**Project Name:Metabase**

**What is Metabase?**

Metabase is a deep product with a lot of tools to simplify business intelligence, from embeddable charts and interactive dashboards, to GUI and SQL editors, to auditing and data sandboxing, and more.Metabase is an open-source business intelligence tool. Metabase lets you ask questions about your data, and displays answers in formats that make sense, whether that’s a bar chart or a detailed table.You can save your questions, and group questions into handsome dashboards. Metabase also makes it easy to share questions and dashboards with the rest of your team.

**Architectural and Component level Modularity:**

Cloud, open-source, and SaaS business models have transformed the software industry and the way that companies think about and build their products. Today, we can set up an entire technology stack in a fraction of the time and cost than before. And it’s no surprise that these transformations have paved the way for the modern data stack.The Modern Data Stack consists of a flexible set of technologies that help businesses store, manage, and learn from their data. Typically, Modern Data Stacks are built on cloud-based services, and increasingly include Low- and No-code tools that empower users to explore and use data.Metabase uses a modern data stack because a Modern Data Stack consists of technologies with generally standard connection points, teams can swap parts of the stack as their needs evolve. This helps them avoid vendor lock-in, and allows the team to grow their stack as their data needs mature.

**Microservices:**

Microservices fundamentally solve the coordination problem in large teams of engineers.They make it easy to independently deploy different portions of a larger application. They also break up the problem into small units of complexity that can be understood, tested and fixed without needing to understand the entire system.Once you get a ginormous monolith, breaking it up can also drastically improve developer productivity as compile, lint and testing times are drastically decreased if you’re only working within the scope of a single service.They also enforce this separation by deployment mechanism, vs convention in using libraries in a single codebase. This makes it harder for people to break encapsulation of your code, by forcing them to use it via a published API.Cynically, they also solve the perennial problem in corporate politics of having to work well with others. Now you can ignore integration issues and just point to your microservice and say “All tests pass and it works!” and look good on your next review.There is a very serious consequence of using microservices on your ability to provide analytics.If you have a data model that is decoupled, at some point you are going to want to well … couple it back together so you can analyse what’s going on with your business. The main proponents of microservices tend to either be engineers at large companies (who have small armies of data engineers to stitch things back together.Metabase uses microservices which are good for initial startups but make iteration difficult.

**Modular Folder Structure:**

Metabase supports a lot of different databases, and ships with a sample database for you to play around with. And once you’ve connected your data sources, Metabase gives you a lot of tools to explore them.

**1)Query builder**

You can use Metabase’s queary builder to filter and summarize data.Use the Notebook editor to ask more sophisticated questions. With custom expressions, you can accomplish pretty much anything you’d be able to do with SQL such as join tables, create custom columns, filter and group results, compare time series, and more. Plus, people who don’t know SQL can duplicate your question and use it as a starting point for another question.

**2)Native Queries:**

Use the native query editor to compose questions in the database’s native query languages (typically SQL for relational databases, but also other query languages for data sources like MongoDB and Google Analytics). For questions written in SQL, you can use variables in your code to create SQL templates, including field filter variables that can create smart dropdown filters.Like query builder questions, you can use the results of models or saved questions as starting points for new questions, just as you would a table or view.

**3)Create Models:**

Models are built with questions from either the query builder or the SQL editor. You can use them to pull together data from multiple tables, with custom, calculated columns, and column descriptions and other metadata, to create great starting data for people to ask new questions. For example, you could build a model for “Active users”, or “Priority orders”, or however you want to model your business.If you find that you’re using the same saved question over and over as your starting data for new questions, you may want to convert that saved question to a model, which will let you add metadata like column descriptions and column types.

**Class Component reuseability:**

Metabase uses the concept of entity loaders to maintain reuseability.

**Entity Loaders:**

If you’re developing a new feature or just generally need to get at some of the application data on the frontend, Entity Loaders are going to be your friend. They abstract away calling the API, handling loading and error state, cache previously loaded objects, invalidating the cache (in some cases) and let you easily perform updates, or create new items.It involves the process and concept of object loading,list loading,controlling overloading and error states and wrapping objects.

**Linters for better code Quality:**

A tool that analyzes source code to flag programming errors, bugs, stylistic errors, and suspicious constructs.More and more teams have adopted linters and other static tools in their development process. Some integrated them in the IDE of their preference, others automated by running them as an additional step in their CI. Also, some run both ways.The first linters used to check the source code and find potential optimizations for compilers. But, over the years, many other checks and analysis would be included in the process of linting.The usage of linters has also helped many developers to write better code for not compiled programming languages. As there is not compiling time errors, finding typos, syntax errors, uses of undeclared variables, calls to undefined or deprecated functions, for instance, helping developers to fix it faster and reduce bugs before execution.Metabase uses the markdown linter to improve the quality of their code and to filter out the junk.

**Performance Optimizations:**

Performance optimization, also known as “performance tuning”, is usually an iterative approach to making and then monitoring modifications to an application and its database. It could involve adjusting the configuration of the database and server, or making changes to the applications and the SQL that maintain and query the database. As authors of this book, we can't participate in the specific modify and monitor iterative processes being carried on by any of our readers and their IT organizations. But we can describe factors that are likely to apply to any asserted versionaning implementation.These factors include the number of users, the complexity of the application and the SQL, the volatility of the data, and the DBMS and server platform. The major DBMSs may optimize varying configurations differently, and may have extensions that can be used to simplify and improve a “plain vanilla” implementation of Asserted Versioning.

**Usage of Cache:**

In Metabase, rather than setting cache settings manually on a per-query basis, we give you two parameters to set to automatically cache the results of long queries: the minimum average query duration, and the cache TTL multiplier.

**Minimum query duration:**

Your Metabase instance keeps track of the average query execution times of your queries, and it will cache the results of all saved questions with an average query execution time longer than the number you put in this box (in seconds).

**Cache Time-to-live (TTL):**

Instead of setting an absolute number of minutes or seconds for a cached result to persist, Metabase lets you put in a multiplier to determine the cache’s TTL. Each query’s cache TTL is computed by multiplying its average execution time by the number you put in this box. So if you put in 10, a query that takes 5 seconds on average to execute will have its cache last for 50 seconds; and a query that takes 10 minutes will have a cached result lasting 100 minutes. This way, each query’s cache is proportional to its execution time.

**Max cache entry size:**

Lastly, you can set the maximum size of each question’s cache in kilobytes, to prevent them from taking up too much space on your server.

**Advanced caching controls:**

Question-specific caching is only available on Pro and Enterprise plans (both self-hosted and on Metabase Cloud).All Metabase editions include global caching controls. Some plans include additional caching options that let you control caching for each database, as well as individual questions.

**Caching per database:**

You can override your default caching options for each database connection, caching the results for more or less time than the default time-to-live (TTL) duration set by your site-wide settings. Setting caching per question is especially useful when data relevant to the question has a different natural cadence than your site-wide caching rule.

Go to Admin settings > Databases and select your database connection. Under Advanced settings, set the Default result cache duration, which determines how long to keep question results for that database. By default, Metabase will use the value you supply on the cache settings page, but if this database has other factors that influence the freshness of data, it could make sense to set a custom duration. You can also choose custom durations on individual questions or dashboards to help improve performance.

**Caching per question:**

You can override your default caching options for questions, caching the results for more or less time than the default time-to-live (TTL) duration set by your site-wide caching settings. Setting caching per question is especially useful when data relevant to the question has a different natural cadence than your site-wide caching rule, such as when the question queries data that doesn’t change often.

**Usage of Content delivery network:**

A content delivery network (CDN) refers to a geographically distributed group of servers which work together to provide fast delivery of Internet content.A CDN allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like Facebook, Netflix, and Amazon.A properly configured CDN may also help protect websites against some common malicious attacks, such as Distributed Denial of Service (DDOS) attacks.The mission of a CDN is to reduce latency. Latency is that annoying delay you experience when trying to access a web page or video stream before it fully loads on your device. Although measured in milliseconds, it can feel like forever, and may even result in a load error or time-out.Metabase is hosted using CSS and Java script in CDN.

**Lazy Loading:**

When it comes to dashboard performance in metabase, there are essentially four ways to get dashboards to load faster:

* Ask for less data.
* Cache answers to questions.
* Organize data to anticipate common questions.
* Ask efficient questions.

What follows is some general guidance for how to get your dashboards to load faster. The bulk of this guidance will focus on that third bullet, or how you can organize data to anticipate the most common questions that data will be used to answer.The usual caveats about premature optimization being the root of all evil apply. Our advice assumes that you have been exploring your data for some time, and are deriving material benefits from the insight that data yields. Only then should you be asking, “How do I get this dashboard to load faster?”

**Ask for less data:**

This point is almost too obvious that it often goes overlooked, but it should be the first place to start. Do you actually need the data you’re querying? And even if you do need all that data, how often do you need it?You can save a lot of time simply by restricting the data you query, such as by including adding a default filter on a dashboard. Be especially mindful of data spanning time and space: do you really need to look at the last quarter’s worth of data every day? Or do you really need every transaction for every country?And even if you do need to know that information, do you need it every day? Could you relocate that question to another dashboard that’s typically only reviewed weekly or monthly?We should be open to all our data when we’re exploring our datasets, but once we settle on the kinds of decisions our organization needs to make—and the data we need to inform those decisions we should be ruthless about excluding data that does not significantly improve our analysis.

**Cache answers to questions:**

You don’t need to wait for data if it’s already loaded. Admins can set up Metabase to cache query results, which will store answers to questions. If you have a set of dashboard that everyone runs when they open up their computers first thing in the morning, run that dashboard ahead of time, and the questions in that dashboard will use the saved results for subsequent runs to load in seconds. People will have the option to refresh the data, but typically this is unnecessary, as most often people will only need to review data from the previous day and before.

**Organize data to anticipate common questions:**

The next best thing you can do is organize your data in such a way that it anticipates the questions that will be asked, which will make it easier for your database to retrieve that data.

* Index frequently queried columns.
* Replicate your database.
* Denormalize data.
* Materialize views: create new tables to store query results.
* Aggregate data ahead of time with summary tables.
* Pull data out of JSON and slot its keys into columns.
* Consider a database specific to analytics.

Database tuned specifically to handle analytics, and should be your last resort, especially for startups.

**PWA:**

A PWA is a website with all the benefits of an app. PWAs give you a faster, more reliable, and more engaging version of your website or eCommerce store. PWAs can do most things that native apps can do, such as operate offline, access your camera and microphone if necessary, GPS, and more.

Metabase offers several types of embedding with different levels of customization and security.Full-app embedding is the only type of embedding that integrates with your permissions and SSO to give people the right level of access to query and drill-down into your data.If you only want to set up a fixed number of filters and drill-down views into your data (i.e., prevent people from creating their own questions), you might prefer Signed embedding.

Prerequisites

1. Make sure you have a license token for a paid plan.
2. Organize people into Metabase groups.
3. Set up permissions for each group.
4. Set up SSO to automatically apply permissions and show people the right data upon sign-in.

If you’re dealing with a multi-tenant situation, check out our recommendations for Configuring permissions for different customer schemas.

Enabling full-app embedding in Metabase

1. Go to Settings > Admin settings > Embedding.
2. Click Enable.
3. Click Full-app embedding.
4. Under Authorized origins, add the URL of the website or web app where you want to embed Metabase (such as https://\*.example.com).

**Setting up embedding on your website**

1. Create an iframe with a src attribute set to:
   * the URL of the Metabase page you want to embed, or
   * an authentication endpoint that redirects to your Metabase URL.
2. Optional: Depending on the way your web app is set up, set environment variables to:
   * Add your license token.
   * Embed Metabase in a different domain.
   * Secure your full-app embed.
3. Optional: Enable communication to and from the embedded Metabase using supported postMessage messages:
   * From Metabase
   * To Metabase
4. Optional: Set parameters to show or hide Metabase UI components.

Once you’re ready to roll out your full-app embed, make sure that people allow browser cookies from Metabase, otherwise they won’t be able to log in.

**Pointing an iframe to a Metabase URL**

Go to your Metabase instance and find the page that you want to embed.

For example, to embed your Metabase home page, set the src attribute to your site URL, such as:

http://metabase.yourcompany.com/

To embed a specific Metabase dashboard, use the dashboard’s URL, such as:

http://metabase.yourcompany.com/dashboard/1

**Pointing an iframe to an authentication endpoint**

Use this option if you want to send people directly to your SSO login screen (i.e., skip over the Metabase login screen with an SSO button), and redirect to Metabase automatically upon authentication.

**Local Storage and Service workers:**

To create an offline website, you’ll need to understand two fundamental web topics, service workers (sw) and browser storage. Let’s start with browser storage.Storing data in the browser.This is a double edged sword in some sort. Storing data in browsers has it’s pro’s and cons. Since web storage was introduced in HTML5, older browsers have limited support though plugins can be installed to make the browser compatible with the new technology. Before HTML5, application data had to be stored in cookies which had to be transferred between the client and server and had a very limited storage capacity (4095 bytes) as opposed to local-storage's 5mb (5,875,280 bytes).Local Storage data is persistent, unlike cookies which have an expiry date. To clear data from local storage, you’d be forced to manually clear the data from the application tab in your browser developer tools. However, your PC can automatically (meaning without your consent or permission) delete data from local storage when it’s running out of file space to make room for more files.Local Storage is the simplest way to store key-value pairs of data on the browser. Other methods are using IndexedDB (A tutorial on this in the making), WebSQL (Deprecated, not advisable to learn or use), File System API and Application cache.Metabase uses load balancing and straightforward horizontal and vertical scaling for local storage and service workers.