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 Program 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 the use of an electronic plagiarismdetection 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 Program Guide).

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

There are two tasks for this assessment:

Task 1

Scenario

Introduction

Millie'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 computerize its customer management and stock recording system. You have been tasked with developing a database application to meet their needs.

Current Position

Currently, MME Ltd records 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 everytransaction 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, and 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.

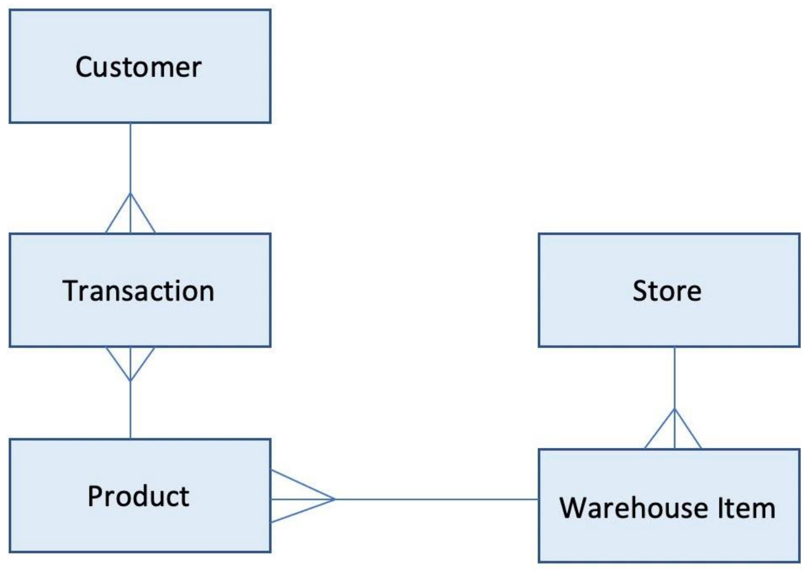
The organization would also like to be able to create management reports, which may, for example, show all sales at a particular store or a group of stores, between certain dates.

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 that 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 development:

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

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

When developing the system you should take into account the important developmentissues 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 theywould 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. writethe relevant GRANT commands for each table and provide a brief explanation.

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 referentialintegrity 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 dataas parameters to your procedure calls.

For Task 2 you will need to submit:

1. A **word** or **pdf** document that contains the answers for the user privileges.