Eugene Wesonga bonus question

1.1. Database Server Comparison:

Create a table comparing the following database server options (focusing on its role as a web application environment):

\* MySQL

\* PostgreSQL

\* Microsoft SQL Server

\* Oracle Database

\* Node.js

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FEATURE/ASPECT** | **MySQL** | **PostgreSQL** | **Microsoft SQL server** | **Oracle database** | **Node.js** |
| Strengths | MySQL is known for its speed and reliability, especially for ready-heavy workloads. It has a large community and it is easy to set up, making it suitable for small to medium-sized web applications | PostgreSQL offers strong support for complex queries, transactions, and extensions. It is ACID-compliant and suitable for applications requiring complex data operations | Provides robust tools for backup, recovery, and data management. It integrates well with other Microsoft, making it suitable for enterprise environments | Known for high performance and reliability in large-scale enterprise environments. It offers advanced security features and supports complex data models | Node.js is not a database but a JavaScript runtime used for building server-side applications. It can connect to various enterprises including MySQL, PostgreSQL, MongoDB etc. via different modules. Node.js excels in handling concurrent connections efficiently |
| Performance | Fast for Read-heavy workloads | First for complex queries and Write-heavy | High performance with large data sets | High performance for Enterprise workloads | Not a database server |
| Cloud support | MySQL Cloud service | Various cloud providers | Azure SQL | Oracle SQL | Using Node.js Modules |
| type | Relational Database Management system | Relational Database Management system | Relational Database Management system | Relational Database Management system | JavaScript Runtime Environment |
|  |  |  |  |  |  |