**VERİ TABANI - 1**

Bahar Dönemi

1. **MATEMATİKSEL İŞLEMLER İÇİN SQL KOMUTLARI**

**9.1. COUNT**

Veri tabanında istenilen veriden kaç tane olduğunu listelemek için kullanılan komuttur.

**SELECT COUNT (alanadi)**

**FROM tablo\_adi**

**WHERE koşul**;

**Tablo adı:** ogrenci\_listesi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ogrenci\_no | ogrenci\_adi | ogrencisoyadi | ogrenci\_bolum | ogrenci\_ders |
| 1 | Ayşe | Yılmaz | sbst | vtys |
| 2 | Ahmet | Yılmaz | isg | ilkyardım |
| 3 | Ahmet | Onur | sbst | ağ temelleri |
| 4 | Nur | Kaya | makine | elektrik bakımı |
| 5 | Fatma | Kurtuluş | sbst | vtys |
| 6 | Mert | Aslan | makine | elektrik bakımı |
| 7 | Ahmet | Şahin | isg | ilkyardım |

1. SELECT COUNT(ogrenci\_adi)

FROM ogrenci\_listesi;

Çıktı:

|  |  |
| --- | --- |
|  | (No column name) |
| 1 | 7 |

1. SELECT COUNT(ogrenci\_adi)

FROM ogrenci\_listesi;

WHERE ogrencisoyadi = ’yılmaz’;

Çıktı:

|  |  |
| --- | --- |
|  | (No column name) |
| 1 | 2 |

* 1. **GROUP BY**

İstenilen alandaki verileri gruplandırarak o veriden kaçar tane olduğunu göstermek için kullanılır.

**SELECT alanadi1, alanadi2,…**

**FROM tablo\_adi**

**WHERE koşul**

**GROUP BY alanadi1, alanadi2,…**

**ORDER BY alanadi1, alanadi2,….**

1. SELECT COUNT(ogrenci\_adi), ogrenci\_adi

FROM ogrenci\_listesi;

GROUP BY ogrenci\_adi;

Çıktı:

|  |  |  |
| --- | --- | --- |
|  | (No column name) | ogrenci\_adi |
| 1 | 1 | Ayşe |
| 2 | 3 | Ahmet |
| 3 | 1 | Nur |
| 4 | 1 | Fatma |
| 5 | 1 | Mert |

1. SELECT COUNT(ogrenci\_adi), ogrenci\_bolum

FROM ogrenci\_listesi;

WHERE ogrenci\_adi = ‘ahmet’

GROUP BY bolum;

Çıktı:

|  |  |  |
| --- | --- | --- |
|  | (No column name) | ogrenci\_adi |
| 1 | 2 | isg |
| 2 | 1 | sbst |

* 1. **AS**

Çoğunlukla yapılan matematiksel işlemlerde ve diğer sorgulamalarda, tabloda olmayan bir sütunun listelenmesini sağlarken, o sütuna isim vermek için kullanılır.

**SELECT fonksiyon(alanadi1) AS yeni\_isim**

**FROM tablo\_adi**

SELECT COUNT(ogrenci\_no) AS ogrenci\_sayisi

FROM ogrenci\_tablosu;

Çıktı:

|  |  |
| --- | --- |
|  | ogrenci\_sayisi |
| 1 | 7 |

* 1. **HAVING**

Gruplama yapılan sorgularda filtreleme yapmak için kullanılır. WHERE yapısına benzer.

**SELECT alanad(lar)i**

**FROM tablo\_adi**

**WHERE koşul**

**GROUP BY alanad(lar)i**

**HAVING koşul**

**ORDER BY alanad(lar)i;**

1. SELECT COUNT(ogrenci\_no) AS sayi, ogrenci\_bolum

FROM ogrenci\_listesi;

GROUP BY bolum;

HAVING bolum = ‘sbst’ or bolum = ‘isg’;

Çıktı:

|  |  |  |
| --- | --- | --- |
|  | sayi | ogrenci\_bolum |
| 1 | 2 | isg |
| 2 | 3 | sbst |

* 1. **SUM**

İstenilen değerlerin toplanması için kullanılır.

**SELECT SUM(alan\_adi)**

**FROM tablo\_adi**

**WHERE koşul;**

Tablo adi: odevteslim\_listesi

|  |  |  |
| --- | --- | --- |
| ogr\_no | odev\_sayisi | ders |
| 1 | 5 | vtys |
| 2 | 4 | vtys |
| 3 | 2 | vtys |
| 4 | 3 | ağ temelleri |

SELECT SUM(odev\_sayisi) AS toplam, ders

FROM odevteslim\_listesi

WHERE ders = ‘vtys’;

Çıktı:

|  |  |  |
| --- | --- | --- |
|  | toplam | ders |
| 1 | 11 | vtys |

* 1. **AVG**

İstenilen nümerik değerlerin ortalamasının hesaplanması için kullanılır.

**SELECT AVG(alanadi)**

**FROM tablo\_adi**

**WHERE koşul;**

Tablo adi: not\_listesi

|  |  |  |
| --- | --- | --- |
| ogr\_no | ortalama | ders |
| 1 | 75 | vtys |
| 2 | 80 | vtys |
| 3 | 90 | vtys |
| 1 | 50 | ağ temelleri |

SELECT AVG(ortalama) as sinif\_ortalamasi

FROM not\_listesi

WHERE ders = ‘vtys’;

Çıktı:

|  |  |
| --- | --- |
|  | sinif\_ortalamasi |
| 1 | 81.6 |

* 1. **MAX**

Seçilen sütundaki değerlerin en büyüğünün listelenmesi için kullanılır.

**SELECT MAX(alanadi)**

**FROM tablo\_adi**

**WHERE koşul;**

SELECT MAX(ortalama) as max

FROM not\_listesi

WHERE ders = ‘vtys’;

Çıktı:

|  |  |
| --- | --- |
|  | max |
| 1 | 90 |

* 1. **MIN**

Seçilen sütundaki değerlerin en küçüğünün listelenmesi için kullanılır.

**SELECT MIN(alanadi)**

**FROM tablo\_adi**

**WHERE koşul;**

SELECT MIN(ortalama) as min

FROM not\_listesi

WHERE ders = ‘vtys;

Çıktı:

|  |  |
| --- | --- |
|  | min |
| 1 | 75 |

* 1. **TEMEL MATEMATİK**

Sütunlardaki veriler ile matematiksel işlemler yaparak bunların sonuçlarının listelenmesini sağlayabiliriz.

**\***Geçme Notu: vize \* 40/100 + final\*60/100;

SELECT ogr\_no, ders, vize\*40/100+final\*60/100 AS ortalama

FROM notlistesi

WHERE bolum = 1 and bolum = 2;

1. **TARİH VE ZAMAN ALANLARI İÇİN SQL KOMUTLARI**

İstenilen tarih alanından o tarihin gününe, ayına veya yılına göre sorgulama yapılabilir.

**SELECT alanlad(lar)i**

**FROM tablo\_adi**

**WHERE alanadi = ‘YYYY-MM-DD’;**

Tablo adi: kayit

|  |  |  |
| --- | --- | --- |
| no | ad | kayit\_tarihi |
| 1 | Ayşe Yılmaz | 2020-10-05 |
| 2 | Fatma Kaya | 2020-10-20 |
| 3 | Ahmet Maden | 2019-05-30 |

SELECT ad

FROM kayit

WHERE kayit\_tarihi = ‘2020-10-05’;

Çıktı:

|  |
| --- |
| ad |
| Ayşe Yılmaz |

|  |  |
| --- | --- |
| * DAY * MONTH * YEAR | * SYSDATETIME() * CURRENT\_TIMESTAMP * GETDATE() |

1. **MANTIKSAL ALANLAR İÇİN SQL KOMUTLARI**

Evet – Hayır, Geçti – Kaldı, Kadın – Erkek gibi iki seçenekli durumlarda bu tip sorgu yapılabilecek şekilde tablo oluşturulur. Bu şekilde sorgu yapılabilmesi için alan tipinin **bit** olarak tanımlanması gerekir.

Tablo adı: ogrenci

|  |  |
| --- | --- |
| isim | cinsiyet |
| Ayşe | 0 |
| Fatma | 0 |
| Ahmet | 1 |

SELECT \* FROM ogrenci

WHERE cinsiyet = false;

Çıktı:

|  |  |
| --- | --- |
| isim | cinsiyet |
| Ayşe | 0 |
| Fatma | 0 |

**Kaynakça:**

* Özdemir S., “C# .Net 2008 ve ASP.NET”, Nirvana Yayınları, Ankara, 2008.
* Obalı M., “Oracle 10g”, Pusula Yayıncılık, İstanbul, 2007.
* Kaya Y. ve Tekin R., “Veritabanı ve Uygulamaları”, Papatya Yayıncılık, İstanbul, 2007.

Öğr. Gör. Kevser ONUR