MODULE 10

Praneesh Sharma

21052264

**Relational database:** A collection of datasets organized as records and columns in tables. In a relational database system, relationships are defined between the database tables. Think of a relational database as a set of data with 1-to-1 and 1-to-many relationships. For example, a database of customers would match each customer with an identifier that uniquely identifies the customer. Developers use structured query language (SQL) to interact with the database.

**Amazon Relational Database Service (Amazon RDS):** Amazon RDS lets developers create and manage relational databases in the cloud. Amazon RDS lets developers track large amounts of data and organize and search through it efficiently.

**Amazon DynamoDB:** The AWS nonrelational database service. Data is stored in key-value pairs.

**Nonrelational Database:** Also called a "NoSQL" or "Not only SQL" database. Each entry is stored in a key-value pair in which each key is attached to values. Each entry can have a different number of values attached to a key.

**Amazon Redshift:** The AWS data-warehousing service that can store massive amounts of data in a way that makes it fast to query for business intelligence (BI) purposes.

**Online transaction processing (OLTP):** A category of data processing that is focused on transaction-oriented tasks. OLTP typically involves inserting, updating, or deleting small amounts of data in a database.

**Online analytic processing (OLAP):** A computing method that lets users efficiently and selectively extract and query data to analyze it from different points of view.

**Amazon Aurora:** A relational database engine compatible with MySQL and PostgreSQL, built for the cloud, combining the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open-source databases.

**MySQL:** An open-source relational database management system.

**OLTP and OLAP:**

There are two types of data processing: online transaction processing (OLTP) and online analytic processing (OLAP).

* ****OLAP****operations are primarily read-only; that is, they read the data and perform various types of aggregation such as sum, group, and sort. Relational database management systems have built-in functions for performing these types of operations. Because they are built in, they are done efficiently. OLAP systems are often used where the system is required to process a lot of related data. Companies often need to analyze a lot of data points that have occurred over a long period of time to determine trends and predict behaviors. This type of system doesn’t necessarily need to be a real-time system—it can run as a background process.
* ****OLTP****operations, however, need to update the database in addition to reading it. Updating can involve adding, changing, or deleting values. Updating can become complex because many of the tables in a relational database are virtual. That is, the tables need to be combined in real time from nonvirtual tables.  OLTP systems are often used where the system is required to handle large volumes of transactions at a high rate. Many ecommerce systems, such as shopping carts, sell a large number of items during the checkout process while simultaneously removing the items from the inventory table. When the integrity of the entire transaction is critical, and when processing needs to happen in near-real time, companies should consider OLTP systems. OLTP systems are not exclusively relational databases, even though there are relationships in the data. It’s becoming more common for nonrelational databases to enforce constraints and enable transactions, so that these databases can be used as OLTP systems.

|  |  |
| --- | --- |
| **OLTP** | **OLAP** |
| Handles recent operational | Handles all historical data |
| Size is smaller | Size is larger |
| Goal is to perform day-to-day operations | Goal is to make decisions from large data sources |
| Uses simple queries | Uses complex queries |
| Faster processing speeds | Slower processing speeds |
| Requires read/write operations | Requires only read operations |

|  |  |
| --- | --- |
| **Applications of OLTP:** | **Applications of OLAP** |
| * Entering orders online * Processing purchases * Storing customer details | * Analyzing shopping patterns to make recommendations * Tracking purchasing trends for targeted advertisement * Analyzing seasonal buying trends to make sure items are in stock |

**AWS database services:**

****Amazon RDS****is the classic relational database that uses SQL, Oracle, Aurora, or other similar database systems. Businesses can use code to search for specific data based on the information in the rows and columns. Amazon RDS is useful for companies that are storing a moderate amount of data that is uniform in structure, meaning each unique ID is attached to the same number of data points. Amazon RDS is primarily used for OLTP because it has better methods for maintaining the integrity and consistency of the database when processing data.

****DynamoDB**** is a nonrelational database, meaning that you can’t use traditional systems such as SQL or Aurora. Each item in the database is stored as a key-value pair or a JavaScript Object Notation (JSON) file. This means that each row can have a different number of columns. The entries do not all have to be matched in the same way. This permits flexibility in processing that works well for blogging, gaming, and advertising.

****Aurora**** is a relational database engine that is specifically made to work with the AWS Cloud. Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases. It is designed to provide the security, availability, and reliability of commercial databases at one-tenth the cost. Aurora is fully managed by Amazon RDS, which automates time-consuming administrative tasks such as hardware provisioning, database setup, patching, and backups.

****Amazon Redshift**** is a fast, fully managed data warehouse that makes it efficient and cost effective to analyze all your data using standard SQL and your existing BI tools.

**Lab 10: Creating an Amazon RDS Database Instance**

1. Services > Database > RDS
2. Create a database through ‘Easy Create’
3. Modify the database to make it publicly accessible and apply all the changes immediately
4. Choose the name of the security group of your database and choose the inbound rules tab
5. Add an inbound rule ot type MSSQL and your workstation IPv4 address as the source
6. Open the Microsoft SQL Server Management Studio and connect it with your database instance
7. The databse can now be exploredz