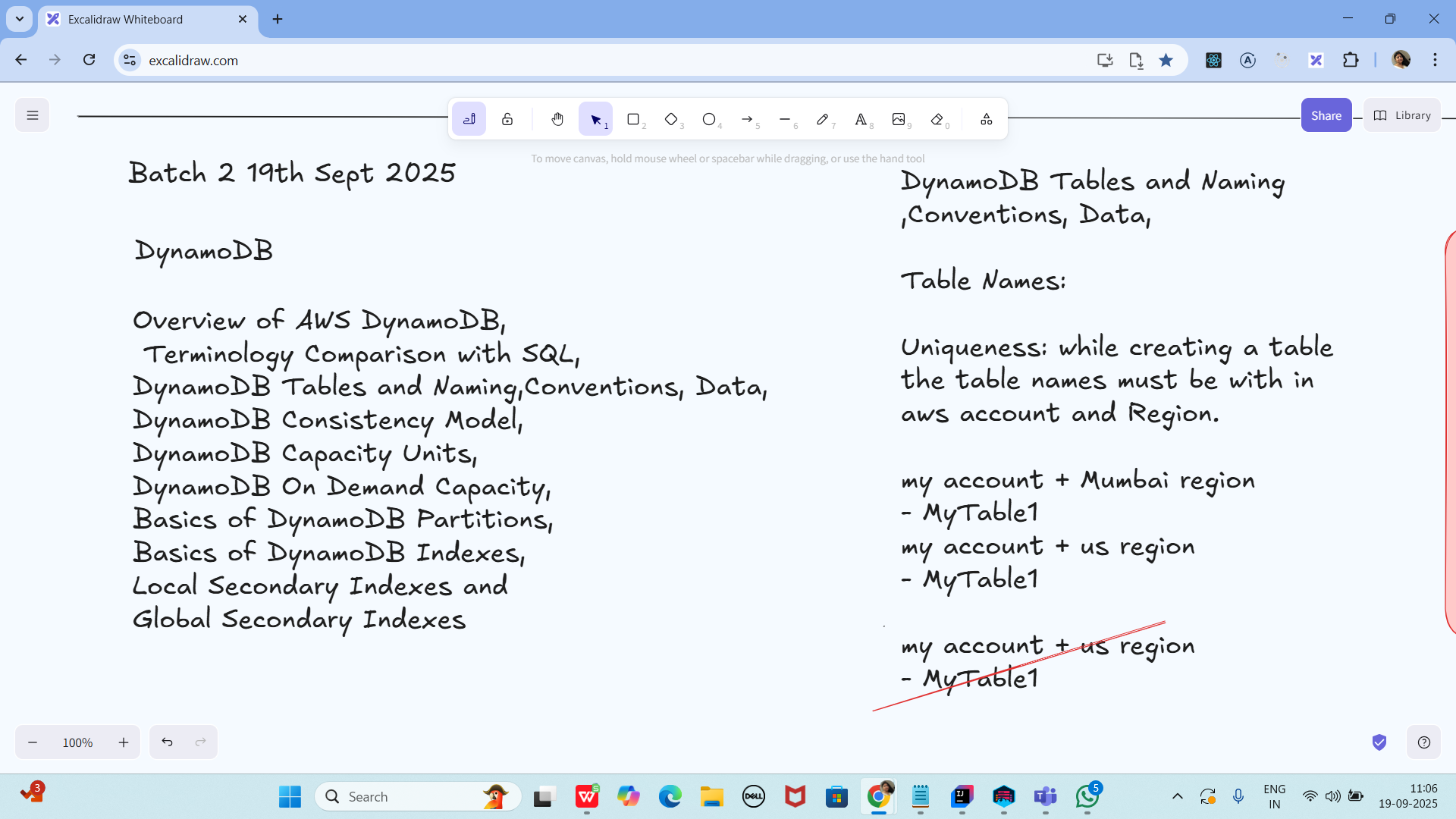
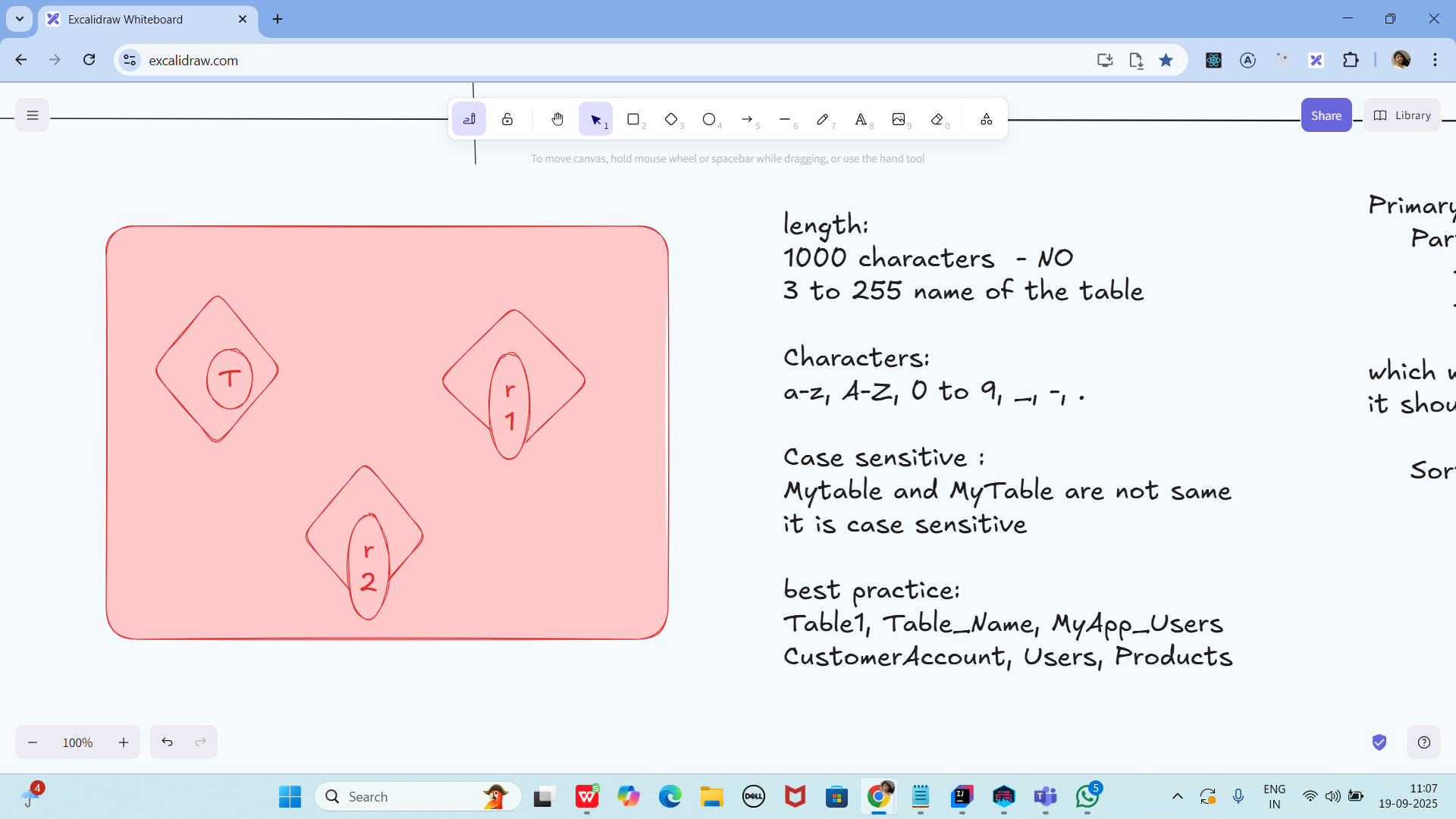
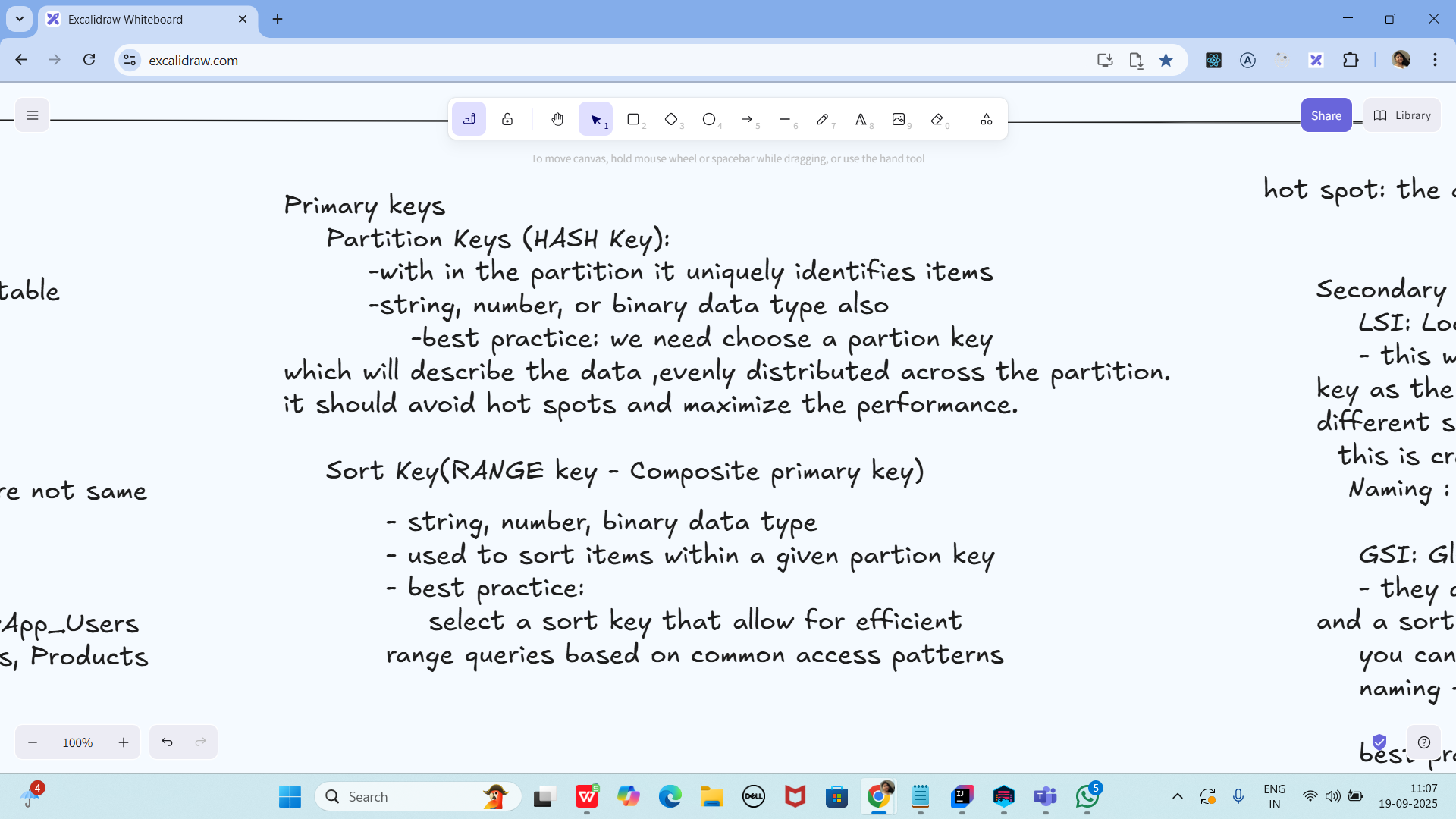
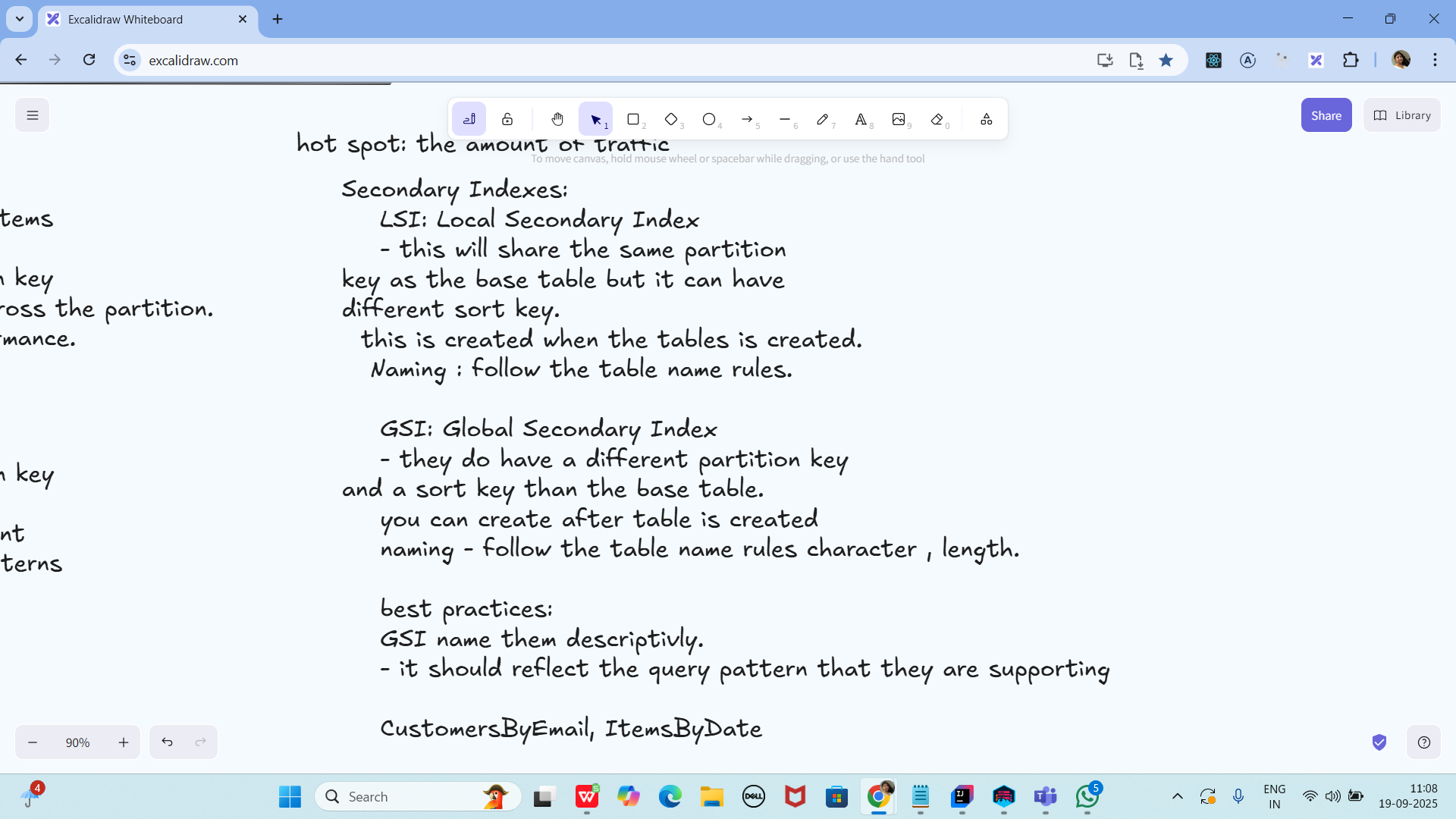
Day 34 19th Sept 2025

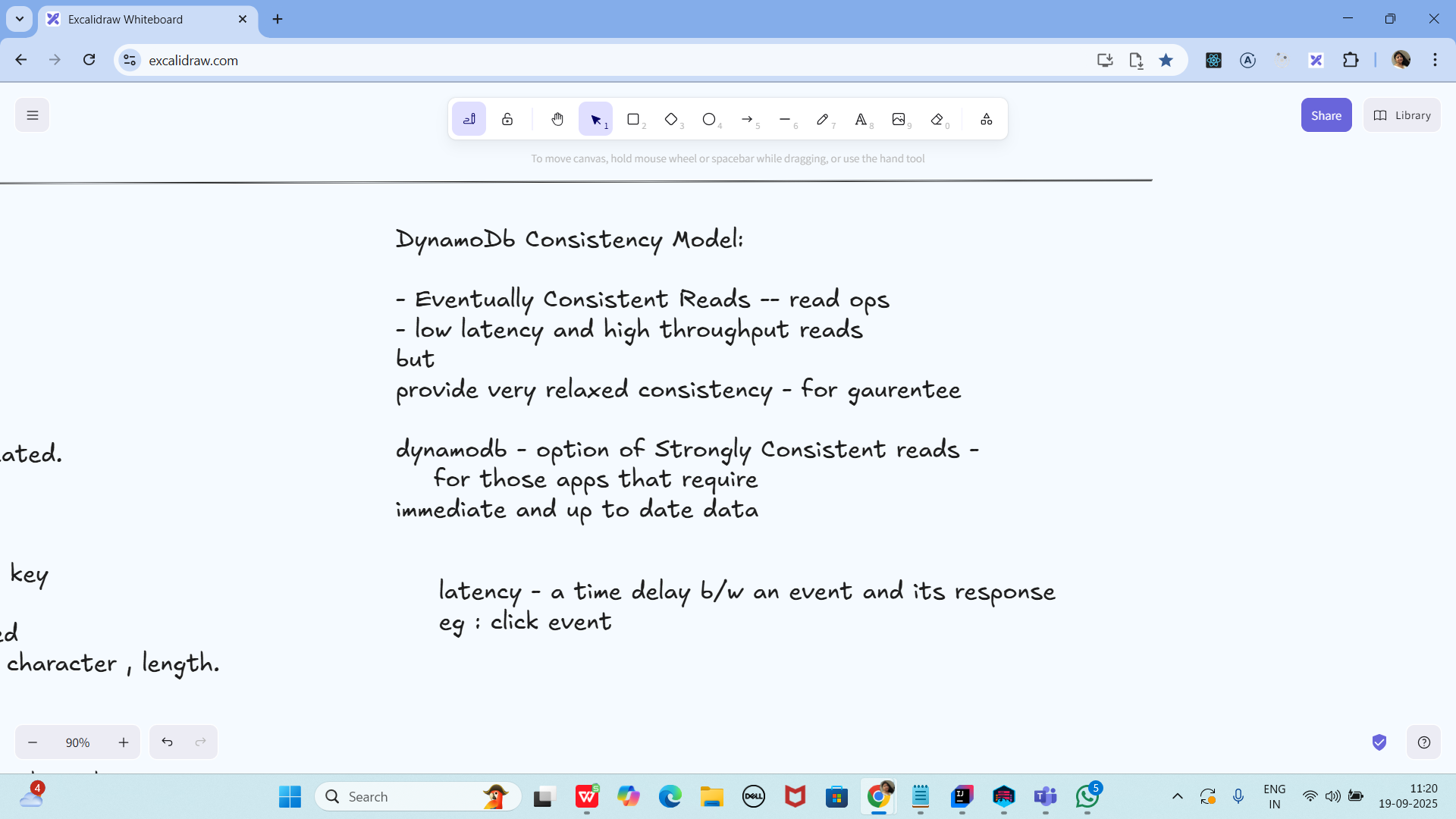
|  |  |  |
| --- | --- | --- |
| **NoSQLs & Dynamo**  **Dev Considerations - Modelling, Indexes, Syntax**  **Architecture - Data Structures, Queries**  **Implementing DB solution via Dynamo/NoSQL** | Dynamo DB | Overview of AWS DynamoDB, Terminology Comparison with SQL, DynamoDB Tables and Naming,Conventions, Data, DynamoDB Consistency Model, DynamoDB Capacity Units, DynamoDB On Demand Capacity, Basics of DynamoDB Partitions, Basics of DynamoDB Indexes, Local Secondary Indexes and Global Secondary Indexes |
| Working with DynamoDB using AWS Console3 | Interacting with DynamoDB -Table-level Operations with AWS Console, Item-level Operations with AWS Console, Additional Features in DynamoDB Console |
| Working with DynamoDB using AWS CL | Working with the AWS CLI, Table level Operations with AWS CLI, Write Operations - Item level Operations with AWS CLI, Read Operations - Item level Operations with AWS CLI |
| Working with DynamoDB using AWS SDK | Working with DynamoDB using AWS SDK - Introduction, Table-level Operations with AWS SDK, Write Operations - Item Level Operations with AWS SDK, Conditional Writes - Item Level Operations with AWS SDK, Atomic Counters - Item Level Operations with AWS SDK, Read Operations - Item Level Operations with AWS SDK, Paginated Reads - Item Level Operations with AWS SDK |
| DynamoDB Data Modeling & Best Practices | DynamoDB Architecture, DynamoDB Partitions in Depth, DynamoDB Efficient Key Design, Hot Keys or Hot Partitions, DynamoDB Design Patterns, Multi-value Sorts and Filters, DynamoDB Limits - Error Handling in DynamoDB, Ways to Lower DynamoDB Costs |

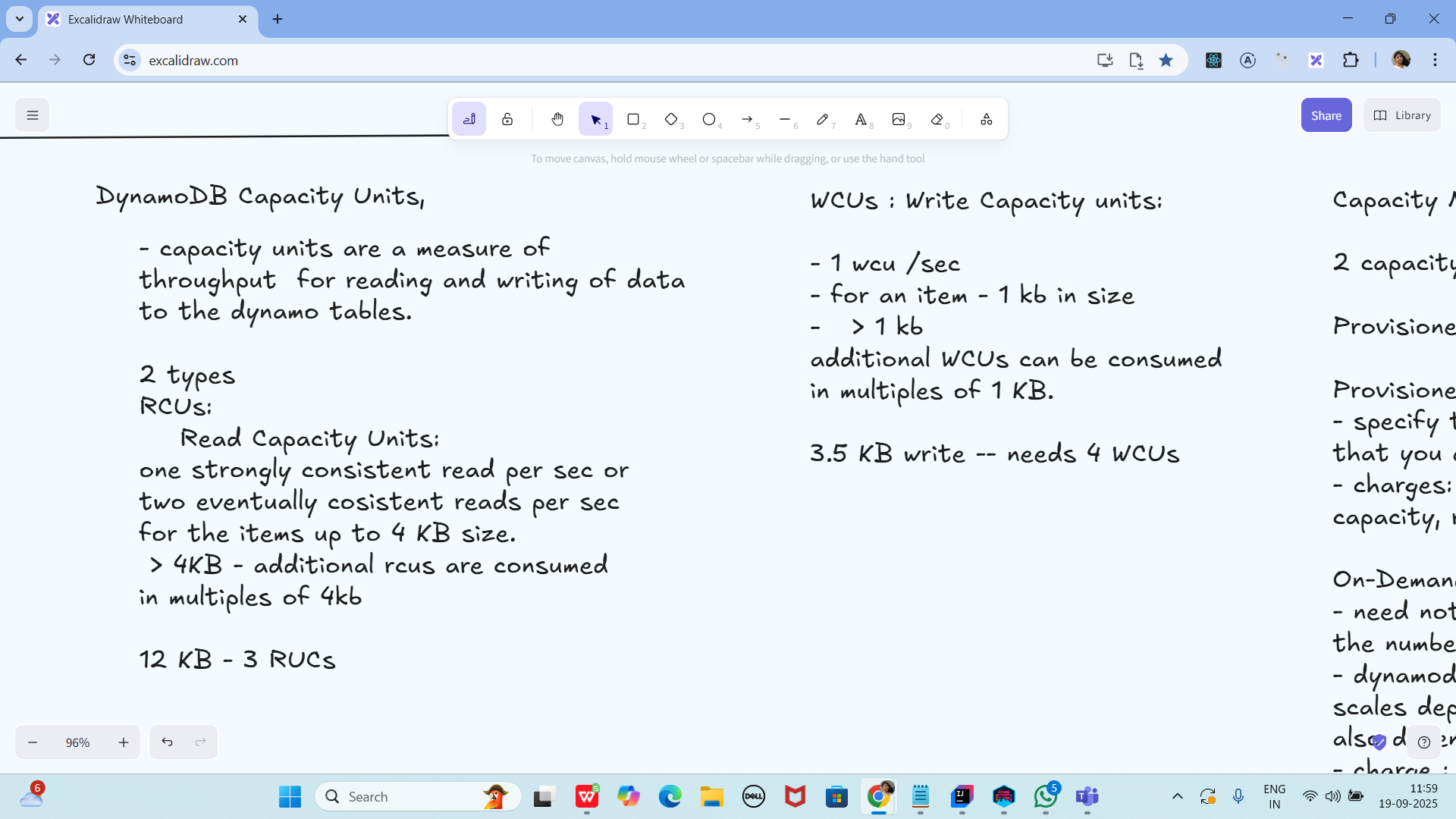


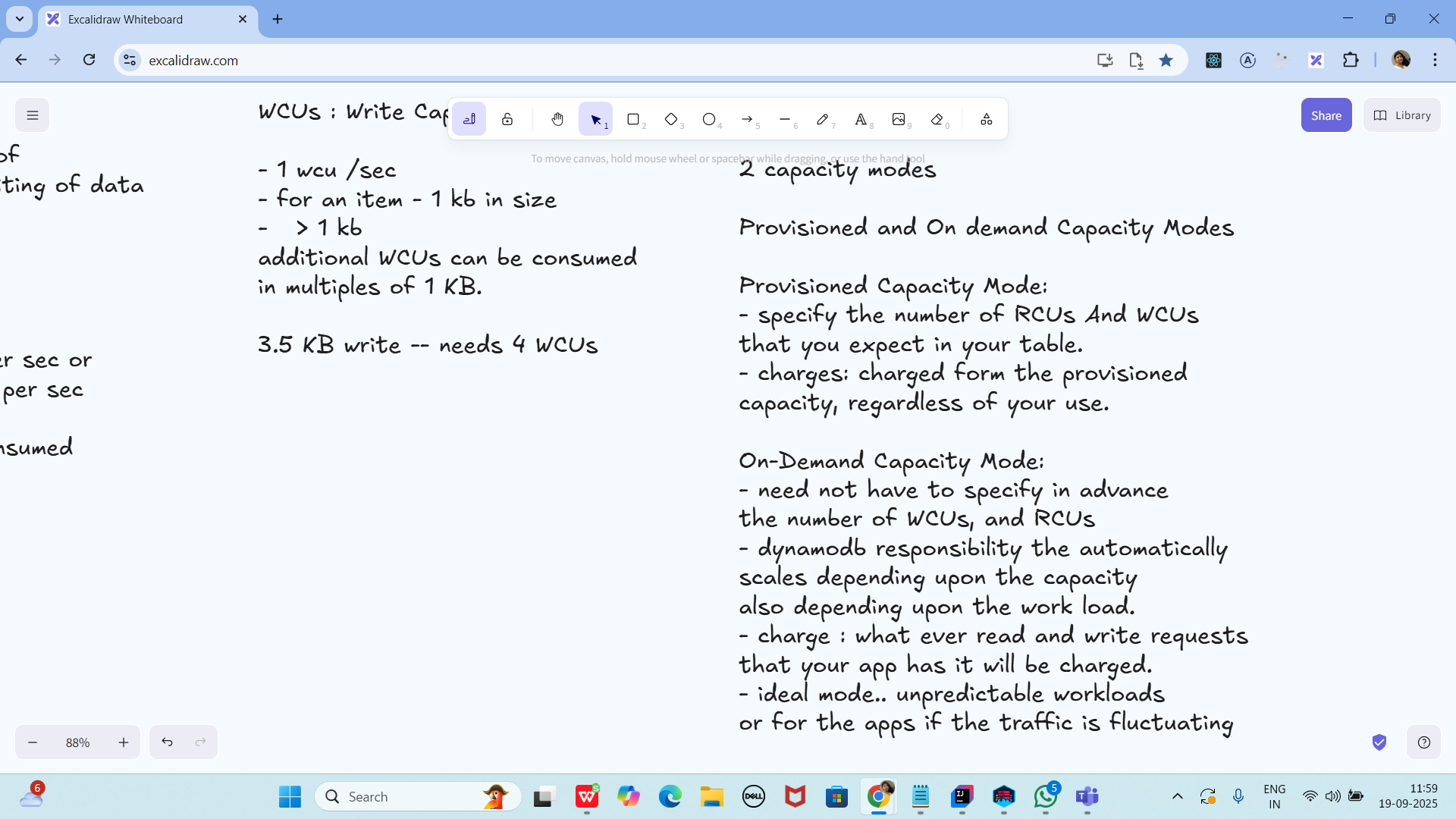


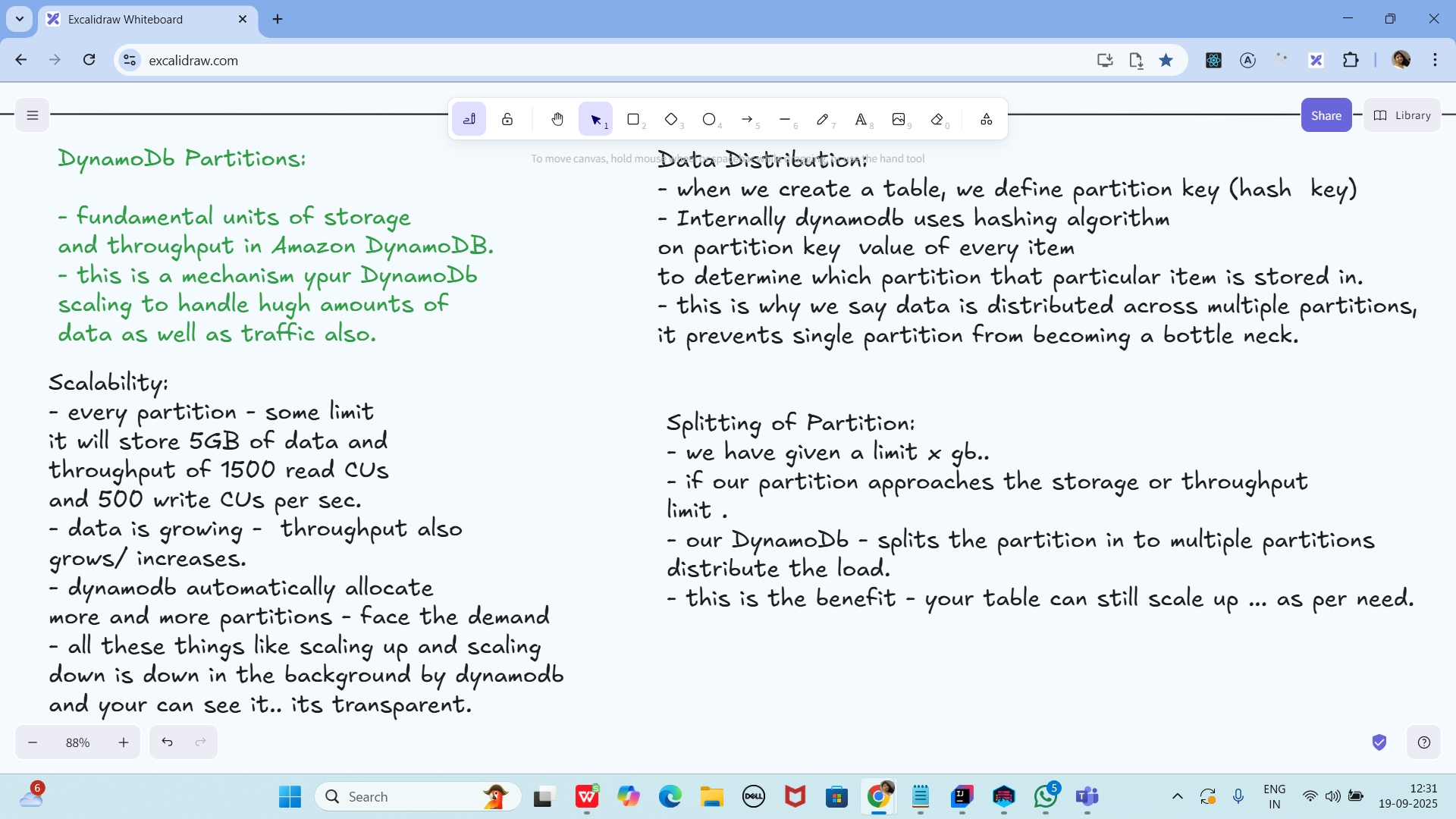


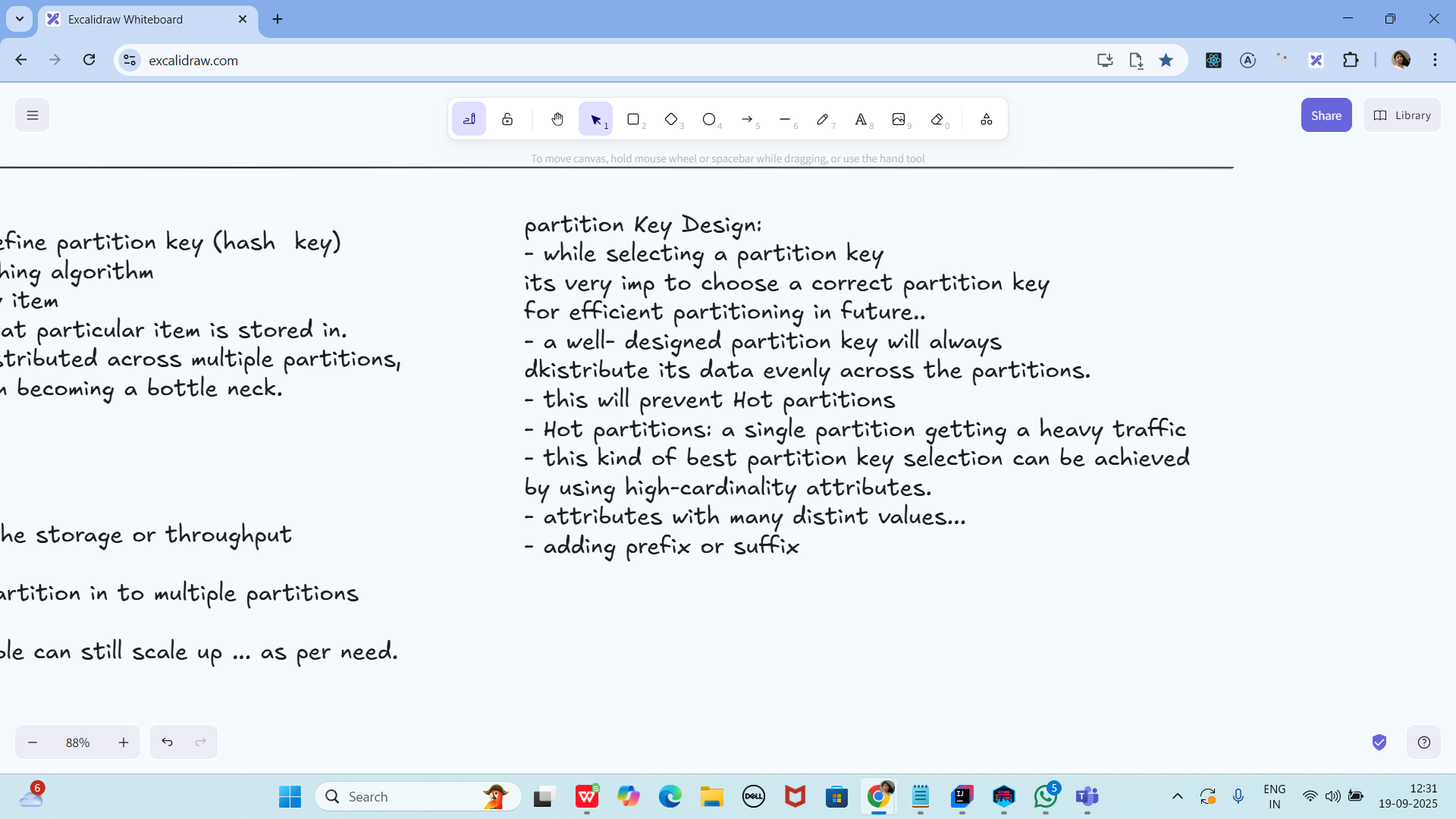


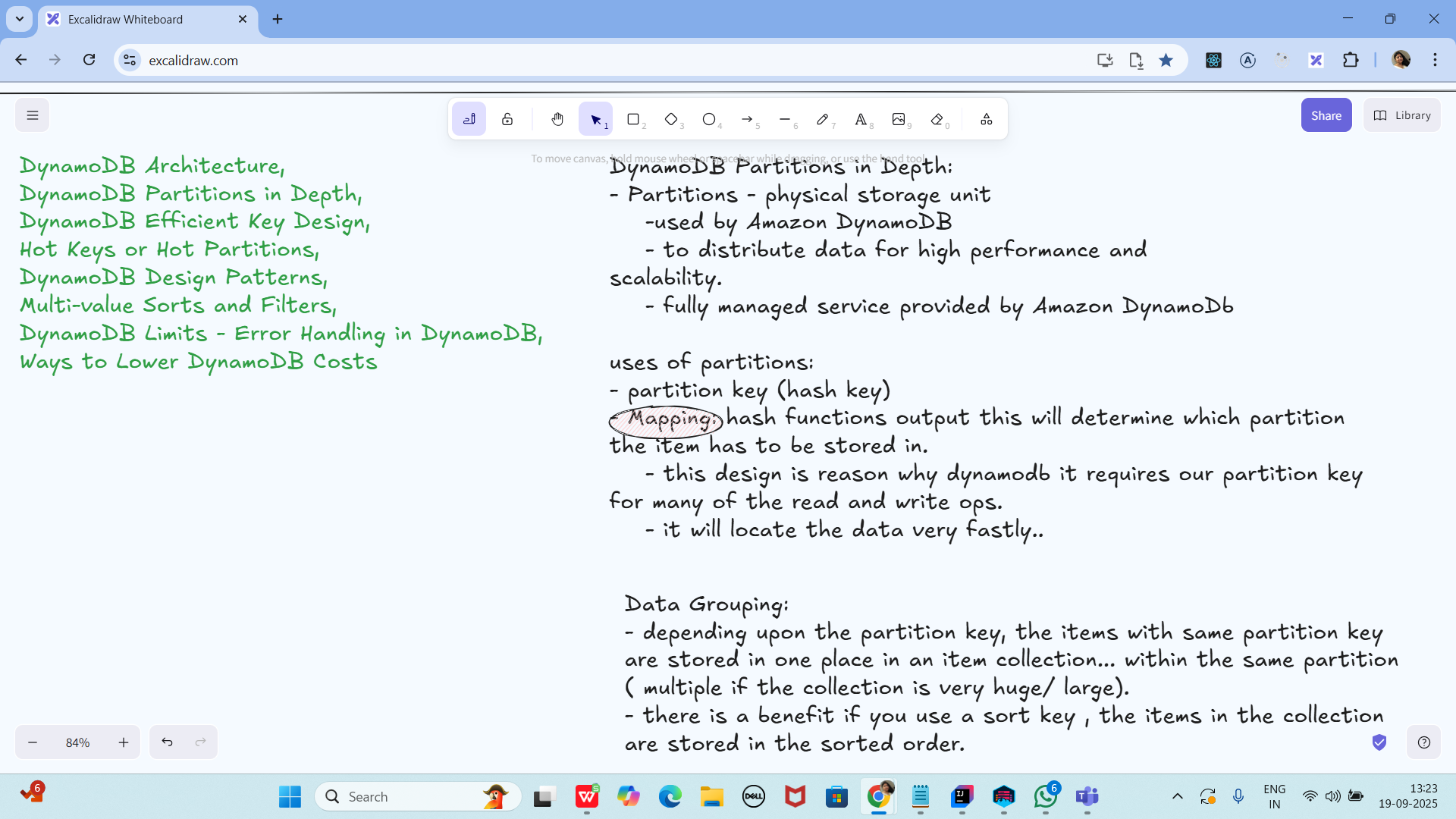


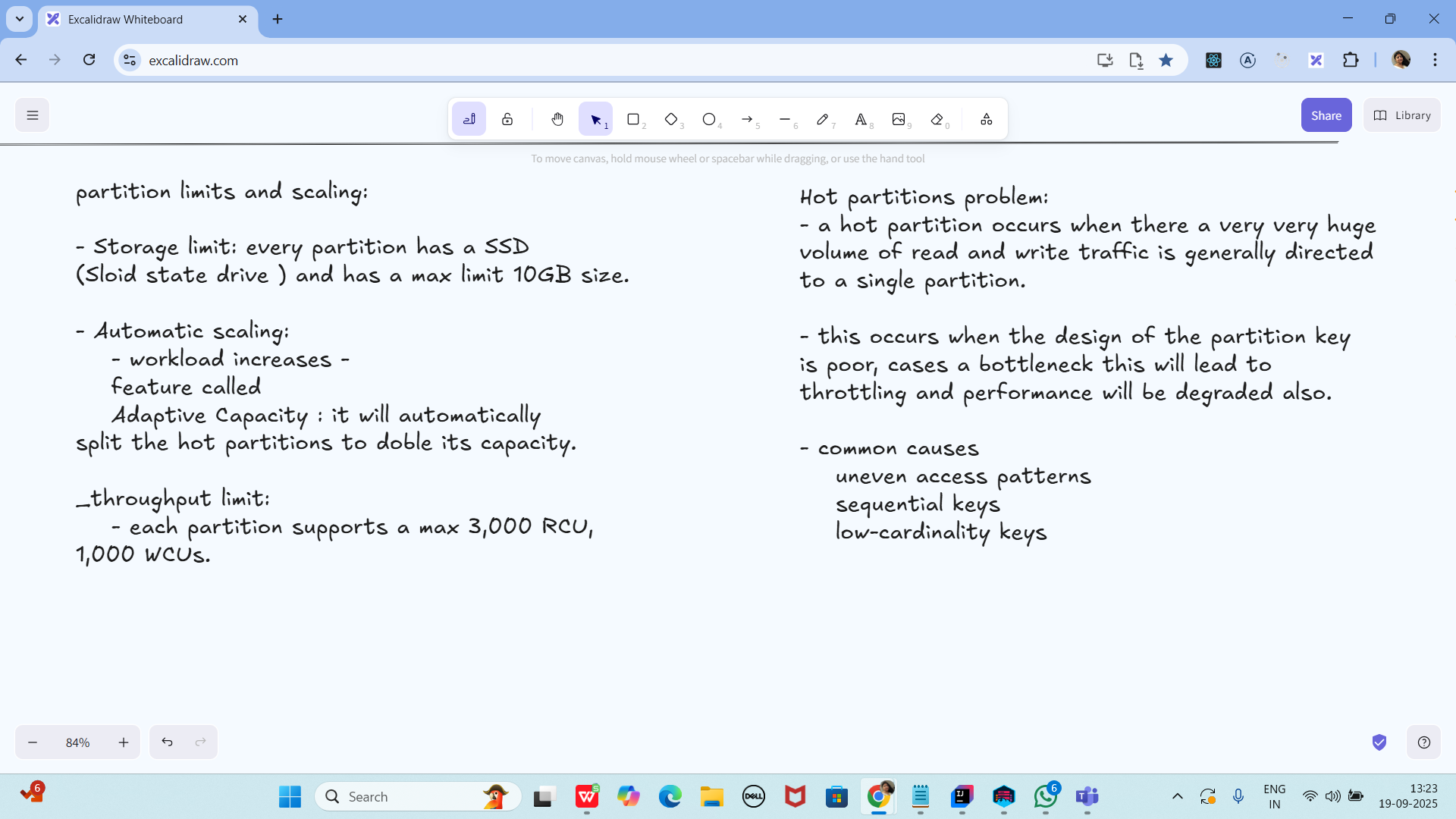


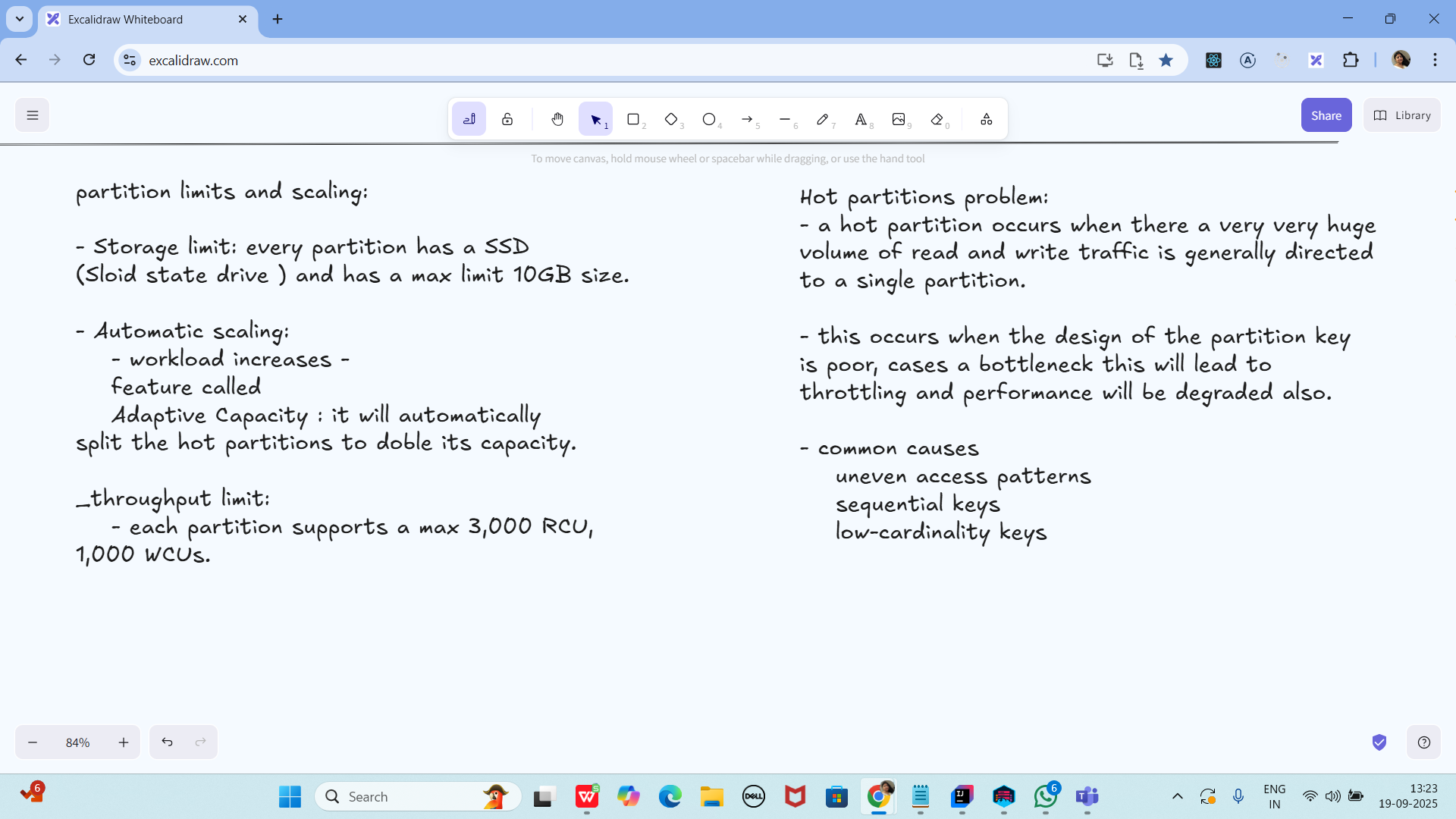




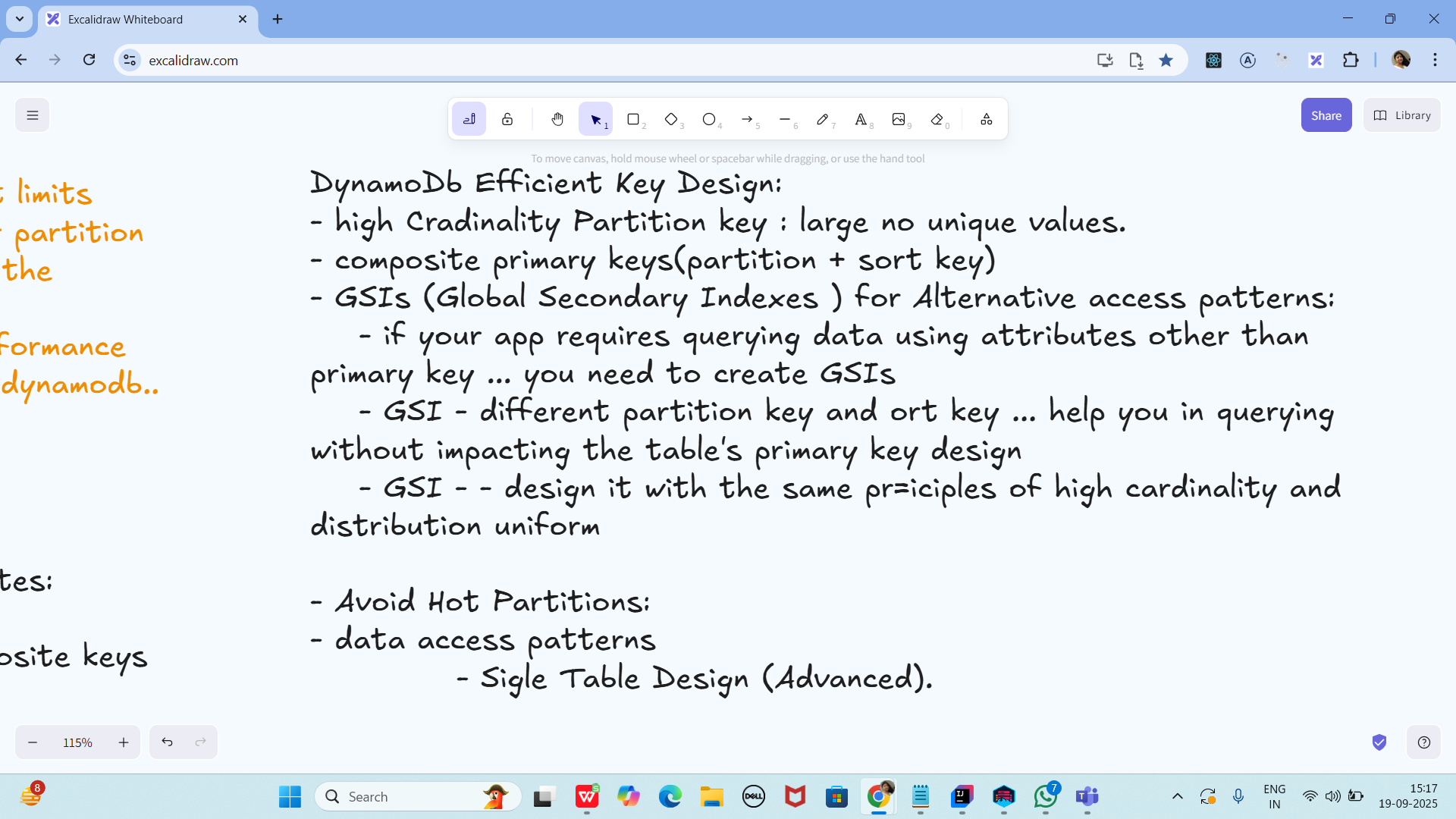


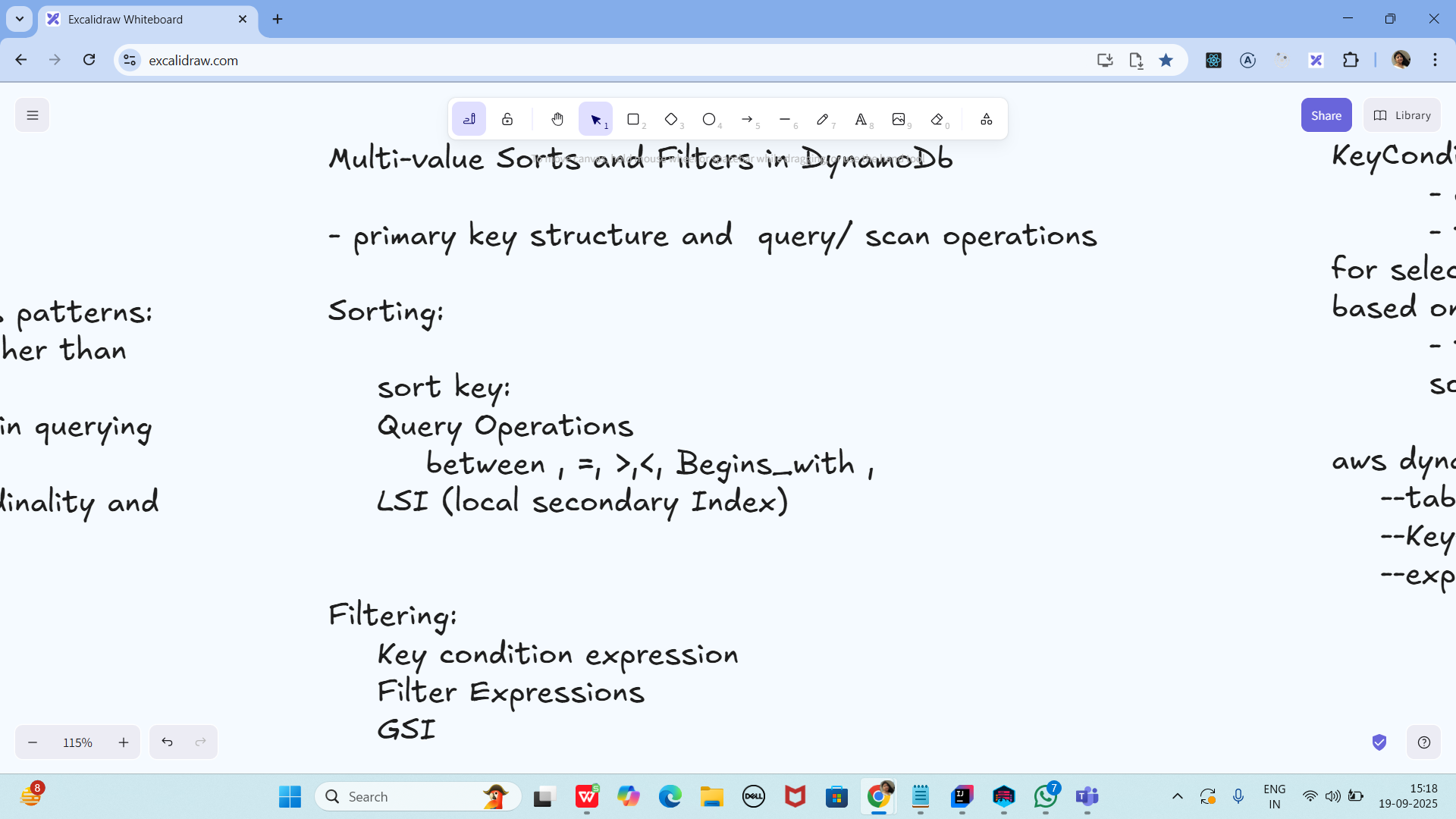


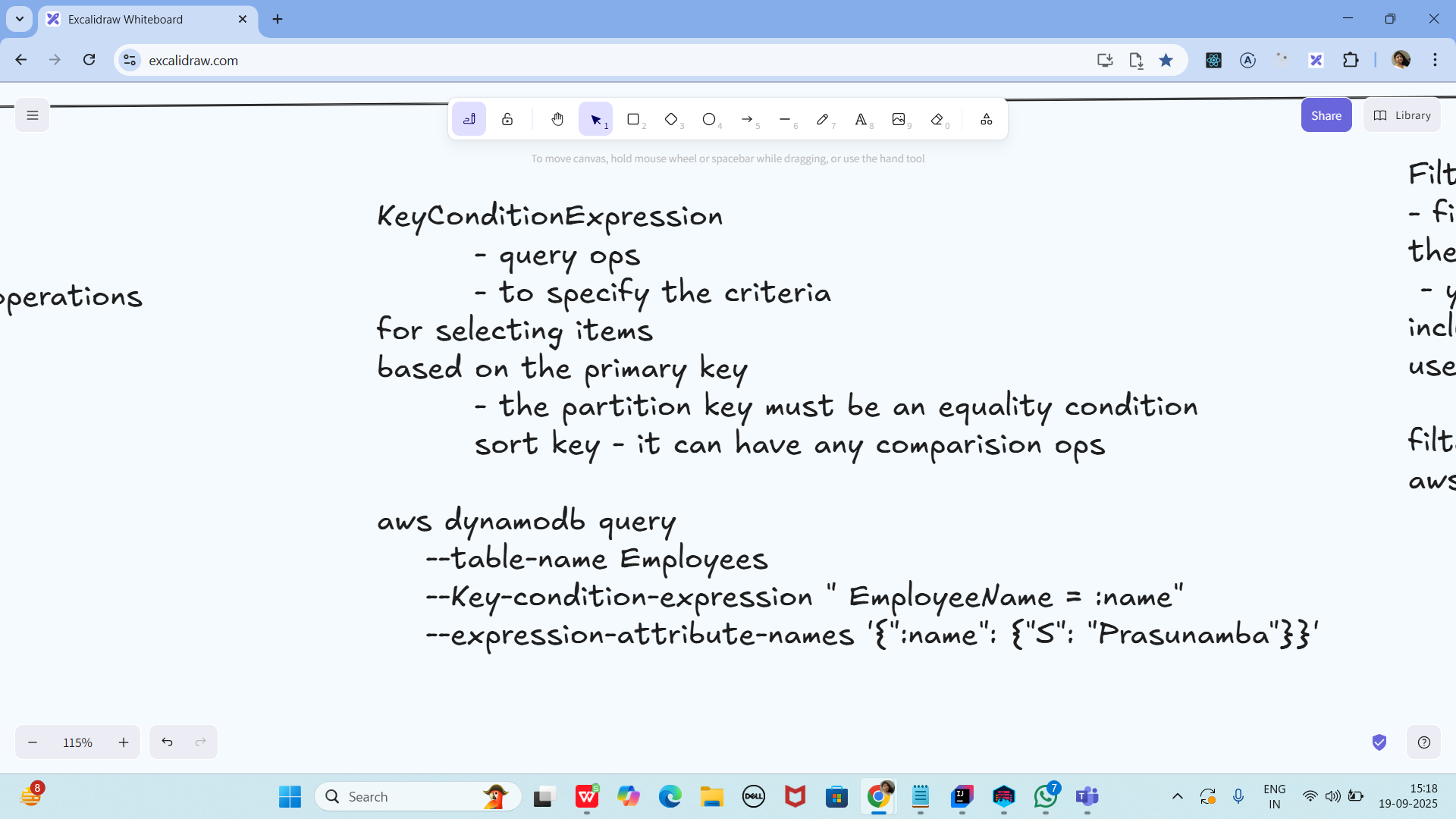


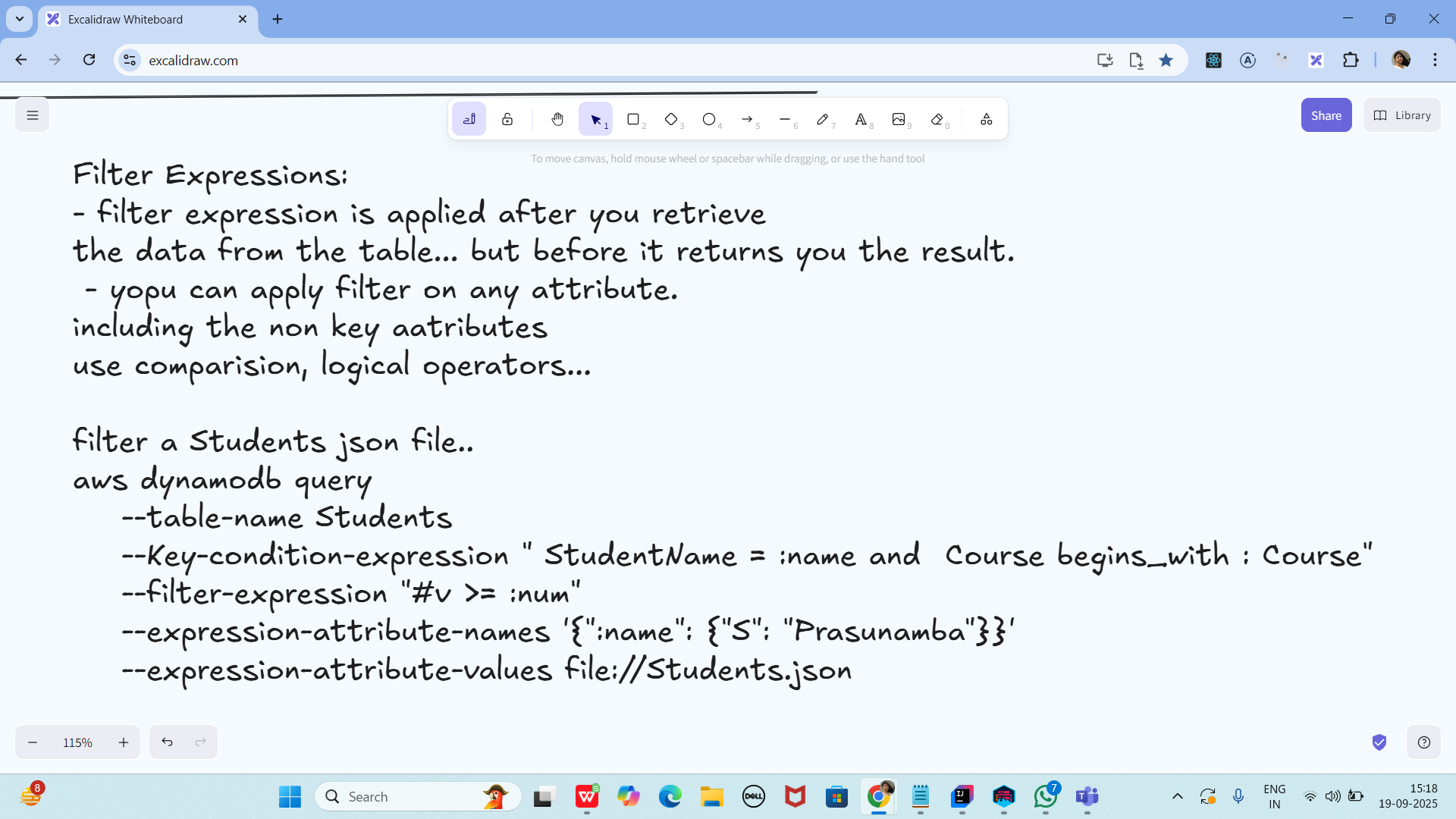












Task 01:

In DynamoDB, which factor triggers the creation of new partitions automatically?

1. Every time a user creates a new GSI on the table.

2. When the storage in an existing partition exceeds 10 GB or throughput requirements surpass limits.

3. When global replication is enabled via global tables.

4. Upon inserting items larger than 400 KB individually.

Task 01:

In DynamoDB, which factor triggers the creation of new partitions automatically?

1. Every time a user creates a new GSI on the table.

2. When the storage in an existing partition exceeds 10 GB or throughput requirements surpass limits.

3. When global replication is enabled via global tables.

4. Upon inserting items larger than 400 KB individually.

Task 02:

Which statement is true about write operations using AWS SDK in DynamoDB?

1. Write operations always replace the entire item unless conditionally filtered.

2. The SDK enforces write isolation by automatically disabling concurrent access.

3. You can use UpdateItem with expressions to modify specific attributes without affecting the rest.

4. All write operations must be wrapped in transactions when using SDK.

Task 03:

In the AWS SDK, which method would you use to increase an existing numeric attribute without retrieving its current value?

1. Use PutItem with full item replacement and a computed value.

2. Use UpdateItem with ADD operation on the attribute.

3. Use GetItem followed by UpdateItem with arithmetic expression.

4. Use IncrementItem method under conditional write configuration.

Task 04:

When using the AWS CLI to delete a DynamoDB table, what precaution must be taken?

1. You must disable all indexes before issuing the delete command.

2. You should manually drain the capacity units to avoid service penalties.

3. Deleting a table is immediate, but recovery is not possible unless point-in-time recovery is enabled.

4. CLI deletes are queued operations and may take up to 48 hours to reflect.

Task 05:

In the AWS Console, what happens if you try to create a table with a duplicate name in the same region?

1. The console will automatically append a timestamp to make the name unique.

2. Table creation will fail with a validation error indicating name conflict.

3. It silently overwrites the existing table with the same name.

4. A new versioned table is created with identical name but different ARN.

Task 06:

What is a key characteristic of DynamoDB partitions that influences table performance?

1. Each partition can contain up to 100 GB of data and supports up to 3,000 read and 1,000 write capacity units.

2. Partitions store only the indexes and are mapped to base tables by a global hash.

3. Partition creation must be triggered manually based on table growth metrics.

4. All partitions share a common throughput quota, evenly distributed regardless of key usage.

Task 07:

In DynamoDB data modeling, what does the multi-value sort key pattern allow you to do?

1. Use multiple sort keys to create unique records across several partition keys.

2. Emulate a relational schema by allowing sort keys to represent hierarchical relationships like time ranges or event types.

3. Store list-type values directly in the sort key for rapid filtering.

4. Enable join-like behavior across different tables by linking sort keys dynamically.

Task 08:

What is a recommended method to lower DynamoDB costs while maintaining access performance?

1. Use multiple GSIs with uniform read patterns to distribute throughput.

2. Store large binary objects directly in DynamoDB to avoid data transfer costs.

3. Optimize access patterns using compound keys and avoid unnecessary indexes or scans.

4. Enable strong consistency on all reads to reduce retries and associated charges.

Task 09:

How can hot partitions be avoided in a high-volume DynamoDB design?

1. Use a fixed partition key for all entries and rely on sort key randomness.

2. Adopt a key sharding strategy where the partition key is prefixed or suffixed with a hashed or random number.

3. Implement partition auto-balancing using Amazon Athena with DynamoDB exports.

4. Enable parallel scans with partition-level throughput reservations.

Task 10:

How do Local Secondary Indexes (LSIs) affect storage limits per partition key in DynamoDB?

1. LSIs have no impact on size limits because they store only index pointers.

2. LSIs share the same item size limit (400 KB) as the base table but have no cumulative partition size limit.

3. The total size of all items with the same partition key (including all versions across LSIs) must not exceed 10 GB.

4. Each LSI adds an extra 10 GB limit per partition key independent of the base table

Task 11:

Which AWS CLI command correctly updates an item's attribute score by adding 5 in a table named Players?

1.  aws dynamodb update-item --table-name Players --key '{"PlayerId":{"S":"101"}}' --update-expression "ADD score :val" --expression-attribute-values '{":val":{"N":"5"}}'

2. aws dynamodb modify-item --table-name Players --primary-key PlayerId=101 --add score 5

3. aws dynamodb increment-item --table-name Players --key PlayerId=101 --value 5

4. aws dynamodb update-item --table-name Players --key '{"PlayerId":{"S":"101"}}' --update-expression "SET score = score + :val" --expression-attribute-values '{":val":{"N":"5"}}'