

# Cursorのおすすめ設定 & Cursorにデータ分析を任せる方法

馬渡 大樹

株式会社GENDA

2025-05-22

# 1. はじめに

## 1.1. 自己紹介

名前 馬渡 大樹 (Mawatari Daiki)

所属 株式会社GENDA - IT戦略部

職種 データエンジニア / 機械学習基盤エンジニア

GitHub [@i9wa4](#)

- 好き
  - ゲームセンター
  - Vim
  - Happy Hacking Keyboard
- 好きではない
  - Zero Configuration



IMG20240922183711 Trimmed image

## 1.2. 私の主なデータ系業務紹介

Cursor の用途を説明するために簡単に業務を紹介します

まとめると SQL を書くことが多いです

### 業務1

- DWH のデータ整備
- dbt (data build tool) により集計テーブルを SQL ファイルとして管理する
  - SQL ファイル主体のプロジェクトがあるという認識で OK

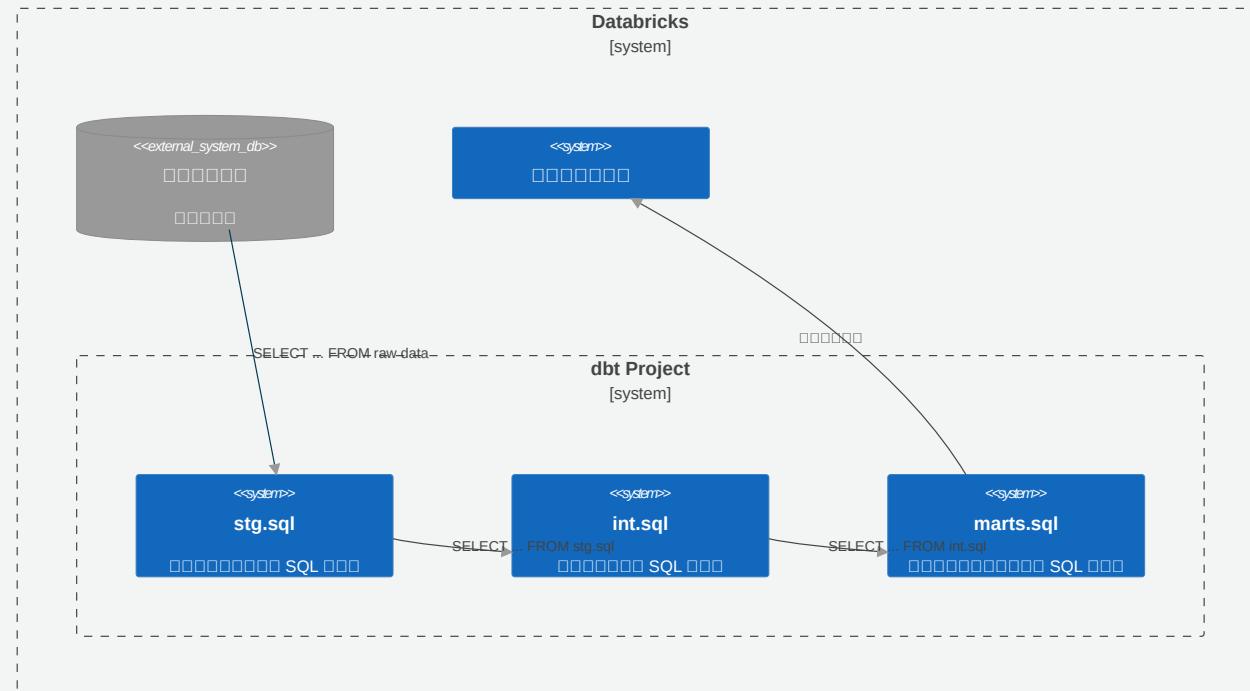
### 業務2

- データ可視化用ダッシュボード作成
- データ分析サポート

### 業務3

- データ基盤の改修

□□□□□□□□□



## 2. Cursor のおすすめ設定

## 2.1. Cursor 設定との向き合い方

Cursor のテキストエディタとしての設定をしっかりやるべきか？

私は No だと思う

- AI エージェントはプロジェクト設定を守ってくれるとは限らない
- 多様性のある Pull Request を許容できるように CI を整備したほうがよい

### Cursor の用途の割り切り

AI エージェント実行アプリとして使いやすくなるグローバル設定を紹介していきます

The screenshot shows a news article from BigGo. The title is "Microsoft が Cursor を含む VSCode フォークで C/C++ 拡張機能をブロック". The author is "BigGo Editorial Team" and the date is "2025-04-05 16:23 (GMT+9)". The article content discusses the blocking of the Cursor extension in a fork of VSCode for C/C++ development. It includes a screenshot of a GitHub issue page titled "Has the VSCode C/C++ Extension been blocked? #2976". The issue has 1.8k forks and 39.2k stars. A comment from a user named "rektbuildr" is shown, stating "While coding just now I got this msg". The issue is marked as "Open".

Extensions Blocked image

Sad News 😢

Microsoft が Cursor を含む VSCode フォークで C/C++ 拡張機能をブロック - BigGo ニュース

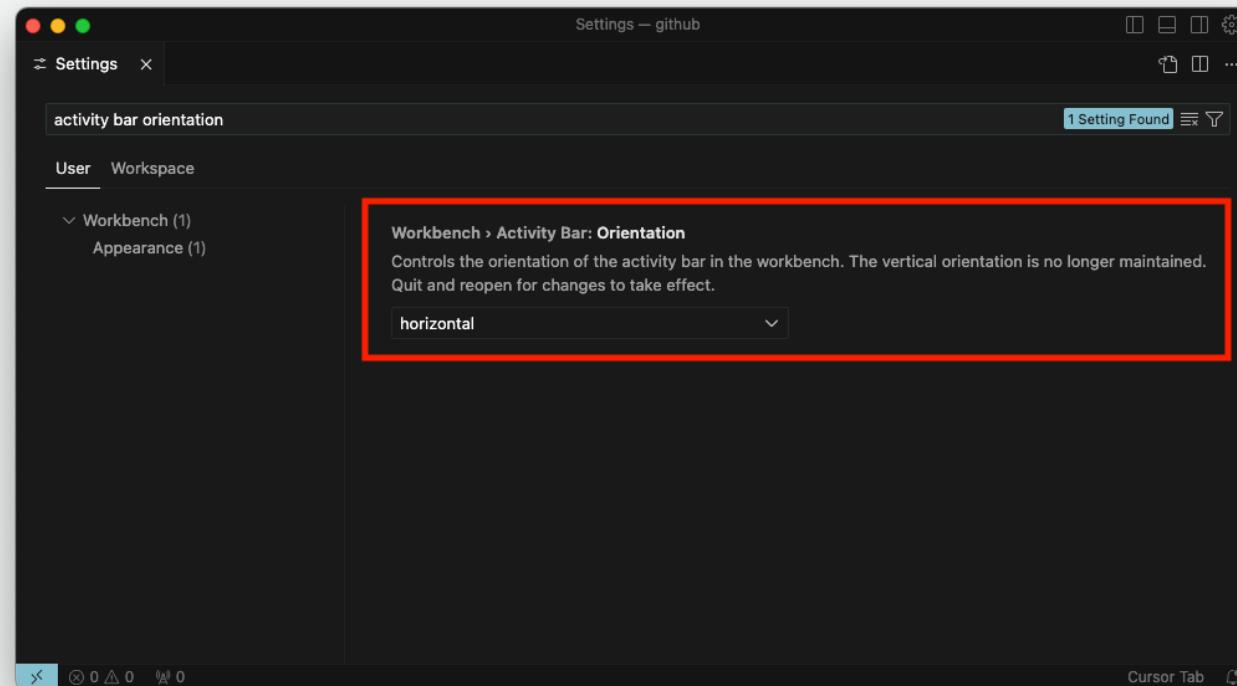
## 2.2. settings.json に反映される設定 (1)

F1 > Preferences: Open Settings

VS Code に存在する左端のバーを表示させる

The vertical orientation is no longer maintained.

使いやすくなりますが自己責任で



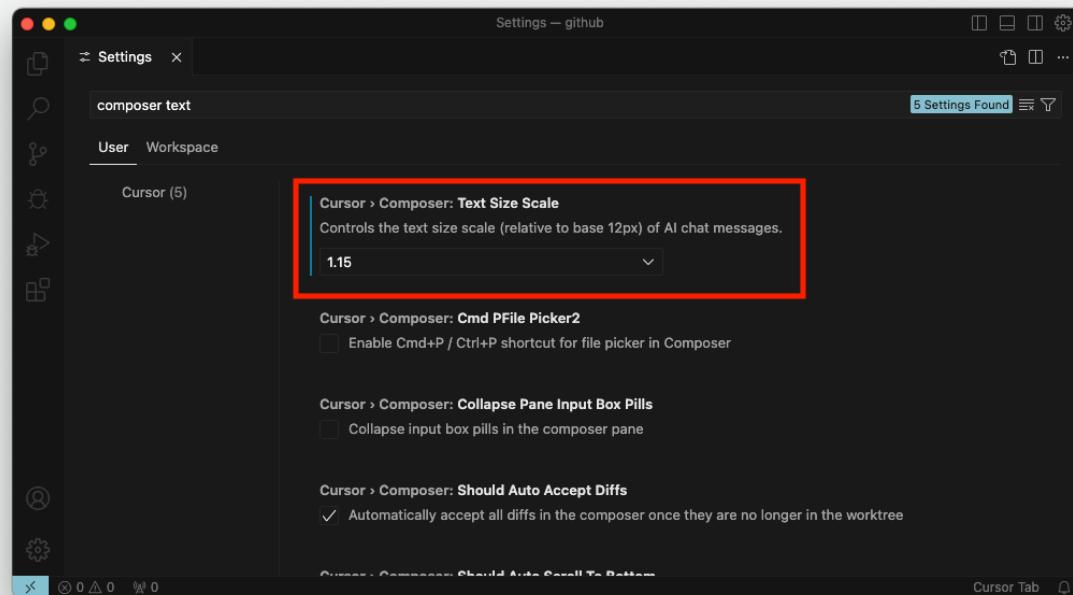
Cursor Settings Activitybar image

## 2.3. settings.json に反映される設定 (2)

F1 > Preferences: Open Settings

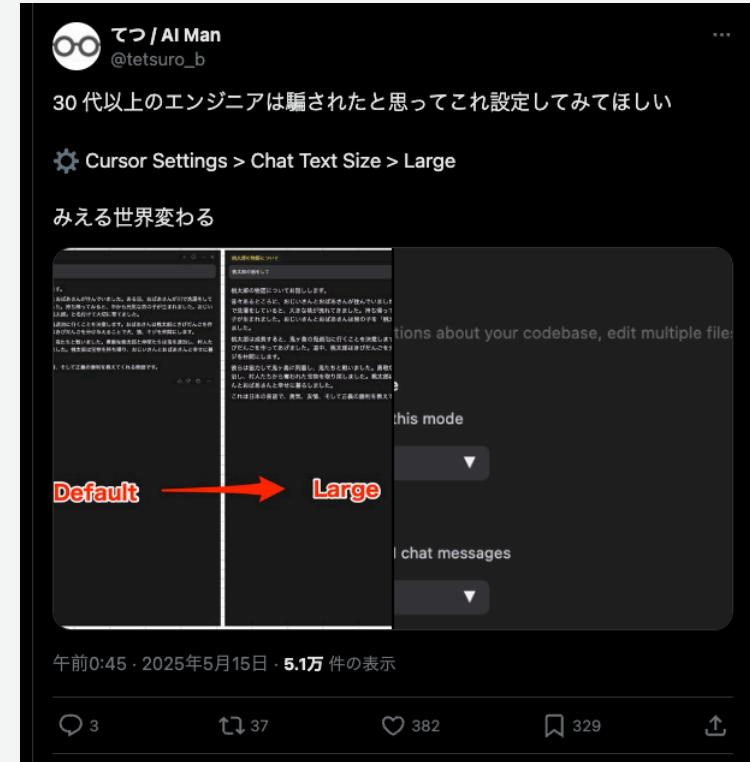
チャットメッセージのサイズを大きくする

チャット欄が Cursor の本体なのでデカくしましょう



Cursor Settings Chat Msg Size image

文字を大きくする決意をさせてくれたポスト  
ポストの内容通り右上歯車からも設定できる



Cursor Settings Chat Msg Size X image

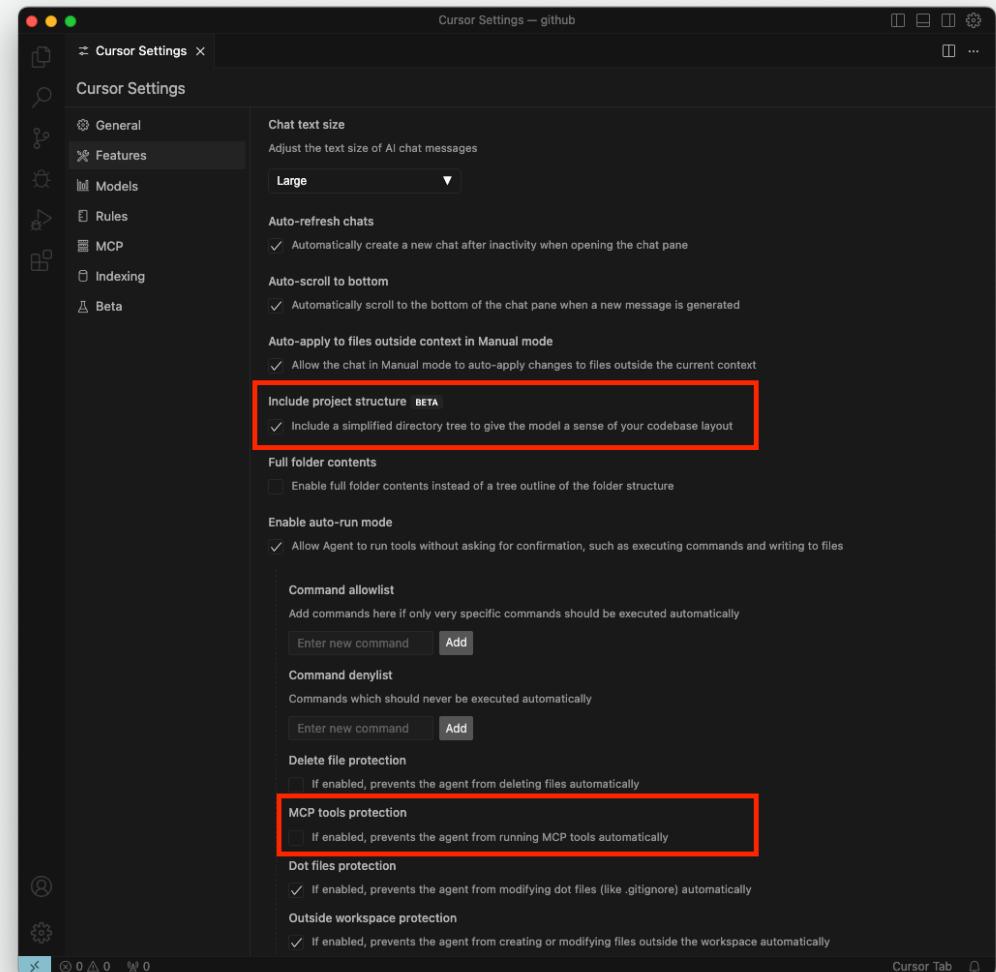
[https://x.com/tetsuro\\_b/status/1922679755863970196](https://x.com/tetsuro_b/status/1922679755863970196)

## 2.4. 右上歯車アイコンから行う設定

見ておくべき項目

- General
  - Privacy mode: **Enabled**
    - ポリシーで設定変更禁止になつていれば安心
- Features
  - Include project structure [BETA]: **ON**
    - 効果は実感できていないが ON
  - Enable auto-run mode: **ON** ! 要注意
    - MCP tools protection: **OFF** ! 要注意

auto-run mode は自分の用途や権限に合つた範囲内で最大限許容すると効率アップ！



Cursor Settings Up Right Features image

## 2.5. 外部設定

### グローバルな gitignore 設定

リポジトリ内に自己と Cursor の作業用ディレクトリを作成する

```
~/.config/git/ignore
1 .i9wa4/
```

~/.gitignore\_global ではなくデフォルト設定に従ったほうが  
Git と仲良くなれる

まだ .gitconfig に core.excludesfile を設定しているの？ #Git -  
Qiita



Gitignore X image

ポスト内容と用途が違うが Cursor に文脈を与えたりアウトプット  
させるのに便利であった

<https://x.com/mizchi/status/1914543131888066561>

## 2.6. MCP 設定 (基本編)

DWH 向けの MCP Server を利用しクエリを実行できるようにする

参考

<https://github.com/RafaelCartenet/mcp-databricks-server>

```
mcp.json
1  {
2      "mcpServers": {
3          "databricks": {
4              "command": "uv",
5              "args": [
6                  "--directory",
7                  "~/ghq/github.com/RafaelCartenet/mcp-databricks-server",
8                  "run",
9                  "main.py"
10             ]
11         }
12     }
13 }
```

実行の様子



The screenshot shows a Databricks notebook interface. At the top, there are two tabs: "Databricks スキーマのテーブル一覧取得" and "Databricks スキーマのテーブル一覧取得". Below them is a text input field with the placeholder "1つのテーブルを選んでレコード数を教えて". Underneath, a section titled "Called MCP tool execute\_sql\_query" is expanded, showing the SQL query: "sql": "SELECT COUNT(\*) AS record\_count;". The result of the query is displayed below, showing a single row with "record\_count" and its value "627".

```
Called MCP tool execute_sql_query
sql: "SELECT COUNT(*) AS record_count;
}
Result:
record_count
-----
627
```

Cursor Mcp Databricks Query image

## 2.7. MCP 設定 (非公式 MCP サーバーとの向き合い方)

前頁で Databricks MCP Server を紹介しましたが非公式 MCP Server です

### 非公式 MCP サーバーとの向き合い方

1. 実装を理解した上でコミットハッシュ指定して使う

- コード量は少なめだし DeepWiki を使えば壁打ちもできる

2. 自分で作る

- 私は Fork して機能追加します

- Databricks MCP Server を Service Principal 認証対応させた

### 心得

- 自分の手足の延長となるツールなので自分で可否判断できるものだけ使いましょう
- 無批判に何でも使うと最終的には不自由な世界になります

## 2.8. MCP 設定 (認証情報隠蔽)

⚠️ 未検証の内容です

プロジェクト共通設定として `.cursor/mcp.json` を置きたい場合に備えて MCP 設定から認証情報を隠蔽するために `envmcp` が利用できます

<https://github.com/griffithsbs/envmcp>

### Before

```
1 {
2   "my_database": {
3     "command": "start-my-mcp-server",
4     "args": [
5       "my secret connection string",
6     ],
7   },
8   "my_other_mcp_server": {
9     "command": "start-my-other-mcp-server",
10    "env": {
11      "MY_API_KEY": "my api key"
12    }
13  }
14 }
```

### After

```
1 {
2   "my_database": {
3     "command": "npx",
4     "args": [
5       "envmcp",
6       "start-my-mcp-server",
7       "$MY_DATABASE_CONNECTION_STRING",
8     ],
9   },
10  "my_other_mcp_server": {
11    "command": "npx",
12    "args": [
13      "envmcp",
14      "start-my-other-mcp-server",
15    ],
16  }
17 }
```

## 2.9. ルール設定

⚠️ いずれも未検証の内容です

プロジェクトに対してルールを設定できると便利

```
1 project/
2   .cursor/rules/      # Project-wide rules
3 backend/
4   server/
5     .cursor/rules/    # Backend-specific rules
6 frontend/
7   .cursor/rules/      # Frontend-specific rules
```

<https://docs.cursor.com/context/rules>

Devin 向けにも別途ルールを整備する必要があり躊躇していました 😢

シンボリックリンクを張ればよいという発想

- Unified `.ai/` folder for all your project-wide AI rules (Markdown)
- Auto-generate:
  - `.cursor/rules/*.mdc`
  - `.cline-rules`
  - `.github/copilot-instructions.md`
  - `devin-guidelines.md`
- Symlink or copy mode (auto-detects OS capability)

<https://github.com/airulefy/Airulefy>

### 3. まとめ

### 3.1. Cursor の設定まとめとデータ分析向け構築

以下の整備をすることで Cursor が自律的に爆速でデータ分析を行えるようになります

今後はルール設定を調整していきたいです！

Cursor 設定 auto-run mode を利用する

---

MCP DWH の MCP Server を利用する

---

プロジェクト dbt プロジェクトを作業場とする

---

その他設定 gitignore で個人作業ディレクトリを用意する