Text

Description automatically generated with medium confidence**Softwarica College of IT & E-Commerce**

STW7086CEM Data Management System

**Assignment Brief [TBA]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Title:**  Data Management System | **Ind/Group:**  Group | **Cohort:**  TBA | **Module Code:**  STW7086CEM |
| **Coursework Title:**  CW – Data Management | | | **Handout Date:**  TBA |
| **Lecturer:**  Manoj Shrestha | | | **Due Date:** TBA |
| **Estimated Time (hrs.) [20]**  **Word Limit: Up-to 600. Applicable to Part D only.** | **Coursework Type:**  Project Development | | **% Of Module Mark**  30% |
| Submission arrangement online via Softwarica Portal:  The link to your GitHub classroom repository containing app source code must be submitted on Softwarica Portal.  File types and method of recording:  Mark and Feedback date: 3 weeks after submission  Mark and Feedback method: written feedback using Softwarica Portal | | | |

|  |
| --- |
| **Module Learning Outcomes Assessed:**   1. Demonstrate a sound understanding of the theoretical and practical issues relevant to data management systems. 2. Critically evaluate a range of conceptual and technical tools and to apply them selectively in the design and implementation of an effective data management system 3. Assess and evaluate the theoretical and technological underpinnings of distributed frameworks 4. Review and comment critically on the current trends in distributed processing |
| **7086CEM – Data Management Systems**  This assignment is made up of four parts:   * Part A refers with database design, using E-R modelling. * Part B deals with SQL programming * Part C covers sequential and distributed processing * Part D involves a research report   **Please note that plagiarism (copying and pasting from online sources) and collusion (submitting the same content as a fellow student) are detected automatically by Turnitin and flagged. You should be able to check Turnitin for similarity. You are expected to submit your own individual work.**  **Part A**  A human resource management (HRM) department would like to build a database to track its personnel. The firm is separated into several divisions, and each employee is allocated to one of them. Employees are recognized by their employee id and name, whereas departments are identified by their department id and title. There are two categories of workers: shop floor workers and office workers. Each worker on the shop floor has a distinct talent and performs a certain duty. Office personnel, on the other hand, are recognized by their function (such as project member, project coordinator, or product lead) and department.  The department has a designated Manager who is ultimately responsible for the department and its personnel. To assist with department management, a number of personnel are designated to supervise groups of employees. When a new employee enters the organization, past job history and credentials must be provided. Each employee is obliged to undergo a review on a regular basis on a set day, which is generally carried out by the Manager but may be outsourced to a chosen representative.  The organization has identified a variety of job types (which differ from roles), such as Manager, Business Analyst, Salesperson, and Secretary, and each type has a number of grades connected with it, which affects the employee's remuneration for most non-senior positions. Positions are assigned to departments based on their workload. For example, two new Business Analyst posts may be assigned to a department. One person will occupy a role, but over time, individuals will fill a variety of positions.   1. Create an ER diagram for the above scenario and indicate the cardinality of the relationships and their nature (mandatory or optional). You should allocate adequate attributes to the entities of interest, and specify the identifiers.   (20%)   1. Generate, with appropriate justification, relational tables from the ER diagram, in their schema form. Clearly indicate the names of the tables, the attributes, the primary keys, and the foreign keys. Explain briefly how the tables were generated and which rules of transformation were used.   (5%)  *Guidance: for each case i) Create the ER diagram and clearly identify any identifiers, and indicate the cardinality of relationships and the nature of the associations (mandatory or optional). ii) Generate tables and include primary and foreign keys. Use the schema notation; you do not have to produce SQL statements. Example of table generation in schema form: Course(courseId, courseName)* |
| **Part B : SQL programming**  Consider the following Employee database and sample data.  You may wish to add more data records.  Employee (empId, name, address, DOB, job, salaryCode, deptId, manager, schemeId) Department (deptId, name)  SalaryGrade (salaryCode, startSalary, finishSalary)  PensionScheme (schemeId, name, rate)  **Table : Employee**   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **EmpID** | **Name** | **Address** | **DOB** | **Job** | **SalaryCode** | **DeptID** | **Manager** | **SchemeID** | | E101 | Young, S. | 199 London Road | 05/03/76 | Clerk | S1 | D10 | E110 | S116 | | E301 | April,  H. | 20 Glade close | 10/03/79 | Sales Person | S2 | D30 | E310 | S124 | | E310 | Newgate,  E. | 10 Heap Street | 28/11/80 | Manager | S5 | D30 |  | S121 | | E501 | Teach,  E | 22 railway road | 12/02/72 | Analyst | S5 | D50 |  | S121 | | E102 | Hawkins,  M. | 3 High Street | 13/07/74 | Clerk | S1 | D10 | E110 | S116 | | E110 | Watkins,  J. | 11 crescent road | 25/06/69 | Manager | S5 | D10 |  | S121 |   **Table : Department**   |  |  | | --- | --- | | **DeptID** | **Name** | | D10 | Administration | | D20 | Finance | | D30 | Sales | | D40 | Maintenance | | D50 | IT Support |   **Table : SalaryGrade**   |  |  |  | | --- | --- | --- | | **SalaryCode** | **StartSalary** | **FinishSalary** | | S1 | 15000 | 18000 | | S2 | 18001 | 22000 | | S3 | 22001 | 25000 | | S4 | 25001 | 29000 | | S5 | 29001 | 38000 |   **Table PensionScheme**   |  |  |  | | --- | --- | --- | | **SchemeID** | **Name** | **rate** | | S110 | AXA | 0.5 | | S121 | Premier | 0.6 | | S124 | Stakeholder | 0.4 | | S116 | Standard | 0.4 |  1. Use appropriate data types and write the SQL statements to create the tables defined in the schema above.   (10%)   1. Write SQL Statements to return the following data from the Employee database: 2. The name (in ascending order), the starting salary and department id of each employee within a descending order of department ids. (5%) 3. Give the number of employees for each of the pension schemes offered by the company. Result listing should include the name of each scheme and its corresponding number of employees who join the scheme. (5%) 4. Give the total number of employees who are not managers but currently receive an annual salary of over £35,000. (5%) 5. List the id and name of each employee along with his/her manager’s name. (5%)   *Guidance: Please use standard SQL. Clearly indicate the primary keys and the foreign keys. State the SQL statements and give the results. The presentation of each query should have a text summary which includes*  *i) the query itself,*  *ii) the corresponding SQL statement solution,*  *iii) the result of the execution of the statement and*  *iv) evidence that you have used standard SQL and implemented each statement on a database (use screenshots or spool facility). A small data sample is given. When appropriate, you can create and insert additional data records in order to make sure that the queries return results.* |
| **Part C : Sequential and distributed processing**  Consider the following data structure for a sales data store, where all values are either integers or reals. Each record has eight attributes; the description specifies the list of permissible attribute values and format (metadata).  OrderNo Integer  ProductNo Integer  Price Price of each product (Real/float)  Quantity Integer  Sales Real/float  QtrId Quarter (1-4)  MonthId Integer (1-12)  YearId Integer  The following table contains sample records:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | OrderNo | ProductNo | Price | Quality | Sales | QtrId | MonthId | YearID | | 10107 | 2 | 85.7 | 30 | 2587 | 1 | 2 | 2003 | | 10107 | 5 | 95.8 | 39 | 3879.49 | 1 | 2 | 2003 | | 10121 | 5 | 71.5 | 34 | 2700 | 1 | 2 | 2003 | | 10134 | 2 | 94.74 | 41 | 3884.34 | 3 | 7 | 2004 | | 10134 | 5 | 100 | 27 | 3307.77 | 3 | 7 | 2004 | | 10159 | 14 | 100 | 49 | 5205.27 | 4 | 10 | 2005 | | 10168 | 1 | 96.66 | 36 | 3479.66 | 4 | 10 | 2006 | | 10180 | 12 | 100 | 42 | 4695.6 | 4 | 11 | 2006 | | … | … | … | … | … | … | … | … |     The managers of the company would like to determine, for each product, the number of products which were sold in each month of each year.   1. Assuming that the data is stored in a relational database produce, with justification, the SQL code to determine, for each product, the number of products which were sold in each month of each year. (5%) 2. Assuming that the data is too large to be processed in a centralised manner in a relational database, and that it is stored in an ordinary file, produce a decentralised solution which applies MapReduce to the data processing. Justify your decisions and all the steps of your solution. Use diagrams if required. (20%)   *Guidance: You should carefully study the examples of mapReduce covered in the lecture notes. You should consider the structure of the key in the (key, value) pair in the original record and in the mapping stage. This is not a programming exercise. The solution should follow the structure given in the lecture notes.* |
| **Part D : Research report**  Consider the following web article quote:  "With over 1.5 million new advertisements posted every day, Craigslist (a classified listing site) users have created over a billion records - some may even call that 'big data.' Furthermore, for statutory compliance, documents must be transferred to an archival area after a 60-day retention term on the active portion of the site.  Craigslist experienced a number of issues as a result of the kind and volume of data kept on its relational MySQL servers. The structure of their data, for example, has altered multiple times throughout the years. This alone made each modification to the database schema an expensive, time-consuming headache, as changes frequently resulted in outages. And if changing the database was tough, imagine how difficult it was to add totally new functionality. Furthermore, each change to the live database structure necessitated a matching modification to the whole archive - a process that took months each time".  Craigslist opted to switch its data and operations from relational MySQL servers to NoSQL MongoDB servers in 2011.  Refer to the advantages and disadvantages of relational and NoSQL databases, and explain in no more than 700 words why Craigslist switched from MySQL to MongoDB.  *Guidance: Use your own words for the report. Copying and pasting is plagiarism. You should include relevant references. The maximum length of the report is 700 words. Longer reports will be penalised.* |

**Marking Rubric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grade** | **Part A** | **Part B** | **Part C** | **Part D** |
| **<40** | * Incorrect interpretation of scenario and Incomplete formulation of solution * Limited identification of entities and poor annotation of relationships * Incorrect generation of relational tables * Limited or absent rationale | * Poor interpretation of requirements and of queries * DDL and DML SQL statements limited in scope * Incomplete and incorrect SQL statements * Absence of rationale | * Partial understanding of requirements and partially correct SQL formulation * Partial understanding of context and relevance of parallel processing * Incomplete steps in the application of MapReduce * Partial justification of design decisions | * Lack of understanding of requirements * Inadequate identification of issues * Incompetent understanding of structural and processing components * Poorly written essay |
| **40 - 49** | * Partial interpretation of scenario and formulation of solution * Partially correct ER diagram with relevant entities and relationships * Partial consistency in the generation of the relational tables * Partial justification of design decisions | * Basic understanding of requirements and partially correct interpretation * Relevant use of DDL and DML statements in solution formulation * Partially complete SQL statements * Limited justification of solution | * Partial understanding of requirements and partially correct SQL formulation * Partial understanding of context and relevance of parallel processing * Incomplete steps in the application of MapReduce * Partial justification of design decisions | * Partial interpretation of requirements * Limited presentation of key issues * Relevant description of structural and processing component * Mostly descriptive essay |
| **50 -59** | Adequate and consistent interpretation of scenario and satisfactory conceptual modelling  · Mostly correct generation of ER diagram with relevant entities and relationships · Relatively competent generation of the relational tables  · Adequately justified design decisions | · Adequate understanding of requirements and mostly correct interpretation  · Adequate use of DDL and DML SQL statements in solution formulation  · Mostly complete SQL statements  · Adequate justification of solution | · Adequate understanding of requirements and correct SQL formulation  · Adequate presentation of context of application of parallel processing · Mostly correct application of the different steps of MapReduce  · Adequately justified solution | · Adequate interpretation of requirements  · Key issues well identified and partially addressed  · Adequate presentation of key structural and processing components  · Adequately written essay |
| **60-69** | · Good and clear interpretation of scenario and satisfactory solution formulation of the two parts  · Correct and complete identification of entities and relationships  · Consistent generation of relational tables  · Well expressed rationale | · Good interpretation of requirements and good formulation of solution  · Correct use of DDL and DML statements in solution formulation  · Complete and relevant treatment of queries  · Good justification of decisions | · Satisfactory solution to the initial query in terms of SQL statements  · Focused presentation of context of application of parallel processing · Clearly stated and correct sequential steps of MapReduce  · Well expressed rationale | · Good understanding and statement of requirements  · Well focused presentation of key issues  · Good description of structural and processing components  · Well-presented essay with some reflection |
| **70+** | · Very good and clear interpretation of scenario and excellent solution formulation of the two parts  · Correct and coherent identification of well annotated entities and types of relationships  · Logical and consistent generation of relational tables  · Excellent justification of design decisions | · Excellent interpretation of requirements and very good formulation of solution  · Excellent use of DDL and DML statements in solution formulation  · Complete treatment of queries and SQL formulation  · Very good rationale | · Very good solution to the initial query in terms of SQL statements  · Excellent presentation on the need for an overall parallel solution  · Clear deployment and annotation of the sequential steps of MapReduce  · Excellent rationale | · Excellent understanding and interpretation of requirements  · Very good identification and formulation of key issues  · Relevant and specific presentation of structural and processing components  · Reflective writing supported by an excellent structure |