



AWS DataBase

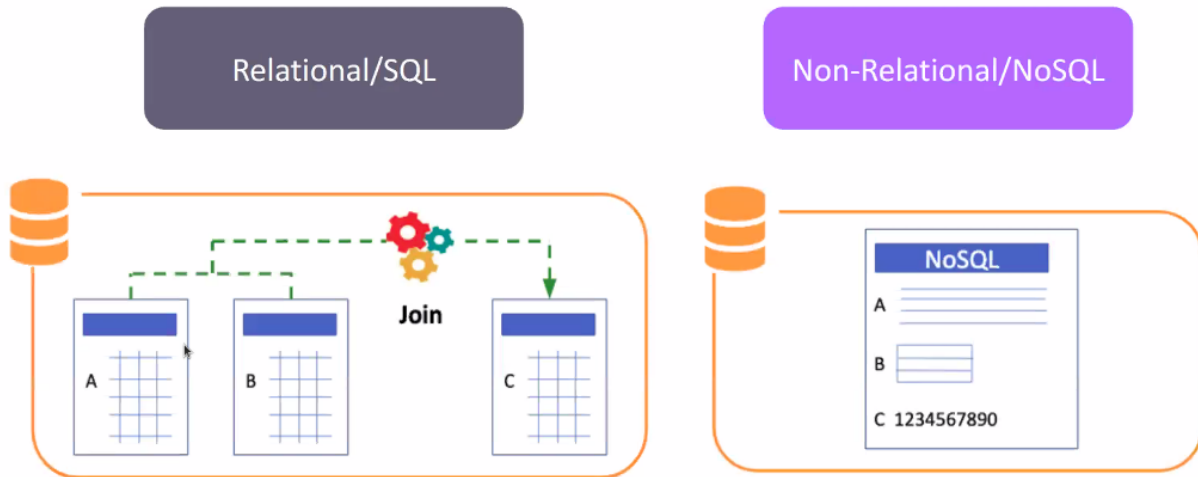
RDS'e bağlamak ve database üzerinde değişiklik yapabilmek için MySQL Workbench isimli bir tool kullanacağız. Aşağıdaki linkten indirdim.

<https://dev.mysql.com/downloads/workbench/>

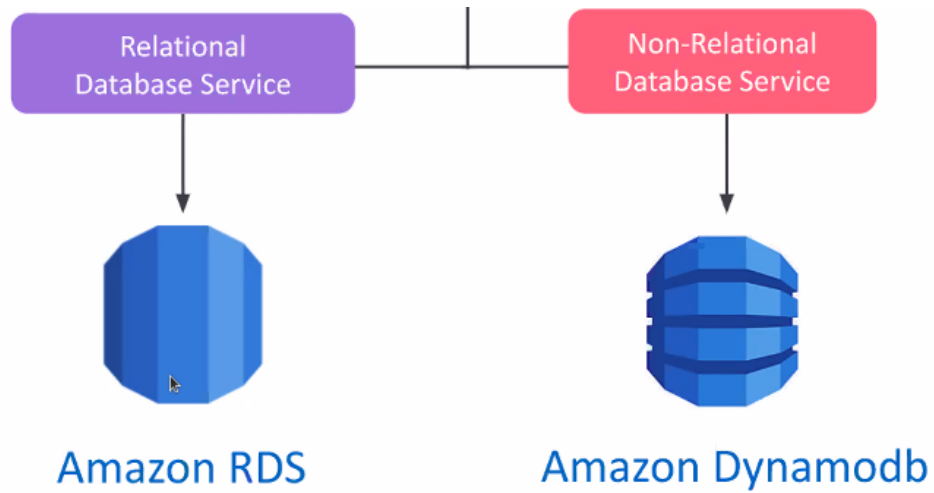
İkiye ayrılır. Relational DataBase (SQL), Non-Relational DataBase (No-SQL)

SQL daha çok veri tabloları ile çalışır.

NoSQL ise daha esnek bir yapı var. JSON formatında tutulur. Key-value den oluşan veriler var



SQL, tutulan verilerden yeni bir veri tabanı elde etmeye de yarar. A ile B'den C yi çıkarma gibi. Bu işleme Join function denir.



yukarıdakiler amazonun databaseleri. Relation tarafında toplam 6 alternatif var. Ancak non-relation da sadece Amazonun makinası var.

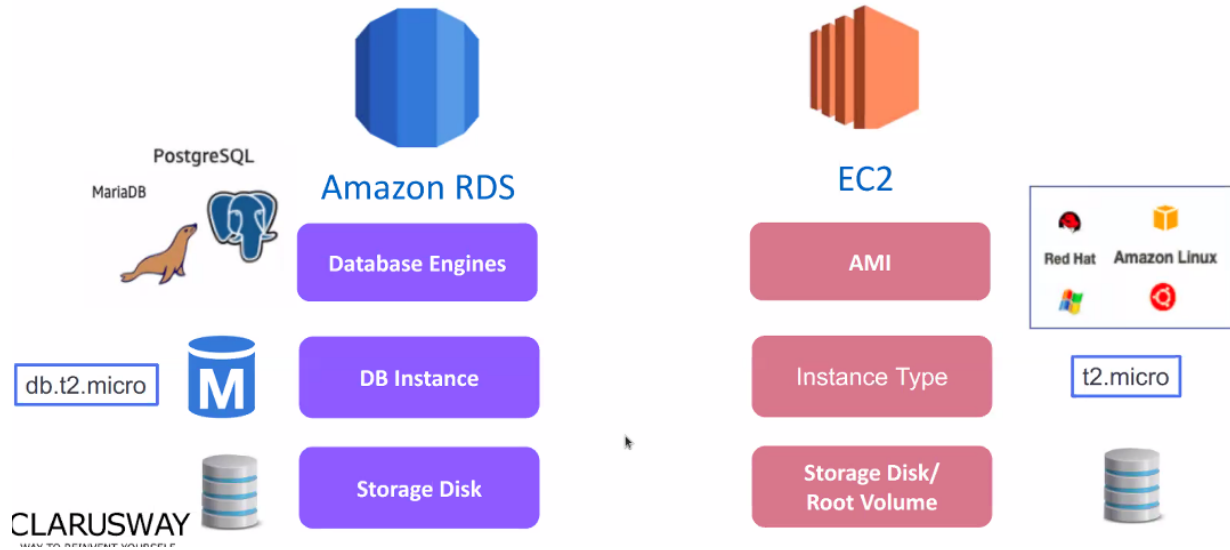
Dynamodb serverless bir yapıdır. RDS başlı başına bir makinadır.

SQL	NoSQL
Relational	Non-Relational
Table-based	Document-based, key-value pairs, graph databases or wide-column stores
Predefined Schema	Dynamic Schema
Uses SQL	As the name suggest, it doesn't use SQL
Used for complex queries	Used for simple queries
Available for Join function	Not available for Join function

Amazon RDS

AWS-manage bir servistir. İşletilmesi, bakımı AWS'ye aittir. Veritabanının tasarım ve veri değiştirme işlemleri kullanıcıya aittir

Basic Components



AWS'nin desteklediği Database Engine'lar aşağıda. Eğer bunlardan başka kullanmak istersek makinaya kurmak gerekecek. Ancak diğerleri için AWS-manage kısmında zorluklar oluyor

- Aurora, Oracle, MariaDB, PostgreSQL, Microsoft SQL Server, MySQL

RDS Çeşitleri

▼ Pricing

On Demand Instance

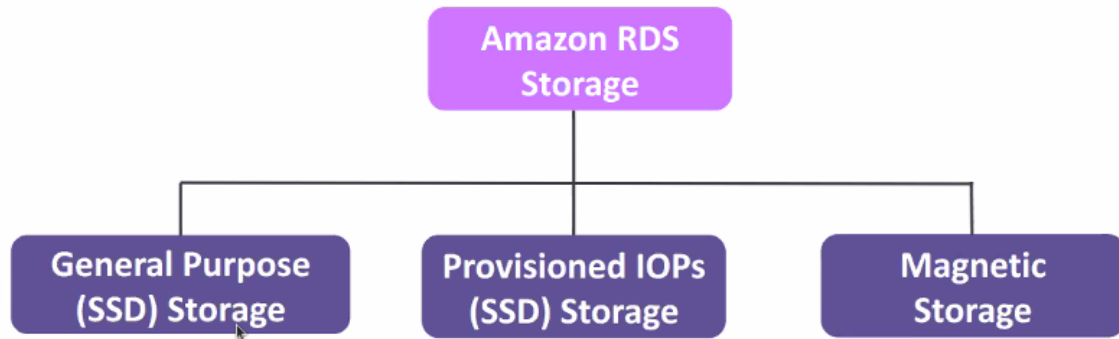
Reserved Instance

▼ Purpose

Memory Optimized (Yüksek performans, Yüksek maliyet)

General Purpose

Burstable



Cost-effective

Fast and consistent I/O

Not recommended

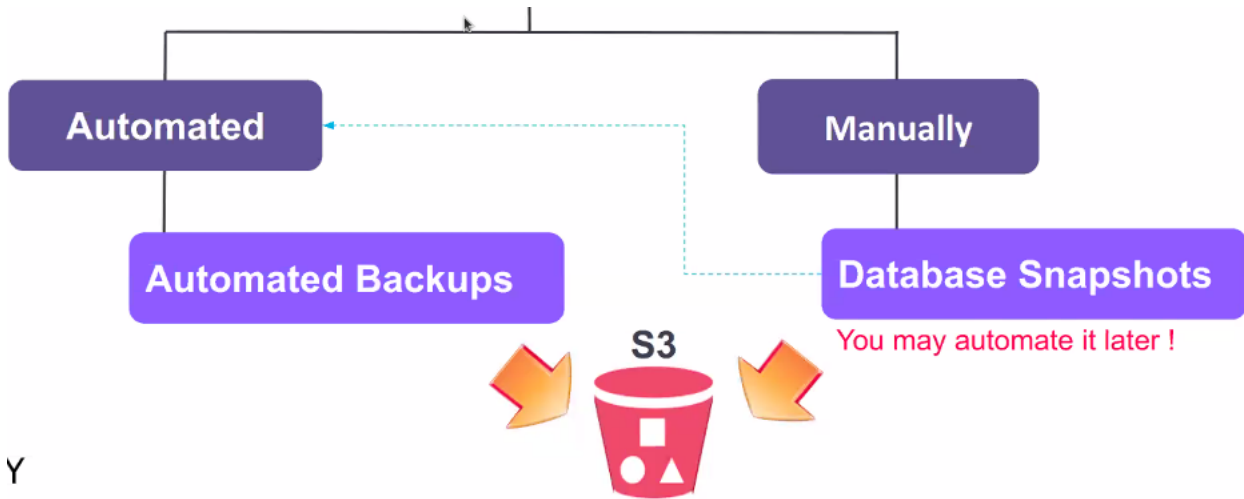
Depolama alanına göre sıralama

Apache Server'a 80 portunu açıp Browser'dan ulaşabiliyorduk. Ancak RDS'e SSH bağlantısı vs. yapılamıyor. Çünkü bunu AWS yönetiyor. Ulaşmak için MySQL WorkBench gibi toolarla ulaşabiliyoruz.



Back-ups

Automated olarak AWS 35 güne kadar 5'er dk.lık log kayıtları alınır. 36. gün gelince 1. günü siler. İstersen manual olarak Snapshot alabilirsin. Database'i silersen otomatik alınanlar silinir ancak manuel snapshotlar kalır.



Y

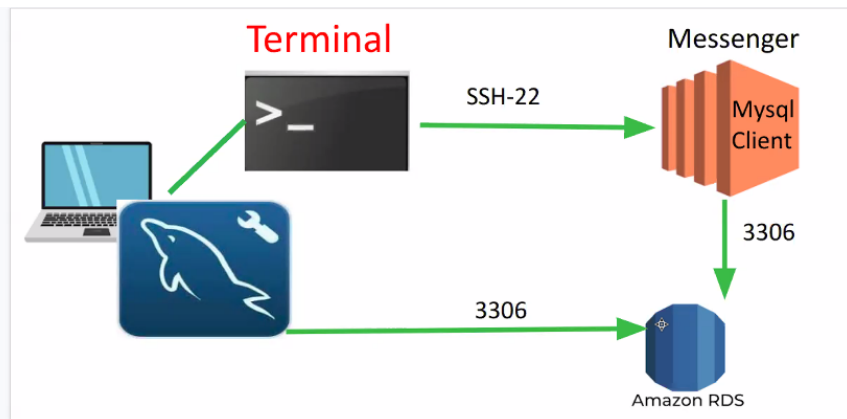
Multi-AZ DB

Multi-AZ'de çalışmak için AWS, DB'yi kopyalatıp diğer AZ de Standby DB Instance oluşturuyor. Ancak bu sadece yedekleme amaçlı Load Balancer gibi çalışmaz. Primary DB arıza verirse Standby'ı active eder. Kayıp olmaz. Performans olarak herhangi bir artış sağlamaz.

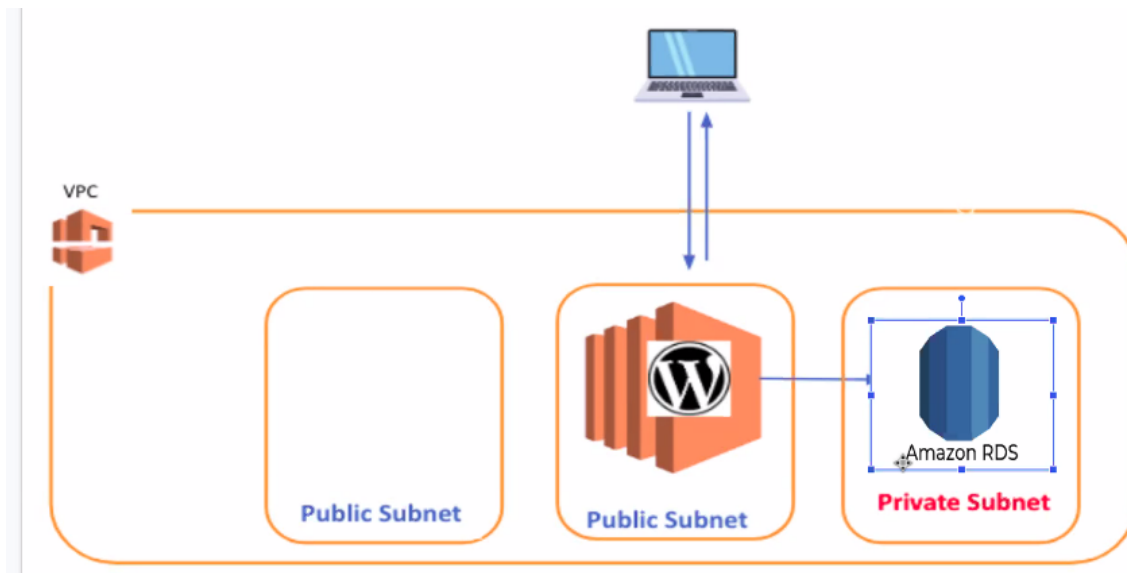
Read Replica

Burada Primary DB sadece Readable olarak kopyalanır. Load Balancer gibi çalışır. DB'nin yükünü alır. Ancak yazdırma olayı yok replicada. Yeni birşey yazılacaksa primarye yazılır replica ile senkronize edilir.

RDS'e ulaşım:



Ancak Best-practise:



Client RDS'le ya da EC2 ile konuşur/ulaşlık yapar ancak bünyesinde database tutamaz sadece komutları iletir.

MariaDB

```
MariaDB [mysql]> SELECT Host, User, Password FROM user;
+-----+-----+-----+
| Host          | User | Password |
+-----+-----+-----+
| localhost     | root |          |
| ip-172-31-82-94.ec2.internal | root |          |
| 127.0.0.1     | root |          |
| ::1          | root |          |
| localhost     |      |          |
| ip-172-31-82-94.ec2.internal |      |          |
+-----+-----+-----+
```

MariaDB'ye yukarıda karşısında root yazmayan adresten bağlandık. Daha sonra root kullanıcı olarak giriş yaparak bu girişleri siliyoruz ki kimse şifresiz bir şekilde giriş yapmasın.

EC2'dan root olarak girmek için **mysql -u root**
password ile girmek için **mysql -u root -p**

Yeni kullanıcı oluşturduktan sonra bu kullanıcının bir client vasıtası ile girmesini isteriz. Ancak server da aynı zamanda bir client'tır ve Database'e ulaşabiliriz.

MYSQL EC2'dan bağlanma:

Database'e bağlanmak için:

database containerin içine gelip ;

mysql -uroot -p bookstore_db yazıyoruz ⇒ bookstore_db database ismi

Şifre P123456p

```
environment:
  MYSQL_ROOT_PASSWORD: P123456p
  MYSQL_DATABASE: bookstore_db
  MYSQL_USER: clarusway
  MYSQL_PASSWORD: Clarusway_1
```

root yerine user olarak da girebiliriz;

mysql -uclarusway -p bookstore_db

Şifre Clarusway_1

SHOW DATABASES;

USE bookstore_db

SHOW TABLES;

SELECT * FROM books;

```

MySQL [(none)]> SHOW DATABASES;
+-----+
MySQL [bookstore_db]> SELECT * FROM books;
+-----+-----+-----+-----+
| book_id | title                                | author                                | is_sold |
+-----+-----+-----+-----+
| 1       | Where the Crawdads Sing              | Delia Owens                         | 1       |
| 2       | The Vanishing Half: A Novel          | Brit Bennett                        | 0       |
| 3       | A Formula de Deus                    | Jose Rodrigues dos Santos           | 1       |
| 4       | The Alchemist                        | Paulo Coelho                        | 0       |
| 5       | One Hundred Years of Solitude        | Gabriel Garcia Marquez              | 0       |
| 6       | To Kill a Mockingbird                 | Harper Lee                           | 0       |
+-----+-----+-----+-----+
6 rows in set (0.000 sec)

```

RDS Snapshot



Dump İşlemleri

Eğer database'imiz EC'2 içindeyse dumping yaparak bilgileri Database'e kopyalarız.

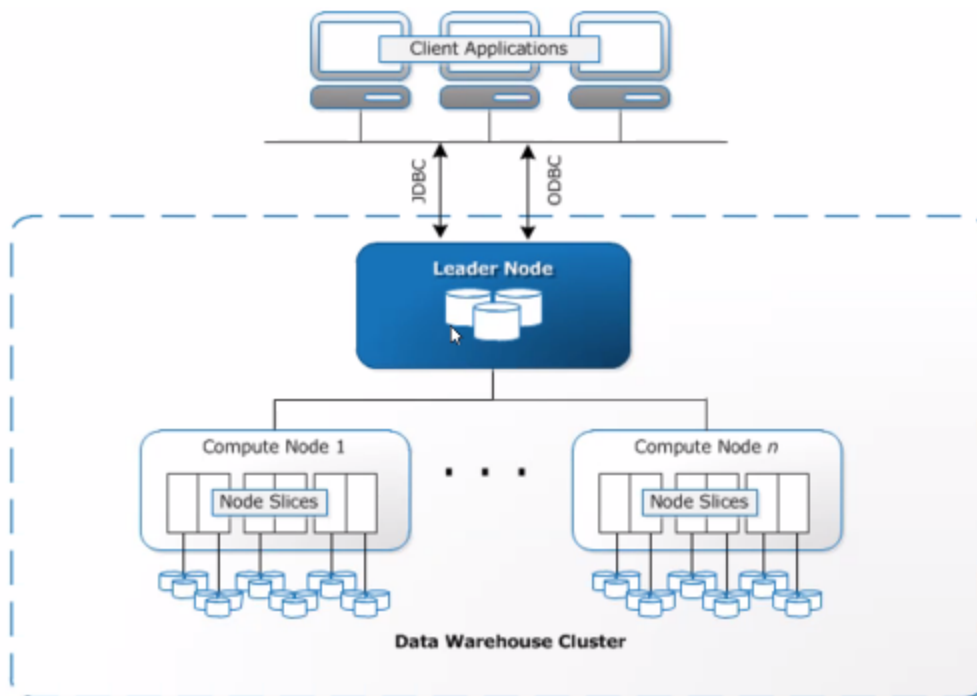
NoSQL

Amazon Redshift

Redshift bir data warehouse dur.Petabayt seviseyinde çok çok büyük bir veri havuzu. Asıl amacı veri analizi yapmak. Bu da AWS fully-managed bir servistir.

Database	Data Warehouse
	
OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Used for data storing and managing	Used for data analyzing

Database(RDS) verilerin sorgulanması, değiştirilmesi saklanması için tasarlanmış bir sistem. Warehouse verilerin analizi için



AWS Redshift Data Warehouse architecture

Serverless çalışan bir sistem.



AWS ElastiCache

Hızlandırma ve database in yükünü hafifletmek için kullanılan bir sistem. Bilgisayardaki RAM gibi bir çalışma mantığı var. İlk sorguda veriyi databaseden çeker. Daha sonraki sorgularda boşuna database'e gitmez. Daha önce çektiği veriyi kullanır.

CacheHit: DataBase'e gitmeden önce Elasticache gider. Eğer sorgulanan bilgi burada varsa cachehit verir. ve boşuna DB'ye gitmez

CacheMiss: Eğer bilgi Elasticach'te yoksa cachemiss verir ve DB'den çeker.

Redis ve Memcached yaygın kullanılanlardan ikisi

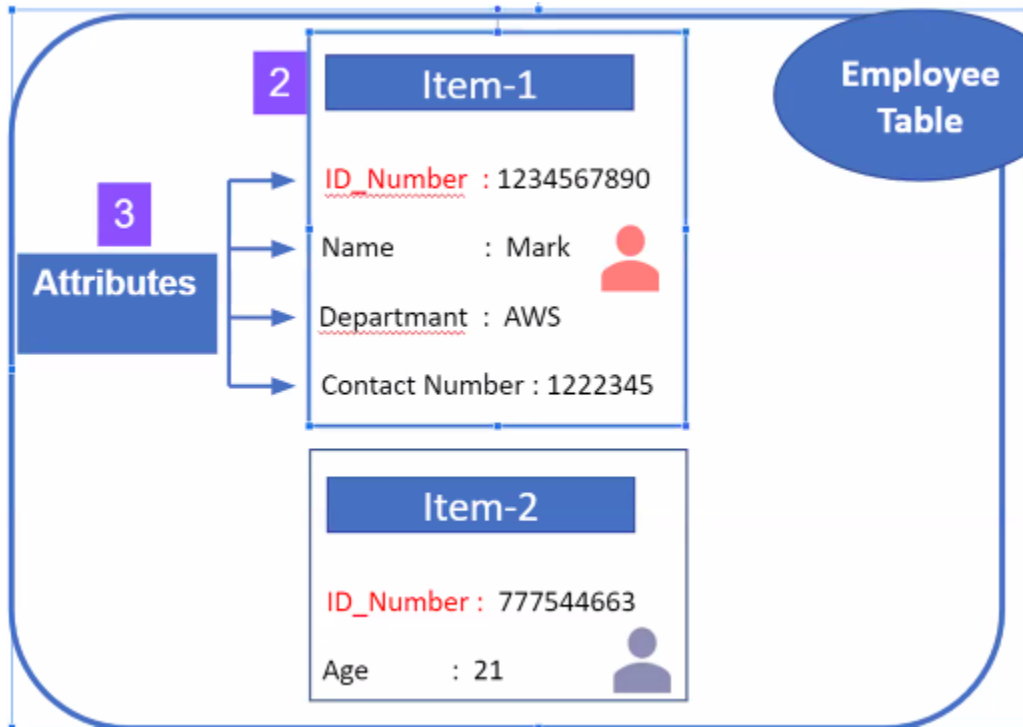
Redis 			Memcached 
Sub-millisecond latency	+	+	Sub-millisecond latency
User friendly syntax	+	+	User friendly syntax
It supports many different programming languages C, C++, java, python, etc.	+	+	It supports many different programming languages C, C++, java, python, etc.
Redis supports strings, lists, sets, sorted sets, hashes, bit arrays, and hyperloglogs	+	-	Memcached supports only strings
It doesn't support multithreaded architecture	-	+	It supports multithreaded architecture. It means that it has multiple processing cores. This allows you to handle multiple operation
It supports Snapshot	+	-	It doesn't support Snapshot
It supports Replica	+	-	It doesn't support Replica

Amazon Dynamo DB

NoSQL bir DB'dir. Amazon, RDS tarafında 6 tane DB servisini destekler ancak NoSQL tarafında Dynamo dışında bir alternatif sunmuyor.

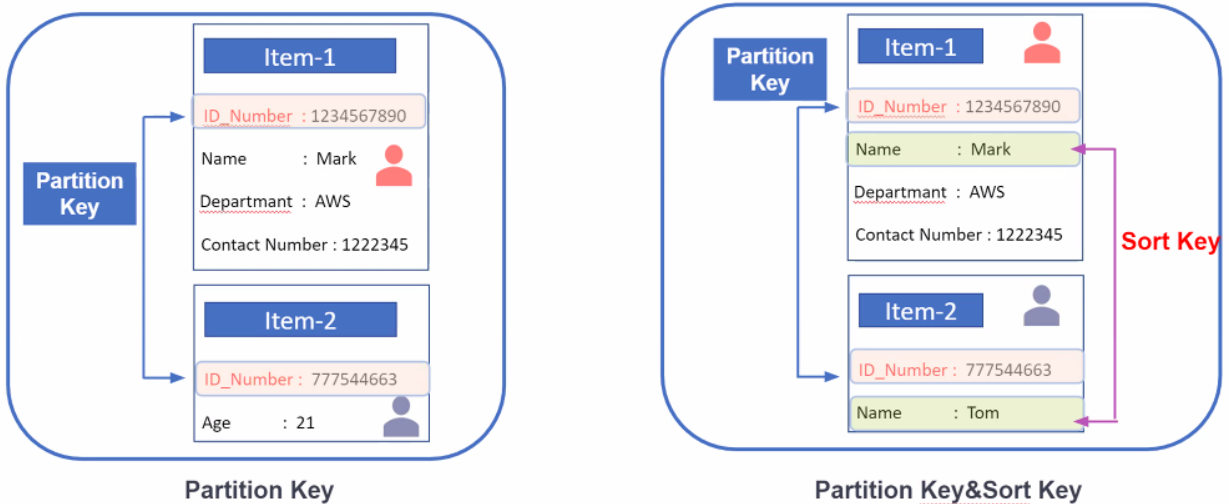
Schema'lar yerine flexible tablolar kullanılıyor. Join function kullanılmaz. Fully-managed by AWS'dir. Hatta RDS'ten farklı olarak kullanılacak instance tipine kadar AWS belirler.

Tablonun içinde item'lar var. Ve bu itemlerin içinde bilgiler yer alıyor



Primary Key sadece Partition Key'den de oluşur. Partition Key ve Sort Key'den de oluşur.

Structure of DynamoDB?



Partition Key ve Sort Key'ler atanır. Partition Key olmak zorunda ancak Sort Key olmak zorunda değil. Bunu eklerseniz bir filtre daha eklemiş oluyorsunuz. Aradığımız veriyi daha kolay bulabiliriz.

Partition key'e Hash Key de denir. Sort Key'e de Range key denir. Çünkü arka planda Hash ve Range algoritmalarına göre tutulur.

Capacity calculator

Read consistent: Eventually consistent ve strongly consistent ve Transactional consistency

Ben bir girdi yaparken başka biri de aynı anda sisteme girdi yapıyor. Eventually'de sen bir sorgu yaptığın an sistemde ne varsa sana sonucu verir. Ancak Strongly'de diğer girdi yapan elemanın irdisini bekler ondan sonra sana sonucu verir. Sonuç olarak Eventually daha hızlı ancak her zaman güncel veriyi veremez.

Transactional da ise daha çok oyun bankacılık sistemlerinde kullanılır. Örneğin banka havalesinde benim hesaptan para gitti karşı tarafın tablosuna yazılmasını beklemeden sonucu gösterdi. O zaman karşı tarafa para ulaşmıyor. Aynı şekilde oyunlarda item gönderirken de oluyor. Bu sistemde iki tablo arasındaki değişiklikler aynı anda yapılıyor. Senin tablodan silinip karşı tabloya yazıldığını görmeden işlem bitmiyor. En yavaş işleyeni bu ancak veri kaybı sıfır. Ayrıca diğer ikisine göre daha pahalı

▼ Capacity calculator

Average item size (KB)

1

Item read/second

1

Read consistency

Eventually consistent ▼

Item write/second

1

Write consistency

Standard ▼

Ortalama item büyüklüğüne göre Read Capacity Unit değişiyor. 1 Reading unit en fazla 8 KB'lık veri okuyabiliyor. unit sayısına göre ücret değişir.

▼ Capacity calculator

Average item size (KB)

17

Item read/second

1

Read consistency

Eventually consistent

Item write/second

1

Write consistency

Standard

Read capacity units

3

Write capacity units

17

Region

us-east-1

Estimated cost

\$10,64 / month

On-Demand ve Provisioned seçenekleri var. On demand ne kadar veri varsa ben okurum sıkıntı yok sen paradan haber ver diyor. Ancak Provisioned kısmında max. min. capacity unit'i belirleeyebiliyorsun. Bu da maddi açıdan daha avantajlı.

Read/write capacity settings [Info](#)

Capacity mode

☐ On-demand

Simplify billing by paying for the actual reads and writes your application performs.

☒ Provisioned

Manage and optimize your costs by allocating read/write capacity in advance.

Read capacity

Auto scaling [Info](#)

Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

☒ On

☐ Off

Minimum capacity units

1

Maximum capacity units

10

Target utilization (%)

70

Write capacity

Auto scaling [Info](#)

Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

☒ On

☐ Off

Minimum capacity units

1

Maximum capacity units

10

Target utilization (%)

70

Bir veriyi ararken Scan işlemi Query'e göre daha pahalı çünkü tüm tabloyu tarayıp sonucu veriyor.