**Question 6 – Cypher querying**

1. *Explain the clauses of the Cypher language and show how it is used for querying.*

**Clauses of the Cypher Language**

MATCH/OPTIONAL MATCH

* **MATCH** finds all paths that satisfy a pattern
* **OPTIONAL MATCH** behaves like a left join, returning **null** when no match is found

WHERE

* Filters results on property conditions or pattern existence (**EXISTS/NOT EXISTS**)

RETURN

* Specifies which variables or expressions to return (like SQL’s **SELECT**)
* Implicit grouping: **RETURN c, COUNT(y)** groups by **c**
* New aggregations: **COLLECT**, etc.

WITH

* Passes intermediate results down the pipeline, enables aliases, and acts as a place to apply further **WHERE, ORDER BY,** or **LIMIT**

UNWIND

* “Explodes” a list into individual rows, useful for processing collections

ORDER BY, SKIP, LIMIT

* Controls result ordering and pagination

SET/DELETE/DETACH DELETE

* **SET** updates properties or labels
* **DELETE** removes relationships or nodes (fails if nodes still have relationships)
* **DETACH DELETE** removes nodes plus all attached relationships

1. *Compare selected queries from your 3 assignments.*

**Comparing Selected Queries Across Assignments**

Explain code

the aggregation pipeline replaces JOIN + GROUP BY in SQL

with a sequence of stages ($lookup, $unwind, $group, etc.),

in GraphQL/Neo4j you often rely on automatic resolvers or custom @cypher directives to achieve the same end.