Database Research

# Table Of Contents

[Preface 2](#_Toc947116479)

[Which database is most suitable for my application? 2](#_Toc1703721498)

[Where do databases differ? 2](#_Toc282464507)

[How reliable is Natan Silnitsky's database study? 3](#_Toc1275388448)

[Who is Natan Silnitsky? 3](#_Toc1586704803)

[What is Wix.com? 3](#_Toc1769286031)

[Other guides to help me choose a database 3](#_Toc2033319316)

[Who is Leitao Guo? 4](#_Toc10557160)

[Who is Anu Upadhyay? 4](#_Toc849674353)

[Conclusion 4](#_Toc371187458)

[Query Patterns 4](#_Toc152023051)

[What kind of queries do I need? 4](#_Toc1637958589)

[What kind of Database would suite my application in terms of Query Patterns? 5](#_Toc1425140816)

[Key-Value Storage 5](#_Toc540034896)

[Wide-Column Database 7](#_Toc44178527)

[Relational Database 7](#_Toc248604561)

[Document Database 7](#_Toc490575773)

[Search-Engine Database 7](#_Toc440410002)

[Conclusion 7](#_Toc1868468485)

[Consistency 8](#_Toc1173555202)

[Storage Capacity 8](#_Toc184154724)

[Object Storage Service 8](#_Toc302261189)

[Conclusion 8](#_Toc444737904)

[Performance 8](#_Toc1047098470)

[Maturity and Stability 9](#_Toc1784994546)

[Individual Project 9](#_Toc779899697)

[Cost 9](#_Toc285864566)

[Free Database 9](#_Toc1644313439)

[Data Types 9](#_Toc291335659)

[Databases that meet my requirements 9](#_Toc2136056296)

[Differences between SQL Databases 10](#_Toc906342208)

[SQL Dialects 10](#_Toc734498456)

[Community Support 10](#_Toc456500433)

[Testing the community 11](#_Toc231825223)

[PostgreSQL 11](#_Toc1151943771)

[MySQL 13](#_Toc1815310052)

[MariaDB 16](#_Toc1668605263)

[MSSQL 18](#_Toc1819259256)

[SQLite 21](#_Toc1230197779)

[IBM DB2 23](#_Toc156652849)

[Conclusion 24](#_Toc1522200774)

[So, what Database Suits my application the best? 24](#_Toc774577080)

[My Database Choice 24](#_Toc541500875)

# Preface

Currently there are so many different databases. Usually, we tend to choose a database that we have already used before simply because it is the fastest and easiest solution. But usually that isn't the best choice. With this research I want to find out what the most suitable database is for my application.

# Which database is most suitable for my application?

To answer this question, I have multiple sub questions. My goal is to answer all my sub questions to answer this, the main question of the research.

# Where do databases differ?

Since this isn't a new topic, I quickly found a study by Natan Silnitsky. He has written a guide to help you choose a database that suits your application. The main topics in his study are Query Patterns, Consistency, Storage Capacity, Performance, Maturity and Stability, and Cost. For more information per topic, I advise reading his study.

Silnitsky, N. (2022, 5 januari). *How to choose the right database for your service - Wix Engineering*. Medium. Geraadpleegd op 7 juni 2022, van <https://medium.com/wix-engineering/how-to-choose-the-right-database-for-your-service-97b1670c5632>

## How reliable is Natan Silnitsky's database study?

His study seems pretty solid to me, but in the end I'm no database expert. Therefore, I want to research how reliable this source is. To do that, I will search for more guides that help me choose a database, also will I try to find Natan his background.

### Who is Natan Silnitsky?

Natan is a senior backend developer that works for Wix.com since 2019. He works there in a Data Steams Team. He's in charge of libraries, services and tools for Kafka messaging and event driven flows. I have added the link to his LinkedIn for if you want to check yourself.

Natan's LinkedIn: <https://il.linkedin.com/in/natansilnitsky?original_referer=https%3A%2F%2Fwww.google.com%2F>

I've tried to search Natan on YouTube and found a presentation of Wix.com. As the leader of the backend infra team. Natan gets the opportunity to tell the audience more about their data architecture. For me this backs him up pretty good, getting on stage like that is quite an achievement. Important to note is that the video on YouTube is only watched 415 times at this moment (12:47 - 10/06/2022). I assume that is because Wix.com isn't a big company since the channel has 6,63k subscribers on youtube. Also by searching his name on youtube i got lots of other videos about topics like microservices, migrate Bazel from Maven or Gradle.

*Kafka Based Global Data Mesh At Wix- Natan Silnitsky*. (2021, November 8). YouTube. <https://www.youtube.com/watch?v=3kEueGs1gkc>

Natan Youtube Video: https://www.youtube.com/results?search\_query=natan+silnitsky

#### What is Wix.com?

Wix.com is an company located in Tel Aviv. Their industry is Internet Publishing. It's a company that is specialized in the following subjects: Free Website Builder, Web Design, Mobile Websites, HTML5 Websites, Web Apps, Create a Website, Website Templates, Domain Registration, Business Website, Personal Website, and Free Blog. Wix.com consists of over 6000 employees. And is founded in 2006.

Wix.com LinkedIn: https://il.linkedin.com/company/wix-com?trk=public\_profile\_topcard-current-company

### Other guides to help me choose a database

I've looked around on the internet for other sources that should help me choose a suitable database for my application. I found this study by Leitao Guo. Also did I search for a third study, this one is written by Anu Upadhyay.

#### Who is Leitao Guo?

Leitao is a Database and middleware manager at iQIYI.com. I did the same for Leitao as I did for Natan. Leitao has more than 6 years experience at China-Mobile. In his job he has worked with MySQL and NoSQL in that time. After that job he became a Research Assistant for iQIYI.com. He already does this job for over 8 years. But unlike Natan Leitao doesn't work with databases in his current job. Also is it significantly harder to find work of Leitao.

Leitan's Study: <https://en.pingcap.com/blog/how-to-efficiently-choose-the-right-database-for-your-applications/#database-selection-criteria>

Leitan's LinkedIn: <https://cn.linkedin.com/in/leitao-guo-ba6518b?trk=people-guest_people_search-card&original_referer=https%3A%2F%2Fwww.linkedin.com%2F>

#### Who is Anu Upadhyay?

The longer I keep searching for studies, the less known the people get. I hardly found Anu her LinkedIn. But I'm still not 100% if it's her. Let's say it is her. She has once worked with extracting data from an Oracle database. But other than that there isn't much to find.

Annu's Study: <https://www.geeksforgeeks.org/how-to-choose-the-right-database-for-your-application/>

Annu's LinkedIn: <https://in.linkedin.com/in/anu-upadhyay-257709138>

Conclusion

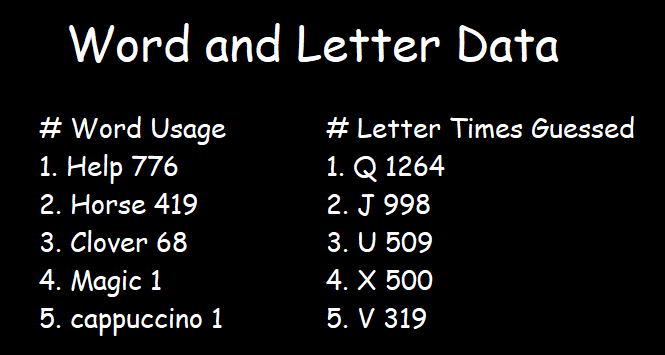
First of all, I doubted Natan when I looked him up. Because he doesn't work in a huge company. The rest I found was actually pretty solid. When I started researching other people that wrote guides, I realised that the people writing them are very hard to find on the internet except Natan. In my eyes Natan has a pretty good status. Therefore, I will use Natan his study for my Research

# Query Patterns

***How complex are your query patterns? Do you just need retrieval by key, or also by various other parameters? Do you also need fuzzy search on the data?***

## What kind of queries do I need?

I'll show a quick example of how my data will look like in my application. This is very prototype-ish but the way the data is visualized is accurate. What you see down below is the amount of times words have been used, as you see its filtered on "Usage". You also see the number of times letters have been guessed, this is filtered on "Times Guessed".



Using this I can already expect what queries I need

* Get a word from a word table
* Save words to a word table
* Update a column in the word table
* Getting the top 100 words ordered on the Usage column
* Getting a letter its usage column
* Update a letter its usage column
* Getting all letters ordered on the Times Guessed column

So long story short, I need basic CRUD operations and I need a ordering option that can order all values, but also the top 100.

## What kind of Database would suite my application in terms of Query Patterns?

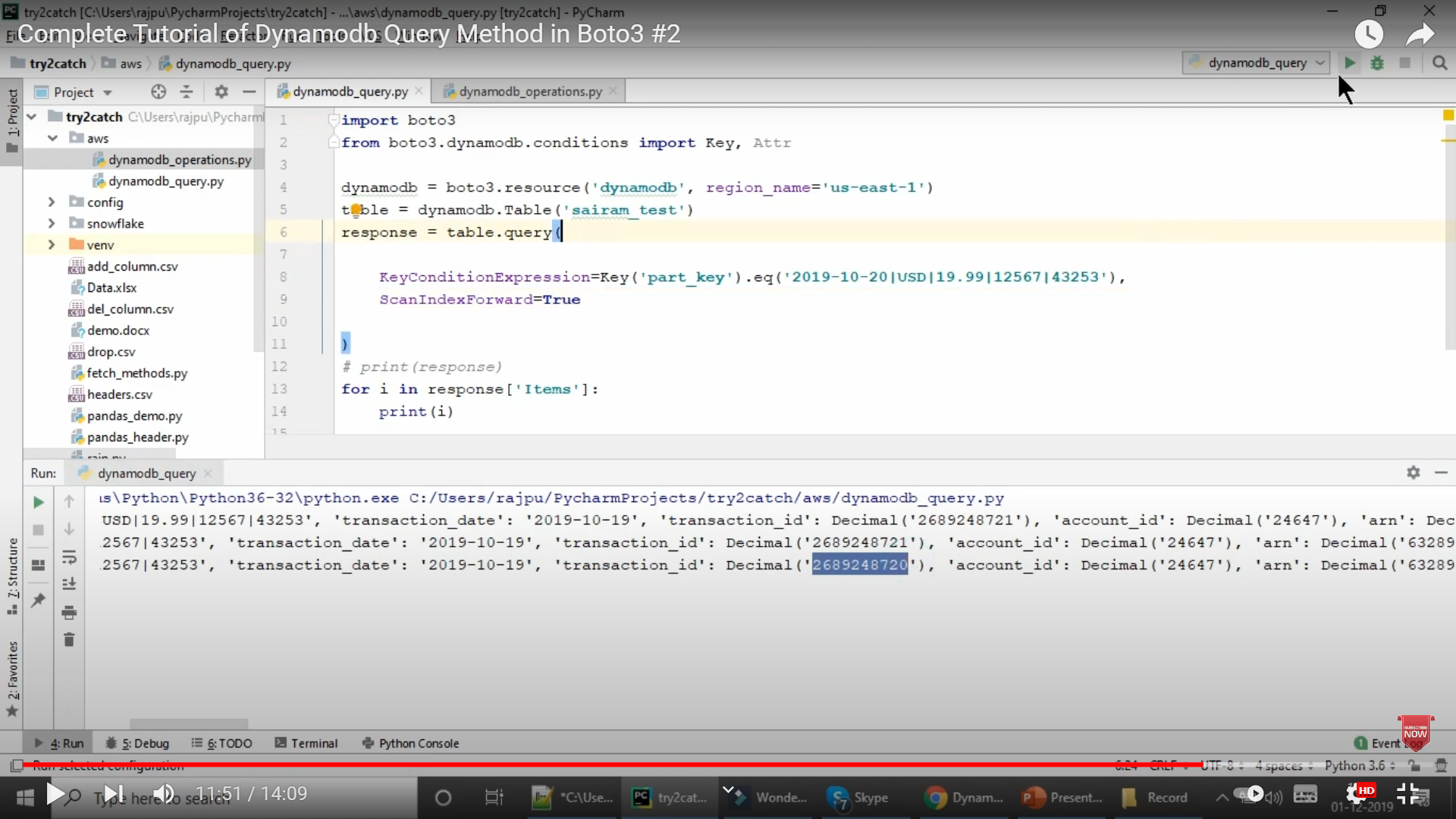
Knowing what we need, we can filter out different types of databases.

### Key-Value Storage

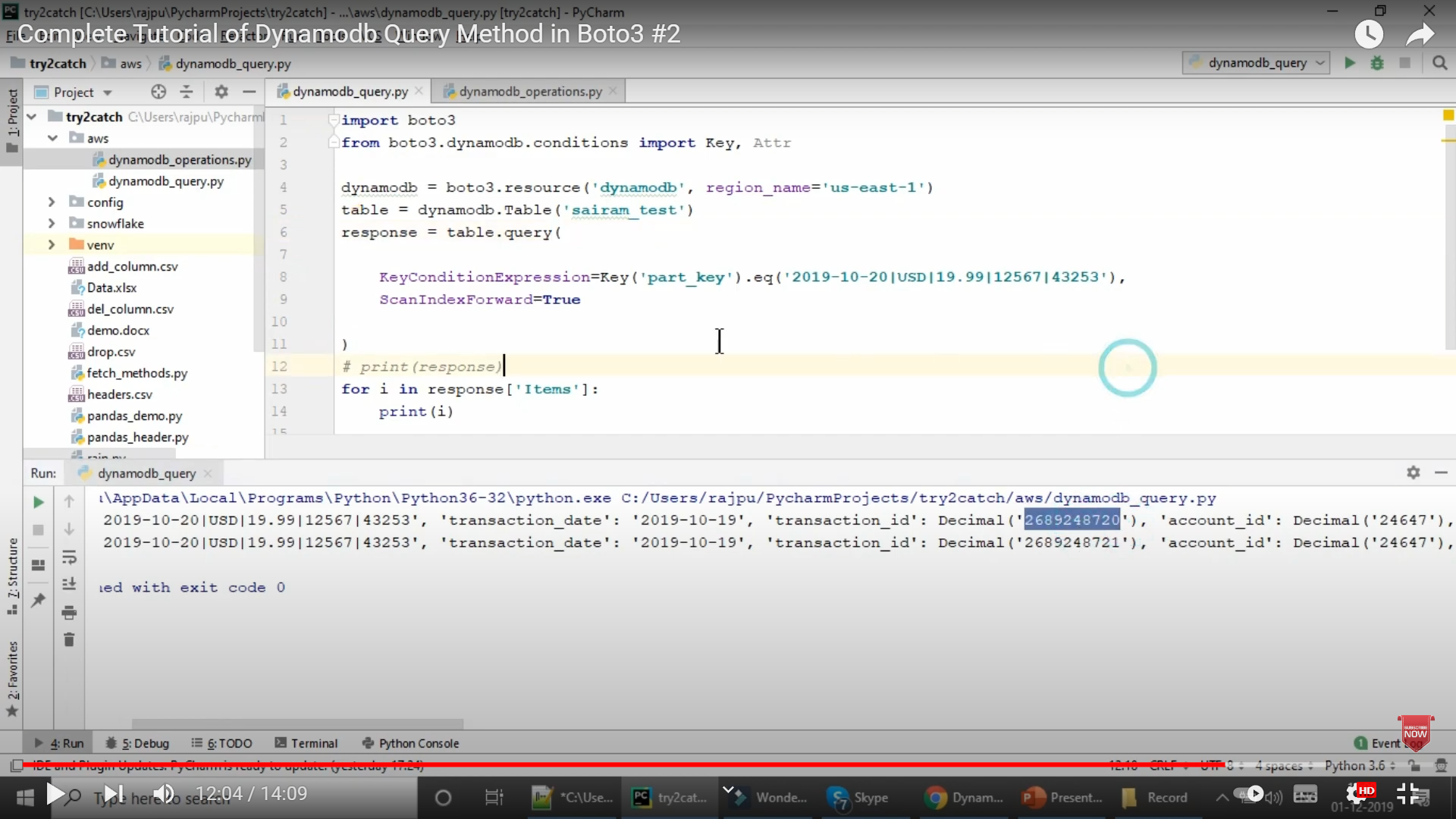
In a key-value storage all the CRUD operations are possible. But I also want to order on a specific column for example "Times Guessed". On stack overflow some guy claims that it should be possible. But it only has 49 upvotes and it doesn't have a green check mark. So, to make sure it's possible I will try it myself.

Source: <https://stackoverflow.com/questions/9297326/is-it-possible-to-order-results-with-query-or-scan-in-dynamodb>

Sadly, I came to the conclusion that I need a credit card to access DynamoDB. And since I don’t have a credit card I can't test it myself. I did find someone doing something similar to what I wanted to try. Here you can hardly see that he put ScanIndexForward=True. The result of this is that it prints the transaction\_id in descending order.



In the following example he puts ScanIndexForward=False, and now it print in an ascending order.



So, getting your data ordered is possible in a key-value storage. Important to note is getting data in a key-value storage is slightly different than I'm used to. For example, if I want to get data from the database, I have to specify a key. I can give every word in my database the same key, but this feels kind of unnatural to me.

### Wide-Column Database

The only real difference between a key-value storage and a wide-column storage, is that a wide-column storage has a better performance when working with bigger datasets. But since my dataset won't be so big, only 2 tables with both 2 columns and lots of rows, a wide-column storage is unnecessary.

Difference: <https://www.techtarget.com/searchdatamanagement/infographic/NoSQL-database-comparison-to-help-you-choose-the-right-store#:~:text=Key%2Dvalue%20stores%20are%20highly,for%20retail%20and%20IoT%20data>.

### Relational Database

In a relational database I can do my basic CRUD operations. Also, can I select any number of rows and order them any way I'd like.

Here's an example that resembles my problem.

Source: <https://stackoverflow.com/questions/12113699/get-top-n-records-for-each-group-of-grouped-results>

### Document Database

If we take MongoDB as example, everything I desire is possible in MongoDB. Here I found an example where they sort on date.

Source: <https://www.educba.com/mongodb-sort-by-date/?source=leftnav>

### Search-Engine Database

Search-engine databases come with fuzzy search. Fuzzy search is a way to find results that are similar to your own results. For example, if I'm looking for "Birate", it might give me "Pirate". I don't have to use anthing like this so this one definitely falls off.

Source: https://aws.amazon.com/nosql/search/#:~:text=A%20search%2Dengine%20database%20is,data%20and%20facilitate%20search%20capability.

Conclusion

If we purely look at query patterns, I'd choose for a relational database or a document database. A key-value storage seems possible, but I might have to use a work around. Same goes for wide-column database. So, these 2 I'd rather not choose. A Search engine database I'll definitely not choose since it comes with one good functionality that I don't need.

# Consistency

***Is strong consistency required (read after write, especially when you switch writes to a different data-center) or eventual consistency is OK?***

If someone has just completed a game of reverse hangman, I'd like to instantly update the data list in-game. If there's a delay between writing and reading data, it might look like my application stopped working. Overall NoSQL databases are less consistent than SQL databases. So earlier it was a choice between Relational databases and Document databases. But now it will be just a battle between relational databases.

What is Data Consistency: https://docs.oracle.com/cd/B10500\_01/server.920/a96524/c21cnsis.htm

# Storage Capacity

*How much storage capacity is needed?*

## Object Storage Service

Object storage services are databases where you can store infinite amounts of data.

## Conclusion

Let’s say my game blows up and everyone on the planet starts playing it. I use an external dictionary API to check the words. This external dictionary does not have an infinite amount of words. This means I don't need an object storage service since my data won't be infinite.

Performance

***What is the needed throughput and latency?***

*"All databases performance degrades as the amount of read/write throughput traffic increases. This is the time when optimizations such as re-indexing and re-sharding of your data come in handy.*

*In case you have very high traffic and require very low latency, Cloud providers solutions like Amazon’s DynamoDB and Google’s Bigtable could be just what you need. As long as your service is deployed on the same data center as the database, you can enjoy latencies that are under 10ms. The downside is of-course the $ cost." - Natan Silnitsky*

My project doesn't need to reach a certain performance. As Natan said, it also costs money, and since my project has no budget this would be impossible anyway.

# Maturity and Stability

***If you choose self-hosted deployment, How much experience does your DBA team have with this technology, how mature is it?***

## Individual Project

Since I'm on my own, there is no team experience that I have to take into account. This gives me the freedom to choose any database.

# Cost

***If you choose a managed cloud solution, What are the costs? What are its limitations?***

## Free Database

I’m looking for a free database since there isn’t any budget. This make every database that is paid fall off.

# Data Types

Natan doesn't say a thing about data types, but in my experience, there are databases like PostgreSQL that come with lots of unique types for example, array, geometry point and even custom types. In my application I only need to save strings and numbers. So, anything that comes extra is unnecessary.

PostgreSQL: https://www.postgresql.org/about/

# Databases that meet my requirements

If we gather all information I've researched we want a,

* Order by functionality
* Select x amount of rows
* SQL based
* No infinite storage necessary
* No better than usual performance necessary
* I don't rely on others' experiences while choosing a database
* Free to use
* String and number datatypes

If we look at this list, this basically sums up every SQL database. Since all SQL databases are so similar I have to find something that separates them.

List with SQL servers: https://learnsql.com/blog/most-popular-sql-databases-2020/

## Differences between SQL Databases

I found a blog from someone named Mona Khalil. I looked up Mona on LinkedIn and found that she is a Data Science Professional. She has experience with working with data and researching so this blog feels very reliable.

In her blog she compares SQL, PostgreSQL, MySQL and SQLite. By reading what she compares, I can use that to compare a lot more databases with each other. So I don’t make a choice between 4 random sql databases, but a choice between a whole lot of databases.

What seperates SQL databases: <https://www.datacamp.com/blog/sql-server-postgresql-mysql-whats-the-difference-where-do-i-start>

Mona Khalil LinkedIn: <https://www.linkedin.com/in/mona-khalil>

### SQL Dialects

I found a blog wherein a software engineer explains the way different SQL dialects structure their queries.

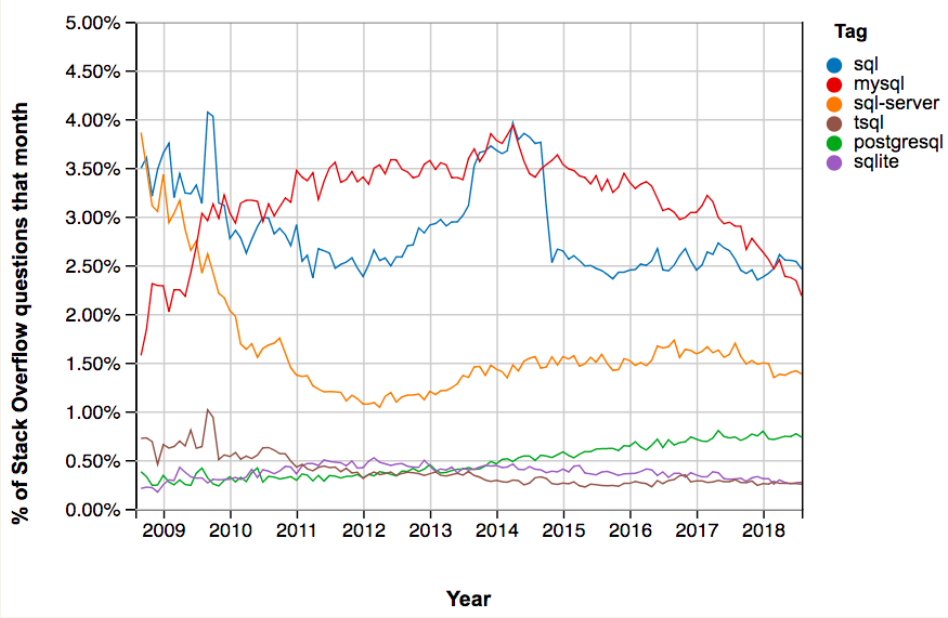
Not all SQL databases use the same SQL language. There are so called different SQL dialects. This means they look similar but are slightly different. I would like to use a dialect I have already used before since this might save up time. The dialect I'm used to looks like the following, SELECT … FROM … WHERE …. But for example the dialect PostgreSQL uses looks like this, DECLARE … BEGIN … END [ label ]. If I look at this, it looks slightly similar but I would have no idea what the label should mean. So I would rather a select, from, where dialect.

Dialects: <https://arctype.com/blog/sql-dialects/>

His LinkedIn: <https://www.linkedin.com/in/everettberry?original_referer=https%3A%2F%2Fwww.google.com%2F>

### Community Support

In Mona's blog, she shows a graph of how often questions get asked about the database. I'll copy paste it into my research.



The ones with lots of questions have a bigger community. So, if I get stuck, the chance of finding the answer online is bigger on these. I decided to put this on the test my way. I will try to google 5 questions on different databases. Like this we will which ones I cant find the answer for online. Like this I test which database has the bigger community.

#### Testing the community

To test the community I first need a few SQL problems. So I looked up 'hard SQL exercises' and found the following.

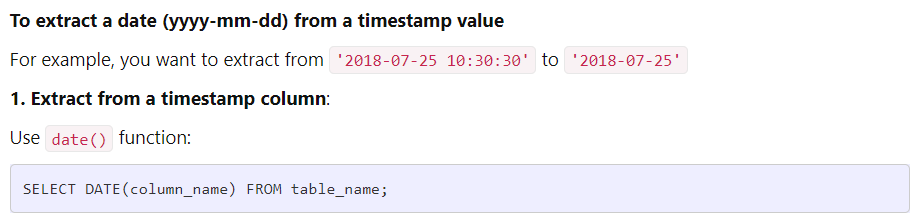
1. Convert datetime to date
2. Get data from multiple tables
3. Get the median
4. Get the average
5. Round to 3 decimals

Important to note these are all fractions of the 'hard' queries I found. I could also search for 'getting the top 5% rounded to 3 decimals' but the chances of finding exactly that is really low, that’s why I chose 5 exercises that serve one purpose.

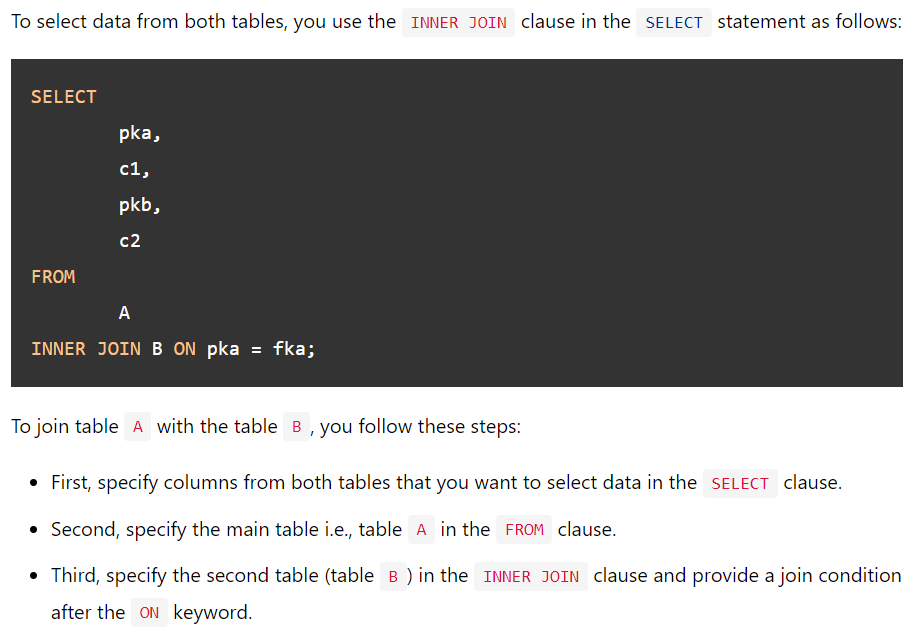
I left out paid SQL databases (snowflake) and less accessible databases (teradata) for obvious reasons.

PostgreSQL

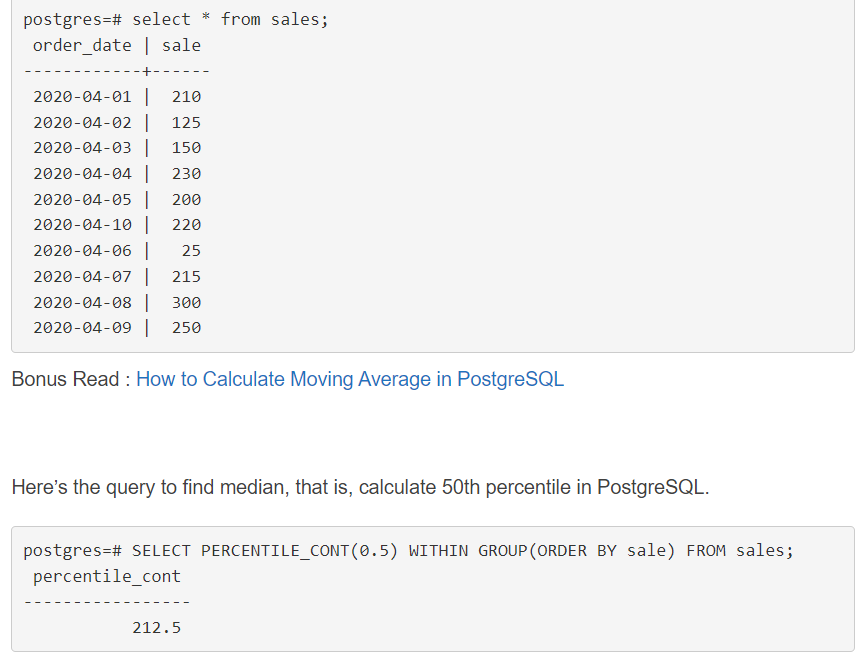
Converting datetime to date



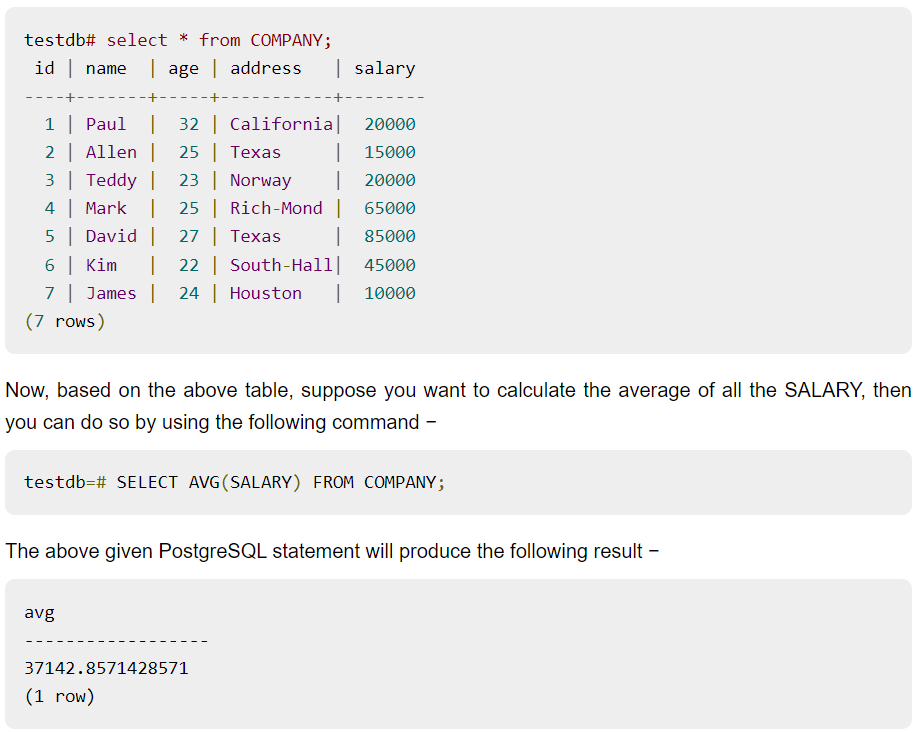
Getting data from multiple tables



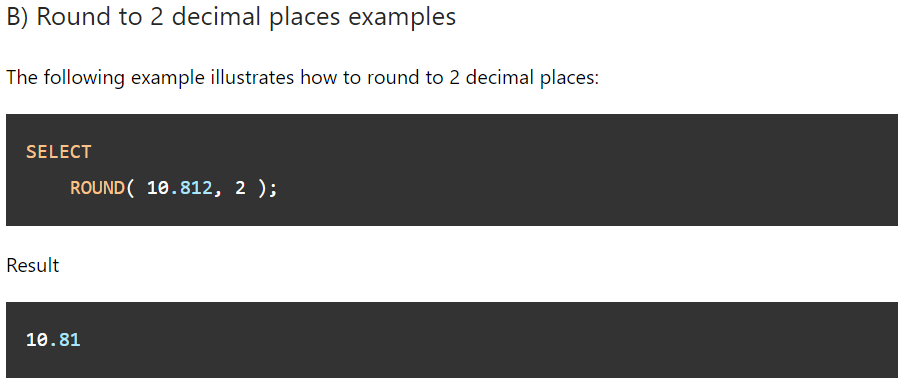
Getting median



Getting average



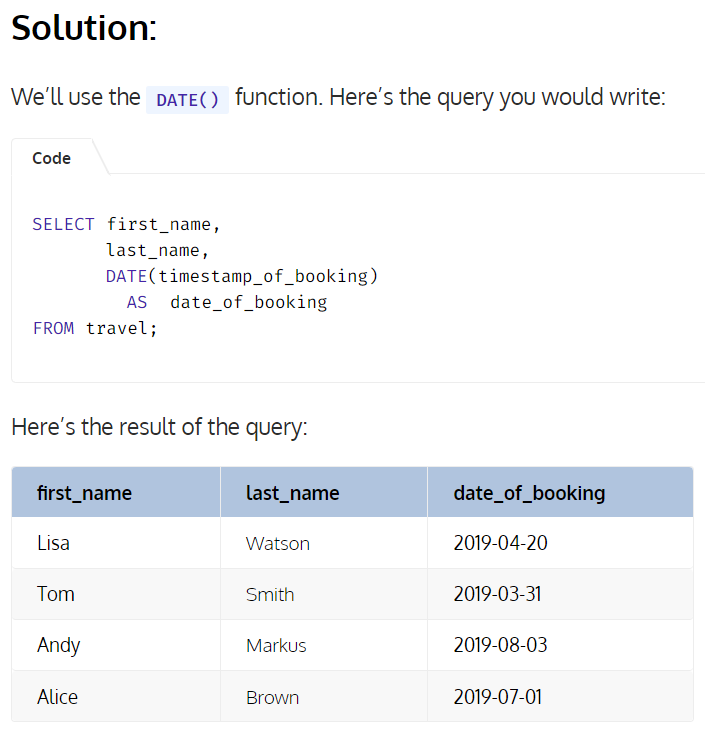
Round to 3 decimal places, here we found an example to round to 2 decimal places but if we simply replace the 2 in the round function we get 3 decimals.



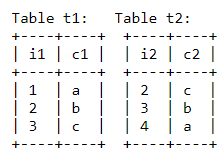
PostgreSQL scores 5/5

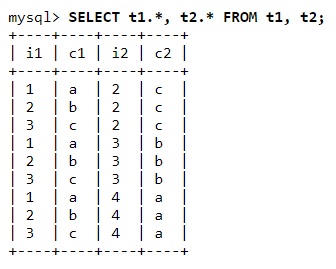
MySQL

Convert datetime to date

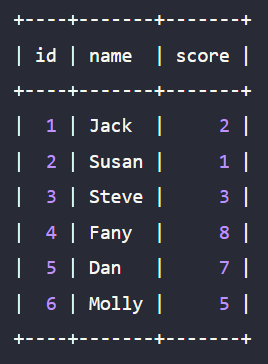


Get data from multiple tables

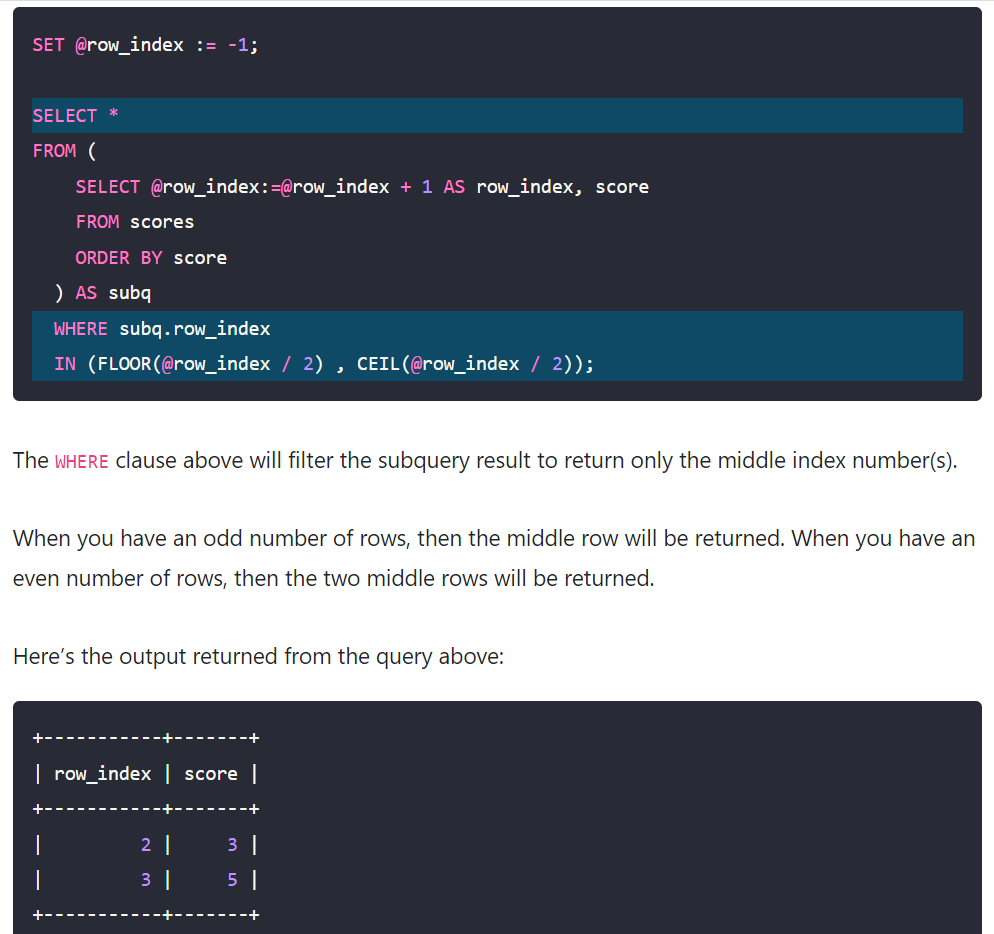




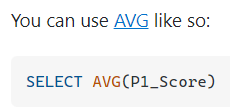
Get the median, As you can see its quite a hassle in MySQL to calculate the median. But since I don't ever have to calculate the median in my application it doesn't matter.



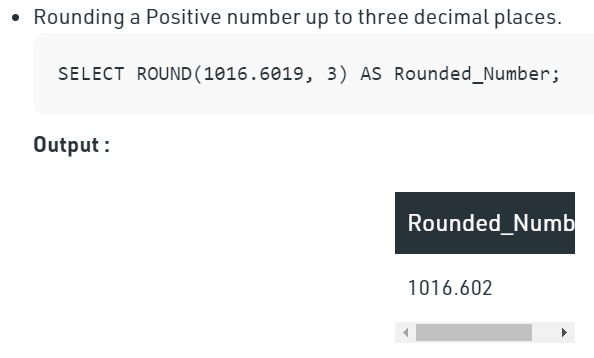




Get the average



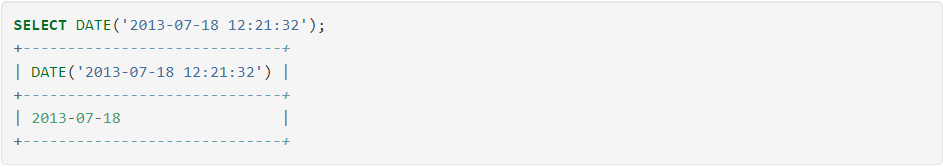
Round to 3 decimals



MySQL scores 5/5

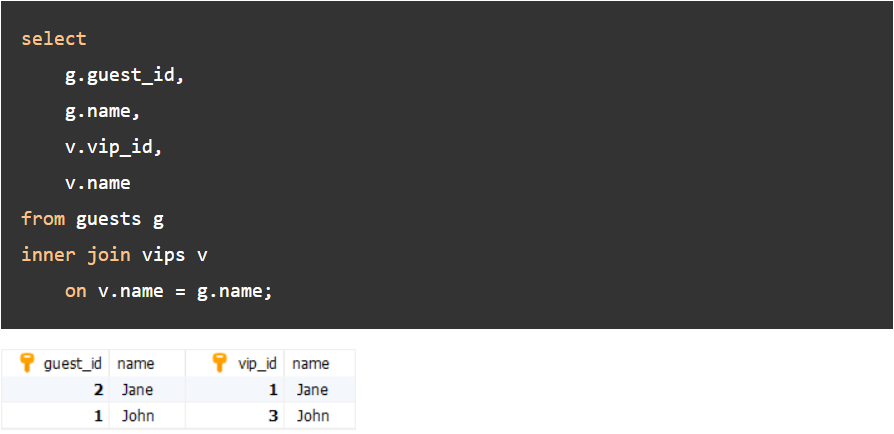
##### MariaDB

Convert datetime to date

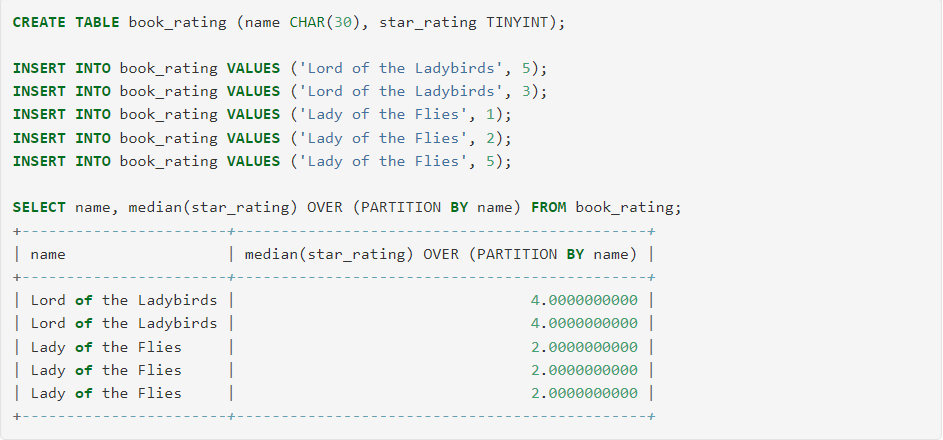


Get data from multiple tables

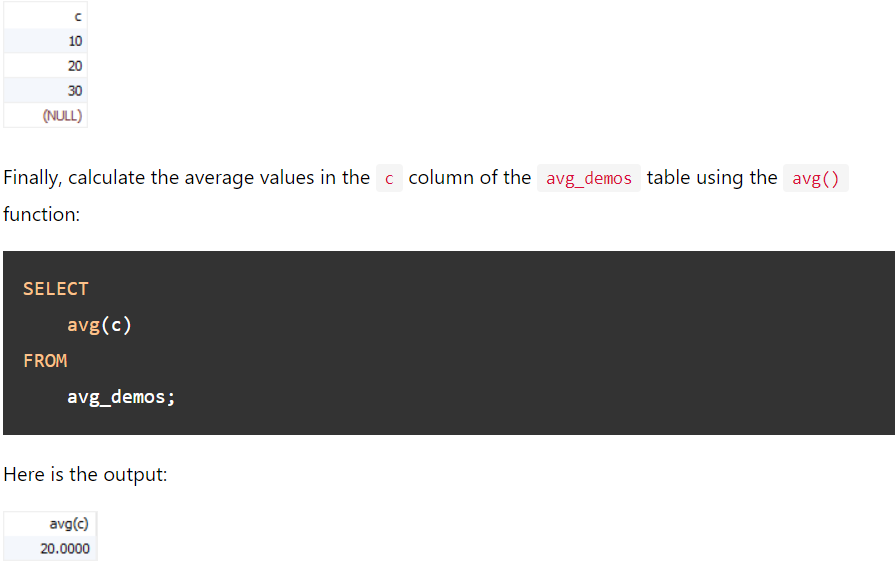




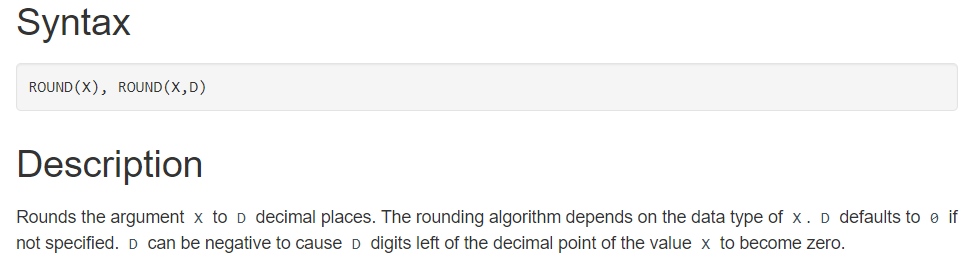
Get the median



Get the average



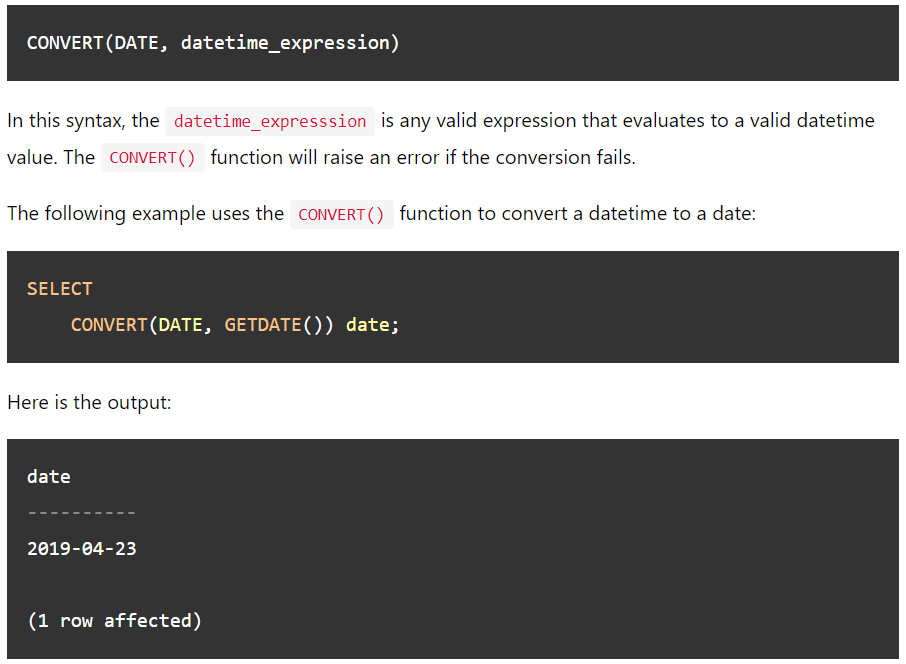
Round to 3 decimals, no example but understandable.



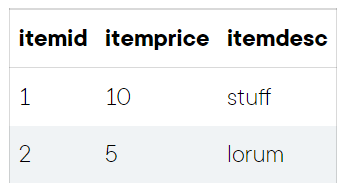
MariaDB scores 5/5

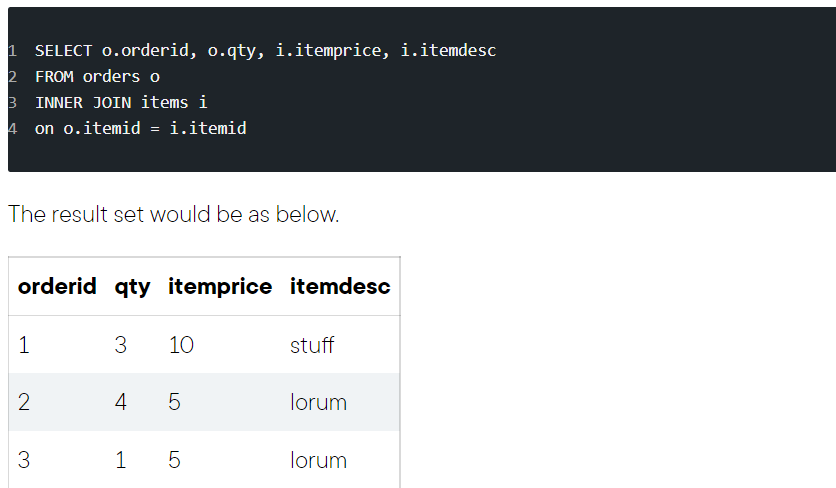
MSSQL

Convert datetime to date

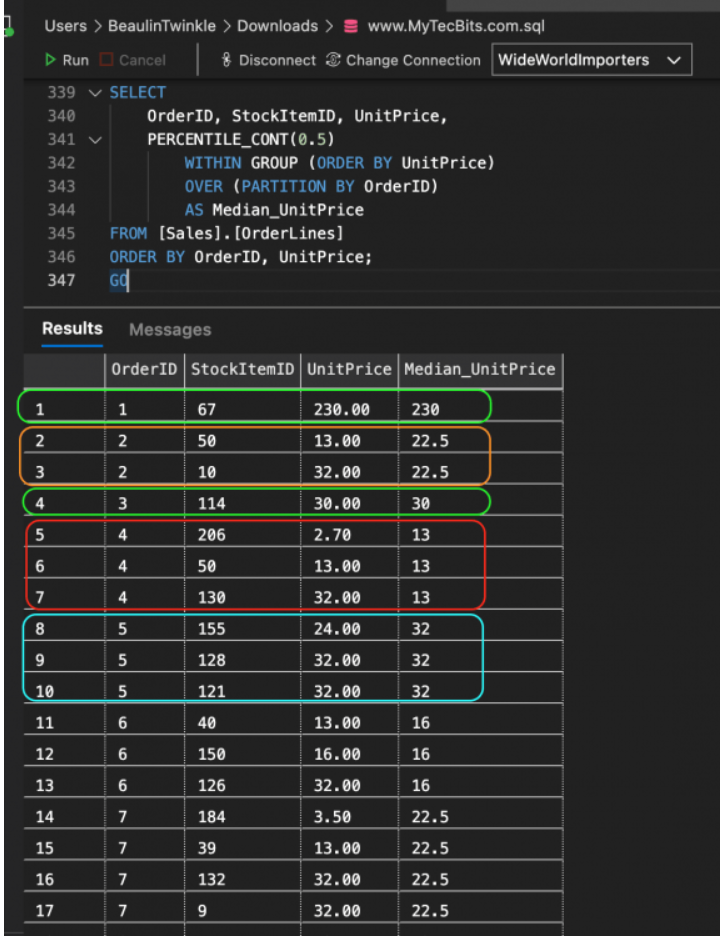


Get data from multiple tables

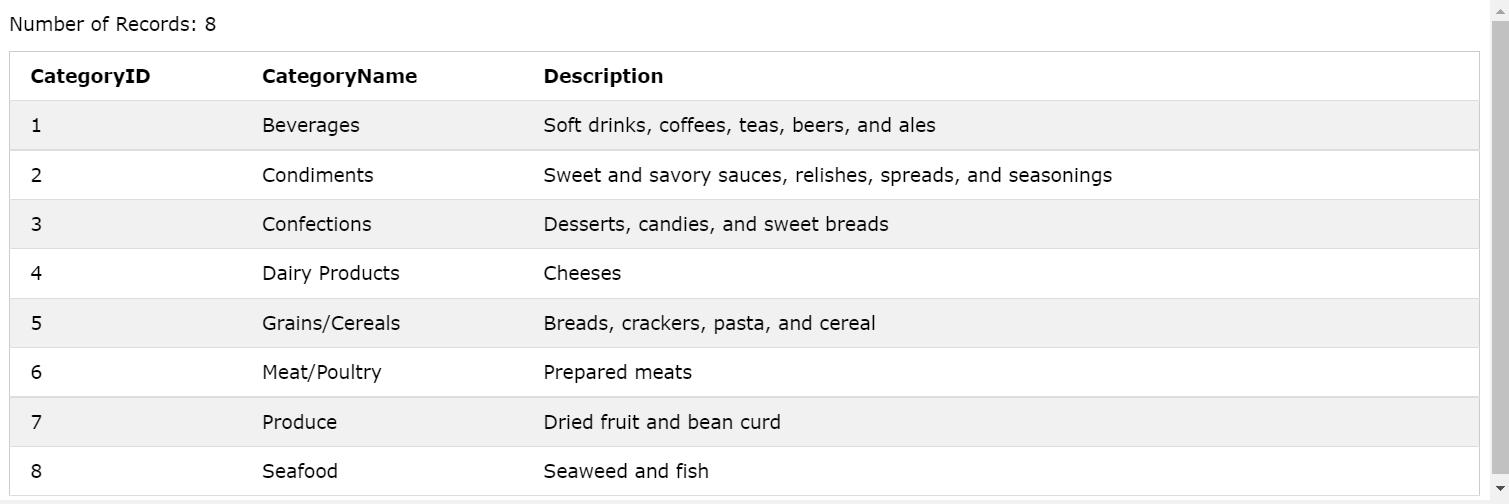


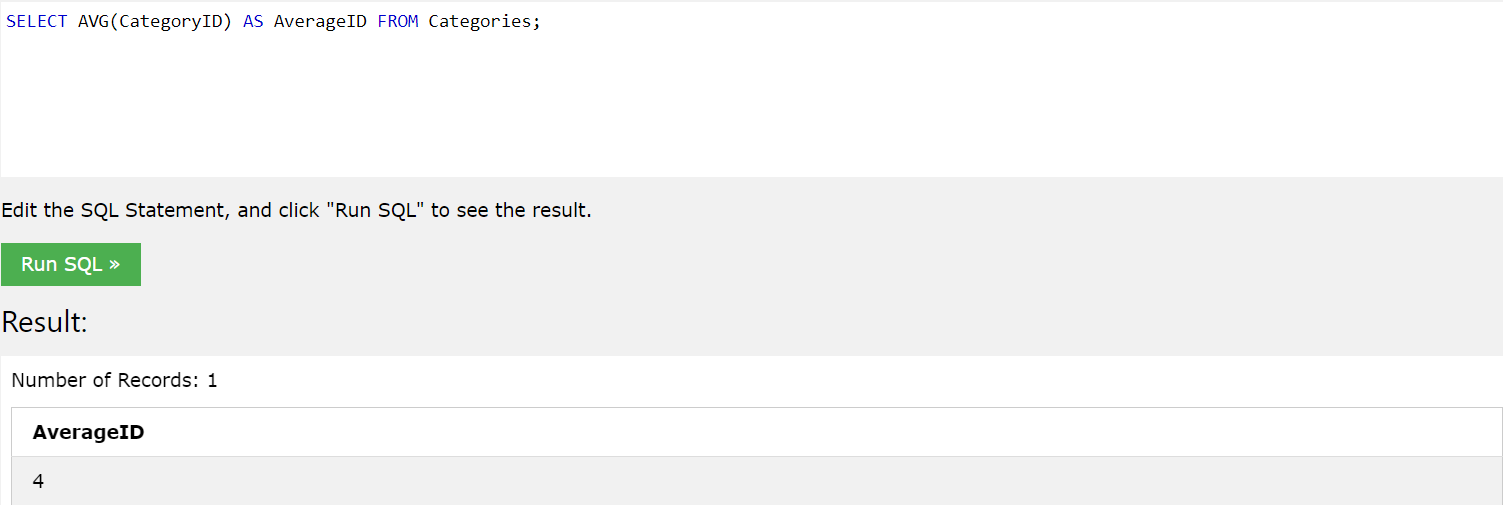


Get the median, here you can see the median per id.

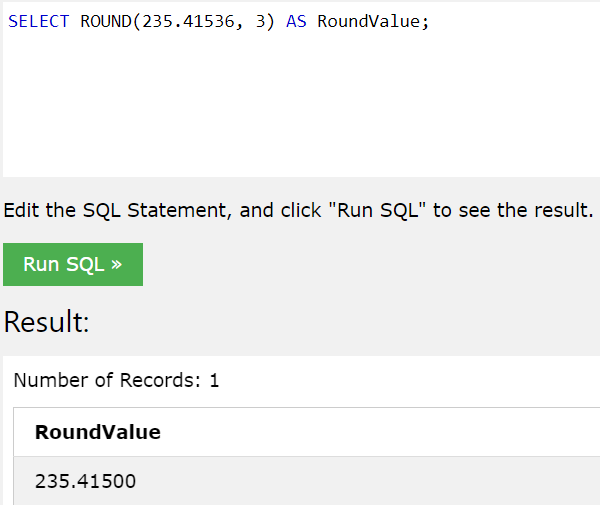


Get the average





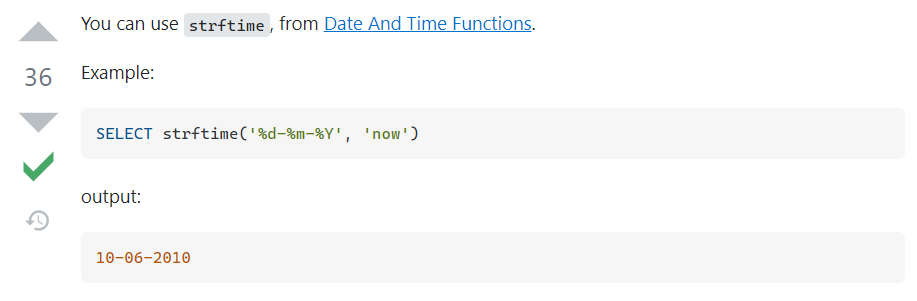
Round to 3 decimals



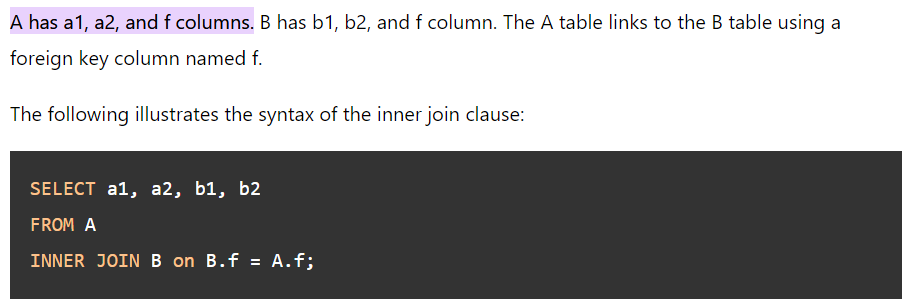
MSSQL scores 5/5

SQLite

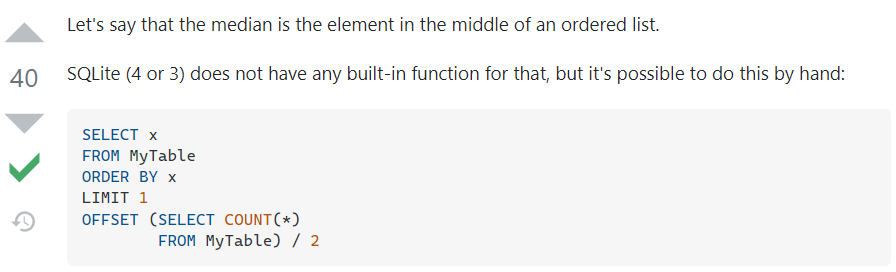
Convert datetime to date



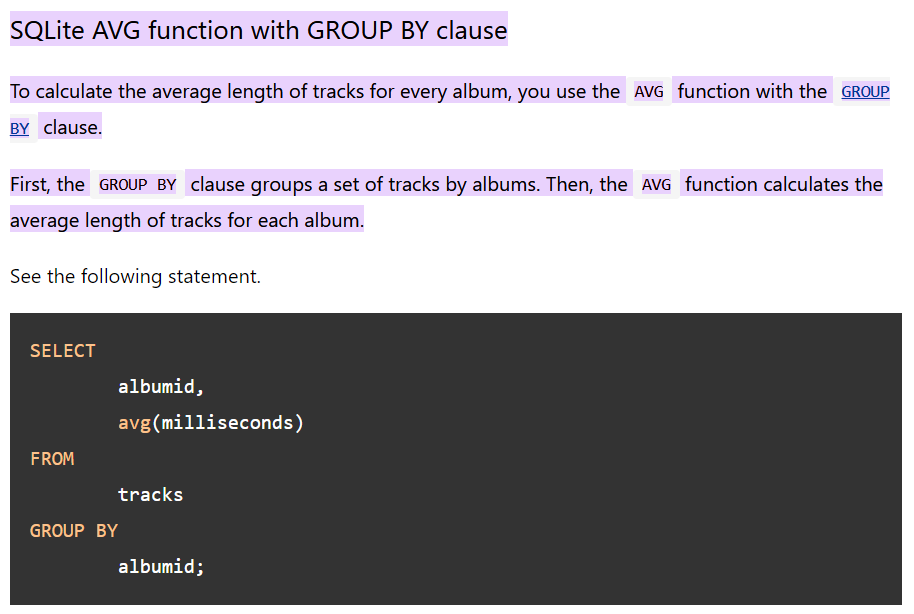
Get data from multiple tables



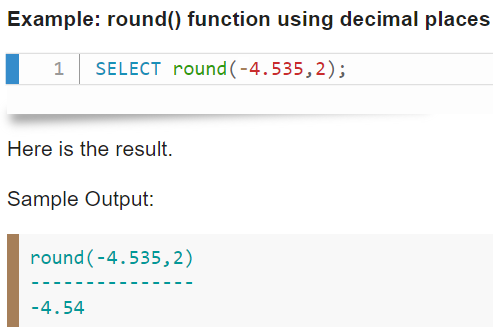
Get the median



Get the average



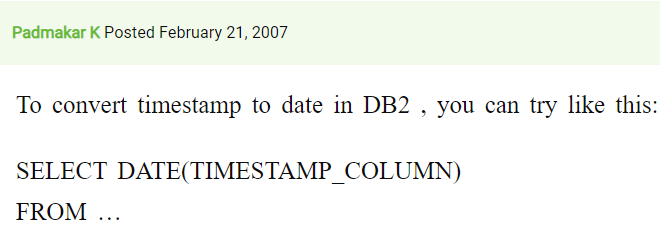
Round to 3 decimals, again an example with 2 decimals



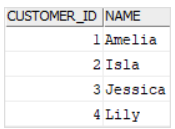
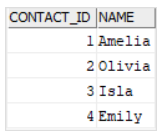
SQLite scores 5/5

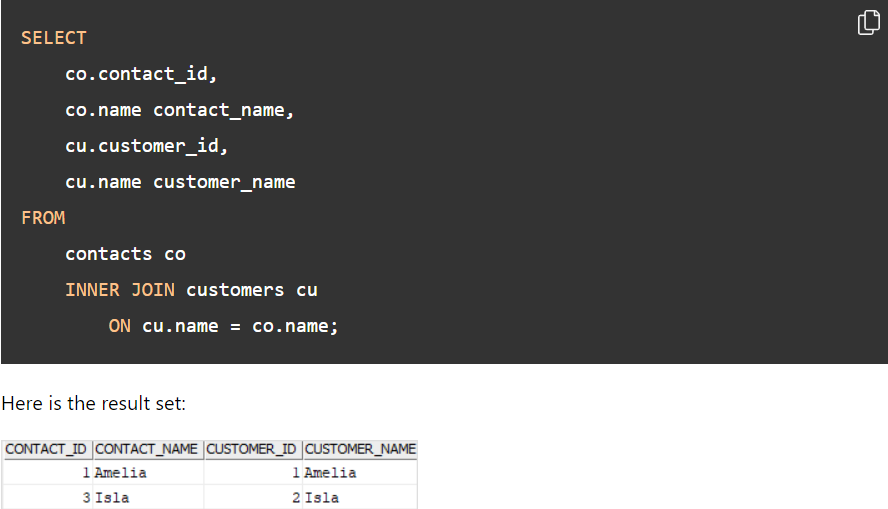
##### IBM DB2

Convert datetime to date, this took a while to find.

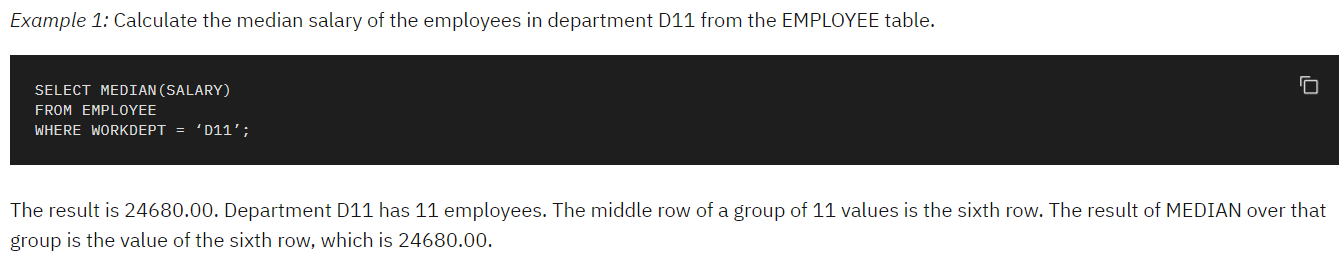


Get data from multiple tables

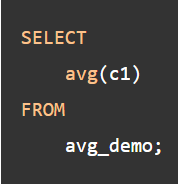




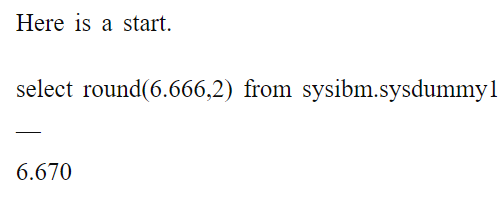
Get the median



Get the average



Round to 3 decimals



IBM DB2 scores 5/5

##### Conclusion

What seemed like a fun and interesting research appeared to be pretty useless. The things I learned are, IBM DB2 is not as known and has a smaller community. For the normal SQL language I liked the sources more than the sources I got for PostgreSQL. For example if I looked up something about SQL I got a W3school source, and I'm a huge fan of W3schools. For the languages with a specific dialect like PostgreSQL the sources were usually understandable as well. With this research I can say that I won't choose IBM DB2 for sure. Classic SQL has an advantage over the others due to the clearest sources. And Dialect SQL databases are neutral.

# So, what Database Suits my application the best?

A relational database with number and string data types. If we look at the following databases, IBM DB2, MariaDB, PostgreSQL, MSSQL, SQLite and MySQL. PostgreSQL comes with lots of extra data types we don't need so we can leave this one out. Other than that, I could choose any database of the ones I just summed up.

My Database Choice

For my application I will use MySQL. I'm choosing MySQL because I have already installed MySQL on my pc, and I recently worked with it for my group project. At this time I have 0 seconds to lose. So I have to save up any bit of time. If time wasn't the issue I might have chosen for any of the others except IBM DB2 since the community is so small.