**103428 3A Fredah Kioko**

**CAT 2**

**QUESTIONS (12 marks)**

1. What is the difference between a temporal trigger and a normal trigger? (2 marks

A temporal trigger or a scheduled event is an action, a consequence or an outcome that may occur periodically based on certain intervals of time and in the case of a database certain SQL statements are written respective to time so that something happens in the future example a supermarket might need a report of all sold out items at the end of every month do that they re-stock this might also lead to the knowledge of most often and less often sought goods by the customer and lead to the managerial decision of less or more purchase to make more profit. While a normal trigger is special stored procedure that is run when specific actions occur within a database. Most triggers are defined to run when changes are made to a table's data. Triggers can be defined to run instead of or after DML (Data Manipulation Language) actions such as INSERT, UPDATE, and DELETE for example one can use the command trg\_before\_update\_on \_stock to store the old data of stock before adding new stock.

1. What is the difference between a row-level trigger and a statement-level trigger? (2 marks

A row-level trigger is activated for each row that is inserted, updated, or deleted. For example, if a table has 50 rows inserted, updated, or deleted, the trigger is automatically fired 50 times for each of the 50 rows affected.

On the other hand, a statement-level trigger is executed once for each database transaction regardless of how many rows are inserted, updated, or deleted.

1. List 2 uses of scheduled events in a database system. (2 marks)

- To generate reports periodically depending on the specified time constraint using temporal triggers either monthly, weekly, daily or even hourly.

- To archive data for future reference that is before any alterations is made to any attributes of the database table.

1. List and explain SIX techniques that can be used to overcome performance bottlenecks caused by disk I/O in database systems. (6 marks)

Most I/O bottlenecks form because of lack of storage to support the needs of the database, which grows over time as more queries are written. Some of the techniques used to overcome performance bottlenecks given to us by Omondi, A. O. (2020). include :-

1. Indexes: an auxiliary data structure that supports fast access to the tuples of a

relation at the cost of additional writes to maintain it. The indexing makes the

execution of the query more efficient because the result set is obtained by performing a lower number of scans (reads) for tuples stored in pages on the HDD

1. Query optimization: choosing the best execution strategy for relational algebra

operations DBMS finds the most cost-effective access path without user intervention. SQL Query Optimization Technique is the process of ensuring that the SQL statements will run in the fastest possible time.

1. Caches: storing a copy of critical data in high-performance memory The cache must be set large enough to permit as many data requests to be serviced from cache as possible.
2. Prefetching: anticipating the data that will be required and fetching it even

before a query for it has been issued. prefetching is used to execute I/O operations in parallel, in a way that is somewhat similar to a read-ahead mechanism. It is used in the nested loops execution plan when there are more than a threshold number of rows in the outer input table. used by viewing the property WithOrderedPrefetch or WithUnorderedPrefetch in the nested loops operator.

1. Redundant Array of Independent Disks (RAID): supports concurrent query processing. RAID level 6 is considered to be affordable and useful for high performance transaction processing in business applications (OLTP systems) RAID 6 arrays provide extra protection for your data because they can recover from two simultaneous disk drive failures.
2. Performance tuning: implementation of configurations geared towards improving a system’s capability of meeting its non-functional requirements, the purpose for tuning is to reduce resource consumption or to reduce the elapsed time for an operation to complete. Either way, the goal is to improve the effective use of a particular resource, the tuning phase is left until the database is in production. At this time, tuning becomes a reactive process, where the most important bottleneck is identified and fixed.

Reference

Omondi, A. O. (2020). BBT 3104 – Advanced Database Systems – Lecture Slides[Concept 3 of 6]. Retrieved from Strathmore University eLearning website

<https://elearning.strathmore.edu/pluginfile.php/222172/mod_resource/content/9/3.a.Concept3of6-Data_Storage.pdf?redirect=1>