# Article information:

Choosing a database solution and ORM for Node.js  
<https://medium.com/@aabedraba/choosing-a-database-solution-and-orm-for-node-js-6c256ced72ff>

# Article summary:

1. The project team chose to use React and Node.js for their frontend and backend, respectively, and TypeScript for increased robustness.

2. The team decided to use SQL as their database solution, and evaluated four ORMs (KnexJs, TypeORM, Sequelize and Prisma) based on type support, migrations, testing and documentation.

3. The team ultimately chose Prisma due to its excellent features in all categories; however they experienced some issues with the library's rapid changes over time.

# Article rating:

May be slightly imbalanced: The article presents the information in a generally reliable way, but there are minor points of consideration that could be explored further or claims that are not fully backed by appropriate evidence. Some perspectives may also be omitted, and you are encouraged to use the research topics section to explore the topic further.

# Article analysis:

The article is generally reliable in terms of its content; it provides a detailed overview of the process the project team went through when selecting an ORM for their Node.js application. It also provides a comprehensive comparison between the four ORMs evaluated by the team (KnexJs, TypeORM, Sequelize and Prisma), which is useful for readers who are considering similar solutions for their own projects.

However, there are some potential biases present in the article that should be noted. For example, the author states that they had "a lot of love" for TypeScript before trying it out - this could indicate a bias towards using TypeScript over plain Javascript from the start of the project. Additionally, while the author does mention some issues with Prisma (such as its rapid changes over time), they do not provide any counterarguments or alternative solutions that could have been used instead - this could be seen as promotional content in favour of using Prisma over other options available on the market.

In conclusion, while this article is generally reliable in terms of its content and provides useful information about selecting an ORM for Node.js applications, readers should be aware of potential biases present in the article that may influence their decision-making process when choosing an ORM solution for their own projects.

# Topics for further research:

* Node.js ORM comparison
* Pros and cons of TypeScript
* Alternatives to Prisma
* ORM selection criteria
* ORM performance evaluation
* ORM scalability considerations

# Report location:

<https://www.fullpicture.app/item/c253452453ddd854006c41e5c1ae09b9>