N5 DDD Clydeview

Tasks

[Task 1 Database Structure 1-1](#_Toc183177672)

[Task 2 Queries Introduction 2-1](#_Toc183177673)

[Task 3 Sorting 3-1](#_Toc183177674)

[Task 4 SQL – Introduction 4-1](#_Toc183177675)

[Task 5 Queries 5-1](#_Toc183177676)

[Task 6 Query Design (Simple) 6-1](#_Toc183177677)

[Task 7 Query Implementation 7-1](#_Toc183177678)

[Task 8 Query Design (Complex) 8-1](#_Toc183177679)

[Task 9 Query Implementation 9-1](#_Toc183177680)

[Task 10 Entity Design 10-1](#_Toc183177681)

[Task 11 Entity Design 11-1](#_Toc183177682)

[Task 12 Entity Design 12-1](#_Toc183177683)

[Task 13 Entity Design 13-1](#_Toc183177684)

[Task 14 Entity and Relationship Design 14-1](#_Toc183177685)

[Task 15 Create Tables 15-1](#_Toc183177686)

[Task 16 Entity and Relationship Design 16-1](#_Toc183177687)

[Task 17 Create Tables – Done 17-1](#_Toc183177688)

[Task 18 Entity and Relationship Design 18-1](#_Toc183177689)

[Task 19 Create Tables 19-1](#_Toc183177690)

[Task 20 Entity and Relationship Design 20-1](#_Toc183177691)

[Task 21 Create Tables 21-1](#_Toc183177692)

[Task 22 Query Design 22-1](#_Toc183177693)

[Task 23 Query Implementation 23-1](#_Toc183177694)

[Task 24 Query Design 24-1](#_Toc183177695)

[Task 25 Query Implementation 25-1](#_Toc183177696)

[Task 26 Modify Data – Introduction 26-1](#_Toc183177697)

[Task 27 Modify Data – Design 27-1](#_Toc183177698)

[Task 28 Modify Data – Implementation 28-1](#_Toc183177699)

[Task 29 Modify Data – Design 29-1](#_Toc183177700)

[Task 30 Modify Data – Implementation 30-1](#_Toc183177701)

[Task 31 Testing 31-1](#_Toc183177702)

[Task 32 Evaluation 32-1](#_Toc183177703)

[Task 33 Testing 33-1](#_Toc183177704)

[Task 34 Evaluation 34-1](#_Toc183177705)

**NB**. Each task is a separate section to assist with printing individual tasks.

Answers

[A1 Database Structure 1-1](#_Toc183177706)

[A2 Queries Introduction 2-1](#_Toc183177707)

[A3 Sorting 3-1](#_Toc183177708)

[A4 SQL – Introduction 4-1](#_Toc183177709)

[A5 Queries 5-1](#_Toc183177710)

[A6 Query Design (Simple) – Done 6-1](#_Toc183177711)

[A7 Query Implementation – Done 7-1](#_Toc183177712)

[A8 Query Design (Complex) – Done 8-1](#_Toc183177713)

[A9 Query Implementation – Done 9-1](#_Toc183177714)

[A10 Entity Design – Done 10-1](#_Toc183177715)

[A11 Entity Design – Mostly done 11-1](#_Toc183177716)

[A12 Entity Design 12-1](#_Toc183177717)

[A13 Entity Design 13-1](#_Toc183177718)

[A14 Entity and Relationship Design 14-1](#_Toc183177719)

[A15 Create Tables – Done 15-1](#_Toc183177720)

[A16 Entity and Relationship Design – Part 16-1](#_Toc183177721)

[A17 Create Tables 17-1](#_Toc183177722)

[A18 Entity and Relationship Design – Mostly 18-1](#_Toc183177723)

[A19 Create Tables 19-1](#_Toc183177724)

[A20 Entity and Relationship Design – Mostly 20-1](#_Toc183177725)

[A21 Create Tables 21-1](#_Toc183177726)

[A22 Query Design – Done 22-1](#_Toc183177727)

[A23 Query Implementation – Done 23-1](#_Toc183177728)

[A24 Query Design – Done 24-1](#_Toc183177729)

[A25 Query Implementation – Mostly 25-1](#_Toc183177730)

[A26 Modify Data – Introduction 26-1](#_Toc183177731)

[A27 Modify Data – Design 27-1](#_Toc183177732)

[A28 Modify Data – Implementation 28-1](#_Toc183177733)

[A29 Modify Data – Design 29-1](#_Toc183177734)

[A30 Modify Data – Implementation 30-1](#_Toc183177735)

[A31 Testing 31-1](#_Toc183177736)

[A32 Evaluation 32-1](#_Toc183177737)

[A33 Testing 33-1](#_Toc183177738)

[A34 Evaluation 34-1](#_Toc183177739)

# Database Structure

## Part 1

Read the information below about fields and field types.

### Fields

A database **field** is one single item of information in a database. Each heading in a database table identifies a field.

A database about pupils would store fields such as name, address and date of birth.



### Field Types

When setting up a database you need to think carefully about what type of data each field is going to store. The **field type** determines the type of data that can be held in a field.

Examples of field types include: text, number, date, time, graphic, link and Boolean.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Description** | **Example** |
| Text | Used for storing characters or combinations of characters and numbers | Peter Parker, PA19 1UX |
| Number | Used for storing numbers | 34, 8.99, -26 |
| Date | Used for storing dates | 23/7/2007 |
| Time | Used for storing time values | 11:37am |
| Boolean | Used for storing True/False or Yes/No values | True, False, Yes, No |

## Part 2

Look at the database table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| KnownAs | Surname | Branch | AccountType | Balance(£) | Rate(%) | InterestDue |
| Leigh | Nelson | Glasgow | Saver | 100.67 | 1.7 | 01/02/2024 |
| Sam | McAteer | Glasgow | Flexi | 304.86 | 2.3 | 02/03/2024 |
| Robyn | Jones | Gourock | ISA | 89.45 | 3.0 | 09/03/2024 |
| Jack | Devlin | Greenock | Saver | 162.91 | 1.7 | 29/01/2024 |

1. How many fields are in the database table above?

|  |
| --- |
|  |

1. How many records are in the database table above?

|  |
| --- |
|  |

1. For each field, write the field name and the data type used.

|  |  |
| --- | --- |
| **Field Name** | **Type** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. You have been asked to create a database to store information about pupils in this class. Complete the shaded cells of the table, adding your own fields and data types. All sensible answers are correct.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Forename | Surname | Eye Colour |  | Year |  |  |
| Text | Text | Text | Date |  | Boolean |  |

1. You have been asked to create a database to store information about products in a shop. Complete the table, adding your own fields and data types. All sensible answers are correct.

|  |  |
| --- | --- |
| **Field name** | **Field type** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Queries Introduction

## Part 1

Read the following information carefully.

What is a query?

A query is used to search or sort a database table. Queries allow the user of a database to quickly find the records that they are looking for or arrange the records into a particular order. Being able to perform queries is a major advantage of using an electronic database over a non-electronic database. Queries are commonly used to perform two tasks: **search** and **sort**.

### Search Queries

A search query is performed to retrieve all the records which **match** particular **search criteria**. A simple search uses only one criterion, while a complex search uses two or more criteria.

Below is an example of a simple search query to find all the people in the database table with the first name John.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database Table** | | **Query** | **Result** | |
| **Forename** | **Surname** | Find all the people with the first name John | **Forename** | **Surname** |
| Joanna | Keetings | John | Brown |
| John | Brown |  | |
| James | Smith |

The result of a search query will return only the records which match the criteria given.

### Sort Queries

A sort query is performed to arrange all the records in the database table into a particular order. There are two different orders that can be used: **ascending****order** and **descending****order**.

**Ascending** **Order** – arranges the records in order from smallest 🡪 largest or alphabetically from A 🡪 Z.

**Descending** **Order** – arranges the records in order from largest 🡪 smallest or in reverse alphabetical order from Z 🡪 A.

## Part 2

A CRIMINAL table in the PRISON database is used to store details of criminals. The tables has 8 fields shown in the sample record below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **firstName** | **surname** | **dob** | **age** | **crime** | **sentence** | **eyeColour** | **hairColour** |
| Patrick | Grant | 19/4/86 | 29 | Burglary | 2 years | Blue | Brown |

**Example**: Find all the criminals named Brian.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_*Brian*\_\_\_\_\_  in the field \_\_\_\_*firstName*\_\_\_\_. |

Complete each of the following sentences by providing the missing search criteria and fields for each query.

1. Find all the criminals with black hair.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with the surname Smith.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals aged 35.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with brown eyes.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with blonde hair and green eyes.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_  AND contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals named Steven and are aged 19.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_  AND contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Sorting

## Part 1

1. For each set of data below, state whether it has been arranged in ascending order or descending order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Set 1** | **Set 2** | **Set 3** | **Set 4** |
| Data | 12  15  19  27  34  56 | Water  Socks  Meat  Food  Coffee  Ball | 125  113  97  64  14  5 | Minister  Mister  Monster  Month  Mother  Mouth |
| Order |  |  |  |  |

## Part 2

1. For each table below, identify the field that has been used to sort the table and state the type of sort has been applied to the field.

### Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Andy | Smith | 25 | Blue |
| Chris | Holdsworth | 27 | Red |
| Ben | Adamson | 30 | Green |
| Fred | Jackson | 36 | Orange |

Field used to sort the table:

Order applied:

### Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Andy | Smith | 25 | Blue |
| Ben | Adamson | 30 | Green |
| Chris | Holdsworth | 27 | Red |
| Fred | Jackson | 36 | Orange |

Field used to sort the table:

Order applied:

### Table 3

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Ben | Adamson | 30 | Green |
| Chris | Holdsworth | 27 | Red |
| Fred | Jackson | 36 | Orange |
| Andy | Smith | 25 | Blue |

Field used to sort the table:

Order applied:

## Part 3

The database tables below have been sorted using a complex sort. This is where a sort has been applied to two different fields.

1. For each database table, state which fields has been sorted and the type of sort that has been applied.

### Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| **First Name** | **Surname** | **Year** | **Class** |
| James | Cook | S1 | Bute |
| Rebecca | Bottle | S1 | Mull |
| Leon | Smith | S1 | Skye |
| Amy | Smith | S2 | Bute |
| Rebecca | Smith | S2 | Skye |
| Patrick | Adams | S3 | Skye |

1st sort:

2nd sort:

### Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| **First Name** | **Surname** | **Year** | **Class** |
| Patrick | Adams | S3 | Skye |
| Rebecca | Bottle | S1 | Mull |
| James | Cook | S1 | Bute |
| Leon | Smith | S2 | Skye |
| Rebecca | Smith | S1 | Skye |
| Amy | Smith | S2 | Bute |

1st sort:

2nd sort:

# SQL – Introduction

If you are setting up a query directly to a database, the coding that goes on behind the scenes will most likely use a formal language called **SQL** (Structured Query Language) to set up the query.

Below is an example of a query written in SQL:

1 SELECT title, introduced, RAM

2     FROM Computer

3     WHERE RAM = 512

4     ORDER BY title ASC;

This query will display the fields title, introduced and RAM from the database table Computer for any record where the RAM field is equal to 512. The records will then be sorted into ascending order of the title field.

Although SQL is a coding language, it doesn’t follow a linear sequence. The instructions are processed in the following order, which is different from the written order:

FROM – which database table is required?

WHERE – what criteria must be met?

SELECT – which fields should be displayed?

ORDER BY – how should the fields be sorted?

## Part 1

1. What does SQL stand for?

|  |
| --- |
|  |

1. Number the following steps into the correct order from 1 to 4, as they would be used in a SQL statement.

|  |  |
| --- | --- |
| **Step** | **SQL Clause** |
|  | FROM |
|  | ORDER BY |
|  | SELECT |
|  | WHERE |

1. Draw lines to match the SQL keywords to the correct description.

|  |  |  |
| --- | --- | --- |
| **SQL Clause** |  | **Description** |
| SELECT |  | The criteria which must be met |
| FROM |  | The fields to display |
| WHERE |  | The field and order used to sort the results |
| ORDER BY |  | The table used in the query |

## Part 2

1. Read each SQL statement below and fill in the blanks to complete the explanation.

### Query 1

|  |
| --- |
| 1 SELECT firstName, surname, age  2     FROM Pupil  3     WHERE age > 16  4     ORDER BY firstName ASC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

### Query 2

|  |
| --- |
| 1 SELECT itemName, price, quantity  2     FROM Stock  3     WHERE price < 9.99  4     ORDER BY price DESC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

### Query 3

|  |
| --- |
| 1 SELECT surname, address, postCode  2     FROM Customer  3     WHERE title = "Mr"  4     ORDER BY dob ASC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

# Queries

## Tasks

Read each SQL statement below and fill in the blanks to complete the explanation.

### Query 1

|  |
| --- |
| 1 SELECT firstName, surname, age  2     FROM Pupil  3     WHERE age > 16 AND surname = "Smith"  4     ORDER BY firstName ASC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

### Query 2

|  |
| --- |
| 1 SELECT itemName, price, quantity  2     FROM Stock  3     WHERE price < 9.99 OR quantity > 5  4     ORDER BY price DESC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

### Query 3

|  |
| --- |
| 1 SELECT surname, address, postCode  2     FROM Customer  3     WHERE title = "Mr" AND age > 50  4     ORDER BY dob ASC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

# Query Design (Simple)

The Customer table stores details of the customers of a mobile phone company. The table has the 8 fields listed below.



Before queries are written, they are designed by stating the tables, conditions, fields, and order that will need to used. Design queries to perform each of the following tasks.

1. Search the database to display the full name (forename and surname) of the customer with the surname "Rice".

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to display the full name and town of all customers who live in Inverkip.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to display the surname, package and town of all the customers who are on the large package.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to display the surname, town and street of all the customers who live in Port Glasgow.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the forename, surname and town of all of the customers and sort these details into alphabetical order of surname.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the forename, surname and paymentDueDate of all the customers and sort these details into descending order of paymentDueDate.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the forename, surname and customerNo of all the customers with a customerNo greater than 110. Sort the details into alphabetical order of forename.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to display the full name and town of all the customers who live in Kilmacolm. Sort the details into alphabetical order of surname.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

# Query Implementation

Open the Clydeview.db database file. Screen snip the evidence, SQL and output, for the tasks below.

Use the query designs from to help create the SQL statement for each task.

1. Search the database to display the full name (forename and surname) of the customer with the surname "Rice". What is their first name?

|  |
| --- |
|  |

1. Search the database to display the full name and town of all customers who live in Inverkip.

|  |
| --- |
|  |

1. Search the database to display the surname, package and town of all the customers who are on the large package.

|  |
| --- |
|  |

1. Search the database to display the surname, town and street of all the customers who live in Port Glasgow.

|  |
| --- |
|  |

1. Display the forename, surname and town of all of the customers and sort these details into alphabetical order of surname. Who appears at the top of the list?

|  |
| --- |
|  |

1. Display the forename, surname and paymentDueDate of all the customers and sort these details into descending order of paymentDueDate. Who appears at the top of the list?

|  |
| --- |
|  |

1. Display the forename, surname and customerNo of all the customers with a customerNo greater than 110. Sort the details into alphabetical order of forename. Who appears at the top of the list?

|  |
| --- |
|  |

1. Search the database to display the full name and town of all the customers who live in Kilmacolm. Sort the details into alphabetical order of surname. Who appears at the top of the list?

|  |
| --- |
|  |

# Query Design (Complex)

The SuperHero table stores details of movie characters. Each of the characters has played a role in a super hero movie. The table has 8 fields shown below.



Design SELECT queries to perform each of the following tasks.

1. Search the database to find a character who played the role of 'Super Hero' with 'Acrobatics' as his/her main ability. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Henchman' whose main ability is 'Strength'. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' whose main ability is 'magic' and whose origin of power is 'training'. Display the character’s name, role, main ability and origin of power.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Hero' with the main ability ‘Magic’ whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' whose origin of power is 'Chemicals' and who has 'Gadgets' as ability 2. Display the character’s name, role, ability2 and origin of power.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Team member' and has the main ability ‘Water breathing’. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Hero' who has the main ability 'Flight', ability 2 'Super-strength' and origin of power 'Technology'. Display the character’s name, role, main ability, ability2 and origin of power.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' with the main ability 'Intelligence' and ability 2 'Martial arts'. Display the character’s name, role, main ability and ability2.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| Search criteria |  |
| Sort order |  |

# Query Implementation

Open the Clydeview.db database file. Paste the SQL and screen snip the output, for the tasks below.

Use the query designs from Task 8 to help create the SQL statement for each task.

1. Search the database to find a character who played the role of 'Super Hero' with 'Acrobatics' as his/her main ability. Display the character’s name, role and main ability.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Henchman' whose main ability is 'Strength'. Display the character’s name, role and main ability.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Super Villain' whose main ability is 'Magic' and whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Super Hero' with the main ability ‘Magic’ whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Super Villain' whose origin of power is 'Chemicals' and who has 'Gadgets' as ability 2. Display the character’s name, role, ability2 and origin of power.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Team member' and has the main ability 'Water breathing'. Display the character’s name, role and main ability.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Super Hero' who has the main ability 'Flight', ability 2 'Super-strength' and origin of power 'Technology'. Display the character’s name, role, main ability, ability2 and origin of power.

|  |
| --- |
|  |

1. Search the database to find a character who played the role of 'Super Villain' with the main ability 'Intelligence' and ability 2 ‘Martial arts’. Display the character’s name, role, main ability and ability2.

|  |
| --- |
|  |

# Entity Design

SurfScotland is a blog used by members to share information about surfing in Scotland. A relational database is used to store details of members and blog posts in two related tables called Member and Post.

* Members must register with SurfScotland and provide an email address before they are allowed to add posts to the blog
* Members must be aged 18 or over
* The number of words in each post is restricted to between 20 and 250 words

Sample data stored in each table is shown below.

### Member Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Member ID** | **Last Name** | **First Name** | **Age in Years** | **Email** |
| 1 | Davies | Jim | 27 | jimbo31@scotmail.co.uk |
| 2 | McKay | Ann | 28 | mckaya218@hotmail.com |
| 3 | Roberts | Carol | 35 | croberts123@teachers.com |
| 4 | Singh | Hardeep | 24 | singh832@scotmail.co.uk |

### Post Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Post ID** | **Title** | **Date** | **Member ID** | **Number of Words** |
| 1 | Welcome to the SurfScotland blog | 2016/08/01 | 1 | 228 |
| 2 | Belhaven Bay Dunbar | 2016/08/08 | 1 | 176 |
| 3 | Coldingham Bay Scottish Borders | 2016/08/13 | 1 | 58 |
| 4 | Hebridean Surf Lewis | 2016/08/15 | 2 | 145 |
| 5 | Broch Open Surf Competition | 2016/08/15 | 4 | 73 |

### Member entity

1. Complete the data dictionary for the Member entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Post entity

1. Complete the data dictionary for the Post entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Member and Post entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Member and Post entities.

|  |
| --- |
|  |

# Entity Design

MyPhotoSpace is an online photo gallery stores details of photos displayed on the site in two separate linked tables called Album and Photo.

To minimise data entry errors, MyPhotoSpace applies the following restrictions:

* Each album can store a maximum of 120 photos
* Five different categories of album are available on the gallery: animals, cars, castles, surfing and towns

Sample data stored in each table is shown below.

### Album Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Album ID** | **Name** | **Category** | **Description** | **Number of Photos** |
| 121 | BMW Cars | Cars | Photos of BMW cars | 25 |
| 122 | Glenrothes | Towns | Photos from around Glenrothes | 4 |
| 123 | Scottish Castles | Castles | Photos of Scottish castles | 17 |

### Photo Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Photo ID** | **Album ID** | **Title** | **Image** |
| 23 | 122 | Thirsty Hippos | hippos\_pmckay.jpg |
| 24 | 122 | Glenrothes Irises | irises\_mharris.jpg |
| 31 | 123 | Newark Castle at Night | newark\_at\_night.png |
| 32 | 122 | Pond at Riverside Park | riverside\_park\_pong.jpg |

### Album entity

1. Complete the data dictionary for the Album entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Photo entity

1. Complete the data dictionary for the Photo entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Album and Photo entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Album and Photo entities.

|  |
| --- |
|  |

# Entity Design

The RetroClothing website uses a relational database to store details of items of women’s clothing for sale and the brand of each item in two separate tables called Item and Brand.

To minimise data entry errors, RetroClothing applies the following restrictions:

* The nationality of the brands used in the website are American, British or Italian
* The eras featured on the site are 1940, 1950s, 1960s and 1970s
* Item codes all have 7 characters
* Item size should be limited to 8, 10, 12, 14 and 16

Sample data stored in each table is shown below.

### Item Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item Code** | **Description** | **Size** | **Era** | **Brand ID** |
| RSS1001 | Red swim suit | 10 | 1950s | B3 |
| FDP1002 | Floral dungarees playsuit | 10 | 1990s | B2 |
| BSC2103 | Brown swing coat | 16 | 1960s | B5 |
| CSP3204 | Circle skirt black white polka dot | 12 | 1950s | B4 |
| FPD3225 | Floral print hostess dress | 10 | 1970s | B5 |

### Brand Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Brand ID** | **Brand** | **Year Established** | **Nationality** |
| B1 | Valentino | 1965 | Italian |
| B2 | Mary Quant | 1970 | British |
| B3 | Rose Marie Reid | 1946 | American |
| B4 | Elmoor |  | British |
| B5 | Susan Small | 1942 | British |

### Item entity

1. Complete the data dictionary for the Item entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Brand entity

1. Complete the data dictionary for the Brand entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Item and Brand entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Item and Brand entities.

|  |
| --- |
|  |

# Entity Design

The SportsStats website uses a relational database to store details of Scottish sports teams and players in two separate tables called Player and Team.

* All players have a unique Player ID that has 4 characters
* Players are given a star rating between 1 and 5
* The website features a limited number of sports: basketball, handball, hockey and netball

Sample data stored in each table is shown below.

### Player table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Player ID** | **Team ID** | **Star Rating** | **First Name** | **Last Name** | **Date of Birth** |
| L18C | 111 | 4 | Lindy | Osborne | 14/01/1990 |
| F19F | 113 | 3 | Fred | Freddricks | 30/07/1987 |
| Y01D | 131 | 5 | Yasmine | Davies | 22/11/1992 |

### Team table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team ID** | **Team Name** | **Sport** | **Manager** | **Coach** | **Home Town** |
| 111 | West Stars | Hockey | Chris Davidson |  | Paisley |
| 112 | Killie Shooters | Basketball |  | Liz Smillie | Ardrossan |
| 113 | Jumpin Jacks | Basketball | Dave Ford | Ali Mustapha | Dunbar |

### Player entity

1. Complete the data dictionary for the Player entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Team entity

1. Complete the data dictionary for the Team entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Player and Team entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Team and Player tables.

|  |
| --- |
|  |

# Entity and Relationship Design

## Part 1 – Design

The ClydeVet veterinary practice uses a relational database called ClydeVet to store details about pets and the pet owners.

The details of pet owners in a table called Owner.

Sample data stored in the Owner table is shown below.

### Owner Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Owner ID** | **First Name** | **Surname** | **Address** | **Town** | **Contact Telephone** |
| 1277 | Hardeep | Singh | 65 Iona Way | Greenock | 01475255707 |
| 2356 | Sally | Chan | 142 Main Street | Greenock | 01475242499 |
| 3510 | Elaine | Bryce | 29 Clyde Drive | Gourock | 01475636321 |
| 3821 | Cameron | Gray | 17 Shuttle Street | Gourock | 01475312245 |

### Owner Entity

1. Complete the data dictionary for the Owner entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 2 – Design

The practice specialises in the treatment of certain types of pet:

* Cat
* Dog
* Budgie
* Gerbil
* Tortoise

Details of any pets treated by the practice are stored in the ClydeVet database in a second table called Pet. This table is linked to the Owner table using a foreign key.

Sample data stored in the Pet table is shown below.

### Pet Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pet Code** | **Pet Name** | **Pet Type** | **Date of Birth** | **Received Vaccination?** | **Owner ID** |
| P0123 | Misty | Cat | 23/04/2012 | True | 2356 |
| P0345 | Rover | Dog | 12/12/2010 | True | 3821 |
| P0887 | Foggy | Cat | 23/01/2012 | True | 1277 |
| P1559 | Gladys | Gerbil | 16/04/2010 | False | 1277 |
| P1985 | Slinky | Tortoise | 24/09/2016 | False | 3510 |
| P2233 | Speedy | Tortoise | 09/06/2013 | True | 1277 |

### Pet Entity

1. Complete the data dictionary for the Pet entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 3 – ERD

1. Draw an entity relationship diagram to represent the relationship between the Owner and Pet entities.

|  |
| --- |
|  |

# Create Tables

Open the N5 DDD Clydeview Blank project in Replit.

1. Create the Owner table, and make sure that the properties of the Owner table match all of the settings indicated in your data dictionary.
2. Create the Pet table, and me sure that the properties of the Pet table match all of the settings indicated in your data dictionary.
3. Enforce referential integrity between the two tables.
4. Paste below, evidence to show that the table have been created.

# Entity and Relationship Design

A hardware store uses a relational database called HardwareStore to store details of the products for sale and the manufacturer of each product.

## Part 1 – Design

A sample record from the Product table of the HardwareStore database is shown below.

|  |
| --- |
| Product Name: Medium Paint Brush (Size 2)  Product Code: MPB2  Number in Stock: 24  On Order: No  Cost Price: £5.65  Manufacturer ID: 531 |

The store applies a business rule to ensure that the Number in Stock is always between 0 and 50. All costs in the database must be over £1.

### Product Entity

1. Complete the data dictionary for the Product entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 2 – Design

The Manufacturer table of the HardwareStore database is used to store details of manufacturers. Sample details stored in this table are shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Manufacturer ID** | **Name** | **Address** | **Telephone Number** |
| 531 | Metal and Wood | Tyne Way Newcastle | 01542123485 |

### Manufacturer Entity

1. Complete the data dictionary for the Manufacturer table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 3 – ERD

1. Draw an entity relationship diagram to represent the relationship between the Product and Manufacturer entities.

|  |
| --- |
|  |

# Create Tables – Done

Open the N5 DDD Clydeview Blank project in Replit.

1. Create the Manufacturer table, and make sure that the properties match all of the settings indicated in your data dictionary.
2. Create the Product table, and make sure that the properties match all of the settings indicated in your data dictionary.
3. Enforce referential integrity between the two tables.
4. Paste below, evidence to show that the table have been created.

# Entity and Relationship Design

## Part 1 – Design

An online music store uses a relational database called MusicStore to store details of CDs and their music labels.

The CD table is used to store details of the CDs while the Label table is used to store details of music labels. Sample data stored in each table is shown below.

### CD Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CD Code** | **Title** | **Artist** | **Label** | **Number of Tracks** | **Cost (£)** | **Genre** |
| 95VW | Grrr | The Rolling Stones | Polydor Records | 51 | 11.99 | R&R |
| 5J8Y | + | Ed Sheeran | Atlantic Records | 13 | 10.00 | Indie |
| 82FH | The Power of Love | Sam Bailey | Syco Music | 11 | 7.50 | Soul |
| 9KYX | Glory Days | Little Mix | Syco Music | 20 | 9.99 | R&B |

### Label Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Founded** | **Parent Company** | **Country of Origin** | **Website** |
| Syco Music | 2002 | Syco | UK | www.sycoentertainment.com |
| Capital Records | 1942 | Capital Music Group | USA | www.capitalrecords.com |
| Polydor Records | 1924 | Universal Music Group | Germany | www.polydor.co.uk |

The music stores applies the following business rules to the data stored in the database:

* The genre of a CD can be one of: Choral, Country, Garage, Indie, Opera, Pop, R&B, R&R, Soul
* The number of tracks on each CD must be between 10 and 60 inclusive
* The cost of each CD must be between 6.99 and 15.00 inclusive
* The country of origin for each label is one of: Germany, Japan, UK or USA

### CD Entity

1. Complete the data dictionary for the CD entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Label Entity

1. Complete the data dictionary for the Label entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 2 – ERD

1. Draw an entity relationship diagram to represent the relationship between the CD and Label entities.

|  |
| --- |
|  |

# Create Tables

1. Open the N5 DDD Clydeview Blank project in Replit.
2. Create the CD and Label tables, and make sure that the properties match all of the settings indicated in your data dictionaries.
3. Remember to enforce referential integrity between the two tables.
4. Paste below, evidence to show that the table have been created.

# Entity and Relationship Design

## Part 1 – Design

Clydeview Library uses a relational database called BookData to store details of books and authors in two tables called Book and Author.

Sample data stored in the Author table is shown below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author Ref** | **First Name** | **Surname** | **Nationality** | **DOB** | **Website** |
| 2864 | Kenneth | Oppel | Canadian | 31/08/1967 | www.kennethoppel.ca |
| 3061 | Dave | Eggers | American |  |  |
| 3197 | Joanne | Rowling | British | 31/07/1965 | www.jkrowling.com |

Sample data stored in the Book table is shown below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Genre** | **Title** | **Author Ref** | **Publisher** | **ISBN** | **Date of Publication** | **Number of Pages** |
| Child | Fantasy | Galactic Snapshots | 2864 | Puffin | 0140373683 | 03/08/2010 | 96 |
| Child | Mystery | Harry Potter and the Chamber of Secrets | 3197 | Bloomsbury | 0747538492 | 02/07/1998 | 251 |
| Adult | Fiction | The Casual Vacancy | 3197 | Little Brown Company | 0751552860 | 27/09/2012 | 503 |

The library applies the following business rules to the data stored in the database:

* The genre of a book in the Book Database can be one of: Autobiography, Fantasy, Fiction, Joke, Mystery, Fiction, Thriller
* The category of a book must be one of: Adult or Child
* The number of pages in each book must be between 32 and 950 inclusive

### Author Entity

1. Complete the data dictionary for the Author entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Book Entity

1. Complete the data dictionary for the Book entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 2 – ERD

1. Draw an entity relationship diagram to represent the relationship between the Book and Author entities.

|  |
| --- |
|  |

# Create Tables

1. Open the N5 DDD Clydeview Blank project in Replit.
2. Create the Author and Book tables, and make sure that the properties match all of the settings indicated in your data dictionaries.
3. Remember to enforce referential integrity between the two tables.
4. Paste below, evidence to show that the table have been created.

# Query Design

ClydeVet veterinary practice uses a relational database store details pets and their owners in two separate tables called Owner and Pet. The structure of the tables is shown below.

A black line with a black line

Description automatically generated with medium confidence

Design SELECT queries to perform each of the following tasks.

1. List the full name and address, and name of their pets, for all of the cat owners.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the full name and contact telephone number, with the codes of their pets, for all tortoise owners.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the full name and address, and the name of their pets, of all owners who have pets that have not yet received their vaccinations.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the name, type of each pet and town of any pet whose owner lives in Greenock.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the name, vaccination details and contact telephone number of their owner, of all pets whose owners live in Gourock.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the full names of all of the owners with the name and type of their pets. These details should be displayed in alphabetical order of owner surname.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the name and full address (inc town) of each pet. These details should be arranged in alphabetical order of town; pets who live in the same town should be arranged in alphabetical order of pet type.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the name, pet type, town and dateOfBirth of all the cats; details of the youngest pet should be displayed first.

|  |  |  |
| --- | --- | --- |
| Field(s) |  |  |
| Table(s) |  |  |
| Search criteria |  |  |
| Sort order |  |  |

# Query Implementation

Use Repl.it to fork the Team template for the task. Screen snip the required information to produce the evidence for the tasks below.

Use your query designs from Part 1 to help create the SQL statement for each task.

1. List the full name and address, and name of their pets, for all of the cat owners.

|  |
| --- |
|  |

1. List the full name and contact telephone number, with the codes of their pets, for all tortoise owners.

|  |
| --- |
|  |

1. List the full name and address, and the name of their pets, of all owners who have pets that have not yet received their vaccinations.

|  |
| --- |
|  |

1. List the name, type of each pet and town of any pet whose owner lives in Greenock.

|  |
| --- |
|  |

1. List the name, vaccination details and contact telephone number of their owner, of all pets whose owners live in Gourock.

|  |
| --- |
|  |

1. Display the full names of all of the owners with the name and type of their pets. These details should be displayed in alphabetical order of owner surname.

|  |
| --- |
|  |

1. Display the name and full address (inc town) of each pet. These details should be arranged in alphabetical order of town; pets who live in the same town should be arranged in alphabetical order of pet type.

|  |
| --- |
|  |

1. Display name, pet type, town and dateOfBirth of all the cats; details of the youngest pet should be displayed first.

|  |
| --- |
|  |

# Query Design

A hardware store uses a relational database to store details of the products for sale and the manufacturer of each product in two separate tables called Product and Manufacturer. The structure of the tables is shown below.

A black rectangular object with a black line

Description automatically generated

Design SELECT queries to perform each of the following tasks.

1. List the product name, cost price and manufacturer name of all products manufactured by Craft Supplies.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the manufacturer name, address, telephone number and cost price of any manufacturer who supplies products that cost more than £100.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the name of any manufacturers with the product names and cost price of any products that are on order.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the product name and manufacturer name of any products with 12 or more in stock.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. List the product names, prices and manufacturer name of all products manufactured by Tool Makers. These details should be displayed in increasing order of price.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the manufacturer name, address, on order details and number in stock information for products that are not on order. These details should be arranged in decreasing order of number in stock.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the product name, number in stock, cost price and manufacturer name of all products that cost less than £20. These details should be arranged in alphabetical order of manufacturer name; products from the same manufacturer should be displayed in alphabetical order of product name.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

1. Display the manufacturer details (name, address and telephone number) together with the product code and number in stock all products that have more than 2 in stock. These details should be displayed in alphabetical order of manufacturer; products from the same manufacturer should be displayed with the largest quantity in stock at the top.

|  |  |
| --- | --- |
| Field(s) |  |
| Table(s) |  |
| Search criteria |  |
| Sort order |  |

# Query Implementation

Open the populated database called Products and complete the queries below.

Use your query designs from Part 1 to help create the SQL statement for each task.

1. List the product name, cost price and manufacturer name of all products manufactured by Craft Supplies.

|  |
| --- |
|  |

1. List the manufacturer name, address, telephone number and cost price of any manufacturer who supplies products that cost more than £100.

|  |
| --- |
|  |

1. List name of any manufacturers with the product names and order details of any products that are on order.

|  |
| --- |
|  |

1. List the product name and manufacturer name of any products with 12 or more in stock.

|  |
| --- |
|  |

1. List the product names, prices and manufacturer name of all products manufactured by Tool Makers. These details should be displayed in increasing order of price.

|  |
| --- |
|  |

1. Display the manufacturer name, address, on order details and number in stock information for manufacturers who do not have outstanding orders. These details should be arranged in decreasing order of number in stock.

|  |
| --- |
|  |

1. Display the product name, number in stock, cost price and manufacturer name of all products that cost less than £20. These details should be arranged in alphabetical order of manufacturer name; products from the same manufacturer should be displayed in decreasing order of cost price.

|  |
| --- |
|  |

1. Display the manufacturer details (name, address and telephone number) together with the product code and number in stock all products that have more than 2 in stock. These details should be displayed in alphabetical order of manufacturer; products from the same manufacturer should be displayed with the largest quantity in stock at the top.

|  |
| --- |
|  |

# Modify Data – Introduction

A mobile phone company uses a database to store details of its customers in a table called Customer. Some of the data stored in this table is shown below.

A screen shot of a computer

Description automatically generated

Read each SQL query below and describe the effect that it will have on the contents of the Customer table above.

## Query 1 – SQL

1 INSERT INTO Customer

2     VALUES (116, "Nial", "Davies", "12 Clyde View",

3             "Gourock", "Premier", TRUE, "08/05/2017");

Query 1 – Description

|  |
| --- |
|  |

## Query 2 – SQL

1 UPDATE Customer

2     SET directDebit = TRUE

3     WHERE foreName = "Grant"

4       AND surname = "Donaldson";

Query 2 – Description

|  |
| --- |
|  |

## Query 3 – SQL

1 DELETE FROM Customer

2     WHERE customerNo = 111;

Query 3 – Description

|  |
| --- |
|  |

## Query 4 – SQL

1 UPDATE Customer

2     SET paymentDueDate = "2017-06-01";

Query 4 – Description

|  |
| --- |
|  |

## Query 5 – SQL

1 INSERT INTO Customer (foreName, surname, directDebit, package,  
                        paymentDueDate, street, town, customerNo)

2     VALUES( "Kelly", "Holmes", FALSE, "Standard",

              "2017-05-16", "5 Lime Grove", "Greenock", 126);

Query 5 – Description

|  |
| --- |
|  |

# Modify Data – Design

ClydeVet veterinary practice uses a relational database store details pets and their owners in two separate tables called Owner and Pet. The structure of the tables is shown below.

A black line with a black line

Description automatically generated with medium confidence

Design INSERT, DELETE and UPDATE queries to perform each of the following tasks.

Add the following details of Goldie the dog to the Pet table.

|  |
| --- |
| Pet Code: P4821  Name: Goldie  Type: Dog  Date of Birth: 26/10/2016  Received vaccination?: True  Owner ID: 3821 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 3905  First Name: Gary  Surname: Hughes  Address: 13 Juniper Place  Town: Wemyss Bay  Contact Telephone: 07998765432 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

|  |
| --- |
| Pet Code: P2751  Pet Name: Usain  Pet Type: Tortoise  Date of Birth: 28/10/2006  Received vaccination?: True |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 2664  First Name: Hannah  Surname: Black  Address: 47 High Road  Town: Greenock  Contact Telephone: 01475633633 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

|  |
| --- |
| Pet Code: P0438  Pet Name: Arnie  Pet Type: Budgie  Date of Birth: 13/03/2017  Received vaccination?: False |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove details of Slinky the tortoise from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove the details of the owner with Owner ID 3510 from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The date of birth of Gladys the gerbil has been stored incorrectly; it should be 16/10/2004. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Owner 3905 has a new pet cat called Bruno with date of birth 14/05/2017 and pet code P1678 (Bruno hasn’t had any vaccinations yet). Add Bruno’s details to the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove the details of the pet called Usain from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

# Modify Data – Implementation

Use Repl.it to fork the Team template for the task. Screen snip the required information to produce the evidence for the tasks below.

Use your query designs from Part 1 to help create the SQL statement for each task.

Add the details of Goldie the dog to the Pet table.

|  |
| --- |
| Pet Code: P4821  Name: Goldie  Type: Dog  Date of Birth: 26/10/2016  Received vaccination?: True  Owner ID: 3821 |

The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.

The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 3905  First Name: Gary  Surname: Hughes  Address: 13 Juniper Place  Town: Wemyss Bay  Contact Telephone: 07998765432 |

|  |
| --- |
| Pet Code: P2751  Pet Name: Usain  Pet Type: Tortoise  Date of Birth: 28/10/2006  Received vaccination?: True |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 2664  First Name: Hannah  Surname: Black  Address: 47 High Road  Town: Greenock  Contact Telephone: 01475633633 |

|  |
| --- |
| Pet Code: P0438  Pet Name: Arnie  Pet Type: Budgie  Date of Birth: 13/03/2017  Received vaccination?: False |

Remove details of Slinky the tortoise from the database.

The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.

Remove the details of the owner with Owner ID 3510 from the database.

The date of birth of Gladys the gerbil has been stored incorrectly; it should be 16/10/2004. Edit the correct record of the database.

Owner 3905 has a new pet cat called Bruno with date of birth 14/05/2017 and pet code P1678 (Bruno hasn’t had any vaccinations yet). Add Bruno’s details to the database.

Remove the details of the pet called Usain from the database.

# Modify Data – Design

A hardware store uses a relational database to store details of the products for sale and the manufacturer of each product in two separate tables called Product and Manufacturer. The structure of the tables is shown below.

A black rectangular object with a black line

Description automatically generated

Design INSERT, DELETE and UPDATE queries to perform each of the following tasks

1. Add details of this new product to the database.

**New Product**

Product Name: Spirit Level

Product Code: SPL3

Number In Stock: 14

On Order?: False

Cost Price(£): 17.99

Manufacturer ID: 531

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. The stock level of the product with Product Code MA16 has fallen to 1 and the product is now on order. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. The manufacturer called Tool Makers has moved. Its new address is: Unit 6, Avon Industrial Estate, Bath and its new phone number is: 01789334456. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove the saw with Product Code SW22 form the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 327  Manufacturer Name: CVA Group  Address: 35 Lomond Way Paisley  Telephone Number: 01414141414 |  | Product Name: Circular Saw  Product Code: CSW2  Number In Stock: 3  On Order?: False  Cost Price(£): 99.00  Manufacturer ID: 327 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 408  Manufacturer Name: Cabinet Makers  Address: 158 Hawthorn Road Carlisle  Telephone Number: 03217329124 |  | Product Name: 6 piece Chisel Set  Product Code: CSP6  Number In Stock: 8  On Order?: True  Cost Price(£): 43.51  Manufacturer ID: 408 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove all products manufactured by the manufacturer with ID 441 from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Manufacturer 327 has a new product called Tin Snips Left-handed and costs £9.67. The code of this new product is TSL1, there are 5 in stock and none on order. Add details of this new product to the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove the details of the manufacturer called Craft Supplies from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Increase the number in stock of all products by 2.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Reduce by £5 the cost of all products that are manufactured by the manufacturer with ID 627.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

# Modify Data – Implementation

Open the populated database called Products DB and complete the queries below.

Use your query designs from Task 1 to help create the SQL statement for each task.

1. Add details of this new product to the database.

**New Product**

Product Name: Spirit Level

Product Code: SPL3

Number In Stock: 14

On Order?: False

Cost Price(£): 17.99

Manufacturer ID: 531

|  |
| --- |
|  |

1. The stock level of the product with Product Code MA16 has fallen to 1 and the product is now on order. Edit the correct record of the database.

|  |
| --- |
|  |

1. The manufacturer called Tool Makers has moved. Its new address is: Unit 6, Avon Industrial Estate, Bath and its new phone number is: 01789334456. Edit the correct record of the database.

|  |
| --- |
|  |

1. Remove the saw with Product Code SW22 form the database.

|  |
| --- |
|  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 327  Manufacturer Name: CVA Group  Address: 35 Lomond Way Paisley  Telephone Number: 01414141414 |  | Product Name: Circular Saw  Product Code: CSW2  Number In Stock: 3  On Order?: false  Cost Price(£): 99.00  Manufacturer ID: 327 |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 408  Manufacturer Name: Cabinet Makers  Address: 158 Hawthorn Road Carlisle  Telephone Number: 03217329124 |  | Product Name: 6 piece Chisel Set  Product Code: CSP6  Number In Stock: 8  On Order?: true  Cost Price(£): 43.51  Manufacturer ID: 408 |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. Remove all products manufactured by the manufacturer with ID 441 from the database.

|  |
| --- |
|  |

1. Manufacturer 327 has a new product called Tin Snips Left-handed and costs £9.67. The code of this new product is TSL1, there are 5 in stock and none on order. Add details of this new product to the database.

|  |
| --- |
|  |

1. Remove the details of the manufacturer called Craft Supplies from the database.

|  |
| --- |
|  |

1. Increase the number in stock of all products by 2.

|  |
| --- |
|  |

1. Reduce by £5 the cost of all products that are manufactured by the manufacturer with ID 627.

|  |
| --- |
|  |

# Testing

An online music store uses a relational database called MusicStore to store details of CDs and their music labels.

The CD table is used to store details of the CDs while the Label table is used to store details of music labels. Data stored in each of the tables is shown below.



### CD table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **code** | **title** | **artist** | **label** | **tracks** | **cost** | **genre** |
| 42AM | The Best of | Eva Cassidy | Blix Street | 20 | 9.99 | Pop |
| 5J8Y | + | Ed Sheeran | Atlantic Records | 13 | 10.0 | Indie |
| 61DS | Stronger Together | Military Wives | Decca Records | 14 | 7.49 | Choral |
| 82FK | The Power of Love | Sam Bailey | Syco Music | 11 | 7.5 | Soul |
| 8G9K | iDos! | Greenday | Warner Bros | 13 | 8.99 | Garage |
| 8PL3 | Opera | Andrea Bocelli | Decca Records | 21 | 10.0 | Opera |
| 8QGC | Our Version of Events | Emeli Sande | Virgin Records | 19 | 8.99 | R&B |
| 91TU | Right Place Right Time | Olly Murs | Epic Records | 12 | 11.99 | Pop |
| 942Y | Take Me Home | One Direction | Syco Music | 13 | 10.0 | Pop |
| 95VW | Grrr | The Rolling Stones | Polydor Records | 51 | 11.99 | R&R |
| 97F2 | Red | Taylor Swift | Mercury Records | 16 | 8.99 | Country |
| 9KYX | Glory Days | Little Mix | Syco Music | 20 | 9.99 | R&B |

### Label table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **label** | **founded** | **parentCompany** | **country** | **website** |
| Atlantic Records | 1947 | Warner Music Group | USA | www.atlanticrecords.com |
| Blix Street | 2006 |  | USA | www.blixstreet.com |
| Decca Records | 1929 | Universal Music Group | UK | www.decca.com |
| Epic Records | 1953 | Sony Music | USA | www.epicrecords.com |
| Mercury Records | 1945 | Universal Music Group | UK | www.mercuryclassics.com |
| Polydor Records | 1924 | Universal Music Group | Germany | www.polydor.co.uk |
| Syco Music | 2002 | Syco Music | UK | www.sycoentertainment.com |
| Virgin Records | 1973 | Universal Music Group | USA | www.virginrecords.com |
| Warner Bros | 1958 | Warner Music Group | Japan | www.warnerbrosrecords.com |

1. Kerry has been asked to list the CD code, title, label and number of tracks of all the CDs that have fewer than 14 tracks. She writes a SQL query to display the details needed.

Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
| **CD Code** | **Title** | **Label** | **Number Of Tracks** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Kerry is now asked to display the artist name, label and genre of any CDs produced by a German record label. Use the table below to predict the output from the query.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Kerry is asked to display the label name, year founded, country of origin and website of any record label that originated in the USA before 1965. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Kerry is asked to display the title, genre and cost of any CDs that were produced by record labels that originated in the USA and cost less than £10. Use the table below to predict the output from the query.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Kerry is asked to display the CD code, label and number of tracks of any CD produced by Syco Music. These details should be listed so that the CD with the most tracks appears first. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of CD listed first |  |  |  |
| Details of CD listed last |  |  |  |

1. Kerry is asked to display the title, artist, label and number of tracks of and CDs that were produced by UK record labels that have 15 or fewer tracks. Use the table below to predict the results of the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Evaluation

1. In Q1, Kerry was asked to list the CD code, title, label and number of tracks of all the CDs that have fewer than 14 tracks.

Here is the answer table produced by Kerry’s query for Q1.

A screenshot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q2, Kerry was asked to display the artist name, label and genre of any CDs produced by a German record label.

Here is the answer table produced by Kerry’s query for Q2.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q3, Kerry was asked to display the label name, year founded, country of origin and website of any record label that originated in the USA before 1965.

Here is the answer table produced by Kerry’s query for Q3.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q4, Kerry was asked to display the title, genre and cost of any CDs that were produced by record labels that originated in the USA and cost less than £10.

Here is the answer table produced by Kerry’s query for Q4.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q5, Kerry was asked to display the CD code, label and number of tracks of any CD produced by Syco Music. These details should be listed so that the CD with the most tracks appears first.

Here is the answer table produced by Kerry’s query for Q5.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q6, Kerry was asked to display the title, artist, label and number of tracks of and CDs that were produced by UK record labels that have 15 or fewer tracks.

Here is the answer table produced by Kerry’s query for Q6.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

# Testing

Clydeview Library uses a relational database called BookData to store details of books and authors in two tables called Book and Author. Data stored in each of the tables is shown below.

### Author table

A screenshot of a computer

Description automatically generated

### Book table

A screenshot of a computer

Description automatically generated

1. Sean has been asked to list the ISBN, category, genre and publisher of all fiction books suitable for adults. He writes a SQL query to display the details needed.

Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
| **ISBN** | **Category** | **Genre** | **Publisher** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Sean is now asked to display the full name and nationality of all American authors. These details should be listed in alphabetical order of author surname. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of first Author listed |  |  |  |
| Details of last Author listed |  |  |  |

1. Sean is asked to display the first name and website of the author with surname ‘Rowling’ together with the ISBN and date of publication of all books written by that author. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Sean is asked to add the website details for the author called Mick Inkpen to the database. The URL of his website is <http://authorpages.hoddersystems.com/MickInkpen/first.asp>. Use the space below to predict the result of this query.

|  |
| --- |
|  |

1. Sean is asked to display the surname of the authors, together with the genre and number of pages, of any fiction books that have more than 200 pages. These details should be listed so that the book with the most pages is listed first. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of first Book listed |  |  |  |
| Details of last Book listed |  |  |  |

1. Sean is asked to display the title, category and number of pages, together with the surname of the author, of all children’s books that have fewer than 50 pages. Use the table below to predict the results of the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Evaluation

1. In Q1, Sean was asked to list the ISBN, category, genre and publisher of all fiction books suitable for adults.

Here is the answer table produced by Sean’s query for Q1.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q2, Sean was asked to display full name and nationality of all American authors in alphabetical order of author surname.

Here is the answer table produced by Sean’s query for Q2.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q3, Sean was asked to display the first name and website of the author with surname ‘Rowling’ together with the ISBN and date of publication of all books written by that author.

Here is the answer table produced by Sean’s query for Q3.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q4, Sean was asked to add the website details for the author called Mick Inkpen to the database (the URL of his website is <http://authorpages.hoddersystems.com/MickInkpen/first.asp>).

Here is the Author table after it had been updated by the query that Sean created for Q4.

A screenshot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q5, Sean was asked to surname of the authors, together with the genre and number of pages, of any fiction books that have more than 200 pages. These details should be listed so that the book with the most pages is listed first.

Here is the answer table produced by Sean’s query for Q5.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q6, Sean was asked to display the title, category and number of pages, together with the surname of the author, of all children’s books that have fewer than 50 pages

Here is the answer table produced by the query that Sean created for Q6.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. Database Structure

## Part 1

Read the information below about fields and field types.

### Fields

A database **field** is one single item of information in a database. Each heading in a database table identifies a field.

A database about pupils would store fields such as name, address and date of birth.



### Field Types

When setting up a database you need to think carefully about what type of data each field is going to store. The **field type** determines the type of data that can be held in a field.

Examples of field types include: text, number, date, time, graphic, link and Boolean.

|  |  |  |
| --- | --- | --- |
| **Field Type** | **Description** | **Example** |
| Text | Used for storing characters or combinations of characters and numbers | Peter Parker, PA19 1UX |
| Number | Used for storing numbers | 34, 8.99, -26 |
| Date | Used for storing dates | 23/7/2007 |
| Time | Used for storing time values | 11:37am |
| Boolean | Used for storing True/False or Yes/No values | True, False, Yes, No |

## Part 2

Look at the database table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| KnownAs | Surname | Branch | AccountType | Balance(£) | Rate(%) | InterestDue |
| Leigh | Nelson | Glasgow | Saver | 100.67 | 1.7 | 01/02/2024 |
| Sam | McAteer | Glasgow | Flexi | 304.86 | 2.3 | 02/03/2024 |
| Robyn | Jones | Gourock | ISA | 89.45 | 3.0 | 09/03/2024 |
| Jack | Devlin | Greenock | Saver | 162.91 | 1.7 | 29/01/2024 |

1. How many fields are in the database table above?

|  |
| --- |
|  |

1. How many records are in the database table above?

|  |
| --- |
|  |

1. For each field, write the field name and the data type used.

|  |  |
| --- | --- |
| **Field Name** | **Type** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. You have been asked to create a database to store information about pupils in this class. Complete the shaded cells of the table, adding your own fields and data types. All sensible answers are correct.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Forename | Surname | Eye Colour | ***DoB*** | Year | ***Siblings*** | ***Hair Colour*** |
| Text | Text | Text | Date | ***S2*** | Boolean | ***Text*** |

*Example answers, anything sensible is valid.*

1. You have been asked to create a database to store information about products in a shop. Complete the table, adding your own fields and data types. All sensible answers are correct.

|  |  |
| --- | --- |
| **Field name** | **Field type** |
| name | text |
| barCode | number |
| manufacturer | text |
| ageCheck | boolean |
| price | number |

1. Queries Introduction

## Part 1

Read the following information carefully.

What is a query?

A query is used to search or sort a database table. Queries allow the user of a database to quickly find the records that they are looking for or arrange the records into a particular order. Being able to perform queries is a major advantage of using an electronic database over a non-electronic database. Queries are commonly used to perform two tasks: **search** and **sort**.

### Search Queries

A search query is performed to retrieve all the records which **match** particular **search criteria**. A simple search uses only one criterion, while a complex search uses two or more criteria.

Below is an example of a simple search query to find all the people in the database table with the first name John.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Database Table** | | **Query** | **Result** | |
| **Forename** | **Surname** | Find all the people with the first name John | **Forename** | **Surname** |
| Joanna | Keetings | John | Brown |
| John | Brown |  | |
| James | Smith |

The result of a search query will return only the records which match the criteria given.

### Sort Queries

A sort query is performed to arrange all the records in the database table into a particular order. There are two different orders that can be used: **ascending****order** and **descending****order**.

**Ascending** **Order** – arranges the records in order from smallest 🡪 largest or alphabetically from A 🡪 Z.

**Descending** **Order** – arranges the records in order from largest 🡪 smallest or in reverse alphabetical order from Z 🡪 A.

## Part 2

A CRIMINAL table in the PRISON database is used to store details of criminals. The tables has 8 fields shown in the sample record below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **firstName** | **surname** | **dob** | **age** | **crime** | **sentence** | **eyeColour** | **hairColour** |
| Patrick | Grant | 19/4/86 | 29 | Burglary | 2 years | Blue | Brown |

**Example**: Find all the criminals named Brian.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_*Brian*\_\_\_\_\_  in the field \_\_\_\_*firstName*\_\_\_\_. |

Complete each of the following sentences by providing the missing search criteria and fields for each query.

1. Find all the criminals with black hair.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with the surname Smith.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals aged 35.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with brown eyes.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals with blonde hair and green eyes.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_  AND contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Find all the criminals named Steven and are aged 19.

|  |
| --- |
| Search the database for records which contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_  AND contain \_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the field \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

1. Sorting

## Part 1

1. For each set of data below, state whether it has been arranged in ascending order or descending order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Set 1** | **Set 2** | **Set 3** | **Set 4** |
| Data | 12  15  19  27  34  56 | Water  Socks  Meat  Food  Coffee  Ball | 125  113  97  64  14  5 | Minister  Mister  Monster  Month  Mother  Mouth |
| Order |  |  |  |  |

## Part 2

1. For each table below, identify the field that has been used to sort the table and state the type of sort has been applied to the field.

### Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Andy | Smith | 25 | Blue |
| Chris | Holdsworth | 27 | Red |
| Ben | Adamson | 30 | Green |
| Fred | Jackson | 36 | Orange |

Field used to sort the table:

Order applied:

### Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Andy | Smith | 25 | Blue |
| Ben | Adamson | 30 | Green |
| Chris | Holdsworth | 27 | Red |
| Fred | Jackson | 36 | Orange |

Field used to sort the table:

Order applied:

### Table 3

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **age** | **colour** |
| Ben | Adamson | 30 | Green |
| Chris | Holdsworth | 27 | Red |
| Fred | Jackson | 36 | Orange |
| Andy | Smith | 25 | Blue |

Field used to sort the table:

Order applied:

## Part 3

The database tables below have been sorted using a complex sort. This is where a sort has been applied to two different fields.

1. For each database table, state which fields has been sorted and the type of sort that has been applied.

### Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| **First Name** | **Surname** | **Year** | **Class** |
| James | Cook | S1 | Bute |
| Rebecca | Bottle | S1 | Mull |
| Leon | Smith | S1 | Skye |
| Amy | Smith | S2 | Bute |
| Rebecca | Smith | S2 | Skye |
| Patrick | Adams | S3 | Skye |

1st sort:

2nd sort:

### Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| **First Name** | **Surname** | **Year** | **Class** |
| Patrick | Adams | S3 | Skye |
| Rebecca | Bottle | S1 | Mull |
| James | Cook | S1 | Bute |
| Leon | Smith | S2 | Skye |
| Rebecca | Smith | S1 | Skye |
| Amy | Smith | S2 | Bute |

1st sort:

2nd sort:

1. SQL – Introduction

If you are setting up a query directly to a database, the coding that goes on behind the scenes will most likely use a formal language called **SQL** (Structured Query Language) to set up the query.

Below is an example of a query written in SQL:

1 SELECT title, introduced, RAM

2     FROM Computer

3     WHERE RAM = 512

4     ORDER BY title ASC;

This query will display the fields title, introduced and RAM from the database table Computer for any record where the RAM field is equal to 512. The records will then be sorted into ascending order of the title field.

Although SQL is a coding language, it doesn’t follow a linear sequence. The instructions are processed in the following order, which is different from the written order:

FROM – which database table is required?

WHERE – what criteria must be met?

SELECT – which fields should be displayed?

ORDER BY – how should the fields be sorted?

## Part 1

1. What does SQL stand for?

|  |
| --- |
|  |

1. Number the following steps into the correct order from 1 to 4, as they would be used in a SQL statement.

|  |  |
| --- | --- |
| **Step** | **SQL Clause** |
| ***2*** | FROM |
| ***4*** | ORDER BY |
| ***1*** | SELECT |
| ***3*** | WHERE |

1. Draw lines to match the SQL keywords to the correct description.

|  |  |  |
| --- | --- | --- |
| **SQL Clause** |  | **Description** |
| SELECT |  | The criteria which must be met |
| FROM |  | The fields to display |
| WHERE |  | The field and order used to sort the results |
| ORDER BY |  | The table used in the query |

## Part 2

1. Read each SQL statement below and fill in the blanks to complete the explanation.

### Query 1

|  |
| --- |
| 1 SELECT firstName, surname, age  2     FROM Pupil  3     WHERE age > 16  4     ORDER BY firstName ASC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

### Query 2

|  |
| --- |
| 1 SELECT itemName, price, quantity  2     FROM Stock  3     WHERE price < 9.99  4     ORDER BY price DESC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

### Query 3

|  |
| --- |
| 1 SELECT surname, address, postCode  2     FROM Customer  3     WHERE title = "Mr"  4     ORDER BY dob ASC; |

This query will display the fields

From the database table called

For any record where the

The records will then be sorted

1. Queries

## Tasks

Read each SQL statement below and fill in the blanks to complete the explanation.

### Query 1

|  |
| --- |
| 1 SELECT firstName, surname, age  2     FROM Pupil  3     WHERE age > 16 AND surname = "Smith"  4     ORDER BY firstName ASC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

### Query 2

|  |
| --- |
| 1 SELECT itemName, price, quantity  2     FROM Stock  3     WHERE price < 9.99 OR quantity > 5  4     ORDER BY price DESC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

### Query 3

|  |
| --- |
| 1 SELECT surname, address, postCode  2     FROM Customer  3     WHERE title = "Mr" AND age > 50  4     ORDER BY dob ASC; |

This query will display the fields

From the table called

For any record where the

The records will then be sorted

1. Query Design (Simple) – Done

The Customer table stores details of the customers of a mobile phone company. The table has the 8 fields listed below.



Before queries are written, they are designed by stating the tables, conditions, fields, and order that will need to used. Design queries to perform each of the following tasks.

1. Search the database to display the full name (forename and surname) of the customer with the surname "Rice".

|  |  |
| --- | --- |
| Field(s) | forename, surname |
| Table(s) | Customer |
| Search criteria | surname = "Rice" |
| Sort order |  |

1. Search the database to display the full name and town of all customers who live in Inverkip.

|  |  |
| --- | --- |
| Field(s) | forename, surname, town |
| Table(s) | Customer |
| Search criteria | town = "Inverkip" |
| Sort order |  |

1. Search the database to display the surname, package and town of all the customers who are on the large package.

|  |  |
| --- | --- |
| Field(s) | surname, package, town |
| Table(s) | Customer |
| Search criteria | package = "Large" |
| Sort order |  |

1. Search the database to display the surname, town and street of all the customers who live in Port Glasgow.

|  |  |
| --- | --- |
| Field(s) | surname, town, street |
| Table(s) | Customer |
| Search criteria | town = "Port Glasgow" |
| Sort order |  |

1. Display the forename, surname and town of all of the customers and sort these details into alphabetical order of surname.

|  |  |
| --- | --- |
| Field(s) | forename, surname, town |
| Table(s) | Customer |
| Search criteria |  |
| Sort order | surname ASC |

1. Display the forename, surname and paymentDueDate of all the customers and sort these details into descending order of paymentDueDate.

|  |  |
| --- | --- |
| Field(s) | forename, surname, paymentDueDate |
| Table(s) | Customer |
| Search criteria |  |
| Sort order | paymentDueDate DESC |

1. Display the forename, surname and customerNo of all the customers with a customerNo greater than 110. Sort the details into alphabetical order of forename.

|  |  |
| --- | --- |
| Field(s) | forename, surname, customerNo |
| Table(s) | Customer |
| Search criteria | customerNo > 110 |
| Sort order | forename ASC |

1. Search the database to display the full name and town of all the customers who live in Kilmacolm. Sort the details into alphabetical order of surname.

|  |  |
| --- | --- |
| Field(s) | forename, surname, town |
| Table(s) | Customer |
| Search criteria | town = "Kilmacolm" |
| Sort order | surname ASC |

1. Query Implementation – Done

Use the Replit project N5 DDD Clydeview. Screen snip the evidence, SQL and output, for the tasks below.

Use the query designs from to help create the SQL statement for each task.

1. Search the database to display the full name (forename and surname) of the customer with the surname 'Rice'. What is their first name?

SELECT forename, surname

FROM Customer

WHERE surname = "Rice";

|  |  |
| --- | --- |
| **forename** | **surname** |
| Margaret | Rice |

1. Search the database to display the full name and town of all customers who live in Inverkip.

SELECT forename, surname, town

FROM Customer

WHERE town = "Inverkip";

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **town** |
| Donald | McAndrew | Inverkip |

1. Search the database to display the surname, package and town of all the customers who are on the large package.

SELECT surname, package, town

FROM Customer

WHERE package = "Large";

|  |  |  |
| --- | --- | --- |
| **surname** | **package** | **town** |
| Rutherford | Large | Port Glasgow |
| Hastings | Large | Gourock |
| Flowers | Large | Greenock |
| Shields | Large | Port Glasgow |

1. Search the database to display the surname, town and street of all the customers who live in Port Glasgow.

SELECT surname, town, street

FROM Customer

WHERE town = "Port Glasgow";

|  |  |  |
| --- | --- | --- |
| **surname** | **town** | **street** |
| Rutherford | Port Glasgow | 5A Panama Place |
| Sweeney | Port Glasgow | 3 Paisley Road |
| Shields | Port Glasgow | 4 Brookside Close |

1. Display the forename, surname and town of all of the customers and sort these details into alphabetical order of surname. Who appears at the top of the list?

SELECT forename, surname, town

FROM Customer

ORDER BY surname ASC;

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **town** |
| Lauren | Calder | Wemyss Bay |
| Abigail | Cameron | Wemyss Bay |
| Ryan | Collins | Gourock |
| Grant | Donaldson | Gourock |
| Christine | Flowers | Greenock |

1. Display the forename, surname and paymentDueDate of all the customers and sort these details into descending order of paymentDueDate. Who appears at the top of the list?

SELECT forename, surname, paymentDueDate

FROM Customer

ORDER BY paymentDueDate DESC;

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **paymentDueDate** |
| Donald | McAndrew | 2020-08-01 |
| Shelby | Sweeney | 2020-05-26 |
| Ryan | Collins | 2020-05-19 |
| Jack | Shields | 2020-05-10 |
| Pauline | Milne | 2020-05-09 |

1. Display the forename, surname and customerNo of all the customers with a customerNo greater than 110. Sort the details into alphabetical order of forename. Who appears at the top of the list?

SELECT forename, surname, customerNo

FROM Customer

WHERE customerNo > 110

ORDER BY forename ASC;

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **customerNo** |
| Jack | Shields | 113 |
| Margaret | Rice | 115 |
| Paul | McGill | 112 |
| Pauline | Milne | 114 |
| Ross | Lambie | 111 |

1. Search the database to display the full name and town of all the customers who live in Kilmacolm. Sort the details into alphabetical order of surname. Who appears at the top of the list?

SELECT forename, surname, town

FROM Customer

WHERE town = "Kilmacolm"

ORDER BY surname ASC;

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **town** |
| Ross | Lambie | Kilmacolm |
| Paul | McGill | Kilmacolm |
| Pauline | Milne | Kilmacolm |
| Justine | O'Docherty | Kilmacolm |

1. Query Design (Complex) – Done

The SuperHero table stores details of movie characters. Each of the characters has played a role in a super hero movie. The table has 8 fields shown below.



Design SELECT queries to perform each of the following tasks.

1. Search the database to find a character who played the role of 'Super Hero' with 'Acrobatics' as his/her main ability. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility |
| Table(s) | SuperHero |
| Search criteria | role = "Super Hero" and  mainAbility = "Acrobatics" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Henchman' whose main ability is 'Strength'. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility |
| Table(s) | SuperHero |
| Search criteria | role = "Henchman" and  mainAbility = "Strength" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' whose main ability is 'magic' and whose origin of power is 'training'. Display the character’s name, role, main ability and origin of power.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility, originOfPower |
| Table(s) | SuperHero |
| Search criteria | role = "Super Villain" and  mainAbility = "magic" and  originOfPower = "training" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Hero' with the main ability ‘Magic’ whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility, originOfPower |
| Table(s) | SuperHero |
| Search criteria | role = "Super Hero" and  mainAbility = "magic" and  originOfPower = "training" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' whose origin of power is 'Chemicals' and who has 'Gadgets' as ability 2. Display the character’s name, role, ability2 and origin of power.

|  |  |
| --- | --- |
| Field(s) | name, role, ability2, originOfPower |
| Table(s) | SuperHero |
| Search criteria | role = "Super Villain" and  originOfPower = "Chemicals" and  ability2 = "Gadgets" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Team member' and has the main ability ‘Water breathing’. Display the character’s name, role and main ability.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility |
| Table(s) | SuperHero |
| Search criteria | role = "Team member" and  mainAbility = "Water breathing" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Hero' who has the main ability 'Flight', ability 2 'Super-strength' and origin of power 'Technology'. Display the character’s name, role, main ability, ability2 and origin of power.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility, ability2, originOfPower |
| Table(s) | SuperHero |
| Search criteria | role = "Super Hero" and  mainAbility = "Flight" and  ability2 = "Super-strength" and  originOfPower = "Technology" |
| Sort order |  |

1. Search the database to find a character who played the role of 'Super Villain' with the main ability 'Intelligence' and ability 2 'Martial arts'. Display the character’s name, role, main ability and ability2.

|  |  |
| --- | --- |
| Field(s) | name, role, mainAbility, ability2 |
| Table(s) | SuperHero |
| Search criteria | role = "Super Villian" and  mainAbility = "Intelligence" and  ability2 = "Martial arts" |
| Sort order |  |

1. Query Implementation – Done

Open the Clydeview.db database file. Paste the SQL and screen snip the output, for the tasks below.

Use the query designs from Task 8 to help create the SQL statement for each task.

1. Search the database to find a character who played the role of 'Super Hero' with 'Acrobatics' as his/her main ability. Display the character’s name, role and main ability.

SELECT name, role, mainAbility

FROM SuperHero

WHERE role = "Super Hero"

AND mainAbility = "Acrobatics";

|  |  |  |
| --- | --- | --- |
| **name** | **role** | **mainAbility** |
| Spider-Man | Super Hero | Acrobatics |

1. Search the database to find a character who played the role of 'Henchman' whose main ability is 'Strength'. Display the character’s name, role and main ability.

SELECT name, role, mainAbility

FROM SuperHero

WHERE role = "Henchman"

AND mainAbility = "Strength";

|  |  |  |
| --- | --- | --- |
| **name** | **role** | **mainAbility** |
| The Iron Avenger | Henchman | Strength |

1. Search the database to find a character who played the role of 'Super Villain' whose main ability is 'Magic' and whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

SELECT name, role, mainAbility, originOfPower

FROM SuperHero

WHERE role = "Super Villain"

AND mainAbility = "Magic"

AND originOfPower = "Training";

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **role** | **mainAbility** | **originOfPower** |
| Lord Voldemort | Super Villain | Magic | Training |

1. Search the database to find a character who played the role of 'Super Hero' with the main ability ‘Magic’ whose origin of power is 'Training'. Display the character’s name, role, main ability and origin of power.

SELECT name, role, mainAbility, originOfPower

FROM SuperHero

WHERE role = "Super Hero"

AND mainAbility = "Magic"

AND originOfPower = "Training";

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **role** | **mainAbility** | **originOfPower** |
| Harry Potter | Super Hero | Magic | Training |

1. Search the database to find a character who played the role of 'Super Villain' whose origin of power is 'Chemicals' and who has 'Gadgets' as ability 2. Display the character’s name, role, ability2 and origin of power.

SELECT name, role, ability2, originOfPower

FROM SuperHero

WHERE role = "Super Villain"

AND originOfPower = "Chemicals"

AND ability2 = "Gadgets";

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **role** | **ability2** | **originOfPower** |
| Green Goblin | Super Villain | Gadgets | Chemicals |

1. Search the database to find a character who played the role of 'Team member' and has the main ability 'Water breathing'. Display the character’s name, role and main ability.

SELECT name, role, mainAbility

FROM SuperHero

WHERE role = "Team member"

AND mainAbility = "Water breathing";

|  |  |  |
| --- | --- | --- |
| **name** | **role** | **mainAbility** |
| Whale Woman | Team member | Water breathing |

1. Search the database to find a character who played the role of 'Super Hero' who has the main ability 'Flight', ability 2 'Super-strength' and origin of power 'Technology'. Display the character’s name, role, main ability, ability2 and origin of power.

SELECT name, role, mainAbility, ability2, originOfPower

FROM SuperHero

WHERE role = "Super Hero"

AND mainAbility = "Flight"

AND ability2 = "Super-strength"

AND originOfPower = "Technology";

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **role** | **mainAbility** | **ability2** | **originOfPower** |
| Green Lantern | Super Hero | Flight | Super-strength | Technology |

1. Search the database to find a character who played the role of 'Super Villain' with the main ability 'Intelligence' and ability 2 ‘Martial arts’. Display the character’s name, role, main ability and ability2.

SELECT name, role, mainAbility, ability2

FROM SuperHero

WHERE role = "Super Villain"

AND mainAbility = "Intelligence"

AND ability2 = "Martial arts";

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **role** | **mainAbility** | **ability2** |
| The Kingpin | Super Villain | Intelligence | Martial arts |

1. Entity Design – Done

### Member entity

1. Complete the data dictionary for the Member entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| memberID | PK | number |  | Y |  |
| lastName |  | text | 20 |  |  |
| firstName |  | text | 20 |  |  |
| age |  | number |  | Y | range: >-18 |
| email |  | text | 30 | Y |  |

### Post entity

1. Complete the data dictionary for the Post entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| postID | PK | number |  | Y |  |
| title |  | text | 50 | Y |  |
| date |  | date |  | Y |  |
| memberID | FK | number |  | Y | Existing memberID in Member table |
| words |  | number |  | Y | Range: >= 20 and <= 250 |

1. Create an Entity Relationship Diagram to represent the relationship between the Member and Post entities.



1. Describe the type of relationship that exists between the Member and Post entities.

Each member can make many posts and each post is made by one member.

1. Entity Design – Mostly done

### Album entity

1. Complete the data dictionary for the Album entity.

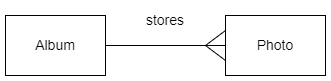
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| album\_id | PK | number |  | Y |  |
| name |  | text | 30 | N |  |
| category |  | text | 7 | Y | restricted choice: animals, cars, castles, surfing, towns |
| description |  | text | 255 | N |  |
| no\_of\_photos |  | number |  | Y | range: <= 120 |

### Photo entity

1. Complete the data dictionary for the Photo entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| photo\_id | PK | rumber |  | Y |  |
| album\_id | FK | number |  | Y | album\_id exists in Album table |
| title |  | text | 50 | N |  |
| image |  | text | 255 | N |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Album and Photo entities.



1. Describe the type of relationship that exists between the Album and Photo entities.

The relationship between Album and Photo is 1 to Many, i.e. one Album stores many photos, or many photos are stored in an Album.

1. Entity Design

The RetroClothing website uses a relational database to store details of items of women’s clothing for sale and the brand of each item in two separate tables called Item and Brand.

To minimise data entry errors, RetroClothing applies the following restrictions:

* The nationality of the brands used in the website are American, British or Italian
* The eras featured on the site are 1940, 1950s, 1960s and 1970s
* Item codes all have 7 characters
* Item size should be limited to 8, 10, 12, 14 and 16

Sample data stored in each table is shown below.

### Item Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item Code** | **Description** | **Size** | **Era** | **Brand ID** |
| RSS1001 | Red swim suit | 10 | 1950s | B3 |
| FDP1002 | Floral dungarees playsuit | 10 | 1990s | B2 |
| BSC2103 | Brown swing coat | 16 | 1960s | B5 |
| CSP3204 | Circle skirt black white polka dot | 12 | 1950s | B4 |
| FPD3225 | Floral print hostess dress | 10 | 1970s | B5 |

### Brand Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Brand ID** | **Brand** | **Year Established** | **Nationality** |
| B1 | Valentino | 1965 | Italian |
| B2 | Mary Quant | 1970 | British |
| B3 | Rose Marie Reid | 1946 | American |
| B4 | Elmoor |  | British |
| B5 | Susan Small | 1942 | British |

### Item entity

1. Complete the data dictionary for the Item entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Brand entity

1. Complete the data dictionary for the Brand entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Item and Brand entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Item and Brand entities.

|  |
| --- |
|  |

1. Entity Design

The SportsStats website uses a relational database to store details of Scottish sports teams and players in two separate tables called Player and Team.

* All players have a unique Player ID that has 4 characters
* Players are given a star rating between 1 and 5
* The website features a limited number of sports: basketball, handball, hockey and netball

Sample data stored in each table is shown below.

### Player table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Player ID** | **Team ID** | **Star Rating** | **First Name** | **Last Name** | **Date of Birth** |
| L18C | 111 | 4 | Lindy | Osborne | 14/01/1990 |
| F19F | 113 | 3 | Fred | Freddricks | 30/07/1987 |
| Y01D | 131 | 5 | Yasmine | Davies | 22/11/1992 |

### Team table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team ID** | **Team Name** | **Sport** | **Manager** | **Coach** | **Home Town** |
| 111 | West Stars | Hockey | Chris Davidson |  | Paisley |
| 112 | Killie Shooters | Basketball |  | Liz Smillie | Ardrossan |
| 113 | Jumpin Jacks | Basketball | Dave Ford | Ali Mustapha | Dunbar |

### Player entity

1. Complete the data dictionary for the Player entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Team entity

1. Complete the data dictionary for the Team entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Create an Entity Relationship Diagram to represent the relationship between the Player and Team entities.

|  |
| --- |
|  |

1. Describe the type of relationship that exists between the Team and Player tables.

|  |
| --- |
|  |

1. Entity and Relationship Design

## Part 1 – Design

The ClydeVet veterinary practice uses a relational database called ClydeVet to store details about pets and the pet owners.

The details of pet owners in a table called Owner.

Sample data stored in the Owner table is shown below.

### Owner Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Owner ID** | **First Name** | **Surname** | **Address** | **Town** | **Contact Telephone** |
| 1277 | Hardeep | Singh | 65 Iona Way | Greenock | 01475255707 |
| 2356 | Sally | Chan | 142 Main Street | Greenock | 01475242499 |
| 3510 | Elaine | Bryce | 29 Clyde Drive | Gourock | 01475636321 |
| 3821 | Cameron | Gray | 17 Shuttle Street | Gourock | 01475312245 |

### Owner Entity

1. Complete the data dictionary for the Owner entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 2 – Design

The practice specialises in the treatment of certain types of pet:

* Cat
* Dog
* Budgie
* Gerbil
* Tortoise

Details of any pets treated by the practice are stored in the ClydeVet database in a second table called Pet. This table is linked to the Owner table using a foreign key.

Sample data stored in the Pet table is shown below.

### Pet Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pet Code** | **Pet Name** | **Pet Type** | **Date of Birth** | **Received Vaccination?** | **Owner ID** |
| P0123 | Misty | Cat | 23/04/2012 | True | 2356 |
| P0345 | Rover | Dog | 12/12/2010 | True | 3821 |
| P0887 | Foggy | Cat | 23/01/2012 | True | 1277 |
| P1559 | Gladys | Gerbil | 16/04/2010 | False | 1277 |
| P1985 | Slinky | Tortoise | 24/09/2016 | False | 3510 |
| P2233 | Speedy | Tortoise | 09/06/2013 | True | 1277 |

### Pet Entity

1. Complete the data dictionary for the Pet entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Part 3 – ERD

1. Draw an entity relationship diagram to represent the relationship between the Owner and Pet entities.

|  |
| --- |
|  |

1. Create Tables – Done

CREATE TABLE Owner (

owner\_id INT NOT NULL,

first\_name VARCHAR(20),

surname VARCHAR(30),

address VARCHAR(50),

town VARCHAR(30),

contact\_phone VARCHAR(13) NOT NULL

CHECK (LENGTH(contact\_phone) >= 11),

PRIMARY KEY (owner\_id)

);

CREATE TABLE Pet (

pet\_code VARCHAR(4) NOT NULL

CHECK (LENGTH(pet\_code) = 5),

pet\_name VARCHAR(20),

pet\_type VARCHAR(10) NOT NULL

CHECK (pet\_type IN ("Cat", "Dog", "Budgie",

"Gerbil", "Tortoise")),

dob DATE,

vaccine BOOL NOT NULL,

owner\_id INT NOT NULL,

FOREIGN KEY (owner\_id)

REFERENCES Owner (owner\_id),

PRIMARY KEY (pet\_code)

);

PRAGMA foreign\_keys = on;

1. Paste below, evidence to show that the table have been created.

SELECT sql

FROM sqlite\_schema

WHERE tbl\_name = "Owner"

OR tbl\_name = "Pet";

1. Entity and Relationship Design – Part

## Part 1 – Design

### Product Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| name |  | text | 30 | N |  |
| code | PK | text | 4 | Y |  |
| no\_in\_stock |  | number |  | Y | range: >= 0 and <= 50 |
| on\_order |  | boolean |  | Y |  |
| price |  | number |  | Y | range: > 1.00 |
| manufacturer\_id | FK | number |  | Y | Exists in Manufacturer table |

*Variations that achieve the same outcome are also valid.*

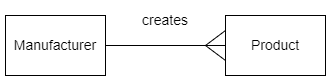
## Part 2 – Design

### Manufacturer Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| manufacturer\_id | PK | number |  | Y |  |
| name |  | text | 30 | N |  |
| address |  | text | 50 | N |  |
| phone |  | text | 13 | Y | length: >= 11 |

## Part 3 – ERD

1. Draw an entity relationship diagram to represent the relationship between the Product and Manufacturer entities.



*The relationship description can be different..*

1. Create Tables

CREATE TABLE Manufacturer (

manufacturer\_id INT NOT NULL,

name VARCHAR(30),

address VARCHAR(50),

phone VARCHAR(13) NOT NULL

CHECK (LENGTH(phone) >= 11),

PRIMARY KEY (manufacturer\_id)

);

CREATE TABLE Product (

name VARCHAR(30),

code VARCHAR(4) NOT NULL,

no\_in\_stock INT NOT NULL

CHECK (no\_in\_stock >= 0 AND no\_in\_stock <= 50),

on\_order BOOL NOT NULL,

price REAL NOT NULL

CHECK (price > 1.00),

manufacturer\_id INT NOT NULL,

FOREIGN KEY (manufacturer\_id)

REFERENCES Manufacturer (manufacturer\_id),

PRIMARY KEY (code)

);

PRAGMA foreign\_keys = on;

SELECT sql

FROM sqlite\_schema

WHERE tbl\_name = "Product"

OR tbl\_name = "Manufacturer";

1. Entity and Relationship Design – Mostly

## Part 1 – Design

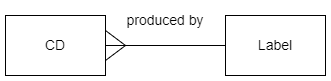
### CD Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| cdCode | PK | text | 4 | Y |  |
| title |  | text | 30 | N |  |
| artist |  | text | 30 | N |  |
| label | FK | text | 30 | Y | Exists in Label table |
| noOfTracks |  | number |  | Y | range: >=10 AND <= 60 |
| cost |  | number |  | Y | range: >=6.99 AND <= 15.00 |
| genre |  | text | 7 | Y | restricted choice: Choral, Country, Garage, Indie, Opera, Pop, R&B, R&R, Soul |

### Label Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| label | PK | text | 30 | Y |  |
| founded |  | number |  | N |  |
| parentCompany |  | text | 30 | N |  |
| country |  | text | 7 | Y | restricted choice: Germany, Japan, UK, USA |
| website |  | text | 50 | N |  |

## Part 2 – ERD



1. Create Tables
2. Open the N5 DDD Clydeview Blank project in Replit.
3. Create the CD and Label tables, and make sure that the properties match all of the settings indicated in your data dictionaries.
4. Remember to enforce referential integrity between the two tables.
5. Paste below, evidence to show that the table have been created.
6. Entity and Relationship Design – Mostly

## Part 1 – Design

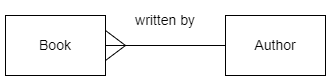
### Author Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| authorRef | PK | number |  | Y |  |
| firstName |  | text | 20 |  |  |
| surname |  | text | 30 |  |  |
| nationality |  | text | 30 |  |  |
| dob |  | date |  |  |  |
| website |  | text | 50 |  |  |

### Book Entity

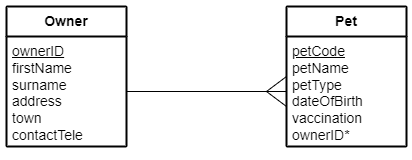
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| category |  | text | 5 | Y | restricted choice: Adult, Child |
| genre |  | text | 13 | Y | restricted choice: Autobiography, Fantasy, Fiction, Joke, Mystery, Fiction, Thriller |
| title | PK | text | 255 | Y |  |
| authorRef | FK | number |  | Y | Exists in Author table |
| publisher |  | text | 30 |  |  |
| isbn |  | text | 10 |  |  |
| dateOfPub |  | date |  |  |  |
| noOfPages |  | number |  | Y | range: >=32 AND <= 950 |

## Part 2 – ERD



1. Create Tables
2. Open the N5 DDD Clydeview Blank project in Replit.
3. Create the Author and Book tables, and make sure that the properties match all of the settings indicated in your data dictionaries.
4. Remember to enforce referential integrity between the two tables.
5. Paste below, evidence to show that the table have been created.
6. Query Design – Done

ClydeVet veterinary practice uses a relational database store details pets and their owners in two separate tables called Owner and Pet. The structure of the tables is shown below.



Design SELECT queries to perform each of the following tasks.

1. List the full name and address, and name of their pets, for all of the cat owners.

|  |  |
| --- | --- |
| Field(s) | firstName, surname, address, name |
| Table(s) | Owner, Pet |
| Search criteria | petType = "Cat" |
| Sort order |  |

1. List the full name and contact telephone number, with the codes of their pets, for all tortoise owners.

|  |  |
| --- | --- |
| Field(s) | firstName, surname, contactTele, petCode |
| Table(s) | Owner, Pet |
| Search criteria | petType = "Tortoise" |
| Sort order |  |

1. List the full name, address, and the name of their pets, of all owners who have pets that have not yet received their vaccinations.

|  |  |
| --- | --- |
| Field(s) | firstName, surname, address, petName |
| Table(s) | Owner, Pet |
| Search criteria | Vaccination = FALSE |
| Sort order |  |

1. List the name, type of each pet and town of any pet whose owner lives in Greenock.

|  |  |
| --- | --- |
| Field(s) | petName, petType, town |
| Table(s) | Owner, Pet |
| Search criteria | town = "Greenock" |
| Sort order |  |

1. List the name, vaccination details and contact telephone number of their owner, of all pets whose owners live in Gourock.

|  |  |
| --- | --- |
| Field(s) | petName, vaccination, contactTele |
| Table(s) | Owner, Pet |
| Search criteria | town = "Gourock" |
| Sort order |  |

1. Display the full names of all of the owners with the name and type of their pets. These details should be displayed in alphabetical order of owner surname.

|  |  |
| --- | --- |
| Field(s) | firstName, surname, petName, petType |
| Table(s) | Owner, Pet |
| Search criteria |  |
| Sort order | Surname ASC |

1. Display the name and full address (inc town) of each pet. These details should be arranged in alphabetical order of town; pets who live in the same town should be arranged in alphabetical order of pet type.

|  |  |
| --- | --- |
| Field(s) | petName, address, town |
| Table(s) | Owner, Pet |
| Search criteria |  |
| Sort order | town ASC, petType ASC |

1. Display the name, pet type, town and dateOfBirth of all the cats; details of the youngest pet should be displayed first.

|  |  |  |
| --- | --- | --- |
| Field(s) |  | petName, petType, town, dateOfBirth |
| Table(s) |  | Owner, Pet |
| Search criteria |  | petType = "Cat" |
| Sort order |  | dateOfBirth DESC |

1. Query Implementation – Done

Use Repl.it to fork the Team template for the task. Screen snip the required information to produce the evidence for the tasks below.

Use your query designs from Part 1 to help create the SQL statement for each task.

1. List the full name and address, and name of their pets, for all of the cat owners.

SELECT firstName, surname, address, town

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

AND petType = "cat";

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **address** | **town** |
| Sally | Chan | 142 Main Street | Greenock |
| Hardeep | Singh | 64 Iona Way | Greenock |

1. List the full name and contact telephone number, with the codes of their pets, for all tortoise owners.

SELECT firstName, surname, contactTele

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

AND petType = "tortoise";

|  |  |  |
| --- | --- | --- |
| **firstName** | **surname** | **contactTele** |
| Elaine | Bryce | 01475636321 |
| Hardeep | Singh | 01475255707 |

1. List the full name and address, and the name of their pets, of all owners who have pets that have not yet received their vaccinations.

SELECT firstName, surname, address, town, petName

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

AND vaccination = False;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **firstName** | **surname** | **address** | **town** | **petName** |
| Hardeep | Singh | 64 Iona Way | Greenock | Gladys |
| Elaine | Bryce | 29 Clyde Drive | Gourock | Slinky |

1. List the name, type of each pet and town of any pet whose owner lives in Greenock.

SELECT petName, petType, town

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

AND town = "Greenock";

|  |  |  |
| --- | --- | --- |
| **petName** | **petType** | **town** |
| Misty | cat | Greenock |
| Foggy | cat | Greenock |
| Gladys | gerbil | Greenock |
| Speedy | tortoise | Greenock |

1. List the name, vaccination details and contact telephone number of their owner, of all pets whose owners live in Gourock.

SELECT petName, vaccination, contactTele

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

AND town = "Gourock";

|  |  |  |
| --- | --- | --- |
| **petName** | **vaccination** | **contactTele** |
| Rover | 1 | 01475312245 |
| Slinky | 0 | 01475636321 |

1. Display the full names of all of the owners with the name and type of their pets. These details should be displayed in alphabetical order of owner surname.

SELECT firstName, surname, petName, petType

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

ORDER BY surname ASC;

|  |  |  |  |
| --- | --- | --- | --- |
| **firstName** | **surname** | **petName** | **petType** |
| Elaine | Bryce | Slinky | tortoise |
| Sally | Chan | Misty | cat |
| Cameron | Gray | Rover | dog |
| Hardeep | Singh | Foggy | cat |
| Hardeep | Singh | Gladys | gerbil |

1. Display the name and full address (inc town) of each pet. These details should be arranged in alphabetical order of town; pets who live in the same town should be arranged in alphabetical order of pet type.

SELECT petName, address, town

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

ORDER BY town ASC,

petType ASC;

|  |  |  |
| --- | --- | --- |
| **petName** | **address** | **town** |
| Rover | 17 Shuttle Street | Gourock |
| Slinky | 29 Clyde Drive | Gourock |
| Misty | 142 Main Street | Greenock |
| Foggy | 64 Iona Way | Greenock |
| Gladys | 64 Iona Way | Greenock |

1. Display name, pet type, town and dateOfBirth of all the cats; details of the youngest pet should be displayed first.

SELECT petName, petType, town, dateOfBirth

FROM Owner, Pet

WHERE Owner.ownerID = Pet.ownerID

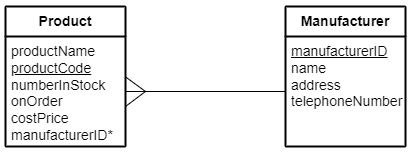
AND petType = "cat"

ORDER BY dateOfBirth DESC;

|  |  |  |  |
| --- | --- | --- | --- |
| **petName** | **petType** | **town** | **dateOfBirth** |
| Foggy | cat | Greenock | 2012-12-24 |
| Misty | cat | Greenock | 2012-01-23 |

1. Query Design – Done

A hardware store uses a relational database to store details of the products for sale and the manufacturer of each product in two separate tables called Product and Manufacturer. The structure of the tables is shown below.



Design SELECT queries to perform each of the following tasks.

1. List the product name, cost price and manufacturer name of all products manufactured by Craft Supplies.

|  |  |
| --- | --- |
| Field(s) | productName, costPrice, name |
| Table(s) | Product, Manufacturer |
| Search criteria | Name = "Craft Supplies" |
| Sort order |  |

1. List the manufacturer name, address, telephone number and cost price of any manufacturer who supplies products that cost more than £100.

|  |  |
| --- | --- |
| Field(s) | name, address, telephoneNumber, costPrice |
| Table(s) | Product, Manufacturer |
| Search criteria | costPrice > 100 |
| Sort order |  |

1. List the name of any manufacturers with the product names and number in stock of any products that are on order.

|  |  |
| --- | --- |
| Field(s) | name, productName, numberInStock |
| Table(s) | Product, Manufacturer |
| Search criteria | On\order = TRUE |
| Sort order |  |

1. List the product name and manufacturer name of any products with 12 or more in stock.

|  |  |
| --- | --- |
| Field(s) | productName, name |
| Table(s) | Product, Manufacturer |
| Search criteria | numberInStock >= 12 |
| Sort order |  |

1. List the product names, prices and manufacturer name of all products manufactured by Tool Makers. These details should be displayed in increasing order of price.

|  |  |
| --- | --- |
| Field(s) | productName, costPrice, name |
| Table(s) | Product, Manufacturer |
| Search criteria | name = "Tool Makers" |
| Sort order | costPrice ASC |

1. Display the manufacturer name, address, and number in stock for manufacturers who do not have outstanding orders. These details should be arranged in decreasing order of number in stock.

|  |  |
| --- | --- |
| Field(s) | Name, address, onOrder |
| Table(s) | Product, Manufacturer |
| Search criteria | onOrder = FALSE |
| Sort order | numberInStock DESC |

1. Display the product name, number in stock, cost price and manufacturer name of all products that cost less than £20. These details should be arranged in alphabetical order of manufacturer name; products from the same manufacturer should be displayed in alphabetical order of product name.

|  |  |
| --- | --- |
| Field(s) | productName, numberInStock, costPrice, name |
| Table(s) | Product, Manufacturer |
| Search criteria | costPrice < 20 |
| Sort order | Name ASC, productName ASC |

1. Display the manufacturer details (name, address and telephone number) together with the product code and number in stock all products that have more than 2 in stock. These details should be displayed in alphabetical order of manufacturer; products from the same manufacturer should be displayed with the largest quantity in stock at the top.

|  |  |
| --- | --- |
| Field(s) | Name, address, telephoneNumber, productCode, numberInStock |
| Table(s) | Product, Manufacturer |
| Search criteria | numberInStock > 2 |
| Sort order | Name ASC, numberInStock DESC |

1. Query Implementation – Mostly

Open the populated database called Products and complete the queries below.

Use your query designs from Part 1 to help create the SQL statement for each task.

1. List the product name, cost price and manufacturer name of all products manufactured by Craft Supplies.

SELECT productName, costPrice, name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND name = "Craft Supplies";

|  |  |  |
| --- | --- | --- |
| **productName** | **costPrice** | **name** |
| Metal File | 8.49 | Craft Supplies |
| Pliers | 11.99 | Craft Supplies |

1. List the manufacturer name, address, telephone number and cost price of any manufacturer who supplies products that cost more than £100.

SELECT name, address, telephoneNumber, costPrice

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND costPrice > 100;

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **address** | **telephoneNumber** | **costPrice** |
| Tool Makers | 231 London Walk Bristol | 01347234987 | 148.77 |
| Tool Makers | 231 London Walk Bristol | 01347234987 | 165.0 |

1. List name of any manufacturers with the product names and cost price of any products that are on order.

SELECT name, productName, costPrice

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND onOrder = TRUE;

|  |  |  |
| --- | --- | --- |
| **name** | **productName** | **costPrice** |
| Metal and Wood | Claw Hammer | 4.99 |
| Tool Makers | Combi Drill | 148.77 |
| Metal and Wood | Medium Paint Brush (Size 2) | 5.65 |
| Craft Supplies | Pliers | 11.99 |
| Tool Makers | Top Handled Jigsaw | 165.0 |

1. List the product name and manufacturer name of any products with 12 or more in stock.

SELECT productName, name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND numberInStock >= 12;

|  |  |
| --- | --- |
| **productName** | **name** |
| Combi Drill | Tool Makers |
| Medium Paint Brush (Size 2) | Metal and Wood |

1. List the product names, prices and manufacturer name of all products manufactured by Tool Makers. These details should be displayed in increasing order of price.

SELECT productName, costPrice, name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND name = "Tool Makers"

ORDER BY costPrice ASC;

|  |  |  |
| --- | --- | --- |
| **productName** | **costPrice** | **name** |
| Combi Drill | 148.77 | Tool Makers |
| Top Handled Jigsaw | 165.0 | Tool Makers |

1. Display the manufacturer name, address, on order details and number in stock information for products that are not on order. These details should be arranged in decreasing order of number in stock.

SELECT name, address, onOrder, numberInStock

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND onOrder = FALSE

ORDER BY numberInStock DESC;

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **address** | **onOrder** | **numberInStock** |
| Craft Supplies | Wishaw Industrial Estate | 0 | 9 |
| Metal and Wood | Tyne Way Newcastle | 0 | 7 |
| Metal and Wood | Tyne Way Newcastle | 0 | 4 |

1. Display the product name, number in stock, cost price and manufacturer name of all products that cost less than £20. These details should be arranged in alphabetical order of manufacturer name; products from the same manufacturer should be displayed in decreasing order of cost price.

SELECT productName, numberInStock, costPrice, name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND costPrice < 20

ORDER BY name ASC,

costPrice DESC;

|  |  |  |  |
| --- | --- | --- | --- |
| **productName** | **numberInStock** | **costPrice** | **name** |
| Pliers | 5 | 11.99 | Craft Supplies |
| Metal File | 9 | 8.49 | Craft Supplies |
| Saw | 4 | 9.99 | Metal and Wood |
| Medium Paint Brush (Size 2) | 34 | 5.65 | Metal and Wood |
| Mallet | 7 | 5.5 | Metal and Wood |

1. Display the manufacturer details (name, address and telephone number) together with the product code and number in stock all products that have more than 2 in stock. These details should be displayed in alphabetical order of manufacturer; products from the same manufacturer should be displayed with the largest quantity in stock at the top.
2. Modify Data – Introduction

A mobile phone company uses a database to store details of its customers in a table called Customer. Some of the data stored in this table is shown below.

A screen shot of a computer

Description automatically generated

Read each SQL query below and describe the effect that it will have on the contents of the Customer table above.

## Query 1 – SQL

1 INSERT INTO Customer

2     VALUES (116, "Nial", "Davies", "12 Clyde View",

3             "Gourock", "Premier", TRUE, "08/05/2017");

Query 1 – Description

|  |
| --- |
|  |

## Query 2 – SQL

1 UPDATE Customer

2     SET directDebit = TRUE

3     WHERE foreName = "Grant"

4         AND surname = "Donaldson";

Query 2 – Description

|  |
| --- |
|  |

## Query 3 – SQL

1 DELETE FROM Customer

2     WHERE customerNo = 111;

Query 3 – Description

|  |
| --- |
|  |

## Query 4 – SQL

1 UPDATE Customer

2     SET paymentDueDate = "01/06/2017";

Query 4 – Description

|  |
| --- |
|  |

## Query 5 – SQL

1 INSERT INTO Customer (foreName, surname, directDebit, package,  
                        paymentDueDate, street, town, customerNo)

2     VALUES( "Kelly", "Holmes", FALSE, "Standard",

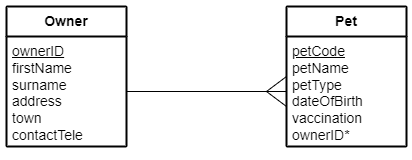
              "16/05/2017", "5 Lime Grove", "Greenock", 126);

Query 5 – Description

|  |
| --- |
|  |

1. Modify Data – Design

ClydeVet veterinary practice uses a relational database store details pets and their owners in two separate tables called Owner and Pet. The structure of the tables is shown below.



Design INSERT, DELETE and UPDATE queries to perform each of the following tasks.

Add the following details of Goldie the dog to the Pet table.

|  |
| --- |
| Pet Code: P4821  Name: Goldie  Type: Dog  Date of Birth: 26/10/2016  Received vaccination?: True  Owner ID: 3821 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.

|  |  |
| --- | --- |
| Field(s) |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 3905  First Name: Gary  Surname: Hughes  Address: 13 Juniper Place  Town: Wemyss Bay  Contact Telephone: 07998765432 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

|  |
| --- |
| Pet Code: P2751  Pet Name: Usain  Pet Type: Tortoise  Date of Birth: 28/10/2006  Received vaccination?: True |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 2664  First Name: Hannah  Surname: Black  Address: 47 High Road  Town: Greenock  Contact Telephone: 01475633633 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

|  |
| --- |
| Pet Code: P0438  Pet Name: Arnie  Pet Type: Budgie  Date of Birth: 13/03/2017  Received vaccination?: False |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove details of Slinky the tortoise from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove the details of the owner with Owner ID 3510 from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

The date of birth of Gladys the gerbil has been stored incorrectly; it should be 16/10/2004. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Owner 3905 has a new pet cat called Bruno with date of birth 14/05/2017 and pet code P1678 (Bruno hasn’t had any vaccinations yet). Add Bruno’s details to the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

Remove the details of the pet called Usain from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New / updated value(s) |  |
| Criteria |  |

1. Modify Data – Implementation

Use Repl.it to fork the Team template for the task. Screen snip the required information to produce the evidence for the tasks below.

Use your query designs from Part 1 to help create the SQL statement for each task.

Add the details of Goldie the dog to the Pet table.

|  |
| --- |
| Pet Code: P4821  Name: Goldie  Type: Dog  Date of Birth: 26/10/2016  Received vaccination?: True  Owner ID: 3821 |

The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.

The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 3905  First Name: Gary  Surname: Hughes  Address: 13 Juniper Place  Town: Wemyss Bay  Contact Telephone: 07998765432 |

|  |
| --- |
| Pet Code: P2751  Pet Name: Usain  Pet Type: Tortoise  Date of Birth: 28/10/2006  Received vaccination?: True |

Add the following details of a new owner and their pet to the database.

|  |
| --- |
| Owner ID: 2664  First Name: Hannah  Surname: Black  Address: 47 High Road  Town: Greenock  Contact Telephone: 01475633633 |

|  |
| --- |
| Pet Code: P0438  Pet Name: Arnie  Pet Type: Budgie  Date of Birth: 13/03/2017  Received vaccination?: False |

Remove details of Slinky the tortoise from the database.

The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.

Remove the details of the owner with Owner ID 3510 from the database.

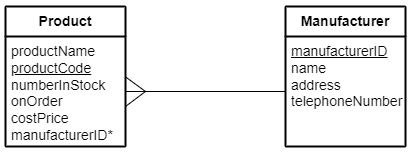
The date of birth of Gladys the gerbil has been stored incorrectly; it should be 16/10/2004. Edit the correct record of the database.

Owner 3905 has a new pet cat called Bruno with date of birth 14/05/2017 and pet code P1678 (Bruno hasn’t had any vaccinations yet). Add Bruno’s details to the database.

Remove the details of the pet called Usain from the database.

1. Modify Data – Design

A hardware store uses a relational database to store details of the products for sale and the manufacturer of each product in two separate tables called Product and Manufacturer. The structure of the tables is shown below.



Design INSERT, DELETE and UPDATE queries to perform each of the following tasks

1. Add details of this new product to the database.

**New Product**

Product Name: Spirit Level

Product Code: SPL3

Number In Stock: 14

On Order?: False

Cost Price(£): 17.99

Manufacturer ID: 531

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. The stock level of the product with Product Code MA16 has fallen to 1 and the product is now on order. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. The manufacturer called Tool Makers has moved. Its new address is: Unit 6, Avon Industrial Estate, Bath and its new phone number is: 01789334456. Edit the correct record of the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove the saw with Product Code SW22 form the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 327  Manufacturer Name: CVA Group  Address: 35 Lomond Way Paisley  Telephone Number: 01414141414 |  | Product Name: Circular Saw  Product Code: CSW2  Number In Stock: 3  On Order?: False  Cost Price(£): 99.00  Manufacturer ID: 327 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 408  Manufacturer Name: Cabinet Makers  Address: 158 Hawthorn Road Carlisle  Telephone Number: 03217329124 |  | Product Name: 6 piece Chisel Set  Product Code: CSP6  Number In Stock: 8  On Order?: True  Cost Price(£): 43.51  Manufacturer ID: 408 |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove all products manufactured by the manufacturer with ID 441 from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Manufacturer 327 has a new product called Tin Snips Left-handed and costs £9.67. The code of this new product is TSL1, there are 5 in stock and none on order. Add details of this new product to the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Remove the details of the manufacturer called Craft Supplies from the database.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Increase the number in stock of all products by 2.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Reduce by £5 the cost of all products that are manufactured by the manufacturer with ID 627.

|  |  |
| --- | --- |
| Type of query |  |
| Table |  |
| New/updated value(s) |  |
| Criteria |  |

1. Modify Data – Implementation

Open the populated database called Products DB and complete the queries below.

Use your query designs from Task 1 to help create the SQL statement for each task.

1. Add details of this new product to the database.

**New Product**

Product Name: Spirit Level

Product Code: SPL3

Number In Stock: 14

On Order?: False

Cost Price(£): 17.99

Manufacturer ID: 531

|  |
| --- |
|  |

1. The stock level of the product with Product Code MA16 has fallen to 1 and the product is now on order. Edit the correct record of the database.

|  |
| --- |
|  |

1. The manufacturer called Tool Makers has moved. Its new address is: Unit 6, Avon Industrial Estate, Bath and its new phone number is: 01789334456. Edit the correct record of the database.

|  |
| --- |
|  |

1. Remove the saw with Product Code SW22 form the database.

|  |
| --- |
|  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 327  Manufacturer Name: CVA Group  Address: 35 Lomond Way Paisley  Telephone Number: 01414141414 |  | Product Name: Circular Saw  Product Code: CSW2  Number In Stock: 3  On Order?: false  Cost Price(£): 99.00  Manufacturer ID: 327 |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. Add the following manufacturer and product details to the database.

|  |  |  |
| --- | --- | --- |
| Manufacturer ID: 408  Manufacturer Name: Cabinet Makers  Address: 158 Hawthorn Road Carlisle  Telephone Number: 03217329124 |  | Product Name: 6 piece Chisel Set  Product Code: CSP6  Number In Stock: 8  On Order?: true  Cost Price(£): 43.51  Manufacturer ID: 408 |

|  |
| --- |
|  |

|  |
| --- |
|  |

1. Remove all products manufactured by the manufacturer with ID 441 from the database.

|  |
| --- |
|  |

1. Manufacturer 327 has a new product called Tin Snips Left-handed and costs £9.67. The code of this new product is TSL1, there are 5 in stock and none on order. Add details of this new product to the database.

|  |
| --- |
|  |

1. Remove the details of the manufacturer called Craft Supplies from the database.

|  |
| --- |
|  |

1. Increase the number in stock of all products by 2.

|  |
| --- |
|  |

1. Reduce by £5 the cost of all products that are manufactured by the manufacturer with ID 627.

|  |
| --- |
|  |

1. Testing

An online music store uses a relational database called MusicStore to store details of CDs and their music labels.

The CD table is used to store details of the CDs while the Label table is used to store details of music labels. Data stored in each of the tables is shown below.

### CD table

A screen shot of a computer

Description automatically generated

### Label table

A screen shot of a computer

Description automatically generated

1. Kerry has been asked to list the CD code, title, label and number of tracks of all the CDs that have fewer than 14 tracks. She writes a SQL query to display the details needed.

Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
| **CD Code** | **Title** | **Label** | **Number Of Tracks** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Kerry is now asked to display the artist name, label and genre of any CDs produced by a German record label. Use the table below to predict the output from the query.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Kerry is asked to display the label name, year founded, country of origin and website of any record label that originated in the USA before 1965. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Kerry is asked to display the title, genre and cost of any CDs that were produced by record labels that originated in the USA and cost less than £10. Use the table below to predict the output from the query.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Kerry is asked to display the CD code, label and number of tracks of any CD produced by Syco Music. These details should be listed so that the CD with the most tracks appears first. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of CD listed first |  |  |  |
| Details of CD listed last |  |  |  |

1. Kerry is asked to display the title, artist, label and number of tracks of and CDs that were produced by UK record labels that have 15 or fewer tracks. Use the table below to predict the results of the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Evaluation
2. In Q1, Kerry was asked to list the CD code, title, label and number of tracks of all the CDs that have fewer than 14 tracks.

Here is the answer table produced by Kerry’s query for Q1.

A screenshot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q2, Kerry was asked to display the artist name, label and genre of any CDs produced by a German record label.

Here is the answer table produced by Kerry’s query for Q2.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q3, Kerry was asked to display the label name, year founded, country of origin and website of any record label that originated in the USA before 1965.

Here is the answer table produced by Kerry’s query for Q3.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Kerry’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q4, Kerry was asked to display the title, genre and cost of any CDs that were produced by record labels that originated in the USA and cost less than £10.

Here is the answer table produced by Kerry’s query for Q4.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q5, Kerry was asked to display the CD code, label and number of tracks of any CD produced by Syco Music. These details should be listed so that the CD with the most tracks appears first.

Here is the answer table produced by Kerry’s query for Q5.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. In Q6, Kerry was asked to display the title, artist, label and number of tracks of and CDs that were produced by UK record labels that have 15 or fewer tracks.

Here is the answer table produced by Kerry’s query for Q6.

A screen shot of a computer

Description automatically generated

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. Testing

Clydeview Library uses a relational database called BookData to store details of books and authors in two tables called Book and Author. Data stored in each of the tables is shown below.

### Author table

A screenshot of a computer

Description automatically generated

### Book table

A screenshot of a computer

Description automatically generated

1. Sean has been asked to list the ISBN, category, genre and publisher of all fiction books suitable for adults. He writes a SQL query to display the details needed.

Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
| **ISBN** | **Category** | **Genre** | **Publisher** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Sean is now asked to display the full name and nationality of all American authors. These details should be listed in alphabetical order of author surname. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of first Author listed |  |  |  |
| Details of last Author listed |  |  |  |

1. Sean is asked to display the first name and website of the author with surname ‘Rowling’ together with the ISBN and date of publication of all books written by that author. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Sean is asked to add the website details for the author called Mick Inkpen to the database. The URL of his website is <http://authorpages.hoddersystems.com/MickInkpen/first.asp>. Use the space below to predict the result of this query.

|  |
| --- |
|  |

1. Sean is asked to display the surname of the authors, together with the genre and number of pages, of any fiction books that have more than 200 pages. These details should be listed so that the book with the most pages is listed first. Use the table below to predict the output from the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Details of first Book listed |  |  |  |
| Details of last Book listed |  |  |  |

1. Sean is asked to display the title, category and number of pages, together with the surname of the author, of all children’s books that have fewer than 50 pages. Use the table below to predict the results of the query.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Evaluation
2. In Q1, Sean was asked to list the ISBN, category, genre and publisher of all fiction books suitable for adults.

Here is the answer table produced by Sean’s query for Q1.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q2, Sean was asked to display full name and nationality of all American authors in alphabetical order of author surname.

Here is the answer table produced by Sean’s query for Q2.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q3, Sean was asked to display the first name and website of the author with surname ‘Rowling’ together with the ISBN and date of publication of all books written by that author.

Here is the answer table produced by Sean’s query for Q3.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q4, Sean was asked to add the website details for the author called Mick Inkpen to the database (the URL of his website is <http://authorpages.hoddersystems.com/MickInkpen/first.asp>).

Here is the Author table after it had been updated by the query that Sean created for Q4.

A screenshot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q5, Sean was asked to surname of the authors, together with the genre and number of pages, of any fiction books that have more than 200 pages. These details should be listed so that the book with the most pages is listed first.

Here is the answer table produced by Sean’s query for Q5.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

1. In Q6, Sean was asked to display the title, category and number of pages, together with the surname of the author, of all children’s books that have fewer than 50 pages

Here is the answer table produced by the query that Sean created for Q6.

A screen shot of a computer

Description automatically generated

Look back at your predicted output for this query and compare your prediction with Sean’s solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output