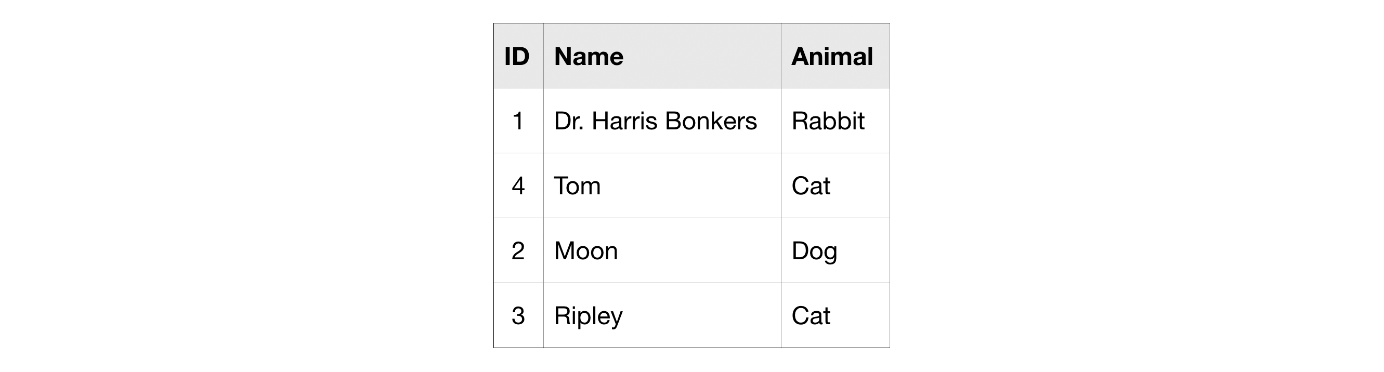
Introduction

Şimdiye kadar birkaç SQL ifadesini nasıl kullanacağınızı öğrendiniz. Örneğin, belirli sütunları bir tablodan çekmek için **SELECT**'i ve belirli kriterleri karşılayan satırları çekmek için de **WHERE**'i nasıl kullanacağınızı biliyorsunuz. Ayrıca, birden çok satırı tek bir grup olarak ele almak için **GROUP BY** ile birlikte **COUNT()** gibi toplama fonksiyonlarını nasıl kullanacağınızı da biliyorsunuz.

Şimdi, **ORDER BY** ifadesini kullanarak sonuçlarınızın sırasını nasıl değiştireceğinizi öğrenecek ve tarihleri sıralamaya uygulayarak popüler bir kullanım durumunu keşfedeceksiniz. Bu eğitimde öğreneceklerinizi göstermek için, bildiğimiz **pets** tablosunun biraz değiştirilmiş bir versiyonuyla çalışacağız.



# ORDER BY[¶](https://www.kaggle.com/code/dansbecker/order-by#ORDER-BY)

ORDER BY genellikle sorgunuzun son cümlesidir ve sorgunuzun geri kalanı tarafından döndürülen sonuçları sıralar.

Satırların ID sütununa göre sıralanmadığına dikkat edin. Aşağıdaki sorguyla bu sorunu hızlıca çözebiliriz.



ORDER BY ifadesi, sonuçların alfabetik sırayla gösterildiği metin içeren sütunlar için de çalışır.



DESC bağımsız değişkenini ('azalan' ifadesinin kısaltması) kullanarak sıralamayı tersine çevirebilirsiniz. Sonraki sorgu, tabloyu Hayvan sütununa göre sıralar ve alfabetik sırada en sonda bulunan değerler ilk döndürülür.



# Dates[¶](https://www.kaggle.com/code/dansbecker/order-by" \l "Dates" \t "_self)

Şimdi, gerçek dünya veritabanlarında sıklıkla karşımıza çıkan tarihlerden bahsedeceğiz. BigQuery'de tarihler iki şekilde saklanabilir: DATE (Tarih) veya DATETIME (Tarih-Saat).

DATE (Tarih) biçiminde önce yıl, sonra ay ve son olarak da gün bulunur. Şöyle görünür:

YYYY-[M]M-[D]D

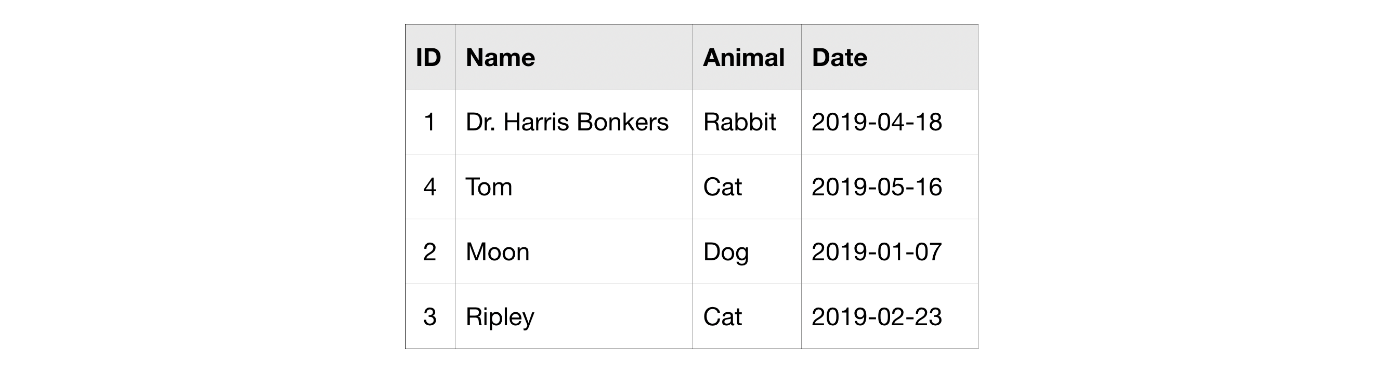
* YYYY: Four-digit year
* [M]M: One or two digit month
* [D]D: One or two digit day

Yani 2019-01-10, 10 Ocak 2019 olarak yorumlanır.

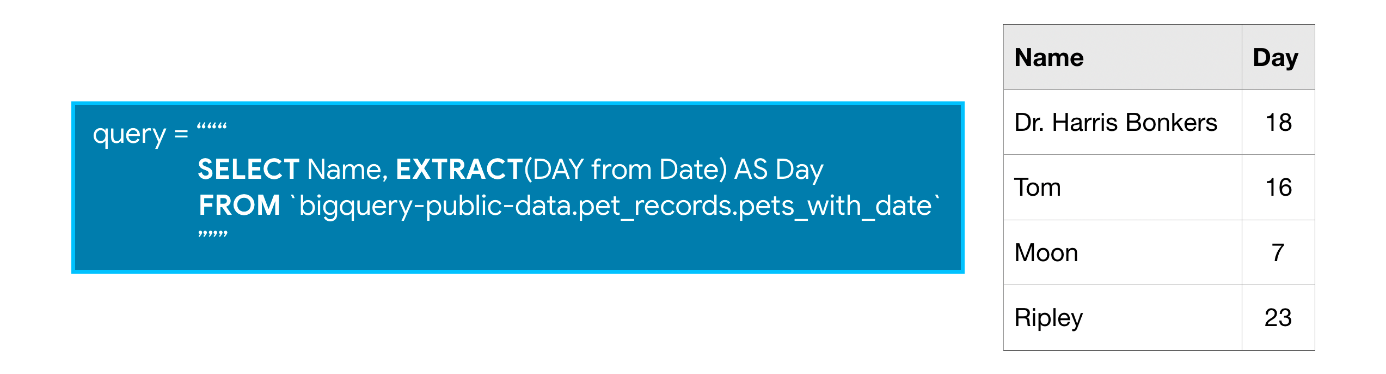
DATETIME biçimi, tarih biçimine benzer... ancak sonuna saat eklenir.

# EXTRACT[¶](https://www.kaggle.com/code/dansbecker/order-by#EXTRACT)

Genellikle bir tarihin yıl veya gün gibi bir bölümüne bakmak istersiniz. Bunu EXTRACT ile yapabilirsiniz. Bunu biraz farklı bir tablo olan pets\_with\_date ile göstereceğiz.



Aşağıdaki sorgu iki sütun döndürür; burada Gün sütunu, pets\_with\_date tablosundaki Tarih sütunundaki her girişe karşılık gelen günü içerir:



SQL tarihler konusunda çok akıllıdır ve hücrenin yalnızca bir kısmını çıkarmakla kalmayıp bilgi de isteyebiliriz. Örneğin, bu sorgu, Tarih sütunundaki her tarih için yalnızca yılın haftasını (1 ile 53 arasında) içeren bir sütun döndürür:



BigQuery'de tarihlerle kullanabileceğiniz tüm fonksiyonları bu dokümantasyonda "Tarih ve saat fonksiyonları" başlığı altında bulabilirsiniz.

# Example: Which day of the week has the most fatal motor accidents?[¶](https://www.kaggle.com/code/dansbecker/order-by#Example:-Which-day-of-the-week-has-the-most-fatal-motor-accidents?)

ABD'de en az bir kişinin hayatını kaybettiği trafik kazaları hakkında bilgi içeren ABD Trafik Ölüm Kayıtları veritabanını kullanalım.

accident\_2015 tablosunu inceleyeceğiz. İşte ilk birkaç satırın görünümü. (İlgili kodu gizledik. Göz atmak için aşağıdaki "Kod" butonuna tıklayın.)

from google.cloud import bigquery

*# Create a "Client" object*

client = bigquery.Client()

*# Construct a reference to the "nhtsa\_traffic\_fatalities" dataset*

dataset\_ref = client.dataset("nhtsa\_traffic\_fatalities", project="bigquery-public-data")

*# API request - fetch the dataset*

dataset = client.get\_dataset(dataset\_ref)

*# Construct a reference to the "accident\_2015" table*

table\_ref = dataset\_ref.table("accident\_2015")

*# API request - fetch the table*

table = client.get\_table(table\_ref)

*# Preview the first five lines of the "accident\_2015" table*

client.list\_rows(table, max\_results=5).to\_dataframe()

Using Kaggle's public dataset BigQuery integration.

/opt/conda/lib/python3.7/site-packages/ipykernel\_launcher.py:19: UserWarning: Cannot use bqstorage\_client if max\_results is set, reverting to fetching data with the tabledata.list endpoint.

Tabloyu kullanarak kaza sayısının haftanın gününe göre nasıl değiştiğini belirleyelim. Şöyle ki:

* **consecutive\_number** sütunu her kaza için benzersiz bir kimlik (ID) içerdiği için ve
* **timestamp\_of\_crash** sütunu kazanın tarihini **DATETIME** formatında içerdiği için,

şunları yapabiliriz:

* **timestamp\_of\_crash** sütunundan haftanın gününü (**aşağıdaki sorguda day\_of\_week** olarak) **EXTRACT** (ayıklar) edebilir, ve
* Her gün için kaza sayısını belirlemek amacıyla **consecutive\_number** sütununu **COUNT** etmeden önce haftanın gününe göre **GROUP BY** yapabiliriz.

Sonra, en çok kazanın olduğu günlerin ilk sırada dönmesi için tabloyu bir **ORDER BY** ifadesi ile sıralarız.

*# Query to find out the number of accidents for each day of the week*

query = """

SELECT COUNT(consecutive\_number) AS num\_accidents,

EXTRACT(DAYOFWEEK FROM timestamp\_of\_crash) AS day\_of\_week

FROM `bigquery-public-data.nhtsa\_traffic\_fatalities.accident\_2015`

GROUP BY day\_of\_week

ORDER BY num\_accidents DESC

"""

Her zamanki gibi, bunu şu şekilde çalıştırıyoruz:

*# Set up the query (cancel the query if it would use too much of*

*# your quota, with the limit set to 1 GB)*

safe\_config = bigquery.QueryJobConfig(maximum\_bytes\_billed=10\*\*9)

query\_job = client.query(query, job\_config=safe\_config)

*# API request - run the query, and convert the results to a pandas DataFrame*

accidents\_by\_day = query\_job.to\_dataframe()

*# Print the DataFrame*

accidents\_by\_day

/opt/conda/lib/python3.7/site-packages/google/cloud/bigquery/client.py:440: UserWarning: Cannot create BigQuery Storage client, the dependency google-cloud-bigquery-storage is not installed.

"Cannot create BigQuery Storage client, the dependency "

| num\_accidents | day\_of\_week |
| --- | --- |
| 0 | 5659 | 7 |
| 1 | 5298 | 1 |
| 2 | 4916 | 6 |
| 3 | 4460 | 5 |
| 4 | 4182 | 4 |
| 5 | 4038 | 2 |
| 6 | 3985 | 3 |

Verilerin, daha fazla trafik kazası olan günlerin ilk sırada göründüğü **num\_accidents** sütununa göre sıralandığına dikkat edin.

**day\_of\_week** sütunu için dönen sayıları gerçek günlerle eşleştirmek için **DAYOFWEEK** fonksiyonu hakkındaki **BigQuery dökümantasyonuna** bakabilirsiniz. Dökümantasyona göre bu fonksiyon, "1 (Pazar) ile 7 (Cumartesi) arasında (bu günler dahil) bir tam sayı" döndürür. Dolayısıyla, 2015 yılında ABD'deki en ölümcül motor kazaları Pazar ve Cumartesi günleri meydana gelirken, en azı Salı günü olmuştur.