University of Sunderland

School of Computer Science

CETM75 Assignment 2

This assessment contributes 65% to your final module mark.

The following learning outcomes will be assessed:

> 3. Display a critical understanding of data models, e.g. relational, NoSQL, and where they should be used.
> 4. Use Database Management Systems (e.g. Oracle, Postgres) and associated technologies in secure information and database systems development

Important Information

You are required to submit your work within the bounds of the University Infringement of Assessment Regulations (see your Programme Guide). Plagiarism, paraphrasing and downloading large amounts of information from external sources, will not be tolerated and will be dealt with severely. Although you should make full use of any source material, which would normally be an occasional sentence and/or paragraph (referenced) followed by your own critical analysis/evaluation. You will receive no marks for work that is not your own. Your work may be subject to checks for originality which can include use of an electronic plagiarism detection service.

Where you are asked to submit an individual piece of work, the work must be entirely your own. The safety of your assessments is your responsibility. You must not permit another student access to your work.

Where referencing is required, unless otherwise stated, the Harvard referencing system must be used (see your Programme Guide).

Please ensure that you retain a duplicate of your assignment. We are required to send samples of student work to the external examiners for moderation purposes. It will also safeguard in the unlikely event of your work going astray.

Submission Date and Time	As advised on Canvas
Submission Location	Electronic submission via Canvas

There are two tasks for this assessment:

Task 1

Scenario

Introduction

Milllie’s Musical Emporium (MME) Ltd has grown from a small company based in a small market town, to one of the country’s leading suppliers of musical instruments and associated media (e.g. printed music, books, CDs and DVDs). However, they have never quite managed to move away from using a paper-based filing system for storing customer, sales and stock information. To cope with their growth and allow for more efficient stock recording, the store has decided to computerise their customer management and stock recording system. You have been tasked with developing a database application to meet their needs.

Current Position

Currently, MME Ltd record details of all customers (including their name, address, telephone number, date of birth and bank details, i.e. bank name, address, sort code and account number) who either purchase a musical instrument or media. Records are also kept of every transaction that takes place in any of the stores.

A stock warehouse is also kept. This makes it possible to see where any given product (including its identifier, type, name, description, cost) is currently stored. The stock warehouse also contains details of all purchases and allows stores to move stock from one store to another. This is particularly useful for those customers who wish to purchase a product that is not available in their local store.

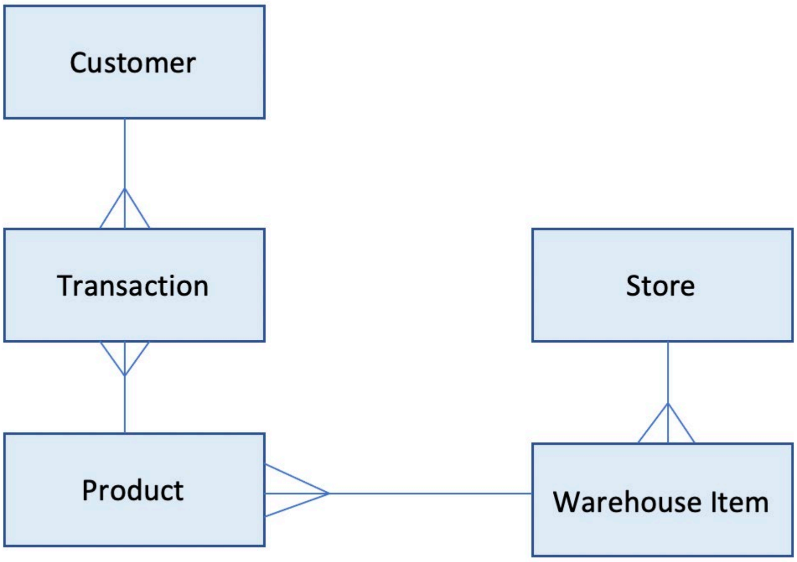
The organisation would also like to be able to create management reports, which may, for example, show all sales at a particular store or a group of stores, between certain dates.

Proposed System

Using PostgreSQL, you are required to design and develop a prototype system that not only satisfies the requirements of the current system, but also has features that you consider to be worthwhile enhancements to the current system.

To achieve this, base your system on the following **entity-relationship diagram** and **partial data dictionary**

Entity-Relationship Diagram



Partial Data Dictionary

Customer

Attribute Name	**Description**	**Data Type**	**Constraint**
Customer_Id	Customer Identifier	VARCHAR(3)	Primary Key
Customer_Sname	Customer Surname	VARCHAR(20)	Not null
…			

Transaction

Attribute Name	**Description**	**Data Type**	**Constraint**
Customer_Id	**Reference to customer identified**	**VARCHAR(3)**	**Foreign Key (To Customer)**
…			

Note: The data dictionary is only partially complete.

Task 1:

a) Subsequently, produce a single SQL script file which can be run within PostgreSQL without error and which drops and creates your tables (correctly ensuring that any referential integrity issues can be resolved), and inserts sample data into each table.

The SQL script file must also contain the code for the PL/pgSQL code that you implement in (b) below.

b) Using PostgreSQL develop:

i. A PL/pgSQL stored procedure (and any associated code) which allows for **registration of new customers**.

ii. A PL/pgSQL stored procedure (and any associated code) which allows an existing customer to **purchase a product**. This transaction must allow the client to specify a specific product to purchase, a delivery date and time, ensuring that the delivery can only be booked if both that product and delivery slot are available.

Each PL/pgSQL stored procedure may require you to develop other PL/pgSQL stored functions, triggers and cursors that you think necessary to fully implement the required functionality.

When developing the system you should take into account the important development issues identified below:

- Data types used should match those used in the tutorial booklet.
- Dates and other relevant data should be validated accordingly.
- Exception Handling must be in place to deal with all errors, e.g. invalid dates, duplicate customers, incorrect products specified, insufficient stock etc.
- Any fields that require mandatory input, i.e. NOT NULL must be validated on input.

Task 2

Millie's Music Emporium has two users: **admin** and **customer**.
For each table specify what privileges you would give each user and briefly explain why they would have this privilege.

> As a reminder here is a link to privileges in PostgreSQL:
> https://www.postgresql.org/docs/13/ddl-priv.html
>
> You do not have to test these privileges in PostgreSQL, just list the commands, i.e. write the relevant GRANT commands for each table and provide a brief explanation.

Task 3:

You are required to prepare and submit a **5 to 10-minute voice-over screencast** in which you present a set of Powerpoint slides and critically discuss the potential benefits of using a NoSQL document store database for the organisation from Task 1. It is recommended that you use a software tool such as 'Screencast-o-matic' (which can be downloaded for free - https://screencast-o-matic.com/) or OBS (which can be downloaded for free - https://obsproject.com/) to record your screencast. **If your screencast exceeds ten minutes then only the first ten minutes will be viewed and assessed**.

You must use up-to-date academic research literature in your review, and provide correctly cited references (using the Harvard referencing system) within the Powerpoint slides. *Note there is no need to submit a copy of your Powerpoint slides.*

It is recommended that you include the following content in your Powerpoint slides:

- A title slide (one slide);
- Slides which provide an overview of NoSQL document stores (two slides maximum);
- Slides which provide a critical discussion of the benefits of NoSQL document stores (two slides maximum);
- A slide (one slide maximum) which gives a summary;
- Two slides maximum which contains your reference list in Harvard format.

Submission Requirements: You should submit your work as a single .ZIP archive

For Task 1 you will need to submit:

1. An **SQL script** file which can be run without error and:
   a. drops and then creates your tables (correctly ensuring that any referential integrity issues can be resolved)
   b. inserts sample data into every table; and
   c. creates the PL/pgSQL code described in part (d) below;
   d. includes calls to your PL/pgSQL stored procedures supplying sample test data. There is no requirement to prompt for user input, just supply test data as parameters to your procedure calls.

For Task 2 you will need to submit:

1. A **word** of **pdf** document which contains the answers for the user privileges

For Task 3 you will need to submit:

1. A voice over screencast in **.mp4** format.

This assignment should be submitted no later than the deadline advised on Canvas.

The mark scheme and breakdown is shown on the following page

CETM75 Assignment 2

Name: **Total Mark:** /65

	Max pts	**Mark**				
		Not Attempted	**Major Issues**	**Minor Issues**	**Satisfactory Attempt**	**Excellent Attempt**
TASK 1 – Millie's Musical Emporium (35/65)						
Table Creation and Sample Data Insertion	**10**	**0** Not attempted or very poor attempt.	**3** Poor match to ERD and data dictionary. Script will not work in PostgreSQL correctly. Poor or no sample data. PL/pgSQL and procedure calls not included	**5** Minor issues in match to ERD and data dictionary. Some problems with script. Reasonable sample data. PL/pgSQL code included but no PL/pgSQL procedure calls.	**7** Good match to ERD and data dictionary. Works completely in PostgreSQL without error. Good sample data. PL/pgSQL code and PL/pgSQL procedure calls included.	**10** Very good match to ERD and data dictionary. Would work completely in PostgreSQL without error. Excellent sample data created. PL/pgSQL code and PL/pgSQL procedure calls included.
PL/pgSQL Task b(i)	**12**	**0** Not attempted or very poor attempt.	**4** PL/pgSQL code developed does not run and does not match requirement. No test examples.	**6** PL/pgSQL code runs or contains only a small number of errors. Close fit to requirement. Test examples given but need more thought about sample data.	**10** PL/pgSQL code runs and matches requirement and some error checking and data generation incorporated. Good use of test examples to demonstrate code running.	**12** Excellent code developed with excellent use of error checking and automated generation of data values. Very good set of test examples to demonstrate code running with both valid and invalid data.

PL/pgSQLTask b(ii)	**13**	**0** Not attempted or very poor attempt.	**4** PL/pgSQL code developed does not run and does not match requirement. No test examples.	**6** PL/pgSQL code runs or contains only a small number of errors. Close fit to requirement. Test examples given but need more thought about sample data.	**11** PL/pgSQL code runs and matches requirement and some error checking and data generation incorporated. Good use of test examples to demonstrate code running.	**13** Excellent code developed with excellent use of error checking and automated generation of data values. Very good set of test examples to demonstrate code running with both valid and invalid data.
Task 2 – Privileges (10/65)						
Privileges	**10**	**0** Not attempted or very poor attempt.	**3** Missing user privileges for some tables and lack of discussion of these choices.	**5** Some good choices made for user privileges. Not all tables covered and lack of discussion.	**7** Privileges for all tables and users covered. Satisfactory discussion of choices made.	**10** Privileges for all tables and users covered and accurately discussed.
TASK 3 – Screencast (20/65)						
Screencast	**20**	**0** Not attempted or very poor attempt.	**5** Limited or no critical comparison of NoSQL databases, with no link to the scenario. No benefits discussed. No, or limited, use of references. Poor presentation.	**10** Some good critical discussion of NoSQL databases, some discussion in context of the scenario. Report does not adequately discuss benefits. Some use of academic referencing with a reference list provided but not necessarily in Harvard format. Good presentation but could be improved in some areas.	**15** Good critical discussion of NoSQL databases with clear relevance to the scenario. Good critical discussion of benefits of NoSQL databases. Good use of academic referencing used and reference list correctly supplied in Harvard format. Good presentation in most areas.	**20** Excellent critical discussion of NoSQL with excellent discussion relevant to the scenario. Report is focused and has excellent discussion of the benefits of NoSQL databases. Very good use of academic referencing and a reference list of high quality sources is provided in Harvard format. Excellent presentation.