T.C.

SELÇUK ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

FİNAL PROJESİ

ÖĞRENCİNİN ADI SOYADI: İSMAİL APAN

ÖĞRENCİ NUMARASI: 213311118

DERS ADI: VERİ TABANI VE YÖNETİM SİSTEMLERİ

PROJE KONUSU: İŞ ARAMA VE İŞVEREN BULUŞMA PLATFORMU

MAYIS-2025
KONYA

İÇİNDEKİLER

KAVRAMSAL MODEL

Bu projede, iş arayan bireylerle işverenlerin buluştuğu bir dijital platformun veri tabanı tasarımı yapılmıştır. Sistem, iş ilanlarının yayınlanması, iş başvurularının alınması, kullanıcıların yeteneklerinin kaydedilmesi ve işverenlerin sektörlere göre sınıflandırılması gibi işlevleri destekleyecek şekilde kurgulanmıştır. Bu amaçla, oluşturulan varlıklar ve ilişkiler doğrultusunda kavramsal model oluşturulmuştur.

Sistem temelde iş arayan ve işverenlerden oluşturulmuştur.

Her iş arayan kullanıcılara ait olmak üzere belirli nitelikler vardır. Bunlar şu şekildedir :

- İş arayanın kendisine ait kullanıcı numarası (`JOBSEEKERID`), isim-soy isim bilgisi (`FULLNAME`), sisteme kayıt olabilmesi ve dijital platformla alakalı bilgi alabilmesi adına kullanıcının kendisine ait mail adresi bilgisi (`EMAIL`), telefon numarası bilgisi (`PHONENUMBER`), ait olduğu şehir (`CITY`), deneyim bilgisi (`EXPERIENCEYEAR`), eğitim düzeyi (`EDUCATIONLEVEL`) bilgileri tutulur.

İş arayanlar bu bilgiler ile sisteme kayıt olduktan sonra platform üzerinden ilanlara başvurabilir. İş araya kullanıcılar yeteneklerini belirtebilir. Bu yapılan başvurular kayıt tablosunda tutulur.

Her işveren firmalara ait bazı bilgiler firmalara ait nitelikler şeklinde veri tabanında tutulur.

- İşverenin kendisine ait işveren numarası (`EMPLOYERID`) veri tabanında tutulur. İşverenin firma adı (`COMPANYNAME`), ait olduğu sektör (`INDUSTRYID`), firmada çalışan personel sayısı (`COMPANYSIZE`), firmaya ait mail adresi (`EMAIL`) ve firmanın iletişim için kullandığı telefon numarası (`PHONENUMBER`) bilgileri veri tabanında tutulur.

Ayrıca varlıklar arasındaki ilişkinin daha düzenli olması ve anlaşılır olması için veri tabanı sistemine diğer varlıklar da eklenmiştir. Bunlar varlıklar ve içerdikleri şu şekildedir :

İş ilanları tablosu, her bir ilanla ilgili başlık, açıklama, maaş, konum, iş ilanı tarihi ve iş türü bilgilerini içerir ve yalnızca bir işverene bağlıdır. Başvurular tablosu, iş arayanların hangi ilana başvurduğunu ve başvurunun durumunu saklar. Yetenekler tablosu, iş arayanların sahip olduğu becerileri ve seviyelerini gösterir. İş türleri ve sektörler, sistemde tanımlı sabit değerlerdir ve ilanlar ile işverenlerle ilişkilidir.

1-)VARLIK - İLİŞKİ MODELLERİ

Veri tabanı tasarımında ilk ve en önemli adım, verilerin yapısını ve birbirleriyle olan ilişkilerini doğru şekilde modellemektir. Bu noktada kullanılan temel yöntemlerden biri Varlık-İlişki Modeli (Entity Relationship Model)'dir. Varlık-İlişki modeli, gerçek dünya kavramlarının veri tabanı ortamına taşınmasını sağlayan mantıksal bir yapıdır.

ER Modeli, veri analizinde önemli bir araçtır çünkü:

- **Chen Notasyonu:** Klasik ve teorik gösterim biçimidir. Varlıklar dikdörtgen, ilişkiler elmas şekliyle temsil edilir; nitelikler ise oval ile gösterilir. Daha çok akademik çalışmalarda tercih edilir.
- **Crow's Foot (Karga Ayağı) Notasyonu:** Uygulamalı veri tabanı tasarımlarında en yaygın kullanılan modeldir. Varlıklar kutu, ilişkiler çizgi ve özel uç sembolleriyle gösterilir. SQL tabanlı sistemlerle birebir uyumlu olduğundan, endüstriyel projelerde tercih edilir.

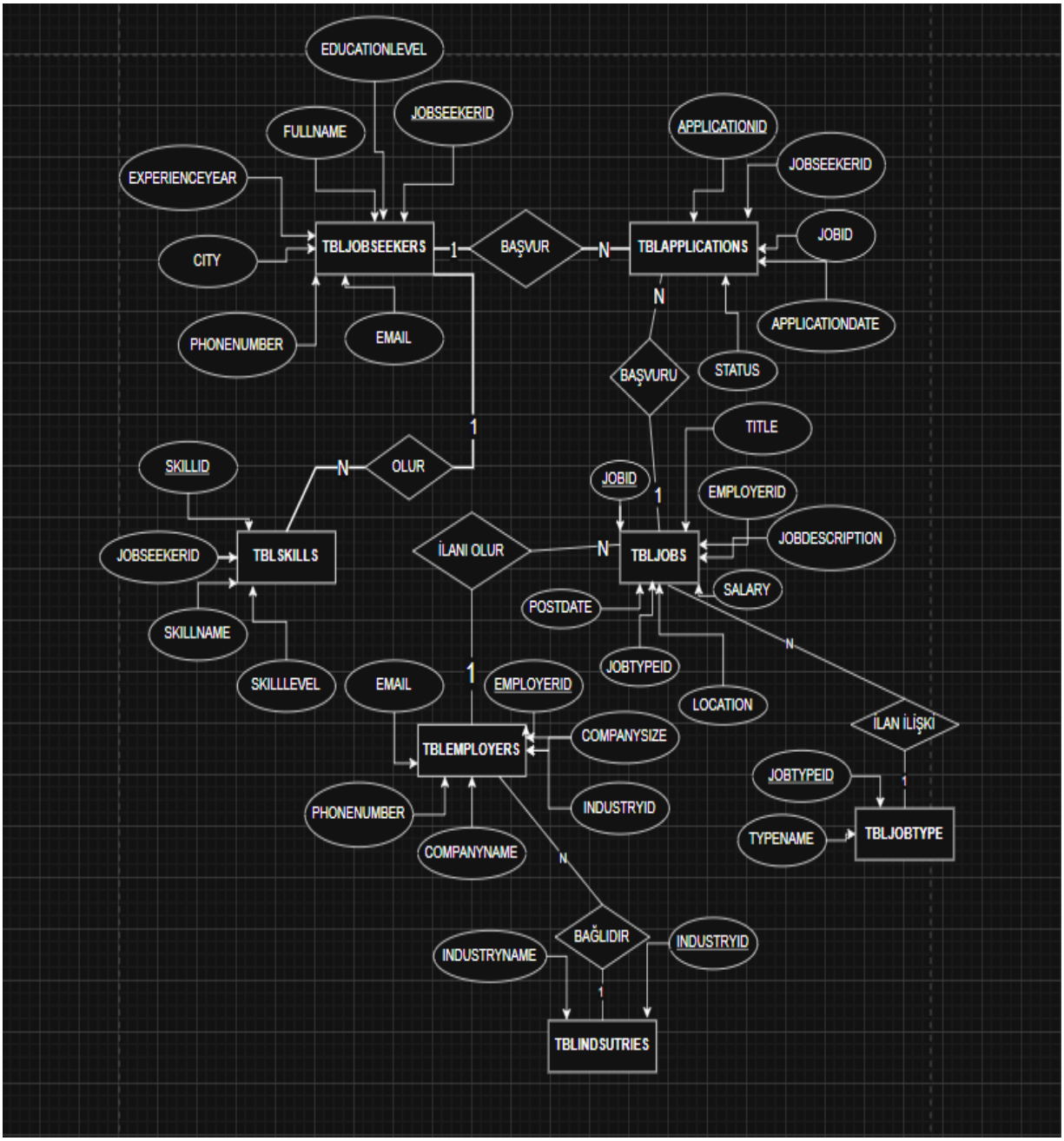
İLİŞKİ	TÜRÜ	AÇIKLAMA
TBLJOBSEEKERS - TBLSKILLS	1:N	Bir iş arayanın birçok becerisi olabilir. Her beceri tek bir iş arayana bağlı.
TBLJOBSEEKERS - TBLAPPLICATIONS	1:N	Bir iş arayan birçok başvuruda bulunabilir.
TBLAPPLICATIONS - TBLJOBS	N:1	Bir iş ilanına birden fazla kişi başvurabilir.
TBLEMPLOYERS - TBLJOBS	1:N	Bir işverenin birden fazla ilanı olabilir.
TBLJOBTYPE - TBLJOBS	1:N	Bir tür birden fazla ilan ilişki ile ilişkilidir
TBLEMPLOYERS - TBLINDUSTRIES	N:1	Her işveren yalnızca sadece bir sektöre bağlıdır.

Tabloda varlıklar ve ilişkileri belirtilmiştir. Her satırda, iki tablo arasındaki ilişkinin yönü ve bunla ilgili açıklama verilmiştir.

Chen Notasyonu:

Aşağıda yer alan varlık-ilişki diyagramı (ER Diyagramı) projemde oluşturulan varlıkları, bu varlıklara ait nitelikleri ve varlıklar arasındaki ilişkileri Chen notasyonuna uygun olarak gösterimi yapılmıştır.

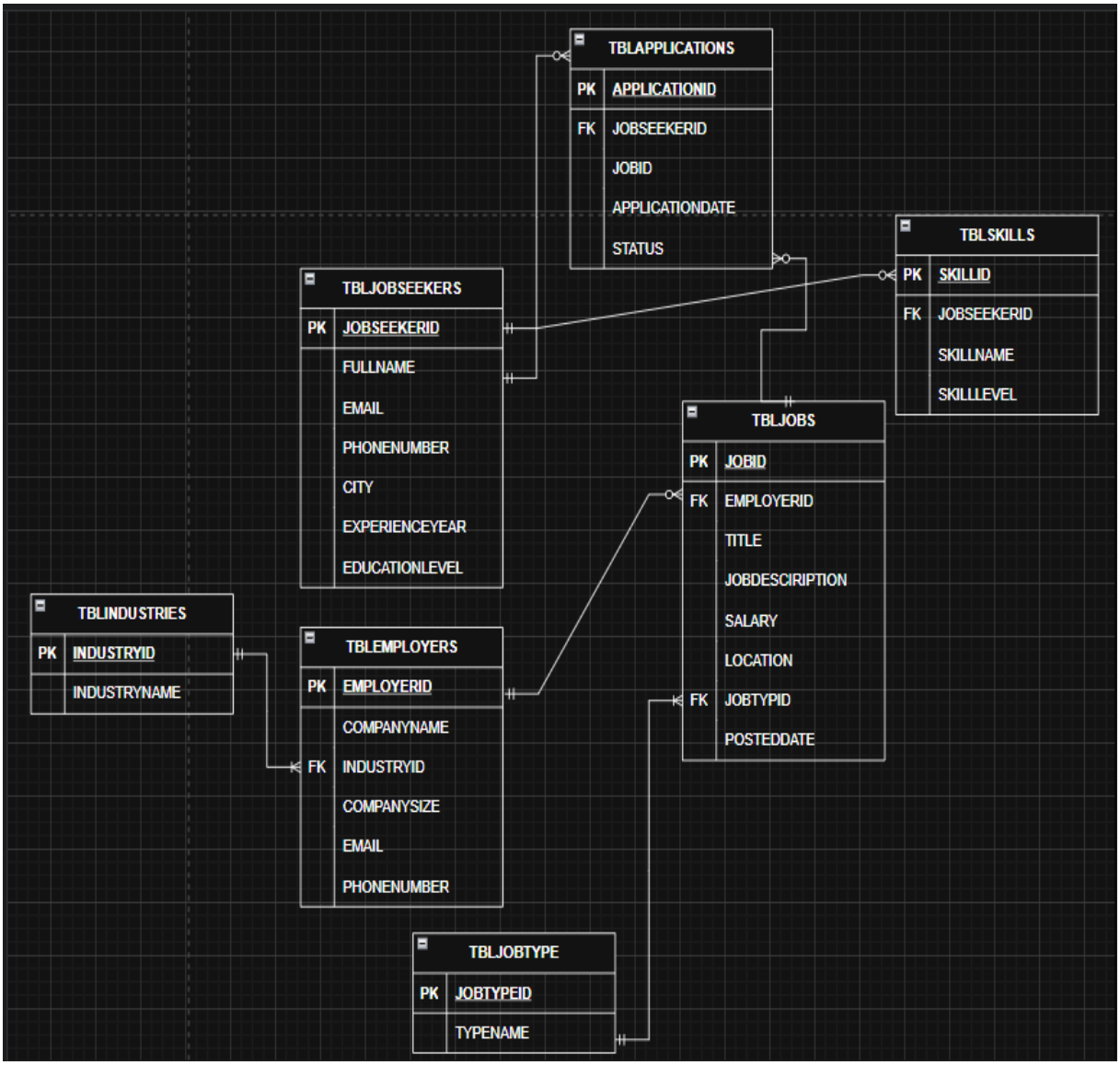
Diyagramda varlıklar dikdörtgen şekiller ile gösterilmiştir. Bu varlıklara ait nitelikler (attributes) oval semboller ile gösterilmiştir. Nitelikler arasından varlıklara ait birincil anahtar olanlar altı çizgili olarak ifade edilmiştir. Varlıklar arasındaki ilişkiler ve bu ilişkilerin türleri belirtilmiştir.



Crow’s Foot Notasyonu:

Aşağıda yer alan varlık-ilişki diyagramı, veri tabanı yapısını crow’s foot (kaz ayağı) notasyonuna uygun olarak verilmiştir. Bu diyagram, veri tabanı içerisindeki tabloları varlıklar şeklinde bunların içerisindeki nitelikleri tablolar içerisindeki satırlar halinde gösterilmiştir. Aynı zamanda bu tablolar arasındaki ilişkileri de semboller ile ifade eder.

Crow’s foot gösterimini varlık-ilişki şemalarında ayrı tutan durum veri tabanı yapısına en uygun gösterim olduğu içindir. Bundan kaynaklı projelerde en yaygın kullanılan tekniktir.



2-)İLİŞKİSEL CEBİR İFADELERİ

İlişkisel cebir (Relational algebra) ilişkisel veri tabanlarında bulunan verilere erişim için kullanılan bir matematiksel sorgulama dilidir. Bu yapı ilişkisel veri tabanlarında birden fazla varlıklar arasında bulunan ilişkilere erişim ve bunlar arasında işlem yapmak için kullanılan bir yapıdır. Aynı zamanda veri kümesini elde etmemizi sağlayan bir yapıdır. Bu yapı sayesinde herhangi bir kodlama işlemine gerek kalmadan ilişkisel veri tabanlarında işlem yapmamıza olanak sağlar.

İlişkisel cebir ifadeleri SQL gibi belirli ortamlarda doğrudan çalıştırılan bir yapı değildir. Üzerinde çalıştığımız veri tabanı üzerinde matematiksel anlamda işlem yapmamızı sağlar. Yani herhangi SQL dilindeki gibi herhangi bir derleyici ve yorumlayıcıya ihtiyaç duymaz. Bu ifadeleri direkt veri tabanında kullanabilmemiz için oluşturulan cebir ifadeleri daha sonra SQL sorgulama diline dönüştürülür ve o şekilde uygulanır.

İlişkisel cebir ifadelerini oluştururken kullandığımız temel işlemler arasında seçim (σ), yansıtma (π), birleşim (∪), fark (-), kartezyen çarpım (×), birleşim (⋈) ve bölme (÷) gibi işlemler bulunur.

Örnek İlişkisel Cebir İfadeleri:

- İş arayanların sadece "Python" becerisine sahip olanların isimleri listeleyiniz.

  πFullName (σ SkillName='Python' (TBLJOBSEEKERS ⋈ TBL_SKILLS))

- Her iş ilanının türünü ve bağlı olduğu işverenin sektörünü listeleyiniz.

  π Title, TypeName, IndustryName

((TBLJOBS ⋈ TBLJOBTYPE) ⋈ TBLEMPLOYERS ⋈ TBLINDUSTRIES)

- Yazılım ve Bilişim Teknolojileri" sektöründe çalışmakta olan işverenlerin ilanlarını ve maaş bilgilerini listeleyiniz.
  π Title, Salary ((σ IndustryName='Yazılım ve Bilişim Teknolojileri' (TBLINDUSTRIES)
  ⋈ TBLEMPLOYERS) ⋈ TBLJOBS)
- Tam Zamanlı" olmayan (≠) tüm iş ilanlarının başlıklarını listeleyiniz.
  π Title (σ TypeName ≠ 'Tam Zamanlı' (TBLJOBS ⋈ TBLJOBTYPE))

3-) NORMALİZASYON

Normalizasyon, çok fazla satır ve sütundan oluşan bir tabloyu gereksiz fazla satır ve sütundan kurtarmaktadır. Veri tabanını daha kolay anlaşılır kılmak ve daha kolaylaştırmaktır. Yönetilebilir daha alt tablolara ayırmaktır. Bu şekilde veri tabanında bulunan verileri daha tutarlı hale getirmektir aynı zamanda gelecekte yapılacak işlemleri daha az maliyetli hale gelir. Yanlış ve eksik yapılan normalizasyon işlemleri sonucunda sistem daha karmaşık hale gelir bu sorunda zamanla başka problemlerin gelmesine sebebiyet verir.

Bu çalışmada, iş arayanlar ve işverenlerin aynı tabloda tutulduğu bir veri tabanı sistemi baz alınmış ve üç aşamalı normalizasyon süreci uygulanmıştır:

- **1. Normal Form (1NF)**: Atomik veriler sağlandı, tekrarlayan sütunlar kaldırıldı.
- **2. Normal Form (2NF)**: Kısmi bağımlılıklar ortadan kaldırıldı, ilişkisel veriler bağımsız hale getirildi.
- **3. Normal Form (3NF)**: Geçişli bağımlılıklar giderildi

Normalizasyon sayesinde veri bütünlüğü artırılmış, veri işleme hızları iyileştirilmiş ve ilişkisel model en verimli hale getirilmiştir.

0.Başlangıç (Denormalized) Karmaşık Tablo

Aykırılıkları, çok değerli alanları (veya ilişkisel olmayan tablo yapılarını), kısmi ve geçişli bağımlılıkları, tekrarlanan veri gruplarını içerir.

Normalizasyon Aşaması	Tablo Adı	Kolonlar	Açıklama
0.Başlangıç (Denormalized)	TBLUSERS	UserID, FullName, Email, PhoneNumber, City, UserType, ExperienceYears, EducationLevel, CompanyName, Industry, CompanySize, JobTitle, JobDescription, SalaryRange, PostedDate	Şu anlık durumda iş arayanlar ve iş verenler aynı tabloda, çok fazla gereksiz veri tekrarı var. Industry ve UserType gibi tekrarlayan veriler mevcut.

1.Normal Form (1NF)

Bu aşamaya geldiğimizde amacımız varsa çok değerli alanları tek değerli hale dönüştürmektir. Her alandaki değer atomik olmalıdır. Birden fazla bilgi tek bir sütunda tutulmaz. Her tabloda birincil anahtar (PK) tanımlanır. Tablolar ilişkisel yapıya dönüştürülür.

Normalizasyon Aşaması	Tablo Adı	Kolonlar	Açıklama
1.Normal Form (1NF)	TBLJOBSEEKERS	JOBSEEKERID, FULLNAME, EMAIL, PHONENUMBER, CITY, EXPERIENCEYEAR, EDUCATIONLEVEL	İş arayanlar ayrıldı. ayrı bir tabloya alındı.
	TBLEMPLOYERS	EMPLOYERID, COMPANYNAME, INDUSTRY, COMPANYSIZE, EMAIL, PHONENUMBER	İşverenler ayrıldı, şirket bilgileri artık bağımsız bir yapıya sahip.
	TBLJOBS	JOBID, EMPLOYERID, TITLE, JOBDESCRIPTION, SALARYRANGE, LOCATION, POSTEDDATE	İş ilanları ayrı bir tabloya taşındı, böylece her ilan tek bir kayıt oldu.

2.Normal Form (2NF)

Bu aşamada ise varsa kısmı bağımlılıkları yok etmektir. Kısmi bağımlılık fonksiyonel bağımlılığın özel durumu olup, bir sütunun birleşik anahtarın sadece bir parçasına bağımlı olmasıdır.

Normalizasyon Aşaması	Tablo Adı	Kolonlar	Açıklama
2.Normal Form (2NF)	TBLINDUSTRIES	INDUSTRYID, INDUSTRYNAME	Sektör bilgileri ayrı bir tabloya taşındı, artık gereksiz veri tekrarları önlendi.
	TBLJOBTYPES	JOBTYPEID, TYPENAME	İş ilanlarının çalışma türleri (Tam Zamanlı, Yarı

			Zamanlı vb.) ayrı bir tabloda tutuluyor.

3.Normal Form (3NF)

2NF'de sadece anahtarlara ilişkin fonksiyonel bağımlılıklar kullanılmıştır. Bunun dışındaki bağımlılıklarda tablolara dönüştürülerek 3NF'e ulaşılır. İkinci normal formdaki sorunlardan kurtulmak için de nitelikler arasındaki geçişli işlevsel bağımlılıkları ortadan kaldırmamız gerekir.

Normalizasyon Aşaması	Tablo Adı	Kolonlar	Açıklama
3.Normal Form (3NF)	TBLSKILLS	SKILLID, JOBSEEKERID, SKILLNAME, SKILLLEVEL	İş arayanların becerileri artık ayrı bir tabloya taşındı, her beceri bağımsız kaydediliyor.

Normalizsayon Süreçlerinin Detaylı Açıklaması

0.Başlangıç Durumu (Denormalized Hali)

İlk olarak, iş arayanlar ve işverenlerin aynı tabloda yer aldığı karmaşık bir veri modeli oluşturulmuştur.

Bu yapıdaki temel problemler şunlardır:

- Farklı türde kullanıcıları (iş arayanlar ve işverenler) aynı tabloda saklamak veri tekrarına ve gereksiz NULL değerlerine sebep olmaktadır.
- Birçok sütun sadece belirli kullanıcı türleri için anlamlıdır. Örneğin:
    - İş arayanlar için EXPERIENCEYEAR, EDUCATIONLEVEL sütunları anlamlıdır.
    - İşverenler için COMPANYNAME, INDUSTRY, COMPANYNAME, JOBTITLE, JOBDESCRIPTION sütunları anlamlıdır.
    - Fakat bu sütunlar tek bir tabloda olduğu için gereksiz boş değerler oluşmaktadır.
- Tekrarlayan bilgiler bulunmaktadır. Örneğin:
    - İş ilanlarındaki INDUSTRY (sektör) bilgisi, her ilanda tekrar yazılmak zorundadır.

- İş türü (Full-Time, Part-Time vb.) bilgisi, her ilanda tekrar yazılmaktadır.

Bu sorunları çözmek için 1. Normal Form'a (1NF) geçiş yapılmalıdır.

1.Normal Form (1NF)

Bu aşamada atomiklik sağlanmış ve tekrarlayan sütunlar kaldırılmıştır.

Yapılan işlemler:

1. İş arayanlar ve işverenler ayrı tablolara bölünmüştür.
   - TBLJOBSEEKERS (iş arayanları saklar).
   - TBLEMPLOYERS (işverenleri saklar).
2. İş ilanları ayrı bir tabloya taşınmıştır.
   - TBLJOBS tablosu oluşturularak ilanlar işverenlere bağlanmıştır.

Sağlanan Avantajlar:
Boş değerler büyük ölçüde kaldırıldı.
Tekrarlayan sütunlar temizlendi.
Her satır atomik hale getirildi.

Bu aşamadan sonra, hâlâ var olan bazı kısmi bağımlılıklar giderilerek 2. Normal Form'a (2NF) geçilmelidir.

2.Normal Form (2NF)

Bu aşamada, kısmi bağımlılıklar ortadan kaldırılarak veri tekrarları önlenmiştir.

Problemler:

- TBLJOBS içinde yer alan INDUSTRY (Sektör) bilgisi her ilanda tekrar etmektedir.
- TBLJOBS içinde EMPLOYMENTTYPE (Çalışma Türü) bilgisi tekrar etmektedir.
- Aynı sektör bilgisi birden fazla işveren için yazılmak zorunda kalmaktadır.

Çözüm:

1. Sektörler ayrı bir tabloya taşındı (TBLINDUSTRIES).

- Artık işverenler sadece INDUSTRYID referansı ile sektörlerini belirtiyor.

2. İş türleri ayrı bir tabloya taşındı (TBLJOBTYPES).
    - Artık iş ilanları JOBTYPEID ile hangi çalışma modeline sahip olduğunu belirtiyor.
3. Sağlanan Avantajlar:
   Tekrarlayan veri azaltıldı.
   Her sütun artık tam bağımlı hale geldi.
   Veri bütünlüğü ve sorgu performansı artırıldı.

Bu aşamadan sonra, hâlâ bazı geçişli bağımlılıklar bulunmaktadır. Bu nedenle 3. Normal Form'a (3NF) geçilmelidir.

3.Normal Form (3NF)

Bu aşamada geçişli bağımlılıklar kaldırılarak veri modeli en optimize hale getirilmiştir.

Problemler:

- TBLJOBS tablosunda EMPLOYMENTTYPE doğrudan yer almaktadır ve bu bilgi tekrar etmektedir.
- İş arayanların becerileri tekrar eden sütunlar halinde tutulursa, ilerleyen aşamalarda yeni beceriler eklemek zorlaşacaktır.

Çözüm:

1. İş türleri için ayrı bir tablo (TBLJOBTYPES) oluşturuldu.
    - Artık ilanlar JOBTYPEID ile ilişkilendirildi ve veri tekrarları önlendi.
2. Beceriler (TBLSKILLS) ayrı bir tabloya alındı.
    - Artık iş arayanlar istediği kadar beceriye sahip olabilir ve sistem esnek hale geldi.

SağlananAvantajlar:
Geçişli bağımlılıklar kaldırıldı. İlişkisel veri tabanı en verimli hale getirildi. Güncellenebilir ve genişletilebilir bir yapı oluşturuldu.

4-)MS SQL'DE PROJEYİ OLUŞTURMA

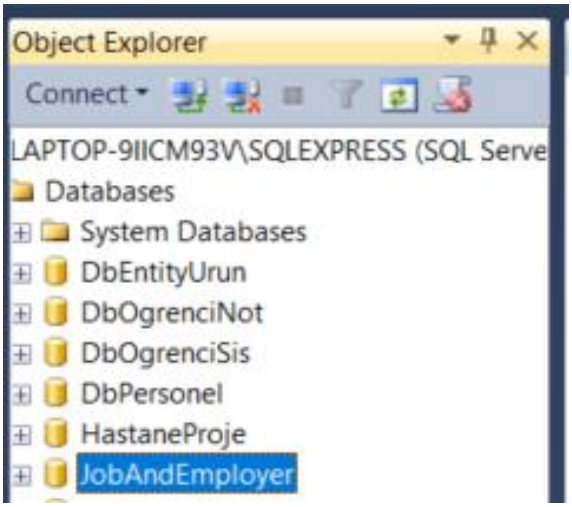
Bu proje, iş arayanlar ve iş verenleri bir araya getiren bir iş bulma platformunun veri tabanı modelini oluşturmayı amaçlamaktadır. Microsoft SQL Server (MS SQL) kullanılarak tasarlanan bu veri tabanı, ilişkisel veri tabanı yönetim sistemleri (RDBMS) prensiplerine uygun olarak yapılandırılmıştır.

Veri tabanı tasarımında, her veri grubunun bağımsız olarak saklanması ve tablolar arasındaki ilişkilerin en verimli şekilde yönetilmesi esas alınmıştır. Bu doğrultuda, projenin veri tabanı modeli aşağıdaki tablolar kullanılarak oluşturulmuştur:

- TBLJOBSEEKERS
- TBLEMPLOYERS
- TBLJOBS
- TBLAPPLICATIONS
- TBLSKILLS
- TBLINDUSTRIES
- TBLJOBTYPE

Öncelikle Managment Studio açıldıktan sonra ilgili sorgu yazılarak proje ile alakalı veri tabanı oluşturuldu.

```
CREATE DATABASE JobAndEmployer
```



Proje kapsamında veri tabanımıza ait tablolar veri tabanın işlevlik kazanması adına özenle hazırlanmıştır. Tablolar oluşturulurken “CREATE TABLE SQL” komutu kullanılmıştır.

TBLJOBSEEKERS

İş arayan kullanıcıların temel bilgilerini saklar.

```
CREATE TABLE TBLJOBSEEKERS(
JOBSEEKERID int primary key identity(1,1),
FULLNAME varchar(100) NOT NULL,
EMAIL varchar(100) UNIQUE NOT NULL,
PHONENUMBER VARCHAR(20) NOT NULL,
CITY VARCHAR (50) NOT NULL,
EXPERIENCEYEAR int,
EDUCATIONLEVEL VARCHAR(20) NOT NULL
)
```

TBLEMPLOYERS

İşverenlerin (şirketlerin) bilgilerini tutar.

```
CREATE TABLE TBLEMPLOYERS(
EMPLOYERID INT PRIMARY KEY IDENTITY(1,1),
COMPANYNAME VARCHAR(100) NOT NULL,
INDUSTRYID INT NOT NULL,
COMPANYSIZE INT NOT NULL,
EMAIL VARCHAR(100) UNIQUE,
PHONENUMBER VARCHAR(100),
FOREIGN KEY (INDUSTRYID) REFERENCES TBLINDUSTRIES(INDUSTRYID)
)
```

TBLINDUSTRIES

Şirketlerin faaliyet gösterdiği sektörleri yönetir.

```
CREATE TABLE TBLINDUSTRIES(
INDUSTRYID INT PRIMARY KEY IDENTITY(1,1),
INDUSTRYNAME VARCHAR(50) UNIQUE NOT NULL,
)
```

TBLJOBS

İşverenlerin oluşturduğu iş ilanlarını saklar.

```
CREATE TABLE TBLJOBS(
JOBID INT PRIMARY KEY IDENTITY(1,1),
EMPLOYERID INT NOT NULL,
TITLE VARCHAR(100) NOT NULL,
JOBDESCRIPTION TEXT NOT NULL,
SALARY DECIMAL(18,2),
LOCATION VARCHAR(100) NOT NULL,
JOBTYPEID INT NOT NULL,
POSTEDDATE DATE NOT NULL,
FOREIGN KEY (EMPLOYERID) REFERENCES TBLEMPLOYERS(EMPLOYERID),
FOREIGN KEY (JOBTYPEID) REFERENCES TBLJOBTYPE(JOBTYPEID)
)
```

TBLAPPLICATIONS

İş arayanların yaptığı iş başvurularını takip eder.

```
CREATE TABLE TBLAPPLICATIONS(
APPLICATIONID INT PRIMARY KEY IDENTITY(1,1),
JOBSEEKERID INT NOT NULL,
JOBID INT NOT NULL,
APPLICATIONDATE DATE NOT NULL,
STATUS VARCHAR(50) NOT NULL,
FOREIGN KEY (JOBSEEKERID) REFERENCES TBLJOBSEEKERS(JOBSEEKERID),
FOREIGN KEY (JOBID) REFERENCES TBLJOBS(JOBID)
)
```

TBLSKILLS

İş arayanların sahip olduğu becerileri saklar.

```
CREATE TABLE TBLSKILLS(
SKILLID INT PRIMARY KEY IDENTITY(1,1),
JOBSEEKERID INT NOT NULL,
SKILLNAME VARCHAR(100) NOT NULL,
SKILLLEVEL VARCHAR(50),
FOREIGN KEY (JOBSEEKERID) REFERENCES TBLJOBSEEKERS(JOBSEEKERID)
)
```

TBLJOBTYPE

İş ilanlarının çalışma türlerini belirler.

```
CREATE TABLE TBLJOBTYPE(
JOBTYPEID INT PRIMARY KEY IDENTITY(1,1),
TYPENAME VARCHAR(50) UNIQUE NOT NULL,
)
```

Projede, tablolar arasındaki ilişkiler belirlenerek MS SQL üzerinde tanımlanmıştır. İlişkisel veri modeli kullanılarak veri tekrarları en aza indirilmiş, performans veri bütünlüğü sağlanmıştır. Veri tabanı oluşturulurken şu adımlar takip edilmiştir :

- ✓ Her veri grubunu temsil eden tablolar belirlenmiş ve oluşturulmuştur.
- ✓ Tablolar arasında ilişkiler kurulmuş, yabancı anahtarlar (FOREIGN KEY) ile bağlar tanımlanmıştır.
- ✓ Tekrarlayan veriler, sektörler ve iş türleri gibi bağımsız tablolar yönetilmiştir.
- ✓ İş başvuruları gibi çoklu ilişkiler ayrı tablolarda modellenmiştir.

Bu yapı sayesinde MS SQL üzerinde oluşturulan veri tabanı, büyük ölçekli bir iş bulma platformunun temelini oluşturacak şekilde yapılmıştır.

5-) VERİ TABANINA ÖRNEK VERİ GİRİŞİ

Bir veri tabanı oluşturulduktan sonra, sistemin tasarlanan yapıya uygun şekilde çalıştığını doğrulamak için test amaçlı veri eklenmesi gerekmektedir. Örnek veri girişleri, veri tabanındaki tabloların yapısının ve ilişkilerinin gerçekçi senaryolarda nasıl çalıştığını görmek, hataları tespit etmek ve sorguların düzgün işleyip işlemediğini anlamak için oldukça önemlidir.

Bu projede, iş arayanlar, işverenler, iş ilanları, başvurular ve diğer ilgili tablolar için gerçekçi örnek veriler eklenerek sistemin işleyişi test edilmiştir. Örnek veriler, SQL INSERT komutları kullanılarak Microsoft SQL Server (MS SQL) üzerinde sisteme dahil edilmiştir.

Veri Girişi Süreci ve Adımları

Tablolara örnek veri girişi yapılırken veri tabanındaki tablolar arasındaki ilişkiler göz önünde bulundurularak adım adım bir veri ekleme sırası takip edilmiştir. Bunlar sırası ile şu şekildedir:

1. **Temel Verilerin Eklenmesi**

   Öncelikle diğer tabloların bağımlı olduğu ana tablolar doldurulmuştur:

   - **İş arayanlar (TBLJOBSEEKERS):** Sisteme kayıtlı iş arayanların temel bilgileri eklenmiştir.
   - **İşverenler (TBLEMPLOYERS):** Farklı sektörlerden çeşitli şirketler eklenerek işveren verileri sisteme dahil edilmiştir.
   - **Sektör bilgileri (TBLINDUSTRIES):** İşverenlerin faaliyet gösterdiği sektörler veri tabanına eklenmiştir.

TBLJOBSEEKERS

```
INSERT INTO TBLJOBSEEKERS(FULLNAME,EMAIL,PHONENUMBER,CITY,EXPERIENCEYEAR,EDUCATIONLEVEL)
VALUES ('Ahmet Yılmaz', 'ahmet.yilmaz@email.com', '05001234567', 'İstanbul', 5, 'Lisans'),
('Zeynep Demir', 'zeynep.demir@email.com', '05002345678', 'Ankara', 3, 'Ön Lisans'),
('Mehmet Kaya', 'mehmet.kaya@email.com', '05003456789', 'İzmir', 7, 'Yüksek Lisans'),
('Elif Aydın', 'elif.aydin@email.com', '05004567890', 'Bursa', 2, 'Lise'),
('Fatih Çelik', 'fatih.celik@email.com', '05005678901', 'Antalya', 10, 'Doktora'),
('Hüseyin Korkmaz', 'huseyin.korkmaz@email.com', '05006789012', 'Konya', 8, 'Lisans'),
('Merve Şahin', 'merve.sahin@email.com', '05007890123', 'Adana', 1, 'Ön Lisans'),
('Ali Güneş', 'ali.gunes@email.com', '05008901234', 'Trabzon', 6, 'Lisans'),
('Deniz Erkan', 'deniz.erkan@email.com', '05009012345', 'Samsun', 3, 'Lise'),
('Buse Özkan', 'buse.ozkan@email.com', '05001023456', 'Kayseri', 12, 'Yüksek Lisans'),
('Can Polat', 'can.polat@email.com', '05002134567', 'Gaziantep', 4, 'Lisans'),
('Selin Demirtaş', 'selin.demirtas@email.com', '05003245678', 'Mersin', 2, 'Lise'),
('Oğuz Kaan', 'oguz.kaan@email.com', '05004356789', 'Malatya', 5, 'Lisans'),
('Ezgi Yalçın', 'ezgi.yalcin@email.com', '05005467890', 'Erzurum', 9, 'Doktora'),
('Emre Sarı', 'emre.sari@email.com', '05006578901', 'Eskişehir', 7, 'Lisans'),
('Gizem Uslu', 'gizem.uslu@email.com', '05007689012', 'Sakarya', 1, 'Ön Lisans'),
('Umut Yıldırım', 'umut.yildirim@email.com', '05008790123', 'Çanakkale', 10, 'Lisans'),
('Cem Kılıç', 'cem.kilic@email.com', '05009801234', 'Denizli', 3, 'Yüksek Lisans'),
('Sibel Aktaş', 'sibel.aktas@email.com', '05001912345', 'Şanlıurfa', 6, 'Lisans'),
('Serkan Tuncel', 'serkan.tuncel@email.com', '05002023456', 'Hatay', 2, 'Lise'),
('Burcu Tekin', 'burcu.tekin@email.com', '05003134567', 'Manisa', 5, 'Lisans'),
('Barış Özdemir', 'baris.ozdemir@email.com', '05004245678', 'Tekirdağ', 7, 'Doktora'),
('Gökhan Aslan', 'gokhan.aslan@email.com', '05005356789', 'Aydın', 4, 'Lisans'),
```

TBLEMPLOYERS:

```
INSERT INTO TBLEMPLOYERS(COMPANYNAME,INDUSTRYID,COMPANYSIZE,EMAIL,PHONENUMBER)
VALUES ('TeknoSoft Yazılım', 1, 150, 'info@teknosoft.com', '02120000000'),
('Mega İnşaat A.Ş.', 2, 500, 'info@megainsaat.com', '03120000001'),
('FinBank', 3, 7000, 'info@finbank.com', '05120000004'),
('SanayiTek Fabrikası', 4, 1200, 'info@sanayitek.com', '07120000006'),
('Hızlı Alışveriş', 7, 350, 'info@hizlialisveris.com', '08120000007'),
('Speed Motors', 6, 2000, 'info@speedmotors.com', '10120000009'),
('Organik Tarım Ltd.', 7, 150, 'info@organiktarim.com', '12120000011'),
('Yeni Medya Ajansı', 8, 80, 'info@yenimedya.com', '13120000012'),
('SavunmaTech', 9, 900, 'info@savunmatech.com', '14120000013'),
('CodeMaster Yazılım', 1, 120, 'info@codemaster.com', '02124567890'),
('Mimari Yapı İnşaat', 2, 450, 'info@mimariyapi.com', '03124567891'),
('Global Taşımacılık', 5, 280, 'info@globaltasimacilik.com', '02224567892'),
('Güven Bank', 3, 5000, 'info@guvenbank.com', '05124567894'),
('Endüstri Makine Sanayi', 4, 1300, 'info@endustrisanayi.com', '07124567896'),
('TrendMarket Alışveriş', 7, 400, 'info@trendmarket.com', '08124567897'),
('Lüks Tatil Otelleri', 5, 600, 'info@luxtatil.com', '09124567898'),
('OtoHız Motors', 6, 1500, 'info@otohizmotors.com', '10124567899'),
('Doğa Tarım A.Ş.', 7, 180, 'info@dogatarim.com', '12124567901'),
('PR Medya Ajansı', 8, 95, 'info@prmedya.com', '13124567902'),
```

TBLINDUSTRIES:

```
INSERT INTO TBLINDUSTRIES(INDUSTRYNAME)
VALUES
('Yazılım ve Bilişim Teknolojileri'),
('İnşaat ve Yapı'),
('Finans ve Bankacılık'),
('Üretim ve Sanayi'),
('Turizm ve Otelcilik'),
('Otomotiv'),
('Gıda ve Tarım'),
('Medya ve İletişim'),
('Havacılık ve Savunma Sanayi'),
('Sigorta'),
('Danışmanlık ve İnsan Kaynakları'),
('Hukuk ve Adalet'),
('Elektrik ve Elektronik'),
('Telekomünikasyon');
```

2. İlişkili Verilerin Eklenmesi

Ana tablolar oluşturulduktan sonra, bu verilerle ilişkili tablolar için girişler yapılmıştır.

- **İş ilanları (TBLJOBS):** İşverenler tarafından oluşturulan iş ilanları sisteme eklenmiştir. İş ilanlarının her biri bir işverene ve bir sektöre bağlıdır.
- **İş arayanların becerileri (TBLSKILLS):** İş arayanların sahip olduğu beceriler, her iş arayanla ilişkilendirilerek eklenmiştir.
- **İş Türleri (TBLJOBTYPE):** Tam zamanlı, yarı zamanlı, freelance gibi iş türleri veri tabanına dahil edilmiştir.
- **Başvurular (TBLAPPLICATIONS):** İş arayanların belirli iş ilanlarına yaptığı başvurular kaydedilmiştir. Bu sayede iş arayanların iş ilanları ile olan ilişkisi test edilmiştir.

TBLJOBS

```
INSERT INTO TBLJOBS(EMPLOYERID,TITLE,JOBDESCRIPTION,SALARY,LOCATION,JOBTYPEID,POSTEDDATE)
VALUES
(4, 'Üretim Mühendisi', 'Üretim hattında makinelerle çalışacak mühendis aranıyor.', 40000, 'Kocaeli', 1, GETDATE()),
(5, 'Satış Temsilcisi', 'Mağazada müşteri ilişkilerini yönetecek satış danışmanı aranıyor.', 28000, 'İzmir', 2, GETDATE()),
(6, 'Otomotiv Mekanikeri', 'Araç bakım ve tamir konusunda yetkin otomotiv teknisyeni aranıyor.', 39000 , 'Bursa', 1, GETDATE()),
(7, 'Tarım Mühendisi', 'Organik tarım üzerine çalışacak ziraat mühendisi aranıyor.', 45000, 'Antalya', 5, GETDATE()),
(8, 'Medya Planlama Uzmanı', 'Reklam kampanyaları için medya planlaması yapacak uzman aranıyor.', 42000, 'Ankara', 3, GETDATE()),
(9, 'Savunma Mühendisi', 'Savunma sanayi projelerinde görev alacak mühendis aranıyor.', 57000, 'Eskişehir', 1, GETDATE()),
(10, 'Yazılım Geliştirici', 'Java ve Python dillerinde deneyimli yazılım mühendisi aranıyor.', 49000, 'İstanbul', 1, GETDATE()),
(11, 'Mimar', 'İnşaat projelerinde yer alacak deneyimli mimar aranıyor.', 46500 , 'Ankara', 1, GETDATE()),
(12, 'Lojistik Uzmanı', 'Depo ve tedarik zinciri yönetimi konusunda tecrübeli uzman aranıyor.', 35000, 'Bursa', 2, GETDATE()),
(13, 'Müşteri Temsilcisi', 'Banka müşteri hizmetlerinde çalışacak temsilci aranıyor.', 28000, 'İstanbul', 5, GETDATE()),
(14, 'Makine Mühendisi', 'Üretim hattında çalışacak deneyimli makine mühendisi aranıyor.', 51250, 'Kocaeli', 1, GETDATE()),
(15, 'E-Ticaret Yöneticisi', 'Online satış kanallarını yönetecek uzman aranıyor.', 30000, 'İstanbul', 6, GETDATE()),
(16, 'Otel Müdürü', 'Lüks otelde yönetim deneyimi olan otel müdürü aranıyor.',55800, 'Antalya', 1, GETDATE()),
(17, 'Otomotiv Mühendisi', 'Araç geliştirme süreçlerinde çalışacak mühendis aranıyor.', 56000, 'Bursa', 1, GETDATE()),
(18, 'Dijital Pazarlama Uzmanı', 'SEO ve sosyal medya yönetimi bilen uzman aranıyor.', 38000, 'İstanbul', 3, GETDATE()),
(19, 'Blockchain Uzmanı', 'Kripto para ve blockchain teknolojileri konusunda uzman aranıyor.', 46700, 'İstanbul', 5, GETDATE()),
(20, 'Veri Bilimci', 'Makine öğrenmesi ve büyük veri analizi yapabilecek veri bilimci aranıyor.', 50000, 'İstanbul', 1, GETDATE());


(6, 'Otomotiv Teknikeri', 'Servis bakım süreçlerinde çalışacak otomotiv teknikeri aranıyor.', 32000, 'Bursa', 1, GETDATE()),
(7, 'Ziraat Mühendisi', 'Tarımsal üretimde çalışacak, organik tarım bilgisine sahip mühendis aranıyor.', 36200, 'Antalya', 5, GETDATE()),
(8, 'Reklam Yöneticisi', 'Dijital reklam kampanyalarını yönetecek medya planlama uzmanı aranıyor.', 41000, 'Ankara', 3, GETDATE()),
(9, 'Silah Sistemleri Mühendisi', 'Savunma sanayi projelerinde yer alacak mekanik mühendis aranıyor.', 47500, 'Eskişehir', 1, GETDATE()),
(10, 'Veritabanı Yöneticisi', 'MS SQL ve Oracle veri tabanlarında deneyimli yönetici aranıyor.',61000, 'İstanbul', 1, GETDATE()),
(11, 'İç Mimar', 'Restorasyon ve iç tasarım projelerinde görev alacak mimar aranıyor.',47000, 'Ankara', 1, GETDATE()),
(12, 'Tedarik Zinciri Uzmanı', 'Depo ve lojistik yönetimi süreçlerini bilen uzman aranıyor.',35000, 'Bursa', 2, GETDATE()),
(13, 'Kredi Uzmanı', 'Banka şubelerinde kredi başvurularını yönetecek uzman aranıyor.',58900, 'İstanbul', 5, GETDATE())
```

TBLJOBTYPE

```
INSERT INTO TBLJOBTYPE(TYPENAME)
VALUES
('Tam Zamanlı'),
('Yarı Zamanlı'),
('Freelance'),
('Stajyer'),
('Uzaktan Çalışma'),
('Dönemsel'),
('Part-Time Hafta Sonu')
```

TBLSKILLS

```
INSERT INTO TBLSKILLS(JOBSEEKERID,SKILLNAME,SKILLLEVEL)
VALUES
(1, 'C#', 'İleri'),
(1, 'MSSQL', 'Orta'),
(2, 'AutoCAD', 'İleri'),
(2, '3D Max', 'Başlangıç'),
(3, 'Python', 'İleri'),
(3, 'Makine Öğrenmesi', 'Orta'),
(4, 'Microsoft Office', 'İleri'),
(4, 'İletişim Becerileri', 'Orta'),
(5, 'Veri Analizi', 'İleri'),
(5, 'SPSS', 'Orta'),
(6, 'SAP ERP', 'Orta'),
(6, 'Excel VBA', 'Başlangıç'),
(7, 'Satış ve Pazarlama', 'İleri'),
(7, 'CRM Sistemleri', 'Orta'),
(8, 'Network Yönetimi', 'İleri'),
(8, 'Linux', 'Orta'),
(9, 'Java', 'İleri'),
(9, 'Spring Boot', 'Orta'),
(10, 'Proje Yönetimi', 'İleri'),
(10, 'Scrum & Agile', 'Orta'),
(11, 'SolidWorks', 'İleri'),
(11, 'Teknik Çizim', 'Orta'),
(12, 'Sosyal Medya Yönetimi', 'İleri'),
(12, 'Google Ads', 'Orta'),
(13, 'Muhasebe', 'İleri'),
(13, 'Mikro ve Logo Yazılımları', 'Orta'),
(14, 'Photoshop', 'İleri'),
(14, 'Illustrator', 'Orta'),
(15, 'Mobil Geliştirme', 'İleri'),
(15, 'Flutter', 'Orta'),
(16, 'Veri Bilimi', 'İleri'),
(16, 'R Programlama', 'Orta'),
(17, 'HTML & CSS', 'İleri'),
(17, 'ReactJS', 'Orta'),
(18, 'C++', 'İleri'),
(18, 'Embedded Systems', 'Orta'),
(19, 'İnsan Kaynakları', 'İleri'),
(19, 'Eğitim Yönetimi', 'Orta'),
(20, 'JavaScript', 'İleri'),
(20, 'Node.js', 'Orta'),
(21, 'Proje Yönetimi', 'İleri'),
(21, 'Scrum & Agile', 'Orta'),
(22, 'Makine Öğrenmesi', 'İleri'),
(22, 'Yapay Zeka', 'Orta'),
(23, 'İş Analizi', 'İleri'),
(23, 'Veri Modelleme', 'Orta'),
(24, 'Sosyal Medya Yönetimi', 'İleri'),
(24, 'Reklam Stratejileri', 'Orta'),
(25, 'Bilgisayar Mühendisliği', 'İleri'),
(25, 'Python', 'Orta'),
(26, 'Elektrik Devre Tasarımı', 'İleri'),
(26, 'PCB Dizayn', 'Orta'),
(27, 'PHP & Laravel', 'İleri'),
(27, 'Web Geliştirme', 'Orta'),
(28, 'Mobil Uygulama Geliştirme', 'İleri'),
(28, 'Android Studio', 'Orta'),
(29, 'İnsan Kaynakları Yönetimi', 'İleri'),
(29, 'Eğitim Yönetimi', 'Orta'),
(30, 'Dijital Pazarlama', 'İleri'),
(30, 'SEO & SEM', 'Orta');
```

TBLAPPLICATIONS

```
INSERT INTO TBLAPPLICATIONS (JOBSEEKERID, JOBID, APPLICATIONDATE, Status)
VALUES
(1, 2, '2025-04-01', 'Beklemede'),
(2, 5, '2025-04-02', 'Kabul Edildi'),
(3, 6, '2025-04-03', 'Beklemede'),
(4, 7, '2025-04-04', 'Reddedildi'),
(5, 8, '2025-04-05', 'Beklemede'),
(6, 9, '2025-04-06', 'Kabul Edildi'),
(7, 10, '2025-04-07', 'Beklemede'),
(8, 11, '2025-04-08', 'Reddedildi'),
(9, 12, '2025-04-09', 'Beklemede'),
(10, 13, '2025-04-10', 'Kabul Edildi'),
(11, 14, '2025-04-11', 'Beklemede'),
(12, 15, '2025-04-12', 'Reddedildi'),
(13, 16, '2025-04-13', 'Beklemede'),
(14, 17, '2025-04-14', 'Kabul Edildi'),
(15, 18, '2025-04-15', 'Beklemede'),
(16, 19, '2025-04-16', 'Beklemede'),
(17, 20, '2025-04-17', 'Reddedildi'),
(18, 5, '2025-04-18', 'Kabul Edildi'),
(19, 6, '2025-04-19', 'Beklemede'),
(20, 7, '2025-04-20', 'Reddedildi');
```

Not :

Temel verilerin ve ilişkili verilerin eklenmesine dair SQL ortamında “INSERT” sorgularının görüntüleri eklenirken resimler parça parça eklenmiştir. Tek bir ekran görüntüsüne sığmadığından kaynaklı bu yol tercih edilmiştir.

6-)Veri Tabanına Ait Örnek Sorgular

Veri tabanına ait varlıklar ve ilişkileri kullanılarak örnek sorgular yazılmıştır. Bu sorgular temel olarak "Select" yapısı kullanılarak yazılmıştır. Böylece veri tabanında bulunan verilerin ne amaçla var olduğu daha da anlaşılır hale gelmiş ve daha da pekiştirilmiştir.

Sorgularda "select" ile birlikte subquery, group by, join ve having yapıları da kullanılmıştır. Birbiri ile ilişkili varlıkların belirtilen yapılar ile birleştirilmesi ve gruplanması sağlanmıştır. Böylece ilişkisel veri tabanının yapısı anlamak daha kolay hale gelmiştir.

Sorgu 1

Bu sorguda işverenler için en çok başvuru alan ilan verilmiştir. Sorguda temel select yapısı ile birlikte ilişkili varlıkların birleşimi için join yapısı, group by ve order by kullanılmıştır. En çok başvuruları görüntülemek adına başvuru sayısı 1'den büyük olanlar seçilmiştir.

```
Select a.JOBID as 'İlan No',COUNT(*) as 'Başvuru Sayısı' From TBLJOBS j
Inner Join TBLAPPLICATIONS a
On a.JOBID = j.JOBID
Group By a.JOBID
Having count(*)>1
Order by count(*) desc
```

Results | Messages

	İlan No	Başvuru Sa...
1	5	2
2	6	2
3	7	2

Sorgu 2

Bu sorgumuzda sistemde başvurusu kabul edilmiş iş arayan kullanıcıların hangi ilana başvurduğu listelenmiştir. İlgili sorgumuzda ilişkili tabloların bir arada bulunması adına join yapısı kullanılmıştır. Yalnızca başvurusu kabul edilen kullanıcıları listelemek için where ile ilgili şart belirtilmiştir.

```
Select FULLNAME,STATUS,o.JOBID as 'İlan No',TITLE From TBLAPPLICATIONS a
Inner Join TBLJOBSEEKERS j
On j.JOBSEEKERID = a.JOBSEEKERID
Inner Join TBLJOBS o
On o.JOBID = a.JOBID
Where STATUS = 'Kabul edildi'
```

Results | Messages

	FULLNAME	STATUS	İlan ...	TITLE
1	Zeynep Demir	Kabul Edildi	5	Üretim Mühendisi
2	Hüseyin Korkmaz	Kabul Edildi	9	Medya Planlama Uzmanı
3	Buse Özkan	Kabul Edildi	13	Lojistik Uzmanı
4	Ezgi Yalçın	Kabul Edildi	17	Otel Müdürü
5	Cem Kılıç	Kabul Edildi	5	Üretim Mühendisi

Sorgu 3

Bu sorguda işverenler arasından bu zamana kadar 1'den fazla ilan yayımlayanlar verilmiştir. Burda ilişkili tabloların bir arada olabilmesi için join yapısı kullanılmıştır. İlan sayısını verebilmek için gruplama yapılmış ve having ile bunun şartı verilmiştir.

```
Select j.EMPLOYERID,COMPANYNAME,count(*) as 'İlan sayısı' From TBLJOBS j
Inner Join TBLEMPLOYERS e
On j.EMPLOYERID = e.EMPLOYERID
Group By j.EMPLOYERID,COMPANYNAME
Having count(*)>=2
```

Results | Messages

	EMPLOYERID	COMPANYNAME	İlan sayısı
1	6	Speed Motors	2
2	7	Organik Tarım Ltd.	2
3	8	Yeni Medya Ajansı	2
4	9	SavunmaTech	2
5	10	CodeMaster Yazılım	2
6	11	Mimari Yapı İnşaat	2
7	12	Global Taşımacılık	2
8	13	Güven Bank	2

Sorgu 4

Bu sorguda isminin baş harfi “m” ile başlayan şehirlerde bulunan kullanıcıların başvurdukları ilanları listelemektir. Sorgu oluşturulurken ilişkili varlıkların bir arada olması için yine join yapısı kullanılmıştır. Baş harfi “m” ile başlayan şehirleri belirlemek için where şartı ile kümeleme operatörlerinden like komutu kullanılmıştır.

```
Select FULLNAME,o.JOBID as 'İlan No',TITLE From TBLAPPLICATIONS a
Inner Join TBLJOBSEEKERS j
On a.JOBSEEKERID = j.JOBSEEKERID
Inner Join TBLJOBS o
On o.JOBID = a.JOBID
where CITY like 'm%'
```

100 %

Results | Messages

	FULLNAME	İlan No	TITLE
1	Selin Demirtaş	15	Makine Mühendisi
2	Oğuz Kaan	16	E-Ticaret Yöneticisi

Sorgu 5

Bu sorguda amacımız iş ilanlarına başvurmamış iş arayan kullanıcıları listelemektir. Amacımız pasif durumda olan kullanıcıları tespit edip sistemin etkinliğini artırmaktır. Sorgu oluşturulurken alt sorgu mantığından faydalanılmıştır. Böylece tüm kullanıcılardan başvurusu olan kullanıcıları çıkarıp pasif durumda olanları tespit ettik.

```
Select FULLNAME from TBLJOBSEEKERS
where JOBSEEKERID not in
(select JOBSEEKERID from TBLAPPLICATIONS)
```

	FULLNAME
1	Burcu Tekin
2	Barış Özdemir
3	Gökhan Aslan
4	Yasemin Karaca
5	Levent Çetin
6	Pelin Öztürk
7	Efe Gündoğan
8	Melis Arslan
9	Hakan Bayraktar

10	Dilara Eren

Sorgu 6

Bu sorguda iş ilanı yayınlamayan işveren kullanıcı firmalar listelenmiştir. Sorguda birbiri ile ilişkili iki tablo bu sefer left join ile birleştirilmiştir. Bunu yapmamızdaki amaç ise left join ile tüm firmaları çağırmak ama where ile verdiğimiz şart ile yalnızca iş ilanı yayınlamayanları bulmaktır. Sorgu ile işverenler açısından sistemin etkinliğini artırmak amaçlanmıştır.

```
Select COMPANYNAME,JOBID,TITLE from TBLEMPLOYERS e
Left Join TBLJOBS j
On j.EMPLOYERID = e.EMPLOYERID
Where JOBID is null
```

Results | Messages

	COMPANYNAME	JOBID	TITLE
1	TeknoSoft Yazılım	NULL	NULL
2	Mega İnşaat A.Ş.	NULL	NULL

Sorgu 7

Bu sorgu ile sistemde hiç başvuru yapılmamış ve yalnızca bir kez başvurulmuş ilanları ve bu ilanların başlıklarını listelemektir. Burada ilişkili iki tablonun birleşimi için right join kullanılmıştır. Başvurular tablosunun tamamı getirilmiş ve ilanların yer aldığı tablodan da gelen veriler ile eşleşmiştir. Hiç başvuru yapılmamış ilanlar için null değer gelmiş ve diğer ilanlar da yalnızca bir kez başvuru yapılmış ilanlardır.

```
Select a.JOBID,TITLE,count(*) From TBLAPPLICATIONS a
Right Join TBLJOBS j
On a.JOBID = j.JOBID
Group By a.JOBID,TITLE
having count(*)=1
```

	JOBID	TITLE	(No column na...
1	20	Blockchain Uzmanı	1
2	19	Dijital Pazarlama Uzmanı	1
3	16	E-Ticaret Yöneticisi	1
4	2	Finansal Analist	1
5	NULL	İç Mimar	1
6	NULL	Kredi Uzmanı	1
7	13	Lojistik Uzmanı	1

	JOBID	TITLE	(No column na...
7	13	Lojistik Uzmanı	1
8	15	Makine Mühendisi	1
9	9	Medya Planlama Uzmanı	1
10	12	Mimar	1
11	14	Müşteri Temsilcisi	1
12	17	Otel Müdürü	1
13	18	Otomotiv Mühendisi	1

	JOBID	TITLE	(No column na...
15	NULL	Reklam Yöneticisi	1
16	10	Savunma Mühendisi	1
17	NULL	Silah Sistemleri Mühen...	1
18	8	Tarım Mühendisi	1
19	NULL	Tedarik Zinciri Uzmanı	1
20	NULL	Veri Bilimci	1
21	NULL	Veritabanı Yöneticisi	1

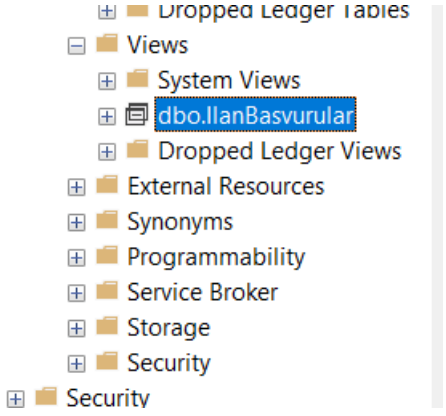
22	11	Yazılım Geliştirici	1
23	NULL	Ziraat Mühendisi	1

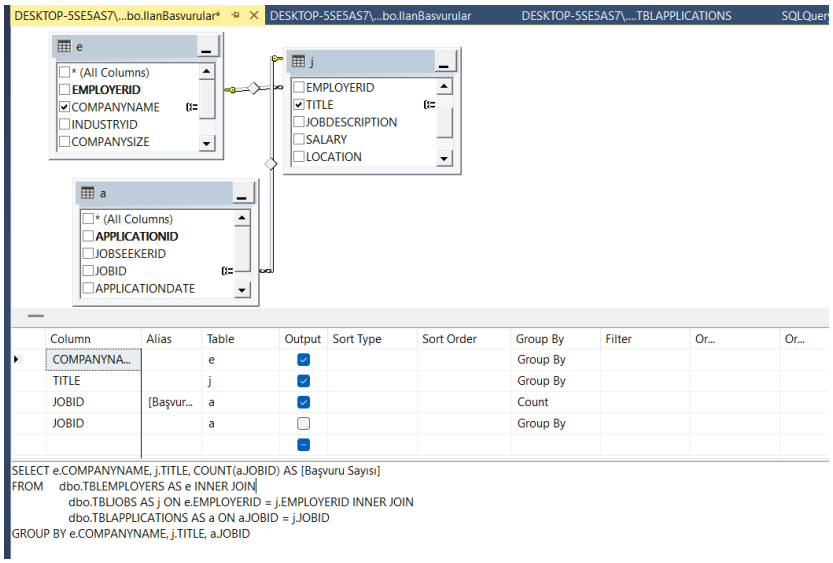
7-)VIEW

Kullanıcı tarafından elde edilen sanal tablolardır diyebiliriz. Aslında bir görünümdür diyebiliriz. Bu sanal tablolar veri tabanında fiziksel olarak yer kaplamazlar. View yapısı veri tabanında yer alan diğer tablolardan elde edilen verilerin kullanıcının isteğine bağlı olarak tek bir sanal tabloda buluşturmaktır. Özellikle fiziksel olarak varolan diğer tablolarda kullanıcılar tarafından sık kullanılan veriler var ise bunları tek bir view da buluşturmaktır amaç. Bu amaç kullanıcılara büyük anlamda bu verilerin bulunması adına kolaylık sağlar.

Proje kapsamında oluşturduğum veri tabanında birkaç view (sanal tablo) oluşturarak bunları örnekler ile sundum.

Öncelikle MS SQL ortamında view oluşturduktan sonra hangi konumda bulunabileceğini ve düzenlenebileceğini bakalım :





Resimlerde görüldüğü üzere MS SQL ortamında view (sanal tablo) nerde bulunabileceği ve nasıl düzenlenebileceği gösterilmiştir.

Proje için oluşturulan veri tabanında kullanıcı kolaylığın sağlanması ve sık kullanılan tablolardaki verilere ait view örnekleri şunlardır:

İşverenler için oluşturulan view (sanal tablo)

"Dbo.IlanBasvurular" adlı view da amacımız işverenlerin verdiği ilanlar ve bu ilanlara başvuru sayısı kaç adet olduğunu görmektir.

```
Create View IlanBasvurular
As
Select e.COMPANYNAME,j.TITLE,count(a.JOBID) as 'Başvuru Sayısı' From TBLEMPLOYERS e
Inner Join TBLJOBS j
on e.EMPLOYERID=j.EMPLOYERID
Inner Join TBLAPPLICATIONS a
On a.JOBID=j.JOBID
Group By e.COMPANYNAME,j.TITLE,a.JOBID
```

Belirtilen sorgu sonucunda ortaya çıkan aşağıdaki şekildeki gibidir.

	COMPANYNAME	TITLE	Başvuru Sayısı
▶	FinBank	Finansal An...	1
	SanayiTek Fabrik...	Üretim Müh...	2
	Hızlı Alışveriş	Satış Temsil...	2
	Speed Motors	Otomotiv M...	2
	Organik Tarım Lt...	Tarım Mühe...	1
	Yeni Medya Aja...	Medya Planl...	1
	SavunmaTech	Savunma M...	1
	CodeMaster Yaz...	Yazılım Geli...	1
	Mimari Yapı İnş...	Mimar	1
	Global Taşımacıl...	Lojistik Uzm...	1
	Güven Bank	Müşteri Tem...	1
	Endüstri Makine...	Makine Müh...	1
	TrendMarket Alı...	E-Ticaret Yö...	1
	Lüks Tatil Otelleri	Otel Müdürü	1
	OtoHız Motors	Otomotiv M...	1
	Doğa Tarım A.Ş.	Dijital Pazar...	1
	PR Medya Ajansı	Blockchain ...	1
*	*NULL*	*NULL*	*NULL*

Bu görünüm sayesinde işveren, yayınladığı tüm ilanları ve her bir ilana yapılan toplam başvuru sayısını tek bir sorgu ile görüntüleyebilir. Bu view yapıları sayesinde işverenler, sistem üzerinde karmaşık sorgulara gerek kalmadan kendi verilerini daha verimli bir şekilde analiz edebilir ve süreçlerini yönetebilir. Bu da sistemi kullanan kullanıcılara kullanım kolaylığı sağlar.

İlan detaylarını görmek için oluşturulan view (sanal tablo)

"Dbo.Ilanlar" adlı oluşturulan sanal tablo hem işveren kullanıcıların hem de iş arayan kullanıcıların ortak olarak bakabileceği, kullanıcıya ilan detaylarını daha kolay sunan bir view'dir. Bu viewda kullanıcılar ilanın temel bilgileri ile birlikte ilan veren kurumun sektörü, hangi konumda bulunduğu, ilanla alakalı maaş bilgisi ve ilanın açıklaması şeklinde bilgilere de erişebilir.

```
Create View Ilanlar
As
Select COMPANYNAME,INDUSTRYNAME,TITLE,JOBDESCRIPTION,SALARY,LOCATION,POSTEDDATE From TBLEMPLOYERS e
Inner Join TBLINDUSTRIES i
On i.INDUSTRYID=e.INDUSTRYID
Inner Join TBLJOBS j
On j.EMPLOYERID=e.EMPLOYERID
```

```
Select * From Ilanlar
```

00 %

Results | Messages

	COMPANYNAME	INDUSTRYNAME	TITLE	JOBDESCRIPTION	SALARY	LOCATION	POSTEDDATE
1	FinBank	Finans ve Bankacılık	Finansal Analist	Bankacılık sektöründe finansal analiz yapabilecek uz...	46000.00	İstanbul	2025-03-19
2	SanayiTek Fabrikası	Üretim ve Sanayi	Üretim Mühendisi	Üretim hattında makinelerle çalışacak mühendis ara...	40000.00	Kocaeli	2025-03-19
3	Hızlı Alışveriş	Gıda ve Tarım	Satış Temsilcisi	Mağazada müşteri ilişkilerini yönetecek satış danış...	28000.00	İzmir	2025-03-19
4	Speed Motors	Otomotiv	Otomotiv Mekanikeri	Araç bakım ve tamir konusunda yetkin otomotiv tekni...	39000.00	Bursa	2025-03-19
5	Organik Tarım Ltd.	Gıda ve Tarım	Tarım Mühendisi	Organik tarım üzerine çalışacak ziraat mühendisi ar...	45000.00	Antalya	2025-03-19
6	Yeni Medya Ajansı	Medya ve İletişim	Medya Planlama Uzmanı	Reklam kampanyaları için medya planlaması yapac...	42000.00	Ankara	2025-03-19
7	SavunmaTech	Havacılık ve Savunma Sanayi	Savunma Mühendisi	Savunma sanayi projelerinde görev alacak mühendi...	57000.00	Eskişehir	2025-03-19
8	CodeMaster Yazılım	Yazılım ve Bilişim Teknolojileri	Yazılım Geliştirici	Java ve Python dillerinde deneyimli yazılım mühendi...	49000.00	İstanbul	2025-03-19
9	Mimari Yapı İnşaat	İnşaat ve Yapı	Mimar	İnşaat projelerinde yer alacak deneyimli mimar aranı...	46500.00	Ankara	2025-03-19
10	Global Taşımacılık	Turizm ve Otelcilik	Lojistik Uzmanı	Depo ve tedarik zinciri yönetimi konusunda tecrübeli ...	35000.00	Bursa	2025-03-19
11	Güven Bank	Finans ve Bankacılık	Müşteri Temsilcisi	Banka müşteri hizmetlerinde çalışacak temsilci aran...	28000.00	İstanbul	2025-03-19
12	Endüstri Makine Sanayi	Üretim ve Sanayi	Makine Mühendisi	Üretim hattında çalışacak deneyimli makine mühen...	51250.00	Kocaeli	2025-03-19
13	TrendMarket Alışveriş	Gıda ve Tarım	E-Ticaret Yöneticisi	Online satış kanallarını yönetecek uzman aranıyor.	30000.00	İstanbul	2025-03-19
14	Lüks Tatil Otelleri	Turizm ve Otelcilik	Otel Müdürü	Lüks otelde yönetim deneyimi olan otel müdürü aran...	55800.00	Antalya	2025-03-19
15	OtoHız Motors	Otomotiv	Otomotiv Mühendisi	Araç geliştirme süreçlerinde çalışacak mühendis ara...	56000.00	Bursa	2025-03-19
16	Doğa Tarım A.Ş.	Gıda ve Tarım	Dijital Pazarlama Uzmanı	SEO ve sosyal medya yönetimi bilen uzman aranıyor.	38000.00	İstanbul	2025-03-19

İş başvuru durumları için oluşturulan view (sanal tablo)

"Dbo.BasvurulanIlanlar" sistemde iş arayan kullanıcıların aktif ilanlara başvuru durumlarını listeleyen bir view'dir. İş arayan kullanıcılar bu view sayesinde iş başvuru durumlarının ne aşamada olduğunu bunların statülerini sistem üzerinde daha kolay sorgulayabilir.

```
Create View BasvurulanIlanlar
As
Select FULLNAME,STATUS,count(a.JOBSEEKERID) as 'Başvurulan İlanlar' From TBLAPPLICATIONS a
Inner Join TBLJOBSEEKERS j
On j.JOBSEEKERID = a.JOBSEEKERID
Group By FULLNAME,STATUS
```

column JOBSEEKERID(int, not null)

```
Select * From BasvurulanIlanlar
```

0 %

Results | Messages

	FULLNAME	STATUS	Başvurulan İlanlar
	Ahmet Yılmaz	Beklemede	1
	Can Polat	Beklemede	1
	Deniz Erkan	Beklemede	1
	Emre Sarı	Beklemede	1
	Fatih Çelik	Beklemede	1
	Gizem Uslu	Beklemede	1
	Mehmet Kaya	Beklemede	1
	Merve Şahin	Beklemede	1
	Oğuz Kaan	Beklemede	1
	Sibel Aktaş	Beklemede	1
	Buse Özkan	Kabul Edildi	1
	Cem Kılıç	Kabul Edildi	1
	Ezgi Yalçın	Kabul Edildi	1
	Hüseyin Korkmaz	Kabul Edildi	1
	Zeynep Demir	Kabul Edildi	1
	Ali Güneş	Reddedildi	1

Query executed successfully.

8-) SAKLI YORDAMLAR (STORED PROCEDURE)

Saklı yordamlar (stored procedure) önceden tanımlanmış bir görevler dizisidir. Sık sık kullanılan işlemleri sürekli olarak tekrarlamak ve yeniden yazmak yerine o işleme bir görev tanımlayıp daha sonra kullanılması gerektiğinde tek bir komutla çalıştırmaktır. Bu sayede daha kullanıcı dostu bir yapı oluşturulur. Kod tekrarının önüne geçilir ve karmaşık işlemler tek bir kod ile yönetilebilir hale gelir.

Saklı yordamlar birden fazla SQL komutlarını bir arada bulunduran yapılardır. Örneğin ekleme, silme , güncelleme, silme ve listeleme gibi işlemler için kullanılır. Ayrıca parametreler de alabilirler. Saklı yordam yapılarını veri tabanı için önemli hale getiren belli kriterlerde vardır. Bunların başında performans gelmektedir. Karmaşık kod yapısını veri tabanında derlenmiş halde bulundurduğu için performans açısında önemli ölçüde avantaj sağlamaktadır. Bunun yanı sıra güvenlik açısından da önem arz etmektedir. Örneğin veri tabanı kullanıcısına sadece o prosedür yetki olarak verilebilir ve kullanıcının veri tabanında sadece o işlemi yapabilmesi sağlanabilir.

Prosedür -1

İş ilanı ekleme prosedürü. Bir işveren sisteme yeni iş ilanı eklemek istediğinde bu prosedür kullanılacak.

```
CREATE PROCEDURE sp_IsIlaniEkle
    @IsverenID INT,
    @Title VARCHAR(100),
    @Aciklama text,
    @Maas DECIMAL(18,2),
    @Konum VARCHAR(100),
    @CalismaSekli INT,
    @YayinTarihi DATE
AS
BEGIN
    INSERT INTO TBLJOBS(EMPLOYERID, TITLE, JOBDESCRIPTION, SALARY, LOCATION,JOBTYPEID,POSTEDDATE)
    VALUES (@IsverenID, @Title, @Aciklama, @Maas,@Konum,@CalismaSekli, @YayinTarihi);
END;
```

İlgili prosedür direkt bize tablo şeklinde bir çıktı vermiyor. Ancak nasıl kullanılması gerektiği aşağıdaki görselde gösterilmiştir.

```
EXEC sp_IsIlaniEkle
    @IsverenID = 7,
    @Title = 'Ziraat Mühendisi',
    @Aciklama = 'Deneyimli bitki biliminde uzman ziraat mühendisi aranmaktadır.',
    @Maas = 27000.00,
    @Konum = 'Konya',
    @CalismaSekli = 1,
    @YayinTarihi = '2025-05-06';
```

```
%
Messages

(1 row affected)

Completion time: 2025-05-06T23:05:09.3063028+03:00
```

Prosedür -2

Bir işverenin yayınladığı tüm ilanları listeleme. Bu prosedür bir işverenin geçmişte yayınladığı tüm ilanları tablo halinde döndürecek.

```
Create Proc IsvereneAitIlanListeleme
@IsverenID INT
As
BEGIN
Select COMPANYNAME,INDUSTRYNAME,TITLE,SALARY,LOCATION,JOBDESCRIPTION From TBLJOBS j
Inner Join TBLEMPLOYERS e
On j.EMPLOYERID = e.EMPLOYERID
Inner Join TBLINDUSTRIES i
On e.INDUSTRYID = i.INDUSTRYID
Where E.EMPLOYERID = @IsverenID
Order By POSTEDDATE desc
End;
```

```
EXEC IsvereneAitIlanListeleme @IsverenID =11
```

Results Messages

COMPANYNAME	INDUSTRYNAME	TITLE	SALARY	LOCATION	JOBDESCRIPTION
Mimari Yapı İnşaat	İnşaat ve Yapı	Mimar	46500.00	Ankara	İnşaat projelerinde yer alacak deneyimli mimar a...
Mimari Yapı İnşaat	İnşaat ve Yapı	İç Mimar	47000.00	Ankara	Restorasyon ve iç tasarım projelerinde görev alac...

Prosedür -3

Lokasyona göre yapılan başvuruları listeleme. Bu prosedür, şehir isimlerine göre yapılan başvuruları listeler ve o başvuruya hangi iş arayan kullanıcılar başvurmuş bunu gösterir.

```
Create Proc LokasyonaBasvuruListeleme
@LokasyonName varchar(100)
As
Begin
Select APPLICATIONID as 'Başvuru No.',FULLNAME,COMPANYNAME,TITLE,SALARY,LOCATION From TBLAPPLICATIONS a
Inner Join TBLJOBSEEKERS j
On a.JOBSEEKERID = j.JOBSEEKERID
Inner Join TBLJOBS o
On o.JOBID = a.JOBID
Inner Join TBLEMPLOYERS e
On e.EMPLOYERID = o.EMPLOYERID
Where LOCATION=@LokasyonName
End;
```

```
EXEC LokasyonaBasvuruListeleme @LokasyonName = 'İstanbul'
```

Results Messages

	Başvuru No.	FULLNAME	COMPANYNAME	TITLE	SALARY	LOCATION
1	21	Ahmet Yılmaz	FinBank	Finansal Analist	46000.00	İstanbul
2	28	Ali Güneş	CodeMaster Yazılım	Yazılım Geliştirici	49000.00	İstanbul
3	31	Can Polat	Güven Bank	Müşteri Temsilcisi	28000.00	İstanbul
4	33	Oğuz Kaan	TrendMarket Alışveriş	E-Ticaret Yöneticisi	30000.00	İstanbul
5	36	Gizem Uslu	Doğa Tarım A.Ş.	Dijital Pazarlama Uzmanı	38000.00	İstanbul
6	37	Umut Yıldırım	PR Medya Ajansı	Blockchain Uzmanı	46700.00	İstanbul

9-) TRIGGER (TETİKLEYİCİ)

Bir veri tabanı üzerinde gerçekleşen insert, update veya delete işlemleri sırasında otomatik olarak çalışan özel SQL kod bloklarıdır. Bu kod blokları kullanıcı tarafından manuel olarak çağırılmaz, ilgili işlemler sırasında otomatik olarak çalışır. Trigger'lar genellikle iş kurallarını otomatikleştirmek, log tutmak, veri bütünlüğünü sağlamak ya da bildirim gibi işlemleri gerçekleştirmek amacıyla kullanılır.

Bir veri tabanında birden fazla trigger tanımlanabilir. Yani aynı tabloya birden fazla eklenebilir ancak veri tabanının iş tanımına göre tetiklenme sırası önemlidir. Önemli olmasının temel sebepleri arasında triggerların amaçlarının önemli olmasıdır çünkü bu tetikleyicilerin amacı tanımlı iş kurallarını otomatik hale getirmek, log kayıtlarını tutmak ve veri bütünlüğünün sağlanması gibi önemli işlevleri vardır.

Aşağıda proje kapsamında oluşturulan veri tabanımıza ait iki adet trigger örneği bulunmaktadır. Örneklerin neyi amaçladığı ve nasıl çalıştığına dair de detaylı bilgiler verilmiştir.

TRİGGER -1

```
CREATE TRIGGER trg_TekrarlananBasvuruEngelle
ON TBLAPPLICATIONS
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM TBLAPPLICATIONS AS mevcut
        JOIN INSERTED AS yeni
        ON mevcut.JOBSEEKERID = yeni.JOBSEEKERID
        AND mevcut.JOBID = yeni.JOBID
    )
    BEGIN
        RAISERROR('Bu ilana daha önce başvuru yapmışsınız.', 16, 1);
    END
    ELSE
    BEGIN
        INSERT INTO TBLAPPLICATIONS (JOBSEEKERID, JOBID, APPLICATIONDATE, STATUS)
        SELECT JOBSEEKERID, JOBID, APPLICATIONDATE, STATUS
        FROM INSERTED;
    END
END;
```

Triggerin amacını şu şekilde belirtebiliriz ;

aynı iş arayan kullanıcının bir iş ilanına birden fazla başvurmasını engeller. Gerçek hayata dair düşündüğümüzde aynı kişi bir ilana bazen birden fazla kez başvurabiliyor bunun sonucunda gereksiz mail oluyor ve spam başvurular oluşabiliyor. Veri tabanına tanımladığımız trigger sayesinde sistem daha temiz çalışıyor.

Trigger'ın çalışma şekli şu şekildedir :

- INSTEAD OF INSERT: Yeni başvuru eklenmeye çalışıldığında işlemi yakalar ve kendisi karar verir.
- INSERTED: Yeni eklenmek istenen başvurunun verilerini içerir (geçici sanal tablo).
- TBLAPPLICATIONS ile karşılaştırma yaparak aynı kullanıcı aynı ilana daha önce başvurmuş mu kontrol eder.
- Eğer başvurmuşsa: RAISERROR ile hata verilir ve kayıt eklenmez.
- Eğer başvurmamışsa: INSERT işlemi gerçekleştirilir.