Elias Leipola

Pre-study of Programming with SQL by Oracle to be implemented as Metropolia's MooC course

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communication Technology

Bachelor’s Thesis

10.6.2025

Tiivistelmä

Tekijä:	Elias Leipola
Otsikko:	Oracle-yhtiön Programming with SQL-kurssin esiselvitys Metropolian MooC-toteutusta varten
Sivumäärä:	19 sivua + 1 liitettä
Aika:	10.6.2025
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Tieto- ja viestintätekniikka
Ohjaajat:	Janne Salonen, Osaamisaluejohtaja (ICT)

Tässä opinnäytetyössä tarkastellaan esitutkimusta Programming with SQL-kurssin mahdollisuutta MOOC-toteutuksena Metropolia Ammattikorkeakoululle. Tutkimuksen menetelmät ja tulokset käydään läpi perusteellisesti. Tehdyn tutkimuksen tuloksia voidaan myöhemmin hyödyntää varsinaista toteutusta varten.

Programming with SQL on Oracle-yhtiön laatima verkkokurssi, joka on saatavilla Oraclen verkkoympäristössä nimeltä Oracle Academy. kurssin tavoitteena on perehdyttää opiskelija SQL-ohjelmointikielen perusteisiin. Kurssi soveltuu opiskelijoille, joilla ei ennestään ole paljoa tai ollenkaan kokemusta tietokannoista tai SQL-kielestä.

SQL on yleinen tietokantojen ohjelmointikieli. Se on pitkäaikainen standardi tietokantojen hallinnassa maailmanlaajuisesti. Kieltä on kehitetty jatkuvasti, eikä se ole menettänyt asemaansa tietokantaohjelmoinnin standardina. SQL:n perusteet ovat siksi erittäin hyödyllinen taito ohjelmoijille ja ohjelmistokehittäjille.

Metropolia Ammattikorkeakoulu hyötyisikin siksi suuresti, jos Oraclen SQL-kurssin saisi sisäistettyä MOOC-toteutuksena. Oppimisalustana toimisi Moodle, joka on yleisesti käytetty oppimisjärjestelmä ja siten tuttu oppilaille ja opettajille. Moodle sopii erityisen hyvin Oraclen kurssin MOOC-toteutusta varten.

Tämä opinnäytetyö käsittelee myös toteutusprosessiin liittyviä haasteita. Esitutkimuksen aikana ilmeni useita ongelmia liittyen kurssin mahdolliseen toteutukseen. Mahdollisia ratkaisuja näihin ongelmiin käsitellään myös. Esitutkimuksen tavoitteena on antaa suuntaa varsinaiselle toteutukselle tulevaisuudessa.

Avainsanat: SQL, Database, ICT, Learning, Learning management systems, MooC, Moodle, Oracle

---

Abstract

Author:	Elias Leipola
Title:	Pre-study of Programming with SQL by Oracle to be implemented as Metropolia's MooC course
Number of Pages:	19 pages + 1 appendices
Date:	10 June 2025
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Professional Major:	Information and Communication Technology
Supervisors:	Janne Salonen, Head of Department (ICT)

This thesis presents the methods and findings of a pre-study about the possibility of implementing the Programming with SQL course as a Mooc for Metropolia University of Applied Sciences. The results of this pre-study can later be used for the actual implementation.

The programming with SQL course was made by the Oracle corporation. It is available in Oracle's learning environment called Oracle Academy. The course is meant to introduce students to Structured Query Language. The course is suitable for students with little to no experience or prior knowledge of databases or SQL.

SQL is a commonly used database programming language. It has been a global standard for a long time for database management. It has been improved as time has passed. Yet it has never lost its status as the standard for database programming language. It is thus a great benefit for programmers and software developers to have at least a basic understanding of the fundamentals of SQL.

It is for this reason that Metropolia UAS would benefit greatly from the implementation of the Oracle course as a MooC. Specifically, Moodle would be used as the Learning Management System (LMS) of choice. Moodle is a very commonly used system and familiar with both students and teachers. It is a flexible solution and ideal for the implementation.

This thesis also addresses the complications with the implementation process. Several problems emerged while researching the possibilities of implementation. Additionally, solutions and workarounds to these problems will also be touched upon. The goal of this pre-study is to have direction for the actual implementation process in the future.

Keywords: SQL, Database, ICT, Learning, Learning management systems, MooC, Moodle, Oracle

Contents

Abbreviations

SQL: *Structured Query Language*. An extremely common, widely used programming language. Used to build and manage databases.

ICT: *Information and Communication Technology*. Technologies that include information processing and communications, such as computers and the internet.

MOOC: *Massive Open Online Course*. An online course that can be enrolled on by students online. Managed by teachers.

RDBMS: *Relational Database Management Systems*. A form of database, in which the records and data are connected to other bits of data.

RM: *Relational Model*. A theoretical mathematical model, which is the basis of relational databases.

ANSI: *American National Standards Institute*. An American organization that manages and oversees standardization on different fields.

ORDBMS: *Object Relational Database Management Systems*. A form of database management system, that allows a degree of object-oriented programming within a database.

LMS: *Learning Management System*. An online service or application that can host online learning such as courses.

1 Introduction

Structured Query Language (SQL) is a standardized programming language meant for use in creating and managing relational databases. It is widely used even after many decades since its creation. With SQL being so commonly used in Information and Communication Technology (ICT), the value of understanding how it works and the skills on how to use it are extremely beneficial - if not crucial – for programmers and software designers to work on complex software.

With SQL being so widely used and important in modern software development and maintenance of older systems and databases, it is a great benefit for Metropolia to have resources and courses for students to easily learn SQL and gain knowledge and understanding of how it works. Since Oracle has made an extensive online course on SQL, it would benefit Metropolia if said course could be implemented into Metropolia's course selection.

Several aspects need to be considered before actual concrete steps can be taken to implement Oracle's course for Metropolia. Several issues need also to be addressed and will be touched upon later in this thesis. In addition, the course materials need to be adapted into the MOOC version of the course, as these do not work with every possible database. This will be touched upon later as well.

This thesis focuses on studying the possibility of implementing an SQL course based on Oracle's Programming with SQL course for Metropolia University of Applied Sciences (UAS) in the form of a Massive Open Online Course (MOOC). This thesis will also touch on problems with such implementation and possible solutions to these issues. In the case of future implementation of the course, Moodle will be used. Moodle is a commonly used Learning Management System. Moodle will also be talked about later.

2 About Structured Query Language

2.1 Database basics

The term “database” might not be immediately obvious to many people, especially for those with no understanding or previous experience with information technology. However, the basic concepts behind databases are relatively simple.

“The term database has fallen into loose use lately, losing much of its original meaning. To some people, a database is any collection of data items (phone books, laundry lists, parchment scrolls . . . whatever). Other people define the term more strictly.” -Allen G. Taylor [1]

In its simplest idea, a database is a collection of data or records. Technically, an office cabinet containing paper folders full of files could be considered a database. These files would be neatly organized, perhaps alphabetically, so that everything needed could be found with as little effort as possible. This would be a form of structure for the database. Without organization, it would take an unnecessarily long time to do even the most basic searches for needed files.

A record in the context of databases is a representation of an object. For example, a veterinarian might have files on all the pets that he has taken care of. These files could be the records in a database. Each pet would have a record. These records would have information known as attributes. These attributes would describe the pet; what species, what age, what color fur etc. The specific content within these attributes is the data.

The database is made of both data and metadata. The purpose of metadata is to describe the structure of the database. The structure of the database needs to be described, so that the data contained in the database can be found and retrieved. All the data is also related to other data within the database, and these relations are also described. The database is self-describing; this means that the database contains the description for itself. In short, the database

structure (columns, tables etc.) are described in the metadata, and through it the database can be operated, and specific data can be retrieved.

2.2 Structured Query Language

Structured Query Language (SQL, pronounced ess-kew-ell) is a programming language meant for easy to understand and straight forward way to create and manage relational databases. It was created in the 1970s by Donald Chamberlin and Raymond Boyce; scientists working for International Business Machines Corporation, commonly known as IBM. In 1979, Relational Software, Inc., later known as Oracle Corporation, released the first commercially available version of SQL, which became widely used. In fact, it became so commonly used that it became the standard globally for relational database management systems (RDBMS). It was standardized by the American National Standards Institute (ANSI) in 1986 and a year later by the International Organization for Standardization. [2]

Chamberlin and Boyce managed to create SQL as a practical implementation of Edgar F. Codd's relational model (RM). While Codd's relational model was a theoretical work, Chamberlin and Boyce had made it into a system that organized data into tables, or relations, and could use commands to manipulate it, i.e. show only the relevant data from a set of tables or create completely new ones without modifying the structure of the database at all. Relational databases are based on Codd's relational model.

Since 1999, the international SQL standard has incorporated part of the object model, resulting in hybrid structures called object-relational databases [1]. Traditional databases store data in tables: each row is a record; each column is an attribute. Simply put, the column is a category that describes a thing, and the value in the column is an attribute (feature) of a thing described. This differs from an object-oriented system. These systems handle nested objects, complex data types and inheritance and polymorphism. The differences between these models can create problems if objects are attempted to be stored in relational

databases. The errors that occur are called “object-relational impedance mismatch” -errors [3].

To fix this error, SQL 1999 and later standards integrated object-relational features into SQL. The resultant systems were called “Object-Relational Database Management Systems (ORDBMS)”. These systems are essentially hybrids between relational database systems and object-oriented ones.

Object-Relational SQL implements several features that standard SQL (pre-1999) doesn’t. These include:

Define custom data types (UDTs)
E.g., a Person type with attributes name, birthdate.

Store complex structures
Arrays, nested records, composite types, or even tables within tables.

Support inheritance
You can define a type Employee that inherits from Person.

Add methods (functions)
Functions or procedures tied to types (like methods in classes).

Use table inheritance
Tables can be created to inherit the structure of other tables.

Relational database management systems (RDBMS) are database management systems that are specifically made to store data in structures known as rows and columns. Rows are oriented horizontally and columns vertically. SQL is used to interact with the database. With SQL, data stored in these columns and rows can be accessed, created, modified and deleted with simple, straight forward commands. The rows and columns form tables, which are collections of related data entries [4]. Several tables can be combined, and new tables can be

temporally created to showcase data in tables that do not really exist in the database. RDBMS is the basis of SQL as well as all modern database systems.

One of the most basic commands in SQL is the SELECT statement. With it, the user can choose and select parts of the database and display the values stored within them. The SELECT statement requires the user to understand the structure of the database. It is vital to understand which tables relate to others, and in what way. Knowing this, the user can select specific parts of the database, such as tables, rows, columns etc. to display. They can also use tools such as the WHERE clause to filter out results that are not desired from the query.

Other commonly used clauses in SQL are, for instance, ORDER BY. This clause is used to sort out results by their values. For example, if a query returns values 1,2,3 and 4, the user can have everything related to these values ordered in ascending order (ASC) or descending order (DESC).

GROUP BY clause is used so that rows that have same values in specific columns can be grouped together. This is useful if the user is looking for data to be ordered by same values in the same column. For example, ordering employees by their department would make excellent use of this clause.

HAVING clause lets the user choose to only display the results in the query that have a specific value in one or multiple columns. For example, if the user wants to select employees from a table whose salaries are over 3000€, using the HAVING clause works perfectly for this task. In addition, arithmetic operators can be used with the HAVING clause. This enables queries to be constructed in a much freer fashion, such as selecting employees who have salaries of over 3000€, even when tax has been subtracted.

The JOIN clause allows two or more rows to be combined if they share related columns. This makes displaying relative data much more intuitive, rather than having the query return two separate tables.

3 Learning Management Systems in General and Online Learning

Learning Management Systems (LMS) are software systems designed to automate learning processes, track the progress of students and make reports of training events [5]. The goal is to provide learning materials and environments for students to research course materials and test their skill and understanding. All this is meant to be achieved with minimal or no teacher involvement in the learning process, as the administrative tasks are also automated. There are several features that LMS's should be able to provide, that will be discussed in this thesis.

One of these features is tools for administrators. Administrators in this case would be teachers, who have access to the LMS. It is rather straight forward to understand, why teachers would need to have access to maintain and modify aspects of the LMS. These tools would enable the teachers to, among other things, manage admissions to the course, monitor student progress in these courses, set schedules for tasks and possible exams and update and add learning materials.

For students, features that would be available should also be considered. These features include access to the course learning materials, checking their course progress, requesting help and assistance from teachers assigned to the course, possible tech support and access to possible exams. After an exam, the student should be able to see if the performance was adequate. It should be clear to the students what aspects of the course are understood and where additional improvement is needed. Teachers should also be able to monitor the progress in exams.

Metropolia UAS offers several MOOC courses for different areas of learning. For example, Metropolia has a wide range of programming courses for several programming languages. These courses are meant to be enrolled on and studied independently by each student. When completed, an administrator for that course is notified, usually by email, and the course is then approved by them,

and the study points are added to the study program. These courses are designed for ease of access and are beneficial for anyone seeking to learn new subjects or advance their knowledge and skills on the topic [6].

For the implementation of Oracle's "Programming with SQL" -course into a MOOC for Metropolia, Moodle will be used. Moodle is a Learning Management system with hundreds of millions of users enrolled in thousands of educational institutions and organizations around the globe. It is one of the most used LMSs in the world. It is thus relatively well known among students and teachers alike [7].

Moodle was launched in 1999, and it has since grown and evolved and is nowadays widely used. It is open source, it is free to use and download and easy to share. The creators of Moodle have stated that they want Moodle to be easily accessed by everyone so that as many people as possible have the chance to access quality learning environments in many different fields and topics.

For students, navigating it is simple and easy to understand. Most students will already be familiar with it and have their accounts and passwords set up. Enrolling on new courses should be simple and easy, without difficulties. Therefore, enrolling on a Moodle course implementation of "Programming with SQL" -course should feel familiar to the students.

As with the students, the teachers are also familiar with using Moodle to manage online courses. For this reason, for the teachers, using Moodle to manage the course and monitor the enrolled students' progress should also be easy. If the students on the course have any questions or if they need assistance from the teacher, they can easily contact them through Moodle. This is because Moodle has a feature for students to ping a teacher from a button, if they need help.

Moodle is also very flexible. It allows for several ways to embed content or create tasks and exams for the students. These tasks can be done to suit any learning situation and goal in the best way possible, deemed by the teacher or

course administrator. The specific tasks suitable for the “Programming with SQL” -course will be discussed later.

4 Aspects of implementing the course for Metropolia’s Moodle

The goal of the work was to explore the possibility of using the oracle SQL programming course in Metropolia’s Moodle. Oracle has an extensive and in-depth programming course for SQL, called “Programming with SQL”. It is accessible only from Oracles own learning platform “Oracle Academy”. It would benefit Metropolia, if this course could be made accessible from a learning management system that is used by Metropolia. It should be possible to implement this course so that it is accessible from Metropolia’s own platform. In addition, student’s learning could be monitored through Moodle, which is not possible for Metropolias’s teachers from Oracle Academy.

4.1 Programming with SQL by Oracle

Programming with SQL is an online course made by Oracle corporation. It is accessible online from oracles learning site Oracle Academy. However, to enroll on this course, a user is going to need access to it. This will create complications, which will be addressed later. The course is autonomous, meaning that the students’ progress on the course independently. This gives the students a great deal of control over their work hours and freedom to progress at their own pace.

The course consists of several chapters. Each chapter touches on a specific aspect of the SQL language. for instance, the first chapters introduce the student to the very basic functions of SQL, so that the student has a solid basis of knowledge about SQL to then progress to more complicated parts of the course. Naturally, the course gets more complicated and demanding as the student progresses further. This means that the overall progress on the course will

be slower the further the student advances on the course. However, progressing through the course from one chapter to another feels natural. This is because each chapter has a scope on the concepts and the chapters make sure to not overcomplicate the topics. This helps the student to grasp the concepts within each chapter intuitively and without overwhelming students with less experience with database concepts.

The learning materials within each chapter consist of pdf-files of information for the students. These pdf-files are accessible on the course through a web browser. Additionally, these files can be downloaded to the user's machine. This is helpful for students who might be struggling with a stable internet connection. Since the course is done entirely through pdf files, the course can be essentially made an offline course, should the student wish to download all the course's pdf- files.

Each chapter consists of approximately three segments. Each of these segments contain two pdf -files. In these pdf -files the chapter topic is introduced and learning materials displayed. The student goes through these materials and learns from there. After the segment's learning materials have been thoroughly examined, the student may mark the materials completed and can progress to the final part of the segment. This part is also a pdf -file, the third in the segment, but instead of materials for learning, there are found tasks that the student needs to complete. However, the course does not include anything in which to give the answers to the tasks. It is unclear whether or not the tasks are meant to be done in Oracle APEX system (Oracle APEX will be talked about later) and be logged there. This is not realistically a problem for the implementation process to Metropolia's Moodle, as will be demonstrated later.

Progress on the course is logged by the student; when a chapter is completed, the student checks the chapter completed by themselves. With the implementation to Metropolia's Moodle, progress will be logged there. This will be clarified later.

The user interface (UI) of the course is good. It is clear and intuitive to navigate. It should not cause the student any confusion on how to access each chapter and where to progress next. The UI is also very responsive. However, the experience may vary depending on the device the course is accessed from. In this work only the browser version will be discussed on PC -platform.

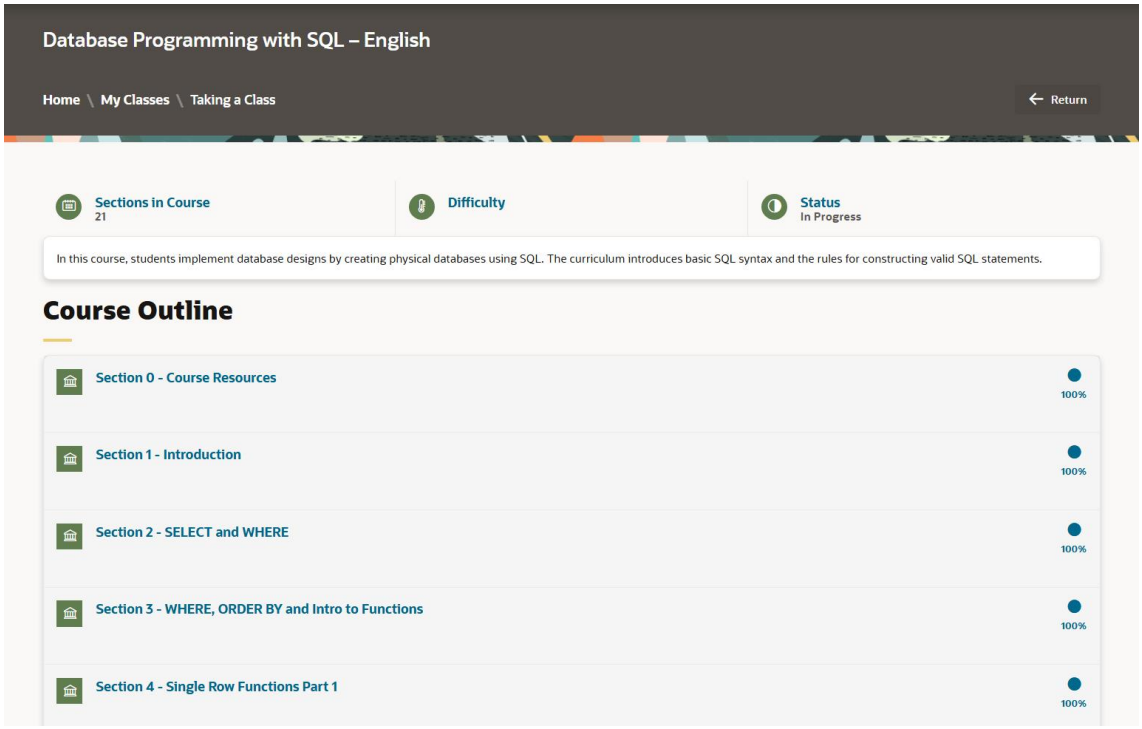


Figure 1. Course chapter selection view in Oracle Academy

The course mentions Oracle APEX. This is Oracle's database software that is meant to be used on this course. Specifically, the tasks within the course require the student to have access to this software. However, a problem was encountered while trying to access APEX. This resulted in APEX being dropped as an option for the Metropolia implementation of the course in Moodle. An alternative solution was considered and experimented on. While the results were promising, some problems and difficulties arose from this. The problem with Oracle Apex and the alternative solution will be touched upon later.

4.2 Studying the possibility of Moodle implementation

As stated before, the goal of this pre-study was to find out the viability of implementing the Oracle course for Metropolia's Moodle. Naturally, the first step to determining this was to enroll on the course and progress on it. This was to gain an idea of the overall course, how it worked, what could be implemented as is and what needed to be done in case of problems or compatibility errors. Access to the course was gained from Janne Salonen, the teacher supervising this work.

There were no issues with logging in to the Oracle environment. Once logged in, access to the course material was gained. First, a look was taken into the course materials. From there, it was found that the course required access to Oracle APEX, a database software in which the tasks, such as writing the proper queries and examining the course database were meant to be done. This was the first and most pressing problem encountered.

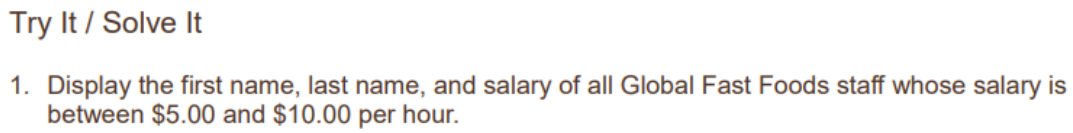
Access to Oracle APEX was not gained despite numerous attempts. It is unclear what exactly caused this practical problem. One theory is that the way to access the software has changed, but the instructions on the course itself remained out of date. An attempt was later made with Salonen, but the problem persisted.

The problem was unable to be solved, so an alternative was decided. There exists an open-source database by MySQL called "classicmodels". It was decided with Salonen that instead of using Oracle APEX to examine the course database, classicmodels would be utilized instead. One of the benefits to this solution was that the classicmodels database is very clear to conceptualize, even to those with limited or no prior experience with databases. Another benefit was that the database could be accessed with many different software that were both free to use and widely available.

However, this solution required a compromise. Since the Oracle course needed to use a specific database (that was essentially inaccessible due to the problem

described formerly), the classicmodels database was not going to be compatible with course content. This required that any references to the course database needed to be formatted so that they applied to the classicmodels database that was now in use. Effectively, this meant that several tasks needed to be reinterpreted to fit into classic models. That said, care was taken to make sure that the actual content of the query, in other words, what the specific task was meant to train and test, was unaltered.

The image below is a task from the third segment of chapter two. In this task, the student needs to write a query that retrieves the first name, last name and salary of all Global Fast Foods staff whose salary is between $5.00 and $10.00 per hour. Obviously, this specific query is impossible to do on classicmodels, since no data or tables exist in it as described in the task. This error was circumvented by modifying the query to fit the classicmodels database, without altering the fundamental way the query works.


Try It / Solve It

1. Display the first name, last name, and salary of all Global Fast Foods staff whose salary is between $5.00 and $10.00 per hour.

Figure 2. An example task from the course.

In the image below can be seen how this problem was solved. Instead of writing the exact query required in the task description, it has been modified to fit classicmodels. Specifically, instead of attempting to retrieve the staff's first and last names of those whose hourly salaries are between $5.00 and $10.00, the query retrieves the customer number of those customers whose payments are between 40000 and 85000. These are obviously major differences between the queries. However, it should be noted that the actual query structure is the same. In other words, the specific training that the query in the course tasks attempts can be achieved on the one modified for classicmodels as well.

```
1 SELECT customerNumber FROM classicmodels.payments where amount between 40000 and 85000;
```

Figure 3. The same query as the previous, but modified to fit classicmodels.

This was the method used to fit the tasks in the course to work also with classicmodels database. However, some of these queries were not compatible at all. In these cases, modifying them would've inadvertently changed the query to such a degree that the fundamental aspect that it meant to train would've been changed to something else. Thus, these queries were ignored.

Since the course itself lacked any way to write and save the queries and tasks (speculated to be done on Oracle APEX as stated before), another solution was needed. Ultimately it was decided that the answers would be recorded simply in a text document. Realistically, almost any kind of text document would've sufficed. It was decided that Notepad++ would be used for its simplicity and ability to show code clearly. Specifically practical was the option to set it to recognize SQL. This made the code very clear to view compared to other text documents like the standard Windows notepad.

In the image below is a demonstration of how the task answers and queries were recorded in Notepad++. It should be noted that the tasks that simply could not be implemented by using classicmodels were marked with three lines (---). Additionally, on line 100 only the letter "c" is typed. This is because some tasks did not require a query. Rather, some of them required a written answer. In this case, the sixth task in the segment was a multiple-choice question. The correct answer was option c, hence the letter c in the answer file.

```
2-3

1 SELECT customerNumber FROM classicmodels.payments where amount between 40000 and 85000;
2 ---
3 SELECT first_name, last_name
FROM f_staffs
WHERE salary >= 20.00 AND salary <= 60.00;
4 SELECT checkNumber FROM classicmodels.payments where checkNumber like '_A%';
5 ---
6 SELECT lastName FROM classicmodels.employees where lastName like '%s';
7 c. WHERE quantity IS NULL;
8 ---


3-1

1 Latter has larger result SET
2 SELECT contactLastName FROM classicmodels.customers where contactLastName like '%e%'
  AND contactLastName LIKE '%i%';
3 SELECT firstName, lastName, jobTitle FROM employees WHERE 7.50 > 6.50
4 SELECT customerName, contactLastName FROM classicmodels.customers WHERE contactLastName LIKE 'D%'
  AND contactLastName LIKE '%a%'
  AND contactLastName LIKE '%e%';
5 ---
6 c
7 ---
8 ---


3-2

```

Figure 4. The tasks were recorded into Notepad++.

As mentioned earlier, classicmodels database was used during the work. This database was accessed using both MySQL Workbench and MySQL Shell. The queries were tested through either of these programs.

In the image below can be seen a query; “SELECT * FROM classicmodels. customers;”. This is how said query appears on MySQL Workbench. It should be noted that the entire result of the query is not shown as it is far too large to fit in a single image.

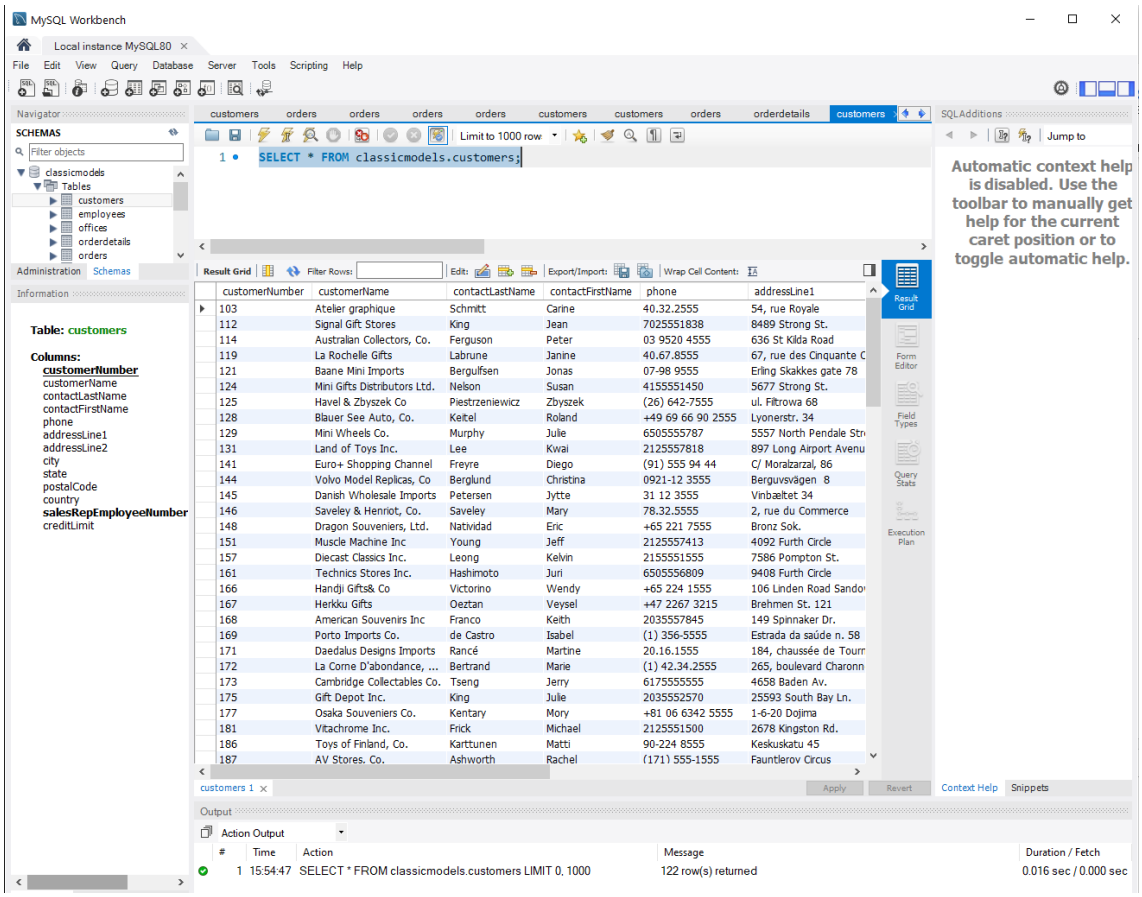
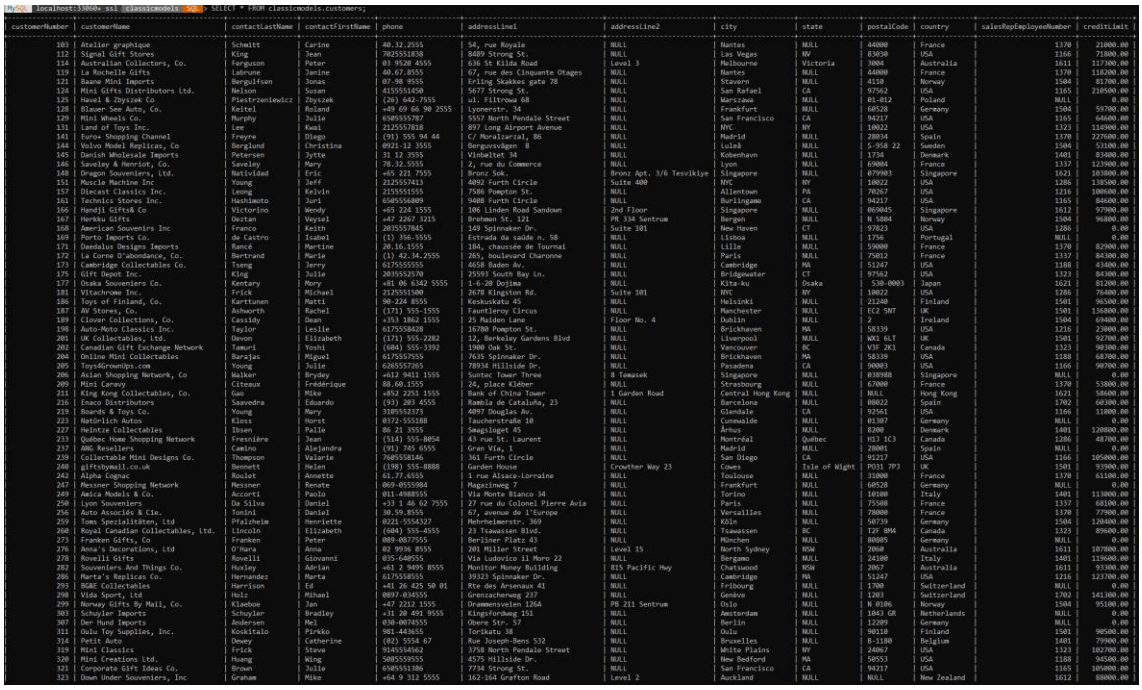


Figure 5. View of MySQL Workbench application.

In the image below there is the same query as in the previous image. This time the query was made in MySQL Shell. MySQL Shell has a more classic appearance, familiar to any database developer. As with the previous image, the entire result of the query is impossible to fit into a single image. Therefore, only the first part is shown.



```
MySQL  localhost:33060+ ssl  classicmodels  SQL > SELECT * FROM classicmodels.customers;
```

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	NULL	Nantes	NULL	44000	France	1370	21000.00
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	NULL	Las Vegas	NV	83030	USA	1166	71800.00
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia	1611	117300.00
119	La Rochelle Gifts	Labrune	Janine	40.67.8555	67, rue des Cinquante Otages	NULL	Nantes	NULL	44000	France	1370	118200.00
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555	Erling Skakkes gate 78	NULL	Stavern	NULL	4110	Norway	1504	81700.00
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA	1165	210500.00
125	Havel & Zbyszek Co	Piestrzeniewicz	Zbyszek	(26) 642-7555	ul. Filtrowa 68	NULL	Warszawa	NULL	01-012	Poland	NULL	0.00
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555	Lyonerstr. 34	NULL	Frankfurt	NULL	60528	Germany	1504	59700.00
129	Mini Wheels Co.	Murphy	Julie	6505555787	5557 North Pendale Street	NULL	San Francisco	CA	94217	USA	1165	64600.00
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	NULL	NYC	NY	10022	USA	1323	114900.00
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44	C/ Moralzarzal, 86	NULL	Madrid	NULL	28034	Spain	1370	227600.00
144	Volvo Model Replicas, Co	Berglund	Christina	0921-12 3555	Berguvsvägen 8	NULL	Luleå	NULL	S-958 22	Sweden	1504	53100.00
145	Danish Wholesale Imports	Petersen	Jytte	31 12 3555	Vinbæltet 34	NULL	Kobenhavn	NULL	1734	Denmark	1401	83400.00
146	Saveley & Henriot, Co.	Saveley	Mary	78.32.5555	2, rue du Commerce	NULL	Lyon	NULL	69004	France	1337	123900.00
148	Dragon Souveniers, Ltd.	Natividad	Eric	+65 221 7555	Bronz Sok.	Bronz Apt. 3/6 Tesvikiye	Singapore	NULL	079903	Singapore	1621	103800.00
151	Muscle Machine Inc	Young	Jeff	2125557413	4092 Furth Circle	Suite 400	NYC	NY	10022	USA	1286	138500.00
157	Diecast Classics Inc.	Leong	Kelvin	2155551555	7586 Pompton St.	NULL	Allentown	PA	70267	USA	1216	100600.00
161	Technics Stores Inc.	Hashimoto	Juri	6505556809	9408 Furth Circle	NULL	Burlingame	CA	94217	USA	1165	84600.00
166	Handji Gifts& Co	Victorino	Wendy	+65 224 1555	106 Linden Road Sandown	2nd Floor	Singapore	NULL	069045	Singapore	1612	97900.00
167	Herkku Gifts	Oeztan	Veysel	+47 2267 3215	Brehmen St. 121	PR 334 Sentrum	Bergen	NULL	N 5804	Norway	1504	96800.00
168	American Souvenirs Inc	Franco	Keith	2035557845	149 Spinnaker Dr.	Suite 101	New Haven	CT	97823	USA	1286	0.00
169	Porto Imports Co.	de Castro	Isabel	(1) 356-5555	Estrada da saúde n. 58	NULL	Lisboa	NULL	1756	Portugal	NULL	0.00
171	Daedalus Designs Imports	Rancé	Martine	20.16.1555	184, chaussée de Tournai	NULL	Lille	NULL	59000	France	1370	82900.00
172	La Corne D'abondance, Co.	Bertrand	Marie	(1) 42.34.2555	265, boulevard Charonne	NULL	Paris	NULL	75012	France	1337	84300.00
173	Cambridge Collectables Co.	Tseng	Jerry	6175555555	4658 Baden Av.	NULL	Cambridge	MA	51247	USA	1188	43400.00
175	Gift Depot Inc.	King	Julie	2035552570	25593 South Bay Ln.	NULL	Bridgewater	CT	97562	USA	1323	84300.00
177	Osaka Souveniers Co.	Kentary	Mory	+81 06 6342 5555	1-6-20 Dojima	NULL	Kita-ku	Osaka	530-0003	Japan	1621	81200.00
181	Vitachrome Inc.	Frick	Michael	2125551500	2678 Kingston Rd.	Suite 101	NYC	NY	10022	USA	1286	76400.00
186	Toys of Finland, Co.	Karttunen	Matti	90-224 8555	Keskuskatu 45	NULL	Helsinki	NULL	21240	Finland	1501	96500.00
187	AV Stores, Co.	Ashworth	Rachel	(171) 555-1555	Fauntleroy Circus	NULL	Manchester	NULL	EC2 5NT	UK	1501	136800.00
189	Clover Collections, Co.	Cassidy	Dean	+353 1862 1555	25 Maiden Lane	Floor No. 4	Dublin	NULL	2	Ireland	1504	69400.00
198	Auto-Moto Classics Inc.	Taylor	Leslie	6175558428	16780 Pompton St.	NULL	Brickhaven	MA	58339	USA	1216	23000.00
201	UK Collectables, Ltd.	Devon	Elizabeth	(171) 555-2282	12, Berkeley Gardens Blvd	NULL	Liverpool	NULL	WX1 6LT	UK	1501	92700.00
202	Canadian Gift Exchange Network	Tamuri	Yoshi	(604) 555-3392	1900 Oak St.	NULL	Vancouver	BC	V3F 2K1	Canada	1323	90300.00
204	Online Mini Collectables	Barajas	Miguel	6175557555	7635 Spinnaker Dr.	NULL	Brickhaven	MA	58339	USA	1188	68700.00
205	Toys4GrownUps.com	Young	Julie	6265557265	78934 Hillside Dr.	NULL	Pasadena	CA	90003	USA	1166	90700.00
206	Asian Shopping Network, Co	Walker	Brydey	+612 9411 1555	Suntac Tower Three	8 Temasek	Singapore	NULL	038988	Singapore	NULL	0.00
209	Mini Caravy	Citeaux	Frédérique	88.60.1555	24, place Kléber	NULL	Strasbourg	NULL	67000	France	1370	53800.00
211	King Kong Collectables, Co.	Gao	Mike	+852 2251 1555	Bank of China Tower	1 Garden Road	Central Hong Kong	NULL	NULL	Hong Kong	1621	58600.00
216	Enaco Distributors	Saavedra	Eduardo	(93) 203 4555	Rambla de Cataluña, 23	NULL	Barcelona	NULL	08022	Spain	1702	60300.00
219	Boards & Toys Co.	Young	Mary	3105552373	4097 Douglas Av.	NULL	Glendale	CA	92561	USA	1166	11000.00
223	Natürlich Autos	Kloss	Horst	0372-555188	Taucherstraße 10	NULL	Cunewalde	NULL	01307	Germany	NULL	0.00
227	Heintze Collectables	Ibsen	Palle	86 21 3555	Smagsloget 45	NULL	Århus	NULL	8200	Denmark	1401	120800.00
233	Québec Home Shopping Network	Fresnière	Jean	(514) 555-8054	43 rue St. Laurent	NULL	Montréal	Québec	H1J 1C3	Canada	1286	48700.00
237	ANG Resellers	Camino	Alejandra	(91) 745 6555	Gran Vía, 1	NULL	Madrid	NULL	28001	Spain	NULL	0.00
239	Collectable Mini Designs Co.	Thompson	Valarie	7605558146	361 Furth Circle	NULL	San Diego	CA	91217	USA	1166	105000.00
240	giftsbymail.co.uk	Bennett	Helen	(198) 555-8888	Garden House	Crowther Way 23	Cowes	Isle of Wight	PO31 7PJ	UK	1501	93900.00
242	Alpha Cognac	Roulet	Annette	61.77.6555	1 rue Alsace-Lorraine	NULL	Toulouse	NULL	31000	France	1370	61100.00
247	Messner Shopping Network	Messner	Renate	069-0555984	Magazinweg 7	NULL	Frankfurt	NULL	60528	Germany	NULL	0.00
249	Amica Models & Co.	Accorti	Paolo	011-4988555	Via Monte Bianco 34	NULL	Torino	NULL	10100	Italy	1401	113000.00
250	Lyon Souveniers	De Sousa	Daniel	+33 1 46 62 7555	27 rue du Colonel Pierre Avia	NULL	Paris	NULL	75508	France	1337	68100.00
256	Auto Associés & Cie.	Tonini	Daniel	30.59.8555	67, avenue de l'Europe	NULL	Versailles	NULL	78000	France	1370	77900.00
259	Toms Spezialitäten, Ltd	Pfalzheim	Henriette	0221-5554327	Mehrheimerstr. 369	NULL	Köln	NULL	50739	Germany	1504	120400.00
260	Royal Canadian Collectables, Ltd.	Lincoln	Elizabeth	(604) 555-4555	23 Tsawassen Blvd.	NULL	Tsawassen	BC	T2F 8M4	Canada	1323	89600.00
273	Franken Gifts, Co	Franken	Peter	089-0877555	Berliner Platz 43	NULL	München	NULL	80805	Germany	NULL	0.00
276	Anna's Decorations, Ltd	O'Hara	Anna	02 9936 8555	201 Miller Street	Level 15	North Sydney	NSW	2060	Australia	1611	107800.00
278	Rovelli Gifts	Rovelli	Giovanni	035-640555	Via Ludovico il Moro 22	NULL	Bergamo	NULL	24100	Italy	1401	119600.00
282	Souveniers And Things Co.	Huxley	Adrian	+61 2 9495 8555	Monitor Money Building	815 Pacific Hwy	Chatswood	NSW	2067	Australia	1611	93300.00
286	Marta's Replicas Co.	Hernandez	Marta	6175558555	39323 Spinnaker Dr.	NULL	Cambridge	MA	51247	USA	1216	123700.00
293	BG&E Collectables	Harrison	Ed	+41 26 425 50 01	Rte des Arsenaux 41	NULL	Fribourg	NULL	1700	Switzerland	NULL	0.00
298	Vida Sport, Ltd	Holz	Mihael	0897-034555	Grenzacherweg 237	NULL	Genève	NULL	1203	Switzerland	1702	141300.00
299	Norway Gifts By Mail, Co.	Klaeboe	Jan	+47 2212 1555	Drammensveien 126A	PB 211 Sentrum	Oslo	NULL	N 0106	Norway	1504	95100.00
303	Schuyler Imports	Schuyler	Bradley	+31 20 491 9555	Kingsfordweg 151	NULL	Amsterdam	NULL	1043 GR	Netherlands	NULL	0.00
307	Der Hund Imports	Andersen	Mel	030-0074555	Obere Str. 57	NULL	Berlin	NULL	12209	Germany	NULL	0.00
311	Oulu Toy Supplies, Inc.	Koskitalo	Pirkko	981-443655	Torikatu 38	NULL	Oulu	NULL	90110	Finland	1501	90500.00
314	Petit Auto	Dewey	Catherine	(02) 5554 67	Rue Joseph-Bens 532	NULL	Bruxelles	NULL	B-1180	Belgium	1401	79900.00
319	Mini Classics	Frick	Steve	9145554562	3758 North Pendale Street	NULL	White Plains	NY	24067	USA	1323	102700.00
320	Mini Creations Ltd.	Huang	Wing	5085559555	4575 Hillside Dr.	NULL	New Bedford	MA	50553	USA	1188	94500.00
321	Corporate Gift Ideas Co.	Brown	Julie	6505551386	7734 Strong St.	NULL	San Francisco	CA	94217	USA	1165	105000.00
323	Down Under Souveniers, Inc	Graham	Mike	+64 9 312 5555	162-164 Grafton Road	Level 2	Auckland	NULL	NULL	New Zealand	1612	88000.00

Figure 6. view of MySQL Shell.

To enroll and progress on the course, access to the Oracle course in Oracle Academy is required. Without access, the course materials cannot be accessed. The access to the Oracle Academy course can be provided by the supervising teacher. After choosing to enroll on the course, a teacher will then provide the student with directions to create an account for the Oracle Academy. After this, the teacher will provide access to the Programming with SQL course.

Once full access has been gained to the course, and by extension the course materials, the student can begin learning. The training will work the same way as it does on the Oracle course, reading through the learning materials found on the pdf -files. If the student has any questions, the supervising teacher can be contacted, and assistance can be requested. Progress through the course is made chapter by chapter, similarly, like on Oracle Academy.

After a chapter is completed, the student moves on to the task section. This can be done in numerous different ways in Moodle. For example, the student will be prompted to write a specific query based on the topic of the chapter. The

student then writes it down in the text field in Moodle. Then, based on whether the answer was correct or not, the student can progress to the next chapter.

In addition, several different types of tasks can be implemented in the task sections of the Moodle course. For example, multiple-choice tasks are a valid option for some of the exercises. This would add a deal of variability into the tasks. The course administrator could tailor each segment of the course, however is seen fit. Additionally, these tasks segments can be modified later if difficulties arise, such as a great number of students finding some tasks unnecessarily confusing.

After the student has passed all (or the required amount) of the chapters, they can attempt a final test. Here would be collected tasks and quizzes about topics of every chapter of the course. Like in the task segments, a mixture of written tasks and multiple-choice tasks could be implemented. Once the student passes the test, they would contact the teacher and have the course marked as completed and receive a grade.

The Oracle Academy course “Programming with SQL” should be reasonably simple to be implemented as a Moodle course for Metropolia. The methods of doing the implementation, as described above, should be relatively straightforward and cause no issues. Since the learning materials are found in Oracle Academy, these do not need to be accessible from Moodle directly. Rather, only the tasks and exams should be made into Moodle for when the student has read the corresponding learning materials for each chapter.

5 Conclusions

5.1 Starting point

The premise of this pre-study was to research the possibility of implementing a programming course called “Programming with SQL” into Metropolia’s Moodle environment. The course was originally made by Oracle, and it is accessible

only from Oracle's online learning environment "Oracle Academy." It would be greatly beneficial for Metropolia University of Applied Sciences, if this course could be accessible from Moodle.

To find out the possible way of doing the implementation, the first step was to enroll on the course and progress on it. This was to gain an idea of how the course progressed and how it would fit the Moodle environment. Access to the course was gained with the help of a teacher. After this, progress was made on the course.

The course consists of the learning materials, which were accessible as pdf-files. Each chapter had three segments dedicated to a specific topic of the chapter. At the end of each segment there was one more pdf-file, containing the tasks for that segment.

To do the tasks, Oracle intended that a software called "APEX" be used. This was Oracle's database software, which was to be used to access a database intended for the tasks of the course. However, despite efforts, this software would not work. An alternative solution was required. It was decided that an alternative, free database software was to be used. MySQL was chosen for its simplicity and familiarity, as well as availability. However, this posed another problem. The database intended for the course could not be accessed without Oracle APEX, and an alternative database was also required. It was decided that the free database "classicmodels" was to be used.

Now the database could be accessed, but since it was a different database from the one intended for the course, some compatibility problems arose. Specifically, the queries in the tasks that were meant for the student to be written into APEX would not work with a different database. To make them work, they needed to be modified. This worked well, because even when they were changed, care was taken to make sure that the query functioned the same way, so that the same training was included.

5.2 Results

After researching the possibility to implement Oracle's "programming with SQL" into Metropolia's Moodle it can be concluded that this is indeed possible. Even though the learning materials are only accessible from Oracle Academy, this should not be a problem. Access to Oracle Academy can be given by a teacher acting as the administrator for this Moodle course. Then the student can access the materials directly from Oracle Academy.

The greatest benefit from a Moodle implementation comes in the form of the tasks. For each chapter of the course (corresponding to the chapters in the learning materials) task pages can be created in Moodle. These tasks can be done in various ways, such as written answers where the student needs to write the correct query into a text field, multiple-choice tasks or anything the course administrator wants. This way the potentially replicable problems that were encountered with Oracle APEX can be avoided altogether.

The implementation of Oracle's "Programming with SQL" course into Metropolia's Moodle environment has several benefits. Specifically, it makes the teachers work to monitor students' progress much easier, since it can be seen in Moodle. Moodle is also a much more familiar environment for teachers and students alike, making it much more intuitive and familiar for students to enroll on the course. Moodle also makes it possible for the teachers to have full control over the course tasks.

Overall, the goal of this pre-study was reached. It can undoubtedly be concluded that the course is indeed possible, if not preferable to be implemented into Metropolia's Moodle environment. Now that the prospect has been evaluated it is possible to start concrete work to make an implementation.

Sources

1 Taylor A. *SQL for Dummies* [Internet]. 5th ed. Hoboken (NJ): Wiley Publishing; 2003 [cited 2025 April 15]. Available from: https://datubaze.wordpress.com/wp-content/uploads/2016/03/a_taylor_sql_for_dummies_2003.pdf

2 TechTarget. SQL (Structured Query Language) [Internet]. TechTarget; [cited 2025 April 17]. Available from: https://www.techtarget.com/searchdatamanagement/definition/SQL

3 ChatGPT. Asked about ORDBMS. [cited 2025 April 22].

4 TutorialsPoint. SQL - RDBMS Concepts [Internet]. [cited 2025 April 23]. Available from: https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm

5 ASTD. Learning Management Systems: The field guide [Internet]. 2009 [cited 2025 May 3]. Available from: https://web.archive.org/web/20140824102458/http://www.astd.org/~/media/Files/Publications/LMS_fieldguide_20091

6 Metropolia University of Applied Sciences. Open University of Applied Sciences [Internet]. [cited 2025 May 10]. Available from: https://www.metropolia.fi/en/academics/open-university

7 Moodle. About Moodle [Internet]. [cited 2025 May 25]. Available from: https://moodle.com/about/