

AWS Databases Modernization Day



DynamoDB 활용하기

강민석

Sr. Database Specialist SA



강연 중 질문하는 방법

▼ Questions

☒ Show Answered Questions

Question	Asker

Type answer here

Go to Webinar “Chat/채팅” 창에 자신이 질문한 내역이 표시됩니다. 기본적으로 모든 질문은 공개로 답변 됩니다만 본인만 답변을 받고 싶으면 (비공개)라고 하고 질문해 주시면 됩니다.

고지 사항(Disclaimer)

본 콘텐츠는 고객의 편의를 위해 AWS 서비스 설명을 위해 온라인 세미나용으로 별도로 제작, 제공된 것입니다. 만약 AWS 사이트와 콘텐츠 상에서 차이나 불일치가 있을 경우, AWS 사이트(aws.amazon.com)가 우선합니다. 또한 AWS 사이트 상에서 한글 번역문과 영어 원문에 차이나 불일치가 있을 경우(번역의 지체로 인한 경우 등 포함), 영어 원문이 우선합니다.

AWS는 본 콘텐츠에 포함되거나 콘텐츠를 통하여 고객에게 제공된 일체의 정보, 콘텐츠, 자료, 제품(소프트웨어 포함) 또는 서비스를 이용함으로써 인하여 발생하는 여하한 종류의 손해에 대하여 어떠한 책임도 지지 아니하며, 이는 직접 손해, 간접 손해, 부수적 손해, 징벌적 손해 및 결과적 손해를 포함하되 이에 한정되지 아니합니다.



Table of contents

- DynamoDB 이해하기
- DynamoDB 실습



DynamoDB 이해하기



Data Platform portfolio

Database, Business Intelligence & Machine Learning



QuickSight



SageMaker

Analytics



Redshift
Data
warehousing



EMR
Hadoop +
Spark



Athena
Interactive
analytics



**Elasticsearch
Service**
Operational
Analytics



**Kinesis Data
Analytics** Real
time

Databases



Aurora
MySQL,
PostgreSQL



DynamoDB
Key value



DocumentDB
Document



RDS
MySQL, PostgreSQL,
MariaDB, Oracle, SQL
Server



ElastiCache
Redis, Memcached



QLDB
Ledger
Database



RDS on VMware



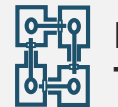
Neptune
Graph



Timestream
Time Series



**Managed
Blockchain**



**Blockchain
Templates**

Data Lake



S3/Glacier



Lake Formation
Data Lakes



Glue
ETL & Data
Catalog

Data Movement

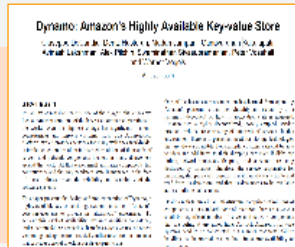
Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams



Amazon에서의 NoSQL 역사

Dec 2004:

Database
scalability
challenges

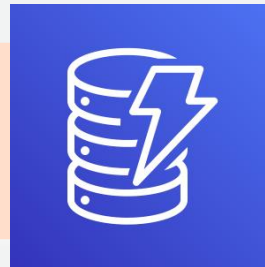


Oct 2007:

Dynamo paper
published

Jan 2012:

DynamoDB
general
availability



Q3 2016:

DynamoDB leader
in Gartner MQ,
Forrester Wave

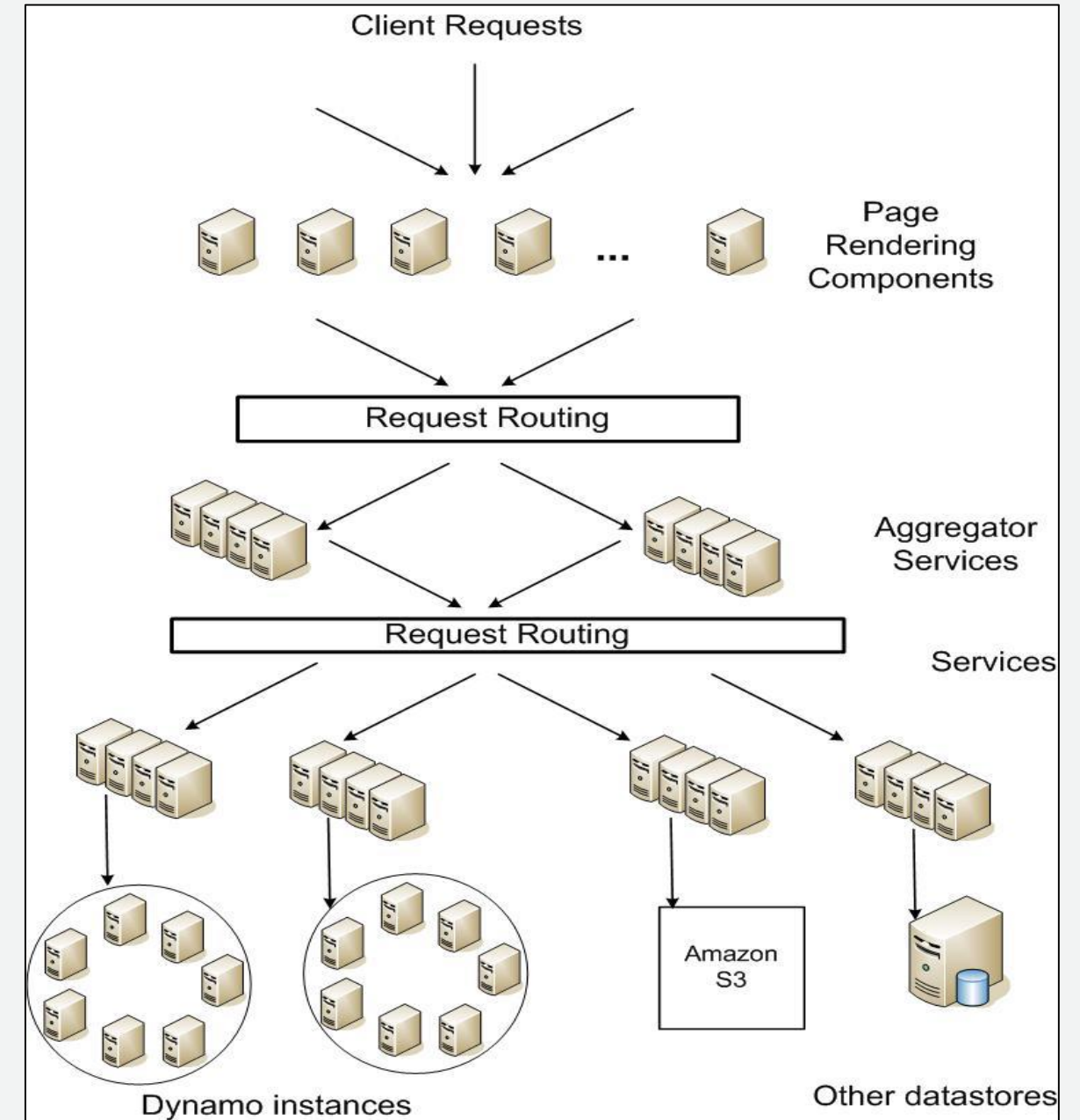
Today:

Tier 0 service
powering most of
Amazon



DynamoDB란 무엇인가?

- 2007년에 Amazon에서 “Dynamo Model”으로 처음 출시
- 2012년도에 DynamoDB로 서비스 됨
- Key-Value기반의 NoSQL Database
 - 성능: Low latency
 - 용량: Almost infinite capacity
 - 관리: No need to worry, Easy
 - 확장: 완벽한(Seamless) scalability
 - 고가용성: High Durability & Availability
 - 쉬운 Planning: (via throughput parameters)

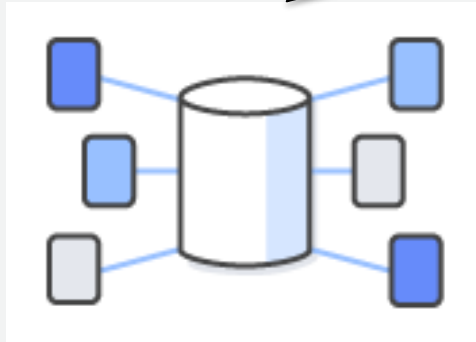


Service-oriented architecture of
Amazon's platform 



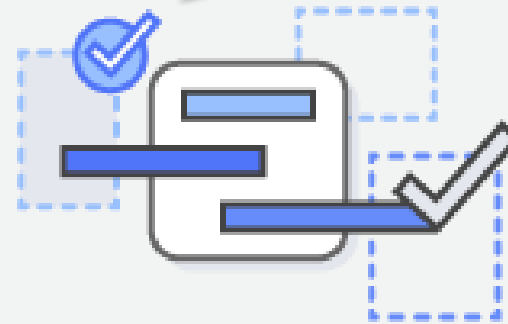
Amazon DynamoDB

3개의 AZ에 3 copy
복제



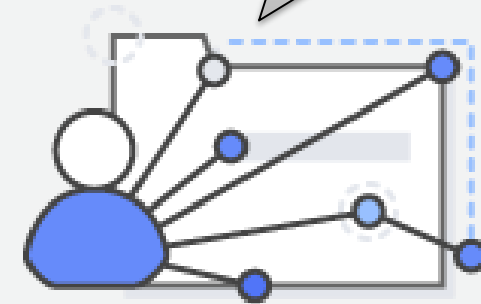
Highly available
and durable

한자리 milli-second
latency



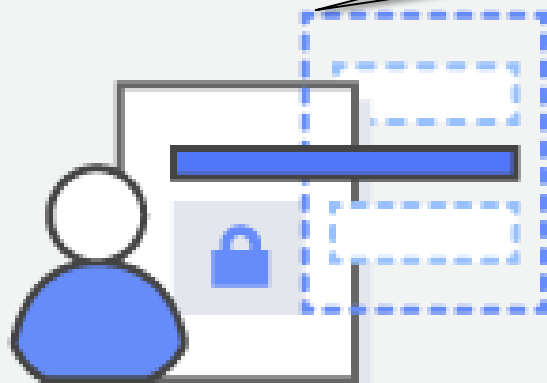
Consistently fast at any scale

Serverless로 관리
비용 절감



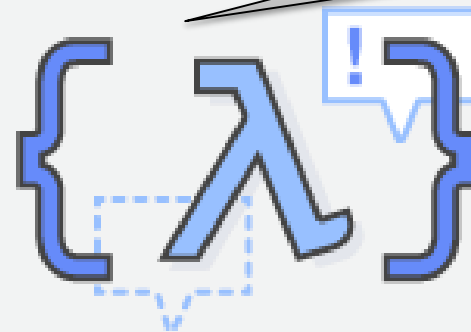
Fully managed

IAM 보안 사용



Secure

AWS 서비스와 통합



Integrates with AWS Lambda,
Amazon Redshift, and more

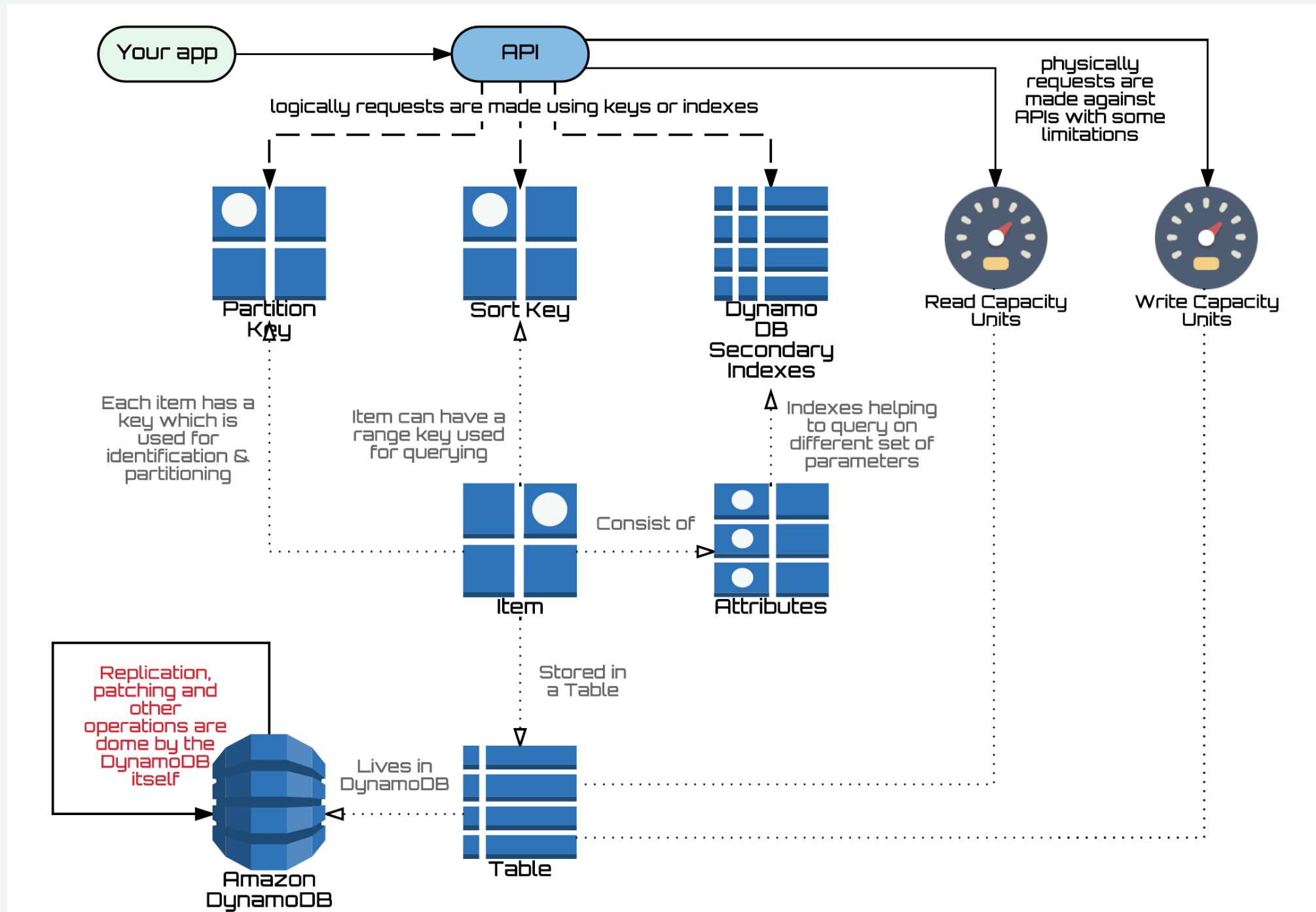
사용한 만큼 비용



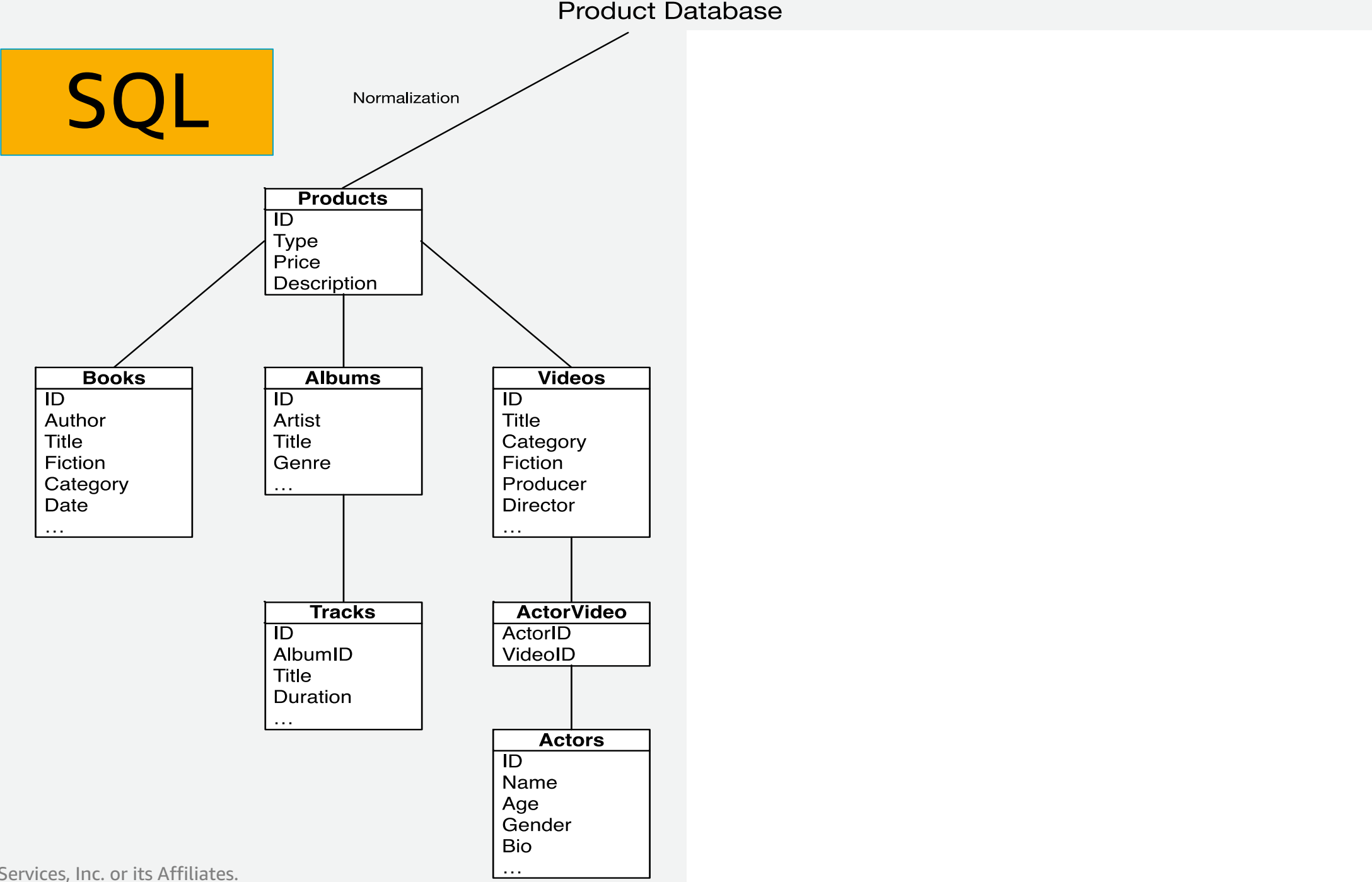
Cost-effective



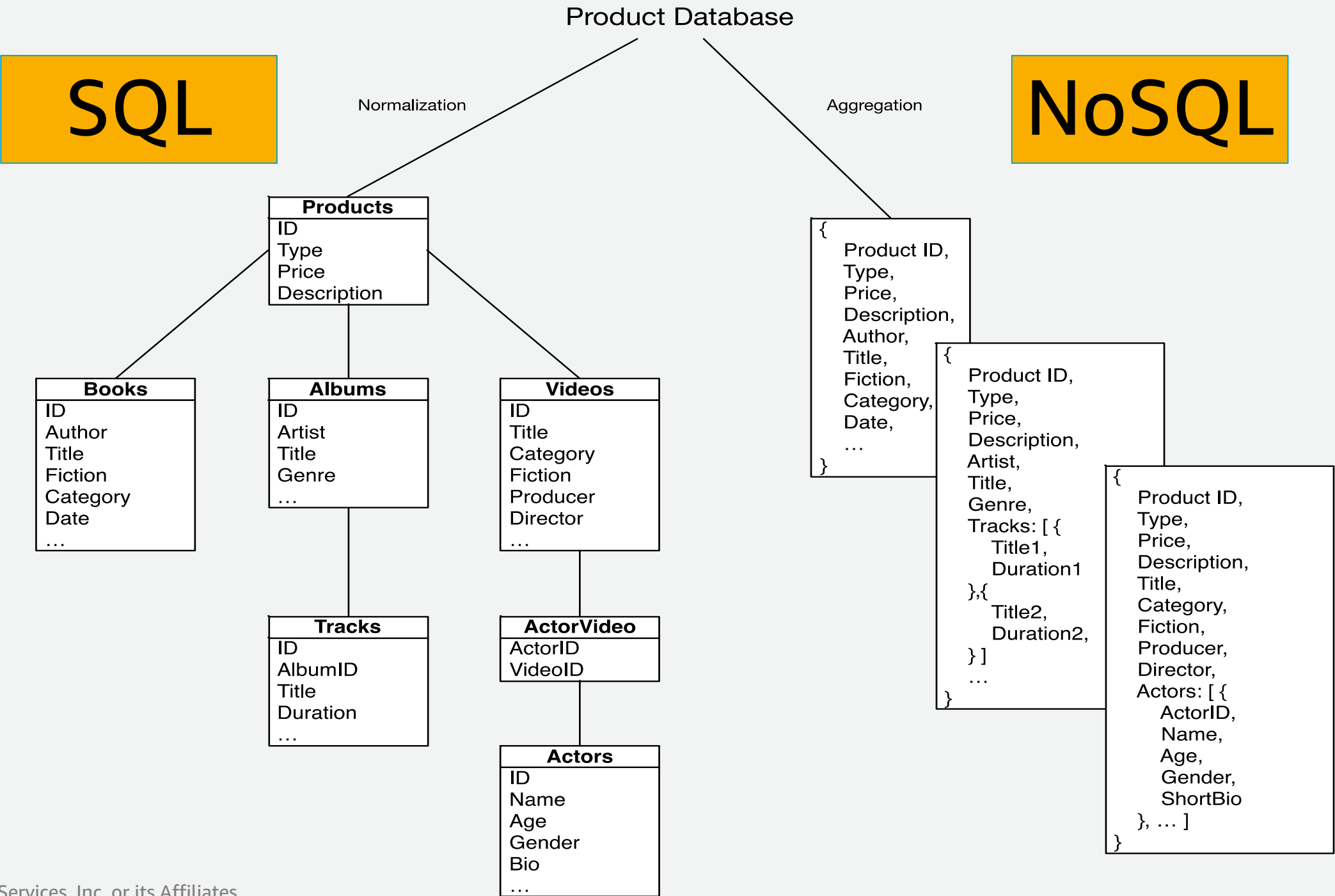
How DynamoDB works?



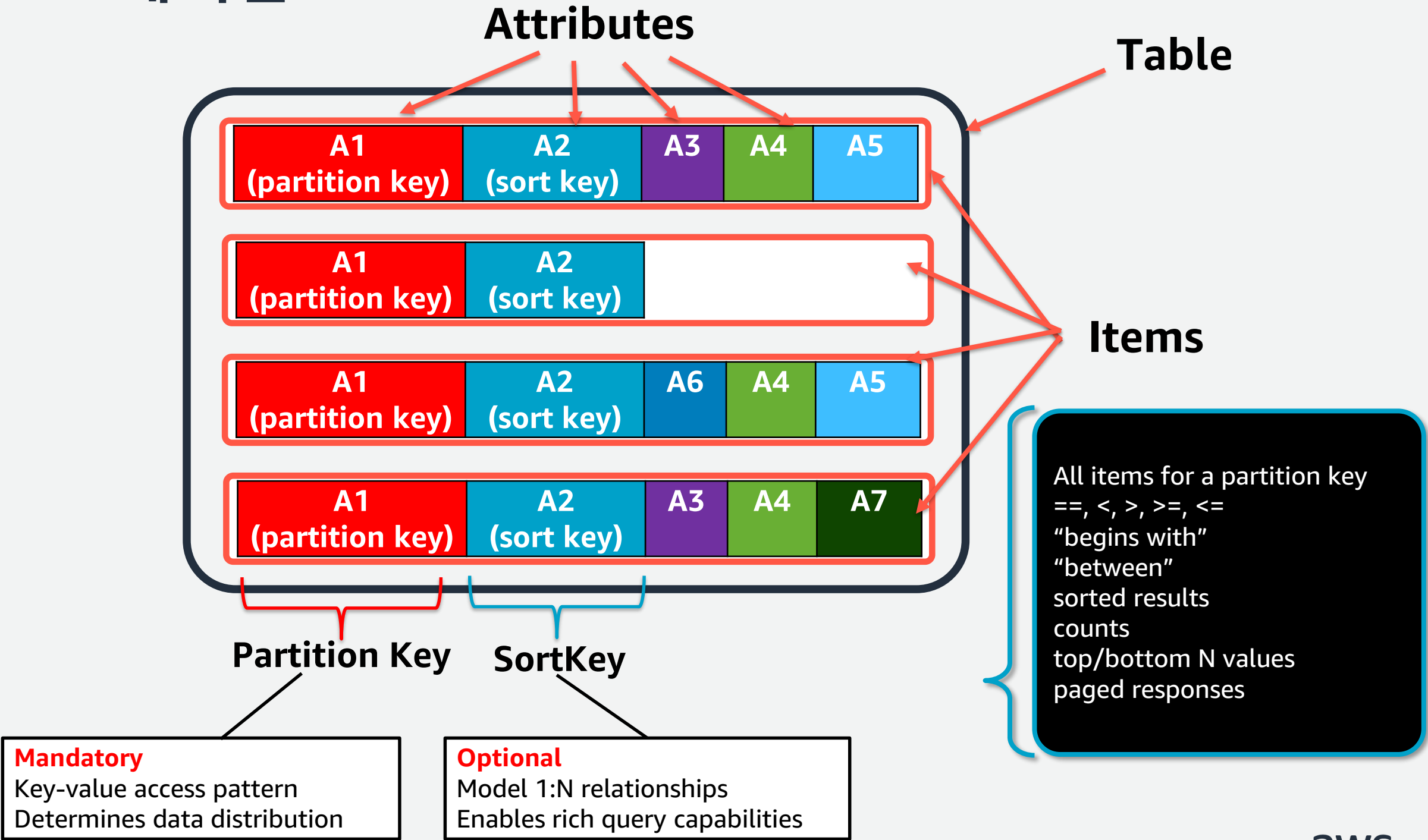
Data modeling: SQL vs. NoSQL



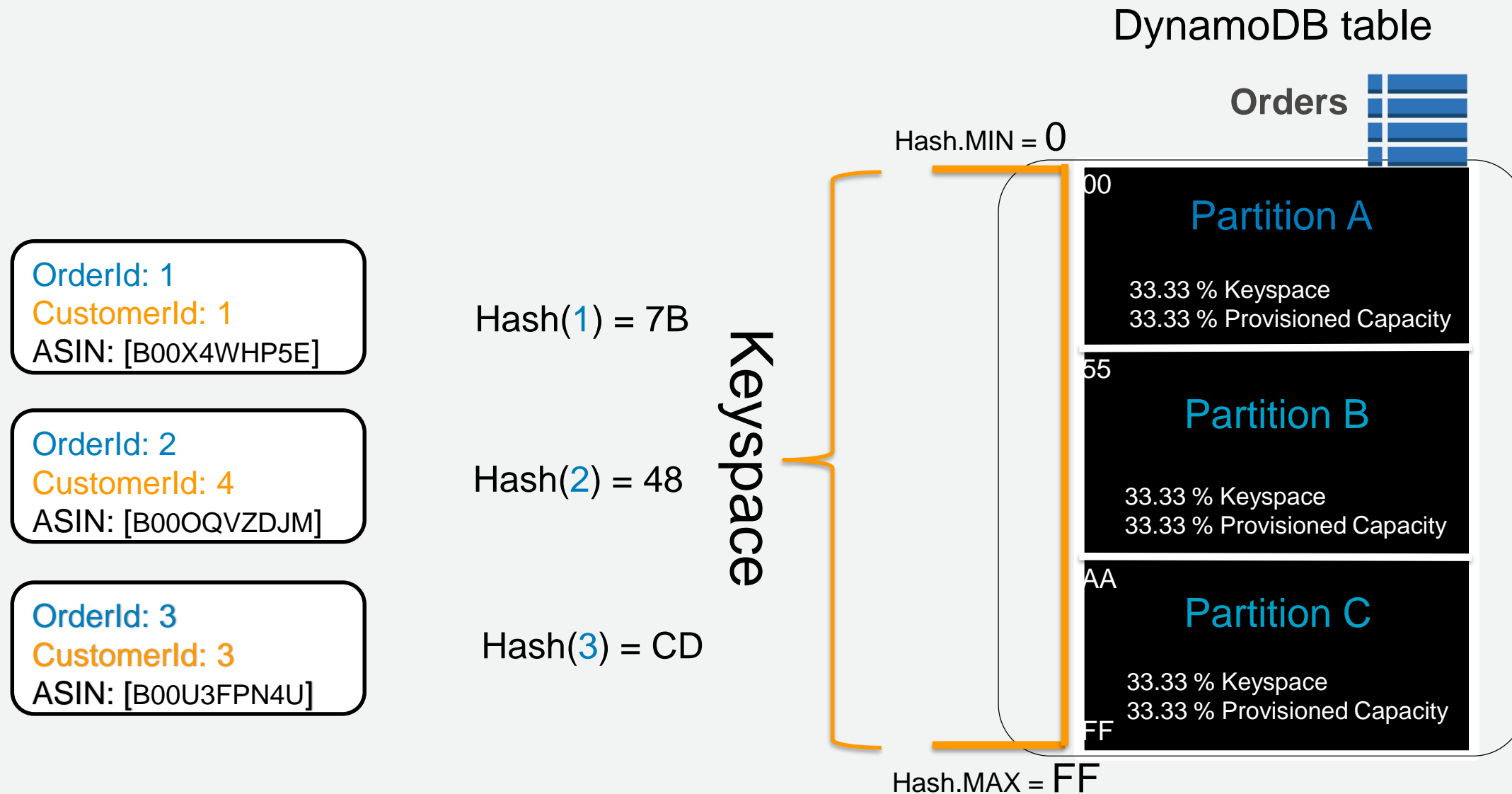
Data modeling: SQL vs. NoSQL



DynamoDB 테이블



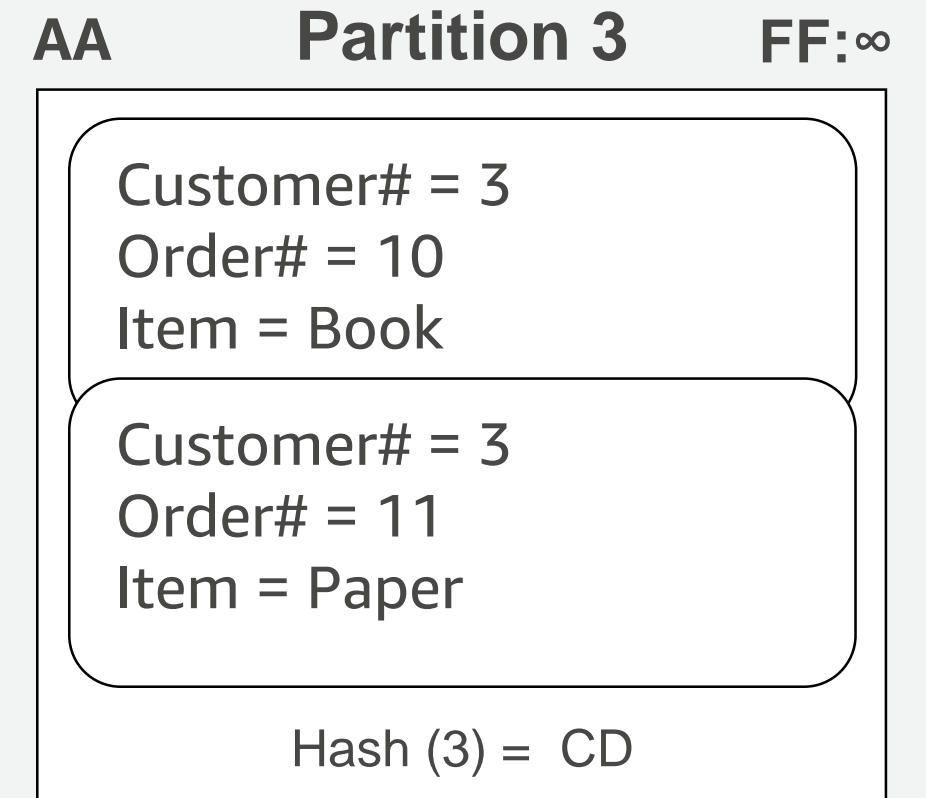
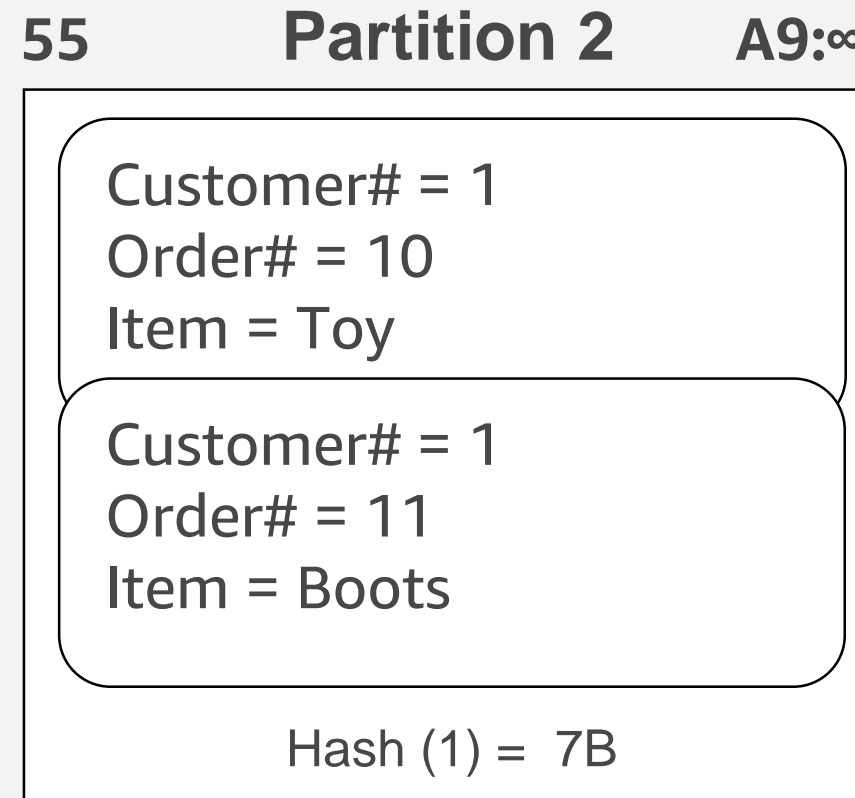
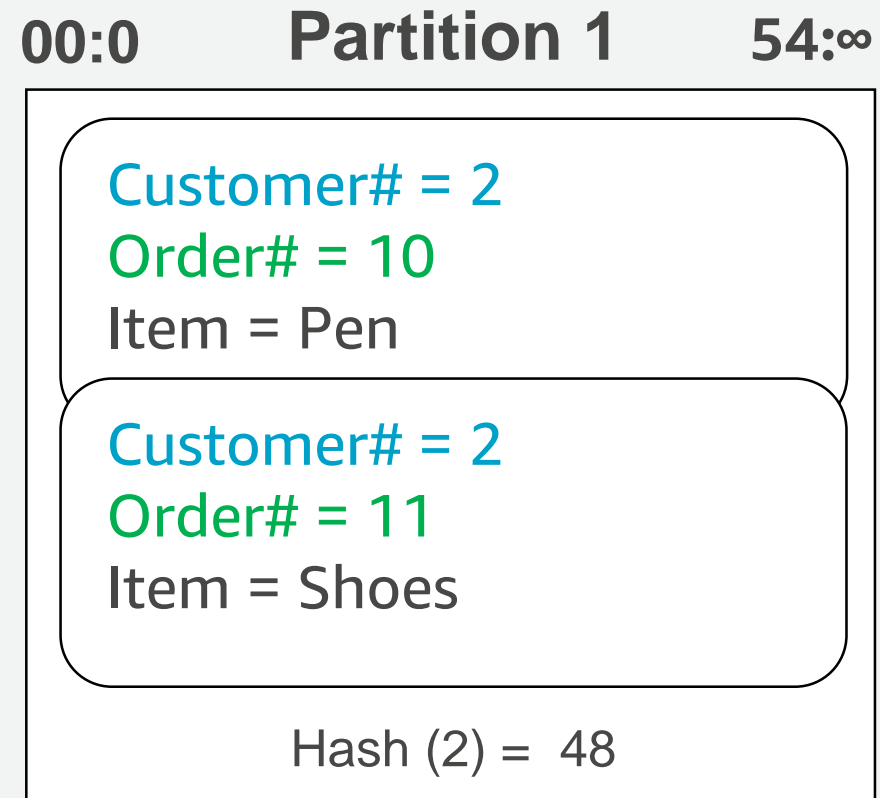
Partitioning



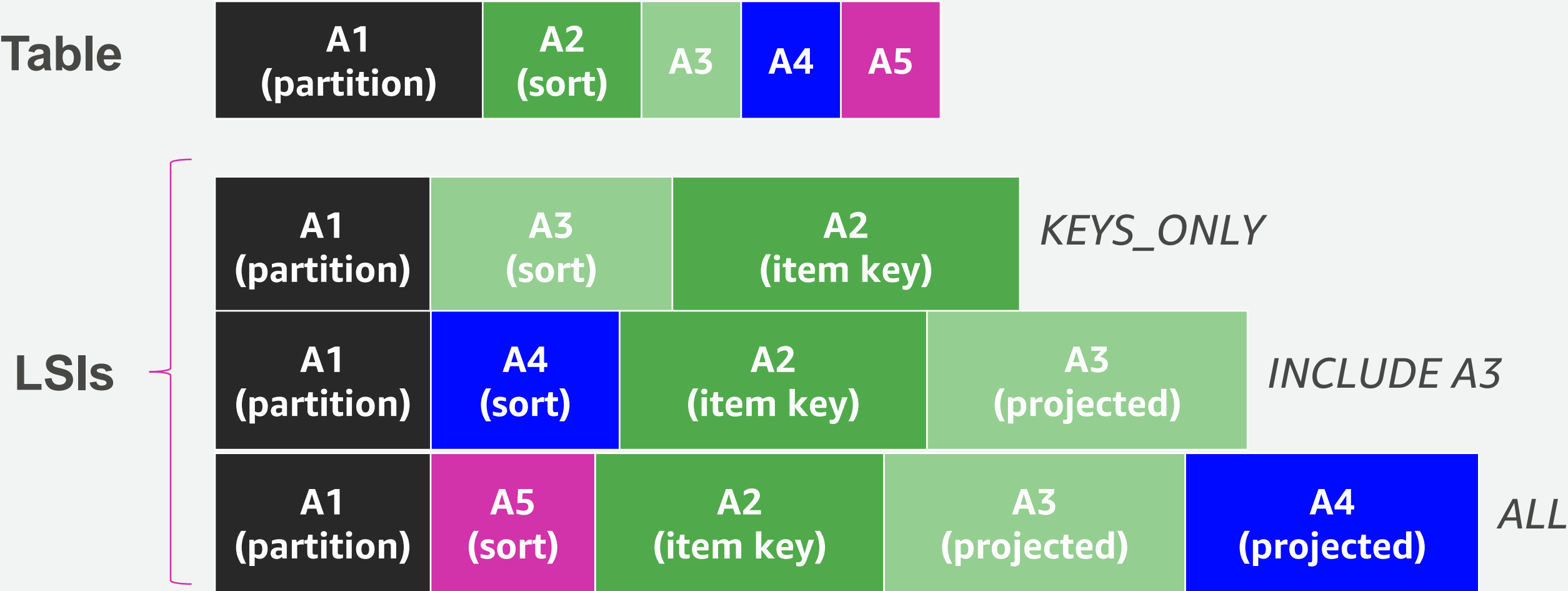
- 1 Partition = 10G, 1 Item = Max 400 KB, 1 Partition 에는 25,000 개의 Item이 들어감
- 1 Partition 의 최고 RCU는 3,000. 최고 WCU는 1,000 입니다.



Partition: Sort Key 예



Local secondary index(LSI)



LSI

- Alternate sort key attribute
- Index is local to a partition key

- 10 GB max per partition key, i.e. LSIs limit the # of range keys!
- Table당 5개



Global secondary index(GSI)

Table

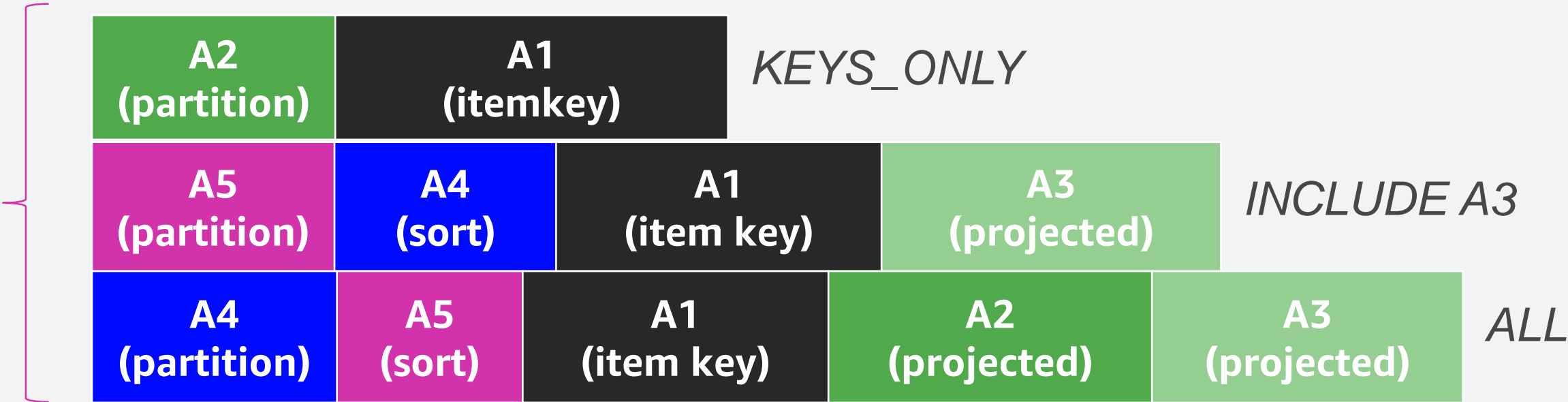


Online indexing

RCUs/WCUs provisioned separately for GSIs

GSI: 테이블당 20개 제약

GSIs



- Alternate partition (+sort) key
- Use composite sort keys for compound indexes
- Can be added or removed anytime



Data types

Type	DynamoDB Type
String	String
Integer, Float	Number
Timestamp	Number or String
Blob	Binary
Boolean	Bool
Null	Null
List	List
Set	Set of String, Number, or Binary
Map	Map



테이블 속성

읽기/쓰기 용량 모드

On-Demand Mode

- 애플리케이션에서 수행할 것으로 예상되는 읽기 및 쓰기 처리량을 지정할 필요가 없음
- 테이블에 대해 수행하는 읽기 및 쓰기에 대해 요금이 부과
- 워크노드에 따라 읽기/쓰기 용량이 변경됨
- 사용예
 - 알 수 없는 워크로드를 포함하는 테이블을 새로 만들 경우
 - 애플리케이션 트래픽이 예측 불가능한 경우
 - 사용한 만큼에 대해서만 지불하는 요금제를 사용하려는 경우

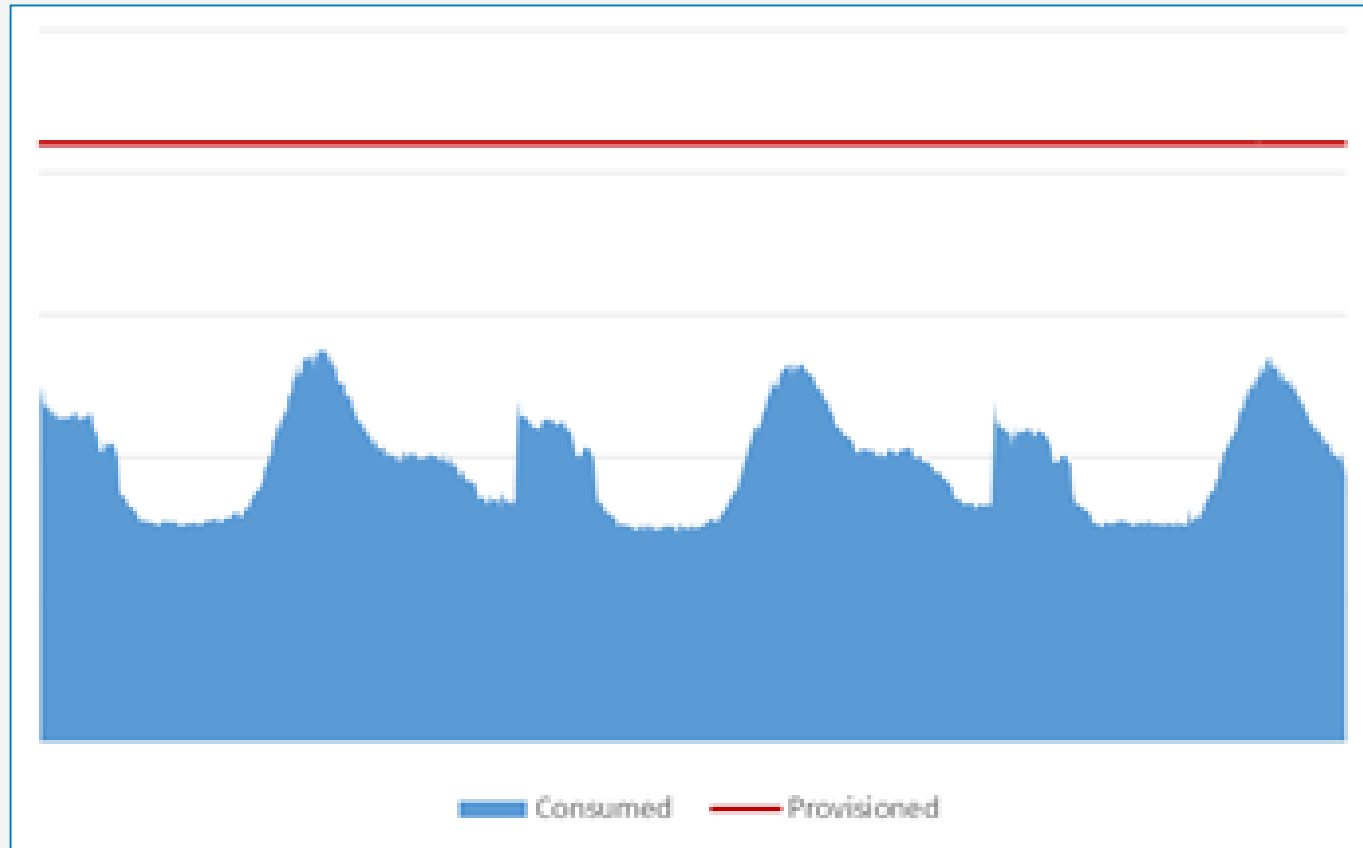
Provisioned Mode

- 테이블의 읽기 용량 단위(RCU) 및 쓰기 용량 단위(WCU)의 처리량을 지정
- DynamoDB는 이 정보를 사용하여 처리량 요구 사항에 맞는 충분한 시스템 리소스 확보
- DynamoDB Auto Scaling에서 테이블의 처리 용량을 관리하도록 허용할 수도 있음



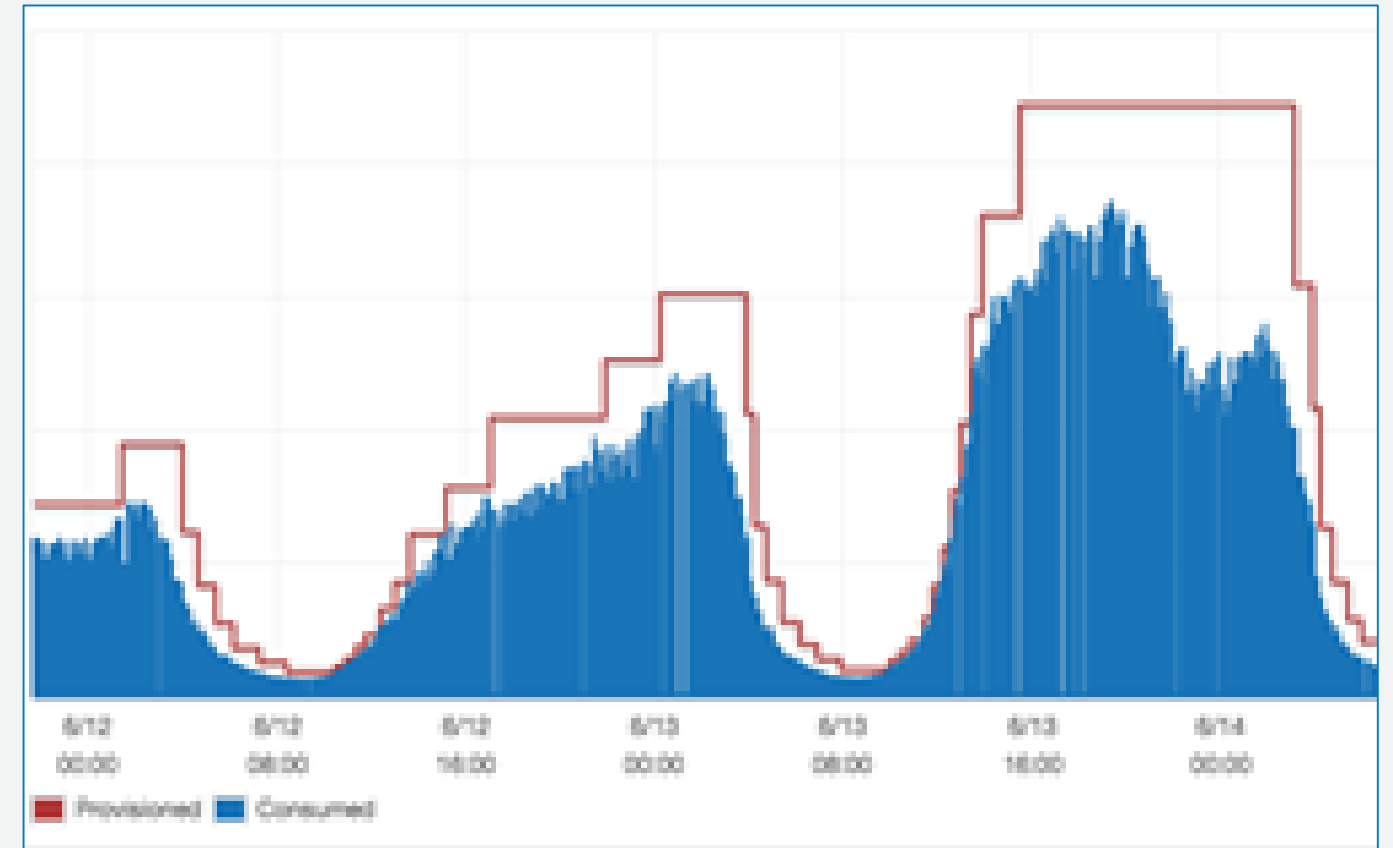
DynamoDB Auto Scaling

Throughput automatically adapts to your actual traffic



Without Auto Scaling

- 목표 사용률 — 테이블의 목표 사용률을 입력합니다.
- 최소 프로비저닝 용량 — Auto Scaling 범위의 하한을 입력합니다.
- 최대 프로비저닝 용량 — Auto Scaling 범위의 상한을 입력합니다.
- 글로벌 보조 인덱스에 동일한 설정 적용 — 이 옵션을 기본 설정(활성화)으로 유지합니다.

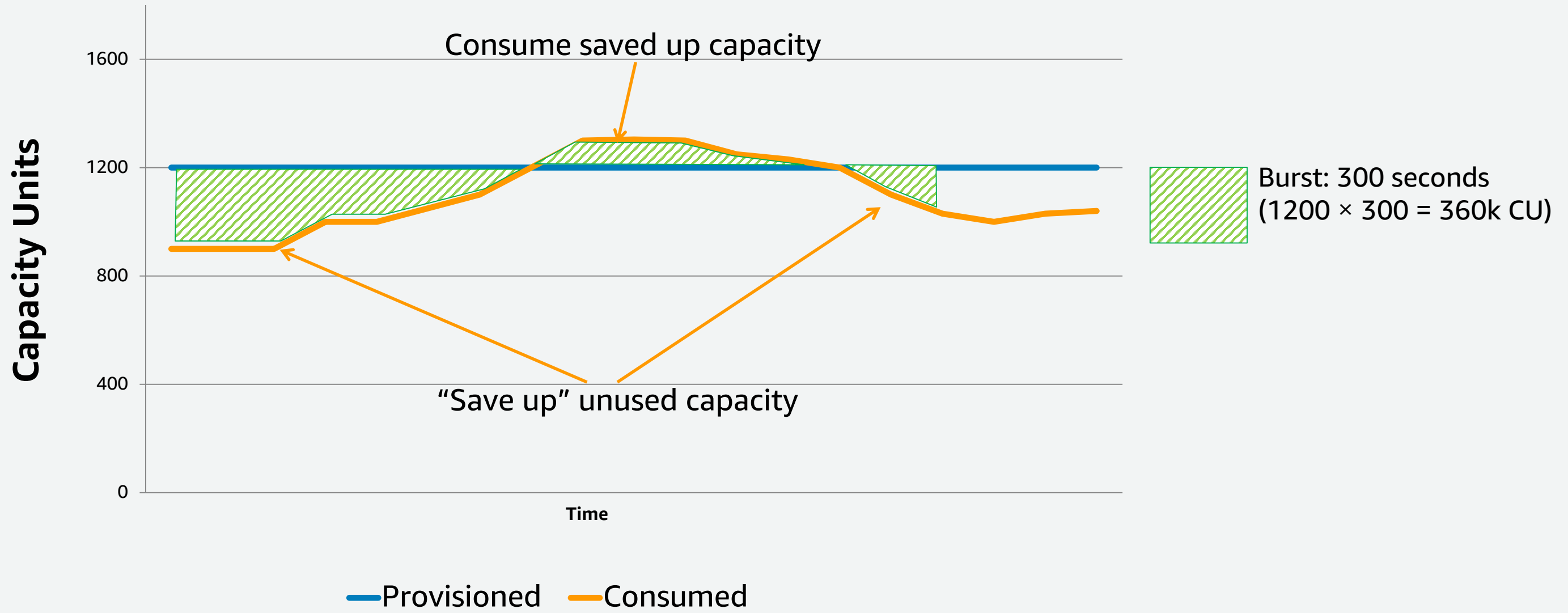


With Auto Scaling



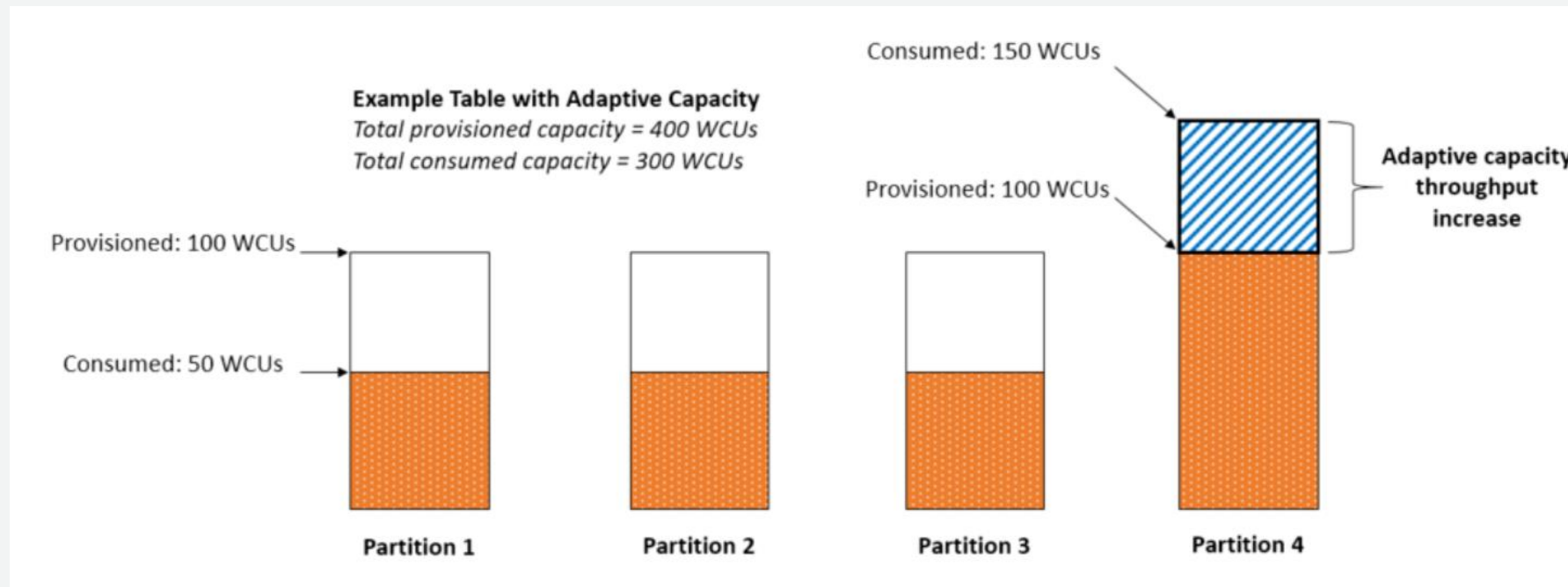
Burst capacity is built in

DynamoDB “saves” 300 seconds of unused capacity per partition



효과적으로 파티션 키를 설계

- 효과적인 버스트 용량 사용
 - 5분(300초)에 해당하는 미사용 읽기 및 쓰기 용량을 비축
- 높은 트래픽 파티션에 대한 처리량 향상
 - 특정 파티션 키에 대한 트래픽이 3,000RCU, 1,000WCU를 초과하지 않도록 유지
 - 핫 파티션의 Item 분산

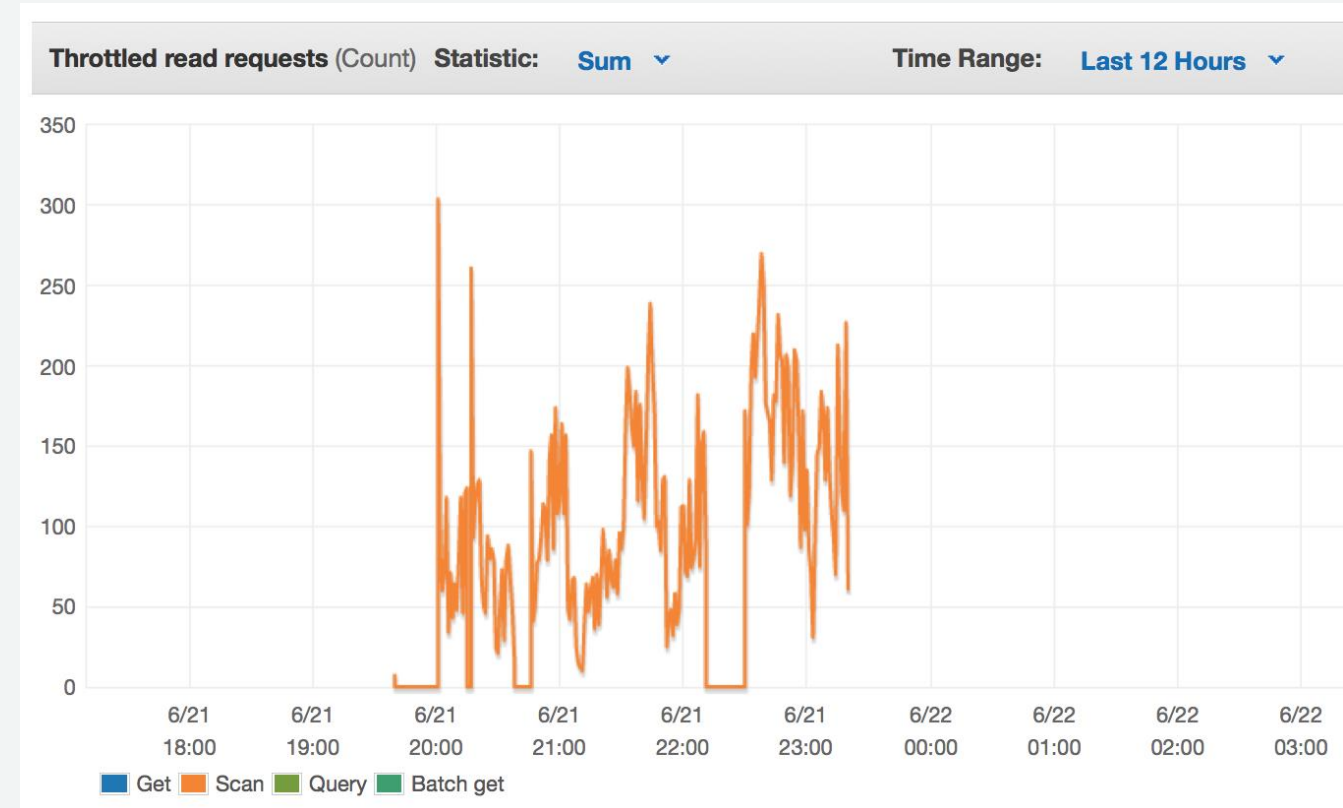


https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/best-practices.html



Throttling

- Occurs if **sustained** throughput goes **beyond provisioned throughput** per partition
- Possible causes
 - Non-uniform workloads
 - Hot keys/hot partitions
 - Very large items
 - Mixing hot data with cold data
 - Remedy: Use TTL or a table per time period
 - Disable retries, write your own retry code, and log all throttled or returned keys



Data modeling: Hierarchical data structures as items

- Use composite sort key to define a hierarchy
- Highly selective result sets with sort queries
- Index anything, scales to any size

	Primary Key		Attributes						
	ProductID	type							
Items	1	bookID	title	author	genre	publisher	datePublished	ISBN	
			Some Book	John Smith	Science Fiction	Ballantine	Oct-70	0-345-02046-4	
	2	albumID	title	artist	genre	label	studio	released	producer
			Some Album	Some Band	Progressive Rock	Harvest	Abbey Road	3/1/73	Somebody
	2	albumID:trackID	title	length	music	vocals			
			Track 1	1:30	Mason	Instrumental			
	2	albumID:trackID	title	length	music	vocals			
			Track 2	2:43	Mason	Mason			
	2	albumID:trackID	title	length	music	vocals			
			Track 3	3:30	Smith	Johnson			
	3	movieID	title	genre	writer	producer			
			Some Movie	Scifi Comedy	Joe Smith	20th Century Fox			
	3	movieID:actorID	name	character	image				
			Some Actor	Joe	img2.jpg				
	3	movieID:actorID	name	character	image				
			Some Actress	Rita	img3.jpg				
	3	movieID:actorID	name	character	image				
			Some Actor	Frito	img1.jpg				



... or as documents (JSON)

- JSON data types (M, L, BOOL, NULL)
- Document SDKs available
- 400 KB maximum item size (limits hierarchical data structure)

	Primary Key	Attributes							
	ProductID								
Items	1	id	title	author	genre	publisher	datePublished	ISBN	
		bookID	Some Book	Some Guy	Science Fiction	Ballantine	Oct-70	0-345-02046-4	
	2	id	title	artist	genre	Attributes			
		albumID	Some Album	Some Band	Progressive Rock	{ label:"Harvest", studio: "Abbey Road", published: "3/1/73", producer: "Pink Floyd", tracks: [{title: "Speak to Me", length: "1:30", music: "Mason", vocals: "Instrumental"},{title: "Breathe", length: "2:43", music: "Waters, Gilmour, Wright", vocals: "Gilmour"},{title: "On the Run", length: "3:30", music: "Gilmour, Waters", vocals: "Instrumental"}]}			
	3	id	title	genre	writer	Attributes			
		movieID	Some Movie	Scifi Comedy	Joe Smith	{ producer: "20th Century Fox", actors: [{ name: "Luke Wilson", dob: "9/21/71", character: "Joe Bowers", image: "img2.jpg"},{ name: "Maya Rudolph", dob: "7/27/72", character: "Rita", image: "img1.jpg"},{ name: "Dax Shepard", dob: "1/2/75", character: "Frito Pendejo", image: "img3.jpg"}]}			



DynamoDB 활용 - Demo

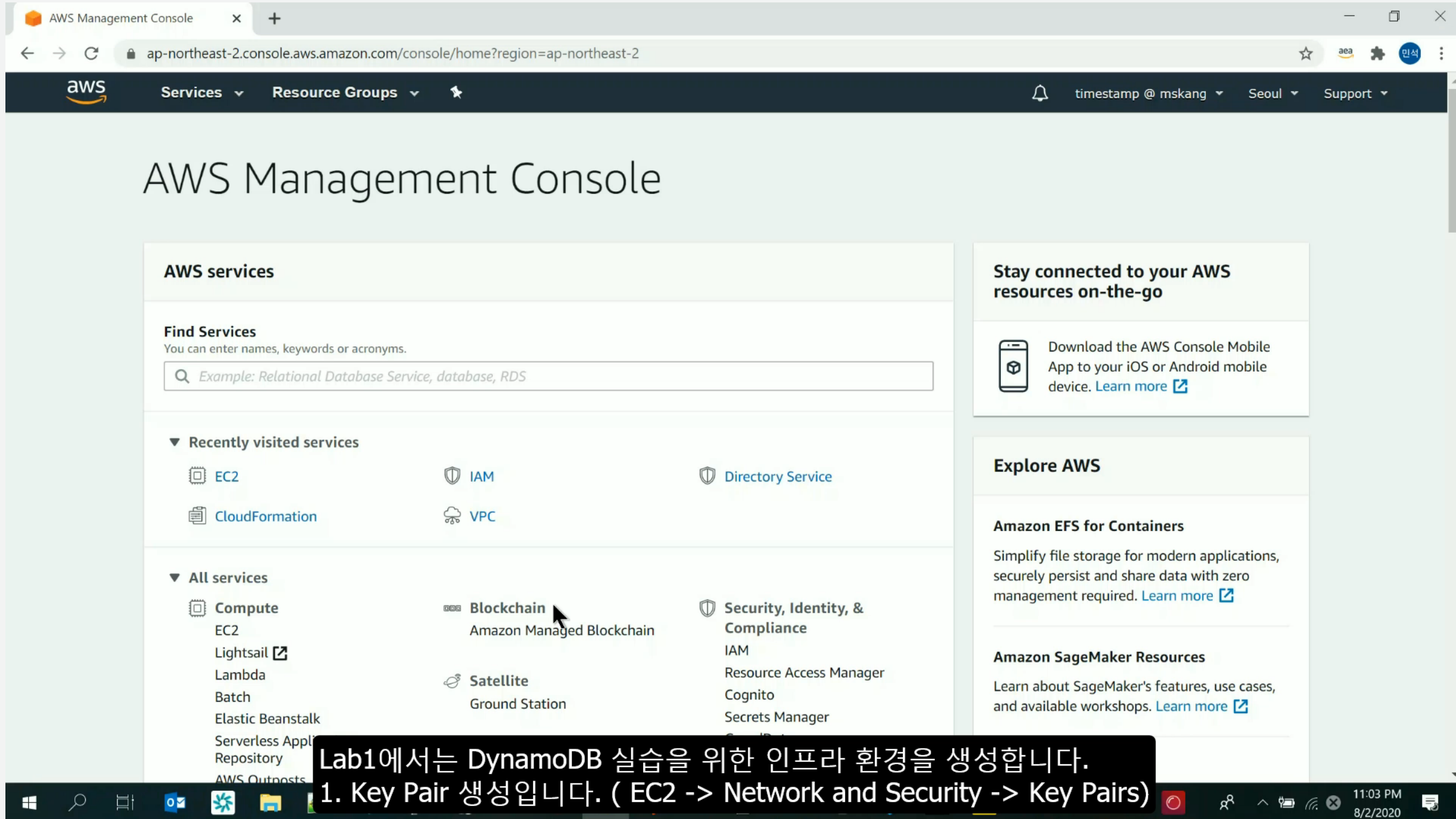


Table of Contents

Lab 개요.....	3
준비 조건.....	4
Lab 1. 실습 환경 구성	5
1-1. Key Pair 생성(기존에 Key Pair 가 있는 사용자는 SKIP).....	5
1-2. CloudFormation 으로 Lab 환경 생성	6
1-3. SSH 로 Bastion 서버 접속(Windows Only).....	9
1-4. Bastion 환경 설정	11
Lab 2. DynamoDB 에서 테이블 Create/Insert/Update 및 Select	16
2-1. 테이블 생성	16
2-2. 데이터 입력	17
2-3. Conditional Query	20
2-4 list_append 를 활용하는 쿼리.....	22
2-5 Update	23
2-6 insert - attribute_not_exist.....	24
2-7 GSI 만들기.....	25
2-8 filter expression select on GSI.....	28
Lab 3. 성능 테스트	31
3-1 테스트 데이터 생성(JSON).....	31
3-2 WCU/RCU 조정 후 성능 테스트	35



Lab1 – 실습환경 구성

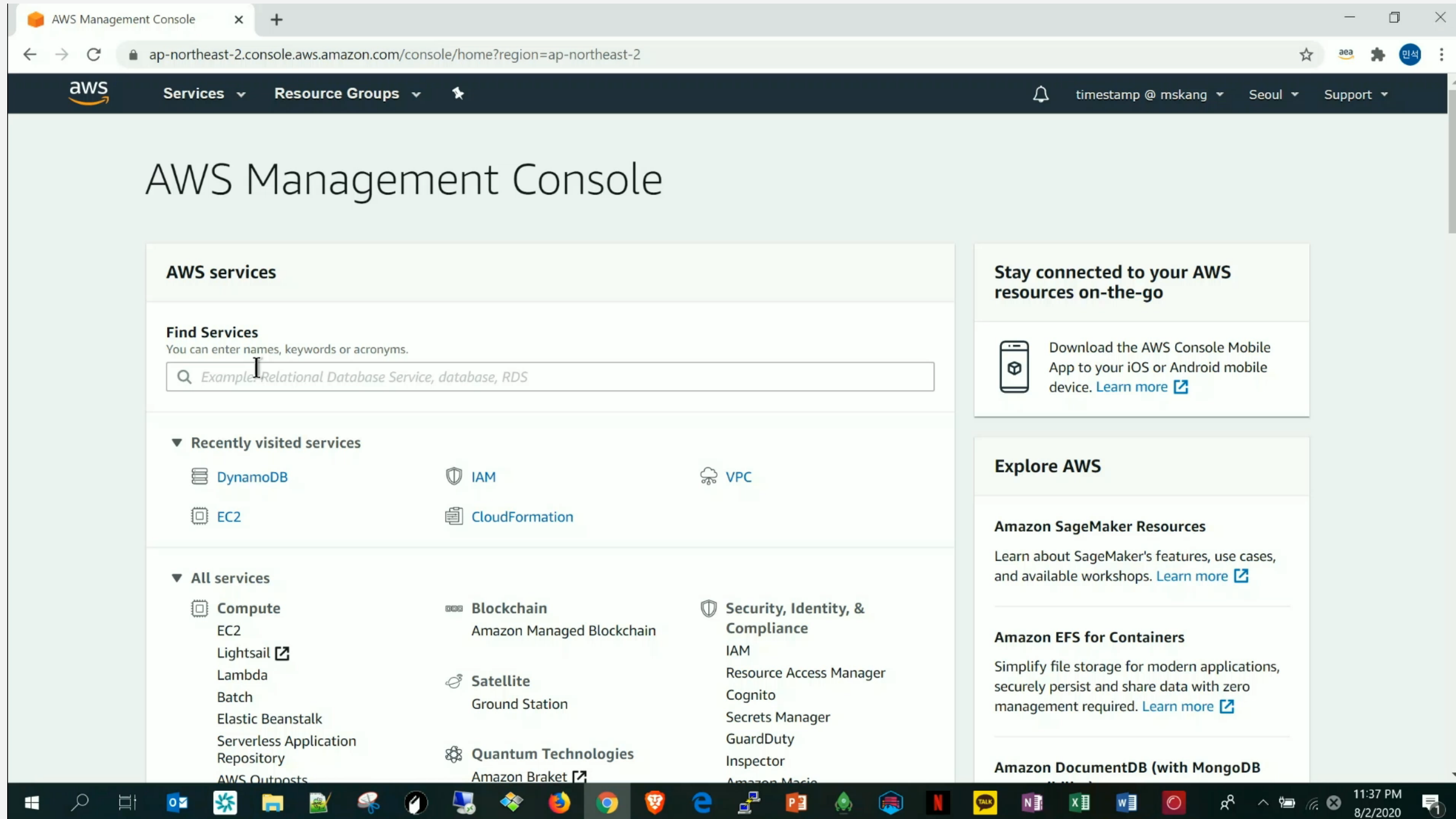


The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'timestamp @ mskang' with location 'Seoul' and 'Support' link. The main heading is 'AWS Management Console'. Below it, there's a 'Find Services' search bar with the placeholder text 'Example: Relational Database Service, database, RDS'. The 'Recently visited services' section lists EC2, CloudFormation, IAM, VPC, and Directory Service. The 'All services' section is expanded, showing categories like Compute (EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless App Repository, AWS Outposts), Blockchain (Amazon Managed Blockchain), Satellite (Ground Station), Security, Identity, & Compliance (IAM, Resource Access Manager, Cognito, Secrets Manager), and others. A mouse cursor is pointing at 'Blockchain'. A black text box is overlaid on the bottom of the console.

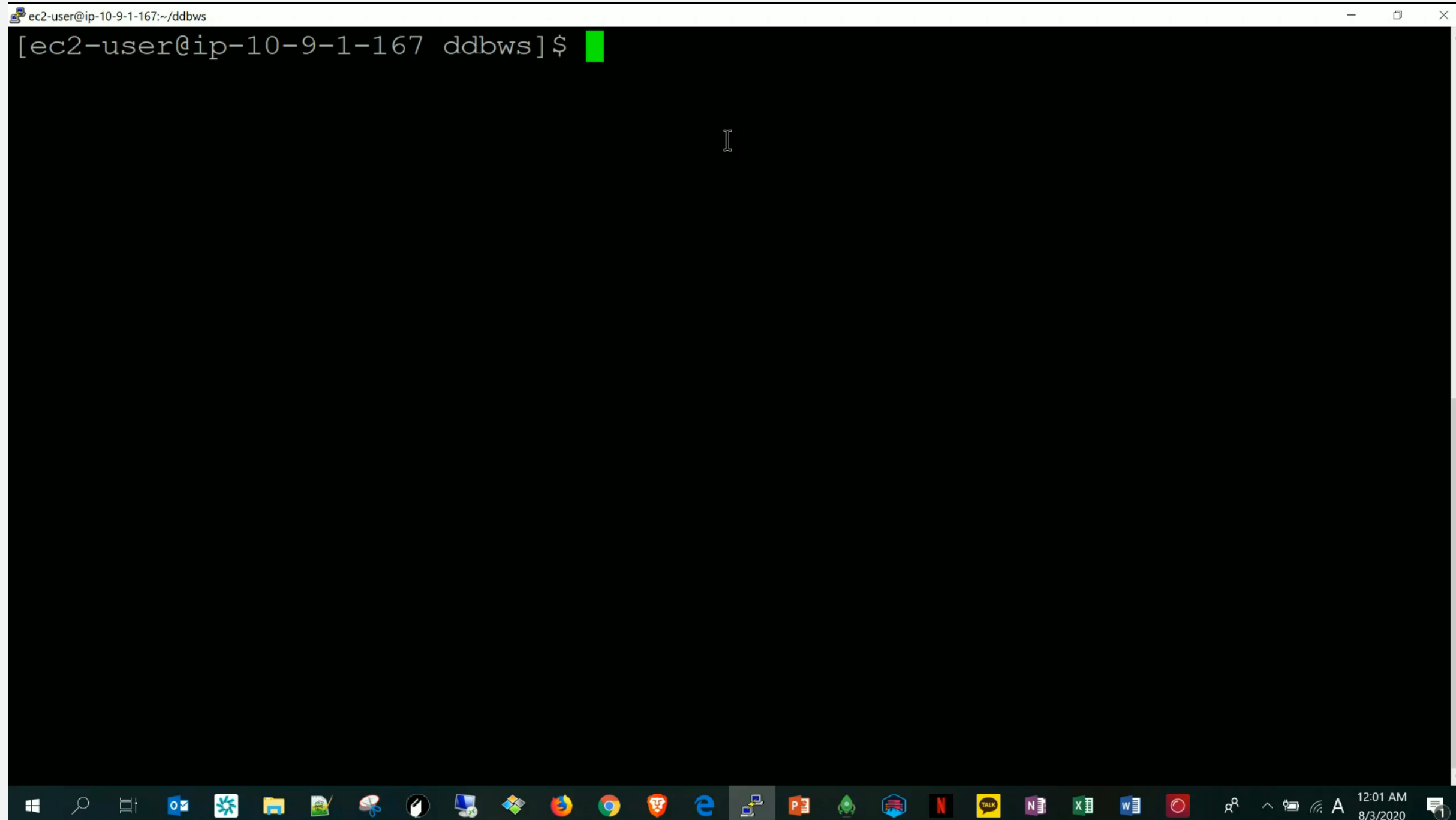
Lab1에서는 DynamoDB 실습을 위한 인프라 환경을 생성합니다.
1. Key Pair 생성입니다. (EC2 -> Network and Security -> Key Pairs)



Lab2 – DynamoDB 테이블 생성 및 활용



Lab3 – 성능 테스트(WCU/RCU – On-Demand)









```
ec2-user@ip-10-9-1-167:~/ddbws
[ec2-user@ip-10-9-1-167 ddbws]$
```



AWS Database Modernization Day에 참석해주셔서 대단히 감사합니다.

저희가 준비한 내용, 어떻게 보셨나요?
더 나은 세미나를 위하여 **설문을 꼭 작성해 주시기 바랍니다.**

-  aws-korea-marketing@amazon.com
-  twitter.com/AWSKorea
-  facebook.com/amazonwebservices.ko
-  youtube.com/user/AWSKorea
-  slideshare.net/awskorea
-  twitch.tv/aws



Thank you!

