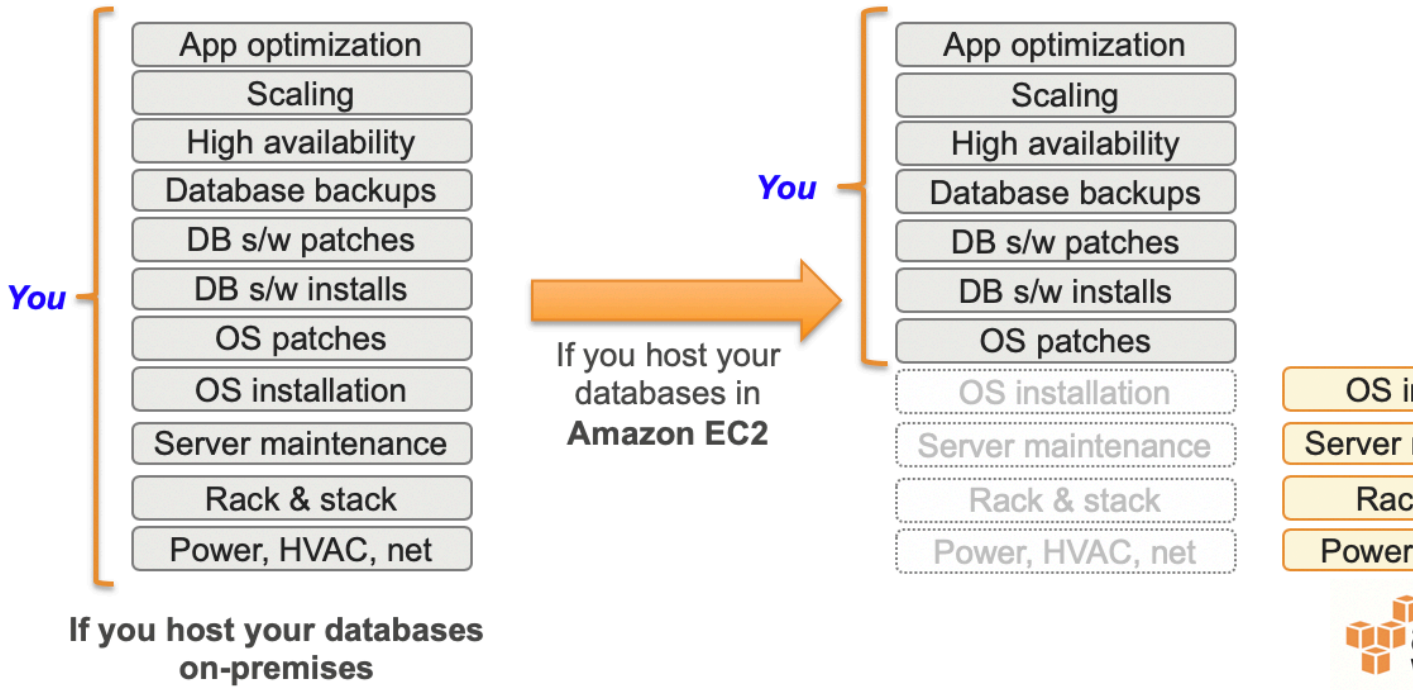
5 Databases

Sunday, 15 September 2019 2:53 PM

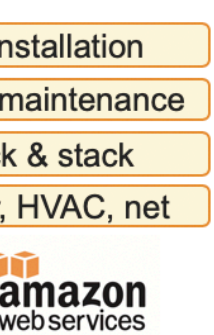
RDS

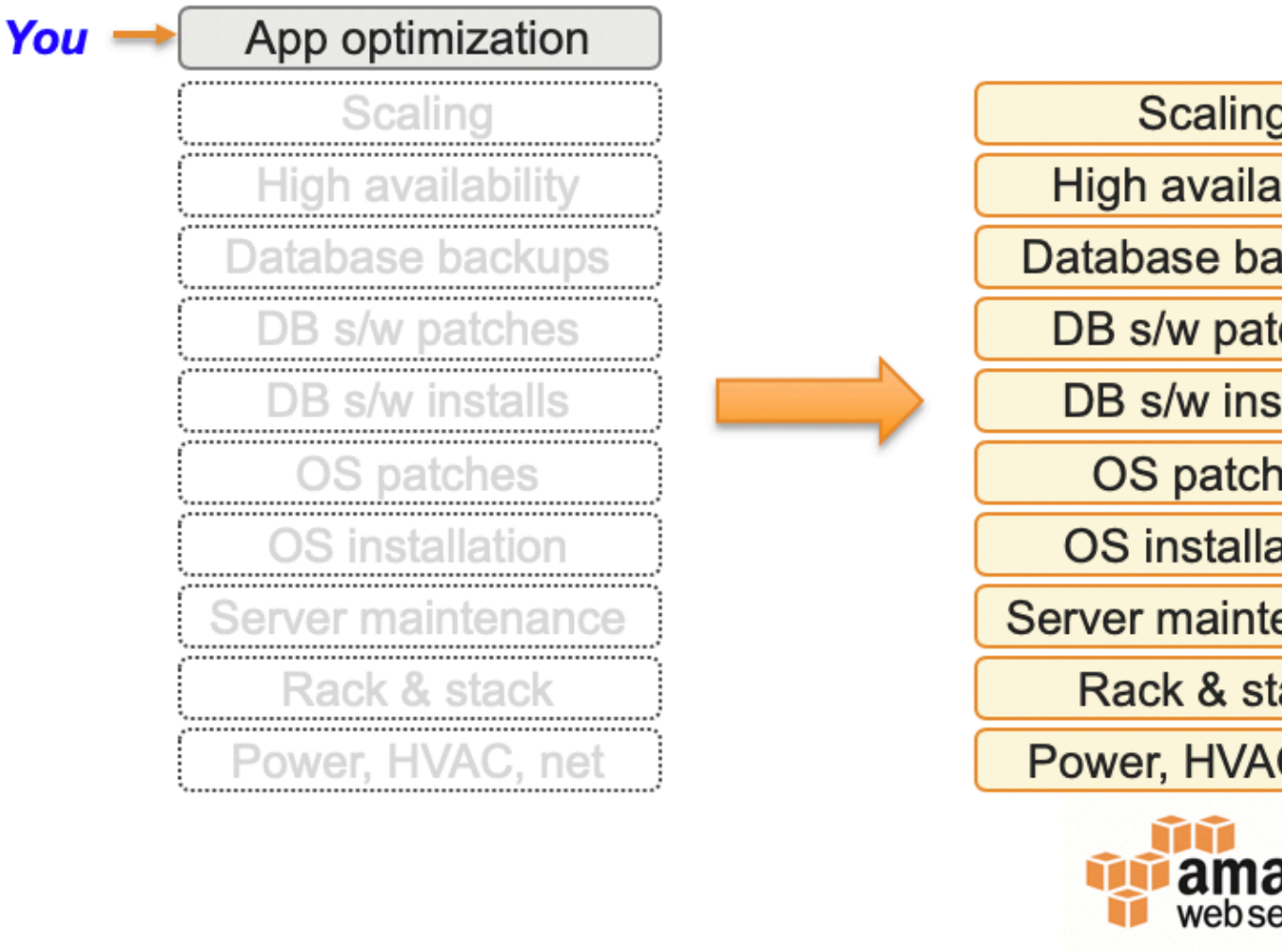
- **Managed Service**
  - DB Setup, HA Replication, Automated Bac Scaling.
- **Resizable Capacity**
  - Compute, Database Storage, Storage IO.
- Deploy **MySQL**, **MariaDB**, **Microsoft SQL Server**, **Oracle**, and **PostgreSQL** database
- **Amazon Aurora** – MySQL, PostgreSQL



kup,

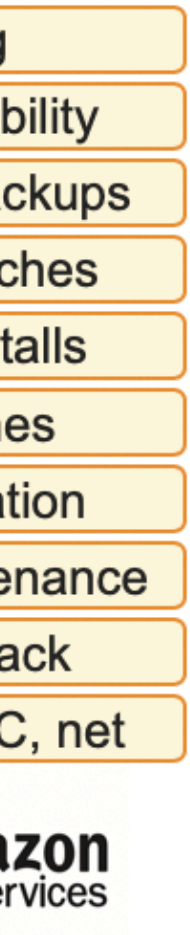
es


nstallation
maintenance
k & stack
, HVAC, net
amazon
web services



- DB Options group
- DB instance class
- Read Replica
- High Availability
- Backup and restore
- Maintenance and patching

Tutorial: Create a Web Server and an Amazon RDS Da



tabase

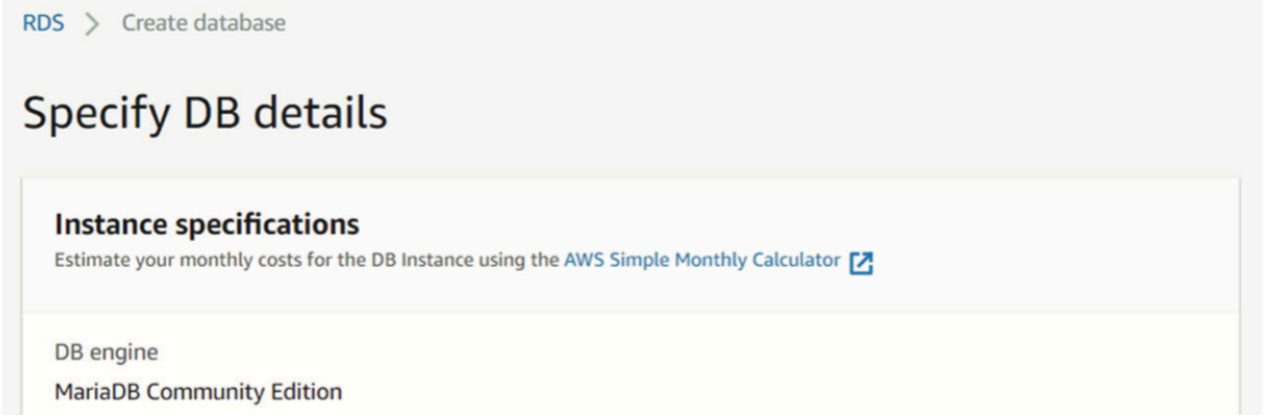
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.htm

Ex 5.1: Create an RDS Database instance

In this exercise, you'll create an RDS database instance using M
database engine. Free Tier accounts won't incur any charges.

1. **Go to the RDS Dashboard.**
2. **Click Create Database.**
3. **On the Select Engine page, select MariaDB and click Nex**
4. **For the use case, select Production - MariaDB. Then click**
5. **For the DB Engine Version, stick with the default.**
6. **For DB Instance Class, select db.t2.micro - 1 vCPU, 1 GB F**
7. **Choose No for Multi-AZ Deployment.**
8. **For Storage Type, select General Purpose (SSD).**
9. **Enter 20 GB as the Allocated Storage setting. Check your example in Figure 5.1 to make sure everything matches.**

Creating an RDS Instance



al

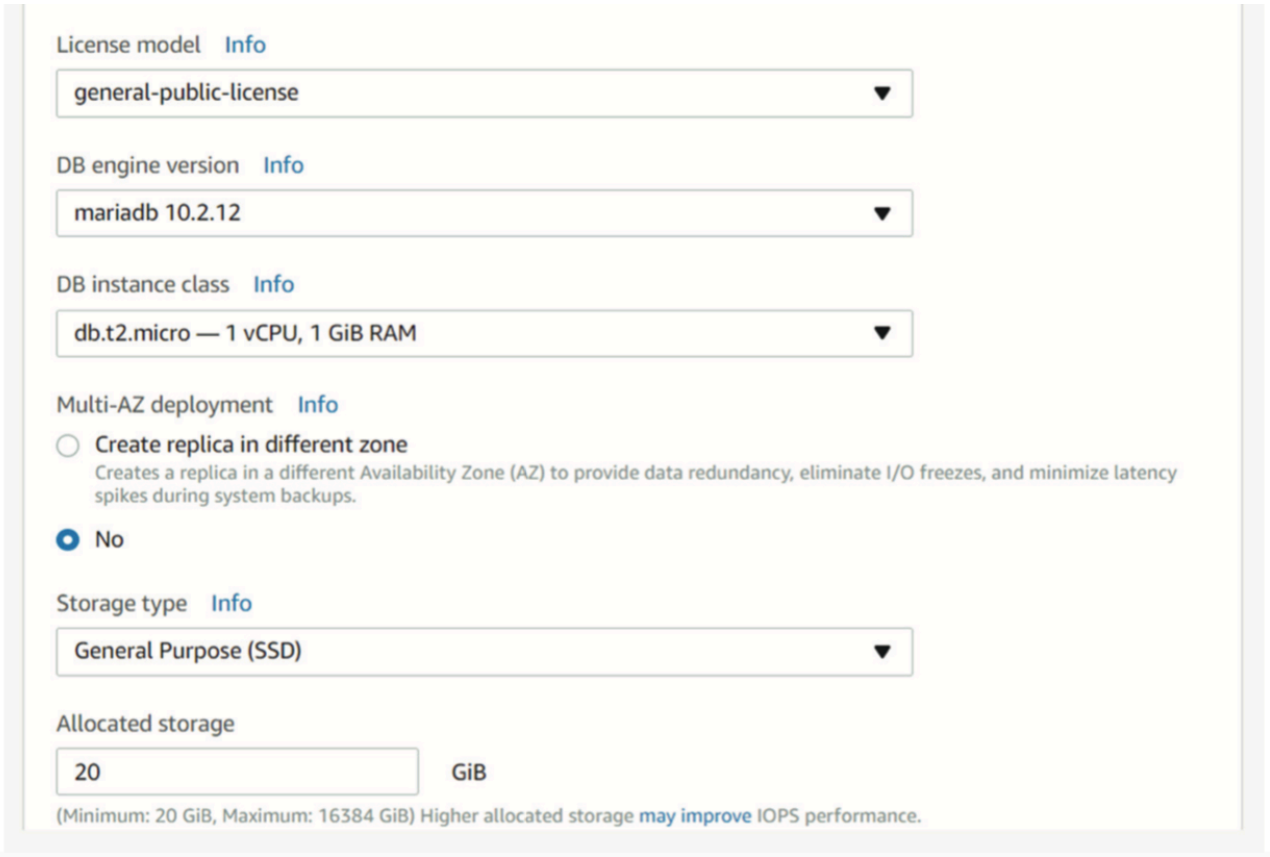
MariaDB as the

t.

Next.

RAM.

entries against the



10. **In the Settings section, enter mydbinstance in the DB Ins field.**
11. **Enter a master username and master password of your c Next.**
12. **In the Network & Security section, choose the Default VI Subnet Group settings.**
13. **For the Public Accessible setting, choose No.**
14. **For Availability Zone, select No Preference.**
15. **For VPC Security Group(s), select Create New VPC Securi**
16. **In the Database Options section, enter mydatabase in th field.**
17. **Leave Database Port set to the default of 3306.**
18. **Leave all the other settings at their defaults and click Cre**
19. **Click View DB Instance Details to view the new database**
20. **Keep the instance up and running, as you'll need it for Ex**

stance Identifier

choice and then click

PC and Default

ty Group.
ne Database Name

eate Database.
e instance.
xercise 5.2.

Ex 5.2:
Create a read replica

1. **In the RDS Dashboard, click Instances.**
2. **Select the instance you created earlier, click Instance Act**
   Create
   Read Replica.
3. **Under the Network & Security heading, stick with the de**
   No for Publicly Accessible.
4. **Under the Settings section, make sure the Read Replica S**
   mydbinstance.
5. **Enter mydbinstance-rr in the Database Instance Identifie**
6. **Click the Create Read Replica button.**

Ex 5.3: Promote the read replica to a master

1. **In the RDS console, click Instances.**
2. **Select the read replica you created.**
3. **Click Instance Actions and then select Promote Read Rep**
4. **Click Continue.**
5. **Click Promote Read Replica.**

Redshift

NoSQL Databases

tions, and then click

efaults, but choose

Source is

er field.

plica.

NoSQL Databases

- **Schemaless**
- **Primary Key (Partition Key)**
- **Denormalized tables**
- **Intelligent Keys**
- **Optimized for queries based on PK**

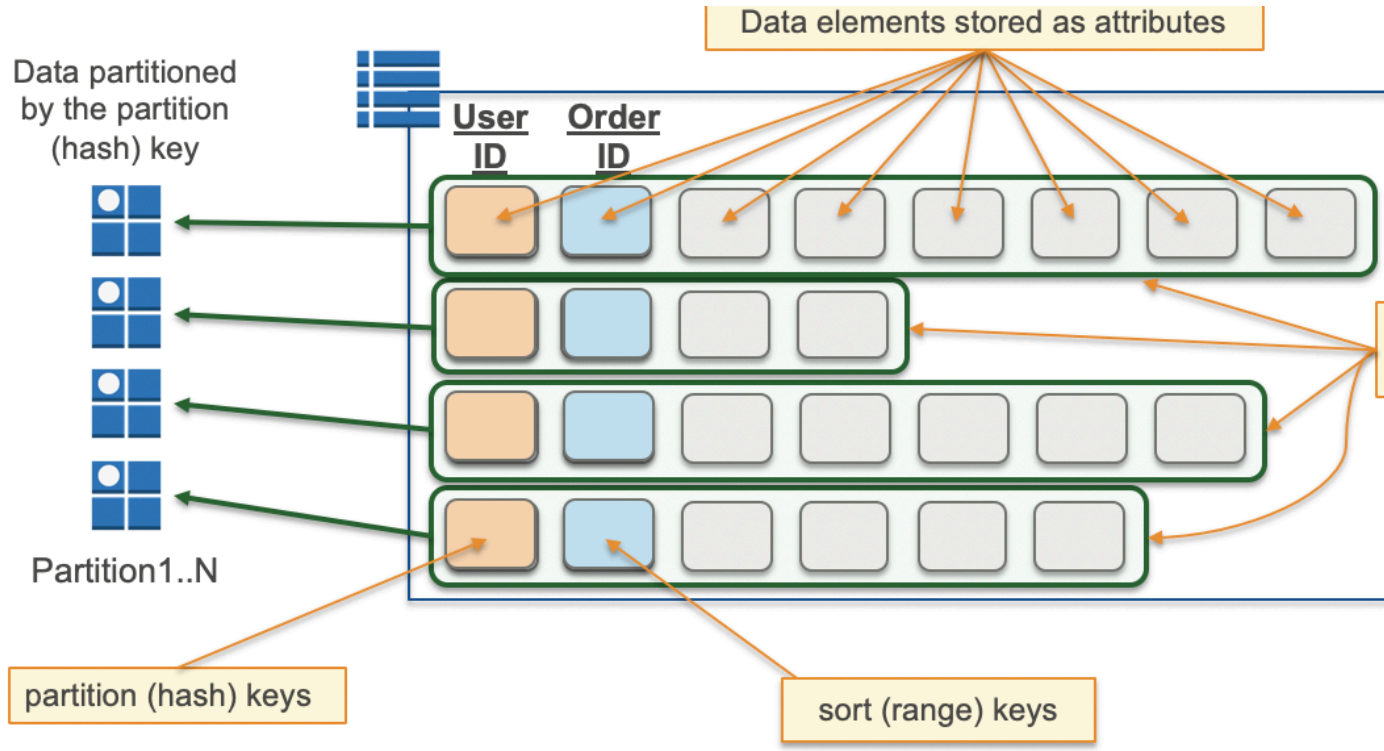
Types of "non-relational databases"

DynamoDB

- Managed/Serverless
- Large reads/writes
- Single digit milliseconds latency
- Scales by using partitions
- SSD
- Partition Key/Hash Key
- Tables/Items/Attributes
- Consistency

DynamoDB Data Model

Table = collection

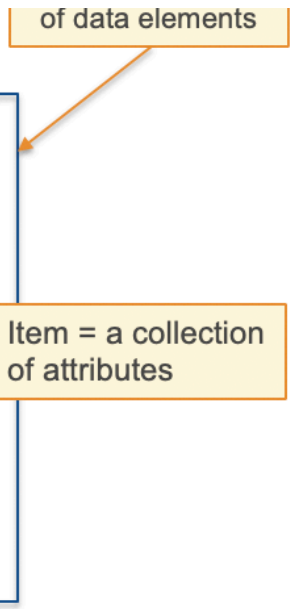


Tables Items and Attributes

People


```
{
    "PersonID": 101,
    "LastName": "Smith",
    "FirstName": "Fred",
    "Phone": "555-4321"
}
```

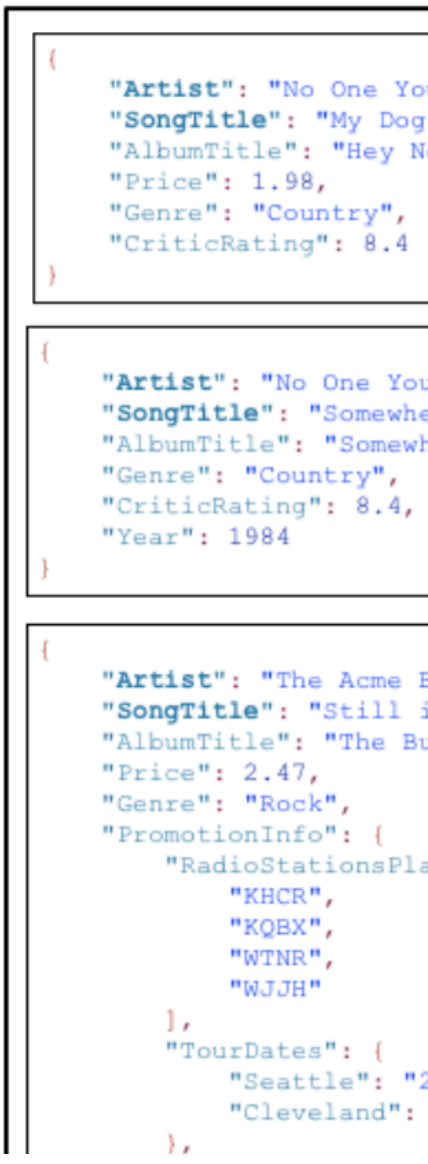
```
{
    "PersonID": 102,
    "LastName": "Jones",
    "FirstName": "Mary",
    "Address": {
        "Street": "123 Main",
        "City": "Anytown",
        "State": "OH",
        "ZIPCode": 12345
    }
```


of data elements
Item = a collection
of attributes

```
}
```

```
{
    "PersonID": 103,
    "LastName": "Stephens",
    "FirstName": "Howard",
    "Address": {
        "Street": "123 Main",
        "City": "London",
        "PostalCode": "ER3 5K8"
    },
    "FavoriteColor": "Blue"
}
```

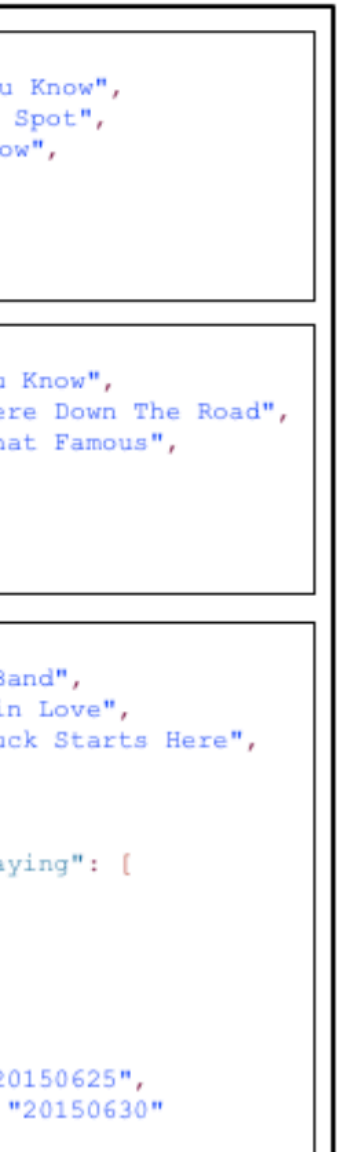
Music

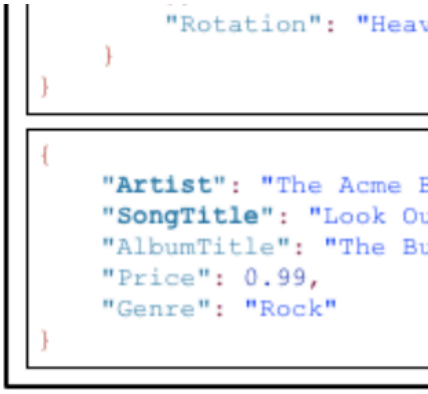


```
{
    "Artist": "No One Yo
    "SongTitle": "My Dog
    "AlbumTitle": "Hey N
    "Price": 1.98,
    "Genre": "Country",
    "CriticRating": 8.4
}
```

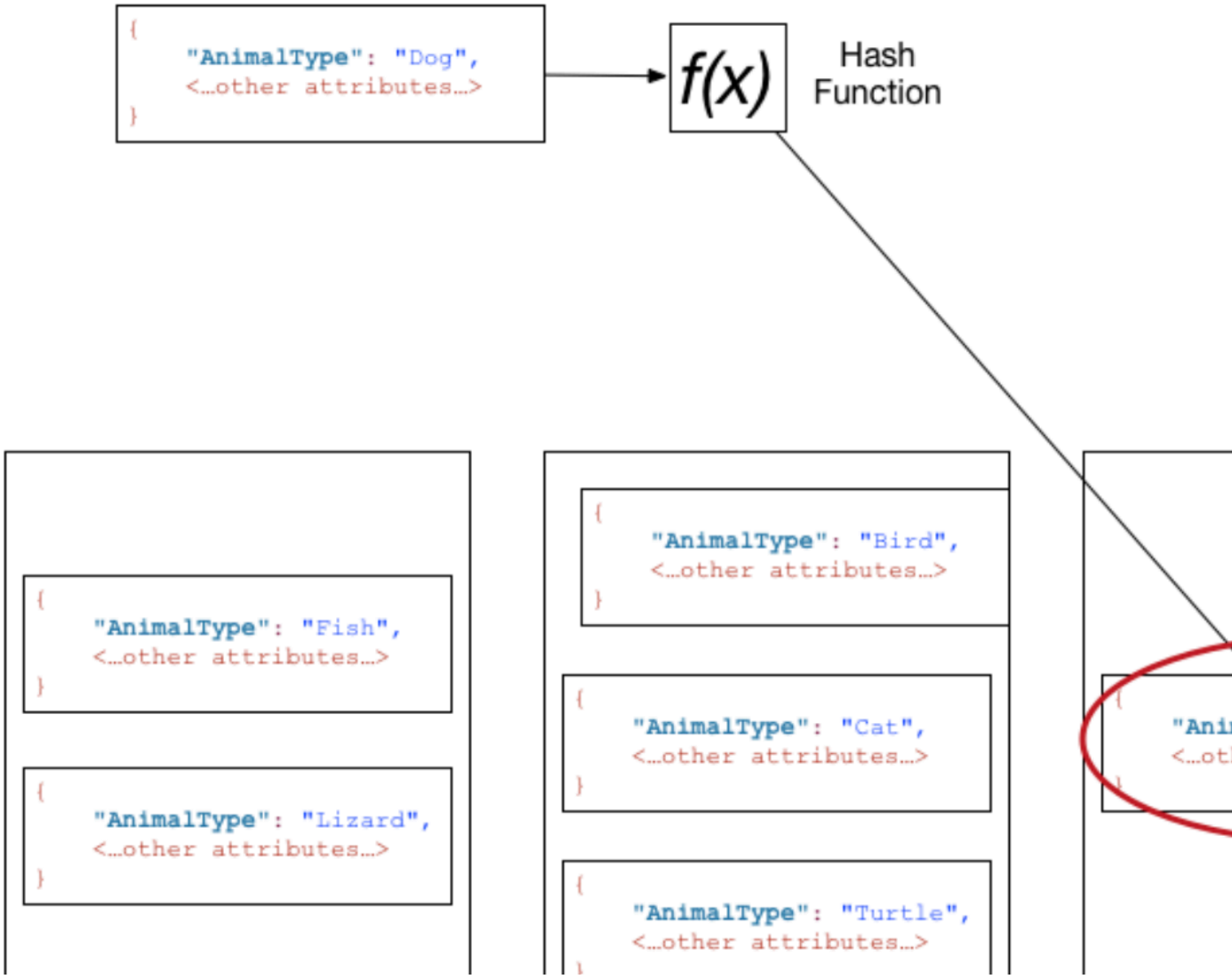
```
{
    "Artist": "No One You
    "SongTitle": "Somewhe
    "AlbumTitle": "Somewh
    "Genre": "Country",
    "CriticRating": 8.4,
    "Year": 1984
}
```

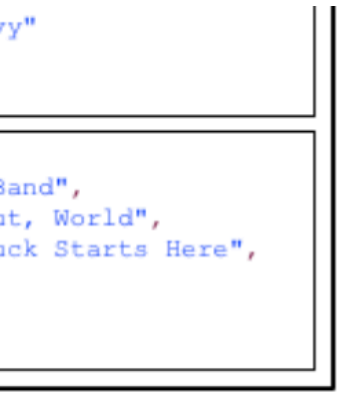
```
{
    "Artist": "The Acme B
    "SongTitle": "Still i
    "AlbumTitle": "The Bu
    "Price": 2.47,
    "Genre": "Rock",
    "PromotionInfo": {
        "RadioStationsPla
            "KHCR",
            "KQBX",
            "WTNR",
            "WJJH"
        ],
        "TourDates": {
            "Seattle": "2
            "Cleveland":
        },
```

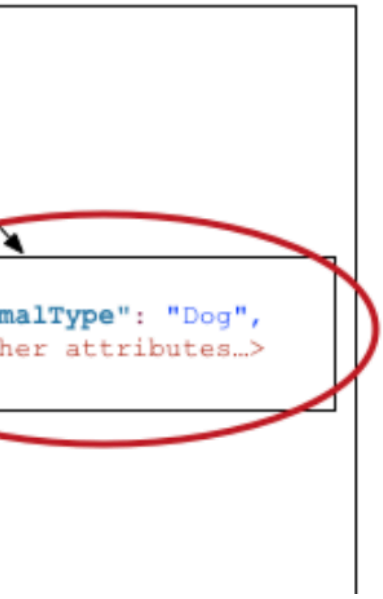

u Know",
Spot",
ow",
n Know",
ere Down The Road",
hat Famous",
Band",
In Love",
ack Starts Here",
aying": [
20150625",
"20150630"



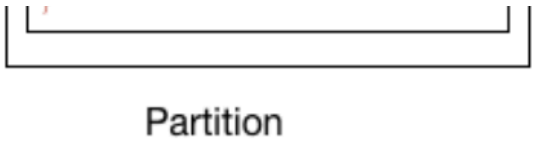
Paritition Keys




yy"
Band",
nt, World",
ack Starts Here",

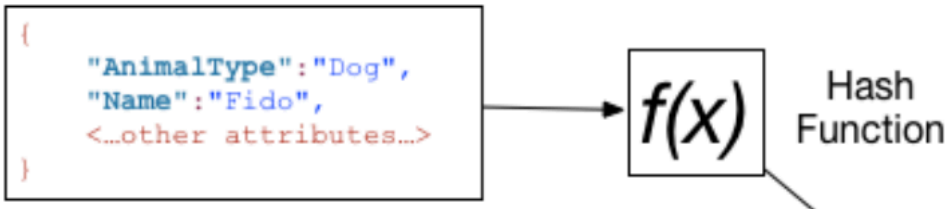

malType": "Dog",
her attributes…>

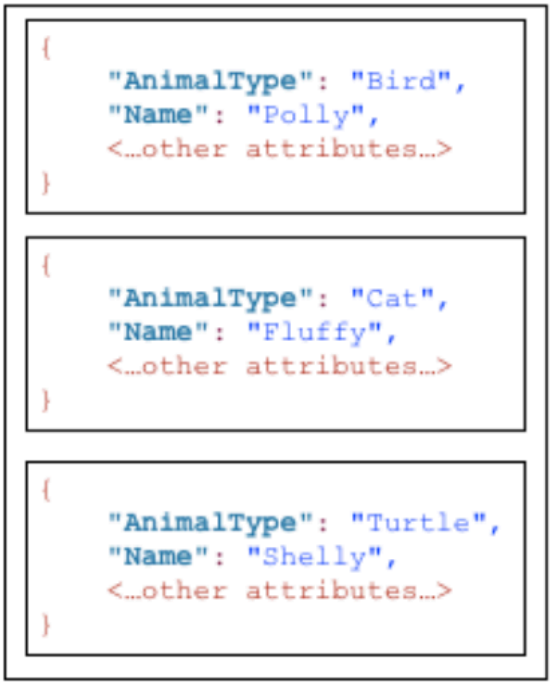
Partition



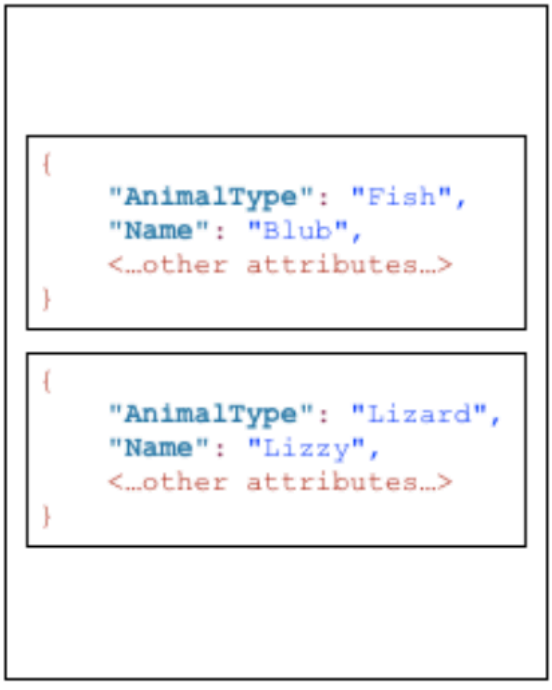
Partition

Partition and Sort Keys



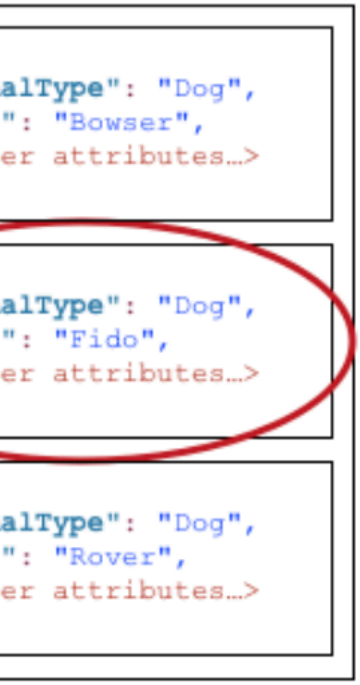


Partition



Partition

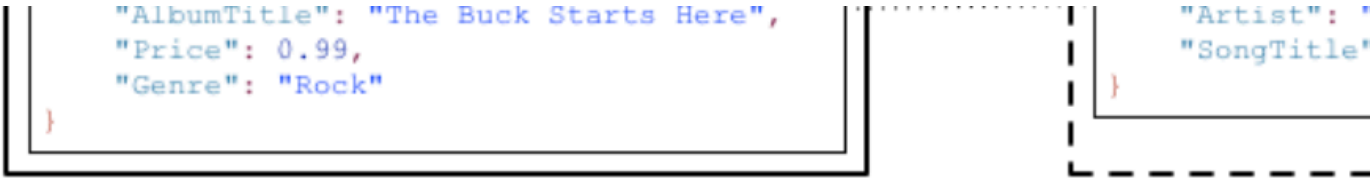
Partition



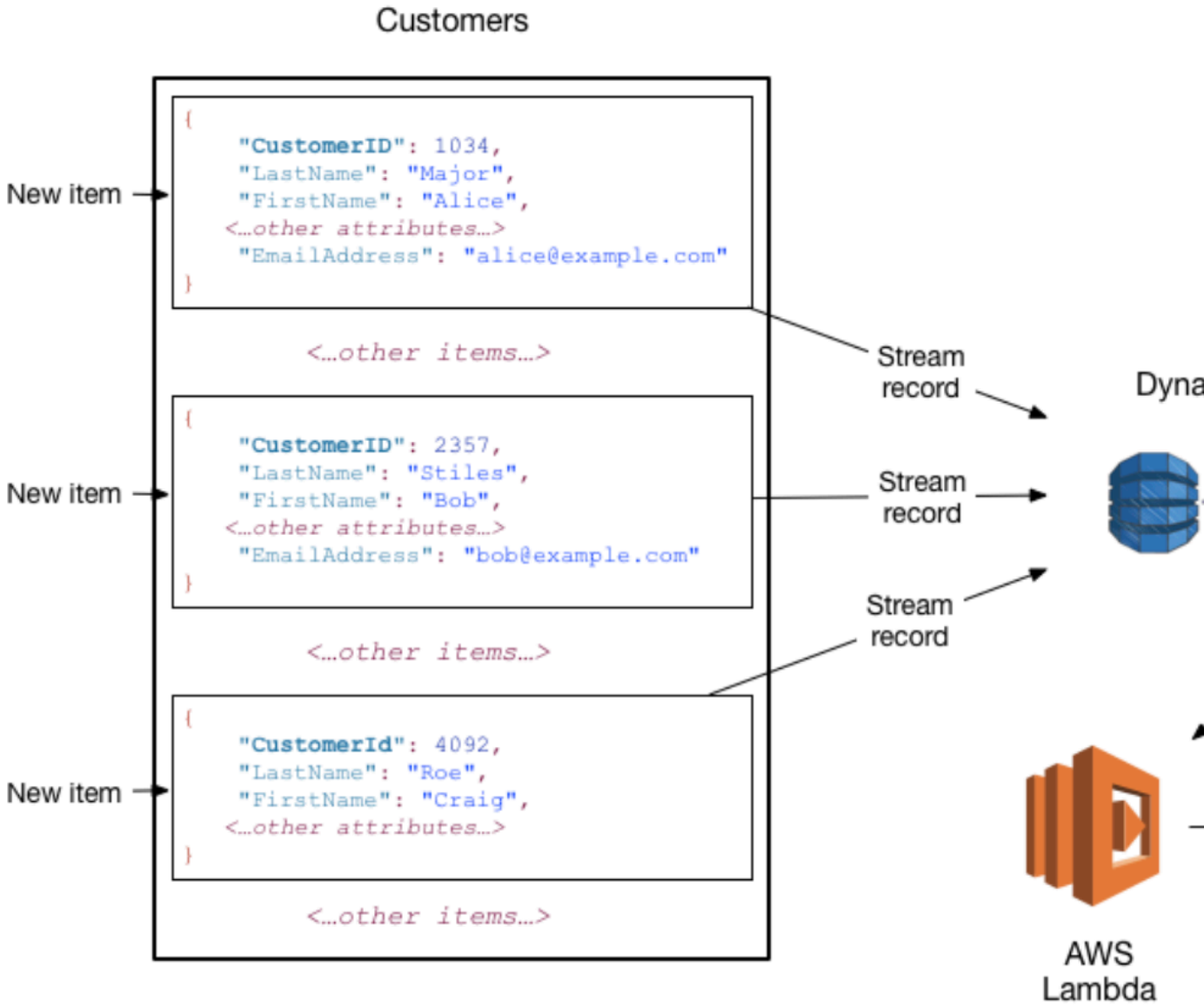
Partition

Secondary Index

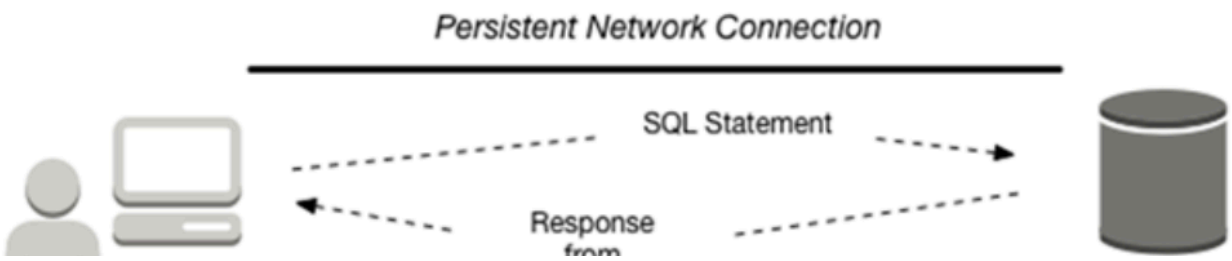
nreAlbumTitle



DDB Streams



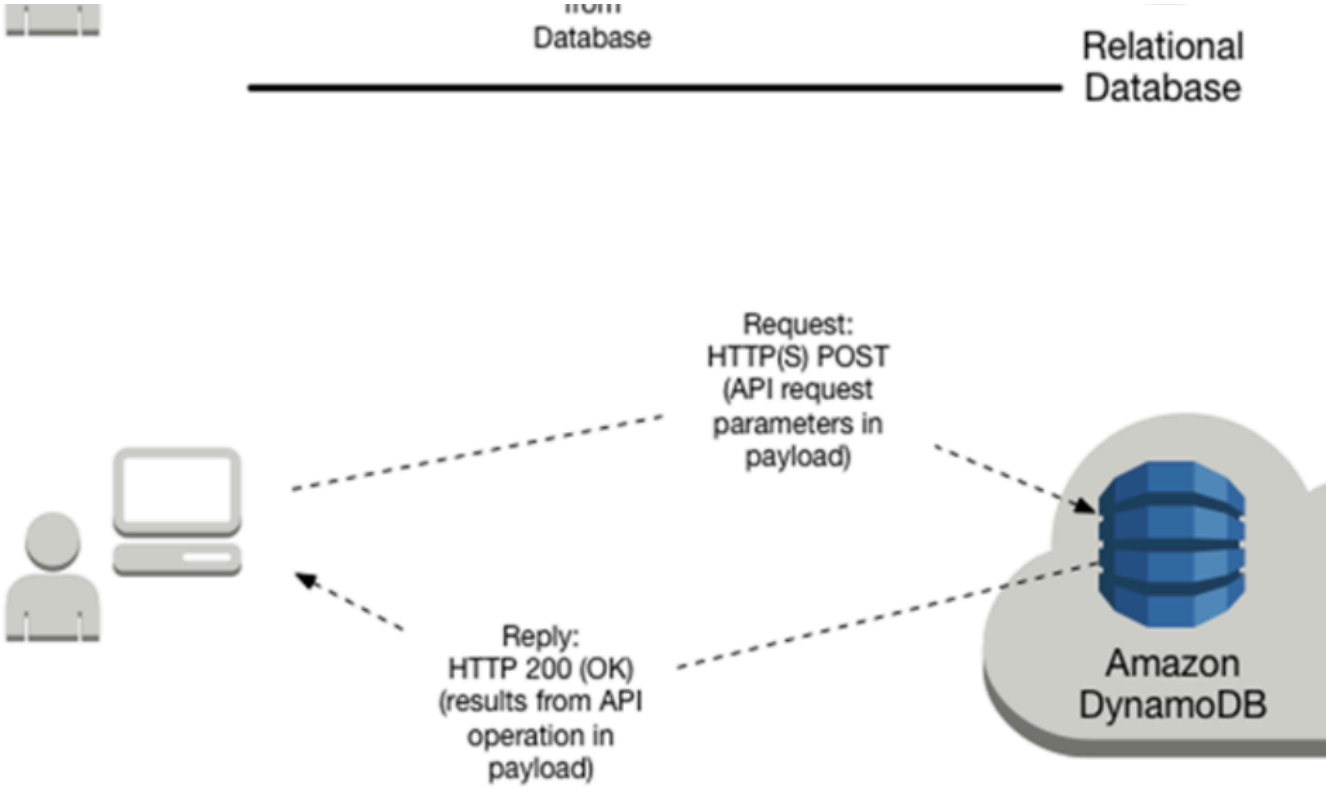
The following diagram shows a client's interaction with a relational database, and with Dynan



```
'The Acme Band",
': "Look Out, World"
```

amoDB Streams

noDB.



https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStartedDI

RCU/WCU

DDB Best Practices

- Keep item size small.
- Store metadata in DynamoDB and large I in Amazon S3.

DynamoDB.htm

BLOBs

- Use table per day, week, month, etc., for time series data.
- Avoid hot keys and hot partitions.

Understand the differences between relational and nonrelat
Know the different database engines RDS supports.
Be able to select the right instance class and storage type giv storage requirements.
Understand the differences between multi-AZ and read repli
Be able to determine the appropriate primary key type for a table
Know how DynamoDB throughput capacity works

storing

tional databases

ven specific

icas
DynamoDB