**Te Hoe Rorohiko**

**Department of Enterprise and Digital Innovation**

Bachelor of Information and Communication Technologies

Database Administration

BCDE214

Assignment Two

Portfolio

Semester Two, 2020

Due date: 20 November 2020 with intermediate dates to be advised

Time: 5.00pm

Instructions:

Hand in a zipped portfolio of scripts, screen shots and other documents as requested during the semester.

**TOTAL MARKS: 50**

Student Name/ID James Aeron Susbilla

*Ara and its faculty members reserve the right to use electronic means to detect and help prevent plagiarism. Students agree that when submitting this assignment, it may be subject to submission for textual similarity review to Turnitin.com.*

Submissions received late will be subject to a penalty of 10% of the student’s mark per working day.

This paper has ***three*** (3) pages including the cover sheet.

**BCDE214 Database Administration Assignment 2– Portfolio**

**Appendix C Database Functions and Spatial Data Analysis**

**Due Date:** Friday 20 November; 11:59 p.m. **(Total Marks 50)**

1. **Recovery**
2. When does a DBA perform forward recovery? What are the forward recovery steps? Explain in detail.

Forward Recovery is performed after a database is destroyed that is due to an event that was catastrophic. The process of performing forward recovery is done in two steps, such steps are that a database admin first copy the latest database backup over a database that is live then the features of database management system forward recovery utilizes logs to employ after images to transaction that was committed

(5 marks)

1. If a transaction is interrupted, what steps does the DBMS take to correct the database? Explain in Detail. Show example with Roll back and Save point in MySQL.

When a transaction is interrupted during a query execution, DBMS rollbacks to the last

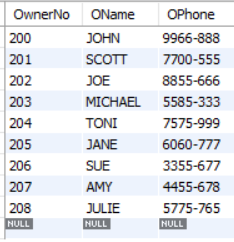
save point of the query, such rollback is required to prevent any database modifications as well as to prevent any creation of redundancies. An example of rollback can be illustrated when inserting new owner records at the homes database. We will first begin by starting a transaction,

USE homes;

START TRANSACTION;

Then we will save the original records of the owner entity of the homes database to use as a rollback save point. Let us call the save point newowner.

SAVEPOINT newowner.



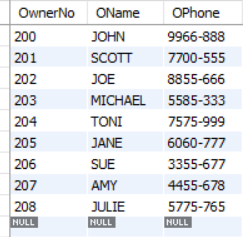
Suppose that while loading new owner records to the owner entity, we encountered an error causing transaction interruption. In this example, DBMS will not attempt to load the records, such attempt could cause data redundancies as some records could be loaded & others may not.

INSERT INTO owner VALUES (209, ‘Adams’,’867-9086’),(210,’Sue’,’123-2834’);

Rather, DBMS will roll back to the last save point created, that is in our case the save point called newowner.

In performing this rollback, the inserted record in the owner table will not exist.





(5 marks)

1. **Transaction Management**
2. Briefly explain what is meant by a “lost update” and give an example to illustrate your answer.

Lost update is when two transactions occurring at the same time attempts to read and update the same column and row, depending on which transactions is committed first, the item value updated by the first transaction committed is overwritten by the second transaction made, such results in the first transaction update being lost and thus, causes inconsistency with the updated item values. An example of lost update can be illustrated below.

Consider an example where 2 different individuals named Glenn (Transaction 1) & Pete (Transaction 2) are buying Keri apple juice in bulk, such individuals will each have their own transaction.

Transaction Two

Transaction One

BEGIN TRANSACTION;

SELECT \* FROM JUICE WHERE Brand = ‘Keri’ and flavour = ‘apple’;

BEGIN TRANSACTION;

SELECT \* FROM JUICE WHERE Brand = ‘Keri’ and flavour = ‘apple’;

Buys 40 bottles

Buys 53 bottles

Updates Keri apple juice stocks to 72 bottles

Updates Keri apple juice stocks to 59 bottles

COMMIT;

COMMIT;

In this example, both individuals will read the same available Keri apple juice stocks that is 112 bottles. Transaction 2 will first buy 40 bottles of apple juice then Transaction 1 will buy 53 bottles. Transaction 2 will finish its transaction first with a new apple juice stock of 72 bottles. Then transaction 1 will finish its transaction with an apple juice stock of 59 bottles. These two Transaction apple juice stocks are wrong as such the right amount of apple juice stocks left should be 19 bottles.

1. In what circumstances can the problem of “deadlock” or “deadly embrace” arise? Support your answer with an example.

Deadlock arises if the following condition are all true,

* when there are multiple processes attempting to use a resource at the same time.
* when a process that is already using a resource, is requesting more resources, that is currently used by other process.
* A process’s resource is not able to be taken away by force, but only the process is able to free the resource it holds.
* When two processes are waiting for each other held resources, such can be shown below in which it creates an infinite wait loop.

Held by

wants

Process 1

Resource 1

Resource 2

wants

Held by

Process 2

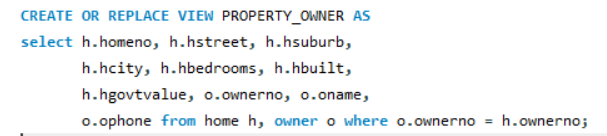
In this example, Process 1 is wanting Resource 2, but it is currently held by Process 2 and Process 2 is wanting Resource 1 that is currently held by Process 1,

1. Outline two (2) ways in which the problem of “deadlock” or “deadly embrace” can be overcome by a database management system.

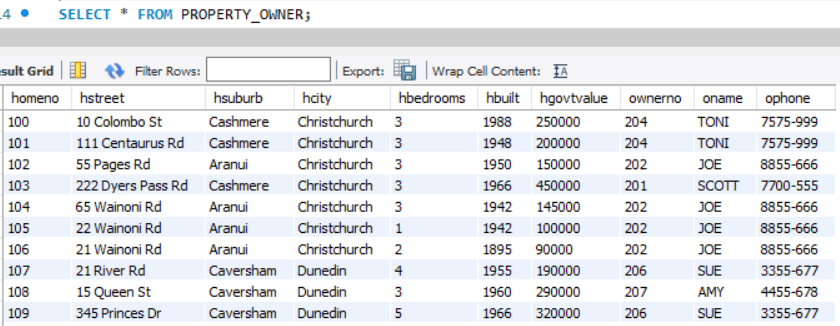
**Database Management System or DBMS can overcome the problem of deadlock or** also known as deadly embrace by either killing the process in which one terminates the process individually, performing this action until the system is fully recovered from a deadly embrace. One can also overcome the deadly embrace by resource preemption in which the resources of a process that is in a deadlock, is given to other process, such actions results in system starvation.

1. **Distributed Databases**
2. Briefly describe the concept of location transparency, using an example to support your answer.

Location Transparency verifies whether an individual user, is able to query on entities or table fragments as if data is stored in the local site. With location transparency, user is not required to know the location of data, however the access to data should be easy in both local & remote site, the only difference between both local and remote site should only be the response time. Location transparency in a distributed database can be achieved by either creating a view, procedure or by using a synonym to hide the object and its location. An example of location transparency which can be shown easily is by creating a view that connects both home and owner entities of the homes database.



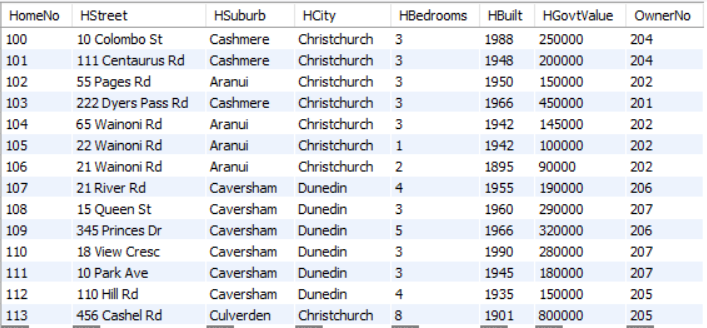
In this example, individuals can access the PROPERTY\_OWNER view without having knowledge of the data location and whether one is accessing more than one entity.



1. Briefly describe the concept of fragmentation transparency and explain why fragmentation transparency is done, using an example to support your answer.

Fragmentation transparency is segregating relations into multiple fragments such fragments are then moved and stored in several sites. With Fragmentation transparency, individuals can query on any entity without knowing that the table one is using, is either fragmented or part of a union with other fragments. With the use of fragmentation transparency in a database, one can enjoy the many benefits it offers, such benefits are security in which the data of local apps that isn’t needed is not saved, thus will be unavailable to any users that are not authenticated. It also offers efficiency that is data is stored near the location, it is most use. Furthermore, Fragmentations transparency offers 3 different type of fragmentation in which they are Horizontal, Vertical and hybrid data fragmentations. To illustrate Fragmentation transparency, I will be using the horizontal data fragmentation to show what fragmentation transparency is.

Consider a scenario in which individuals are owning a home at different locations like Christchurch and Dunedin etc. the number of individuals owning a home at these locations are currently small right now, however the number of individuals owning a property at these locations has the potential to further grow as such when any properties details are needed, the entire home entity has to be access in order to get property details.

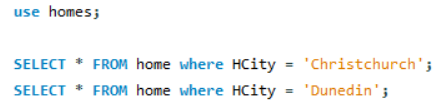


The home entity will be access multiple times by individuals who are around New Zealand such access of data needs to be fast in terms of database performance as such to improve database performance, we’ll have to separate the home entity horizontally by all means through the property location.

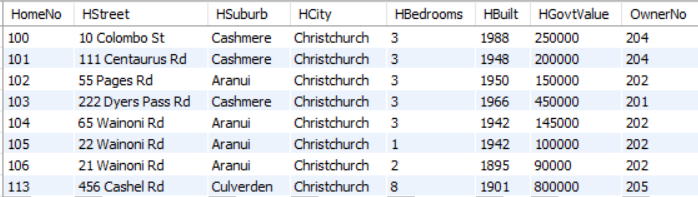
FRAGMENT 1 - Christchurch

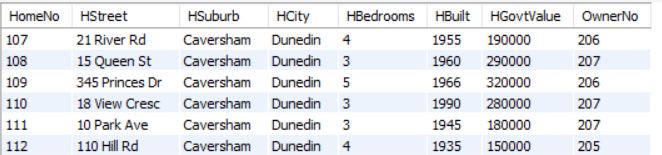
HOME

FRAGMENT 2 - Dunedin



The queries above will become a subset of home table records, such subset would be stored in their locations. Any transactions on the records of home entity would be made in the database at their respective locations. The transactions made will then affect the home entity periodically, such transactions can either be select, insert, update, delete etc.





1. **Data Warehousing**
2. What is a data warehouse and what are its main characteristics?

A Data warehouse is an organization repository that is containing data that are static or unchangeable meaning a user cannot modify it directly, such that it can utilized by the organization itself, for decision support tool to make wise decisions on the prospect of the business. Data Warehouse characteristics are subject oriented - one can do analysis on a particular theme i.e. sales, integrated – integrates data from multiple sources such as RDBMS, flat files etc. Non-volatile – data is static, it cannot be edited directly once in a data warehouse. Time-variant – data collected is systematic that is based on time such as month, years etc.

1. What does it mean when a data warehouse is non-volatile?

Non-volatile means that data is neither destroyed nor deleted in a data warehouse as such once it is in a data warehouse, it is read only – a user cannot just edit it directly. Data that is archive in a Data Warehouse are used for reporting in the future when its needed.

1. Why would an organization invest in a data warehouse?

Data Warehouse helps deliver enterprises/organizations an excellent decision-support tool, such tool can be utilized to obtain meaningful insights of data that are beneficial to one’s organization.

1. What is a fact table in a data warehouse?

Fact Table is a denormalized entity containing each primary key of the surrounding dimension entities that it relates to. A fact table can also have other attributes containing optionally that is for instance a datetime field. The design of a fact table is not utilized to produce data reports, but it is mainly use for physical activities within the database. A fact table consist of facts as well as measurements about the processes of an organization/enterprise. It is centrally located in the star schema.

1. What is a dimension table in a data warehouse?

A dimension table in a data warehouse consist of one primary key field as well as descriptive fields which is related to the dimension table primary key. The dimension table primary key is also utilized to be an index in any fact tables that it relates to. Additionally, constraints and groupings which are from business intelligence apps and queries, is commonly targeting the attributes of a dimension entity since these detailed attributes helps provide businesses data insights which can be used to provide measures & metrics.

1. **Communication Time Calculation (Marks 5)**

A Toy Company wants to list all items in TOY category with a price that is more than $100.00. For this query assume the following set of assumptions:

1. The Product table contains 1000 rows and is stored in a remote site;
2. Each record in the Product table is 500 bits long;
3. There is no special structure, such as an index, that would be helpful in processing this query faster; and
4. Only 10 of the 1000 rows in the Product table satisfy the conditions.

How would you process this query? Calculate the communication time for your query strategy.

SELECT \* FROM Product WHERE category = ‘TOY’ AND price > 100;

Supposed that the access delay is 5 seconds and transmission rate about 50000 bps. Then our communication time will be,

Communication time = access delay + (data volume/transmission rate)

= 5 + ((10 \* 500)/50000)

= 5 + 0.1

Communication time = 5.1 sec

Query Process

Sending a request to the remote site from local site about the full query result. Remote site searches every line of record of the Product table and identifies rows, which matches the query condition. The remote site then sends a form of acknowledgement to the local site, acknowledging that it has found the rows that satisfies the query condition.

1. **Spatial Query (Marks 5)**

Write a query that will:

Calculate Distance between any **two** locations (in Kms.) as provided in SQL file. Test your query by providing at least **5** suitable test scenarios. Use the Zip.SQL file for this task.

SEE THE ATTACHED CalculateDistance.sql SCRIPT

# References

CodesCracker. (2020, November 9). *OS Deadlock Recovery*. Retrieved from CodesCracker: https://codescracker.com/operating-system/deadlock-recovery.htm

Course Hero. (2020, November 9). *Data Warehouse Time Variant data is the data*. Retrieved from Course Hero: https://www.coursehero.com/file/p58g0f3/Data-warehouse-time-variant-Data-is-the-data-warehouse-is-characterised-by-the/

Frankenfield, J. (2020, November 9). *Data Warehousing*. Retrieved from Investopedia: https://www.investopedia.com/terms/d/data-warehousing.asp

GeeksforGeeks. (2020, November 9). *Deadlock Detection And Recovery*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/deadlock-detection-recovery/?ref=lbp

Oracle. (2020, November 9). *Creating Location Transparency*. Retrieved from Oracle: https://docs.oracle.com/cd/B28359\_01/server.111/b28310/ds\_admin006.htm#ADMIN12176

Oracle. (2020, November 9). *What Is a Data Warehouse?* Retrieved from Oracle: https://www.oracle.com/uk/database/what-is-a-data-warehouse/

Perucci, D. (2020, November 9). *What Is a Lost Update in Database Systems?* Retrieved from DZone: https://dzone.com/articles/what-is-a-lost-update-in-database-systems

Pratt, P., & Last, M. (2014). *Concepts of Database Management 8th Edition.* Boston: Cengage Learning.

Richardson, B. (2020, November 9). *The Lost Update Problem in Concurrent Transactions*. Retrieved from coding sight: https://codingsight.com/the-lost-update-problem-in-concurrent-transactions/

Smith, C. (2020, November 9). *Deadlocks and their implications in databases*. Retrieved from KnowTechie: https://knowtechie.com/deadlocks-and-their-implications-in-databases/

Tatum, M. (2020, November 9). *What Is Location Transparency?* Retrieved from wisegeek: https://www.wisegeek.com/what-is-location-transparency.htm

Thakur, D. (2020, November 9). *What is Fragmentation in Database? Advantages & Disadvantages of fragmentation.* Retrieved from Computer Notes: https://ecomputernotes.com/database-system/adv-database/fragmentation

Thakur, D. (2020, November 9). *What is Transparences in DDBMs?* Retrieved from Computer Notes: https://ecomputernotes.com/database-system/adv-database/transparences-in-ddbms

Tutorial Cup. (2020, November 9). *Data fragmentation in DBMS*. Retrieved from Tutorial Cup: https://www.tutorialcup.com/dbms/data-fragmentation.htm

TutorialRide.com. (2020, November 9). *Deadlock in Databases*. Retrieved from TutorialRide.com: https://www.tutorialride.com/dbms/deadlock-in-databases.htm

tutorialspoint. (2020, November 9). *DDBMS - Distribution Transparency*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/distributed\_dbms/distributed\_dbms\_distribution\_transparency.htm

tutorialspoint. (2020, November 9). *Deadlock Characterization*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/deadlock-characterization

tutorialspoint. (2020, November 9). *Methods for Handling Deadlocks*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/methods-for-handling-deadlocks

tutorialspoint. (2020, November 9). *Process Deadlocks in Operating System*. Retrieved from tutorialspoint: https://www.tutorialspoint.com/process-deadlocks-in-operating-system