriyle veriler analiz edilebilir, raporlanabilir ve görselleştirilebilir.

Ayrıca güvenli bir veri tabanı sistemi için Güvenlik adı altında erişim kontrolünün sağlandığı bir sistem de bulunmaktadır.

**Anahtar Sözcükler:** Veri Tabanı, Create Table, Veri Seti, Closure Seti, Buluntu, Envanter, Diyagram, Relation Seçme, Nitelik, Sql

# Archaeological Inventory Find System

## Abstract

The Archaeological Find Inventory System enables archaeologists and researchers to easily record their finds, to keep the find inventory updated, recorded and recorded, and to make it accessible for reference.

It also provides tools for researchers to analyze finds and better understand their archaeological deterioration.

This system is a database system used by archaeologists to preserve, maintain and analyze their finds and excavations.

Within the database; There are tables where find data are kept.

There is a User Interface for users to access the database. This distribution is a web-based application.

It includes Find Entry Forms, which are forms where archaeologists and researchers can enter finds and can also be changed and edited.

With the Reporting and Analysis Tool features, data can be analyzed, reported and visualized.

In addition, for a secure database system, there is also a system in which access control is provided under the name of Security.

**Keywords:** Database, Create Table, Data Set, Closure Set, Find, Inventory, Diagram, Relation Select, Attribute, Sql

# Şekiller Listesi

Şekil 2.1	Diyagram Çizimi .....	4
Şekil 2.2	Diyagram Çizimi .....	6

# Tablolar Listesi

Tablo 3.1	Closure Set'i.....	9
Tablo 3.2	Closure Set'in Tüm Attribute'ları İçermesi .....	10

# İçindekiler

Öz .....	i
Abstract .....	ii
Şekiller Listesi .....	iii
Tablolar Listesi .....	iv
<b>1 Bölüm 1 .....</b>	<b>1</b>
1.1 Giriş .....	1
<b>2 Bölüm 2 .....</b>	<b>4</b>
2.1 Diyagram Çizimi ve Açıklaması .....	4
<b>3 Bölüm 3 .....</b>	<b>8</b>
3.1 Relation Seçme Closure Set Bulma .....	8
3.2 Set'in Tüm Attribute'ları İçermesi .....	10
<b>4 Bölüm 4 .....</b>	<b>11</b>
4.1 Create Table .....	11
4.2 Tablo Açıklaması .....	14
<b>5 Bölüm 5 .....</b>	<b>17</b>
5.1 Arkeolojik Buluntu Envanter Sistemi Gereksinim Listesi .....	17
5.2 Fonksiyonel Gereksinimler .....	18
5.3 Fonksiyonel Olmayan Gereksinimler .....	19
<b>6 Bölüm 6 .....</b>	<b>21</b>
6.1 Veri Seti Örneği .....	21
6.2 Örneğin Açıklanması .....	22
<b>7 Kaynaklar .....</b>	<b>25</b>

# Bölüm 1

## Giriş

"Veri tabanı tasarımı, bir veri modelinin geliştirilmesi sürecidir ve bu model, sistemdeki veri varlıkları ve bu varlıklar arasındaki ilişkileri temsil eder" (Silberschatz, Korth, & Sudarshan, 2011, s. 42).

Bu çalışmada kullanılan veri tabanı tasarımı süreci, veri varlıkları ve ilişkilerinin belirlenmesi üzerine kuruludur. Bu, Silberschatz, Korth ve Sudarshan (2011) tarafından açıklandığı gibi, veri modelinin oluşturulmasının temel bir aşamasıdır.

"SQL, veri tabanı şemasının tanımlanması, verilerin depolanması, güncellenmesi ve sorgulanması için kullanılan standart bir dildir" (Elmasri & Navathe, 2010, s. 134).

Bu çalışmada SQL, veri tabanı şemasının oluşturulması ve yönetimi için kullanılmıştır. Elmasri ve Navathe'nin (2010) belirttiği gibi, SQL veri tabanı işlemlerini standartlaştıran bir dildir.

"Veri tabanlarındaki güvenlik açıklarının minimize edilmesi için verilerin yetkilendirilmiş kullanıcılar dışında erişime kapalı olması gerekmektedir" (Kimball & Ross, 2013, s. 212).

Veri güvenliği hakkında yazarken bu alıntıyı kullanabilirsiniz: Veri güvenliği, bu çalışmanın önemli bir bileşenidir ve Kimball ve Ross'un (2013) belirttiği gibi, verilerin yetkisiz erişime karşı korunması gereklidir.

"Veri madenciliği, büyük veri setlerinden anlamlı kalıpların ve ilişkilerin keşfedilmesi işlemidir" (Han, Kamber, & Pei, 2011, s. 28).

Veri madenciliği teknikleri, büyük veri setlerinden değerli bilgilerin çıkarılması için kullanılmıştır. Han, Kamber ve Pei'nin (2011) belirttiği gibi, bu süreç verilerin anlamlı kalıplar ve ilişkiler üretmesi açısından kritik öneme sahiptir.

Arkeolojik Buluntu Envanter Sistemi, arkeolog ve arařtırmacıların buluntu verilerini kolayca kaydedebilmesini, buluntu envanterinin güncel tutulmasını, verilerin korunmasını ve gelecekteki arařtırmalara referans olacak şekilde erişilebilir olmasını sağlamaktadır.

Bu sistem, arkeologların buluntu ve kazı verilerini saklamak, yönetmek ve analiz etmek için kullanılan bir veri tabanı sistemidir.

Kullanıcıların veri tabanına erişimini mümkün kılmak adına bir Kullanıcı Arayüzü bulunmaktadır. Bu arayüz web tabanlı bir uygulamadır.

Arkeolog ve arařtırmacıların buluntu verilerini girebileceđi ve aynı zamanda da deđiřtirip, düzenleyebileceđi formlar olarak da Buluntu Giriř Formları içerir.

Raporlama ve Analiz Aracı özelliđi ile veriler analiz edilip, raporlanıp, görselleřtirilmesi yapılabilmektedir.



Veri tabanı içerisinde; buluntu verilerinin tutulduğu tablolar mevcuttur.

Projemi gerçekleştirirken; ER (varlık-ilişki) diyagramı çizdim. Diyagram:

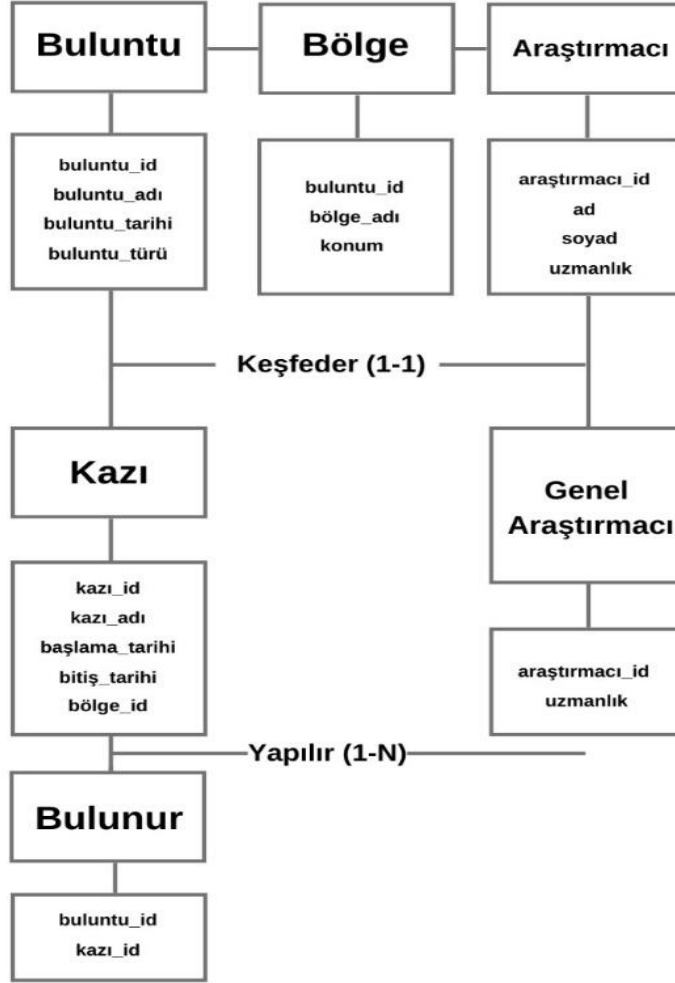
- 4 tane varlık (entity)
- En az 1 tane ilişki (1-1 relationship),
- En az 1 tane many-to-one ilişki (1-to-n relationship)
- En az 1 tane many-to-many (n-to-n relationship)
- Weak entity ilişkisi kısıtlarını içermektedir.

ER modelini Relational (İlişkisel) modele dönüştürdüm. Bu modele uygun bir relation seçtim;

- Geçerli tüm FD'leri (fonksiyonel bağımlılık) belirledim.
- Bulduğum closure setinin relation içerisindeki tüm özellikleri (attribute) gösterdim.
- SQL DDL komutları da bulunmaktadır.
- Excel ortamında hazırlanan bir Veri Seti örneği bulunmaktadır.
- Arkeolojik Buluntu Envanter Sistemi Gereksinim Listesi bulunmaktadır.

## BÖLÜM 2

### Diyagram Çizimi ve Açıklaması:



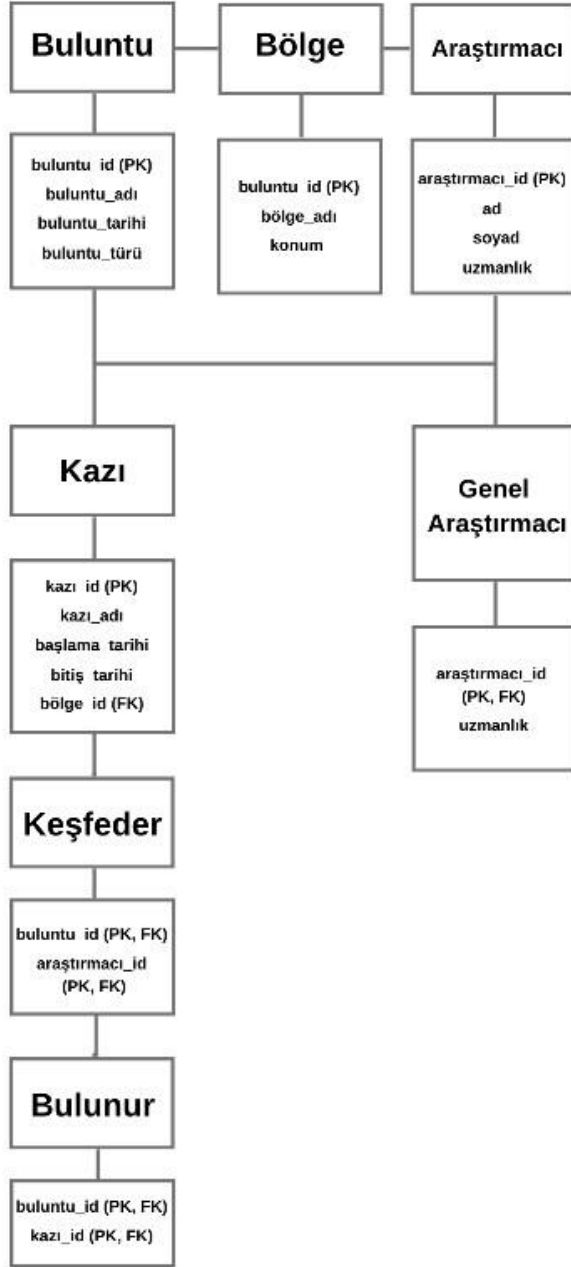
Şekil 2.1: Diyagram Çizimi

Keşfeder İlişkisi (1-1): Buluntu ve Araştırmacı arasında. Bir buluntu sadece bir araştırmacı tarafından keşfedilir.

Yapılır İlişkisi (1-N): Bölge ve Kazı arasında. Bir bölgede birçok kazı yapılabilir.

Bulunur İlişkisi (N-N): Buluntu ve Kazı arasında. Bir buluntu birden fazla kazıda bulunabilir ve bir kazıda birden fazla buluntu bulunabilir.

ISA İlişkisi: Araştırmacılar uzmanlık alanlarına göre genel ve uzman araştırmacılar olarak ayrılabilir.



Şekil 2.2: Diyagram Çizimi

Bulutnu, Bölge ve Arařtırmacı tabloları ana varlıkları temsil eder.

Kazı tablosu, bölge\_id yabancı anahtarını ile Bölge tablosuna baęlıdır.

Keşfeder tablosu, Buluntu ve Arařtırmacı tabloları arasında bir-bir ilişkiyi temsil eder.

Bulunur tablosu, Buluntu ve Kazı tabloları arasında çok-çok ilişkiyi temsil eder.

Genel\_Arařtırmacı ve Uzman\_Arařtırmacı tabloları, Arařtırmacı tablosunun alt tiplerini temsil eder.

Bu diyagram, ilişki modelinin görsel bir temsilidir ve ilişkileri açıkça gösterir.

## Bölüm 3

# RELATION SEÇME VE CLOSURE SET BULMA

Kazı(kazı\_id, kazı\_adi, başlama\_tarihi, bitiş\_tarihi, bölge\_id)

Step 1 ( $X \rightarrow x$ ) : { kazı\_id }

Step 2: { kazı\_id, kazı\_adi }

Step 3: { kazı\_id, kazı\_adi, başlama\_tarihi }

Step 4: { kazı\_id, kazı\_adi, başlama\_tarihi, bitiş\_tarihi }

Step 5: { kazı\_id, kazı\_adi, başlama\_tarihi, bitiş\_tarihi, bölge\_id }

Step 6: { kazı\_id, kazı\_adi, başlama\_tarihi, bitiş\_tarihi, bölge\_id, bölge\_adi }

Step 7: { kazı\_id, kazı\_adi, başlama\_tarihi, bitiş\_tarihi, bölge\_id, bölge\_adi,  
konum }

F	All Attributes
kazi_id -> kazi_adı	kazi_id
kazi_id -> başlama_tarihi	kazi_adı
kazi_id -> bitiş_tarihi	başlama_tarihi
kazi_id -> bölge_id	bitiş_tarihi
bölge_id -> bölge_adı	bölge_id
bölge_id -> konum	bölge_adı konum

Tablo 3.1: Closure Seti

## Closure Set'in Tüm Attribute'ları İçermesi:

Fonksiyonel Bağımlılıklar	Closure Set (X+)	Açıklama
kazı_id -> kazı_adı	{ kazı_id }	Başlangıç (X+=x)
	{ kazı_id, kazı_adı }	FD1'in uygulanması
kazı_id -> başlama_tarihi	{ kazı_id, kazı_adı, başlama_tarihi }	FD2'nin uygulanması
kazı_id -> bitiş_tarihi	{ kazı_id, kazı_adı, başlama_tarihi, bitiş_tarihi }	FD3'ün uygulanması
kazı_id -> bölge_id	{ kazı_id, kazı_adı, başlama_tarihi, bitiş_tarihi, bölge_id }	FD4'ün uygulanması
bölge_id -> bölge_adı	{ kazı_id, kazı_adı, başlama_tarihi, bitiş_tarihi, bölge_id, bölge_adı }	FD5'in uygulanması
bölge_id -> konum	{ kazı_id, kazı_adı, başlama_tarihi, bitiş_tarihi, bölge_id, bölge_adı, konum }	FD6'nın uygulanması

Tablo 3.2: Closure Set'in Tüm Attribute'ları İçermesi



# BÖLÜM 4

## CREATE TABLE

```
CREATE TABLE Buluntu (  
  
    buluntu_id INT PRIMARY KEY,  
  
    buluntu_adi VARCHAR(255),  
  
    buluntu_tarihi DATE,  
  
    buluntu_turu VARCHAR(255)  
  
);
```

```
CREATE TABLE Bölge (  
  
    bölge_id INT PRIMARY KEY,  
  
    bölge_adi VARCHAR(255),  
  
    konum VARCHAR(255)  
  
);
```

```
CREATE TABLE Araştırmacı (  
  
    araştırmacı_id INT PRIMARY KEY,  
  
    ad VARCHAR(255),  
  
    soyad VARCHAR(255),  
  
    uzmanlık VARCHAR(255)  
  
);
```

```

CREATE TABLE Genel_Arařtırmacı (
    arařtırmacı_id INT PRIMARY KEY,
    uzmanlık VARCHAR(255),
    FOREIGN KEY (arařtırmacı_id) REFERENCES
Arařtırmacı(arařtırmacı_id)
);

```

```

CREATE TABLE Uzman_Arařtırmacı (
    arařtırmacı_id INT PRIMARY KEY,
    uzmanlık_alani VARCHAR(255),
    FOREIGN KEY (arařtırmacı_id) REFERENCES
Arařtırmacı(arařtırmacı_id)
);

```

```

CREATE TABLE Kazı (
    kazı_id INT PRIMARY KEY,
    kazı_adi VARCHAR(255),
    başlama_tarihi DATE,
    bitiş_tarihi DATE,
    bölge_id INT,
    FOREIGN KEY (bölge_id) REFERENCES Bölge(bölge_id)
);

```

```

CREATE TABLE Keşfeder (
    buluntu_id INT,

```

```
arařtırmacı_id INT,  
  
PRIMARY KEY (buluntu_id, arařtırmacı_id),  
  
FOREIGN KEY (buluntu_id) REFERENCES Buluntu(buluntu_id),  
  
FOREIGN KEY (arařtırmacı_id) REFERENCES  
Arařtırmacı(arařtırmacı_id)  
  
);
```

```
CREATE TABLE Bulunur (  
  
buluntu_id INT,  
  
kazı_id INT,  
  
PRIMARY KEY (buluntu_id, kazı_id),  
  
FOREIGN KEY (buluntu_id) REFERENCES Buluntu(buluntu_id),  
  
FOREIGN KEY (kazı_id) REFERENCES Kazı(kazı_id)  
  
);
```

## Tablo Açıklaması

### Buluntu Tablosu:

buluntu\_id: Buluntunun benzersiz kimliği (Primary Key).

buluntu\_adi: Buluntunun adı (örneğin, bir fosil ya da seramik parçası).

buluntu\_tarihi: Buluntunun bulunduğu tarih.

buluntu\_turu: Buluntunun türü (örneğin, taş, metal, seramik).

### Bölge Tablosu:

bölge\_id: Bölgenin benzersiz kimliği (Primary Key).

bölge\_adi: Bölgenin adı (örneğin, "Troy").

konum: Bölgenin coğrafi konumu (örneğin, "Türkiye").

### Araştırmacı Tablosu:

araştırmacı\_id: Araştırmacının benzersiz kimliği (Primary Key).

ad: Araştırmacının adı.

soyad: Araştırmacının soyadı.

uzmanlık: Araştırmacının uzmanlık alanı.

Genel Arařtırmacı Tablosu:

arařtırmacı\_id: Arařtırmacının benzersiz kimlięi (Primary Key ve Foreign Key olarak Arařtırmacı tablosundan alınır).

uzmanlık: Genel arařtırmacının uzmanlık alanı.

Uzman Arařtırmacı Tablosu:

arařtırmacı\_id: Arařtırmacının benzersiz kimlięi (Primary Key ve Foreign Key olarak Arařtırmacı tablosundan alınır).

uzmanlık\_alani: Uzman arařtırmacının uzmanlık alanı (bu tablodaki bilgi Arařtırmacı tablosunda da bulunur).

Kazı Tablosu:

kazı\_id: Kazının benzersiz kimlięi (Primary Key).

kazı\_adi: Kazının adı.

bařlama\_tarihi: Kazının bařlangıç tarihi.

bitiř\_tarihi: Kazının bitiř tarihi.

bölge\_id: Kazının yapıldığı bölgenin kimlięi (Foreign Key olarak Bölge tablosundan alınır).

Keřfeder Tablosu:

buluntu\_id: Buluntunun kimlięi (Foreign Key olarak Buluntu tablosundan alınır).

arařtırmacı\_id: Arařtırmacının kimlięi (Foreign Key olarak Arařtırmacı tablosundan alınır).

Bu tablo, bir buluntunun hangi arařtırmacı tarafından keřfedildięini belirtir (many-to-many iliřkisi).

Bulunur Tablosu:

buluntu\_id: Buluntunun kimliđi (Foreign Key olarak Buluntu tablosundan alınır).

kazı\_id: Kazının kimliđi (Foreign Key olarak Kazı tablosundan alınır).

Bu tablo, bir buluntunun hangi kazıda bulunduđunu belirtir (many-to-many iliřkisi).

Genel olarak;

Primary Key: Her tablonun birincil anahtarları, tablodaki her bir kaydın benzersiz řekilde tanımlanmasını sađlar.

Foreign Key: Diđer tablolardan referanslar içerir ve iliřkileri tanımlar.

Many-to-Many İliřkiler: Keřfeder ve Bulunur tabloları, oktan ođa iliřkileri ifade eder, yani bir buluntu birden fazla kazıda bulunabilir ve bir kazı birok buluntu içerebilir.

## Bölüm 5

# Arkeolojik Buluntu Envanter Sistemi Gereksinim Listesi

Gereksinim Gereksinim Tipi Yazar Arkeolojistler, kullanıcı adları ve şifreleriyle sisteme giriş yapabilmelidir. Fonksiyonel İnsan Yeni buluntu, keşif veya kazılar, kayıt formu doldurularak sisteme eklenebilmelidir. Fonksiyonel İnsan Sisteme kayıt olurken belirtilen şifrelerin güçlü olup olmadığını kontrol eden bir mekanizma bulunmalıdır. Fonksiyonel İnsan Arkeolojistler, hatalı giriş yaptıklarında uygun hata mesajları almalı ve tekrar giriş yapmaları istenmelidir. Fonksiyonel İnsan Buluntuların tipi, keşif yeri, tarih ve diğer detaylar sisteme kaydedilmelidir. Fonksiyonel ChatGPT Buluntu, keşif veya kazılar, onaylandıktan sonra sisteme kaydedilmelidir. Fonksiyonel ChatGPT Buluntular, buldukları yerin coğrafi koordinatlarına göre sisteme kaydedilmelidir. Fonksiyonel ChatGPT Buluntu verileri, arkeolojik araştırmalarda kullanılmak üzere analiz edilmelidir. Fonksiyonel ChatGPT Buluntu verileri, korunması gereken türdeki buluntular için güvenlik protokolleriyle korunmalıdır. Fonksiyonel olmayan İnsan Sistem, yüksek performansta ve kesintisiz bir şekilde çalışmalıdır. Fonksiyonel olmayan İnsan Sistem, günlük kullanıcı etkileşimlerini izlemeli ve hata durumlarını kaydetmelidir. Fonksiyonel olmayan ChatGPT Sistem, düzenli olarak yedeklenmeli ve veriler güvenli bir şekilde saklanmalıdır. Fonksiyonel olmayan ChatGPT Sistem, kullanıcıların erişilebilirlik ihtiyaçlarını karşılamak için uygun özellikler sunmalıdır. Fonksiyonel olmayan ChatGPT Sistem, farklı senaryolarda test edilebilir ve hatalar hızlıca düzeltilebilir olmalıdır. Fonksiyonel olmayan ChatGPT

Arkeolojik Buluntu Envanter Sistemi'nin veri tabanı tasarımıyla doğrudan ilişkilidir çünkü veri tabanı tasarımı, bu gereksinimleri karşılayacak şekilde yapılandırılmalıdır.

### Fonksiyonel Gereksinimler:

Arkeologlar, kullanıcı adları ve şifreleriyle sisteme giriş yaparlar:

Bu gereksinim, kullanıcı bilgilerini (kullanıcı adı, şifre vb.) depolayacak bir kullanıcı tablosuna ihtiyaç duyulduğunu gösterir. Bu tablo, kullanıcı kimlik doğrulamasını destekler.

Yeni buluntu, keşif veya kazılar, kayıt formu doldurularak sisteme eklenir:

Bu gereksinim, buluntular, keşifler ve kazılarla ilgili bilgileri depolayan tabloların tasarlanmasını gerektirir. Bu tablolar, veri giriş formundan gelen bilgileri tutar.

Sisteme kayıt olurken belirtilen şifrelerin güçlü olup olmadığını kontrol eden bir mekanizma bulunur:

Bu gereksinim, kullanıcı bilgileri tablosundaki şifre alanının uygun şekilde şifrelenmiş ve güvenli bir yapıda olmasını gerektirir. Ayrıca, şifre karmaşıklığını kontrol eden iş kurallarını destekleyen bir yapı gerektirir.

Arkeologlar, hatalı giriş yaptıklarında uygun hata mesajları alırlar ve tekrar giriş yapmaları istenir:

Bu gereksinim, kullanıcı tablosunda başarısız giriş denemelerinin kaydedilmesi veya hata mesajları için gerekli bilgilerin tutulmasını gerektirir.

Buluntuların tipi, keşif yeri, tarih ve diğer detaylar sisteme kaydedilir:

Bu gereksinim, buluntularla ilgili ayrıntılı bilgileri depolayacak bir tabloya ihtiyaç olduğunu belirtir. Bu tablo, buluntunun tipi, keşif yeri, tarihi gibi bilgileri içerir.



Buluntu, keşif veya kazılar, onaylandıktan sonra sisteme kaydedilir:

Bu gereksinim, veri onay sürecini destekleyen bir yapı gerektirir. Örneğin, bir "onay durumu" alanı ile veri onaylanana kadar sisteme tam olarak kaydedilmemesi sağlanabilir.

Buluntular, buldukları yerin coğrafi koordinatlarına göre sisteme kaydedilir:

Bu gereksinim, buluntuların konum verilerini depolayacak alanlara ihtiyaç duyulduğunu gösterir. Konum bilgisi, coğrafi koordinatları içeren veri tiplerini destekleyen bir tasarım gerektirir.

Buluntu verileri, arkeolojik araştırmalarda kullanılmak üzere analiz edilir:

Bu gereksinim, verilerin analiz edilebilmesi için uygun şekilde yapılandırılmış bir veri tabanı tasarımı gerektirir. Veri normalizasyonu ve ilişkisel veri tabanı tasarımı bu tür analizlerin yapılmasını kolaylaştırır.

### Fonksiyonel Olmayan Gereksinimler:

Buluntu verileri, korunması gereken türdeki buluntular için güvenlik protokolleriyle korunmalıdır:

Bu gereksinim, veri tabanında hassas verilerin güvenliğini sağlamak için uygun güvenlik önlemlerinin alınmasını gerektirir. Şifreleme, erişim kontrolleri ve veri bütünlüğü gibi güvenlik önlemleri tasarımın bir parçası olmalıdır.

Sistem, yüksek performansta ve kesintisiz bir şekilde çalışmalıdır:

Bu gereksinim, veri tabanının performansını optimize edecek şekilde tasarlanmasını gerektirir. İndeksleme, sorgu optimizasyonu ve veri tabanı ölçeklendirme stratejileri bu performansı sağlar.

Sistem, günlük kullanıcı etkileşimlerini izler ve hata durumlarını kaydeder:

Bu gereksinim, kullanıcı aktivitelerini izlemek için bir günlük kayıt sistemi ve hata izleme mekanizmasının veri tabanı içinde yapılandırılmasını gerektirir.

Sistem, düzenli olarak yedekler ve veriler güvenli bir şekilde saklanır:

Bu gereksinim, veri tabanının yedeklenme ve veri kurtarma stratejilerini içermesini gerektirir. Yedekleme planları, yedeklerin güvenli bir şekilde saklanması da kapsar.

Sistem, kullanıcıların erişilebilirlik ihtiyaçlarını karşılamak için uygun özellikler sunar:

Bu gereksinim, kullanıcı arayüzü ve veri erişimi ile ilgili gereksinimlerin desteklenmesi için veri tabanının uygun şekilde yapılandırılmasını gerektirir.

Sistem, farklı senaryolarda test edilebilir ve hatalar hızlıca düzeltilebilir:

Bu gereksinim, veri tabanının esnek ve test edilebilir bir yapıda olmasını gerektirir. Bu, veri tabanı tasarımında test verileri kullanarak farklı senaryoların simüle edilmesini içerir.

Bu gereksinimler, veri tabanı tasarımının sadece verilerin depolanması değil, aynı zamanda bu verilerin güvenliği, erişimi, analizi ve sistemin genel performansı açısından da dikkate alınması gereken unsurlar olduğunu gösterir.

# Bölüm 6

## VERİ SETİ ÖRNEĞİ

Gereksinim Gereksinim Tipi Yazar

Arkeolojistler, kullanıcı adları ve şifreleriyle sisteme giriş yapabilmelidir.  
Fonksiyonel İnsan

Yeni buluntu, keşif veya kazılar, kayıt formu doldurularak sisteme eklenebilmelidir.  
Fonksiyonel İnsan

Sisteme kayıt olurken belirtilen şifrelerin güçlü olup olmadığını kontrol eden bir mekanizma bulunmalıdır. Fonksiyonel İnsan

Arkeolojistler, hatalı giriş yaptıklarında uygun hata mesajları almalı ve tekrar giriş yapmaları istenmelidir. Fonksiyonel İnsan

Buluntuların tipi, keşif yeri, tarih ve diğer detaylar sisteme kaydedilmelidir.  
Fonksiyonel ChatGPT

Buluntu, keşif veya kazılar, onaylandıktan sonra sisteme kaydedilmelidir.  
Fonksiyonel ChatGPT

Buluntular, buldukları yerin coğrafi koordinatlarına göre sisteme kaydedilmelidir.  
Fonksiyonel ChatGPT

Buluntu verileri, arkeolojik araştırmalarda kullanılmak üzere analiz edilmelidir.  
Fonksiyonel ChatGPT

Buluntu verileri, korunması gereken türdeki buluntular için güvenlik protokolleriyle korunmalıdır. Fonksiyonel olmayan İnsan

Sistem, yüksek performansta ve kesintisiz bir şekilde çalışmalıdır. Fonksiyonel olmayan İnsan

Sistem, günlük kullanıcı etkileşimlerini izlemeli ve hata durumlarını kaydetmelidir.

Fonksiyonel olmayan ChatGPT

Sistem, düzenli olarak yedeklenmeli ve veriler güvenli bir şekilde saklanmalıdır.

Fonksiyonel olmayan ChatGPT

Sistem, kullanıcıların erişilebilirlik ihtiyaçlarını karşılamak için uygun özellikler sunmalıdır. Fonksiyonel olmayan ChatGPT

Sistem, farklı senaryolarda test edilebilir ve hatalar hızlıca düzeltilebilir olmalıdır.

Fonksiyonel olmayan ChatGPT

### Örneğin Açıklanması:

Veri seti örneği, veri tabanı tasarımının nasıl işlediğini gösteren pratik bir uygulamadır. Bu veri seti, veri tabanı tasarımının gereksinimlerini karşılayacak şekilde nasıl yapılandırıldığını ve gerçek dünyadaki verilerin bu tasarımda nasıl saklanacağını, işleneceğini ve kullanılacağını anlamamıza yardımcı olur.

#### Veri Yapısını Belirlemek:

Veri seti, hangi bilgilerin hangi tablolarda saklanacağını belirler. Örneğin, "Bulutnu" tablosunda buluntulara ilişkin veriler, "Bölge" tablosunda kazı bölgelerine ilişkin veriler yer alır. Bu yapı, verilerin doğru tablolara eşleşmesini ve doğru ilişkilerin kurulmasını sağlar.

#### 2. İlişkisel Modelin Uygulanması:

Veri seti, tablolar arasındaki ilişkilerin nasıl kurulduğunu gösterir. Örneğin, bir kazının birden fazla buluntuyla ilişkilendirildiği durumda "Bulunur" gibi ilişki tabloları oluşturulur. Bu tablolar, veri tabanı tasarımının ilişki modelini yansıtır ve veri setindeki bilgilerin nasıl bir arada tutulacağını gösterir.

### 3. Veri Doğruluğu ve Tutarlılığı:

Veri seti, veri doğruluğunu ve tutarlılığını sağlamak için tasarlanmış kuralları gösterir. Örneğin, "kazı\_id" gibi birincil anahtarların benzersiz olması, yabancı anahtarların uygun şekilde referans verilmesi ve verilerin tekrarlanmasını önlemek için normalizasyon ilkelerinin uygulanması gibi.

### 4. Fonksiyonel Gereksinimlerin Karşılanması:

Veri seti, fonksiyonel gereksinimlerin nasıl karşılandığını gösterir. Örneğin, arkeologların sisteme giriş yapması, yeni buluntuların eklenmesi veya kazıların kaydedilmesi gibi işlemler, bu veri setiyle doğrudan ilişkilidir. Veri tabanı tasarımı, bu gereksinimlerin karşılanmasını sağlar.

### 5. Analiz ve Raporlama İçin Altyapı:

Veri seti, arkeolojik bulguların analiz edilmesi ve raporlanması için gerekli olan verilerin nasıl depolanacağını gösterir. Bu, veri tabanının analiz ve raporlama işlevlerini destekleyecek şekilde tasarlandığını gösterir.

### 6. Veri Güvenliği ve Erişim Kontrolleri:

Veri seti, hangi verilerin kimler tarafından erişilebileceğini ve hangi güvenlik önlemlerinin alınması gerektiğini belirler. Örneğin, hassas bilgilerin (şifreler gibi) şifrelenmiş olarak saklanması ve yetkisiz erişimlerin engellenmesi gereksinimi veri tabanı tasarımıyla ilişkilidir.

### 7. Veri Tablosunun Kullanıcılarla Etkileşimi:

Veri seti, kullanıcıların veri tabanı ile nasıl etkileşimde bulunacağını gösterir. Örneğin, bir arkeolog sisteme giriş yaptığında, hangi verileri görebileceği ve hangi işlemleri yapabileceği bu veri seti ile belirlenir.

Bu veri seti, veri tabanının hangi bilgileri nasıl saklayacağını ve bu bilgilerin hangi fonksiyonları destekleyeceğini gösterir. Böylece, gereksinimlerin karşılanması, veri bütünlüğü, ilişkisel yapı ve veri güvenliği gibi kritik tasarım unsurları bu veri seti aracılığıyla uygulanır ve doğrulanır.

# Kaynaklar

Silberschatz, Abraham, Korth, F., Henry, Sudarshan, S. (2010). *Database System Concepts*(6. Baskı). McGraw Hill Yayınları.

Elmasri, Ramez, Navathe, Shamkant (2007). *Fundamentals of Database Systems* (7. Baskı). Pearson Yayınevi.

Ross, Margy, Kimball, Ralph (2013) *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3. Baskı). Wiley Yayınevi.

Kamber, Micheline, Han, Jiawei, (2011). *Data Mining: Concepts and Techniques* (*The Morgan Kaufmann Series in Data Management Systems*) (3.Baskı). Morgan Kaufmann Yayınevi.