ST207 DATABASES (AT2025) – ASSIGNMENT 2 – ASSESSMENT CRITERIA

Question	Fail	Pass	Merit	Distinction
1A	No database. Missing or incorrect number of documents. Missing or incorrect number of attributes (key/value pairs). Missing or poorly formed relationships. Presence of errors that compromise model understanding and target queries. Poor/missing explanations in the report.	Database successfully created. Correct number of documents. Minor errors related to the structure of document collection or attribute definition (key/value pairs). Minor errors related to relationship definitions or other aspects not mapped from the given context (missing specification) with moderated influence in the resulting database and target queries. Satisfactory explanation in the report, with points to be better detailed.	Database successfully created. Correct number of documents and attributes, with no errors. Correct mapping of relationships, leading to a consistent document collection. Minor errors impacting the target queries. Consistent and clear explanation in the report, with minor points to clarify.	Database successfully created. Correct number of documents and attributes, with no errors. Correct mapping of relationships, with excellent use of embedded vs referenced relationships, leading to a consistent document collection. Absence of any errors that can impact the final database and target queries. Consistent and clear explanation in the report, with no points to clarify.
1B	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report). Faulty or partial solution not addressing all the required points. Faulty results (retrieval of partial information to fulfil the query). Poor/missing explanation of the query in the report. No code submitted.	Good solution based on standard NoSQL constructs, with minor points for improvement and/or errors not affecting the overall results. Good results but with room for improvement (for instance, presence of duplicated documents per supplier, missing aggregated information per supplier or other aspects influencing the results). Satisfactory explanation in the report, with points to be better detailed. Legible and documented code, with points for improvement.	Good solution based on complex NoSQL constructs (when necessary), with minor points for improvement and no errors. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Some redundancy of code is tolerated. Good results with no need for improvement. Consistent and clear explanation in the report, with minor points to clarify. Legible and well-documented code, with no points for improvement.	Excellent solution based on complex NoSQL constructs (when necessary), with no points for improvement and no errors. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Efficient queries (minimal redundancy, optimal pipelines). Proper handling of edge cases (null values, empty results). Excellent results, covering the entire input dataset. Consistent and clear explanation in the report, with no points to clarify. Legible and well-documented code, with no points for improvement.
1C	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report). Faulty or partial solution not addressing all the required points. Faulty results (retrieval of partial information to fulfil the query). Poor/missing explanation of the query in the report. No code submitted.	Good solution based on standard NoSQL constructs, with minor points for improvement and/or errors not affecting the overall results. Good results but with room for improvement (for instance, presence of duplicated documents per supplier, missing aggregated information per supplier or other aspects influencing the	Good solution based on complex NoSQL constructs (when necessary), with minor points for improvement and no errors. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Some redundancy of code is tolerated. Good results with	Excellent solution based on complex NoSQL constructs (when necessary), with no points for improvement and no errors. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Efficient queries (minimal redundancy, optimal pipelines). Proper handling of edge cases (null

		results). Satisfactory explanation in the report, with points to be better detailed. Legible and documented code, with points for improvement.	no need for improvement. Consistent and clear explanation in the report, with minor points to clarify. Legible and well-documented code, with no points for improvement.	values, empty results). Excellent results, covering the entire input dataset. Consistent and clear explanation in the report, with no points to clarify. Legible and well-documented code, with no points for improvement.
1D	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report). Faulty or partial solution not addressing all the required points, including lack of materialised view. Faulty results (retrieval of partial information to fulfil the query). Poor/missing explanation of the query in the report. No code submitted.	Good solution based on standard NoSQL constructs, with minor points for improvement and/or errors not affecting the overall results. Materialised view created. Good results but with room for improvement (for instance, presence of duplicated documents per supplier, missing aggregated information per supplier or other aspects influencing the results). Satisfactory explanation in the report, with points to be better detailed. Legible and documented code, with points for improvement.	Good solution based on complex NoSQL constructs (when necessary), with minor points for improvement and no errors. Materialised view created, with no extensions. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Some redundancy of code is tolerated. Good results with no need for improvement. Consistent and clear explanation in the report, with minor points to clarify. Legible and well-documented code, with no points for improvement.	Excellent solution based on complex NoSQL constructs (when necessary), with no points for improvement and no errors. Materialised view created, with any extensions to improve understanding and summarisation of existing data. Good use of pipeline stages, document lookup or other functions necessary to retrieve the required data. Efficient queries (minimal redundancy, optimal pipelines). Proper handling of edge cases (null values, empty results). Excellent results, covering the entire input dataset. Consistent and clear explanation in the report, with no points to clarify. Legible and well-documented code, with no points for improvement.
2A.1	No property graph data model. Missing or incorrect number of nodes. Missing or incorrect number of properties (key/value pairs). Missing or poorly formed relationships. Presence of errors that compromise model understanding and target queries. Missing figure/diagram of the resulting graph model. Poor/missing explanations in the report.	Correct property graph data model. Correct number of nodes. Minor errors related to the structure of graph model or properties definition (key/value pairs). Minor errors related to relationship definitions or other aspects not mapped from the given context (missing specification) with moderated influence in the resulting database and target queries. Figure/diagram of the resulting graph model provided. Satisfactory explanation in the report, with points to be better detailed.	Correct property graph data model. Correct number of nodes and properties, with no errors. Correct mapping of relationships, leading to a consistent graph database. Minor errors impacting the target queries. Figure/diagram of the resulting graph model provided. Consistent and clear explanation in the report, with minor points to clarify.	Correct number of nodes and properties, with no errors. Excellent mapping of relationships, leading to a consistent, optimal graph database. Absence of any errors that can impact the final database and target queries. Figure/diagram of the resulting graph model provided. Consistent and clear explanation in the report, with no points to clarify.
	No graph database created. Missing or incorrect number of	Graph database successfully created. Correct number of nodes.	Graph database successfully created. Correct number of nodes	Graph database successfully created. Correct number of nodes

2A.2	nodes and/or properties (key/value pairs). Missing or poorly formed relationships. Presence of errors that compromise data loading and target queries. Missing data preparation (CSV headers). Missing figure/diagram of the resulting database. Poor/missing explanations in the report. No code submitted.	Minor errors related to the structure of the graph database or mapping of properties and relationships from the property graph data model, with moderated influence in the resulting database and target queries. Figure/diagram of the resulting database provided. Satisfactory explanation in the report, with points to be better detailed.	and properties, with no errors. Correct mapping of relationships, leading to a consistent graph database. Minor errors impacting the target queries. Figure/diagram of the resulting graph model provided. Consistent and clear explanation in the report, with minor points to clarify.	and properties, with no errors. Excellent mapping of relationships, leading to a consistent, optimal graph database. Absence of any errors that can impact the final database and target queries. Figure/diagram of the resulting database provided. Consistent and clear explanation in the report, with no points to clarify.
2B	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report). Faulty or partial solution not addressing all the required points. Faulty results (retrieval of partial information to fulfil the query). No discussion/justification of high-rate threshold. Poor/missing explanation of the query in the report. No code submitted.	Good solution based on standard Cypher constructs, with minor points for improvement and/or errors not affecting the overall results. Good results but with room for improvement (for instance, presence of duplicated information, missing average ratings or other aspects influencing the results). Satisfactory explanation in the report, including threshold decision, with points to be better detailed. Legible and documented code, with points for improvement.	Good solution based on complex Cypher constructs (when necessary), with minor points for improvement and no errors. Good use of Cypher commands or other functions necessary to retrieve the required data. Some redundancy of code is tolerated. Good results with no need for improvement. Consistent and clear explanation in the report, including threshold decision, with minor points to clarify. Legible and well-documented code, with no points for improvement.	Excellent solution based on complex Cypher constructs (when necessary), with no points for improvement and no errors. Cypher commands or other functions necessary to retrieve the required data. Efficient queries (minimal redundancy, optimal pipeline). Proper handling of edge cases (null values, empty results). Excellent results, generalisable across different users. Consistent and clear explanation in the report, including threshold decision, with no points to clarify. Legible and well-documented code, with no points for improvement.
2C	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report), unless no triangles exist in the dataset. Faulty or partial solution not addressing all the required points. Faulty results (retrieval of partial information to fulfil the query). No discussion/justification of high-rate threshold. Poor/missing explanation of the query in the report. No code submitted.	Good solution based on standard Cypher constructs, with minor points for improvement and/or errors not affecting the overall results. Good results but with room for improvement. Satisfactory explanation in the report, including threshold decision, with points to be better detailed. Legible and documented code, with points for improvement.	Good solution based on complex Cypher constructs (when necessary), with minor points for improvement and no errors. Good use of Cypher commands or other functions necessary to retrieve the required data. Some redundancy of code is tolerated. Good results with no need for improvement. Consistent and clear explanation in the report, including threshold decision, with minor points to clarify. Legible and well-documented code, with no points for improvement.	Excellent solution based on complex Cypher constructs (when necessary), with no points for improvement and no errors. Cypher commands or other functions necessary to retrieve the required data. Efficient queries (minimal redundancy, optimal pipeline). Proper handling of edge cases (null values, empty results). Excellent results, generalisable across different users. Consistent and clear explanation in the report, including threshold decision, with no points to clarify. Legible and well-

				documented code, with no points for improvement.
2D	Lack of solution (question not attempted). Lack of results (no outputs shown in the code or report). Faulty or partial solution not addressing all the required points (for example, missing score calculation or no changes to the graph database to reflect such scores). Faulty results (retrieval of partial information to fulfil the query). No justification of thresholds or other aspects influencing query understanding. Poor/missing explanation of the query in the report. No code submitted.	Good solution based on Cypher constructs and/or necessary Python functions, with minor points for improvement and/or errors not affecting the overall results. Good results (output) but with room for improvement. Satisfactory explanation in the report, including threshold decision and other aspects of score calculation, with points to be better detailed. Correct inclusion of total score in the graph database, with points for improvement. Legible and documented code, with points for improvement.	Good solution based on Cypher constructs and/or necessary Python functions, with minor points for improvement and no errors. Some redundancy is tolerated. Good results (output) with no need for improvement. Consistent and clear explanation in the report, including threshold decision and other aspects of score calculation, with minor points to clarify. Legible and well-documented code, with no points for improvement.	Excellent solution based on Cypher constructs and/or necessary Python functions, with no points for improvement and no errors. Efficient queries (minimal redundancy, optimal pipeline) and score calculation approach. Proper handling of edge cases (null values, empty results). Excellent results, generalisable across all movies. Consistent and clear explanation in the report, including threshold decision, and other aspects of score calculation, with no points to clarify. Legible and well-documented code, with no points for improvement.
Presentation	Poor presentation. Messy and/or missing sections in the report. Messy and/or non-documented code. Missing or unclear figures and/or explanations. Report does not respect page limit. Any missing deliverables.	Good report with the necessary sections and within page limit, with points to be better detailed. Clear figures/outputs and explanations, with points for improvement. Good quality and structure of all delivered products.	Well-structured report with the necessary sections, within the page limit and minor points for improvement. Clear figures/outputs and explanations, with minor improvement. Good quality and structure of all delivered products, allowing reproducibility.	Well-structured report with the necessary sections, within the page limit and no points for improvement. Clear figures/outputs and sound explanations, with no need for improvement. Justification of extensions/modifications to the proposed scenarios and/or data, aiming to improve the target queries. Good quality and structure of all delivered products, allowing reproducibility.

Observation: as per School and course-specific policy, you may acknowledge the use of any generative AI tool in any part of your summative work. You may note that marks can be deducted if no acknowledgement is made and/or a substantial part of your work (especially coding) is done by these tools. You may use these tools literally as a "co-pilot" to help you prototype your database models, generate synthetic data, and/or structure your SQL queries, but the results must be your own, validated work.