

# AWS Gyakorlat Database

Felde Imre

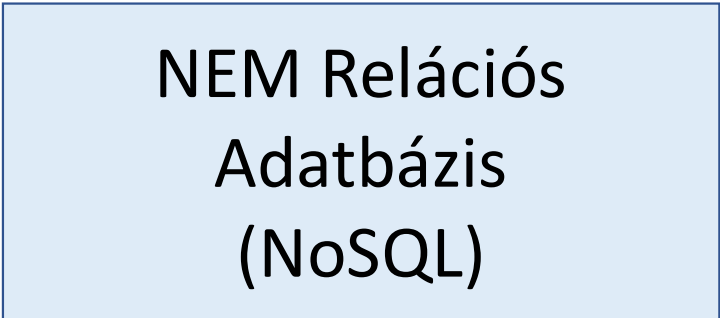
# AWS Adatbázis koncepció



Relációs Adatbázis

**Strukturált adatok**

AWS = Relációs DB szolgáltatás



NEM Relációs  
Adatbázis  
(NoSQL)

**NEM Strukturált adatok**

AWS = Dynamo DB szolgáltatás

# AWS Adatbázis koncepció

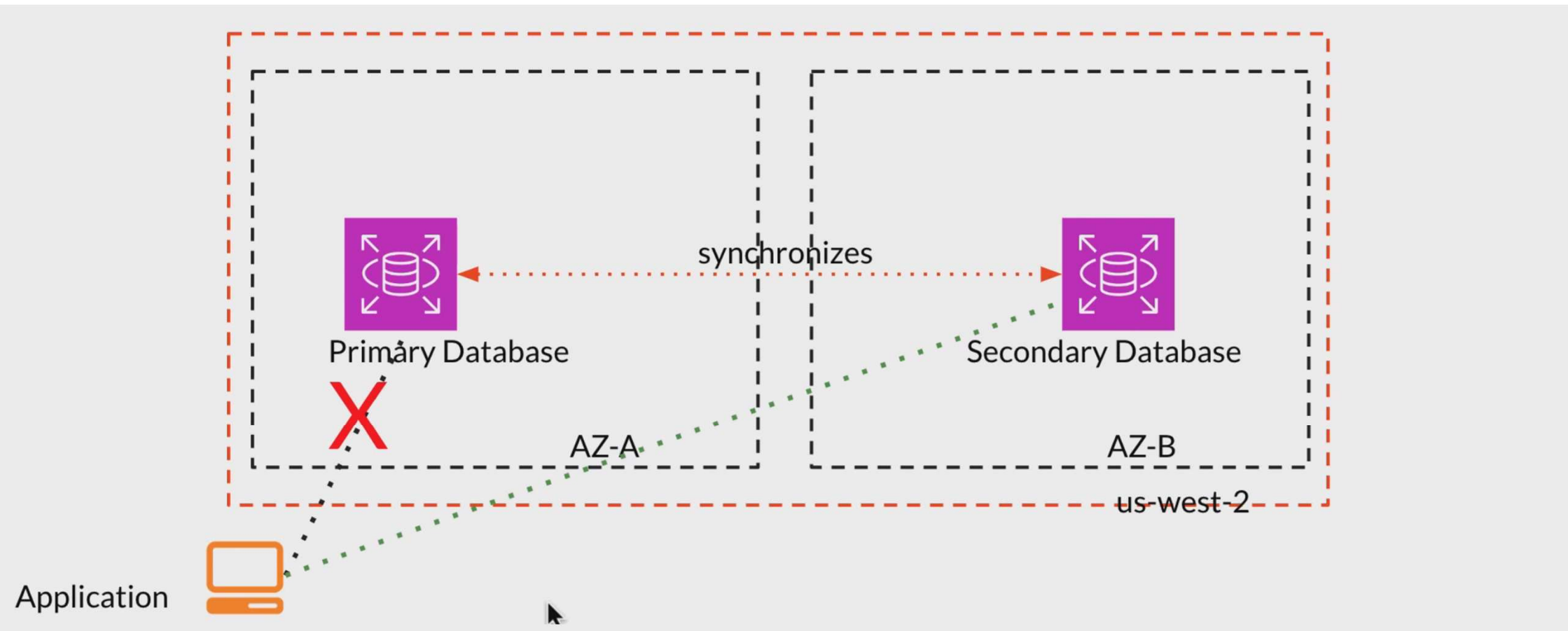
- **Felügyelt adatbázis-szolgáltatás (Managed Database Service)**

- Az Amazon elvégzi az adminisztratív feladatokat (helyreállítás, biztonsági mentés, belső konfigurációk stb.)
- Nekünk csak a saját igényeinknek megfelelően kell beállítani, és... máris készen áll az adatbázis!

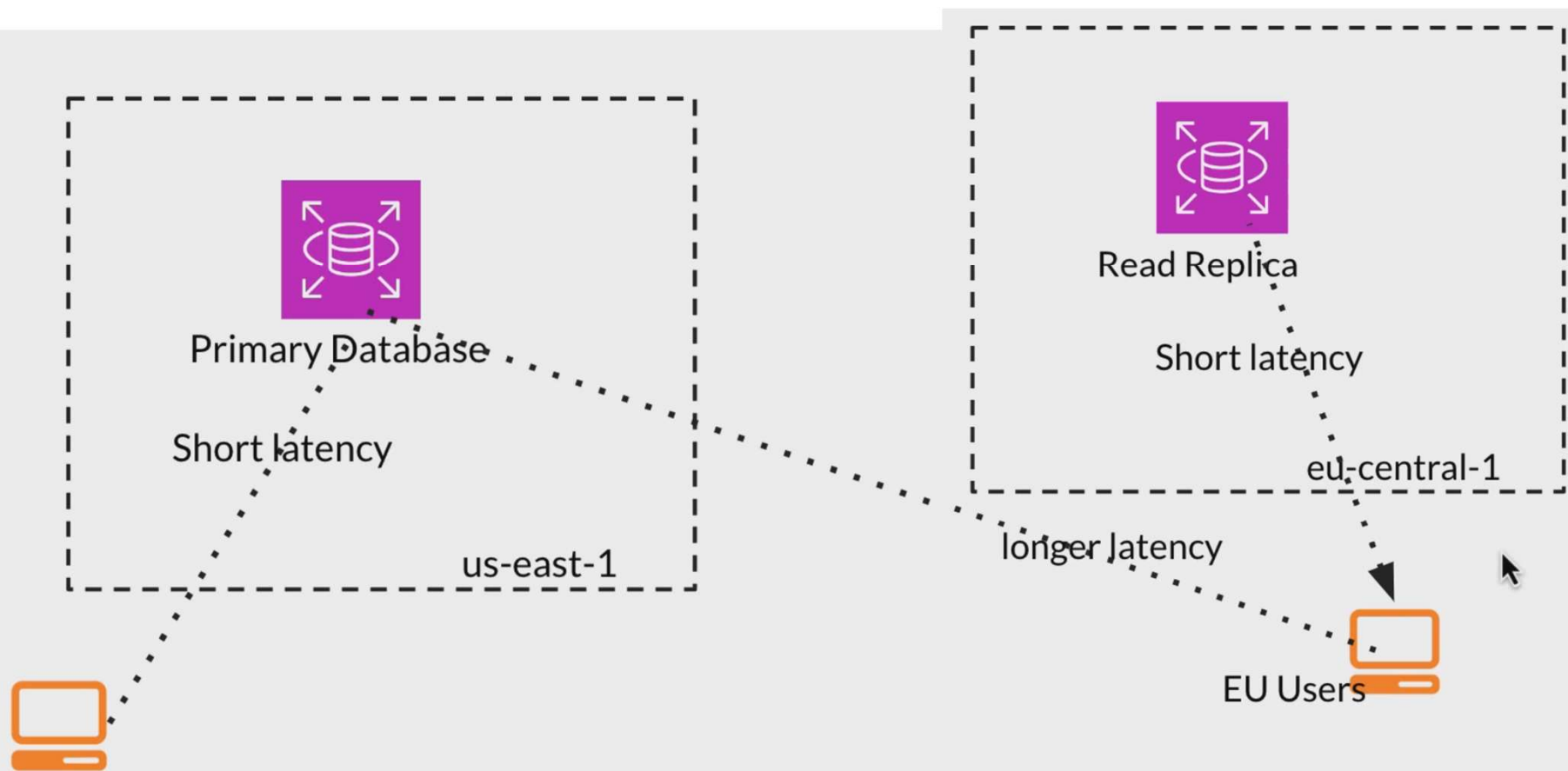
- **Az RDS hat különböző adatbázis-motort támogat:**

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

# AWS AZ rugalmasság



# AWS RDS – Konfigurálható másodpéldányok



# AWS RDS - Autóskálázás

- **Felskálázás (Upscaling)** – az adatbázisnak több erőforrásra (például CPU-ra vagy tárhelyre) van szüksége.
- **Leskálázás (Downscaling)** – az adatbázisnak kevesebb erőforrásra van szüksége a költségek megtakarítása érdekében.
- **Könnyen megvalósítható** az automatikus skálázási szabályok (auto-scaling policies) beállításával az RDS-ben, és az AWS elintézi a többit!

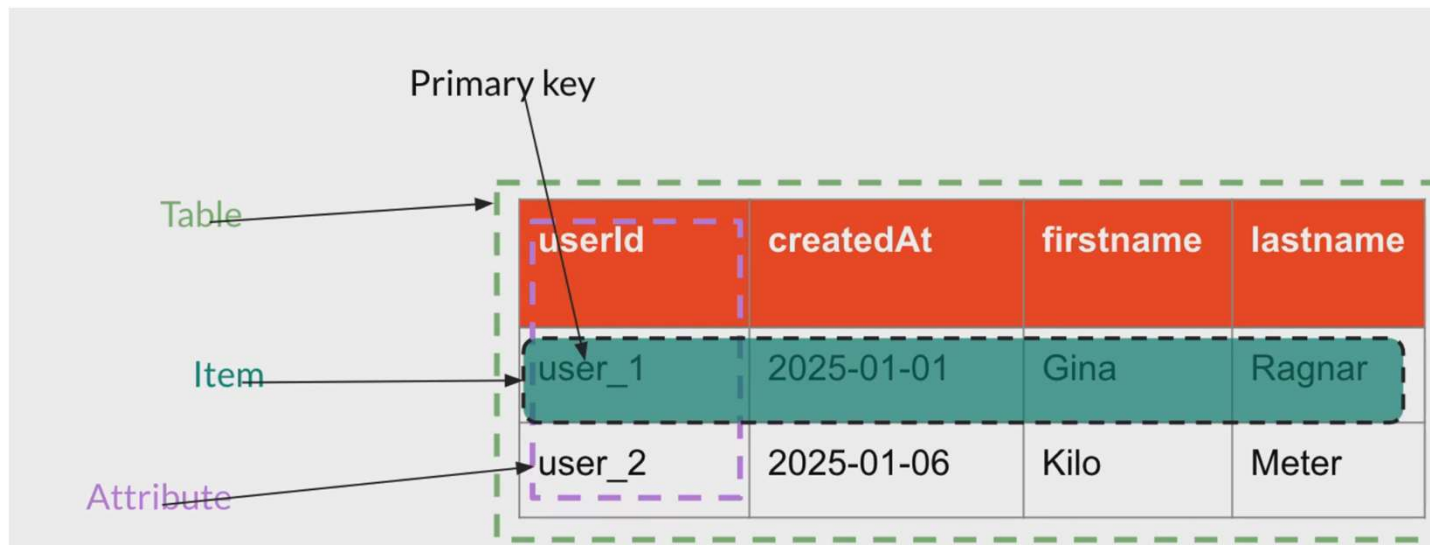
# DnyamoDB – NoSQL Database



- **Teljesen felügyelt NoSQL adatbázis** – könnyen kezelhető, és bármilyen méretű terhelést képes kezelni.
- **NoSQL** – azt jelenti, hogy nem relációs adatbázis, mint az RDS.
- **Nincs séma (No Schema)** – Bármit írhatasz az adatbázisba, nincs előre meghatározott szerkezet! (kivéve az elsődleges kulcsot)
- **Nem relációs (Non-relational)** – (Igen, de... az adatoknak továbbra is vannak kapcsolataik\* – nem lehet táblák között **\*\*JOIN\*\*** műveleteket végezni, de az adatok között ettől még léteznek kapcsolatok.

# DynamoDB – Alapvető koncepció

- **Tábla (Table)** – elemek gyűjteménye
- **Elemek (Items)** – egy tábla sora
- **Attribútumok (Attributes)** – egy elem mezői, hasonlóan a JSON dokumentum kulcsaihoz
- **Típusok (Types)** – Szám (Number), Szöveg (String), Logikai érték (Boolean)...





# DynamoDB – Alapvető koncepció

- Az adatok JSON formában kerülnek tárolásra

```
{
  "userId": "user_1",
  "createdAt": "2025-01-01",
  "firstname": "Gina",
  "lastname": "Ragnar"
}
```

A user in normal JSON

```
{
  "userId": {
    "S": "user_1"
  },
  "createdAt": {
    "S": "2025-01-01"
  },
  "firstname": {
    "S": "Gina"
  },
  "Lastname": {
    "S": "Ragnar"
  }
}
```

A user in DynamoDB JSON

# DynamoDB

Database

## Amazon DynamoDB

A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

### How it works

aws

What is Amazon DynamoDB? | Amazon We...

Link masol...

### Get started

Create a new table to start exploring DynamoDB.

Create table

### Pricing

DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to turn on. DynamoDB has on-demand capacity mode and provisioned capacity mode, and these modes have pricing for processing reads and writes on your tables.

[Learn more about pricing](#)

aws

Search

[Alt+S]

DynamoDB

Tables

Create table

## Create table

### Table details

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

#### Table name

This will be used to identify your table.

USER-table-01

Between 3 and 255 characters, containing only letters, numbers, underscores (`_`), hyphens (`-`), and periods (`.`).

#### Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data.

user-id

String

1 to 255 characters and case sensitive.

#### Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing

Enter the sort key name

String

1 to 255 characters and case sensitive.

### Table settings

☒ Default settings

☐ Customize settings

The fastest way to create your table. You can modify most of these settings after your table has been created. To modify these settings now, choose 'Customize settings'.

### Default table settings

These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value
Table class	DynamoDB Standard

# DynamoDB

aws [Search] [Alt+S]

DynamoDB > Tables > USER-table-01

## USER-table-01

Settings Indexes Monitor Global tables Backups Ex

**Protect your DynamoDB table from accidental writes and deletes**  
When you turn on point-in-time recovery (PITR), DynamoDB backs up your table data a second in the preceding 1 to 35 days. Additional charges apply. [Learn more](#)

### General information [Info](#)

<b>Partition key</b> user-id (String)	<b>Sort key</b> -	<b>Capacity mode</b> On-demand
<b>Alarms</b> No active alarms	<b>Point-in-time recovery (PITR)</b> <a href="#">Info</a> Off	<b>Item size</b> 0 bytes
<b>Average item size</b> 0 bytes	<b>Resource-based policy</b> <a href="#">Info</a> Not active	
<b>Amazon Resource Name (ARN)</b> <a href="#">arn:aws:dynamodb:eu-north-1:493354280892:table/USER-table-01</a>		

► Additional info

### Read/write capacity [Info](#)

The read/write capacity mode controls how you are charged for read and write throughput.

Capacity mode

aws [Search] [Alt+S] Europe (Stockholm) Account ID: 493354280892

DynamoDB > Explore items > USER-table-01

✓ The item has been saved successfully.

## USER-table-01 [Autopreview](#) [View table](#)

**Tables (1)**

Any tag key  
Any tag value  
[Find tables](#)

< 1 > ⚙

USER-table-01 ☆

### Scan or query items

☒ Scan ☐ Query

Select a table or index: Table - USER-table-01

Select attribute projection: All attributes

► Filters - optional

[Run](#) [Reset](#)

✓ **Completed** · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 2

### Table: USER-table-01 - Items returned (2)

Scan started on October 14, 2025, 18:52:40

< 1 >

	user-id (String)	age	name
<input type="checkbox"/>	2	22	Charles
<input type="checkbox"/>	1	21	Amy

[Add new attribute](#)

[Form](#) | [JSON](#)

[Remove](#) [Remove](#) [Cancel](#) [Create](#)

# DynamoDB

The screenshot displays the AWS DynamoDB console interface. The top navigation bar shows the AWS logo, a search bar, and the region 'Europe (Stockholm)'. The breadcrumb trail indicates the path: **DynamoDB** > **Explore items** > **USER-table-01**.

**Left Sidebar: Tables (1)**

- Any tag key
- Any tag value
- Find tables
- USER-table-01** (selected)

**Main Content: USER-table-01**

**Scan or query items**

- Scan** (selected) / Query
- Select a table or index: Table - USER-table-01
- Select attribute projection: All attributes

**Filters - optional**

Attribute name	Condition	Type	Value	
user-id	Greater than	String	0	<a href="#">Remove</a>

[Add filter](#)

**Run** [Reset](#)

**Completed** · Items returned: 2 · Items scanned: 2 · Efficiency: 100% · RCUs consumed: 2

**Table: USER-table-01 - Items returned (2)** [Actions](#)

Scan started on October 14, 2025, 19:00:07

	user-id (String)	age	name
<input type="checkbox"/>	<a href="#">2</a>	22	Charles
<input type="checkbox"/>	<a href="#">1</a>	21	Amy