

SQL

1-SQL FUNDAMENTALS

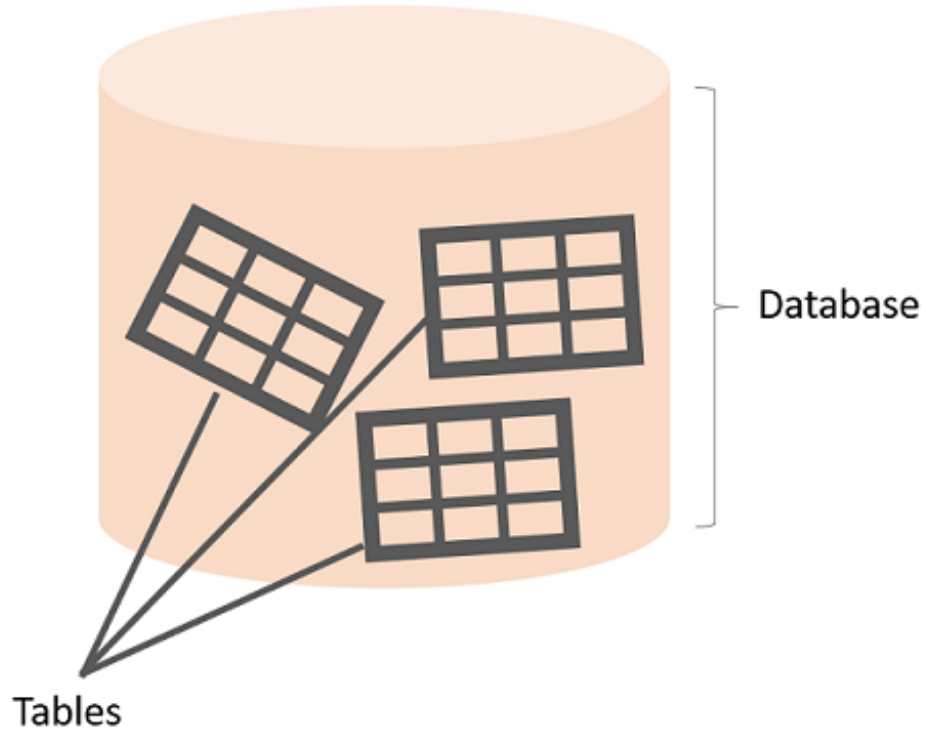
A.Veritabanı nedir?

- Bir veritabanı, tipik olarak bir bilgisayar sisteminde elektronik olarak depolanan yapılandırılmış bir bilgi veya veri topluluğudur.
- Veritabanları hemen hemen her yerdedir. Bankanız, bakkalınız, cep telefonunuzdaki bir uygulama, web sitelerinin tümü verilerinizi takip etmek için veritabanlarını kullanır.
- Örneğin; ticari bir web sitesinde bir ürün sipariş ettiğinizde, siparişiniz bir veritabanında saklanır.
- Bir veritabanı tipik olarak bir **veritabanı yönetim sistemi (DBMS)** tarafından kontrol edilir . Veri ve DBMS ile bunlarla ilişkili uygulamalara, genellikle yalnızca veritabanı olarak kısaltılan bir veritabanı sistemi denir.

A.a-Bir veritabanında neler var?

- Veritabanındaki bilgiler tablolar halinde gruplandırılmıştır. Veritabanındaki bir tabloya **veritabanı tablosu** denir
- **Yapılandırılmış veri** , tablo anlamına gelir. Bunu bir Excel veya Google elektronik tablosu olarak düşünebilirsiniz.

Basic structure of a database



A.b-Bir Tablonun Anatomisi

- Tablo, sütun ve satırlardan oluşur. Sütun, tablo tarafından depolanan bir veri parçasıdır. Bir satır, tek bir şeyin niteliklerini tanımlayan tek bir sütun kümesidir. Sütunların benzersiz bir adı olmalıdır. Sütunlar ve satırlar birlikte bir tablo oluşturur.

💡 İpuçları:

- Veritabanı dünyasında;
- Bir sütuna alan veya nitelik de denir,
- Bir satır aynı zamanda kayıt veya demet olarak da adlandırılır.
- Tabloya ilişki de denir.

emp_id	first_name	last_name	salary	job_title	gender	hire_date
26650	Elvis	Ritter	86000	Sales Manager	Male	11/24/2017
70950	Rodney	Weaver	87000	Project Manager	Male	12/20/2018
97927	Billie	Lanning	67000	Web Developer	Female	6/25/2018
67323	Lisa	Wiener	75000	Business Analyst	Female	8/9/2018
17679	Robert	Gilmore	110000	Operations Director	Male	9/4/2018
76589	Jason	Christian	99000	Project Manager	Male	1/21/2019
51821	Linda	Foster	95000	Data Scientist	Female	4/29/2019
71329	Gayle	Meyer	77000	HR Manager	Female	6/28/2019
49714	Hugo	Forester	55000	IT Support Specialist	Male	11/22/2019
30840	David	Barrow	85000	Data Scientist	Male	12/2/2019

- Sütun başlığı adları küçük harfle yazılır ve tek bir ad içinde boşluk olmamalıdır.
- Bir veritabanı bir veya daha fazla tablodan oluşabilir. Çoğu durumda, birden fazla tablo. Her tablonun çalışanlar, departmanlar veya müşteriler gibi benzersiz bir adı vardır.

B.Veritabanlarının Türü

Tipik olarak, iki ana veritabanı depolama türü vardır:

- **İlişkisel Veritabanı - SQL**
- **İlişkisel Olmayan Veritabanı - NoSQL**
- İlişkisel veritabanı, birbiriyle ilişkili veri noktalarını depolayan ve bunlara erişim sağlayan bir veritabanı türüdür.
- İlişkisel bir veritabanında, tablodaki her satır, anahtar adı verilen benzersiz bir kimliğe sahip bir kayıttır.
- Tablonun sütunları, verilerin özniteliklerini tutar ve her kaydın genellikle her öznitelik için bir değeri vardır, bu da veri noktaları arasındaki ilişkileri kurmayı kolaylaştırır.

- Aslında, 'ilişkisel veritabanı' terimi 1970 yılında IBM'de E. F. Codd tarafından icat edildi, Codd'un 12 kuralından oluşur.
- Bununla birlikte, ilişkisel modelin hiçbir ticari uygulaması Codd'un kurallarının tümüne uymamaktadır, bu nedenle terim yavaş yavaş daha geniş bir veritabanı sistemleri sınıfını tanımlamaya başlamıştır ve en azından:
- **Verileri kullanıcıya ilişkiler olarak sunun (tablo biçiminde bir sunum, yani her tablo bir dizi satır ve sütundan oluşan bir tablo koleksiyonu olarak);**
- **Verileri tablo biçiminde işlemek için ilişkisel operatörler sağlayın.**
- İlişkisel veritabanlarını korumak için kullanılan bir yazılım sistemine **Database Management System (RDBMS)**.
- İşte bazı RDBMS örnekleri Amazon Aurora, Amazon RDS ,Microsoft SQL Server , Oracle Database, MySQL, IBM DB2, Maria DB ,PostgreSQL, SQLite

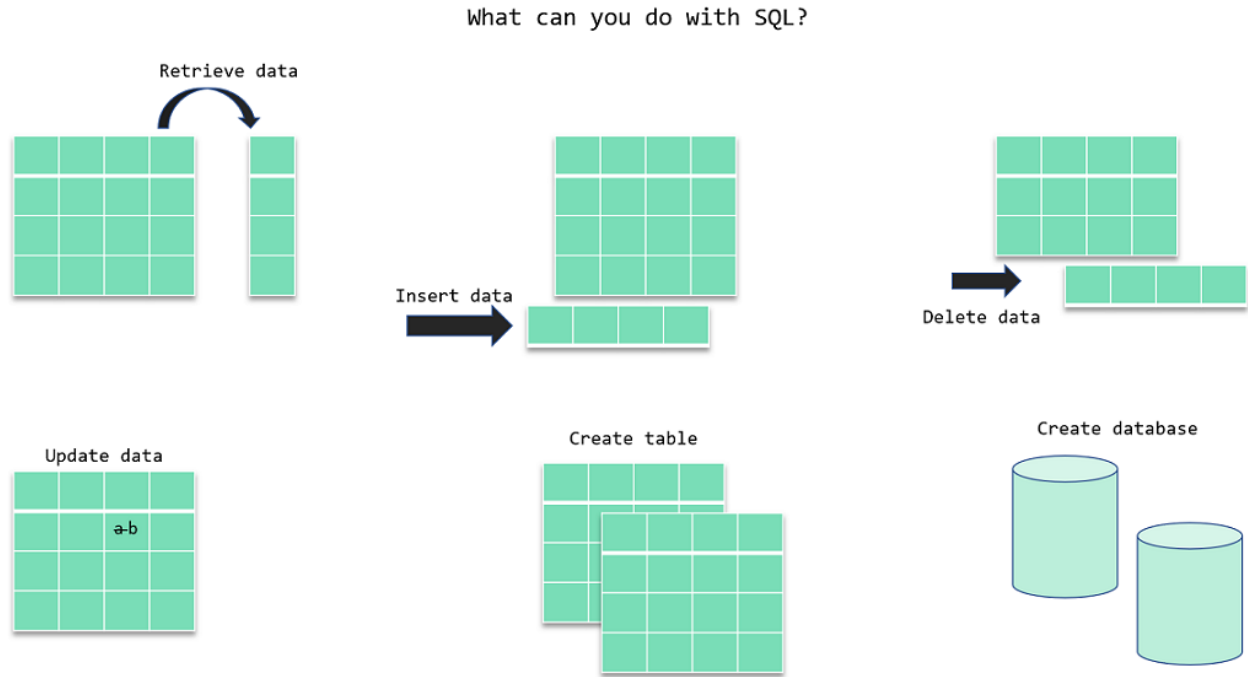
👉 **SQL (Yapılandırılmış Sorgu Dili anlamına gelir), standart İlişkisel Veritabanı Yönetim Sistemi (RDBMS) dili olarak kabul edilir. Bu yüzden genellikle İlişkisel Veritabanını SQL ve İlişkisel Olmayan veritabanını NoSQL olarak adlandırmayı tercih ederiz.**

- SQLite'ı bir RDBMS aracı olarak kullanacağız.SQLite, bir C kütüphanesinde bulunan ilişkisel bir veritabanı yönetim sistemidir. Diğer birçok veritabanı yönetim sisteminin aksine, SQLite bir istemci-sunucu veritabanı motoru değildir. Bunun yerine, son programa gömülüdür.

C. Structured Query Language (SQL)

- **SQL**, Structured Query Language anlamına gelir ve bir veritabanı ile iletişim kurmak için kullanılır. SQL ile veritabanında depolanan verilere erişebilir veya bunları değiştirebilirsiniz.
- Bildirimsel paradigma, nasıl yapacağınızı söylemek zorunda kalmadan istediğinizi söylediğiniz yerdir.
- Prosedürel paradigma ile (JAVA, C'de kullanılır), sonucu almak için kesin adımları belirtmeniz gerekir.
- SQL, yordamdan çok bildirimseldir, çünkü sorgular sonucu üretmek için adımlar belirtmez.
- SQL, en saf haliyle bir programlama dili değil, bir sorgu dilidir. Çünkü döngüleri ve kontrol yapılarını gerçekleştirebilmesi gerekiyor.

- Veri tabanına farklı erişim türleri vardır bunlar ; **Veritabanından veri alınması, Veritabanına yeni verilerin eklenmesi, Veritabanındaki verilerin güncellenmesi, Veritabanından verilerin silinmesi**



- **Query**, veritabanından bilgi alınmasını isteyen bir ifadedir

D-SQL Language Elements

SQL Language Elements

```
SELECT first_name FROM employees;
```

Color coding

Keyword

Identifiers

Terminating Semicolon

Statement

- SELECT ifadesi, bir veritabanından veri seçmek için kullanılır.
- Bu deyim kullanarak tablonun sütunlarından satır alabilirsiniz.
- SELECT ifadesi FROM anahtar sözcüğü ile birlikte kullanılır.
- Bu sorguyu kullanarak, verilen tablodaki bir sütundan/sütunlardan gelen verileri görmek istediğinizi SQL'e açıklarsınız.
- Sorgunun sonucu, sonuç kümesi adı verilen bir sonuç tablosunda saklanır.

D-a- Basic Syntax

- SQL deyimleri SELECT, INSERT, UPDATE, DELETE vb. bir anahtar sözcükle başlar ve tüm deyimler noktalı virgül (;) ile biter.
- Sondaki noktalı virgül, ifadenin tamamlandığını ve yürütülmeye hazır olduğunu gösterir.
- SQL ayrıca büyük/küçük harfe duyarlı değildir; bu, sorgunuzda hem SELECT hem de select kullanabileceğiniz anlamına gelir. SQL için aynı şeyi ifade ederler. Ancak büyük harfle yazmak tercih eddir.
- SQL'de beyaz boşluklar ve boş satırlar yoksayılır. Dolayısıyla, aşağıdaki sorgu kesinlikle doğrudur

```
SELECT                column_name(s)

FROM                  table_name;
```

D-b-Selecting Multiple Columns

column1	column2	column3
column1_value1	column2_value1	column3_value1
column1_value2	column2_value2	column3_value2
column1_value3	column2_value3	column3_value3

- query :

```
SELECT column1, column2 FROM table1;
```

- output :

```
column1      column2
-----
column1_value1 column2_value1
column1_value2 column2_value2
column1_value3 column2_value3
```

- **Yıldız karakteri “*”** SQL'de özel bir anlama sahiptir.
- Tablonuzdan tüm bilgileri almak için SELECT komutundan sonra **bir yıldız (*)** karakteri kullanılabilir.
- query :

```
SELECT * FROM table1;
```

- output :

```

column1      column2      column3
-----
column1_value1 column2_value1 column3_value1
column1_value2 column2_value2 column3_value2
column1_value3 column2_value3 column3_value3

```

E.DISTINCT Clause

- `SELECT DISTINCT column_name(s) FROM table_name;`
- yinelenen satırları tek geçer set fonksiyon gibi
-

student_table

	student	lesson	grade
1	Student1	Mathematics	95
2	Student2	Literature	65
3	Student3	Mathematics	45
4	Student4	Chemistry	85
5	Student5	Physics	70
6	Student6	Physics	75
7	Student7	Mathematics	75

- query :

```
SELECT DISTINCT lesson FROM student_table;
```

- output:

```

lesson
-----
Mathematics
Literature
Chemistry
Physics

```

D-c- WHERE & LIMIT Clauses

D-c-a-WHERE

- WHERE yan tümcesi çoğunlukla SELECT deyimiyle birlikte kullanılır.
- SELECT deyimine ek olarak, DELETE ve UPDATE gibi diğer bazı deyimlerle de kullanılabilir.
- Aşağıdaki örnekte olduğu gibi FROM yan tümcesinden sonra bir sorguda kullanılır.
- `SELECT column_name(s) FROM table_name WHERE condition(s);`

<u>Aa</u> Operator	<u>≡</u> Description
<code>=</code>	Equal to
<code>></code>	Greater than
<code><</code>	Less than
<code>>=</code>	Greater than or equal
<code><=</code>	Less than or equal
<code><></code>	Not equal. This operator may be written as != in some versions of SQL
<u><code>BETWEEN</code></u>	Test if a value is between a certain range of values
<u><code>LIKE</code></u>	Determine if a character string matches a predefined pattern
<u><code>IN</code></u>	Test whether or a value matches any value in a list

student	lesson	grade
Student1	Mathematics	95
Student2	Literature	60
Student3	Mathematics	45
Student4	Chemistry	85
Student5	Physics	70
Student6	Physics	75
Student7	Mathematics	75

- **query :**

```
SELECT * FROM student_table WHERE grade > 70
```

After the execution of the query, you will get a result set like below.

- **output:**

student	lesson	grade
Student1	Mathematics	95
Student4	Chemistry	85
Student6	Physics	75
Student7	Mathematics	75

- ```
SELECT * FROM student_table WHERE lesson = "Mathematics";
```

| student  | lesson      | grade |
|----------|-------------|-------|
| Student1 | Mathematics | 95    |
| Student3 | Mathematics | 45    |
| Student7 | Mathematics | 75    |

- ```
SELECT * FROM student_table WHERE lesson="Mathematics" and grade < 66;
```

student	lesson	grade
---------	--------	-------

Student3	Mathematics	45
----------	-------------	----

D-c-b -LIMIT Clause

- Bir sorgu tarafından döndürülen satır sayısını sınırlar. Sorgunuzun bin satır döndürdüğünü varsayın. Ancak sonuç kümesinde yalnızca ilk 10 satırı görmek istiyorsunuz. Bu gibi durumlarda, istenen çıktıyı elde etmek için LIMIT yan tümcesini kullanırsınız.
- ```
SELECT column_name(s) FROM table_name LIMIT number_rows;
```
- LIMIT ile WHERE'ı da birleştirebiliriz. LIMIT yan tümcesi, WHERE yan tümcesinden sonra gelir.

query:

```
SELECT * FROM student_table WHERE grade > 70 LIMIT 2;
```

output:

| student  | lesson      | grade |
|----------|-------------|-------|
| Student1 | Mathematics | 95    |
| Student4 | Chemistry   | 85    |

## D-d-Order By Clause

- SELECT ifadesi, kayıtları belirtilmemiş bir sırada döndürür.
- Alfabetik veya sayısal sırayla veri almak istemeniz durumunda ORDER BY anahtar sözcüğünü kullanırsınız.
- Varsayılan olarak ORDER BY anahtar sözcüğü, kayıtları artan düzende sıralar.
- Kayıtları azalan düzende sıralamak için DESC anahtar sözcüğünü kullanın. Verileri artan düzende sıralamak için ASC'yi de kullanabilirsiniz. İkisinden birini kullanmak zorundasın.
- `SELECT * FROM employees ORDER BY first_name ASC;` # first name artan düzeyde sıralanır.  
ASC yazmasaydı aynı sonucu verirdi.
- Azalan düzende DESC anahtar sözcüğünü kullanıyorum
- `SELECT * FROM employees ORDER BY first_name DESC;`
- birden çok sütuna göre sıralamak için;
- `SELECT column_name(s) FROM table_name ORDER BY column1 ASC|DESC, column2 ASC|DESC, columnN ASC|DESC;`
- 💡 **İpucu:** Verileri sıraladığınızda, orijinal tablonun sırası etkilenmez. Bir sorgunun bir sonuç tablosu döndürdüğünü önceki derslerden hatırlayın. Böylece orijinal tabloyu değil sonuç tablosunu sıralıyoruz.

## D-e- ORDER BY Clause with WHERE Clause

- Teknik olarak, herhangi bir SQL ifadesi tek bir satıra yazılabilir. Ancak uzun sorgular yazmaya başladığınızda okuması zorlaşacaktır. Bu gibi durumlarda çözüm, kodu yalnızca

yatay olarak değil dikey olarak da düzenlemektir. Buna güzelleştirme denir.

- 

```
SELECT column_name(s)
FROM table_name
WHERE condition
ORDER BY column_name(s) ASC|DESC;
```