|  |  |  |  |
| --- | --- | --- | --- |
| Surname, Name: | **Henry Hayes** | Student Nr: | **30326052** |
|  |  |  |  |
| module: | **CMPG321 Advanced Databases** |  |  |
|  |  |  |  |
| Assessment: | **Assignment 2 – SU 3** | Due date: | **2020/10/06** |



**SU3 – Distributed Database Management Systems**

* 1. Name and briefly describe THREE data allocation strategies. /

Centralized data allocation: The entire database is stored at one site

Partitioned data allocation: The database is divided into two or more disjointed fragments and stored at two or more sites.

Replicated data allocation: Copies of one or more database fragments are stored at several sites*.* **[3]**

* 1. Name and briefly describe THREE data replication scenarios. /

Fully replicated: This scenario refers to multiple copies of each database fragment being stored at multiple sites

Partially replicated: This scenario refers to multiple copies of some database fragments being stored at multiple sites

Unreplicated: Stores each database fragment at a single site.**[3]**

* 1. Identify the different types of database requests and transactions based on the descriptions below. /   
     *Identifiseer die verskillende tipes databasis-versoeke en trasaksies aan die hand van die onderstaande beskrywings .* **[4]**

A \_\_remote request\_\_ accesses data located at a single remote database processor (or DP site). In other words, an SQL statement (or request) can reference data at only one remote DP site.

A \_\_\_remote transaction**\_\_\_\_**, composed of several requests, accesses data at only a single remote DP site.

A \_\_**\_\_distributed request\_** provides the ability to reference and access data located at several different DP sites.

A \_\_\_distributed transaction**\_\_\_** allows a transaction to reference several different local or remote DP sites. Although each single request can reference only one local or remote DP site, the complete transaction can reference multiple DP sites because each request can reference a different site.

* 1. Transaction transparency is a Distributed Database Management System (DDBMS) property that ensures that database transactions will maintain the distributed database’s integrity and consistency. Consider the DDBMS scenario below (see figure) which is spread over three sites (A, B and C) and answer the questions that follow. Specify the minimum type(s) of operation(s) that the database must support in order to perform the following SQL statements. /

*Transaksie deursigtigheid is ŉ Verspreide Databasisbestuurstelsel (VDBBS) eienskap wat verseker dat databasis transaksies die verspreide databasis se integriteit en konsekwentheid onderhou. Beskou die VDBBS scenario hieronder (sien onderstaande figuur) wat oor drie plekke (A, B en C) versprei is en beantwoord die vrae wat volg. Spesifiseer die minimum tipe(s) bewerking(s) wat die databasis moet ondersteun om die volgende SQL stellings te kan uitvoer.* **[5]**



|  |  |  |
| --- | --- | --- |
| **SQL Stelling /  *SQL Statement*** | **Vanaf plek / *From Site*** | **Tipe bewerking /  *Type of operation*** |
| ***e.g.*** *SELECT \* FROM CUSTOMER* | *B* | ***Remote Request*** |
| SELECT CUST\_NAME, INV\_TOTAL FROM CUSTOMER, INVOICE WHERE (INV\_TOTAL > 1000) AND (CUSTOMER.CUST\_NUM = INVOICE.CUST\_NUM); | A | **1.4.a)Distributed request** |
| SELECT \* FROM CUSTOMER  WHERE CUST\_BAL < 10000 | B | **1.4.b)Remote request** |
| UPDATE PRODUCT SET PROD\_QOH = PROD\_QOH – 1 WHERE PROD\_CODE = 1337;  UPDATE CUSTOMER SET CUST\_BAL = CUS\_BAL + 420 WHERE CUST\_NUM = 007;  COMMIT; | C | **1.4.c)Distributed transaction** |
| SELECT \* FROM PRODUCT  WHERE PROD\_QOH < 10 | C | **1.4.d)Distributed request** |
| SELECT \* FROM PRODUCT  WHERE PROD\_QOH < 10 | A | **1.4.e)Remote Request** |