



**YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BLM3041 VERİTABANI YÖNETİMİ DERSİ
PROJE ÖDEVİ**

PETSET – HAYVAN SAHİPLENME SİSTEMİ

20011006 - Abdullah Memiş Belikırık - memis.belikirik@std.yildiz.edu.tr

20011093 - Müdafer Kaymak - mudafer.kaymak@std.yildiz.edu.tr

20011052 - Ayşenaz Konan - aysenaz.konan@std.yildiz.edu.tr

20011061 - Mehmet Fatih Akkuş – fatih.akkus@std.yildiz.edu.tr

İSTANBUL, 2023 GÜZ

İÇİNDEKİLER

1. Giriş
2. Ekip Yapısı
3. Kullanım Senaryoları
 - 3.1. Kullanıcı Senaryoları
 - 3.2. Yönetici Senaryoları
4. Projede Kullanılan Veritabanına Ait ER Diyagramı
5. Veritabanındaki Tabloların Ekran Görüntüleri
 - 5.1. Users Tablosu
 - 5.2. Ad Tablosu
 - 5.3. Application Tablosu
 - 5.4. Orders Tablosu
 - 5.5. Products Tablosu
6. Projede Duyurusunda İstenilen Maddelerin Sağlandığına Dair Kod Blokları
 - 6.1. Veritabanındaki Tabloların SQL Kodları
 - 6.1.1. Users Tablosu
 - 6.1.2. Ad Tablosu
 - 6.1.3. Application Tablosu
 - 6.1.4. Orders Tablosu
 - 6.1.5. Products Tablosu
 - 6.2. Insert, Delete ve Update İşlemleri İçin Örnek Kodlar
 - 6.2.1. Insert İşlemi
 - 6.2.2. Delete İşlemi
 - 6.2.3. Update İşlemi
 - 6.3. Girilen Değere Göre Listeleme İşlemi İçin Örnek Kod
 - 6.4. View Olarak Tanımlanan Sorgunun SQL Kodu
 - 6.5. Except Kullanım Örneği
 - 6.6. Kullanılan SQL Fonksiyonları
 - 6.7. Tanımlanan ve Kullanılan SQL Triggerları

1. Giriş

PetSet Hayvan Sahiplenme Sistemi, hayvan severleri bir araya getirerek evsiz hayvanlara sevgi dolu bir yuva bulmayı amaçlayan bir uygulamadır. Projeyi geliştirme amacımız, hayvan sahiplenme sürecini daha şeffaf, etkili ve duyarlı hale getirerek, kullanıcıların kolayca sahiplenme ilanlarına başvurabilmelerini ve kendi evcil hayvanlarını da sahiplendirebilmelerini sağlamaktır.

Kullanıcılar, uygulama üzerinden belirledikleri hayvan türüne (kedi, köpek, kuş vb.) özel olarak listelenen sahiplenme ilanlarına kolayca erişebilirler. Uygulamamız, kullanıcıların başvurdukları ilanlara dair detaylı bilgileri görebilmelerini sağlamaktadır. Sistem, başvuranların hangi şehirlerden geldiğini listeler. Ayrıca, kullanıcılar kendi sahiplenme ilanlarını oluşturabilirler, bu sayede kendi evcil hayvanlarını da yeni bir yuva bulmalarına yardımcı olabilirler.

Sistem, kullanıcıların kişisel bilgilerini güncelleyebilecekleri ve başvurdukları ilanları takip edebilecekleri bir kullanıcı paneli sunar. Ayrıca, alışveriş modülü aracılığıyla çeşitli evcil hayvan ürünlerini satın alma imkanı da sunarak kullanıcılara geniş bir deneyim sunmaktadır. PetSet'in amacı, hayvanların sevgi dolu bir aile bulmalarını destekleyerek, toplumda daha bilinçli ve sorumlu bir hayvan sahiplenme kültürü oluşturmaktır.

2. Ekip Yapısı

- Proje yöneticisi olan Müdafer Kaymak, projenin başından sonuna kadar tüm aşamalarını planlayan, koordine eden ve yöneten kişidir. Proje ekibinin her bir üyesinin görev ve sorumluluklarını belirleyerek, işlerin uyumlu ve zamanında tamamlanmasını sağlamıştır. Ek olarak, proje hedeflerine ulaşmak için gerekli kaynakları organize etmiş ve bu kaynakların etkili kullanımını sağlamıştır.
- Sistem tasarımcısı olan Mehmet Fatih Akkuş, geliştirilen sistemin tasarımını yapmakla sorumlu olan kişidir. Bu kişi, projenin hedeflerini, ihtiyaçlarını ve gereksinimlerini analiz eden sistem analistinin ortaya koyduğu verileri kullanarak sistem bileşenlerini, veritabanı yapılarını, arayüzleri, veri akışını ve diğer detayları içeren bir sistem tasarım dokümanı hazırlamış, sistemi tasarlamış ve teknik çözümler üretmiştir. Tasarımın doğru bir şekilde uygulanmasını sağlamış ve geliştiriciler ile çalışarak, sistem tasarımının uygun bir şekilde kodlanmasını sağlamıştır.
- Arayüz geliştirici olan Ayşenaz Konan, kullanıcının sistemle etkileşim kurmak için kullandığı arayüzleri tasarlayan ve geliştiren kişidir. Kullanıcı arayüzü, bir yazılım uygulamasının kullanıcılar tarafından etkileşimde bulunulabilecek kısmıdır. Bu kişi, kullanıcı dostu, erişilebilir ve etkili bir arayüz tasarımı oluşturmak için proje gerekliliklerine uygun olarak çalışmıştır. Bunun yanı sıra, arayüzün doğru bir şekilde çalışması ve hata ayıklanması için yazılımın geri kalan kısmıyla iş birliği yapmıştır. Bu süreçte, çeşitli yazılım geliştirme araçları, programlama dilleri, grafik tasarım araçları ve diğer teknolojiler kullanmıştır.
- Yazılım geliştirici olan Abdullah Memiş Belikırık, yazılım tasarımından başlayarak, kodlama, test etme ve projenin sonlandırılması aşamalarına kadar bir dizi görevi yerine getiren kişidir. Bu kişi, tasarımcılarla ve proje yöneticileriyle yakın bir şekilde çalışmalar gerçekleştirerek; yazılımın geliştirme aşamalarından tasarımın oluşturulmasında, kodlamanın yapılmasında, test edilmesinde ve müşteriye sunulacak son ürünün dağıtımında yer almıştır. Ayrıca proje dokümantasyonunu da hazırlamıştır, böylece diğer ekip üyeleri projenin ilerleyişini anlamış ve sorunları hızlı bir şekilde çözebilmiştir.

3. Kullanım Senaryoları

3.1 Kullanıcı Senaryoları

Senaryo 1: Kullanıcı uygulamayı açar, login ekranından kaydol butonuna tıklar, bilgilerini girip uygulamaya kaydolur.

Senaryo 2: Kullanıcı uygulamayı açar, LOGIN ekranından kullanıcı adı ve şifresinin girip uygulamaya giriş yapar ve karşısına MY ADS sayfası çıkar.

Senaryo 3: Kullanıcı MY ADS sayfasından yeni ilan ekleyebilir; ilanlarını görüntüleyebilir, silebilir, düzenleyebilir.

Senaryo 4: Kullanıcı PRODUCTS sayfasından evcil hayvanlarla ilgili ürünleri görüntüleyebilir, sipariş verebilir.

Senaryo 5: Kullanıcı ADS sayfasından sahiplendirme ilanlarını görüntüleyebilir, bu ilanları seçtiği hayvan cinsine göre filtreleyebilir, listeden seçtiği ilana başvurabilir.

Senaryo 6: Kullanıcı PROFILE sayfasından kendisine ait bilgileri görüntüleyebilir, düzenleyebilir; başvurduğu sahiplendirme ilanlarını ve sipariş ettiği ürünleri görüntüleyebilir

3.2 Yönetici Senaryoları

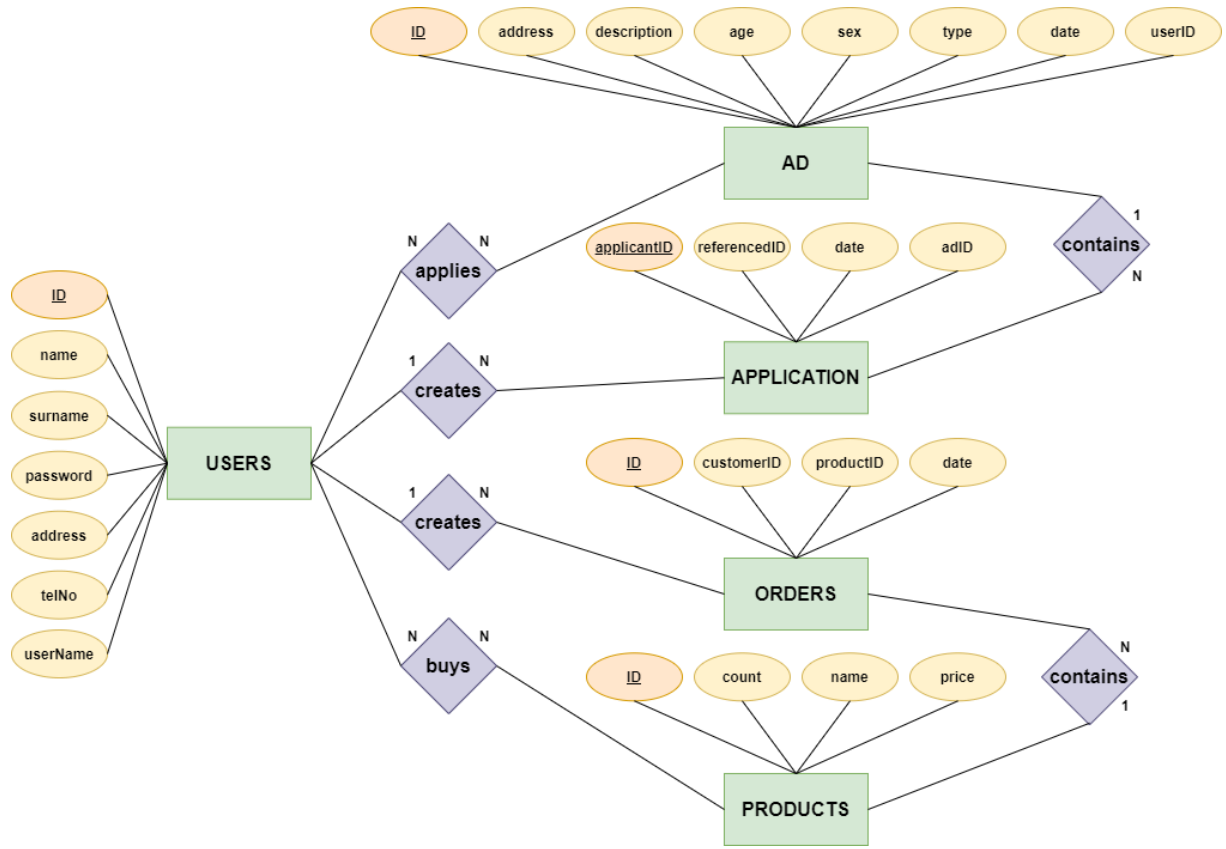
Senaryo 1: Yönetici PRODUCTS sayfasından satılık ürünleri görüntüleyebilir, silebilir, düzenleyebilir, yeni ürünler ekleyebilir.

Senaryo 2: Yönetici ADS sayfasından sahiplendirme ilanlarını görüntüleyebilir, silebilir.

Senaryo 3: Yönetici USERS sayfasından programa kayıtlı kullanıcıları görüntüleyebilir.

Senaryo 4: Yönetici APPLICATIONS AND ORDERS sayfasından sahiplendirme ilanlara yapılan başvuruları ve ürünler için oluşturulan siparişleri görüntüleyebilir.

4. Projede Kullanılan Veritabanına Ait ER Diyagramı



5. Veritabanındaki Tabloların Ekran Görüntüleri

5.1. Users Tablosu

	id [PK] integer	name character varying (255)	surname character varying (255)	password character varying (255)	address character varying (255)	telno character varying (15)	username character varying (50)
1	1	Fatih	Akkuş	kütüphane123	İzmir	+905553331111	coolwhp
2	2	Ayşenaz	Konan	fakülte123	Denizli	+905553332222	konannaz
3	3	Müdafer	Kaymak	yemekhane123	İstanbul	+905553333333	kaymakmudafer
4	4	Abdullah	Belikırık	ortabahçe123	İstanbul	+905553334444	belikirik123
5	5	Melih Tuna	İpek	bilgisayar123	Kırıkkale	+905553335555	melih_tuna78
6	6	Zehra Bengü	Emül	elektrik123	İzmir	+905553336666	zehra_emül
7	7	Büşra	Güral	biyomedikal123	Ankara	+905553337777	buşragüral97
8	8	Mert Tuna	Kurnaz	metalurji123	Ankara	+905553338888	flutter
9	9	Emir	Oğuz	mikro123	Antalya	+905553339999	heckon
10	10	Ömer	Aşkın	blockchain123	İstanbul	+905553330000	aşkın_ömer

5.2. Ad Tablosu

	id [PK] integer	address character varying (255)	description character varying (255)	age integer	sex character var	type character varying (5)	date date	userid integer
1	1	İzmir	3 kardeşten tekir olan sakın uysal bir kedir. Diğer kedilerle pek anlaşamaz. Kendisine yeni bir ev arıyor.	1	erkek	kedi	2024-01-10	1
2	2	İzmir	Sarmanlardan dişi olanı olan çok hareketli zıpr bir minik. Kendisine yeni bir ev arıyor.	1	erkek	kedi	2024-01-10	1
3	3	İzmir	Sarmanlardan erkek olanı. Çok tatlı bir bebek. Kendisine yeni bir ev arıyor.	1	dişi	kedi	2024-01-10	1
4	4	Ankara	Ağı kartı mevcut, tüm aşıları tamamlanmış Pofimizi sahiplendirmek istiyoruz. Son derece uysal, tuvalet alışkanlığı tam, kırıp dökmesi olmayan bir teki..	5	dişi	kedi	2024-01-02	7
5	5	Ankara	2 yaşında bir gozu alınmış asilari tam, oyle sevecen uysal oyuncu bir cocuk onu hic terksetmeyecek sevecen ailesi olmak ister misiniz?	2	dişi	kedi	2024-01-03	7
6	6	İstanbul	İlk aşıları yaptı. Tuvalet eğitimi almaya başladı. Sahiplenen kişi yavru köpek baktığı için bilinciyle hareket etmesini istiyorum.	1	dişi	köpek	2023-12-25	3
7	7	Antalya	Aşıları tam çok sevecen bir dostumuz. Sokakta bulundu tedavisi yapıldı.	2	dişi	köpek	2024-01-10	9
8	8	Kırıkkale	65 cm Seba marka salma kafes, akıllı yemlik ve Vege marka yuvallığı ile birlikte sadece şehir içinde teslim edilecek.	2	dişi	muhabbet kuşu	2024-01-10	5
9	9	Denizli	Yavrumuz sokaktan kurtarılmıştır. Veteriner kontrolleri yapıldı. Mikro chip takıldı. İç ve dış parazit uygulamaları yapıldı. Aşılarına başlandı.	2	erkek	köpek	2024-01-10	2
10	10	İzmir	mama ağı yal parası hiç bir ücret talep edilmeyecektir iyi aile olsun bakılın yeter	5	dişi	köpek	2024-01-10	6

5.3. Application Tablosu

	applicantid integer	referencedid integer	date date	adid integer
1	1	7	2024-01-13	4
2	1	5	2024-01-13	8
3	2	6	2024-01-13	10
4	2	1	2024-01-13	2
5	3	7	2024-01-13	4
6	3	5	2024-01-13	8
7	9	7	2024-01-13	5
8	9	7	2024-01-13	4
9	5	2	2024-01-13	9
10	5	9	2024-01-13	7

5.4. Orders Tablosu

	id [PK] integer	customerid integer	productid integer	date date
1	1	2	1	2024-01-13
2	2	2	2	2024-01-13
3	3	2	6	2024-01-13
4	4	3	2	2024-01-13
5	5	3	3	2024-01-13
6	6	3	8	2024-01-13
7	7	9	2	2024-01-13
8	8	9	4	2024-01-13
9	9	9	6	2024-01-13
10	10	5	5	2024-01-13
11	11	5	9	2024-01-13

5.5. Products Tablosu

	id [PK] integer	count integer	name character varying (255)	price integer
1	1	100	Kuru Kedi Maması	50
2	2	90	Kuru Köpek Maması	40
3	3	20	Yaş Kedi Maması	100
4	4	30	Yaş Köpek Maması	80
5	5	10	Kuş Kafesi	200
6	6	50	Kedi Kumu	90
7	7	20	Kedi Tasması	60
8	8	20	Köpek Tasması	70
9	9	75	Kuş Yemi	30
10	10	30	Köpek Tasması	70

6. Projede Duyurusunda İstenilen Maddelerin Sağlandığına Dair Kod Blokları

6.1. Veritabanındaki Tabloların SQL Kodları

6.1.1. Users Tablosu

```
54 CREATE SEQUENCE user_sequence START 1;
55
56 CREATE TABLE users (
57     ID INT DEFAULT nextval('user_sequence') PRIMARY KEY,
58     name VARCHAR(255),
59     surname VARCHAR(255),
60     password VARCHAR(255),
61     address VARCHAR(255),
62     telNo VARCHAR(15),
63     userName VARCHAR(50)
64 );
```

6.1.2. Ad Tablosu

```
91 CREATE SEQUENCE ad_sequence START 1;
92
93 CREATE TABLE ad (
94     ID INT DEFAULT nextval('ad_sequence') PRIMARY KEY,
95     address VARCHAR(255),
96     description VARCHAR(255),
97     age INT,
98     sex VARCHAR(10),
99     type VARCHAR(50),
100     date DATE,
101     userid INT
102 );
103
104 ALTER TABLE ad ADD FOREIGN KEY (userid) REFERENCES users (ID) ON DELETE CASCADE;
```

6.1.3. Application Tablosu

```
67 CREATE SEQUENCE prod_sequence START 1;
68
69 CREATE TABLE Application(
70     applicantID INT,
71     referencedID INT,
72     date DATE,
73     adID INT
74 );
75
76 ALTER TABLE APPLICATION ADD FOREIGN KEY (applicantID) REFERENCES users (ID) ON DELETE CASCADE;
77
78 ALTER TABLE APPLICATION ADD FOREIGN KEY (referencedID) REFERENCES users (ID) ON DELETE CASCADE;
```


6.1.4. Orders Tablosu

```
51 CREATE SEQUENCE orders_sequence START 1;
52
53 CREATE TABLE orders(
54     ID INT DEFAULT nextval('orders_sequence') PRIMARY KEY,
55     customerID int,
56     productID int,
57     date DATE
58 );
```

6.1.5. Products Tablosu

```
77 CREATE TABLE Products(
78     ID INT DEFAULT nextval('prod_sequence') PRIMARY KEY,
79     count INT,
80     name VARCHAR(255),
81     price INT,
82     CONSTRAINT price_bound check (price is null or (price >= 0))
83 );
```

6.2. Insert, Delete ve Update İşlemleri İçin Örnek Kodlar

6.2.1. Insert İşlemi

```
private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if (priceField.getText().isEmpty() || nameField.getText().isEmpty() || amountField.getText().isEmpty()) {
        JOptionPane.showMessageDialog(parentComponent:this, message:"Please enter all required values.");
    } else {
        try {
            String checkQuery = "SELECT COUNT(*) FROM products WHERE name = ?";
            PreparedStatement checkStatement = conn.prepareStatement(string: checkQuery);
            checkStatement.setString(i: 1, string: nameField.getText());

            ResultSet resultSet = checkStatement.executeQuery();
            resultSet.next();
            int cnt = resultSet.getInt(i: 1);

            if(cnt == 0){
                String insertQuery = "insert into Products(name, count, price) VALUES(?,?,?)";
                PreparedStatement insertStatement = (PreparedStatement) conn.prepareStatement(string: insertQuery);
                String countString = amountField.getText();
                int count = Integer.parseInt(s: countString);
                String priceString = priceField.getText();
                int price = Integer.parseInt(s: priceString);
                insertStatement.setString(i: 1, string: nameField.getText());
                insertStatement.setInt(i: 2, i1: count);
                insertStatement.setInt(i: 3, i1: price);
                insertStatement.executeUpdate();
                JOptionPane.showMessageDialog(parentComponent:this, message:"Product added.");
                displayProducts();
            }
            else{
                JOptionPane.showMessageDialog(parentComponent:this, message:"This product already exist.");
            }
        } catch (SQLException ex) {
            Logger.getLogger(name: AdminProducts.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
        }
    }
}
```

6.2.2. Delete İşlemi

```
private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        PreparedStatement deleteStatement = (PreparedStatement) conn.prepareStatement("delete from products where name = ?");

        int selectedRowIndex = productTable.getSelectedRow();
        if (selectedRowIndex != -1) {
            String productName = (String) productTable.getValueAt(row:selectedRowIndex, column: 1).toString();
            deleteStatement.setString(1, productName);
            deleteStatement.executeUpdate();
            JOptionPane.showMessageDialog(parentComponent:this, message:"Product deleted.");
            displayProducts();
        } else {
            JOptionPane.showMessageDialog(parentComponent:this, message:"You have to choose a product to delete.");
        }
        deleteStatement.close();
    } catch (SQLException ex) {
        Logger.getLogger(name: AdminProducts.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
    }
}
```

6.2.3. Update İşlemi

```
private void editButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String updateQuery = "Update products Set name = ? , count = ?, price = ? where name = ?";
        PreparedStatement updateStatement = (PreparedStatement) conn.prepareStatement(updateQuery);

        int selectedRowIndex = productTable.getSelectedRow();
        if (selectedRowIndex != -1) {
            String oldProductName = (String) productTable.getValueAt(row:selectedRowIndex, column: 0).toString();

            if (nameField.getText().isEmpty() || priceField.getText().isEmpty() || amountField.getText().isEmpty()) {
                JOptionPane.showMessageDialog(parentComponent:this, message:"Please enter all required fields.");
            } else {
                String counttxt = amountField.getText();
                int count = Integer.parseInt(s: counttxt);
                String pricetxt = priceField.getText();
                int price = Integer.parseInt(s: pricetxt);
                updateStatement.setString(1, nameField.getText());
                updateStatement.setInt(2, count);
                updateStatement.setInt(3, price);
                updateStatement.setString(4, oldProductName);
                updateStatement.executeUpdate();

                JOptionPane.showMessageDialog(parentComponent:this, "You updated product " + oldProductName);
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent:this, message:"You have to choose a product to update.");
        }
        updateStatement.close();
        displayProducts();
        displayOutOfStock();
    } catch (SQLException ex) {
        Logger.getLogger(name: AdminProducts.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
        JOptionPane.showMessageDialog(parentComponent:this, message:ex);
    }
}
```

6.3. Girilen Değere Göre Listeleme İşlemi İçin Örnek Kod

```
private void listele_btnActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String selectedType = (String) jComboBox1.getSelectedItem();  
        PreparedStatement preparedStatement = null;  
        String selectQuery = "SELECT * FROM ad WHERE type = ?";  
        preparedStatement = conn.prepareStatement(selectQuery);  
        preparedStatement.setString(1, selectedType);  
        ResultSet resultSet = preparedStatement.executeQuery();  
        ResultSetMetaData metaData = resultSet.getMetaData();  
        int columnCount = metaData.getColumnCount();  
        String[] columnNames = new String[columnCount];  
        for (int i = 1; i <= columnCount; i++) {  
            columnNames[i - 1] = metaData.getColumnName(i);  
        }  
  
        DefaultTableModel tableModel = new DefaultTableModel(columnNames, rowCount: 0);  
        Object[] row ;  
  
        while (resultSet.next()) {  
            row = new Object[columnCount];  
            for (int i = 1; i <= columnCount; i++) {  
                row[i - 1] = resultSet.getObject(i);  
            }  
            tableModel.addRow(rowData: row);  
        }  
  
        adTable.setModel(dataModel: tableModel);  
  
        preparedStatement.close();  
        resultSet.close();  
        // TODO add your handling code here:  
    } catch (SQLException ex) {  
        Logger.getLogger(Ads.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
    }  
}
```

6.4. View Olarak Tanımlanan Sorgunun SQL Kodu

```
107 CREATE OR REPLACE VIEW user_info_view AS  
108 SELECT  
109     userName,  
110     telNo,  
111     address  
112 FROM  
113     users;
```

6.5. Except Kullanım Örneği

```
void displayOutOfStock() throws SQLException{
    try {
        conn = DriverManager.getConnection(url:dbUrl, user:dbUsername, password:dbPassword);
    } catch (SQLException e) {
        System.out.println(x: e.getMessage());
    }

    Statement selectStatement = conn.createStatement();
    String selectQuery = "SELECT name,count,price FROM products EXCEPT SELECT name,count,price FROM products where count>1";
    ResultSet resultSet = selectStatement.executeQuery(string: selectQuery);

    ResultSetMetaData metaData = resultSet.getMetaData();
    int columnCount = metaData.getColumnCount();
    String[] columnNames = new String[columnCount];
    for (int i = 1; i <= columnCount; i++) {
        columnNames[i - 1] = metaData.getColumnName(i);
    }

    DefaultTableModel tableModel = new DefaultTableModel(columnNames, rowCount: 0);
    Object[] row ;

    while (resultSet.next()) {
        row = new Object[columnCount];
        for (int i = 1; i <= columnCount; i++) {
            row[i - 1] = resultSet.getObject(i);
        }
        tableModel.addRow(rowData: row);
    }

    oosTable.setModel(dataModel: tableModel);

    selectStatement.close();
    resultSet.close();
}
```

6.6. Aggregate Fonksiyon İçeren Sorgu Örneği

```
private void displayCount() throws SQLException {
    String selectIDQuery = "SELECT id FROM users WHERE username = ?";
    PreparedStatement selectIDStatement = conn.prepareStatement(string: selectIDQuery);
    selectIDStatement.setString(i: 1, string: Login.userName);
    ResultSet idset = selectIDStatement.executeQuery();

    if (idset.next()) {
        Profile.userid = idset.getInt(string: "id");
    }

    String selectQuery = "SELECT count(*) AS total FROM application GROUP BY applicantid HAVING applicantID = ?";
    PreparedStatement selectStatement = conn.prepareStatement(string: selectQuery);
    selectStatement.setInt(i: 1, i1: Profile.userid);
    System.out.println(x: Profile.userid);
    ResultSet resultSet = selectStatement.executeQuery();

    int count = 0;
    if (resultSet.next()) {
        count = resultSet.getInt(string: "total");
    }
    jLabel15.setText("Your " + String.valueOf(i: count) + " Application(s)");
}
```

6.6 Kullanılan SQL Fonksiyonları

```
81 CREATE OR REPLACE FUNCTION get_user_by_name(p_name VARCHAR)
82 RETURNS TABLE (
83     ID INT,
84     name VARCHAR(255),
85     surname VARCHAR(255),
86     password VARCHAR(255),
87     address VARCHAR(255),
88     telNo VARCHAR(15),
89     username VARCHAR(50)
90 ) AS $$
91 BEGIN
92     RETURN QUERY SELECT * FROM users WHERE users.username = p_name;
93 END;
94 $$ LANGUAGE plpgsql;
```

```
132 CREATE OR REPLACE FUNCTION get_all_products()
133 RETURNS TABLE (
134     ID INT,
135     count INT,
136     name VARCHAR(255),
137     price INT
138 ) AS $$
139 DECLARE
140     product_info Products%ROWTYPE;
141     product_cursor CURSOR FOR SELECT * FROM Products;
142 BEGIN
143     OPEN product_cursor;
144     LOOP
145         FETCH product_cursor INTO product_info;
146         EXIT WHEN NOT FOUND;
147
148         ID := product_info.ID;
149         count := product_info.count;
150         name := product_info.name;
151         price := product_info.price;
152
153         RETURN NEXT;
154     END LOOP;
155
156     CLOSE product_cursor;
157
158
159 END;
160 $$ LANGUAGE plpgsql;
```

6.7 Tanımlanan ve Kullanılan SQL Triggerları

```
65 CREATE OR REPLACE FUNCTION delete_application_after_ad_delete()
66 RETURNS TRIGGER AS $$
67 ▼ BEGIN
68     DELETE FROM Application
69     WHERE adID = OLD.ID;
70     RETURN OLD;
71 END;
72 $$ LANGUAGE plpgsql;
73
74 CREATE TRIGGER trigger_delete_application
75 AFTER DELETE ON AD
76 FOR EACH ROW
77 EXECUTE FUNCTION delete_application_after_ad_delete();
```

```
164 CREATE OR REPLACE FUNCTION check_price_threshold()
165 RETURNS TRIGGER AS $$
166 ▼ BEGIN
167 ▼     IF NEW.price > 1000 THEN
168         -- Burada yapılacak işlemleri ekleyebilirsiniz.
169         RAISE EXCEPTION 'Uyarı: Fiyat 1000 üzerine çıktı!';
170     END IF;
171     RETURN NEW;
172 END;
173 $$ LANGUAGE plpgsql;
174
175 CREATE TRIGGER trigger_check_price
176 BEFORE INSERT OR UPDATE ON products
177 FOR EACH ROW
178 EXECUTE FUNCTION check_price_threshold();
```