UOW MALAYSIA
KDU PENANG
UNIVERSITY COLLEGE
—
PART OF THE UNIVERSITY
OF WOLLONGONG AUSTRALIA
GLOBAL NETWORK

ASSIGNMENT COVER PAGE

Validated by
UNIVERSITY OF
LINCOLN
UNITED KINGDOM

Programme	**Course Code and Title**
Bachelor of Computer Science (Hons)	CDB3033N (Database Programming)
Student's name / student's id	**Lecturer's name**
0204677 / Lim Zhe Yuan 0207778 / Saw Keat Loon 0203640 / Otheya Kumar	Ts. Chng Chern Wei

Date issued	**Submission Deadline**	**Indicative Weighting**
Week 8 - 30/10/2023	Week 11 - 20/11/2023	30%
Assignment [2]	Cursor Processing and Physical Database Design (Group Assignment)	

This assessment assesses the following course learning outcomes

# as in Course Guide	**UOWM KDU Penang University College Learning Outcome**
CLO3	Develop database applications implementing business rules with stored procedures. (C5, PLO3)
CLO4	Apply access methods to improve database performance and to build cost models. (C3, PLO7)
# as in Course Guide	**University of Lincoln Learning Outcome**
CLO1	Use appropriate tools and techniques to design a database
CLO2	Appraise the structure of a database design using standard evaluation mechanisms
CLO3	
CLO4	

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature:

Zhe Yuan

Otheya Kumar

Saw Keat Loon

Submission Date:

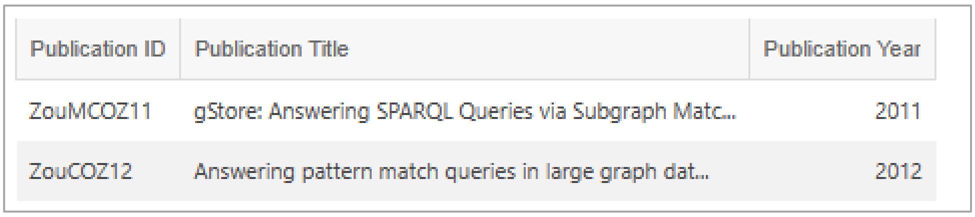
20/11/2023

Table of Contents

Main Report

Task 1

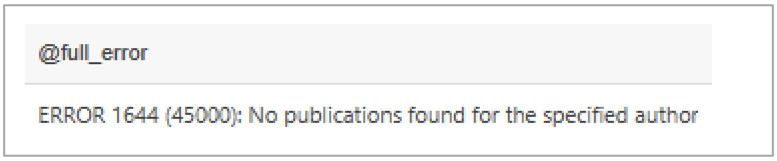
The screenshot below shows the results of the stored procedure executed, which is the list of publications records. If there is a publication records for the author that was selected, the results will show the publication id and publication title as well as ordered by publication year.



Publication ID	Publication Title	Publication Year
ZouMCOZ11	gStore: Answering SPARQL Queries via Subgraph Matc...	2011
ZouCOZ12	Answering pattern match queries in large graph dat...	2012

Figure 1: *Results of Task 1 stored procedure*

This screenshot shows that there are error handling functions implemented in the stored procedure so if there are errors such as an invalid author name has been entered, the result will be as shown in the screenshot.



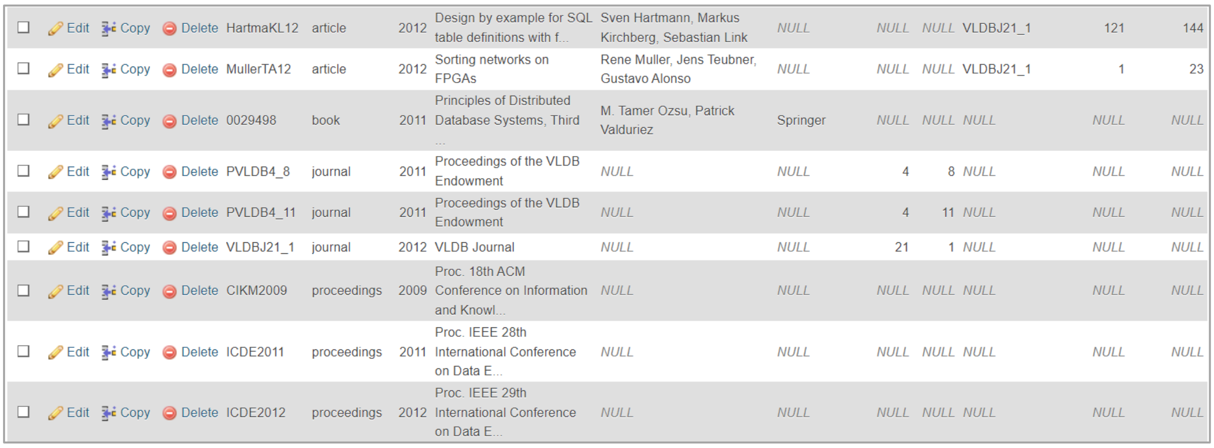
@full_error
ERROR 1644 (45000): No publications found for the specified author

Figure 2: *Error handling in stored procedure*

Task 2

a)

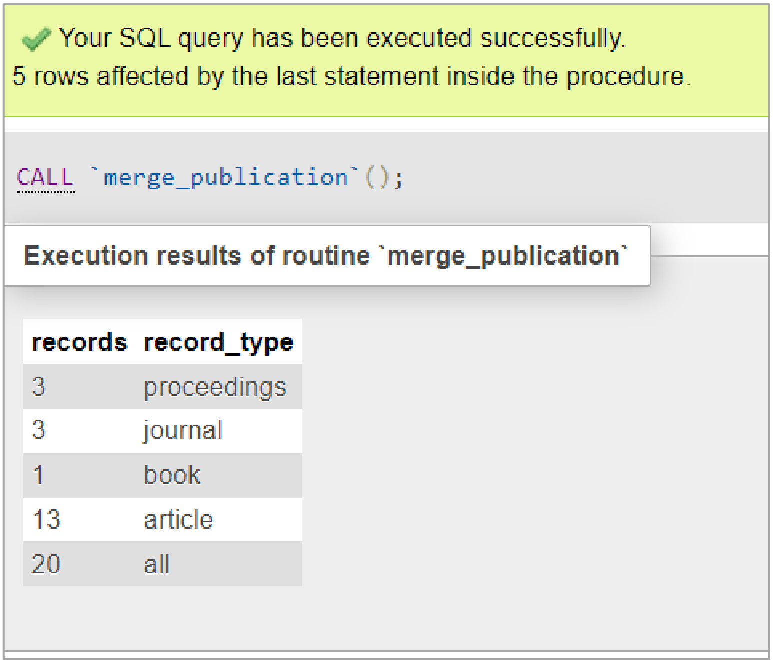
The *publication_master* table consolidates all publication records into a single table and categorizes each publication based on their type. It leaves NULL values in record attributes that are unrelated to the record's publication type. Additionally, the publication name, publication year, and author line is also retrieved respectively to their records. The year of articles settles on the year of the publication it has appeared in.



☐	Edit	Copy	Delete	HartmaKL12	article	2012	Design by example for SQL table definitions with f...	Sven Hartmann, Markus Kirchberg, Sebastian Link	NULL	NULL	NULL	VLDBJ21_1	121	144
☐	Edit	Copy	Delete	MullerTA12	article	2012	Sorting networks on FPGAs	Rene Muller, Jens Teubner, Gustavo Alonso	NULL	NULL	NULL	VLDBJ21_1	1	23
☐	Edit	Copy	Delete	0029498	book	2011	Principles of Distributed Database Systems, Third ...	M. Tamer Ozsu, Patrick Valduriez	Springer	NULL	NULL	NULL	NULL	NULL
☐	Edit	Copy	Delete	PVLDB4_8	journal	2011	Proceedings of the VLDB Endowment	NULL	NULL	4	8	NULL	NULL	NULL
☐	Edit	Copy	Delete	PVLDB4_11	journal	2011	Proceedings of the VLDB Endowment	NULL	NULL	4	11	NULL	NULL	NULL
☐	Edit	Copy	Delete	VLDBJ21_1	journal	2012	VLDB Journal	NULL	NULL	21	1	NULL	NULL	NULL
☐	Edit	Copy	Delete	CIKM2009	proceedings	2009	Proc. 18th ACM Conference on Information and Knowl...	NULL	NULL	NULL	NULL	NULL	NULL	NULL
☐	Edit	Copy	Delete	ICDE2011	proceedings	2011	Proc. IEEE 28th International Conference on Data E...	NULL	NULL	NULL	NULL	NULL	NULL	NULL
☐	Edit	Copy	Delete	ICDE2012	proceedings	2012	Proc. IEEE 29th International Conference on Data E...	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3: *Publication_master table values*

On invoking the stored procedure successfully, the system displays the total result of each publication and the total record sum inserted into the master table.



✓ Your SQL query has been executed successfully.
5 rows affected by the last statement inside the procedure.

```
CALL `merge_publication`();
```

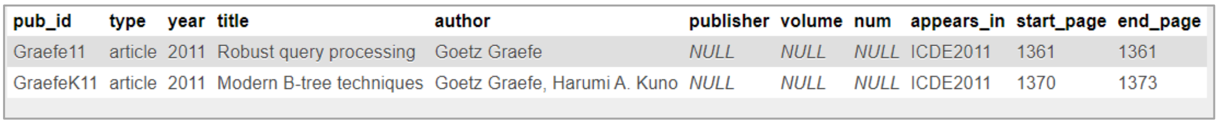
Execution results of routine `merge_publication`

records	record_type
3	proceedings
3	journal
1	book
13	article
20	all

Figure 4: *Results of Task 2a stored procedure*

b)

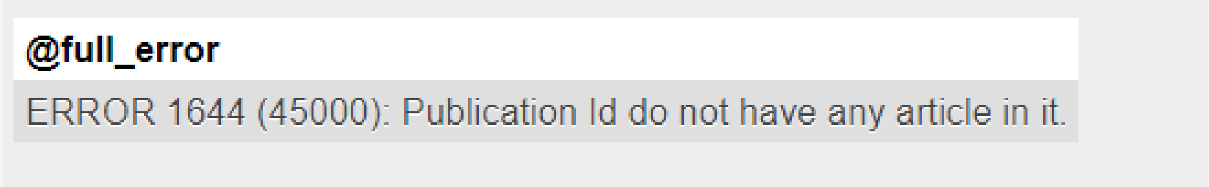
The screenshot below shows the output from successfully running the stored procedure. If there are articles found based on the given pubid, hence, it will display a list of articles that are found.



pub_id	type	year	title	author	publisher	volume	num	appears_in	start_page	end_page
Graefe11	article	2011	Robust query processing	Goetz Graefe	*NULL*	*NULL*	*NULL*	ICDE2011	1361	1361
GraefeK11	article	2011	Modern B-tree techniques	Goetz Graefe, Harumi A. Kuno	*NULL*	*NULL*	*NULL*	ICDE2011	1370	1373

Figure 5: *Results of Task 2b stored procedure*

However, it is important to include error handling functions. The screenshot below shows the output when the user input a pubid where there is no article in it.



@full_error
ERROR 1644 (45000): Publication Id do not have any article in it.

Figure 6: *Error handling in stored procedure*

Task 3

- **Database design review**

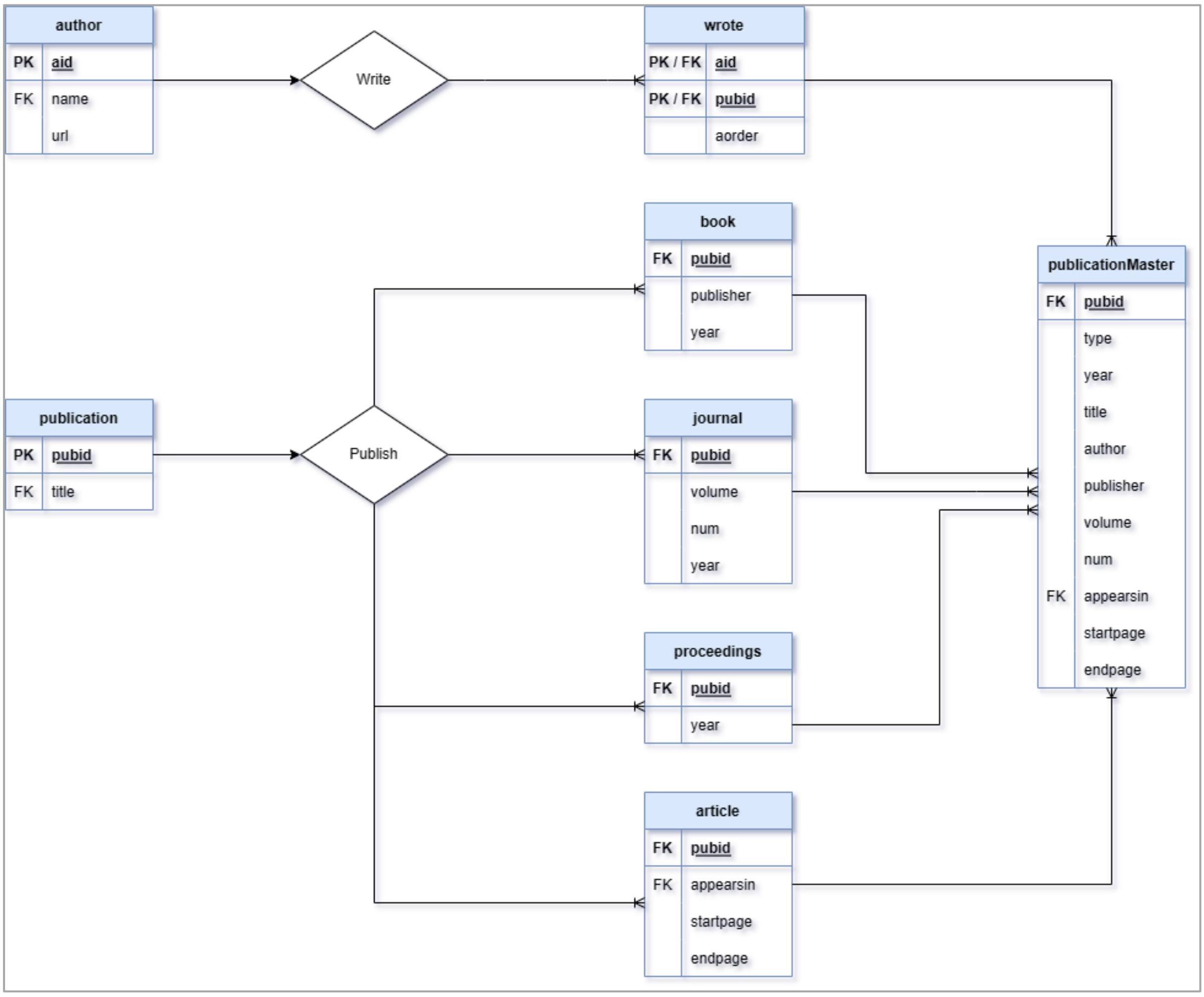


Figure 7: *ERD of all assignment DB entities*

As a brief explanation of the ER diagram above, it shows eight entities which are author, wrote, publication, book, journal, proceedings, article and publicationMaster. It also shows the relationships between them such as an author wrote a book, a book is a publication, a publication can be a journal, a publication can be a proceedings, a publication can be an article and publicationMaster contains all information from the other tables.

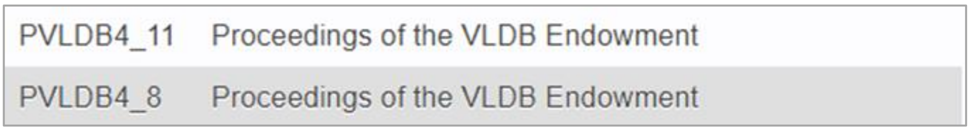
This above ER diagram shows the structure and relationship of the database. Each box is an entity with its attributes and they can be identified uniquely. For example, the entity author has aid, name and url as its attributes.

This ER diagram also uses cardinality lines to represent relationships which is the connection between entities in a database. There are many types of relationships but the prominent one in this database is one-to-many relationships. For example, the first relationship we see in the diagram is the relationship between author and wrote, the cardinality line shows a one-to-many relationship by the number on lines at the end of the line. This means that an author can write many books, but a book can only be written by one author.

The diagram also shows the abbreviations for primary key and foreign key which are important concepts in database as each of them represent and identity to the database. A primary key (PK) is a unique identifier for an entity in a database table as it uniquely identifies each record in the table so it cannot contain duplicate or null values. In the ER diagram above the PK is denoted by the bold letters beside the attribute as well as the underlining of the attributes. A foreign key (FK) on the other hand is a reference to the primary key in another table and it establishes a relationship between two entities by linking them through a common attribute so because of that, it can contain duplicate or null values. In the ER diagram above the FK is denoted by the letters FK beside the attributes (Mitty, 2014).

- **Database performance improvement in accessing and searching records**

Choosing primary keys is one of the performance factors. The presence of primary key prevents the duplication of data in a database table. It is important to pick the right column to be the primary key.



PVLDB4_11	Proceedings of the VLDB Endowment
PVLDB4_8	Proceedings of the VLDB Endowment

Figure 8: *Same entries with different primary key*

The figure above shows a problem with the wrongly chosen primary key candidate. There is duplication of the same publication name due to pubid being the primary key. It is recommended to use integer with auto increment as primary key. Auto increment provides each row with a unique generated primary key. Love (2023) states that “they enable efficient range scans, sorting, and joining operations”.

Searching data in the current database tables might be within an instance. However, as data increases over time, the time taken to search for a row based on rules will take some time. To combat such a thing, applying index to database table is recommended. Cioloca and Georgescu (2011) mentioned that data in a table is sorted in a logical manner after being copied by indexing. Index creates a new data structure that contains the copied primary key and pointer address. Searching with index prevents searching every row instead points towards the memory address of the desired data.

Database performance can also be improved by optimizing query techniques. It can be achieved by following best database querying practices that were already cemented through community trial and errors. This includes minimizing the use of wildcard characters, avoiding subquery usages, using row limiters such as LIMIT or TOP, using EXISTS instead of IN, and using stored procedures (Sarang, 2023), as doing the opposite may cause inefficient or unnecessary CPU utilization. To gauge query performances, developers can use SQL EXPLAIN statements to analyze a query’s execution plan (MySQL, n.d.).

Finally, a periodical data defragmentation maintenance should be enacted to achieve optimal database performance. The process of data undergoing constant rewriting and removal from a storage device will inevitably cause its data to become fragmented in the storage space, which bottlenecks the data retrieval process (Buchanan Technologies, n.d.). Data defragmentation helps by rearranging data on the storage medium and group relevant data together, eliminating holes between the data storage space (Yasar, 2023). Ultimately, this allows I/O related operations to be more responsive and efficient, including database processes (Buchanan Technologies, n.d.).

Personal review

- **Lim Zhe Yuan**

Reflecting over the group assignment, I have learnt about a few database programming techniques aside from simple querying using SQL, such as utilizing stored procedures, structuring tables, and optimizing query processes. Specifically, the most rewarding knowledge I have gained was creating stored procedures, as I could group logical SQL operations together and execute them on command to fulfill business cases. I have also gained the insight of analyzing query execution plans and find potential performance bottlenecks to eliminate unnecessary or deprecated computational use that are cost ineffective.

I made small achievements as well over the course of the assignment, such as successfully managing to find the correct command syntaxes through online means while writing normal queries and stored procedures, even though I am not fluent in the SQL language. I have also managed to logically infer relationships between the relations for the purpose of understanding and completing the assignment questions. There were less database design or implementation problems as a logical, relevant database structure have been systematically planned prior to any queries and operations, which we were able to achieve with the guidance and advice of the lecturer.

Teamwork was also an important contributor to the successful completion of the assignment. In terms of team contribution, I have made significant impact to the group as I was the main coordinator who managed work distributions and motivated other group members to progress on their work steadily. I actively communicated with my group members about assignment requirements and returned feedback whenever possible in order to ensure that each member put in the required effort and quality into their assigned tasks. Aside from delegating tasks for other group members, I have also done my part in the assignment questions spectacularly by validating my code out for system and logical errors and ensuring that the produced results are correct and as expected.

My group members have been a huge part in making the assignment bearable and easy. For example, one of my group members, Otheya, also actively communicated with other group members about assignment requirements to keep themselves on the right track. He was assertive and initiated the process of task choosing for the assignment. He asked for feedback once he completes any of his part in the assignment to ensure that his work is valid and does not contain errors. He did his work responsibly and was able to complete them fully within a set deadline even though he has multiple other subject assignments to attend to, which shows his level of discipline.

Much like Otheya, Keat Loon was also an asset to the group assignment. He displayed the same level of responsibility as Otheya and is also well aware of the assignment requirements. His awareness is commendable as he is able to correctly identify and rectify some illogical assignment work approaches that we planned to follow. Besides, he is capable of working on his tasks independently with little intervention once he briefly understands their requirements at the start.

All in all, the group I am in is all-rounded and balanced, which makes us equipped to complete the assignment within the lecturer’s expectations with little obstacles.

- **Otheya Kumar**

Completing Assignment 2 group work has taught me various new knowledge in database programming that I did not possess when I learned the fundamentals of database during my diploma studies. Since I did possess a basic grip on SQL queries from my diploma times, I did find it easier to understand the simple queries to get results from database but during this assignment I have learned about stored procedure and its use cases which is all new to me as I have never used it previously. Other than that, I have also learned some new SQL tools that are used for transaction in stored procedures as well as getting results that have same attributes.

There were some problems that I faced when completing this assignment and one of them is the stored procedures. This was expected for me as I have just learned how to create stored procedures and was very new and inexperienced in using the stored procedure tools such as transaction. I was finally able to overcome this challenge by grasping the technique as well as the way to build a stored procedure thanks to my lecturer who held a special session for SQL learning. This session not only taught about stored procedures but also went through the basic of SQL query so that I was able to complete the other task with normal SQL query with ease as it was very long ago that I refreshed my SQL language.

Another challenge, albeit slightly less of a problem that I have faced during this assignment is the use of advanced SQL tools such as the various join tools as well as the union tool. Although I have learned to use these tools back in my diploma, I have actually never used them in my previous assignments, so I was not really familiar with the use of them as well as their differences compared to normal join tool. Thanks to the special session I understood the difference as well as get many examples of the use cases of the tools. My teammates also helped me a lot when my task had weird results by identifying and fixing the join tool where the mistake was.

As this is a group assignment, teamwork is a vital part of the assignment as we will have to work together to complete these tasks as well as help each other with solving the problems they might face. My experience in working with my teammates in this assignment was really pleasant and insightful as I was able to choose and complete the task that I chose as well as having great communication with teammates to solve problems. My contribution to this group assignment was task 1 as well as helping my teammates task 3 in creating an entity relationship diagram of the final database structure.

My teammates Zhe Yuan and Saw have also contributed as much as me if not more. Zhe Yuan was also acting as a leader, so he oversees us with our task and validates the results as well as contributing to task 2 and 3. Saw on the other hand contributed to task 2 and task 3 as well while also making sure that our assignment requirements are met. He also volunteered to help each of us when we were facing any problems and guide us to the correct solution to the problem. Overall, this group assignment went smoothly with no hitches thanks to the skilled and smart teammates I had while also being a good learning experience from them.

- **Saw Keat Loon**

Upon completing assignment 2, the course has allowed me to learn new things about database programming. As I have learned database programming during my diploma time, learning it back again brings back some memories. Some knowledge that is learned from a diploma is used when completing the assignment. However, the need of doing stored procedure is something new to me. Overall, the course further strengthens my knowledge and skills in writing SQL queries and proper way of designing databases. Creating procedure is something interesting because we could have multiple SQL queries within a stored procedure. Hence, there is no need to call multiple SQL queries instead just call the stored procedure once.

Some of the problem that I have faced when completing my assignment is learning back SQL syntax. It has been a long time since I have touched SQL upon finishing my diploma. Besides that, learning other programming languages such as Python and Kotlin changes the way I code stuff. Coming back to SQL brings some challenges because the way we type SQL and Python is different as SQL is much stricter while Python is more forgiving. The way I overcome this challenge is to dig back out my old SQL work and study them back. Doing some research on the Internet also helps me in figuring out solutions to errors and additional functions.

The second challenge that I faced is creating stored procedures. Stored procedure enables multiple SQL queries to be done within a command. However, learning how to write it brings me some issues. As the number of tables increases, the process of picking which table to join and pull data brings some challenges as I have not dealt with more than 4 database tables before. Besides that, creating error handling queries in the stored procedure also poses some challenges to me as the syntax of writing is something completely new to me. Errors popped out as I was trying to code queries. The way I solve such challenges is to do more research on the Internet and see examples. Examples help me the most as I am able to copy and modify their codes to suit my needs. Modifying codes enables me to learn coding stored procedure much faster due to it involving me to physically type it.

Teamwork is an essential component when working with other people. Teamwork enables faster work done and communication is key. My overall experience working with my current team has been pleasant and fast. The teammates I have known for 3 semesters, thus our chemistry is quite good to begin with. The overall group dynamics have been good as we are able to understand one another and so far, have no communication issues. Besides that, splitting up the work is easily done as we weigh each of the tasks and decide who is the best to deal with it.

Some of my contribution towards the group assignment is by dividing the task. Initially the task is taken based on a first come first serve basis. However, I have noticed that task 2 is quite hard and there are a lot of things that have to be done. I assisted my friend by taking another half of the task and he did it back to the task that I take. I keep track of the time and initiated the work on the assignment. Keeping track of the time enables us to know how slow we are or how long do we have left as we need to complete other assignments.

Besides me, Otheya and Zhe Yuan also contributed some into the group assignment. Each of them volunteered by taking tasks individually. They also help in validating our answers by checking the results and recommended some improvements such as error checking queries. Getting ideas from them also enabled me to figure out ways to solve my task as there were times where I could not figure out the answer. Zhe Yuan leads us from the start till the end as he is the smartest while Otheya keeps track on the progress to make sure that we are not behind schedule.

References

Buchanan Technologies (n.d.) How to Optimize Database Performance: Everything You Need to Know. Available from https://www.buchanan.com/improve-database-performance/ [accessed 7 November 2023].

Cioloca, C., & GeorgeScu, M. (2011) Increasing Database Performance using Indexes. *Database Systems Journal*, *2*, pp. 13–22. Available from https://www.dbjournal.ro/archive/4/2_Ciologa_Georgescu.pdf [accessed 7 November 2023].

Love, C. (2023) *Guid vs. Sequential Integers: A Great Debate in Database Design*. Available from https://www.c-sharpcorner.com/article/guid-vs-sequential-integers-a-great-debate-in-database-design/ [accessed 7 November 2023].

Mitty, W. (2014) How to represent foreign key in an ER diagram?, Database Administrators Stack Exchange. Available at: https://dba.stackexchange.com/questions/76532/how-to-represent-foreign-key-in-an-er-diagram (Accessed: 17 November 2023).

MySQL (n.d.) *EXPLAIN Statement*. Available from https://dev.mysql.com/doc/refman/8.0/en/explain.html [accessed 7 November 2023].

Sarang, S. (n.d.) *12 Tips for Optimizing SQL Queries for Faster Performance*. Medium. Available from https://medium.com/learning-sql/12-tips-for-optimizing-sql-queries-for-faster-performance-8c6c092d7af1 [accessed 7 November 2023].

Yasar, K. (2023) *defragmentation*. TechTarget. Available from https://www.techtarget.com/searchwindowsserver/definition/defragmentation#:~:text=Defragmentation%2C%20also%20known%20as%20defragging,or%20laptop%27s%20performance%20and%20speed [accessed 7 November 2023].

Group members work contribution form

In percentage, please indicate the work contribution of each member. This should be agreed by all group members**. The total of all members work must add to 100%**
You must **submit this form in your final report.** Put your initials in the signature columns. This copy must be signed by all members.

Group/Team Name: ___Zhe Yuan Group_________________________

Team member name	Student ID	Individual overall work contribution (%)	Signature
Student: **Lim Zhe Yuan**	0204677	34	Zhe Yuan
Student: **Otheya Kumar**	0203640	33	Otheya Kumar
Student: **Saw Keat Loon**	0207778	33	Saw
		Total 100%	

Appendix

- **Task 1 source code**

```
DELIMITER //
CREATE PROCEDURE print_publication(IN authorName VARCHAR(22))
BEGIN
    DECLARE results INT;
    DECLARE exit handler for SQLEXCEPTION
    BEGIN
        GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
        @errno = MYSQL_ERRNO, @text = MESSAGE_TEXT;
        SET @full_error = CONCAT("ERROR ", @errno, " (", @sqlstate, "): ",
@text);
        SELECT @full_error;
    END;

    SELECT COUNT(*) INTO results
    FROM author a
    LEFT JOIN wrote w ON (a.aid = w.aid)
    LEFT JOIN publication p ON (w.pubid = p.pubid)
    WHERE a.name = authorName;

    IF results = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No publications found for the specified author';

    ELSE
        SELECT p.pubid AS 'Publication ID', p.title AS 'Publication Title',
b.`year` AS 'Publication Year'
        FROM author a
        LEFT JOIN wrote w ON (a.aid = w.aid)
        LEFT JOIN publication p ON (w.pubid = p.pubid)
        INNER JOIN book b ON (b.pubid = p.pubid)
        WHERE a.name = authorName

        UNION

        SELECT p.pubid AS 'Publication ID', p.title AS 'Publication Title',
COALESCE(pc.`year`, jc.`year`) AS 'Publication Year'
        FROM author a
        LEFT JOIN wrote w ON (a.aid = w.aid)
        LEFT JOIN publication p ON (w.pubid = p.pubid)
        INNER JOIN article aa ON (aa.pubid = p.pubid)
        LEFT JOIN proceedings pc ON (pc.pubid = aa.appearsin)
        LEFT JOIN journal jc ON (jc.pubid = aa.appearsin)
        WHERE a.name = authorName

        ORDER BY 'Publication Year' ASC;
    END IF;
END //
DELIMITER ;
```

- **Task 2a source code**

```
DROP PROCEDURE IF EXISTS merge_publication;
DELIMITER //
CREATE PROCEDURE merge_publication()
BEGIN
    DECLARE exit handler for SQLEXCEPTION
    BEGIN
        GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
        @errno = MYSQL_ERRNO, @text = MESSAGE_TEXT;
        SET @full_error = CONCAT("ERROR ", @errno, " (", @sqlstate, "): ",
@text);
        SELECT @full_error;
    END;

    DROP TABLE IF EXISTS publication_master;
    CREATE TABLE publication_master (
        pub_id VARCHAR(10),
        type VARCHAR(20) NOT NULL,
        year INT(11) NOT NULL,
        title VARCHAR(70),
        author VARCHAR(500),
        publisher VARCHAR(50),
        volume INT(11),
        num INT(11),
        appears_in VARCHAR(10),
        start_page INT(11),
        end_page INT(11),
        CONSTRAINT pk_publication_master PRIMARY KEY (pub_id)
    );

    DROP TABLE IF EXISTS all_result;
    CREATE TEMPORARY TABLE all_result (
        records INT,
        record_type char(70)
    );

    /* Insert proceedings data */
    INSERT INTO publication_master(`pub_id`, `type`, `title`, `year`)
    SELECT proceedings.pubid AS pubid, 'proceedings' AS `type`, title, year
    FROM proceedings
    INNER JOIN publication ON (proceedings.pubid = publication.pubid);

    INSERT INTO all_result(records, record_type)
    VALUES(ROW_COUNT(), 'proceedings');

    /* Insert journal data */
    INSERT INTO publication_master(`pub_id`, `type`, `title`, `year`,
`volume`, `num`)
    SELECT journal.pubid AS pubid, 'journal' AS `type`, title, `year`, volume,
num
    FROM journal
    INNER JOIN publication ON (journal.pubid = publication.pubid);

    INSERT INTO all_result(records, record_type)
    VALUES(ROW_COUNT(), 'journal');

    /* Insert book data */
    INSERT INTO publication_master(`pub_id`, `type`, `year`, `title`,
`publisher`, `author`)
    SELECT book.pubid AS pubid, 'book' AS `type`, `year`, title, publisher,
GROUP_CONCAT(REPLACE(author.name, '_', ' ') ORDER BY aorder SEPARATOR ", ") AS
author_line
    FROM book
    INNER JOIN publication ON (book.pubid = publication.pubid)
    LEFT JOIN wrote ON (book.pubid = wrote.pubid)
    LEFT JOIN author ON (wrote.aid = author.aid)
    GROUP BY pubid;

    INSERT INTO all_result(records, record_type)
    VALUES(ROW_COUNT(), 'book');

    /* Insert article data */
    INSERT INTO publication_master(`pub_id`, `type`, `title`, `year`,
`appears_in`, `start_page`, `end_page`, `author`)
    SELECT article.pubid AS pubid, 'article' AS `type`, title,
COALESCE(proceedings.`year`, journal.`year`) AS year, article.appearsin,
startpage, endpage,
    GROUP_CONCAT(REPLACE(author.name, '_', ' ') ORDER BY aorder SEPARATOR ",
") AS author_line
    FROM article
    LEFT JOIN journal ON (appearsin = journal.pubid)
    LEFT JOIN proceedings ON (appearsin = proceedings.pubid)
    INNER JOIN publication ON (article.pubid = publication.pubid)
    LEFT JOIN wrote ON (article.pubid = wrote.pubid)
    LEFT JOIN author ON (wrote.aid = author.aid)
    GROUP BY pubid;

    INSERT INTO all_result(records, record_type)
    VALUES(ROW_COUNT(), 'article');

    /* Insert info for sum of all results */
    INSERT INTO all_result(records, record_type)
    SELECT SUM(records), 'all' AS `type` FROM all_result;
    SELECT * FROM all_result;

END//

DELIMITER ;

CALL merge_publication();
```

- **Task 2b source code**

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `print_article`(IN `pub_id`
VARCHAR(255))
BEGIN
    DECLARE results INT;
    DECLARE exit handler for SQLEXCEPTION
    BEGIN
        GET DIAGNOSTICS CONDITION 1 @sqlstate = RETURNED_SQLSTATE,
        @errno = MYSQL_ERRNO, @text = MESSAGE_TEXT;
        SET @full_error = CONCAT("ERROR ", @errno, " (", @sqlstate, "): ",
@text);
        SELECT @full_error;
    END;

    SELECT COUNT(publication_master.type) INTO results FROM publication_master
WHERE       publication_master.appears_in = pub_id AND publication_master.type
= "article";

    IF results =0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Publication Id do not have any article in it.';
    ELSE
        SELECT * FROM publication_master WHERE appears_in = pub_id AND type
="article" ORDER BY start_page ASC;

    END IF;
 END$$
DELIMITER ;
```

CDB3033N DATABSE PROGRAMMING

MARKING RUBRIC

ASSIGNMENT [2]

STORED PROCEDURE, CURSOR, & ACCESS METHODS

Section (1)-80%

LEARNING OUTCOME	MARKING CRITERIA	SCALE							
		Fail (0-49)	3rd Class (50-59)	2nd Lower Class (60-69)	2nd Upper Class (70-79)	1st Class (80-100)	YOUR MARKS/COMMENTS		
							100%	Weightage	Actual Marks
CLO3	**Task 1 (40%)** • **Store procedure implementation (15%)** • **Cursors Implementation (15%)** • **Correct output (10%)**	Little or no attempt to implement the feature correctly	A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors.	Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented.	A mostly complete implementation of the feature which works correctly, although the coding could be clearer.	A complete implementation of the feature, clearly coded.		0.4	
	Task 2a (25%) • **Correct table creation (5%)** • **Insertion procedure with business rules (20%)**	Little or no attempt to implement the feature correctly	A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors.	Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented.	A mostly complete implementation of the feature which works correctly, although the coding could be clearer.	A complete implementation of the feature, clearly coded.		0.25	
	Task 2b (15%) • **Display procedure (10%)** • **Correct output (5%)**	Little or no attempt to implement the feature correctly	A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors.	Appropriate implementation of the features but with some minor flaws and some aspects are not well implemented.	A mostly complete implementation of the feature which works correctly, although the coding could be clearer.	A complete implementation of the feature, clearly coded.		0.15	
						Total (80%)			

CDB3033N DATABSE PROGRAMMING

MARKING RUBRIC

ASSIGNMENT [2]

STORED PROCEDURE, CURSOR, & ACCESS METHODS

Section (1)-80%									
LEARNING OUTCOME	MARKING CRITERIA	SCALE							
		Fail (0-49)	3rd Class (50-59)	2nd Lower Class (60-69)	2nd Upper Class (70-79)	1st Class (80-100)	YOUR MARKS/COMMENTS		
							100%	Weightage	Actual Marks
CLO4	**Task 3 (20%)** • **Physical database review (10%)** • **Improvement (10%)**	A poor piece of documentation has been produced. The coverage is unclear, and/or there are significant omissions	A fair piece of documentation has been produced, although there are some weaknesses—either the coverage is not particularly clear, or some aspects have been omitted	An appropriate piece of documentation has been produced but with minor flaws or some aspects are not well elaborated.	A good piece of documentation has been produced, providing detailed and clear coverage of the aspect concerned, although there may be a number of minor flaws which prevent it being regarded as excellent	An excellent piece of documentation has been produced, providing full and clear coverage of the aspect concerned			
Total (20%)									
Overall Score (100%)									