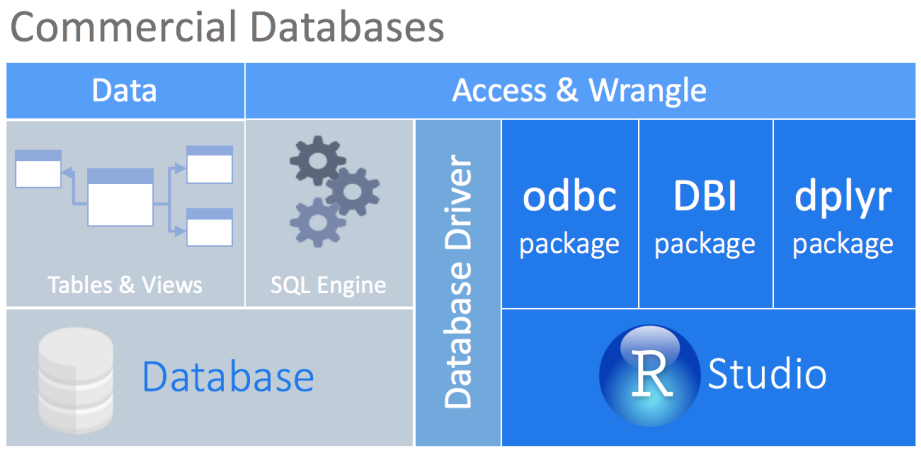
SQLite Veri Tabanına Bağlanma, SQL ve dplyr ile Sorgulama

İçindekiler

- RSQLite paketi ile veri tabanı oluşturma, bağlanma ve tablo sorgulama
- SQLDF paketi ile sorgulama
- DPLYR paketi ile sorgulama




101010
010101
101001

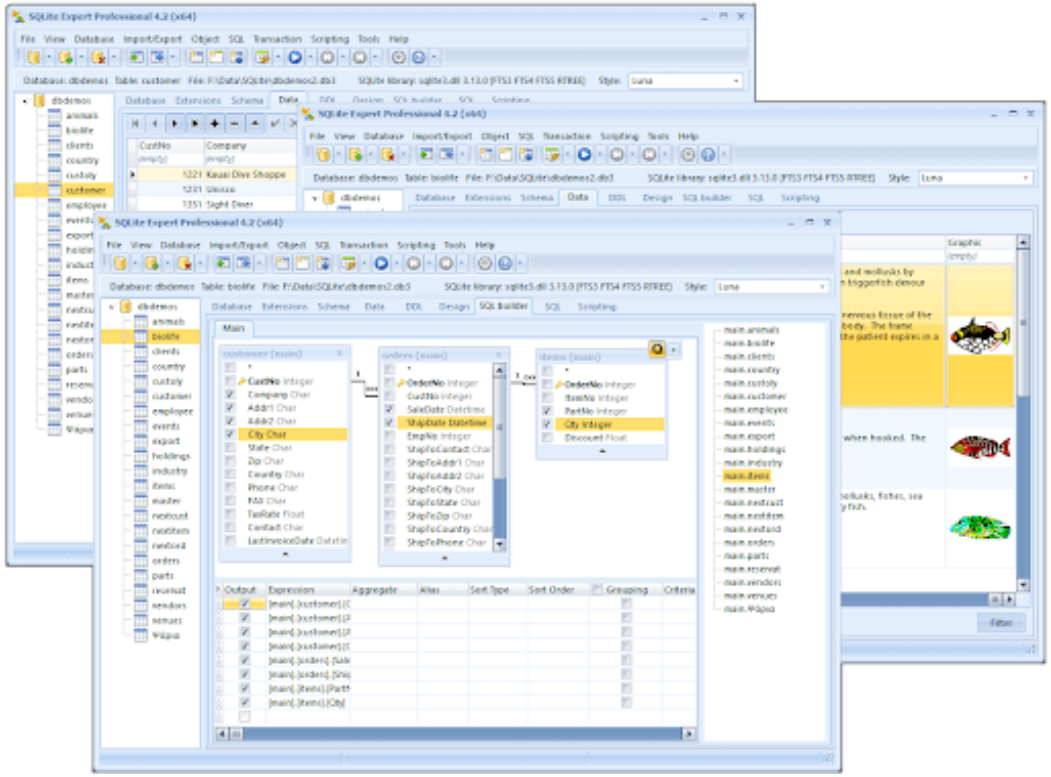
SQL Nedir?

- SQL (Structed Query Language) bir veri tabanı sorgu dilidir. Özellikle bir veri tabanı ile etkileşim için tasarlanmış bir dildir.
- Veri eklemek, çıkarmak, değiştirmek, oluşturmak vb. için söz dizimi (syntax) sunar. Bu amaçla, genellikle bir sunucu veri tabanındaki verilere erişirken kullanılır.
- Verilere erişmek için SQL kullanan çeşitli veri tabanları (SQLite, Microsoft SQL Server, PostgreSQL vb.) vardır. Her bir veri tabanına bağlanma yöntemi farklı olabilir.
- RSQLite, SQLDF, DPLYR paketleri verilere erişmek, temizlemek, filtrelemek ve değiştirmeye izin verir.

NELERE DİKKAT ETMELİ ?

SQL de değişken ve veri tabanı adları küçük harfle yazılır. SQL dizimi(syntax) büyük küçük harf duyarsızdır.

- "SELECT * FROM iris" ✓
- "select * from iris" ✓
- "SELECT * from IRIS" ✗



Değişken ismi "." ile bağlanmışsa tırnak içinde çağırılmalıdır.

- 'SELECT "Petal.Width" FROM iris' 
- 'SELECT Petal.Width FROM iris' 

Hangi Paketlerle Veri Tabanına Bağlanıyoruz?

- DBI, R ve ilişkisel veri tabanı yönetim sistemleri arasındaki iletişim için bir arayüz tanımlar. Bu paketteki tüm sınıflar sanaldır ve çeşitli R / DBMS uygulamaları (SQLite, MySQL, PostgreSQL, MonetDB, vb.) tarafından genişletilmelidir.

```
library(DBI)
# Create an ephemeral in-memory RSQLite database
con <- dbConnect(RSQLite::SQLite(), ":memory:")
```

- RSQLite () ve dbConnect () birlikte bir SQLite veri tabanı dosyasına bağlanmanıza izin verir.

RSQLite

RPostgreSQL

RMySQL

R ortamında RSQLite Paketi

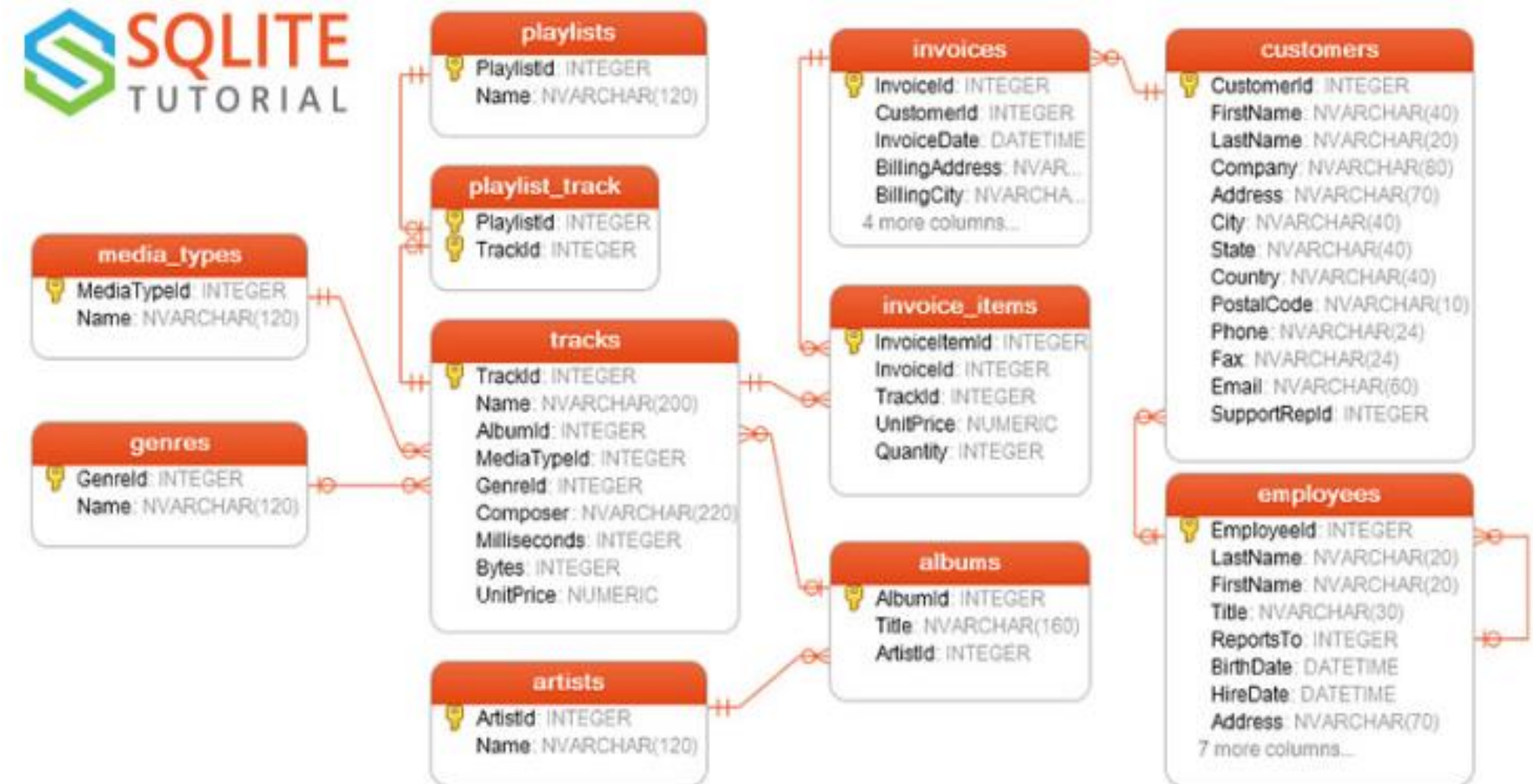
SQLite,dünyada en yaygın kullanılan veri tabanından biri. Android, iPhone ve IOS cihazlarında ve Firefox, Chrome ve Safari web tarayıcılarında bulunur.

- **Install.package("RSQLite")**
- **library(RSQLite)**

SQLite veri tabanına bağlanma

```
> conn<-dbConnect(SQLite(),"chinook.db")
```

conn: dbConnect ile veri tabanına bağlantı nesnesi oluşturulur.



https://www.sqlitetutorial.net/sqlite-sample-database/

DbListTables(): Veri tabanı içindeki tabloları gösterir.
DbListFields(): Tablo içindeki sütunları gösterir.

```
> dbListTables(conn)
 [1] "albums"         "artists"        "customers"      "employees"
 [5] "genres"         "invoice_items"  "invoices"       "media_types"
 [9] "playlist_track" "playlists"      "sqlite_sequence" "sqlite_stat1"
[13] "tracks"

> dbListFields(conn,"albums")
[1] "AlbumId"  "Title"    "ArtistId"
```

dbReadTable(): Belirli bir tabloya ulaşır. Aynı şekilde bir isimle data frame dönüştürülür.

```
> Table_1<-dbReadTable(conn,"genres")
```

	GenreId	Name
1	1	Rock
2	2	Jazz
3	3	Metal
4	4	Alternative & Punk
5	5	Rock And Roll
6	6	Blues
7	7	Latin
8	8	Reggae
9	9	Pop
10	10	Soundtrack
11	11	Bossa Nova
12	12	Easy Listening
13	13	Heavy Metal
14	14	R&B/Soul
15	15	Electronica/Dance
16	16	World
17	17	Hip Hop/Rap
18	18	Science Fiction
19	19	TV Shows
20	20	Sci Fi & Fantasy
21	21	Drama
22	22	Comedy
23	23	Alternative
24	24	Classical
25	25	Opera

dbGetQuery(): İstediğimiz sorgu sütunları gerçekleştirilir.

```
> Table_2<-dbGetQuery(conn, "SELECT * FROM artists LIMIT 25")
```

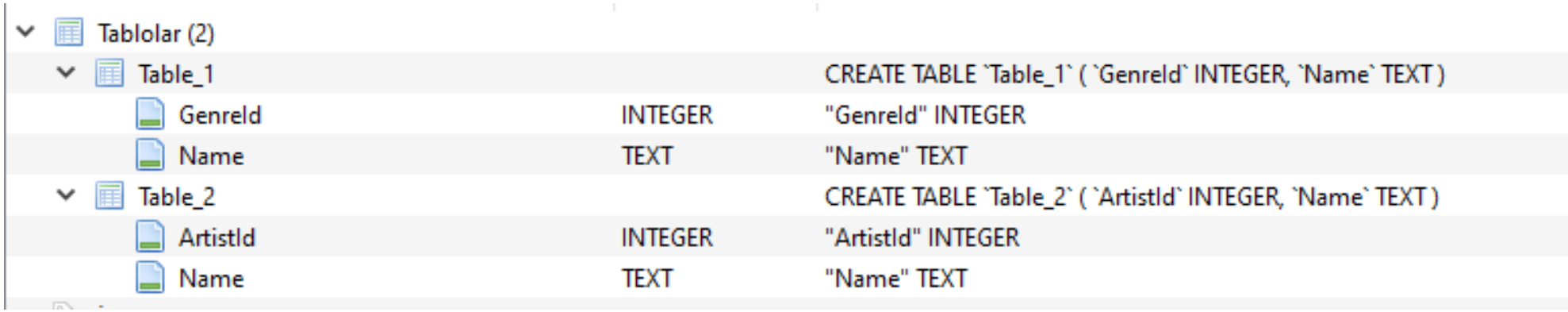
	ArtistId	Name
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette
5	5	Alice In Chains
6	6	Antônio Carlos Jobim
7	7	Apocalyptica
8	8	Audioslave
9	9	BackBeat
10	10	Billy Cobham
11	11	Black Label Society
12	12	Black Sabbath
13	13	Body Count
14	14	Bruce Dickinson
15	15	Buddy Guy
16	16	Caetano Veloso
17	17	Chico Buarque
18	18	Chico Science & Nação Zumbi
19	19	Cidade Negra
20	20	Cláudio Zoli
21	21	Various Artists
22	22	Led Zeppelin
23	23	Frank Zappa & Captain Beefheart
24	24	Marcos Valle
25	25	Milton Nascimento & Bebeto

Seçtiğimiz(Table_1 ve Table_2) tabloları yeni veri tabanı(newdb) içine **dbWriteTable()** ile yazdıralım.

	GenreId	Name
1	1	Rock
2	2	Jazz
3	3	Metal
4	4	Alternative & Punk
5	5	Rock And Roll
6	6	Blues
7	7	Latin
8	8	Reggae
9	9	Pop
10	10	Soundtrack
11	11	Bossa Nova
12	12	Easy Listening
13	13	Heavy Metal
14	14	R&B/Soul
15	15	Electronica/Dance
16	16	World
17	17	Hip Hop/Rap
18	18	Science Fiction
19	19	TV Shows
20	20	Sci Fi & Fantasy
21	21	Drama
22	22	Comedy
23	23	Alternative
24	24	Classical
25	25	Opera

	ArtistId	Name
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette
5	5	Alice In Chains
6	6	Antônio Carlos Jobim
7	7	Apocalyptica
8	8	Audioslave
9	9	BackBeat
10	10	Billy Cobham
11	11	Black Label Society
12	12	Black Sabbath
13	13	Body Count
14	14	Bruce Dickinson
15	15	Buddy Guy
16	16	Caetano Veloso
17	17	Chico Buarque
18	18	Chico Science & Nação Zumbi
19	19	Cidade Negra
20	20	Cláudio Zoli
21	21	Various Artists
22	22	Led Zeppelin
23	23	Frank Zappa & Captain Beefheart
24	24	Marcos Valle
25	25	Milton Nascimento & Bebeto

```
> con<-dbConnect(SQLite(),"newdb")
> dbWriteTable(con, "Table_1", Table_1)
> dbWriteTable(con, "Table_2", Table_2)
> dbListTables(con)
[1] "Table_1" "Table_2"
```



dbExecute() ile çağrılan satır güncellenir.

```
> dbExecute(conn, "DELETE FROM Table_1 WHERE Name= 'Rock'")
[1] 1
>dbExecute(conn, "INSERT INTO Table_1 VALUES (1,'Jazz')")
[1] 1
>dbGetQuery(conn, "SELECT * FROM Table_1")
```

	GenreId	Name
1	2	Jazz
2	3	Metal
3	4	Alternative & Punk
4	5	Rock And Roll
5	6	Blues
6	7	Latin
7	8	Reggae
8	9	Pop
9	10	Soundtrack
10	11	Bossa Nova
11	12	Easy Listening
12	13	Heavy Metal
13	14	R&B/Soul
14	15	Electronica/Dance
15	16	World
16	17	Hip Hop/Rap
17	18	Science Fiction
18	19	TV Shows
19	20	Sci Fi & Fantasy
20	21	Drama
21	22	Comedy
22	23	Alternative
23	24	Classical
24	25	Opera
25	1	Jazz

dbDisconnect() veri bağlantısıyla bağlantı kesilir.

```
> dbDisconnect(con)
```

R ortamında SQLDF Paketi

RSQLite harici çeşitli sqldf paketini kullanarak veri tabanında sorgulama yapmayı sağlar. Veri çerçevesini bir tabloymuş gibi sorgulamanızı sağlar ve genellikle sorguyu bir dize olarak sqldf işlevine geçirmek kadar basittir.

- **Install.package("sqldf")**
- **library(sqldf)**

- R, merkezi olarak konumlandırılmış ilişkisel veri tabanlarından veri almak için SQL kullanımını destekler. Bununla birlikte, R'deki birkaç paket, bu alanın ötesine geçmenize ve verilerin orijinal kaynağından veya nihai hedefinden bağımsız olarak, verileri işlemenin ve analiz etmenin ortasında geçici veri kümeleri oluşturmanıza ve sorgulamanıza olanak tanır.

R da bulunan 2 veri seti kullanıldı.

- data(Orange)
- data(chickwts)

LIMIT ile istenilen gözlem sayısı döndürülür.

```
> sqldf('SELECT * FROM Orange LIMIT 15')
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142
7	1	1582	145
8	2	118	33
9	2	484	69
10	2	664	111
11	2	1004	156
12	2	1231	172
13	2	1372	203
14	2	1582	203
15	3	118	30

WHERE ile şart ifadeleri belirtilir.

```
> sqldf('SELECT age,circumference FROM Orange WHERE Tree <3')
```

	age	circumference
1	118	30
2	484	58
3	664	87
4	1004	115
5	1231	120
6	1372	142
7	1582	145
8	118	33
9	484	69
10	664	111
11	1004	156
12	1231	172
13	1372	203
14	1582	203

ORDER BY ile "ASC(ARTAN)" "DESC(AZALAN)" göre sıralama yapar.

> sqldf("SELECT * FROM Orange ORDER BY age ASC, circumference DESC LIMIT 15")

	Tree	age	circumference
1	2	118	33
2	4	118	32
3	1	118	30
4	3	118	30
5	5	118	30
6	2	484	69
7	4	484	62
8	1	484	58
9	3	484	51
10	5	484	49
11	4	664	112
12	2	664	111
13	1	664	87
14	5	664	81
15	3	664	75

COUNT ile satır(rows) sayısı döndürülür.

```
> sqldf("SELECT COUNT() FROM Orange")
> sqldf("SELECT COUNT(tree)  FROM Orange")
```

	COUNT()
1	35

Eğer Tree değişkenin de bir "NA" değeri olsaydı COUNT(Tree) bu satırları döndürmeyecektir.

AND ve OR ile işlem sırası belirtilir.

> sqldf('SELECT * FROM Orange WHERE (Tree < 3 AND circumference < 200) OR age > 1500')

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142
7	1	1582	145
8	2	118	33
9	2	484	69
10	2	664	111
11	2	1004	156
12	2	1231	172
13	2	1582	203
14	3	1582	140
15	4	1582	214
16	5	1582	177

IN ile istenilen veya istenilmeyen satırlar direk yazılır.

```
> sqldf('SELECT * FROM Orange
WHERE Tree IN (1,4)')
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142
7	1	1582	145
8	4	118	32
9	4	484	62
10	4	664	112
11	4	1004	167
12	4	1231	179
13	4	1372	209
14	4	1582	214

```
> sqldf('SELECT * FROM Orange WHERE Tree NOT IN (2,5)')
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142
7	1	1582	145
8	3	118	30
9	3	484	51
10	3	664	75
11	3	1004	108
12	3	1231	115
13	3	1372	139
14	3	1582	140
15	4	118	32
16	4	484	62
17	4	664	112
18	4	1004	167
19	4	1231	179
20	4	1372	209
21	4	1582	214

LIKE ile herhangi kelimeyle biten ya da başlayan satırlar seçilir veya **NOT LIKE** ile çıkarılır.

"bean" ile biten satırları görüntüler.

```
> sqldf('SELECT * FROM chickwts WHERE feed LIKE "%bean" LIMIT 5')
```

```
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
4    227 horsebean
5    217 horsebean
```

"bean" bulunmayan satırları görüntüler.

```
> sqldf('SELECT * FROM chickwts WHERE feed NOT LIKE "%bean" LIMIT 10')
```

```
   weight    feed
1     309 linseed
2     229 linseed
3     181 linseed
4     141 linseed
5     260 linseed
6     203 linseed
7     148 linseed
8     169 linseed
9     213 linseed
10    257 linseed
```

- "case" ile başlayan satırları görüntüler.

```
> sqldf('SELECT * FROM chickwts WHERE feed LIKE "%case%" LIMIT 10')
```

	weight	feed
1	368	casein
2	390	casein
3	379	casein
4	260	casein
5	404	casein
6	318	casein
7	352	casein
8	359	casein
9	216	casein
10	222	casein

Ortalama, toplam,medyan ve min-max değeri değerleri için sırasıyla;

AVG, SUM **MEDIAN**, **MIN ve MAX** kullanılır.

```
> sqldf("SELECT feed, MIN(weight) AS min_weight FROM chickwts GROUP BY feed")
```

	feed	min_weight
1	casein	216
2	horsebean	108
3	linseed	141
4	meatmeal	153
5	soybean	158
6	sunflower	226

"AS" ile sütuna yeni isim verilir.

```
> sqldf("SELECT feed, AVG(weight) AS avg_weight FROM chickwts GROUP BY feed")
```

	feed	avg_weight
1	casein	323.5833
2	horsebean	160.2000
3	linseed	218.7500
4	meatmeal	276.9091
5	soybean	246.4286
6	sunflower	328.9167

İç içe(Nested) Sorgular İç içe seçimlerin kullanıldığı durumlardır. Değişkenlerin fazla olduğu durumlarda kolaylıkla seçim yapılır.

> sqldf("SELECT COUNT() FROM Orange where age = (select max(age) from Orange)")

	COUNT()
1	5

> sqldf("SELECT * FROM Orange where age = (select max(age) from Orange)")

	Tree	age	circumference
1	1	1582	145
2	2	1582	203
3	3	1582	140
4	4	1582	214
5	5	1582	177

R ortamında DPLYR Paketi İle Sorgulama

Dplyr paketi, dbplyr ile birlikte, yaygın olarak kullanılan açık kaynak veri tabanları SQLite, MySQL ve PostgreSQL ve ayrıca Google'ın bigquery'sine bağlanmayı destekler ve diğer veri tabanı türlerine de genişletilebilir.

- **Install.package("dplyr")**
- **Install.package("dbplyr")**
- **library(dplyr)**
- **library(dbplyr)**

- Dplyr kullanarak veri tabanlarıyla arabirim oluşturmak, SELECT SQL ifadeleri oluşturarak veri kümelerini almaya ve analiz etmeye odaklanır ancak veri tabanın kendisini değiştirmez.
- Dplyr'i UPDATE veya DELETE girişlerine işlev sunmaz. Bu işlevlere ihtiyacınız olursa, ek R paketleri kullanmak gerekir.
- R syntax kullanılır, SQL bilgisi gerekmez.
- Yapmak istenilen her şeyi bir araya toplayıp ardından bir adımda veri tabanına gönderir.

İlk olarak portal_mammals.sqlite dosyasında bulunuan SQLite veri tabanına bağlanıyoruz.

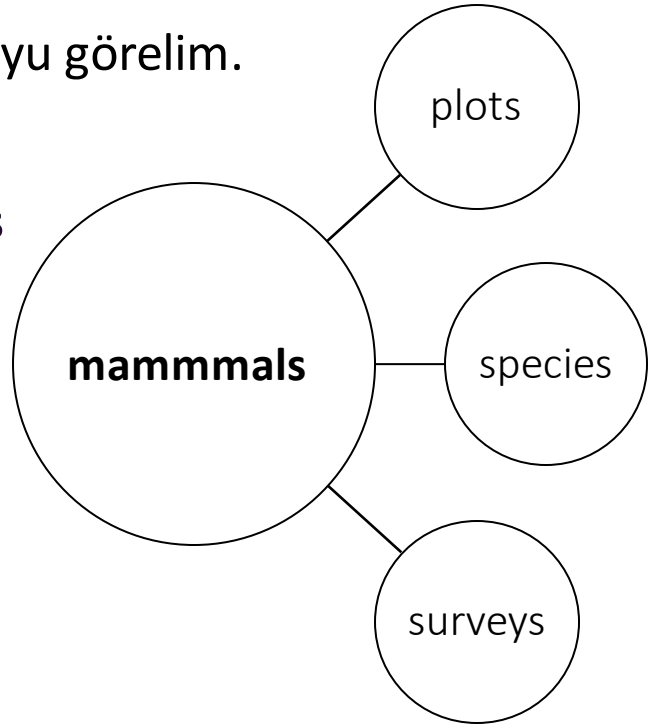
```
dir.create("data_raw", showWarnings = FALSE)
download.file(url = "https://ndownloader.figshare.com/files/2292171",
              destfile = "data_raw/portal_mammals.sqlite", mode = "wb")
```

> mammals <- DBI::dbConnect(RSQLite::SQLite(), "data_raw/portal_mammals.sqlite")

Veri tabanındaki 3 farklı tabloyu görelim.

> src_dbi(mammals),

> tbls: plots, species, surveys



Bir veri tabanı içindeki tabloları çağırmak için dplyr'den **tbl ()** fonksiyonu kullanılır.

Veri tabanını dplyr ile SQL syntax kullanarak survey tablosunu çağıralım.

```
> tbl(mammals, sql("SELECT year, species_id, plot_id FROM surveys"))
```

Veri tabanını dplyr syntax ile survey tablosunu çağıralım.

```
> surveys <- tbl(mammals, "surveys")
        surveys %>%
        select(year, species_id, plot_id)
```

```
   year species_id plot_id
  <int> <chr>         <int>
1  1977 NL                2
2  1977 NL                3
3  1977 DM                2
4  1977 DM                7
5  1977 DM                3
6  1977 PF                1
7  1977 PE                2
8  1977 DM                1
9  1977 DM                1
10 1977 PF                6
# ... with more rows
```

Collect() fonksiyonu tüm satırları toplar.

```
> data_subset <- surveys %>%
      select(species_id, sex, weight) %>%
       collect()
```

	species_id	sex	weight
1	NL	M	*NA*
2	NL	M	*NA*
3	DM	F	*NA*
4	DM	M	*NA*
5	DM	M	*NA*
6	PF	M	*NA*
7	PE	F	*NA*
8	DM	M	*NA*

wing 1 to 9 of 35,549 entries, 3 total columns

Filter() koşul belirlenir.
Select(-) sütun çıkarma yapılır.

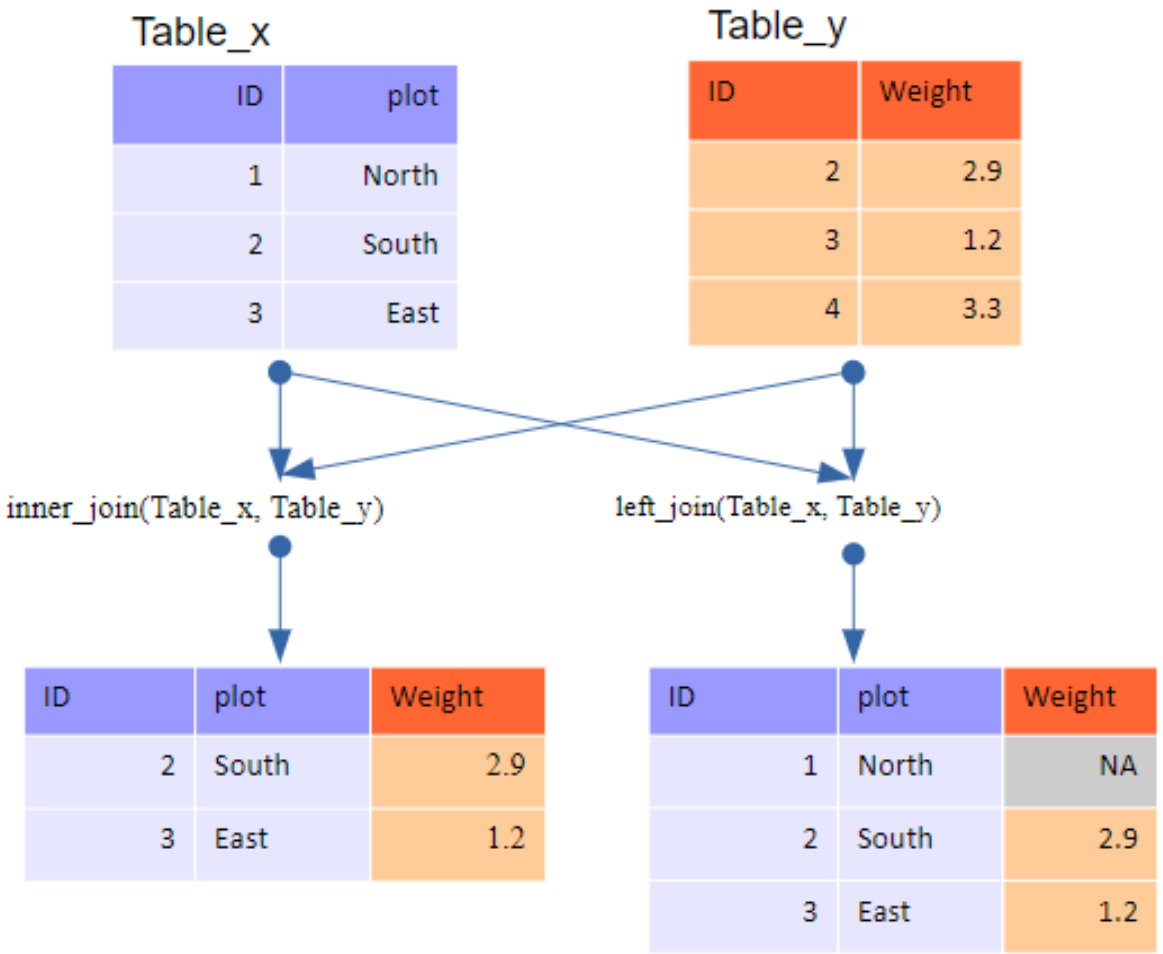
```
> data_subset1<- surveys %>%
    filter(weight < 5) %>%
    select(species_id, weight)
```

```
  species_id weight
  <chr>       <int>
1 PF              4
2 PF              4
3 PF              4
4 PF              4
5 PF              4
6 RM              4
7 RM              4
8 RM              4
9 RM              4
0 RM              4
... with more rows
```

```
> data_subset1 %>%
  select(-weight)
```

```
   species_id
   <chr>
 1 PF
 2 PF
 3 PF
 4 PF
 5 PF
 6 RM
 7 RM
 8 RM
 9 RM
10 RM
# ... with more rows
```

Inner join() ve left join() fonksiyonları ile tablolar birleştirilir.



https://datacarpentry.org/R-ecology-lesson/05-r-and-databases.html

plots ve surveys tabloları ortak sütun olan "plot_id" ile birleştirelim.

```
> plots <- tbl(mammals, "plots")
>  plots %>%
    filter(plot_id == 1) %>%
    inner_join(surveys) %>%
    collect()

Joining, by = "plot_id"
```

	plot_id	plot_type	record_id	month	day	year	species_id	sex	hindfoot_length	weight
1	1	Spectab exclosure	6	7	16	1977	PF	M	14	NA
2	1	Spectab exclosure	8	7	16	1977	DM	M	37	NA
3	1	Spectab exclosure	9	7	16	1977	DM	F	34	NA
4	1	Spectab exclosure	78	8	19	1977	PF	M	16	9
5	1	Spectab exclosure	80	8	19	1977	DS	M	48	NA
6	1	Spectab exclosure	218	9	13	1977	PF	M	13	4
7	1	Spectab exclosure	222	9	13	1977	DS	M	52	NA
8	1	Spectab exclosure	239	9	13	1977	DS	M	48	NA

Left join kullanarak tabloları birleştirelim.

```
> left_join(plots,surveys ) %>%
    filter(plot_type== "Control") %>%
    group_by(plot_id, species_id) %>%
    tally %>%
    collect()
```

	plot_id	species_id	n
	<int>	<chr>	<int>
1	2	NA	3
2	2	AB	14
3	2	AH	7
4	2	BA	1
5	2	CM	1
6	2	CQ	1
7	2	DM	578
8	2	DO	313
9	2	DS	137
10	2	DX	4

... with 190 more rows

Burada **tally ()** sınıfa uygun gözlem sayısını verir.

	species_id	plot_id	n
	<chr>	<int>	<int>
1	NA	2	3
2	NA	4	9
3	NA	8	10
4	NA	11	6
5	NA	12	2
6	NA	14	5
7	NA	17	2
8	NA	22	12
9	AB	2	14
10	AB	4	3

... with 190 more rows

Aynı çıktıyı SQL syntax ile yazalım.

```
> query <- paste("
SELECT a.plot_id, b.species_id,count(*) as count
FROM plots a
JOIN surveys b
ON a.plot_id = b.plot_id
 AND a.plot_type = 'Control'
GROUP BY a.plot_id, b.species_id",
            sep = "" )
tbl(mammals, sql(query))
```

```
   plot_id species_id     n
     <int> <chr>      <int>
 1       2 NA             3
 2       2 AB            14
 3       2 AH             7
 4       2 BA             1
 5       2 CM             1
 6       2 CQ             1
 7       2 DM           578
 8       2 DO           313
 9       2 DS           137
10       2 DX             4
# ... with 190 more rows
```

3 tablo birleştirme

Bunun için species tablosu da çağıralım.

```
> species <- tbl(mammals, "species")
```

```
> unique_genera <- left_join(surveys, plots) %>%
    left_join(species) %>%
    group_by(plot_type) %>%
    summarize(
     n_genera = n_distinct(genus)
     ) %>%
     collect()
```

	plot_type	n_genera
1	Control	24
2	Long-term Krat Exclosure	22
3	Rodent Exclosure	21
4	Short-term Krat Exclosure	24
5	Spectab exclosure	17

```
Joining, by = "plot_id"
Joining, by = "species_id"
> View(unique_genera)
```

Bir sütunda bulunan benzersiz sınıfların sayısını bulmak için **n_distinct ()** kullanılır.

SONUÇ

- R ortamında paketlerle doğrudan veri tabanıyla bağlantı kurup istenilen tablolarda kalıcı değişiklikler yapılabilir. Ya da kendi veri tabanınızı oluşturup ilişkisel tablolar kaydedediliriz.

- Ayrıca SQL tablolarını .csv olarak kaydederek SQLDF veya DPLYR paketleriyle de sorgulamalar gerçekleştirilir.

- RSQLite, SQLDF ve DPLYR paketleri, harici verilere erişmek veya önceden yüklenmiş verileri temizlemek, filtrelemek ve değiştirmek gibi SQL becerilerinizi geliştirmenize izin verir. SQLServer serbest bırakıldığında, içerilen R sürümünü kullanmanın entegrasyon kolaylığı bu sinerjiyi artıracaktır. R'nin diğer veri kaynaklarına erişim esnekliği, yalnızca SQL kullanarak geleneksel olarak mümkün olanın çok ötesinde, veri merkezli ürünler ve hizmetler oluşturmak için yeni fırsatlar sağlayacaktır.

KAYNAKÇA

- SQLDF
- Db.RStudio
- Dplyr
- Chinook.db

TEŞEKKÜRLER