





| **LEARNER** | Kalon Markides |
| --- | --- |
| **TRAINER** | Craig Reeves |
| **DATE** | 13/01/2022 |

| **ACTIVITY 1: DBMS** |
| --- |
| Research on different DBMS applications that can be used to manage relational databases.  Include information such as:   * Identify at least 2 DBMS * Are there any special features of these DBMS? * How does it compare to the other DBMS? * Do you have a preferred DBMS – and why?   There are many DBMS such as MySQL, MS Access, PostgreSQL and Oracle to name a few. Many relational database management systems use SQL for querying and maintaining the database, RDBMS is an advanced form of a DBMS. DBMS stores data as a file where RDBMS stores data in tabular form. Some features of DBMS are: Minimum redundancy and duplication , Reduced amount of space and money spent on storage, Data organisation, Customisation of the database , Data retrieval, Usage of query languages, multi user access, Maintained data integrity, Data durability , Enhanced file uniformity and A high level of data security.  MySQL is an open source RDBMS which means that it is free to use under the terms of the GNU general public licence. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL supports horizontal partitioning which is dividing a larger database into multiple smaller parts. MySQL allows data replication which is the electronic copying of data from a database in one computer or server to a database in another so that all users share the same level of information. MySQL has different types of security and can be configured with SSL support.  MS Access can run as DBMS or RDBMS. It allows a choice of different database engines besides the default Jet Database Engine such as SQL Server, Oracle, DB2 or find support for the Open Database Connectivity. MS Access is a paid for software. MS Access does not support partitioning. MS Access does not support data replication.MS Access only has username/password support for security. MS Access is easier to navigate as well as being able to create ERD’s. MS Access also works with standard SQL syntax and scripts meaning a new language does not have to be learnt. |

| **ACTIVITY 2: NORMALISATION** |
| --- |
| Using the Normalisation Excel Spreadsheet attempt to put this information in third normal form. |

| **ACTIVITY 3: RETRIEVING DATA** |
| --- |
| In the example database, retrieve the following information. Screenshot the SQL query used and the results obtained.  **Obtain all information on the Students not attending course 1**    **Obtain the first name, surname and Date of Birth for the student with the email address: val.bolger@example.com**    **Obtain a list of the modules which have the subject Economics**    **Obtain a list of class numbers and their dates which are scheduled before 21st September 2020** |
|  |

| **ACTIVITY 4: INSERTING DATA** |
| --- |
| In the example database, write inserts to insert the following information. Screenshot the SQL query and the results obtained.  **Insert a record for a new course named Deep-Space Radar Telemetry**    **Write an insert to insert records for the following modules:**  String Theory  Exotic Matter  Harnessing the Einstein-Rosen Bridge  Supercollision and miniature Black Holes  (these modules are worth 20 credits each, at level 6 and are taught on the Quantum Physics Course)    **Using the information from the previous example and the LecturerID of 6, create a class for each new module.**    **Using the 3rd Normal Form Student Data from Activity 2, write an insert to enter this student data into the Student Table of the Example Database** |









| **LEARNER** | Kalon Markides |
| --- | --- |
| **TRAINER** | Craig Reeves |
| **DATE** | 17/01/2022 |

| **ACTIVITY 1: CREATING CALCULATIONS** |
| --- |
| In the existing database from last week:  **Count how many students attend our university**    **Calculate the sum of full time fees for every full-time course**    **Identify the cost of the least and most expensive course**    **Calculate the average cost of all part time courses**    **Calculate the fee of each full time course after applying (subtracting) the scholarship discount**    Extension:  **Select only the course number of the cheapest full-time course**      **Find cost of the most expensive course after applying the scholarship discount**    **(Not sure why displaying 2 results )**  **Count the number of applications for History courses made between 01/03/2020 and 30/08/2020** |

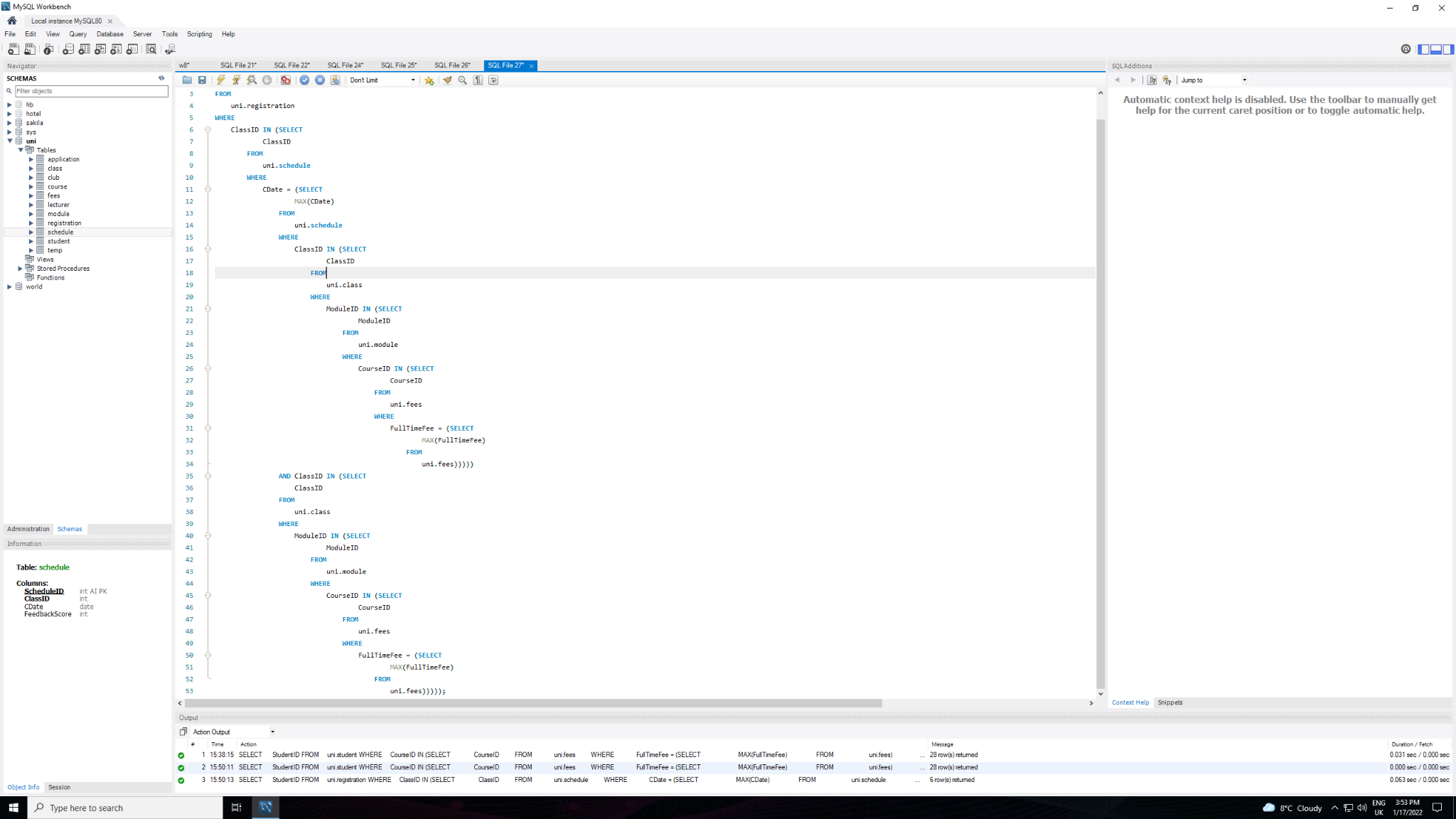
| **ACTIVITY 2: DATABASE FUNCTIONS** |
| --- |
| Combine what you have learned about SQL functions to write solutions for the following problems:  **Write a select statement to obtain all of the student information for successful applications made for Course 11 which do not relate to current students**    **Modify the select statement from the previous example into an insert statement and insert the data into the student table**      **Write a select statement to obtain all the information for the unsuccessful applications made for Course 1**    **Modify the select statement from the previous example into a delete statement and delete the unsuccessful Course 1 applications**    **Write a select statement to identify the unsuccessful applications for course 11 made after 01/08/2020**    **Using the select statement from the previous example, modify it into an update statement and update the applications to successful**    **Roll back the previous update**    **Modify the previous update to include applications for course 11 made after 01/09/2020**    **Commit the Update** |

S

| **EXTENSION: CREATE A TABLE** |
| --- |
| Use the create table functions in the previous example to create an application archive table for retaining information about successful applications. Decide on what your primary key is, as well as whatever foreign keys and data types you may need.  HINT: Always remember to write your code into a select statement before writing the insert for your data query, this means you can be sure that you are inserting the correct data BEFORE you insert it. |

| **ACTIVITY 3: USING SUBQUERIES** |
| --- |
| Using Lists or subqueries  **Obtain all the course information for courses with the CourseIDs of 1, 3, 5 and 7**    **Obtain a list of all modules taught on courses which have a Full Time Fee greater than 9000**    **Obtain a list of classes for modules taught on courses which have a Full Time Fee greater than 9000**    Extension  **Find a list of studentIDs for the latest class on the most expensive course** |

Correct Answer



| **ACTIVITY 4: INNER JOIN** |
| --- |
| **Obtain a list of Students and the name of the Courses they are studying**    **Obtain a list of course names, full time fees and part time fees for each course**    **Obtain a list of classIDs for the Economics Course and the modules they relate to** |



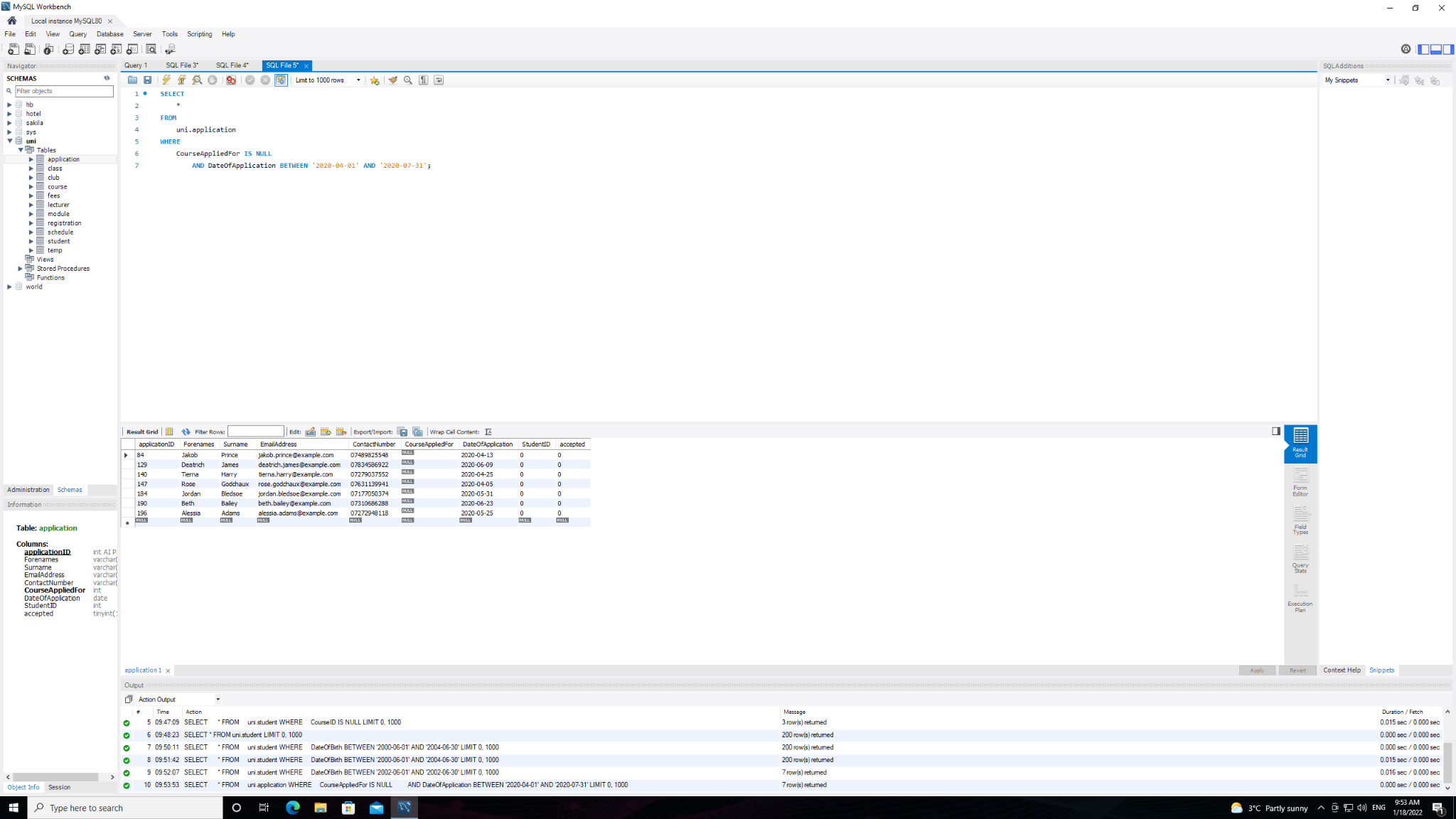




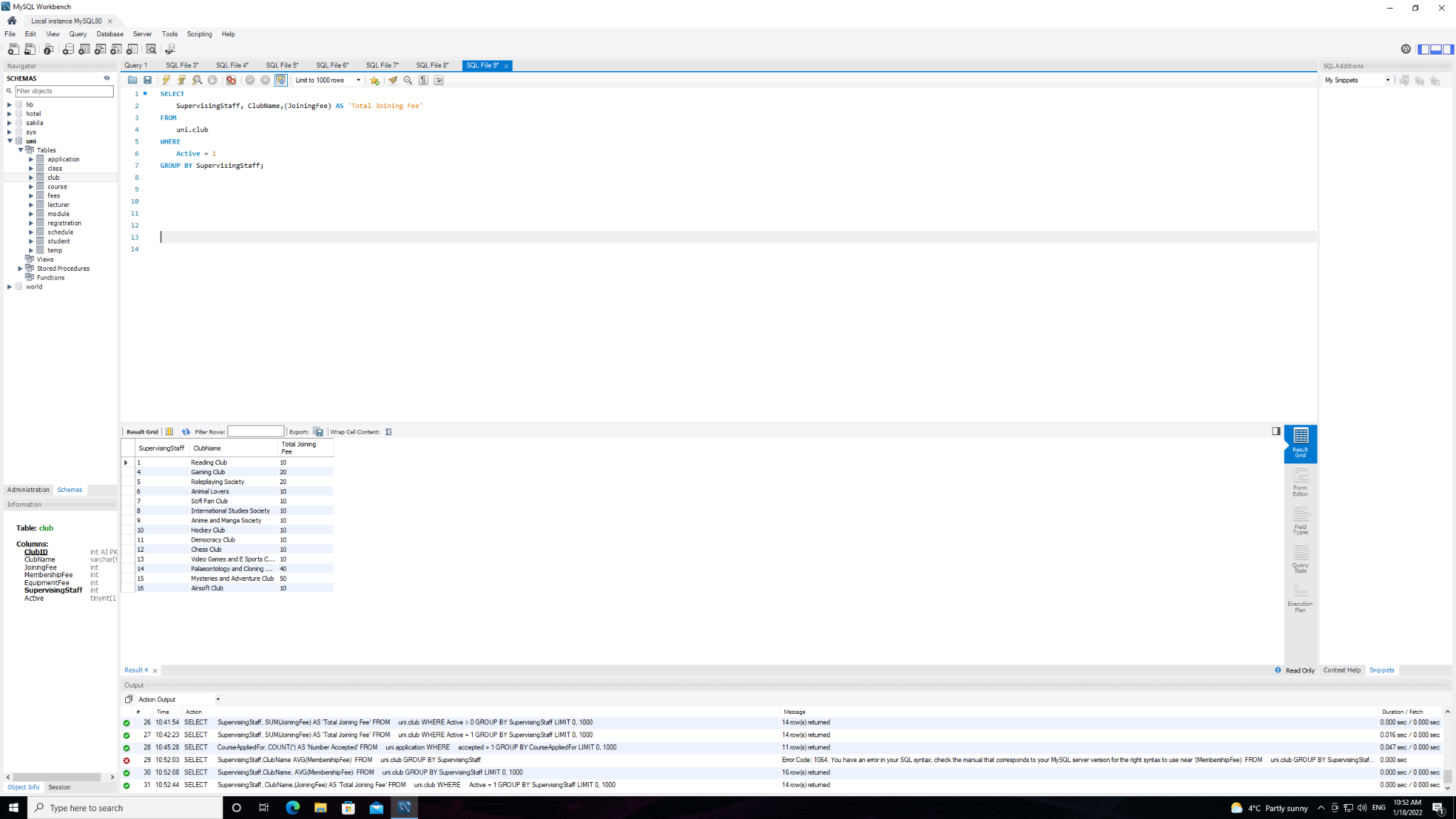


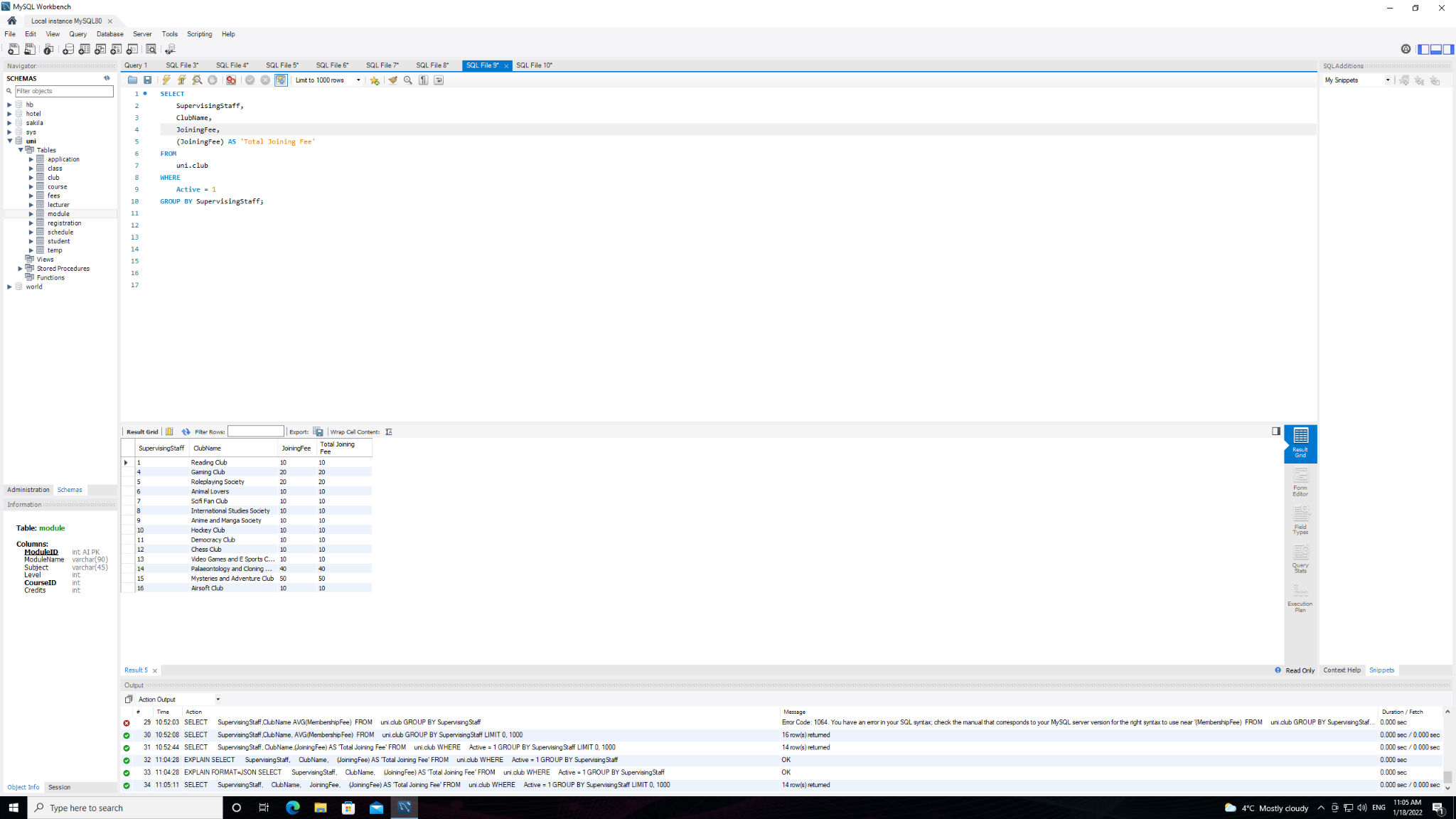
| **LEARNER** | Kalon Markides |
| --- | --- |
| **TRAINER** | Craig Reeves |
| **DATE** | 18/01/2022 |

| **ACTIVITY 1: BETWEEN TWO NULLS** |
| --- |
| Using SQL, screenshot the following queries and results:  **Obtain a list of applications where the CourseID is unknown**    **Obtain a list of students where their CourseID is not unknown**    **Obtain a list of students whom were born in the month of June 2002**    **Obtain a list of applications where CourseID is unknown and the applications were made between 01/04/2020 and 31/07/2020** |



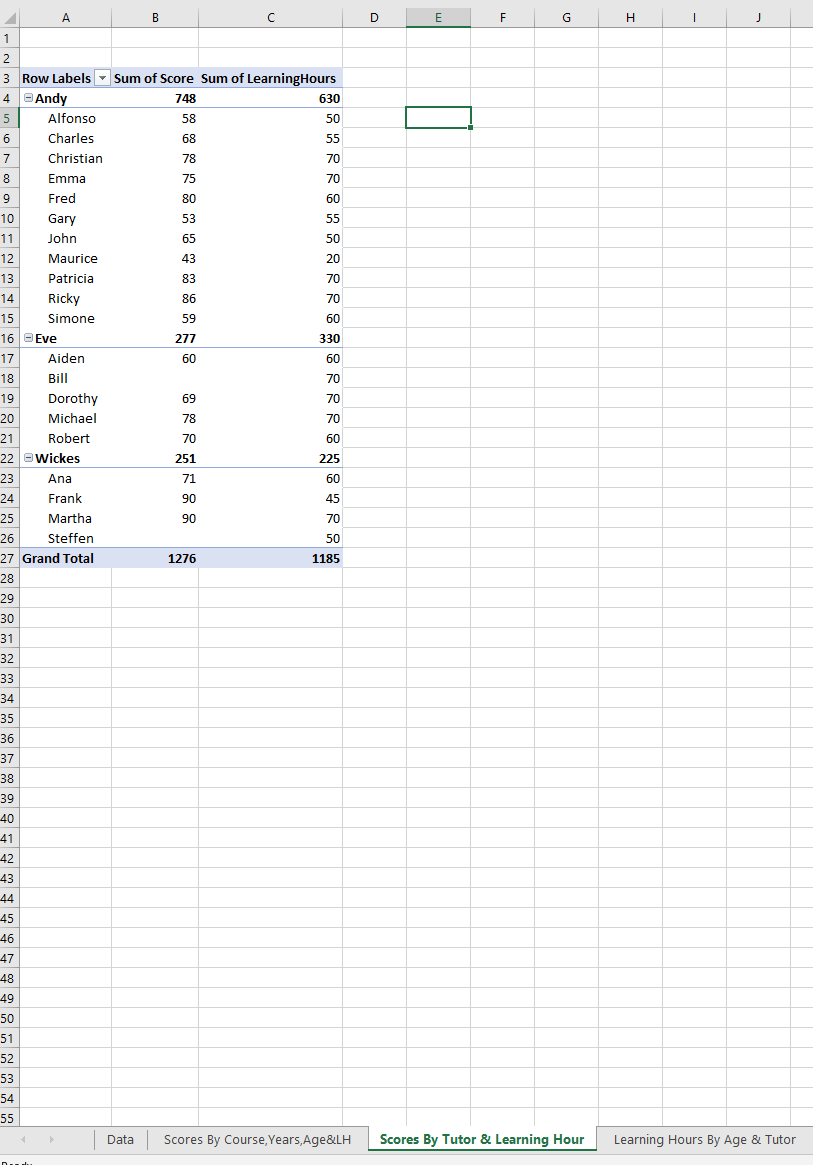
| **ACTIVITY 2: GROUPING FUNCTIONS** |
| --- |
| Using SQL, screenshot the following queries and results:  **Obtain the number of modules which are assigned to each course**    **Retrieve Information on the number of successful applications per course**    **Find the average Membership Fee of Student Clubs by the ID of the Staff member (Lecturer) supervising it**      **Find the Sum total of Joining Fees for all active clubs by Staff Member supervising them** |



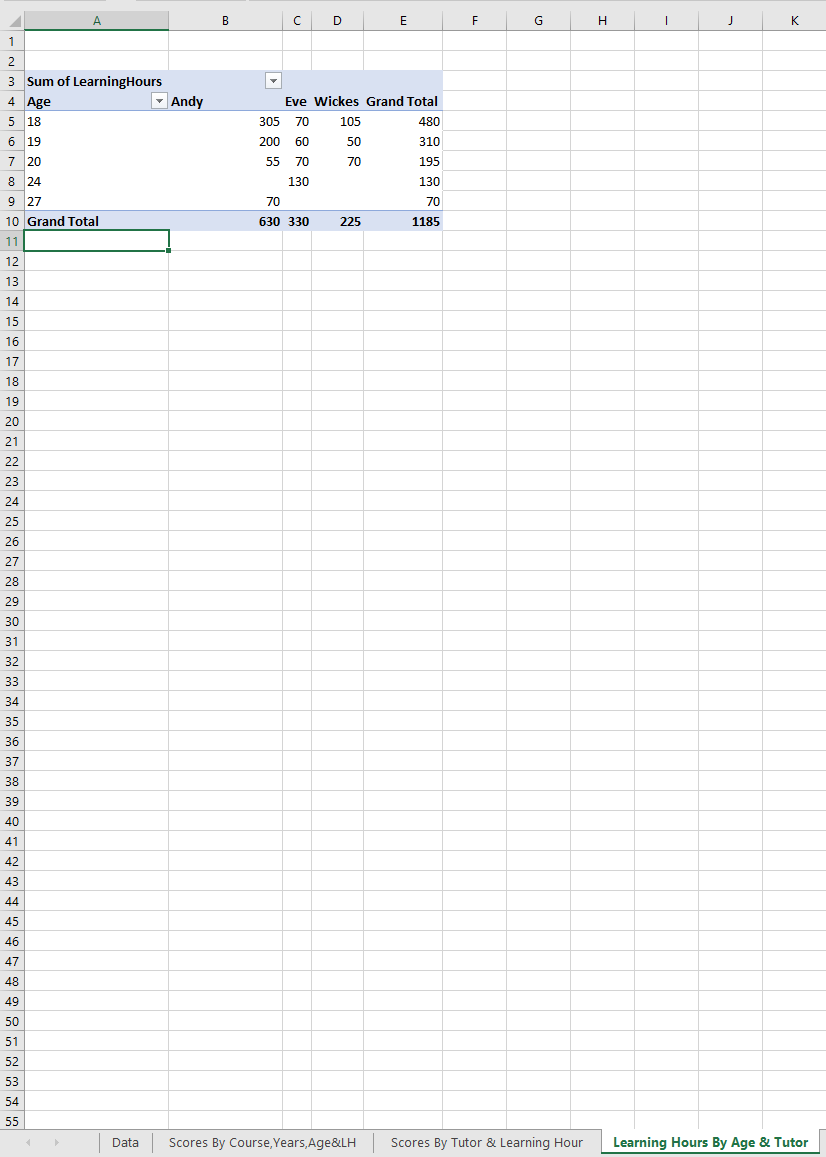


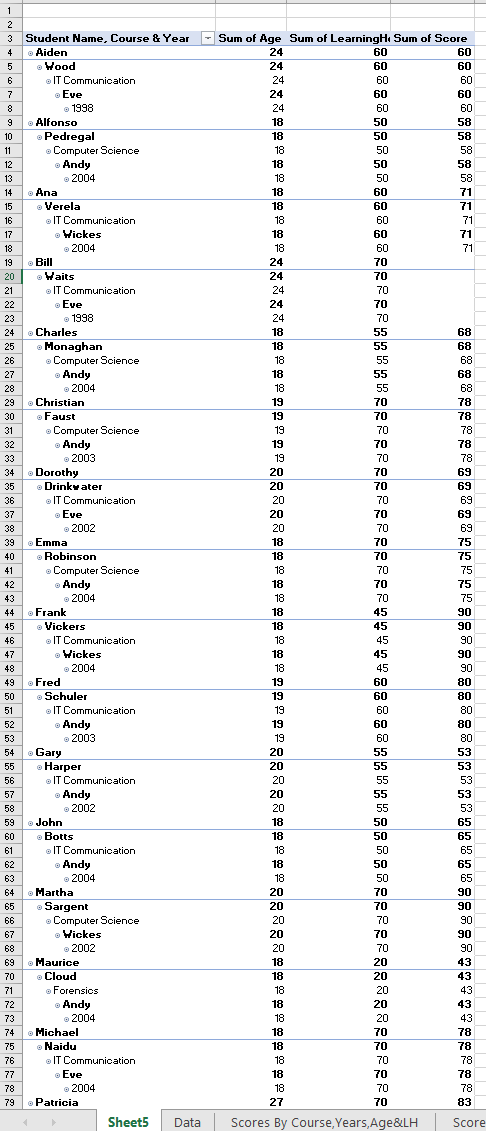
| **ACTIVITY 3: ADVANCED JOINS** |
| --- |
| Using SQL, screenshot the following queries and results:  **Obtain a list of all modules and the names of any courses they may be taught on (include modules without courses)**    **Obtain a list of students along with any related application numbers if they have them**      **Obtain the Class ID, Class Date and Feedback score of the latest class scheduled for each Class ID** |

| **ACTIVITY 4: DATA TRANSFORMATION 1** |
| --- |
| Using the Example Spreadsheet Design a number of Pivot Tables to show the data in a way that you think is useful. Screenshot the Pivot Tables you have created for your workbook.    Shows Courses in the years they were taken with the total scores, sum of learning hours and age of students. Useful too see how students performed each year , the score they obtained versus the amount of learning hours spent and the student age. Can be used to see if more learning hours means a better score , what age groups score better and if certain courses taken in a specific year performed better than others. |



Useful to see Students scores versus hours spent learning. Can be used to find a correlation between hours spent learning and score achieved.





A complete overview of each course , the students who took the course and in what year, the students age, the tutor who taught the course, the amount of hours spent learning and the score.

Can be used to find a correlation between Student age, hours spent learning , Score achieved , what year the course was taken and who taught the course.

| **ACTIVITY 5: DATA TRANSFORMATION 2** |
| --- |
| Using SQL, screenshot the following queries and results.  Using any combination of SQL queries, pivot tables or the spreadsheet skills you have acquired thus far, extract information from the database and present a series of reports which you think display useful information about the state of our university |

