|  |  |
| --- | --- |
| **Databases**  Diploma in IT / FI / CSF  Year 1 (2019/20) Semester 2 | Week **1** |
| **2** hours |
| **PRACTICAL 1**  **SELECT (Part 1)** | |

**OBJECTIVES**

At the end of this practical, you should know how to:

* construct a simple database query using the SELECT statement

**REFERENCES**

Please refer to the following:

* Installation Guide – MS SQLServer 2017 Express
* Appendix B in Database textbook: Tables in *NP40 Book Rental System’s* Database
* Appendix E in Database textbook: Data Dictionary for *NP40 Book Rental System*
* Database Textbook: pages 2-5 to 2-10 and 2-20 to 2-22
* PolyMall: Database Systems – [Topic 0 Case study](https://polymall.polytechnic.edu.sg/webapps/blackboard/content/listContentEditable.jsp?content_id=_30277_1&course_id=_813_1&mode=reset&courseTocLabel=Topic+0+Case+study)
* PolyMall: Database Systems – Topic 1 Basic Select

[1.1 Introduction to Basic SQL](https://polymall.polytechnic.edu.sg/webapps/scor-scormengine-BB5784d4c32fccb/delivery?action=launchPackage&course_id=_813_1&content_id=_30610_1)

[1.2 Introduction to Basic SELECT – Part 1](https://polymall.polytechnic.edu.sg/webapps/scor-scormengine-BB5784d4c32fccb/delivery?action=launchPackage&course_id=_813_1&content_id=_30611_1)

[1.3 Introduction to Basic SELECT – Part 2](https://polymall.polytechnic.edu.sg/webapps/scor-scormengine-BB5784d4c32fccb/delivery?action=launchPackage&course_id=_813_1&content_id=_30612_1)

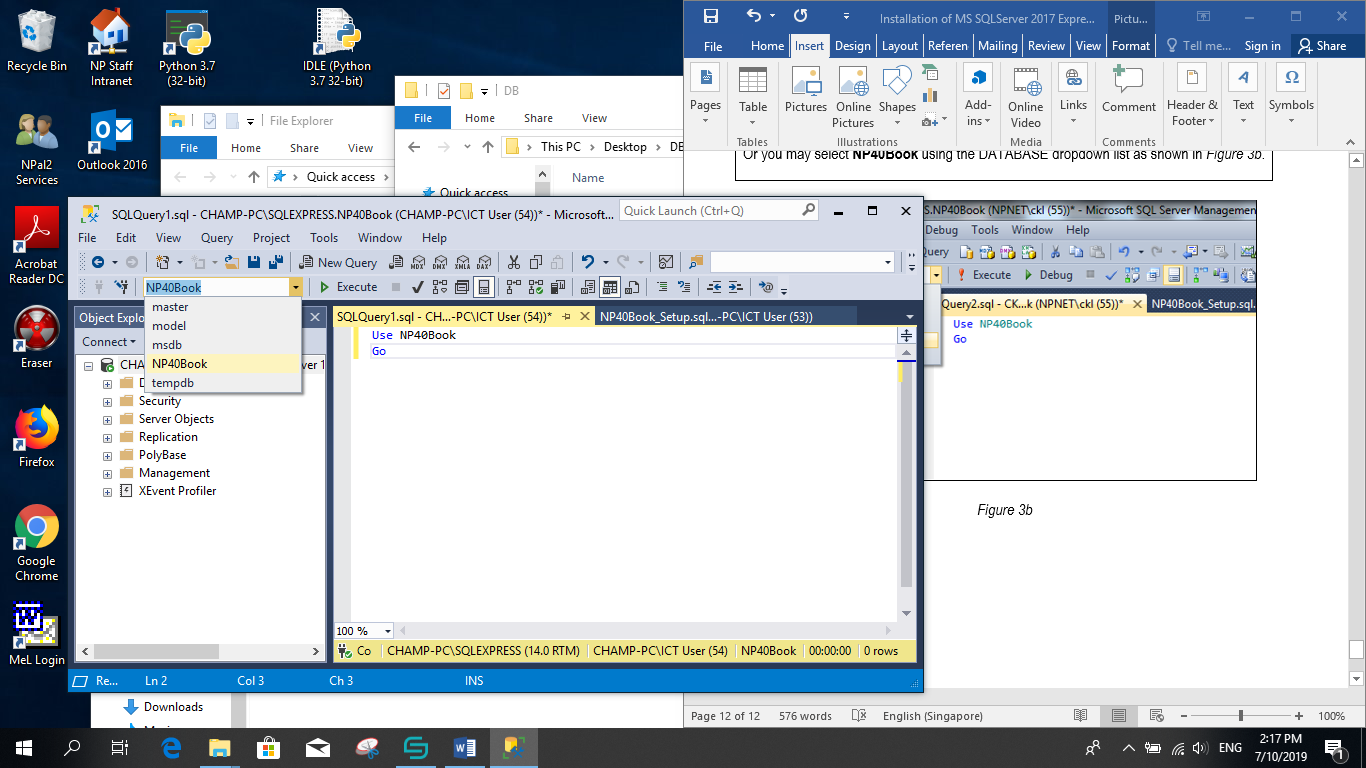
Before your start your practical. Refer to the Installation Duide – MS SQLServer 2017 Express to learn and perform the following:

**A. STARTING Microsoft SQL Server Management Studio Express**

**B. Create Databases**

**C. Starting a New Query**

|  |
| --- |
| **IMPORTANT: For Practical 1 to 6** You should only **create** the NP40Book database **once**.    After you have connected successfully to the database server via SQL Server Management Studio Express, you should either execute the following commands before you issue any other SQL statements:  **use NP40Book**  Or you may select **NP40Book** using the DATABASE dropdown list as shown in *Figure 1*. |



*Figure 1*

**D. QUESTIONS**

With reference to Appendix E: Data Dictionary for *NP40Book Rental System*, write the appropriate **SQL statements** for the following queries. In this practical, you will construct your retrieval query (**SELECT statement**) on a **single table** only.

Syntax:

**SELECT [ ALL | DISTINCT ] { \***

**| { table\_name | table\_alias }.\***

**| { column\_name | express } [ [ AS ] column\_alias ]**

**| column\_alias = expression**

**} [ , … n ]**

**FROM table\_name [ [ AS ] table\_alias ] [ , … n ]**

**[ WHERE search\_condition ]**

**[ ORDER BY { order\_expression [ ASC | DESC ] } [ , … n ] ]**

**Brief note on convention**

An **SQL statement** consists of reserved words and user-defined words. **Reserved words** are a fixed part of the SQL language, and have a fixed meaning. They must be spelt ***exactly*** as required and cannot be split across lines. **User-defined words** are made up by the user and represent names of various database objects such as table name, column name, etc.

We will use the following convention to define SQL statements:

* Words in **UPPERCASE** letters are used to represent reserved words and must be spelt exactly as shown;
* Words in **lowercase** or **TitleCase** letters are used to represent user-defined words;
* A vertical bar ( **|** ) separating syntax items within brackets or braces means that you can choose only one of the items. For example, AVG | COUNT | MAX | MIN | SUM;
* Curly braces **{ }** indicate a required syntax item. Do not type the braces;
* Square brackers **[ ]** indicate an optional syntax item. Do not type the brackets;
* **[ , … n]** indicates that the preceding item can be repeated **n** number of times. The occurrences are separated by commas;
* **[ … n]** indicates that the preceding item can be repeated **n** number of times. The occurrences are separated by blanks.

**Comments** are placed in your SQL statements in two ways:

* **Block comment** makes use of /\* (forward slash and asterisk) at the beginning of a comment and \*/ (asterisk and forward slash) to close the comment;
* **Inline comment** makes use of - - (dash dash) to begin and end the comment.

**Literals** are constants used in SQL statements. There are different forms of literals for every data types supported by SQL. However, for simplicity, we can distinguish between literals that are enclosed in single quotes and those that are not.

* All **non-numeric** data values **must** be enclosed in single quotes (e.g. ‘Gabriel Tan’);
* All **numeric** data must **not** be enclosed in single quotes (e.g. 25).

**List all rows, all columns and selected columns**

1. List every detail of all staff.

Which table in the NP40Book database contains details of all staff?

SELECT \* FROM Staff

1. Now list every detail of all books.

Again, which table contains details all books?

SELECT \* FROM Book

1. List StaffID, Name and Gender of all staff.

Please note that you are not expected to retrieve every detail of all staff.

SELECT StaffID, Name, Gender FROM Staff

1. Now list ISBN, Title, PublisherID and BookCat of all books.

Again, you are not expected to retrieve every details of all books.

SELECT ISBN, Title, PublisherID, BookCat FROM Book

# **Remove Duplicate Rows in Results**

1. List the SupervisorID of all the staff.

Notice that certain SupervisorID is repeated in the results? This is because there are staff that are supervised by the same supervisor.

SELECT SupervisorID FROM Staff

1. Now remove the duplicate SupervisorID from the results generated for the previous instruction.

**Hint: use the DISTINCT keyword**

SELECT DISTINCT SupervisorID FROM Staff

1. List the BranchNo of all the members.

SELECT BranchNo FROM Staff

1. Now remove the duplicate BranchNo from the results generated for the previous instruction.

SELECT DISTINCT BranchNo FROM Staff

# **Calculated Column & Define New Column Heading**

1. NP40Book is considering reducing the current rental rate for all books by 2%. Display the current RentalRate and new RentalRate for each copy of books as follows :

ISBN CopyNo RentalRate New RentalRate

….. …… …………. ………….

**Hint: find out from Reference A on how to create calculated column and define new column heading**

You will notice that there is no change to the values of RentalRate in the BookCopy table after the query is executed. You may verify this by executing a SELECT query statement or Open Table from the Object Explorer.

SELECT ISBN, CopyNo, RentalRate, (RentalRate\*1.02)as New\_RentalRate FROM BookCopy

1. NP40Book is considering increasing the current Salary of all staff by 10%. Display the current Salary and the proposed New Salary for each staff as follows :
   1. StaffID Name Salary New Salary
   2. ….. …… …………. ………….

# SELECT StaffID, Name, Salary, (Salary\*1.1) as New\_salary FROM Staff

# **Sorting Results**

1. List every detail of all staff in ascending order of Name.

**Hint: use the ORDER BY keyword**

You will notice that by default the results are sorted in ascending order of Name when using the ORDER BY keyword in your query.

SELECT \* FROM Staff

ORDER BY Name ASC

1. List every detail of all staff in descending order of Name.

Find out the keywords to generate results in ascending or descending order explicitly.

SELECT \* FROM Staff

ORDER BY Name DESC

1. List StaffID and ContactNo of all staff in ascending order of StaffID.

**Hint: you have to find the database table that has both StaffID and ContactNo**

SELECT StaffID, ContactNo FROM StaffContact

ORDER BY StaffID ASC

1. List Name and Salary of all staff with the highest Salary first.

SELECT Name, Salary FROM Staff

ORDER BY Salary DESC

1. List every detail of all books, showing the most recently acquired copies first.
2. SELECT \* FROM Book
3. ORDER BY YearPublish DESC
4. List BranchNo, Name, and Salary of all staff in ascending order of BranchNo.

Staff that work in the same branch are to be placed together and displayed in ascending order of Name.

SELECT BranchNo, Name, Salary FROM Staff

ORDER BY BranchNo ASC