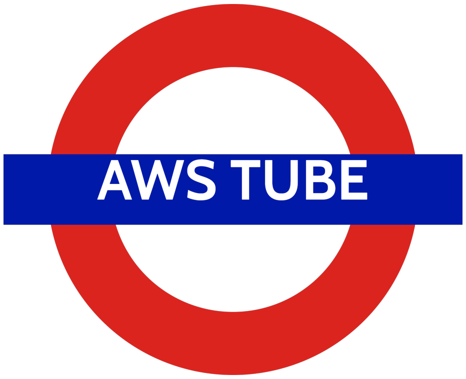

Figure 1: CLICK HERE TO ENTER STATION

Recovery Sheet 7

SJ Green

Recovering from the session delivered on the night of Wednesday 26th February 2025. We talked about databases on AWS. You can view the recovery sheet for the previous week here.

1 Active-Active

Description

1. **Database** “A database is a place to store data in an organised way” (Alice Zhao 2021). “In computing, a database is an organized collection of data or a type of data store based on the use of a database management system (DBMS), the software that interacts with end users, applications, and the database itself to capture and analyze the data” Wikipedia.

2. **SQL** Stands for Structured Query Language. “Imagine you have an app that remembers all of your friend’s birthdays. SQL is the most popular language you would use to talk to that app. English: “Hey app. When is my husband’s birthday?”. SQL:

```
‘‘SELECT * FROM birthdays WHERE person = ‘husband’; .
```

3. **Edgar Codd**. Ted Codd "was a British computer scientist who, while working for IBM, invented the relational model for database management" writes Wikipedia. Codd was a pilot in the RAF Coastal Command during the Second World War. Codd grew up in Dorset and coined the term "Online analytical processing" (OLAP). The **relational model** is an approach to managing data using a structure and language consistent with first-order predicate logic. Codd was a mathematician. He used the term "relation" in its mathematical sense of finitary relation. It is too technical for us to decipher here, but I still think it is important to know how Codd defined a relation. He wrote:

   > "Given sets $D_{1,}D_{2,}...D_n$ (not necessarily distinct), R is a relation on these sets if it is a set of elements of the form $d_{1,}d_{2,}...d_n$ where $d_{j,}\epsilon d_{2,}...D_j$ for each j=1,2,...n. [Codd 1971]

   Codd talked of *attributes*, which are essentially columns, and *tuples*, which are essentially rows. *If a database uses the relational model, we call it a relational database.*

4. **Redis** Description.

5. **Amazon RDS**. RDS stands for Relational Database Service. Amazon launched their RDS product in 2009, the year the VPC service was released and Obama became US President. "Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks" writes The User Guide.

6. **DynamoDB**. "Amazon DynamoDB is a serverless, NoSQL, fully managed database with single-digit millisecond performance at any scale" writes the User Guide. It was launched in 2012. "In DynamoDB, tables, items, and attributes are the core components that you work with. A table is a collection of *items*, and each item is a collection of *attributes*. DynamoDB uses primary keys to uniquely identify each item in a table. You can use DynamoDB Streams to capture data modification events in DynamoDB tables" (User Guide 2025).

7. **Amazon Aurora**. "Aurora is an open-source-compatible relational database" says Anurag Gupta, who helped to design Aurora. Aurora is often described as a database *engine*; I don't think it's worth reading too much into this. Charles Babbage called his mechanical computer an "analytical engine" and Google have their "search engine". Aurora is described as a database engine because it provides the core software responsible for storing, processing, and retrieving data. "Aurora is a fully managed relational database engine that's compatible with MySQL and PostgreSQL" writes the User Guide.

Aurora's performance is often compared to MySQL, as if they are two separate things, so it is important to note that you can run MySQL on Amazon Aurora. Amazon Aurora is a way of using a database on the cloud: the architects of Aurora describe how they separate the computers that deal with the database queries from the underlying storage and ensure that the storage volumes holding the Aurora data are replicated across three AZs.

8. **Caching** "In computing, a cache is a hardware or software component that stores data so that future requests for that data can be served faster" (Wikipedia).

9. **Memcached** It is pronounced mem-cash-dee. It has three syllables. It was released in 2003 by Brad Fitzpatrick for his website LiveJournal. (In this scene from *The Social Network* the actor Jesse Eisenberg types a post on Live Journal.) Memcached, three syllables, is a cache. But it is a cache that lives inside the memory of a computer, not its storage. "It is often used to speed up dynamic database-driven websites by caching data and objects in RAM to reduces the number of times an external data source (such as a database or API) must be read" (Wikipedia).

   Brad Fitzpatrick was born in Iowa. This state is where the butterfly stroke was born. That's right. The University of Iowa is the birthplace of the butterfly stroke. It was invented by David Armbruster, head coach of the University of Iowa swimming team, in 1935. Well, Brad Fitzpatrick was born in Iowa as well.

   Brad Fitzpatrick developed Memcached (three syllables). The system uses a **client-server architecture**. The servers maintain a key-value associative array. The clients populate this array and query it by key. Keys are up to 250 bytes long and values can be at most 1 megabyte in size.

10. **Amazon Elasticache**. In 2011, an early year when AWS Direct Connect was launched (which provides physical cables for people to connect securely to AWS data centres), AWS announced Amazon ElastiCache. It's one word but with two uppercase letters. ElastiCache is an "in-memory cache for web applications running in the AWS cloud. The new service improves the performance of web applications by enabling customers to retrieve information from a fast, managed, in-memory caching system in the cloud, instead of relying on slower disk-based databases. Amazon ElastiCache is compliant with Memcached, a widely adopted memory object caching system, so code, applications, and tools that developers use today with their existing Memcached environments work seamlessly with the service, easing the migration process" writes the press release from 2011.

2 Warm Standby

1. **Database** Description
2. **SQL** Description
3. **Edgar Codd** Description.
4. **Term 4** Description.
5. **Amazon RDS** Description.
6. **DynamoDB** Description.
7. **Amazon Aurora** Description
8. **Caching** Description
9. **Memcached** Description
10. **Amazon ElastiCache** Description

3 Pilot Light

The pilot comes aboard ships in unfamiliar waters to sort out shit.

1. **Database** Description
2. **SQL** Description
3. **Edgar Codd** Description.
4. **Term 4** Description.
5. **Amazon RDS** Description.
6. **DynamoDB** Description.
7. **Amazon Aurora** Description
8. **Caching** Description
9. **Memcached** Description
10. **Amazon ElastiCache** Labib, Michael (2016). Amazon Elasticache Deep Dive. *Reinvent 2016* [Conference]. Available at: `https://www.youtube.com/watch?v=e9sN15a7utI`

    Labib begins by talking about Memcached. “Memcached has been around and available since 2003... It supports a data structure which is a string. You can support up to one megabyte in that value. It has no persistence.” Labib explains that the lack of persistence is okay, but for some situations it is not and this is why you might want Redis. Labib moves on to compare Memecached and Redis.

Suryadevara, Pratibha (2019). Supercharge your real-time apps with Amazon ElastiCache. Available at: `https://www.youtube.com/watch?v=vOGfpL5jfns`
Suradevara is the General Manager (GM) for ElastiCache in this presentation; the previous year she delivered a deep dive on Elastic Load Balancing with Will Rose from Netflix.

4 Backup

1. **Database** Description

   Date, Christopher (1975). An Introduction to Database Systems.

2. **SQL** Description

3. **Security Group (SG)** Description.

4. **Redis** Description.

5. **Amazon RDS**
   Ward, Scott (2016). Deep Dive on Amazon Relational Database Service. Reinvent 2016 [Conference]. Available at: `https://www.youtube.com/watch?v=pPLPzPYY5uU`

   Barr, Jeff (2013). Resource-Level Permissions for EC2 and RDS Resources. Available at: `https://aws.amazon.com/blogs/aws/resource-permissions-for-ec2-and-rd`

6. **DynamoDB** AWS (2012). Amazon DynamoDB - What's It All About? YouTube Channel: Amazon DynamoDB. Available at: `https://www.youtube.com/watch?v=nMhWJJACZSA`

   Green, Sam (2024). The AWS Knowledge Base. Available at: `https://www.youtube.com/playlist?list=PLn2pda68RIHWScIl8Ipl4aEkdKrHrrupK`

   This is a practical project I carried out, implementing a DynamoDB database behind a web page. The project took several hours and I documented it in several YouTube videos. Go to the video called *Creating a DynamoDB table*.

   Houlihan, Rick (2018). Advanced Design Patterns for DynamoDB. *Reinvent 2018* [Coonference]. Available at: `https://www.youtube.com/watch?v=HaEPXoXVf2k`

A framework for Amazon DynamoDB Transactions. Available at: `https://aws.amazon.com/blogs/database/a-framework-for-amazon-dynamodb-transactions/`

Hunter, Jason and Vivek Natarajan (2023). Scaling DynamoDB: How partitions, hot keys, and split for heat impact performance. *AWS Database Blog*. Available at: `https://aws.amazon.com/blogs/database/part-1-scaling-dynamodb-how-part`

AWS (2012). AWSLIVE: Introducing Amazon DynamoDB, a fully managed NoSQL database service. Available at: `https://www.youtube.com/watch?v=wjJ3Dkl6VS4`

Lang, Dave (2013). Production NoSQL in an Hour: Introduction to Amazon DynamoDB. YouTube Channel: Amazon Web Services. Available at: `https://www.youtube.com/watch?v=eyUalcd_Ks0`

AWS (2013). Getting Started with Fine-Grained Access Control for DynamoDB. Available at: `https://www.youtube.com/watch?v=uAUYphLWL5w`

Matthew, Sajee (2016). DynamoDB Design Patterns and Best Practices. YouTube Channel: Amazon Web Services. Available at: `https://www.youtube.com/watch?v=iKrKfY2IcZQ`

Beabetterdev, Daniel (2019). Available at: `https://www.youtube.com/watch?v=ihMOlb8EZKE`

7. **Amazon Aurora** Description

   Gupta, Anurag (2016). Deep Dive on Amazon Aurora. Available at: `https://www.youtube.com/watch?v=duf5uUsW3TM`

   Verbitski, Alexandre, and Anurag Gupta et al (2017). Design considerations for high throughput cloud-native relational databases. SIGMOD'17. Available at: `https://www.amazon.science/publications/amazon-aurora-design-considerations-`

   Green, Sam (2025). Amazon Aurora Paper Readthrough. Available at: `https://www.youtube.com/playlist?list=PLn2pda68RIHXnAzEdEa7pKMHyPguwwFWW`

8. **Caching** Description

   Porter, Mark (2016). Introducing PostgreSQL Compatibility for Amazon Aurora. Reinvent 2016 [Conference]. Available at: `https://www.youtube.com/watch?v=ztmtJJTC8_Y`

9. **Memcached**

   Zuckerberg, Mark (2013). Facebook and Memcached. Available at: `https:`

//www.youtube.com/watch?v=UH7wkvcf0ys

10. **Amazon Elasticache**

    Amazon ElastiCache User Guide. Available at: https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

    Caching Strategies for Memcached. Available at: https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html

    Labib, Michael (2016). Amazon Elasticache Deep Dive. *Reinvent 2016* [Conference]. Available at: https://www.youtube.com/watch?v=e9sN15a7utI

    Suryadevara, Pratibha (2019). Supercharge your real-time apps with Amazon ElastiCache. Available at: https://www.youtube.com/watch?v=v0GfpL5jfns

    Database Caching Strategies Using Redis [Whitepaper]. Available at: https://docs.aws.amazon.com/whitepapers/latest/database-caching-strategies-using-redis/caching-patterns.html