

AWS Gyakorlat Database

Felde Imre

AWS Adatbázis koncepció

Relációs Adatbázis

NEM Relációs
Adatbázis
(NoSQL)

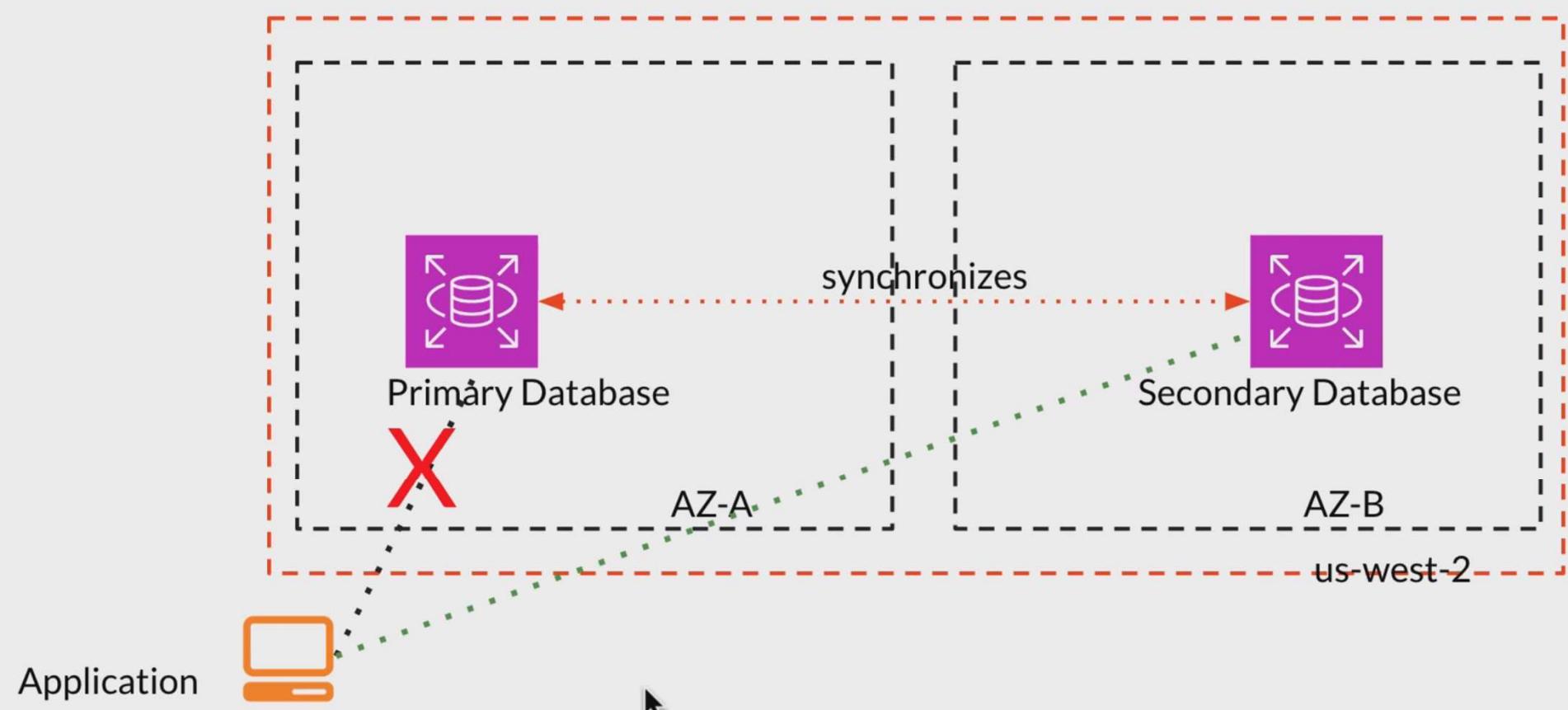
Strukturált adatok
AWS = Relációs DB szolgáltatás

NEM Strukturált adatok
AWS = Dynamo DB szolgáltatás

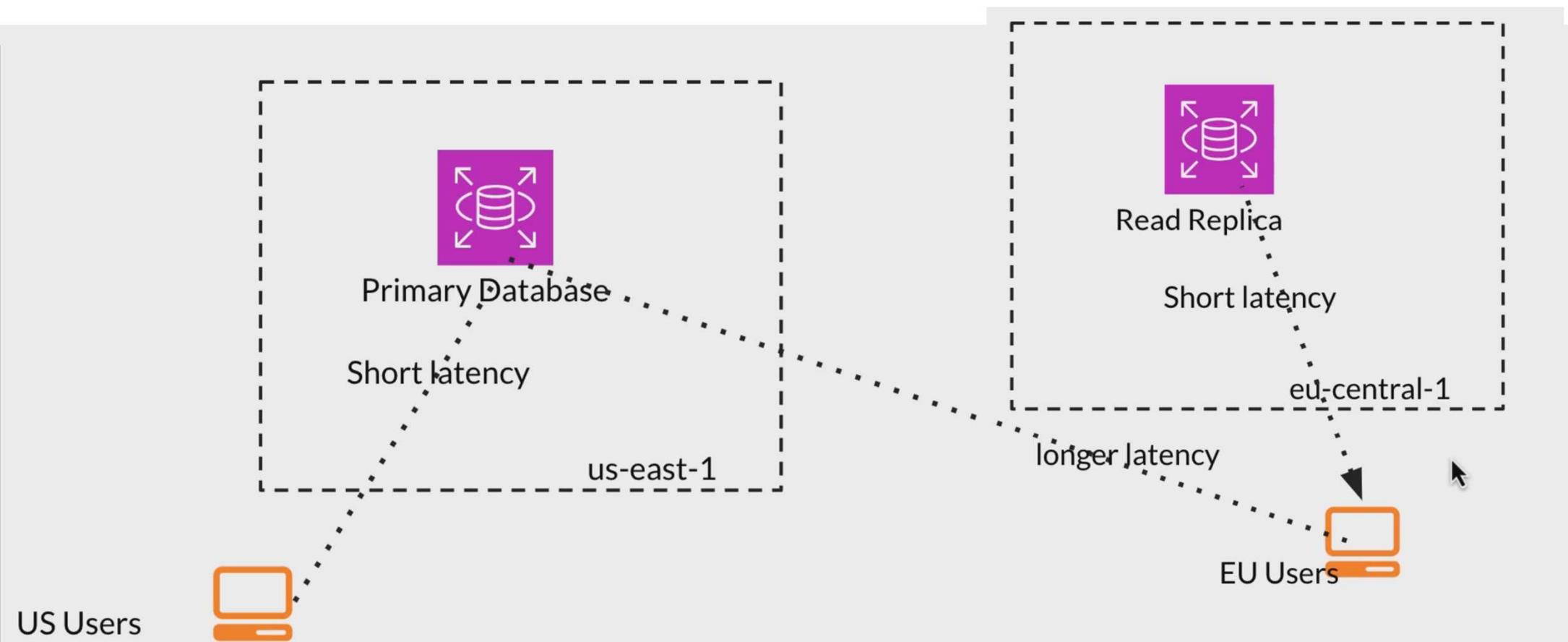
AWS Adatbázis koncepció

- **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... már 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!

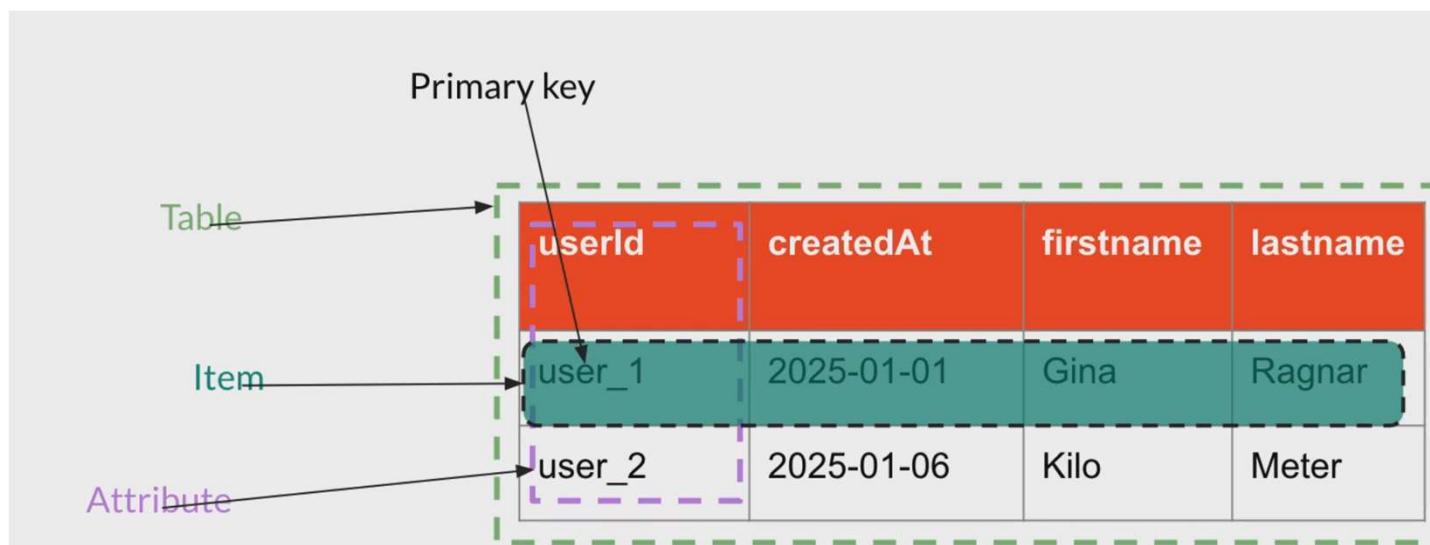
DnyamoDB – NoSQL Database



- **Teljesen felügyelt NoSQL adatbázis** – könnyen kezelhető, és bármilyen méretű terhelést képes kezelní.
- **NoSQL** – azt jelenti, hogy nem relációs adatbázis, mint az RDS.
- **Nincs séma (No Schema)** – Bármit írhatsz 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

The screenshot shows the Amazon DynamoDB homepage. At the top, there's a navigation bar with the AWS logo, a search bar, and account information. Below the header, a blue banner encourages users to share feedback. The main content area features a large heading "Amazon DynamoDB" and a sub-headline "A fast and flexible NoSQL database service for any scale". A paragraph explains that DynamoDB is a fully managed, key-value, and document database. On the right side, there's a "Get started" section with a "Create table" button, which is circled in red. Below this, there's a "Pricing" section with a link to learn more about pricing.

The screenshot shows the "Create table" wizard in the AWS DynamoDB console. The top navigation bar includes the AWS logo, a search bar, and a breadcrumb trail: DynamoDB > Tables > Create table. The main title is "Create table". Under "Table details", it says "DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table". The "Table name" field is set to "USER-table-01". The "Partition key" is set to "user-id" with a type of "String". The "Sort key - optional" field is empty. In the "Table settings" section, the "Default settings" option is selected, with a note that it's the fastest way to create the table. The "Customize setting" option is also available. The "Default table settings" section lists the table class as "DynamoDB Standard".

DynamoDB

Screenshot of the AWS DynamoDB console showing the configuration of the 'USER-table-01' table.

General information

- Partition key: user-id (String)
- Sort key: -
- Alarms: No active alarms
- Average item size: 0 bytes
- Amazon Resource Name (ARN): arn:aws:dynamodb:eu-north-1:493354280892:table/USER-table-01

Read/write capacity

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

Screenshot of the AWS DynamoDB console showing the 'USER-table-01' table details and a successful scan operation.

Tables (1)

- Any tag key
- Any tag value
- Find tables
- USER-table-01

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

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

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

DynamoDB

The screenshot shows the AWS DynamoDB console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and account information for Europe (Stockholm). Below the navigation, the path is shown as **DynamoDB > Explore items > USER-table-01**. On the left, a sidebar titled "Tables (1)" lists "USER-table-01". The main area is titled "USER-table-01" and contains sections for "Scan or query items" and "Filters - optional". Under "Scan or query items", the "Scan" button is selected. Under "Filters - optional", there is one filter defined: "user-id" is Greater than 0. Below these sections are "Run" and "Reset" buttons. A success message at the bottom says "Completed · Items returned: 2 · Items scanned: 2 · Efficiency: 100% · RCUs consumed: 2". The table section below shows two items:

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