Database Research

Table Of Contents

Preface 2

Which database is most suitable for my application? 2

Where do databases differ? 2

How reliable is Natan Silnitsky's database study? 3

Who is Natan Silnitsky? 3

What is Wix.com? 3

Other guides to help me choose a database 3

Who is Leitao Guo? 3

Who is Anu Upadhyay? 4

Conclusion 4

Query Patterns 4

What kind of queries do I need? 4

What kind of Database would suite my application in terms of Query Patterns? 5

Key-Value Storage 5

Wide-Column Database 6

Relational Database 6

Document Database 8

Search-Engine Database 8

Conclusion 8

Consistency 8

Storage Capacity 8

Object Storage Service 8

Conclusion 8

Performance 8

Maturity and Stability 9

Individual Project 9

Cost 9

Free Database 9

Data Types 9
Databases that meet my requirements 10
Differences between SQL Databases 10
SQL Dialects 10
Community Support 10
Testing the community 11
PostgreSQL 11
MySQL 13
MariaDB 16
MSSQL 18
SQLite 21
IBM DB2 23
Conclusion 24
So, what Database Suits my application the best? 24
My Database Choice 24
Bibliography 25

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.

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.

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.

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.

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.

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.

Conclusion

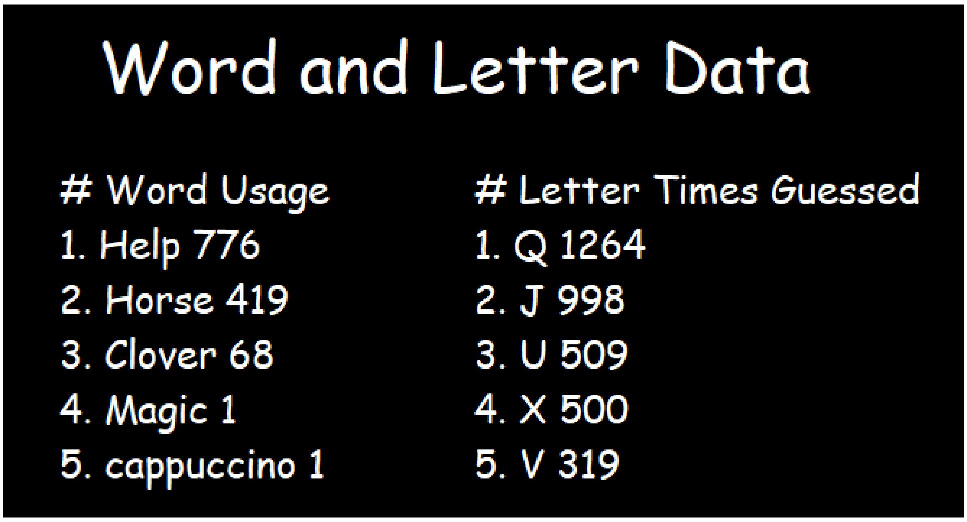
First of all, I doubted Natan when I looked him up. Because he doesn't work in a huge company. The rest I found was actually pretty solid. When I started researching other people that wrote guides, I realised that the people writing them are very hard to find on the internet except Natan. In my eyes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



Word and Letter Data

#	Word	Usage	#	Letter	Times Guessed
1.	Help	776	1.	Q	1264
2.	Horse	419	2.	J	998
3.	Clover	68	3.	U	509
4.	Magic	1	4.	X	500
5.	cappuccino	1	5.	V	319

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.

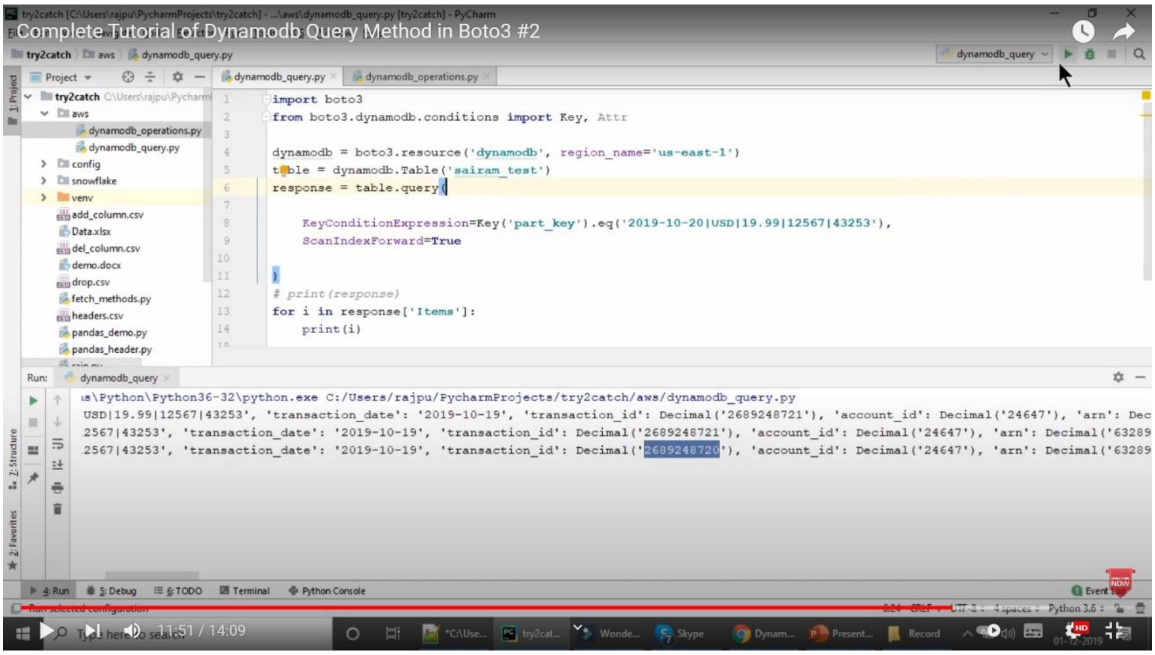
What kind of Database would suite my application in terms of Query Patterns?

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.

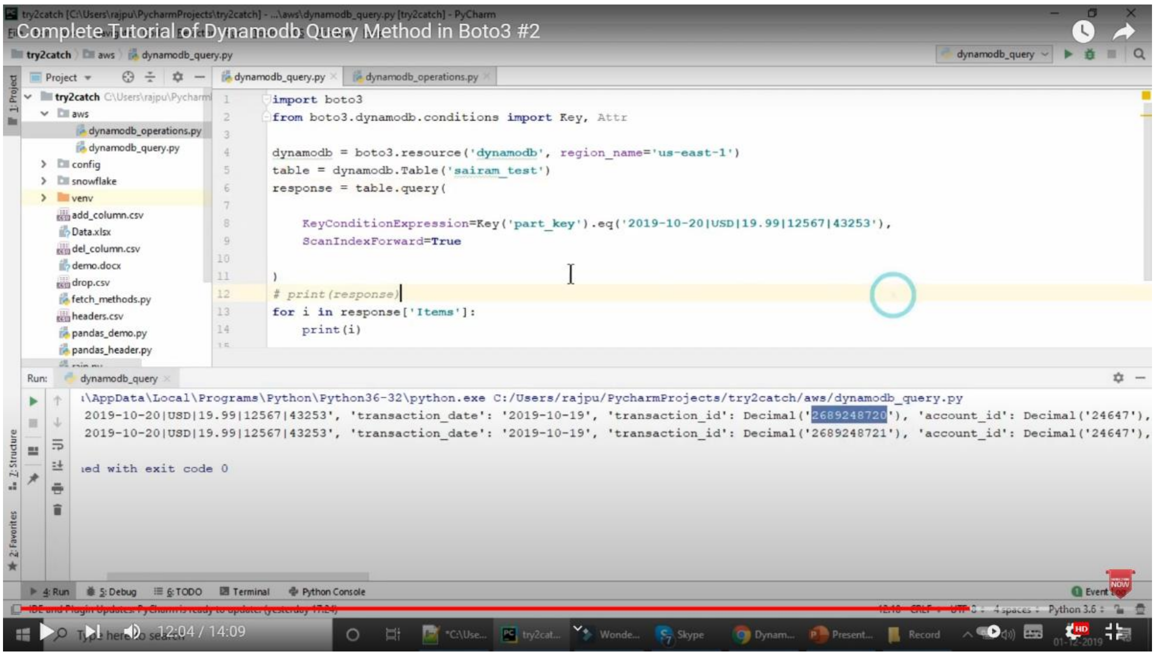
Sadly, I came to the conclusion that I need a credit card to access DynamoDB. And since I don’t have a credit card I can't test it myself. I did find someone doing something similar to what I wanted to try. Here you can hardly see that he put ScanIndexForward=True. The result of this is that it prints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



```
import boto3
from boto3.dynamodb.conditions import Key, Attr

dynamodb = boto3.resource('dynamodb', region_name='us-east-1')
table = dynamodb.Table('sairam_test')
response = table.query(

    KeyConditionExpression=Key('part_key').eq('2019-10-20|USD|19.99|12567|43253'),
    ScanIndexForward=True

)
# print(response)
for i in response['Items']:
    print(i)
```

```
s\Python\Python36-32\python.exe C:/Users/rajpu/PycharmProjects/try2catch/aws/dynamodb_query.py
USD|19.99|12567|43253', 'transaction_date': '2019-10-19', 'transaction_id': Decimal('2689248721'), 'account_id': Decimal('24647'), 'arn': Dec
2567|43253', 'transaction_date': '2019-10-19', 'transaction_id': Decimal('2689248721'), 'account_id': Decimal('24647'), 'arn': Decimal('63289
2567|43253', 'transaction_date': '2019-10-19', 'transaction_id': Decimal('2689248720'), 'account_id': Decimal('24647'), 'arn': Decimal('63289
```

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



```
import boto3
from boto3.dynamodb.conditions import Key, Attr

dynamodb = boto3.resource('dynamodb', region_name='us-east-1')
table = dynamodb.Table('sairam_test')
response = table.query(

    KeyConditionExpression=Key('part_key').eq('2019-10-20|USD|19.99|12567|43253'),
    ScanIndexForward=True

)
# print(response)
for i in response['Items']:
    print(i)
```

```
\AppData\Local\Programs\Python\Python36-32\python.exe C:/Users/rajpu/PycharmProjects/try2catch/aws/dynamodb_query.py
2019-10-20|USD|19.99|12567|43253', 'transaction_date': '2019-10-19', 'transaction_id': Decimal('2689248720'), 'account_id': Decimal('24647'),
2019-10-20|USD|19.99|12567|43253', 'transaction_date': '2019-10-19', 'transaction_id': Decimal('2689248721'), 'account_id': Decimal('24647'),

ıed with exit code 0
```

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

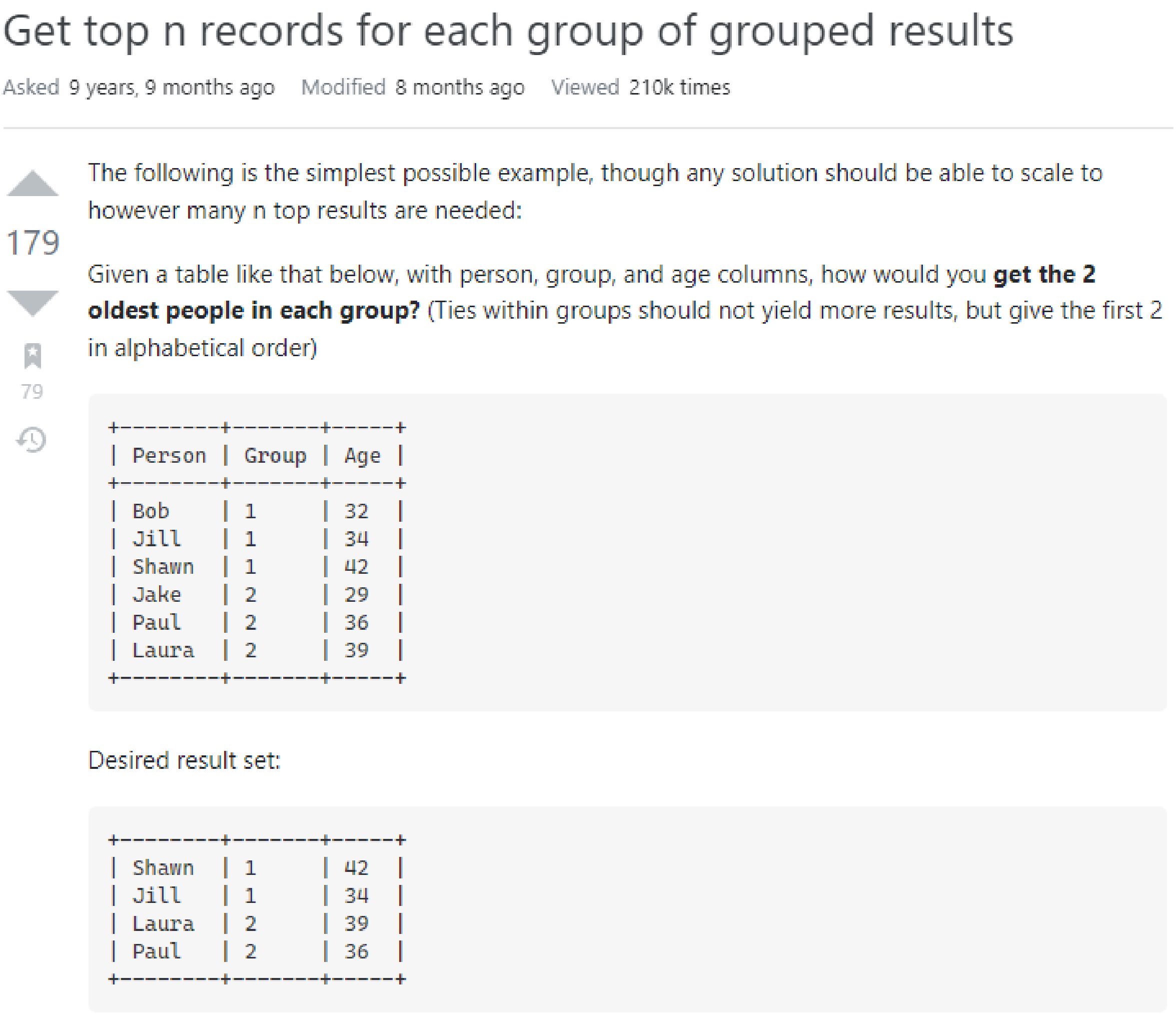
The only real difference between a key-value storage and a wide-column storage, is that a wide-column storage has a better performance when working with bigger datasets. But since my dataset won't be so big, only 2 tables with both 2 columns and lots of rows, a wide-column storage is unnecessary.

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.

Question:



Get top n records for each group of grouped results

Asked 9 years, 9 months ago Modified 8 months ago Viewed 210k times

The following is the simplest possible example, though any solution should be able to scale to however many n top results are needed:

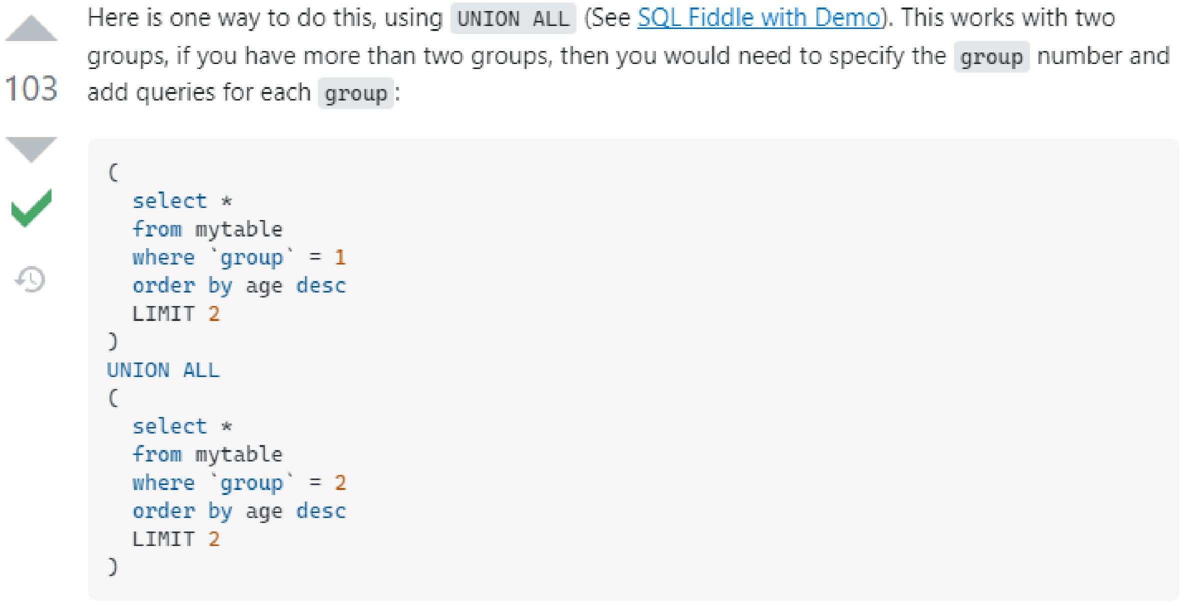
Given a table like that below, with person, group, and age columns, how would you **get the 2 oldest people in each group?** (Ties within groups should not yield more results, but give the first 2 in alphabetical order)

```
+--------+-------+-----+
| Person | Group | Age |
+--------+-------+-----+
| Bob    | 1     | 32  |
| Jill   | 1     | 34  |
| Shawn  | 1     | 42  |
| Jake   | 2     | 29  |
| Paul   | 2     | 36  |
| Laura  | 2     | 39  |
+--------+-------+-----+
```

Desired result set:

```
+--------+-------+-----+
| Shawn  | 1     | 42  |
| Jill   | 1     | 34  |
| Laura  | 2     | 39  |
| Paul   | 2     | 36  |
+--------+-------+-----+
```

Answer:



Here is one way to do this, using `UNION ALL` (See SQL Fiddle with Demo). This works with two groups, if you have more than two groups, then you would need to specify the `group` number and add queries for each `group`:

```
(
  select *
  from mytable
  where `group` = 1
  order by age desc
  LIMIT 2
)
UNION ALL
(
  select *
  from mytable
  where `group` = 2
  order by age desc
  LIMIT 2
)
```

(Taryn, 2012)

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.

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 anything like this so this one definitely falls off.

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.

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.

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.

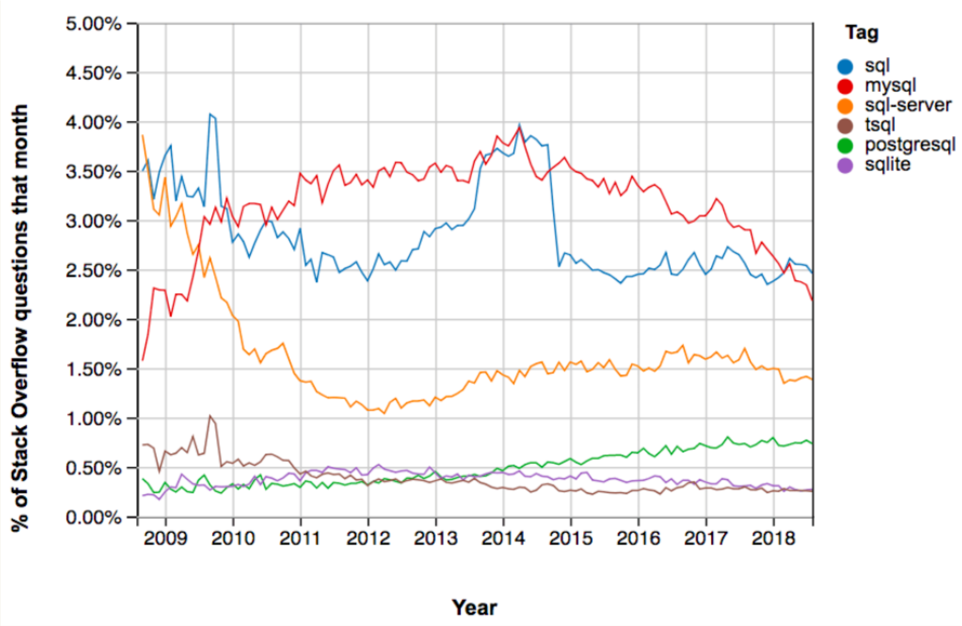
SQL Dialects

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.

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

To extract a date (yyyy-mm-dd) from a timestamp value

For example, you want to extract from `'2018-07-25 10:30:30'` to `'2018-07-25'`

1. Extract from a timestamp column:

Use `date()` function:

```
SELECT DATE(column_name) FROM table_name;
```

Getting data from multiple tables

To select data from both tables, you use the `INNER JOIN` clause in the `SELECT` statement as follows:

```
SELECT
        pka,
        c1,
        pkb,
        c2
FROM
        A
INNER JOIN B ON pka = fka;
```

To join table `A` with the table `B`, you follow these steps:

- First, specify columns from both tables that you want to select data in the `SELECT` clause.
- Second, specify the main table i.e., table `A` in the `FROM` clause.
- Third, specify the second table (table `B`) in the `INNER JOIN` clause and provide a join condition after the `ON` keyword.

Getting median

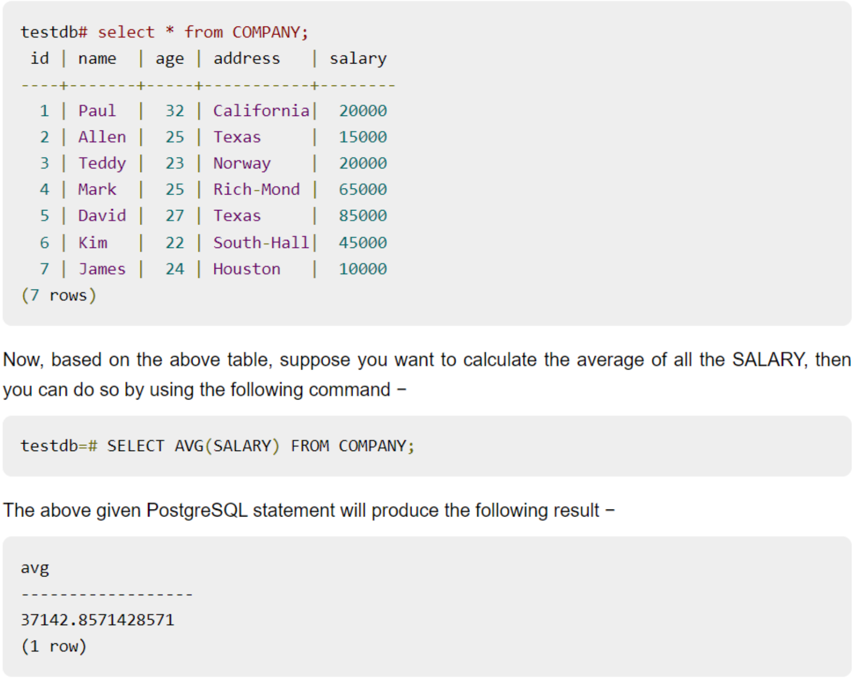
```
postgres=# select * from sales;
 order_date | sale
------------+------
 2020-04-01 |  210
 2020-04-02 |  125
 2020-04-03 |  150
 2020-04-04 |  230
 2020-04-05 |  200
 2020-04-10 |  220
 2020-04-06 |   25
 2020-04-07 |  215
 2020-04-08 |  300
 2020-04-09 |  250
```

Bonus Read : How to Calculate Moving Average in PostgreSQL

Here's the query to find median, that is, calculate 50th percentile in PostgreSQL.

```
postgres=# SELECT PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY sale) FROM sales;
 percentile_cont
-----------------
           212.5
```

Getting average



```
testdb# select * from COMPANY;
 id | name  | age | address    | salary
----+-------+-----+-----------+--------
  1 | Paul  |  32 | California|  20000
  2 | Allen |  25 | Texas      |  15000
  3 | Teddy |  23 | Norway     |  20000
  4 | Mark  |  25 | Rich-Mond  |  65000
  5 | David |  27 | Texas      |  85000
  6 | Kim   |  22 | South-Hall|  45000
  7 | James |  24 | Houston    |  10000
(7 rows)
```

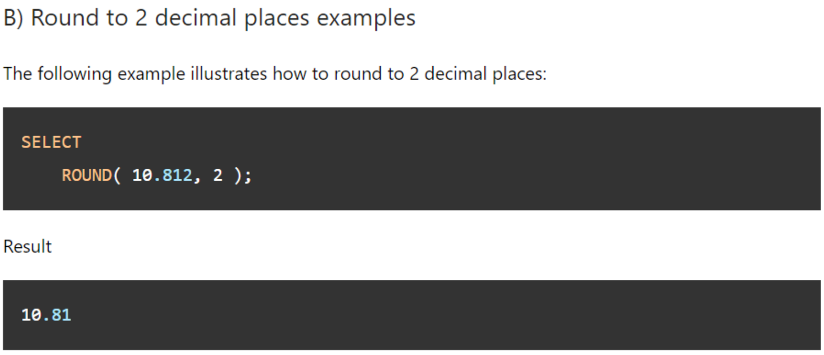
Now, based on the above table, suppose you want to calculate the average of all the SALARY, then you can do so by using the following command −

```
testdb=# SELECT AVG(SALARY) FROM COMPANY;
```

The above given PostgreSQL statement will produce the following result −

```
avg
------------------
37142.8571428571
(1 row)
```

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.



B) Round to 2 decimal places examples

The following example illustrates how to round to 2 decimal places:

```
SELECT
    ROUND( 10.812, 2 );
```

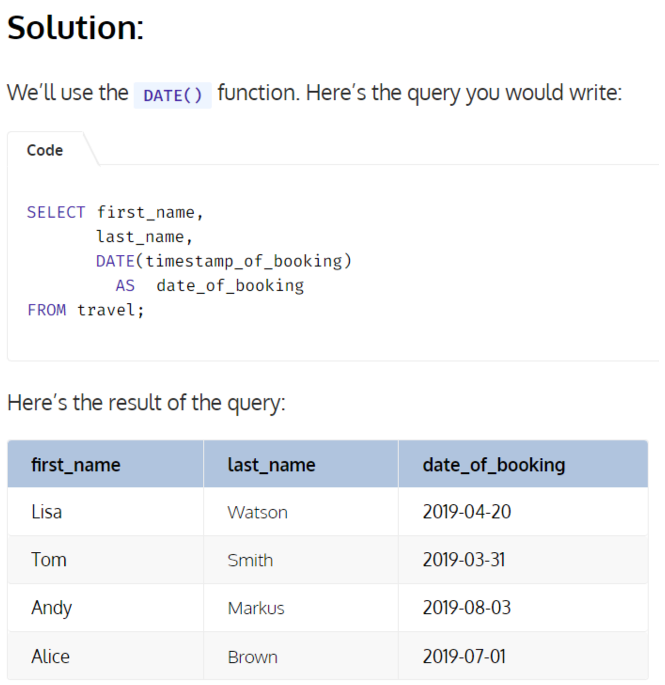
Result

```
10.81
```

PostgreSQL scores 5/5

MySQL

Convert datetime to date



Solution:

We'll use the `DATE()` function. Here's the query you would write:

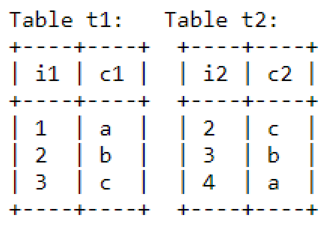
Code

```
SELECT first_name,
       last_name,
       DATE(timestamp_of_booking)
         AS  date_of_booking
FROM travel;
```

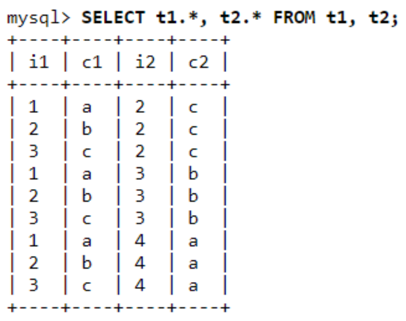
Here's the result of the query:

first_name	last_name	date_of_booking
Lisa	Watson	2019-04-20
Tom	Smith	2019-03-31
Andy	Markus	2019-08-03
Alice	Brown	2019-07-01

Get data from multiple tables

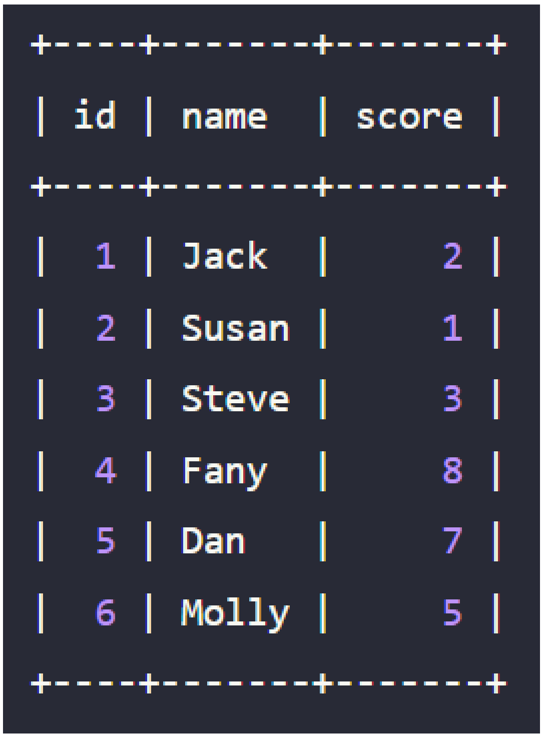


```
Table t1:    Table t2:
+----+----+  +----+----+
| i1 | c1 |  | i2 | c2 |
+----+----+  +----+----+
| 1  | a  |  | 2  | c  |
| 2  | b  |  | 3  | b  |
| 3  | c  |  | 4  | a  |
+----+----+  +----+----+
```

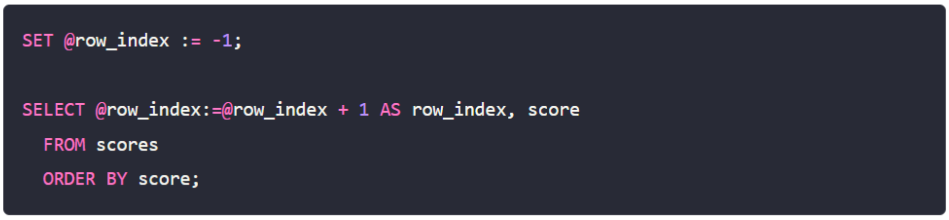


```
mysql> SELECT t1.*, t2.* FROM t1, t2;
+----+----+----+----+
| i1 | c1 | i2 | c2 |
+----+----+----+----+
| 1  | a  | 2  | c  |
| 2  | b  | 2  | c  |
| 3  | c  | 2  | c  |
| 1  | a  | 3  | b  |
| 2  | b  | 3  | b  |
| 3  | c  | 3  | b  |
| 1  | a  | 4  | a  |
| 2  | b  | 4  | a  |
| 3  | c  | 4  | a  |
+----+----+----+----+
```

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.

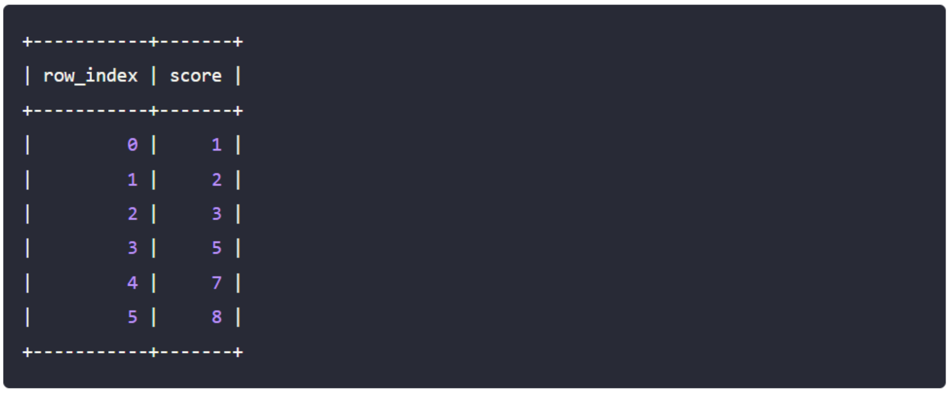


```
+----+-------+-------+
| id | name  | score |
+----+-------+-------+
|  1 | Jack  |     2 |
|  2 | Susan |     1 |
|  3 | Steve |     3 |
|  4 | Fany  |     8 |
|  5 | Dan   |     7 |
|  6 | Molly |     5 |
+----+-------+-------+
```

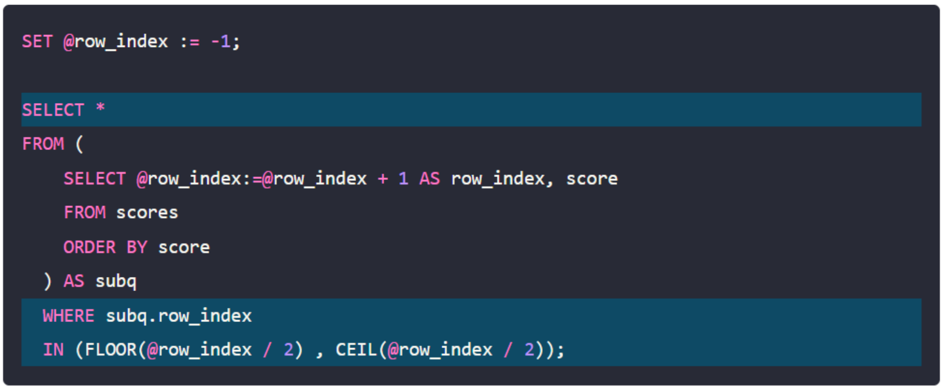


```sql
SET @row_index := -1;

SELECT @row_index:=@row_index + 1 AS row_index, score
  FROM scores
  ORDER BY score;
```

The above SQL query will produce the following result set:



```
+-----------+-------+
| row_index | score |
+-----------+-------+
|         0 |     1 |
|         1 |     2 |
|         2 |     3 |
|         3 |     5 |
|         4 |     7 |
|         5 |     8 |
+-----------+-------+
```

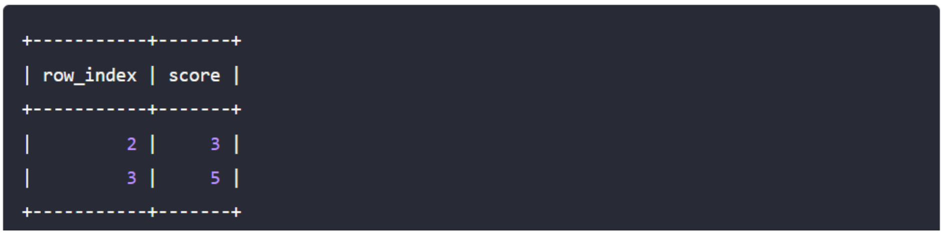


```
SET @row_index := -1;

SELECT *
FROM (
    SELECT @row_index:=@row_index + 1 AS row_index, score
    FROM scores
    ORDER BY score
  ) AS subq
  WHERE subq.row_index
  IN (FLOOR(@row_index / 2) , CEIL(@row_index / 2));
```

The `WHERE` clause above will filter the subquery result to return only the middle index number(s).

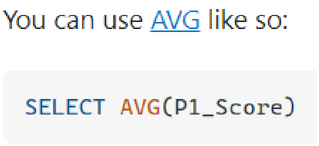
When you have an odd number of rows, then the middle row will be returned. When you have an even number of rows, then the two middle rows will be returned.

Here's the output returned from the query above:



```
+-----------+-------+
| row_index | score |
+-----------+-------+
|         2 |     3 |
|         3 |     5 |
+-----------+-------+
```

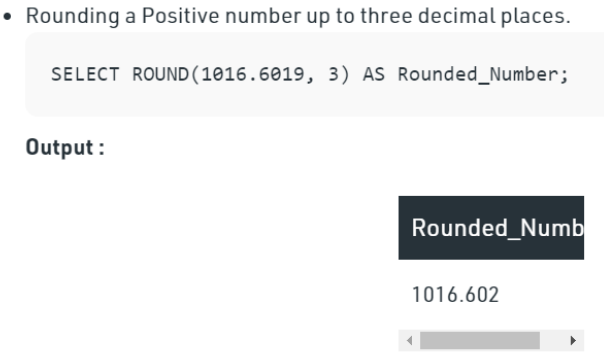
Get the average



You can use AVG like so:

```
SELECT AVG(P1_Score)
```

Round to 3 decimals



- Rounding a Positive number up to three decimal places.

```
SELECT ROUND(1016.6019, 3) AS Rounded_Number;
```

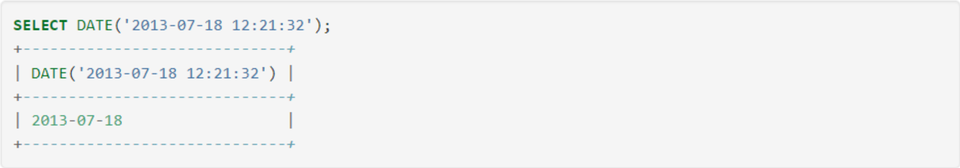
Output :

Rounded_Numb
1016.602

MySQL scores 5 / 5

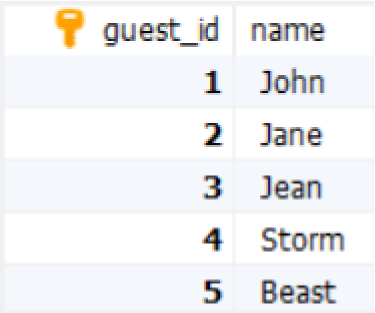
MariaDB

Convert datetime to date

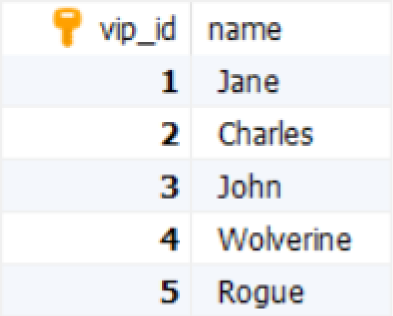


```
SELECT DATE('2013-07-18 12:21:32');
+-----------------------------+
| DATE('2013-07-18 12:21:32') |
+-----------------------------+
| 2013-07-18                  |
+-----------------------------+
```

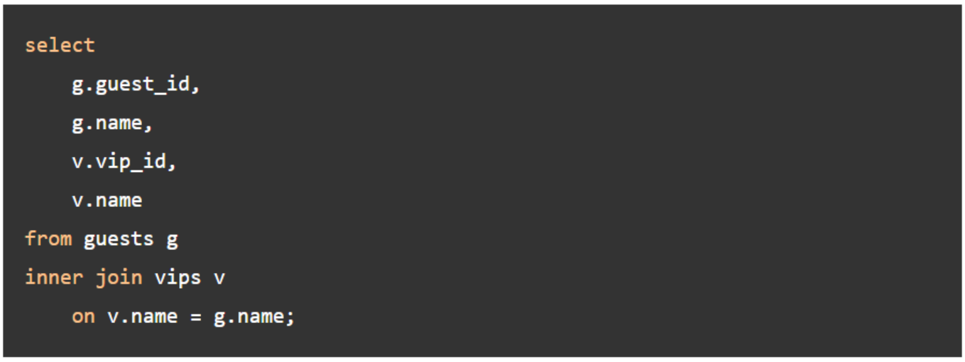
Get data from multiple tables



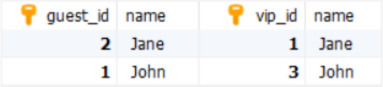
guest_id	name
1	John
2	Jane
3	Jean
4	Storm
5	Beast



vip_id	name
1	Jane
2	Charles
3	John
4	Wolverine
5	Rogue

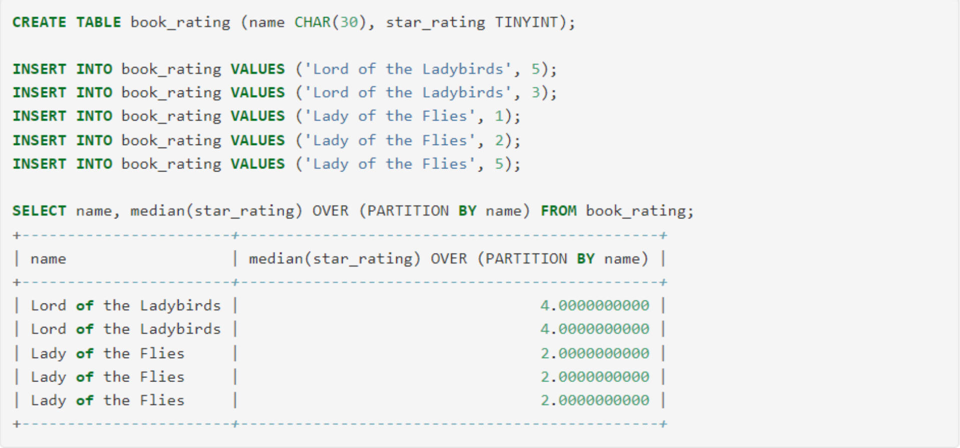


```
select
    g.guest_id,
    g.name,
    v.vip_id,
    v.name
from guests g
inner join vips v
    on v.name = g.name;
```



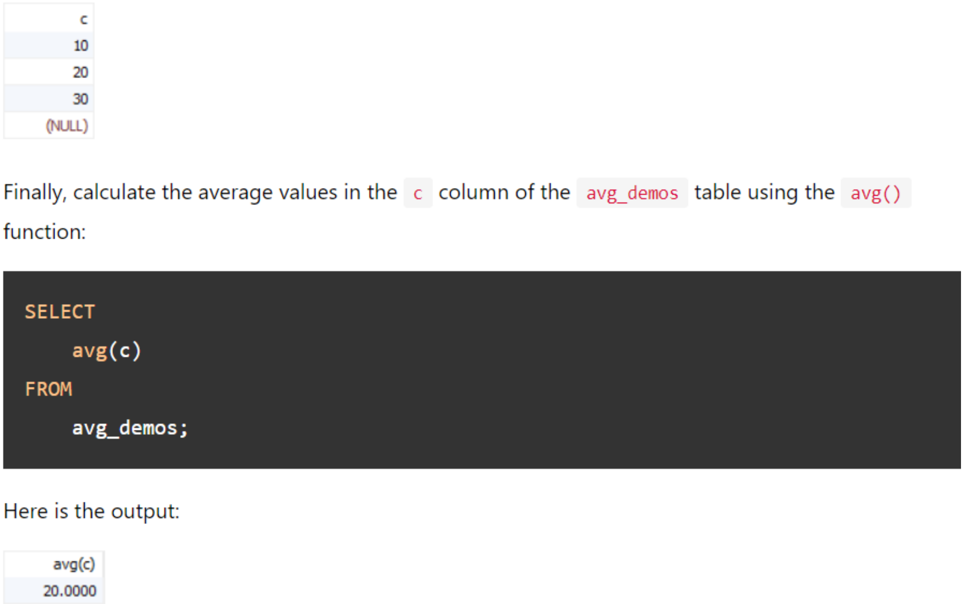
guest_id	name	vip_id	name
2	Jane	1	Jane
1	John	3	John

Get the median



```
CREATE TABLE book_rating (name CHAR(30), star_rating TINYINT);

INSERT INTO book_rating VALUES ('Lord of the Ladybirds', 5);
INSERT INTO book_rating VALUES ('Lord of the Ladybirds', 3);
INSERT INTO book_rating VALUES ('Lady of the Flies', 1);
INSERT INTO book_rating VALUES ('Lady of the Flies', 2);
INSERT INTO book_rating VALUES ('Lady of the Flies', 5);

SELECT name, median(star_rating) OVER (PARTITION BY name) FROM book_rating;
+-----------------------+-----------------------------------------------+
| name                  | median(star_rating) OVER (PARTITION BY name) |
+-----------------------+-----------------------------------------------+
| Lord of the Ladybirds |                                  4.0000000000 |
| Lord of the Ladybirds |                                  4.0000000000 |
| Lady of the Flies     |                                  2.0000000000 |
| Lady of the Flies     |                                  2.0000000000 |
| Lady of the Flies     |                                  2.0000000000 |
+-----------------------+-----------------------------------------------+
```

Get the average



c
10
20
30
(NULL)

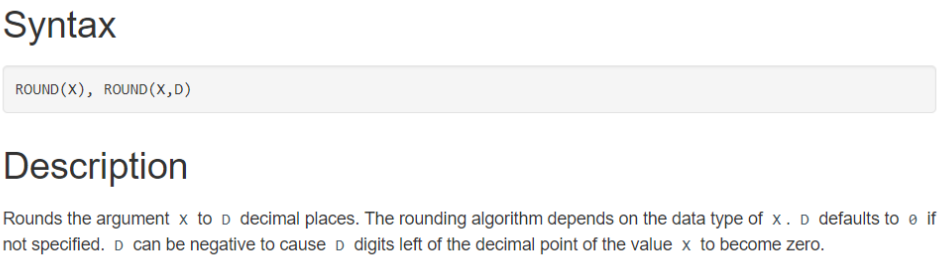
Finally, calculate the average values in the `c` column of the `avg_demos` table using the `avg()` function:

```
SELECT
    avg(c)
FROM
    avg_demos;
```

Here is the output:

avg(c)
20.0000

Round to 3 decimals, no example but understandable.



Syntax

```
ROUND(X), ROUND(X,D)
```

Description

Rounds the argument `X` to `D` decimal places. The rounding algorithm depends on the data type of `X`. `D` defaults to `0` if not specified. `D` can be negative to cause `D` digits left of the decimal point of the value `X` to become zero.

MariaDB scores 5/5

MSSQL

Convert datetime to date

```
CONVERT(DATE, datetime_expression)
```

In this syntax, the `datetime_expresssion` is any valid expression that evaluates to a valid datetime value. The `CONVERT()` function will raise an error if the conversion fails.

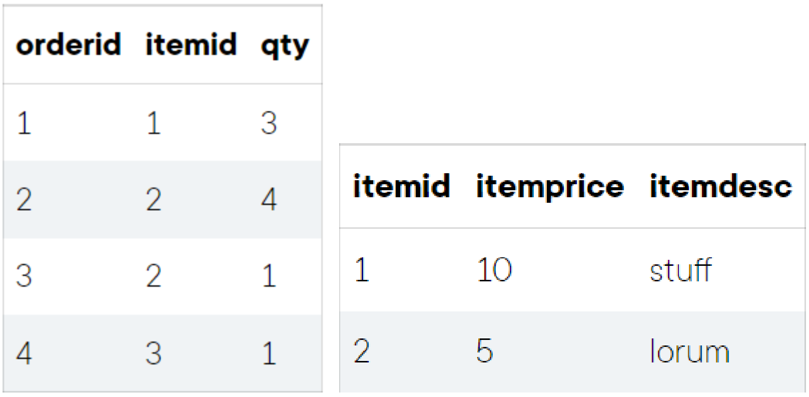
The following example uses the `CONVERT()` function to convert a datetime to a date:

```
SELECT
    CONVERT(DATE, GETDATE()) date;
```

Here is the output:

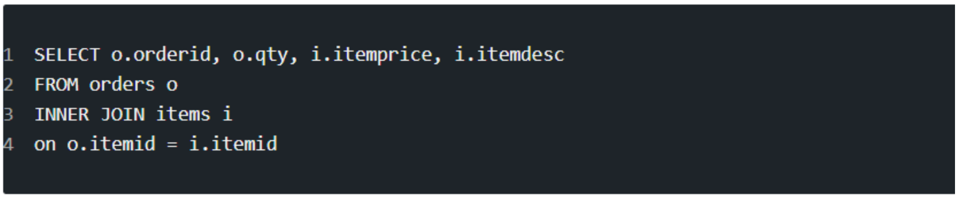
```
date
----------
2019-04-23

(1 row affected)
```

Get data from multiple tables



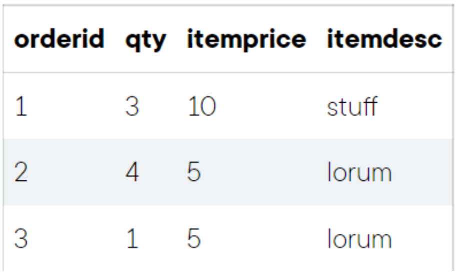
orderid	itemid	qty
1	1	3
2	2	4
3	2	1
4	3	1

itemid	itemprice	itemdesc
1	10	stuff
2	5	lorum



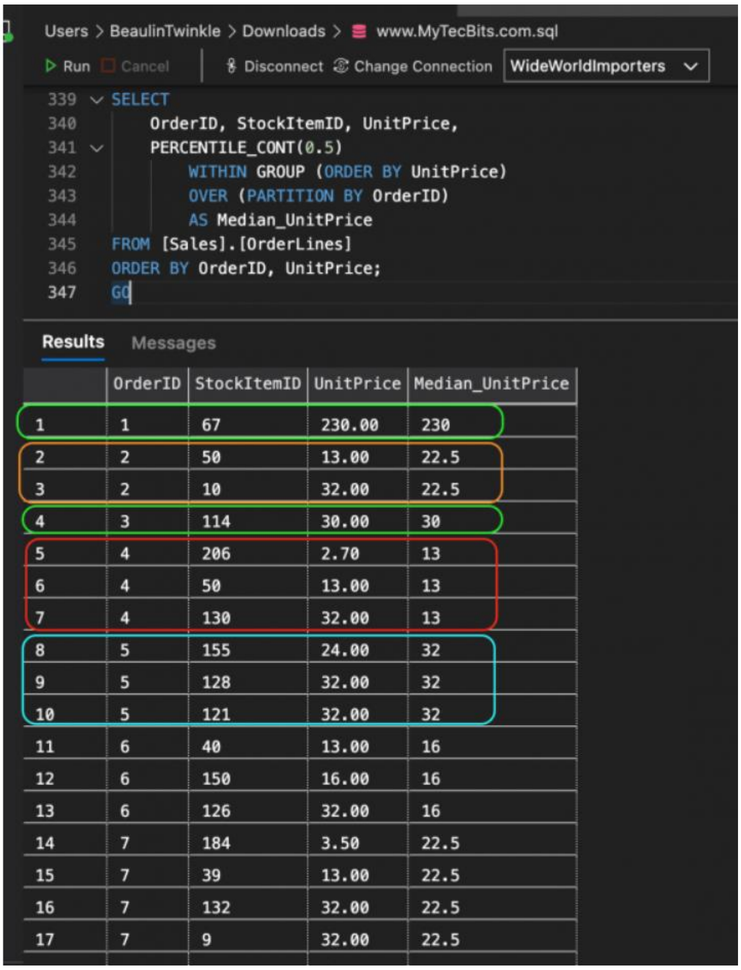
```
SELECT o.orderid, o.qty, i.itemprice, i.itemdesc
FROM orders o
INNER JOIN items i
on o.itemid = i.itemid
```

The result set would be as below.



orderid	qty	itemprice	itemdesc
1	3	10	stuff
2	4	5	lorum
3	1	5	lorum

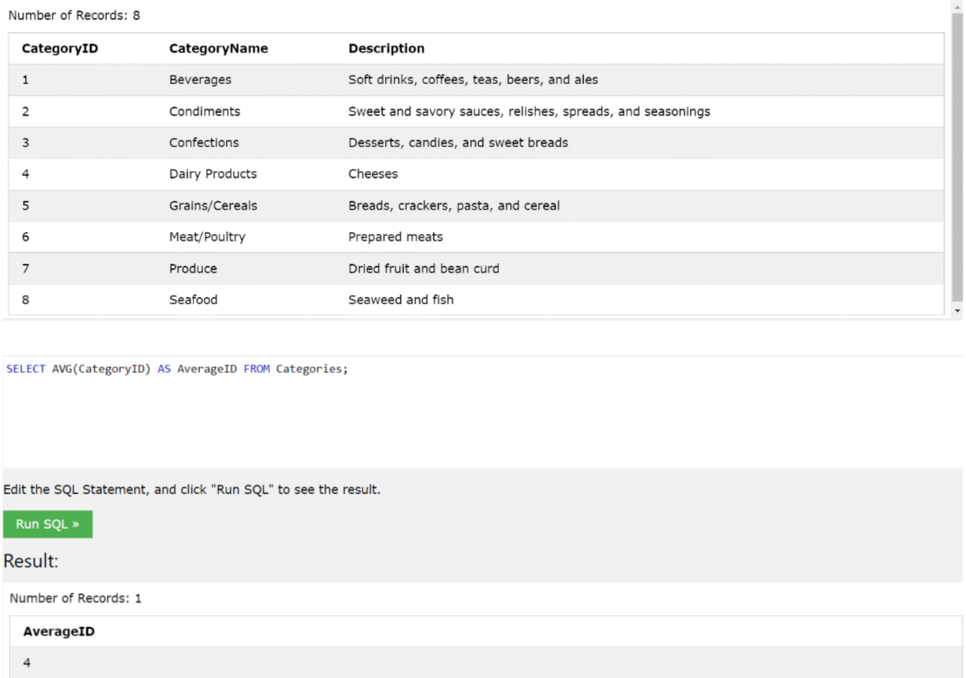
Get the median, here you can see the median per id.



```
SELECT
    OrderID, StockItemID, UnitPrice,
    PERCENTILE_CONT(0.5)
        WITHIN GROUP (ORDER BY UnitPrice)
        OVER (PARTITION BY OrderID)
        AS Median_UnitPrice
FROM [Sales].[OrderLines]
ORDER BY OrderID, UnitPrice;
GO
```

	OrderID	StockItemID	UnitPrice	Median_UnitPrice
1	1	67	230.00	230
2	2	50	13.00	22.5
3	2	10	32.00	22.5
4	3	114	30.00	30
5	4	206	2.70	13
6	4	50	13.00	13
7	4	130	32.00	13
8	5	155	24.00	32
9	5	128	32.00	32
10	5	121	32.00	32
11	6	40	13.00	16
12	6	150	16.00	16
13	6	126	32.00	16
14	7	184	3.50	22.5
15	7	39	13.00	22.5
16	7	132	32.00	22.5
17	7	9	32.00	22.5

Get the average



Number of Records: 8

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

```
SELECT AVG(CategoryID) AS AverageID FROM Categories;
```

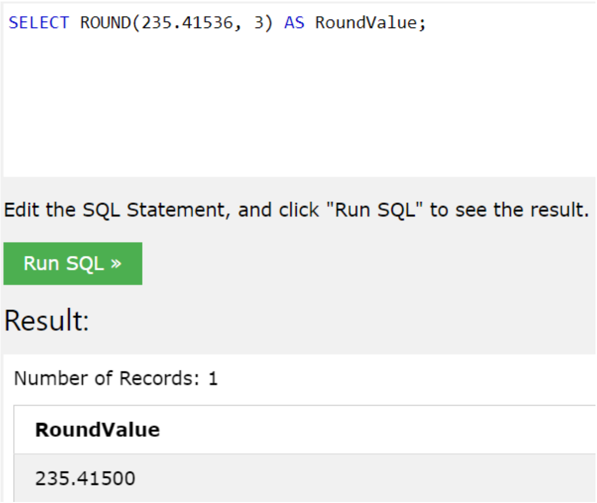
Edit the SQL Statement, and click "Run SQL" to see the result.

Result:

Number of Records: 1

AverageID
4

Round to 3 decimals



```
SELECT ROUND(235.41536, 3) AS RoundValue;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Result:

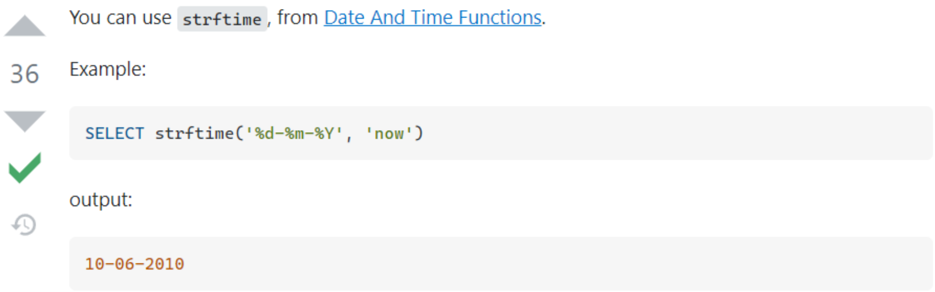
Number of Records: 1

RoundValue
235.41500

MSSQL scores 5 / 5

SQLite

Convert datetime to date



You can use `strftime`, from Date And Time Functions.

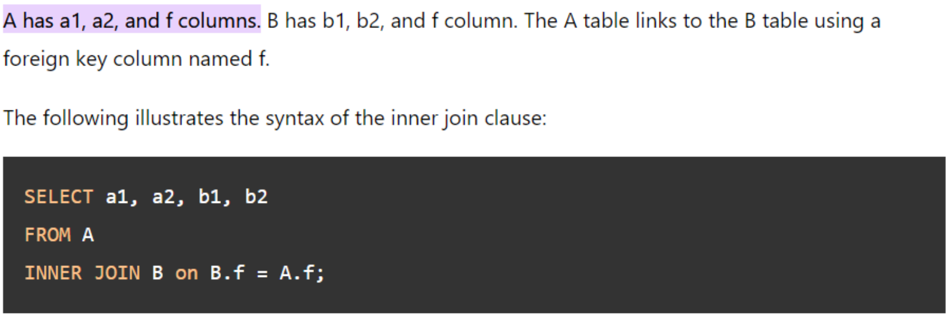
Example:

```
SELECT strftime('%d-%m-%Y', 'now')
```

output:

```
10-06-2010
```

Get data from multiple tables

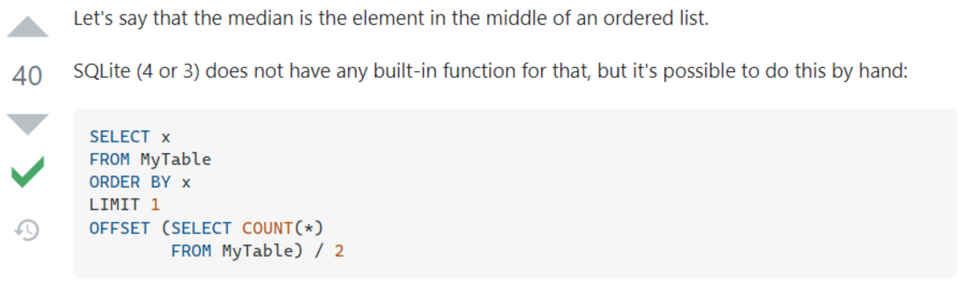


A has a1, a2, and f columns. B has b1, b2, and f column. The A table links to the B table using a foreign key column named f.

The following illustrates the syntax of the inner join clause:

```
SELECT a1, a2, b1, b2
FROM A
INNER JOIN B on B.f = A.f;
```

Get the median

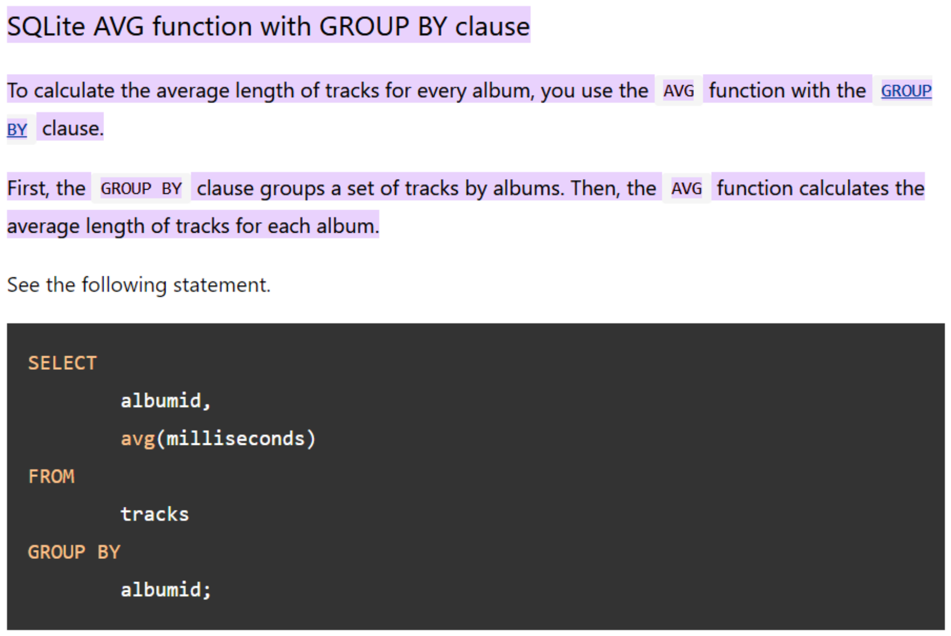


Let's say that the median is the element in the middle of an ordered list.

SQLite (4 or 3) does not have any built-in function for that, but it's possible to do this by hand:

```
SELECT x
FROM MyTable
ORDER BY x
LIMIT 1
OFFSET (SELECT COUNT(*)
        FROM MyTable) / 2
```

Get the average



SQLite AVG function with GROUP BY clause

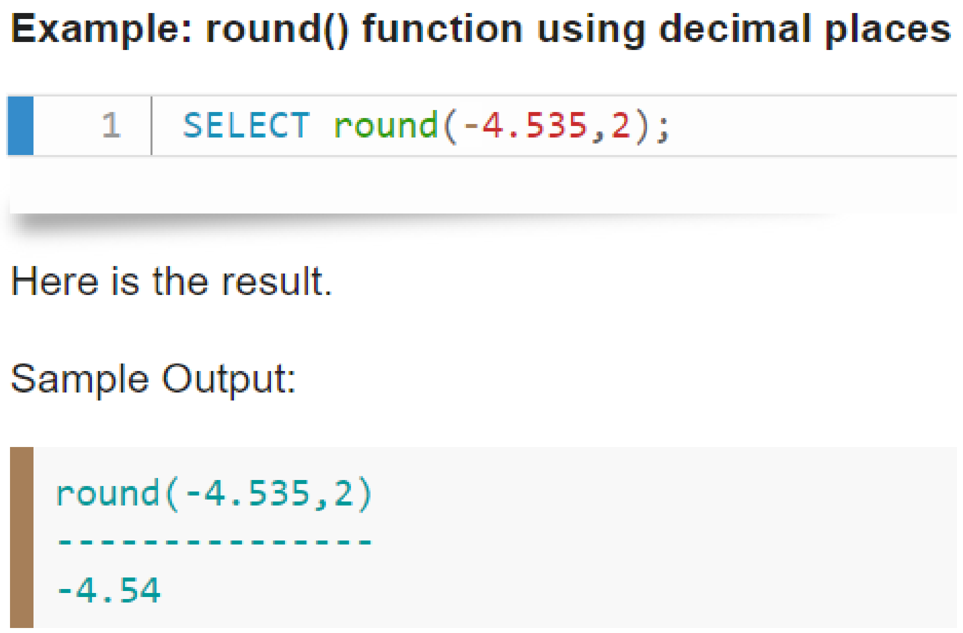
To calculate the average length of tracks for every album, you use the `AVG` function with the `GROUP BY` clause.

First, the `GROUP BY` clause groups a set of tracks by albums. Then, the `AVG` function calculates the average length of tracks for each album.

See the following statement.

```
SELECT
        albumid,
        avg(milliseconds)
FROM
        tracks
GROUP BY
        albumid;
```

Round to 3 decimals, again an example with 2 decimals



Example: round() function using decimal places

```
SELECT round(-4.535,2);
```

Here is the result.

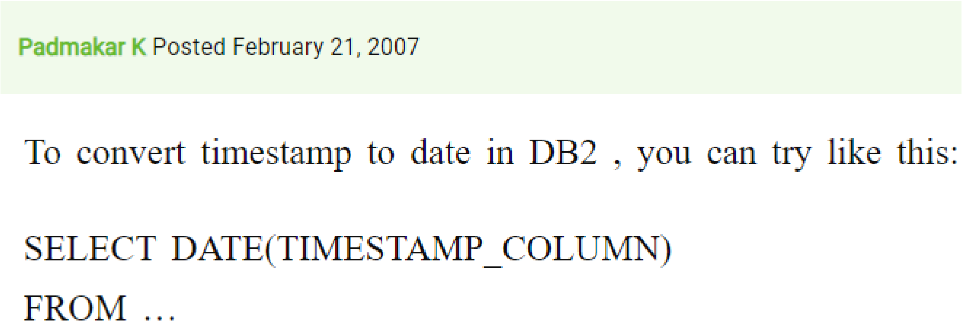
Sample Output:

```
round(-4.535,2)
---------------
-4.54
```

SQLite scores 5 / 5

IBM DB2

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

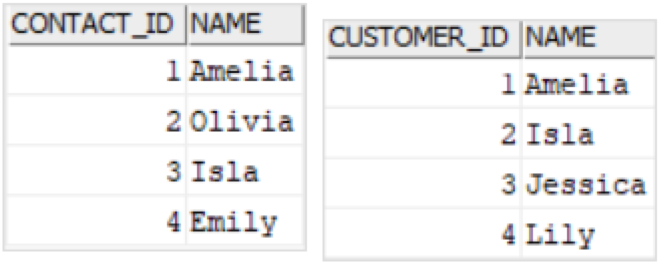


Padmakar K Posted February 21, 2007

To convert timestamp to date in DB2 , you can try like this:

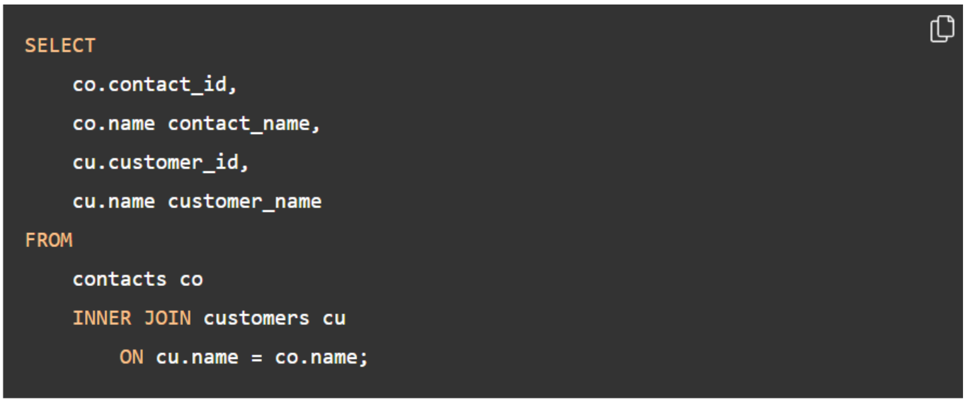
SELECT DATE(TIMESTAMP_COLUMN)
FROM ...

Get data from multiple tables

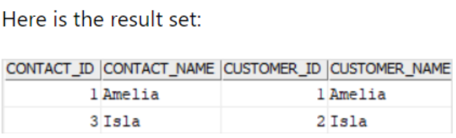


CONTACT_ID	NAME
1	Amelia
2	Olivia
3	Isla
4	Emily

CUSTOMER_ID	NAME
1	Amelia
2	Isla
3	Jessica
4	Lily



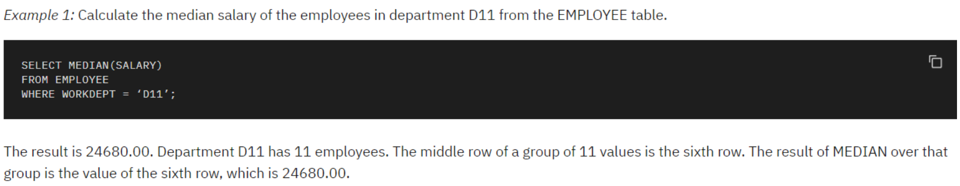
```
SELECT
    co.contact_id,
    co.name contact_name,
    cu.customer_id,
    cu.name customer_name
FROM
    contacts co
    INNER JOIN customers cu
        ON cu.name = co.name;
```



Here is the result set:

CONTACT_ID	CONTACT_NAME	CUSTOMER_ID	CUSTOMER_NAME
1	Amelia	1	Amelia
3	Isla	2	Isla

Get the median

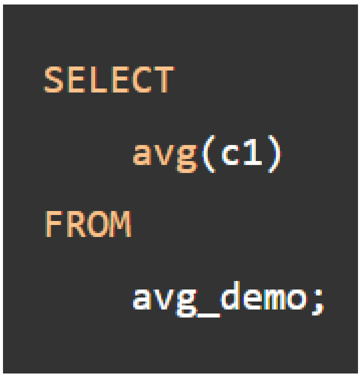


Example 1: Calculate the median salary of the employees in department D11 from the EMPLOYEE table.

```
SELECT MEDIAN(SALARY)
FROM EMPLOYEE
WHERE WORKDEPT = 'D11';
```

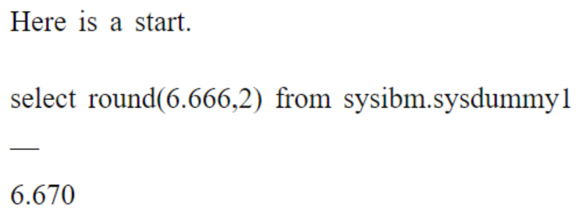
The result is 24680.00. Department D11 has 11 employees. The middle row of a group of 11 values is the sixth row. The result of MEDIAN over that group is the value of the sixth row, which is 24680.00.

Get the average



```
SELECT
    avg(c1)
FROM
    avg_demo;
```

Round to 3 decimals



Here is a start.

select round(6.666,2) from sysibm.sysdummy1

—

6.670

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.

Bibliography

About. (sd). Opgehaald van PostgreSQL: https://www.postgresql.org/about/

Berry, E. (2022, 6 13). *Everett Berry LinkedIn Profile.* Opgehaald van LinkedIn: https://www.linkedin.com/in/everettberry

Berry, E. (2022, 3 17). *Understanding SQL Dialects.* Opgehaald van arctype: https://arctype.com/blog/sql-dialects/

Data Concurrency and Consistency. (sd). Opgehaald van Oracle: https://docs.oracle.com/cd/B10500_01/server.920/a96524/c21cnsis.htm

Guo, L. (2021, 2 16). *How to Efficiently Choose the Right Database for Your Applications.* Opgehaald van PingCap: https://en.pingcap.com/blog/how-to-efficiently-choose-the-right-database-for-your-applications/#database-selection-criteria

Guo, L. (2022, 6 13). *Leitao Guo LinkedIn Profile.* Opgehaald van LinkedIn: https://cn.linkedin.com/in/leitao-guo-ba6518b?trk=people-guest_people_search-card&original_referer=https%3A%2F%2Fwww.linkedin.com%2F

Joyce, K. E. (2020, 2 10). *NoSQL database comparison to help you choose the right store.* Opgehaald van TechTarget: https://www.techtarget.com/searchdatamanagement/infographic/NoSQL-database-comparison-to-help-you-choose-the-right-store

Khalil, M. (2018, 10 15). *SQL Server, PostgreSQL, MySQL... what's the difference? Where do I start?* Opgehaald van datacamp: https://www.datacamp.com/blog/sql-server-postgresql-mysql-whats-the-difference-where-do-i-start

Khalil, M. (2022, 6 13). *Mona Khalil LinkedIn Profile.* Opgehaald van LinkedIn: https://www.linkedin.com/in/mona-khalil

kometen. (2016, 4 27). *Is it possible to ORDER results with query or scan in DynamoDB?* Opgehaald van stack overflow: https://stackoverflow.com/questions/9297326/is-it-possible-to-order-results-with-query-or-scan-in-dynamodb

Salunke, R. (2021, 7 15). *MongoDB sort by date.* Opgehaald van EDUCBA: https://www.educba.com/mongodb-sort-by-date/?source=leftnav

Silnitsky, N. (2021, 9 25). *How to choose the right database for your service.* Opgehaald van Medium: https://medium.com/wix-engineering/how-to-choose-the-right-database-for-your-service-97b1670c5632

Silnitsky, N. (2022, 6 13). *Natan's LinkedIn Profile*. Opgehaald van LinkedIn: https://il.linkedin.com/in/natansilnitsky

Taryn. (2012, 8 24). *Get top n records for each group of grouped results*. Opgehaald van stack overflow: https://stackoverflow.com/questions/12113699/get-top-n-records-for-each-group-of-grouped-results

Upadhyay, A. (2020, 12 23). *How to Choose The Right Database for Your Application?* Opgehaald van GeeksForGeeks: https://www.geeksforgeeks.org/how-to-choose-the-right-database-for-your-application/

Upadhyay, A. (2022, 6 13). *Anu Upadhyay LinkedIn Profile*. Opgehaald van LinkedIn: https://in.linkedin.com/in/anu-upadhyay-257709138

What Is a Search-Engine Database? (sd). Opgehaald van Amazon Web Services, Inc.: https://aws.amazon.com/nosql/search/

Wix.com. (2021, 11 8). *Kafka Based Global Data Mesh At Wix- Natan Silnitsky*. Opgehaald van YouTube: https://www.youtube.com/watch?v=3kEueGs1gkc

Wix.com. (2022, 6 13). *Wix.com LinkedIn Profile*. Opgehaald van LinkedIn: https://il.linkedin.com/company/wix-com