Amazon DynamoDB

What is DynamoDB?

- Amazon DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS).
- It offers seamless scalability, high performance, and low latency for applications requiring fast and flexible data storage.

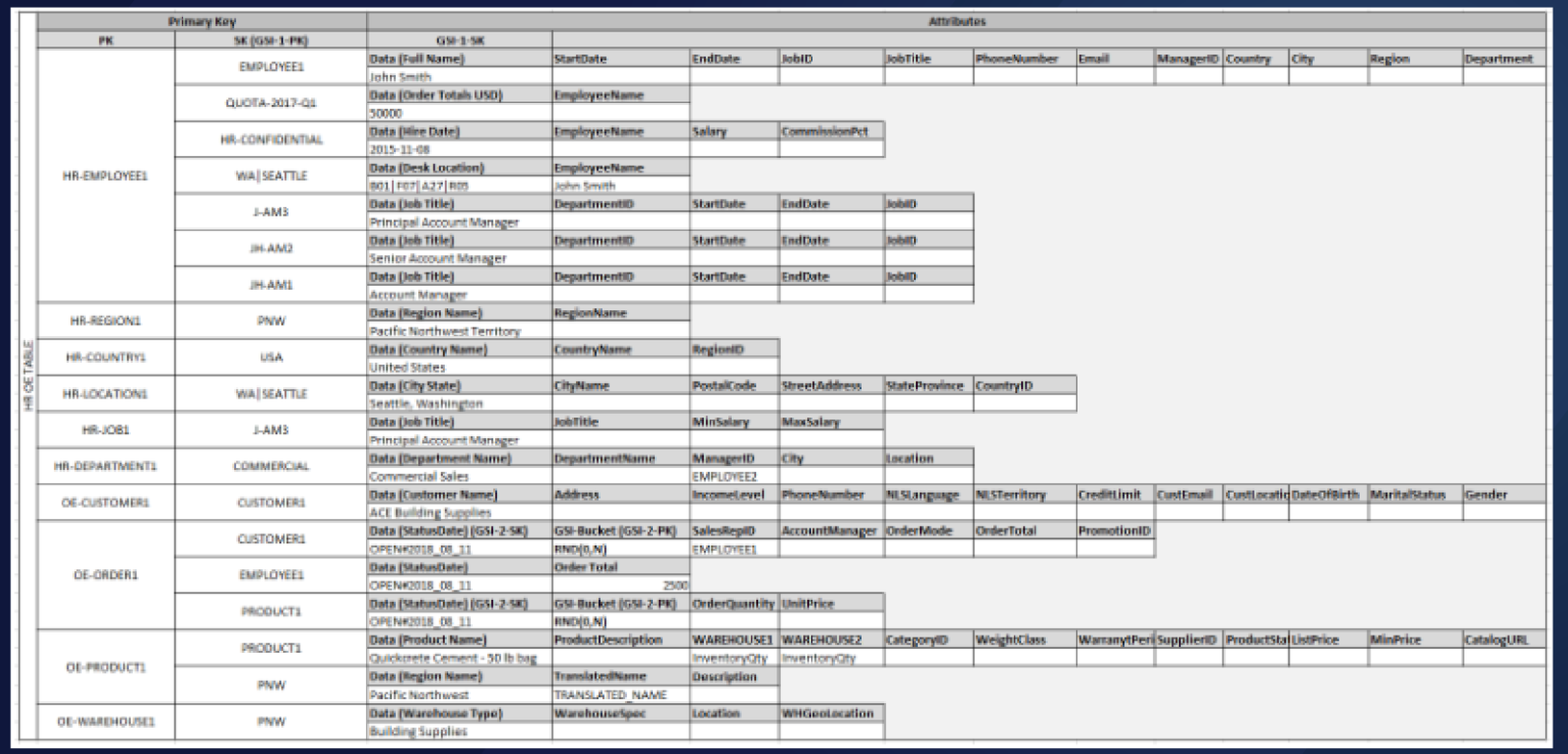
Overview of NoSQL Databases and DynamoDB's Position

- NoSQL databases are designed for handling large volumes of unstructured or semi-structured data.
- DynamoDB is positioned as a key player in the NoSQL landscape, offering features tailored for modern application needs.

Key Features and Benefits

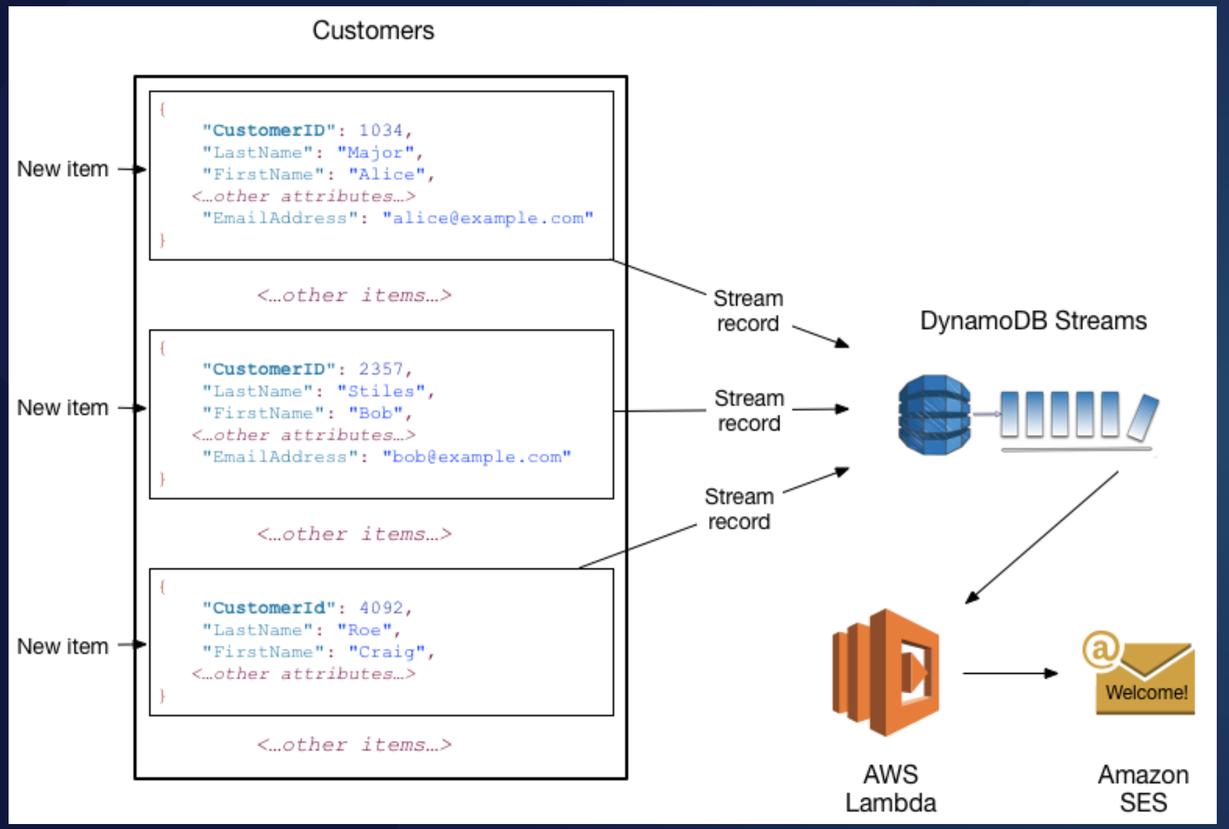
- **Managed Service:** AWS handles infrastructure provisioning and maintenance, allowing developers to focus on application development.
- **Scalability**: Dynamically scales to accommodate growing workloads without downtime.
- **Performance**: Provides consistent, single-digit millisecond latency for real-time data access.
- **Flexible Data Model**: Supports both key-value and document data models, enabling versatile data storage.
- **High Availability**: Replicates data across multiple Availability Zones for enhanced reliability.
- **Security**: Offers encryption at rest and in transit, along with fine-grained access control features.

Data Model



HR OE TABLE

Primary Key		Attributes											
PK	SK (GSI-1-PK)	GSI-1-SK											
HR-EMPLOYEE1	EMPLOYEE1	Data (Full Name)	StartDate	EndDate	JobID	JobTitle	PhoneNumber	Email	ManagerID	Country	City	Region	Department
		John Smith											
	QUOTA-2017-Q1	Data (Order Totals USD)	EmployeeName										
		50000											
	HR-CONFIDENTIAL	Data (Hire Date)	EmployeeName	Salary	CommissionPct								
		2015-11-08											
	WA|SEATTLE	Data (Desk Location)	EmployeeName										
		801|F07|A27|R05	John Smith									
	J-AM3	Data (Job Title)	DepartmentID	StartDate	EndDate	JobID							
		Principal Account Manager											
	JH-AM2	Data (Job Title)	DepartmentID	StartDate	EndDate	JobID							
		Senior Account Manager											
	JH-AM1	Data (Job Title)	DepartmentID	StartDate	EndDate	JobID							
		Account Manager											
HR-REGION1	PNW	Data (Region Name)	RegionName										
		Pacific Northwest Territory											
HR-COUNTRY1	USA	Data (Country Name)	CountryName	RegionID									
		United States											
HR-LOCATION1	WA|SEATTLE	Data (City State)	CityName	PostalCode	StreetAddress	StateProvince	CountryID						
		Seattle, Washington											
HR-JOB1	J-AM3	Data (Job Title)	JobTitle	MinSalary	MaxSalary								
		Principal Account Manager											
HR-DEPARTMENT1	COMMERCIAL	Data (Department Name)	DepartmentName	ManagerID	City	Location							
		Commercial Sales		EMPLOYEE2									
OE-CUSTOMER1	CUSTOMER1	Data (Customer Name)	Address	IncomeLevel	PhoneNumber	NLSLanguage	NLSTerritory	CreditLimit	CustEmail	CustLocatic	DateOfBirth	MaritalStatus	Gender
		ACE Building Supplies											
OE-ORDER1	CUSTOMER1	Data (StatusDate) (GSI-2-SK)	GSI-Bucket (GSI-2-PK)	SalesRepID	AccountManager	OrderMode	OrderTotal	PromotionID					
		OPEN#2018_08_11	RND(0,N)	EMPLOYEE1									
	EMPLOYEE1	Data (StatusDate)	Order Total										
		OPEN#2018_08_11	2500										
	PRODUCT1	Data (StatusDate) (GSI-2-SK)	GSI-Bucket (GSI-2-PK)	OrderQuantity	UnitPrice								
		OPEN#2018_08_11	RND(0,N)										
OE-PRODUCT1	PRODUCT1	Data (Product Name)	ProductDescription	WAREHOUSE1	WAREHOUSE2	CategoryID	WeightClass	WarrantyPeri	SupplierID	ProductSta	ListPrice	MinPrice	CatalogURL
		Quickcrete Cement - 50 lb bag		InventoryQty	InventoryQty								
	PNW	Data (Region Name)	TranslatedName	Description									
		Pacific Northwest	TRANSLATED_NAME										
OE-WAREHOUSE1	PNW	Data (Warehouse Type)	WarehouseSpec	Location	WHGeoLocation								
		Building Supplies											


Customers
New item
{
"CustomerID": 1034,
"LastName": "Major",
"FirstName": "Alice",
<…other attributes…>
"EmailAddress": "alice@example.com"
}
<…other items…>
New item
{
"CustomerID": 2357,
"LastName": "Stiles",
"FirstName": "Bob",
<…other attributes…>
"EmailAddress": "bob@example.com"
}
<…other items…>
New item
{
"CustomerId": 4092,
"LastName": "Roe",
"FirstName": "Craig",
<…other attributes…>
}
<…other items…>
Stream record
Stream record
Stream record
DynamoDB Streams
Welcome!
AWS Lambda
Amazon SES

Overview of DynamoDB's Flexible Data Model

- DynamoDB offers a schema-less design, allowing for flexible and dynamic data structures.
- Data is organized into tables, each containing multiple items.

Explanation of Primary Concepts

Tables:

- DynamoDB stores data in tables, similar to how data is stored in tables in relational databases.
- Each table contains a collection of items.

Items:

- Items are the fundamental unit of data in DynamoDB.
- An item is a collection of attributes, similar to a row in a relational database table.

Attributes:

- Attributes are key-value pairs that represent data within an item.
- Attributes can be of various data types, including strings, numbers, binary data, lists, and maps.

Primary Keys:

- Every DynamoDB table must have a primary key.
- The primary key uniquely identifies each item in the table.

It consists of one or two attributes:

- **Partition Key**: A single attribute that uniquely identifies each item in the table.
- **Composite Key**: A combination of two attributes: a partition key and a sort key. Together, they uniquely identify each item, with the partition key determining the partition where the item is stored and the sort key determining the order of items within the partition.

Comparison with Relational Databases

- Unlike relational databases, DynamoDB does not require a fixed schema.
- DynamoDB's schema-less design allows for more flexible data structures.
- While relational databases use tables, rows, and columns, DynamoDB uses tables, items, and attributes.

Scalability and Performance

Horizontal Scalability:

- DynamoDB scales horizontally to handle large workloads by distributing data and traffic across multiple servers.

Adaptive Capacity:

- DynamoDB automatically adjusts throughput capacity to accommodate changing workload patterns, ensuring consistent performance.

Provisioned Throughput vs. On-Demand Capacity:

- Provisioned Throughput allows you to specify read and write capacity units to meet your performance requirements.
- On-Demand Capacity automatically scales throughput capacity in response to traffic fluctuations, with no need for capacity planning.

Data Consistency and Durability

Strongly Consistent Reads vs. Eventually Consistent Reads:

- Strongly Consistent Reads provide the most up-to-date data but may incur higher latency.
- Eventually Consistent Reads may return stale data but offer lower latency.

Data Durability and Replication:

- DynamoDB ensures data durability by replicating data across multiple Availability Zones within a region.
- This replication enhances both availability and durability of data.

Security and Access Control

Encryption at Rest and in Transit:

- DynamoDB offers encryption of data both at rest and in transit to protect data integrity and confidentiality.

IAM Integration:

- Integration with AWS Identity and Access Management (IAM) enables granular control over access to DynamoDB resources.

Fine-Grained Access Control:

- IAM policies allow you to define fine-grained access control, specifying who can access specific DynamoDB resources and what actions they can perform.

Integration and Ecosystem

AWS SDKs:

- AWS provides SDKs for various programming languages, facilitating integration with DynamoDB in applications.

Integration with Other AWS Services:

- DynamoDB seamlessly integrates with other AWS services such as AWS Lambda, Amazon S3, and more, enabling powerful and scalable applications.

DynamoDB Streams:

- DynamoDB Streams capture changes to items in a DynamoDB table, enabling reactive and event-driven architectures.

Use Cases

- Real-time analytics
- Gaming
- IoT
- Mobile and web applications

References:

Official Amazon DynamoDB Documentation:
Amazon Web Services. (n.d.). Amazon DynamoDB documentation. Retrieved from hello@reallygreatsite.com

Book on NoSQL Databases:
Sadalage, P. J., & Fowler, M. (2012). NoSQL distilled: A brief guide to the emerging world of polyglot persistence (1st ed.). Addison-Wesley.

Research Paper on DynamoDB Performance:
Wiesmann, M., Gubri, K., & Puchner, T. (2018). Performance Evaluation of Amazon DynamoDB. In Proceedings of the 2018 International Conference on Management of Data (pp. 1421–1424). Association for Computing Machinery. Sadalage, P. J., & Fowler, M. (2012). NoSQL distilled: A brief guide to the emerging world of polyglot persistence (1st ed.). Addison-Wesley.

References:

"5 Use Cases for DynamoDB" from rockset.com:

Rockset. (n.d.). 5 Use Cases for DynamoDB. Rockset. Retrieved from https://rockset.com/blog/5-use-cases-for-dynamodb/