N5 DDD Clydeview

**Tasks and Answers**

[Task 1 Database Structure 1-1](#_Toc187835621)

[A1 Database Structure 1-4](#_Toc187835622)

[Task 2 Queries Introduction 2-1](#_Toc187835623)

[A2 Queries Introduction 2-4](#_Toc187835624)

[Task 3 Sorting 3-1](#_Toc187835625)

[A3 Sorting 3-4](#_Toc187835626)

[Task 4 SQL – Introduction 4-1](#_Toc187835627)

[A4 SQL – Introduction 4-4](#_Toc187835628)

[Task 5 Queries 5-1](#_Toc187835629)

[A5 Queries 5-3](#_Toc187835630)

[Task 6 Query Design (Simple) 6-1](#_Toc187835631)

[A6 Query Design (Simple) 6-4](#_Toc187835632)

[Task 7 Query Implementation (Simple) 7-1](#_Toc187835633)

[A7 Query Implementation 7-4](#_Toc187835634)

[Task 8 Query Design (Complex) 8-1](#_Toc187835635)

[A8 Query Design (Complex) 8-4](#_Toc187835636)

[Task 9 Query Implementation (Complex) 9-1](#_Toc187835637)

[A9 Query Implementation 9-4](#_Toc187835638)

[Task 10 Entity and ERD Design (1) 10-1](#_Toc187835639)

[A10 Entity and ERD Design (1) 10-4](#_Toc187835640)

[Task 11 Entity and ERD Design (2) 11-1](#_Toc187835641)

[A11 Entity and ERD Design (2) 11-4](#_Toc187835642)

[Task 12 Entity and ERD Design (3) 12-1](#_Toc187835643)

[A12 Entity and ERD Design (3) 12-4](#_Toc187835644)

[Task 13 Entity and ERD Design (4) 13-1](#_Toc187835645)

[A13 Entity and ERD Design (4) 13-4](#_Toc187835646)

[Task 14 Entity and ERD Design (5) 14-1](#_Toc187835647)

[A14 Entity and ERD Design (5) 14-4](#_Toc187835648)

[Task 15 Create Tables (1) 15-1](#_Toc187835649)

[A15 Create Tables (1) 15-2](#_Toc187835650)

[Task 16 Entity and ERD Design (6) 16-1](#_Toc187835651)

[A16 Entity and ERD Design (6) 16-3](#_Toc187835652)

[Task 17 Create Tables (2) 17-1](#_Toc187835653)

[A17 Create Tables (2) 17-2](#_Toc187835654)

[Task 18 Entity and ERD Design (7) 18-1](#_Toc187835655)

[A18 Entity and ERD Design (7) 18-4](#_Toc187835656)

[Task 19 Create Tables (3) 19-1](#_Toc187835657)

[A19 Create Tables (3) 19-2](#_Toc187835658)

[Task 20 Entity and ERD Design (8) 20-1](#_Toc187835659)

[A20 Entity and ERD Design (8) 20-4](#_Toc187835660)

[Task 21 Create Tables (4) 21-1](#_Toc187835661)

[A21 Create Tables (4) - TBC 21-3](#_Toc187835662)

[Task 22 Query Design (Equi-join) (1) 22-1](#_Toc187835663)

[A22 Query Design (1) 22-4](#_Toc187835664)

[Task 23 Query Implementation (Equi-join) (1) 23-1](#_Toc187835665)

[A23 Query Implementation (1) 23-4](#_Toc187835666)

[Task 24 Query Design (Equi-join) (2) 24-1](#_Toc187835667)

[A24 Query Design (2) 24-4](#_Toc187835668)

[Task 25 Query Implementation (Equi-join) (2) 25-1](#_Toc187835669)

[A25 Query Implementation (2) – TBC 25-4](#_Toc187835670)

[Task 26 Modify Data – Introduction 26-1](#_Toc187835671)

[A26 Modify Data – Introduction – TBC 26-4](#_Toc187835672)

[Task 27 Modify Data – Design (1) 27-1](#_Toc187835673)

[A27 Modify Data – Design (1) – TBC 27-6](#_Toc187835674)

[Task 28 Modify Data – Implementation (1) 28-1](#_Toc187835675)

[A28 Modify Data – Implementation (1) – TBC 28-3](#_Toc187835676)

[Task 29 Modify Data – Design (2) 29-1](#_Toc187835677)

[A29 Modify Data – Design (2) – TBC 29-6](#_Toc187835678)

[Task 30 Modify Data – Implementation (2) 30-1](#_Toc187835679)

[A30 Modify Data – Implementation (2) – TBC 30-5](#_Toc187835680)

[Task 31 Testing (1) 31-1](#_Toc187835681)

[A31 Testing (1) 31-4](#_Toc187835682)

[Task 32 Evaluation (1) 32-1](#_Toc187835683)

[A32 Evaluation (1) – TBC 32-7](#_Toc187835684)

[Task 33 Testing (2) 33-1](#_Toc187835685)

[A33 Testing (2) 33-4](#_Toc187835686)

[Task 34 Evaluation (2) 34-1](#_Toc187835687)

[A34 Evaluation (2) – TBC 34-7](#_Toc187835688)

[Task 35 Predict - 18 marks 35-1](#_Toc187835689)

[A35 Predict - 18 marks – TBC 35-4](#_Toc187835690)

[Task 36 Run - 10 marks 36-1](#_Toc187835691)

[A36 Run - 10 marks – TBC 36-3](#_Toc187835692)

[Task 37 Investigate - 5 marks 37-1](#_Toc187835693)

[A37 Investigate - 5 marks – TBC 37-2](#_Toc187835694)

[Task 38 Modify - 18 marks 38-1](#_Toc187835695)

[A38 Modify - 18 marks – TBC 38-3](#_Toc187835696)

[Task 39 Make - 12 marks 39-1](#_Toc187835697)

[A39 Make - 12 marks – TBC 39-3](#_Toc187835698)

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

# 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 | 2025-07-23 |
| Time | Used for storing time values | 11:37:00 |
| Boolean | Used for storing True or False | True, False, |

## Part 2

Look at the database table below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| knownAs | surname | branch | accountType | balance | rate | interestDue |
| Leigh | Nelson | Glasgow | Saver | 100.67 | 1.7 | 2024-02-01 |
| Sam | McAteer | Glasgow | Flexi | 304.86 | 2.3 | 2024-03-02 |
| Robyn | Jones | Gourock | ISA | 89.45 | 3.0 | 2024-03-09- |
| Jack | Devlin | Greenock | Saver | 162.91 | 1.7 | 2024-01-29 |

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 | eyeColour |  | 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** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

##### Database Structure

###### Part 2

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

**7**

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

**4**

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

|  |  |
| --- | --- |
| **Field Name** | **Type** |
| knownAs | Text |
| surname | Text |
| branch | Text |
| accountType | Text |
| balance | Number |
| rate | Number |
| interestDue | Date |

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

*Example answers, anything sensible is valid.*

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

# 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** | **hairColour** |
| Patrick | Grant | 1986-04-19 | 29 | Burglary | 2 years | 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 found guilty of hacking.

|  |
| --- |
| 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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

##### Queries Introduction

###### Part 2

1. Find all the criminals with black hair.

**Black hairColour**

1. Find all the criminals with the surname Smith.

**Smith surname**

1. Find all the criminals aged 35.

**35 age**

1. Find all the criminals with found guilty of hacking

**Hacking crime**

1. Find all the criminals with blonde hair and sentenced to 5 years

**Blonde hairColour**

**5 years sentence**

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

**Steven firstName**

**19 age**

# 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 |

1s Field: Order:

2nd Filed: Order:

### 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 |

1s Field: Order:

2nd Filed: Order:

##### 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** |
| Order | **ASC** | **DESC** | **DESC** | **ASC** |

###### Part 2

Table 1

Field used to sort the table: **age**

Order applied: **ASC**

Table 2

Field used to sort the table: **firstName**

Order applied: **ASC**

Table 3

Field used to sort the table: **surname**

Order applied: **ASC**

###### Part 3

Table 1

1s Field: **Year ASC**

2nd Filed: **Class ASC**

Table 2

1st sort: **Surname ASC**

2nd sort: **Class DESC**

# 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 use database table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

### Query 2

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

This query will use database table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

### Query 3

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

This query will use database table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

##### SQL – Introduction

###### Part 1

1. What does SQL stand for?

**Structured Query Language**

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

This query will use database table called **Pupil**

It will choose any record where **age > 16**

The selected records will then be sorted **firstName ASC**

It will then display the fields **firstName, surname, age**

Query 2

This query will use database table called **Stock**

It will choose any record where **price < 9.99**

The selected records will then be sorted **price DESC**

It will then display the fields **itemName, price, quantity**

Query 3

This query will use database table called **Customer**

It will choose any record where **title = "Mr"**

The selected records will then be sorted **dob ASC**

It will then display the fields **surname, address, postCode**

# 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 use the database table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

### 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 use the database table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

### 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 use database the table called

It will choose any record where

The selected records will then be sorted

It will then display the fields

##### Queries

###### Tasks

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

Query 1

This query will use database table called **Pupil**

It will choose any record where **age > 16 AND surname = "Smith"**

The selected records will then be sorted **firstName ASC**

It will then display the fields **firstName, surname, age**

Query 2

This query will use database table called **Stock**

It will choose any record where **price < 9.99** **OR quantity > 5**

The selected records will then be sorted **price DESC**

It will then display the fields **itemName, price, quantity**

Query 3

This query will use database table called **Customer**

It will choose any record where **title = "Mr"** **AND age > 50**

The selected records will then be sorted **dob ASC**

It will then display the fields **surname, address, postCode**

# 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 Design (Simple)

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

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

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

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

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

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

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

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

# Query Implementation (Simple)

Open the Clydeview.db database file and use the query designs from Task 6 to help create the SQL statement for each task. Paste the SQL, as text, and the output, as an image, below.

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

|  |
| --- |
|  |

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.

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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.

|  |
| --- |
|  |

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.

|  |
| --- |
|  |

##### Query Implementation

SELECT forename, surname

FROM Customer

WHERE surname = "Rice";

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

SELECT forename, surname, town

FROM Customer

WHERE town = "Inverkip";

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

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 |

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 |

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 |

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 |



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 |

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 |

# 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 Design (Complex)

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

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

**Marks: 32**

# Query Implementation (Complex)

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.

|  |
| --- |
|  |

##### 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.

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 |

# Entity and ERD Design (1)

## Information

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 |

## Design

### 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, including attributes, 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 and ERD Design (1)

###### 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.

# Entity and ERD Design (2)

## Information

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** | **File Name** |
| 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 |

## Design

### 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, including attributes, 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 and ERD Design (2)

###### Album entity

1. Complete the data dictionary for the Album entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| albumID | PK | Number |  | Y |  |
| name |  | Text | 30 | N |  |
| category |  | Text | 7 | Y | Restricted choice: animals, cars, castles, surfing, towns |
| description |  | Text | 255 | N |  |
| noOfPhotos |  | Number |  | Y | Range: >= 0 and <= 120 |

###### Photo entity

1. Complete the data dictionary for the Photo entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| photoID | PK | Number |  | Y |  |
| albumID | FK | Number |  | Y | albumID exists in Album table |
| title |  | Text | 50 | N |  |
| fileName |  | Text | 50 | 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 is 1 to Many, i.e. An album stores many photos.

# Entity and ERD Design (3)

## Information

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 |

## Design

### 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, including attributes, 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 and ERD Design (3)

###### Item entity

1. Complete the data dictionary for the Item entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| code | PK | Text | 7 | Y | Length = 7 |
| description |  | Text | 50 |  |  |
| size |  | Number |  |  | Restricted choice: 8, 10, 12, 14, 16 |
| era |  | Text | 5 |  | Restricted choice: 1940s, 1950s, 1960s, 1970s |
| brandID | FK | Text | 4 | Y | brandID exists in Brand table |

###### Brand entity

1. Complete the data dictionary for the Brand entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| brandID | PK | Text | 4 | Y |  |
| brand |  | Text | 30 |  |  |
| established |  | Number |  |  |  |
| nationality |  | Text | 8 |  | Restricted choice: American, British, Italian |



1. The relationship is Many to 1, i.e. many items are created by a brand.

# Entity and ERD Design (4)

## Information

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 |

## Design

### 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, including attributes, 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 ERD Design (4)

###### Player entity

1. Complete the data dictionary for the Player entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| playerID | PK | Text | 4 | Y | Length = 4 |
| teamID | FK | Number |  | Y | teamID exists in Team table |
| starRating |  | Number |  |  | Range >= 1 AND <= 5 |
| firstName |  | Text | 20 |  |  |
| lastName |  | Text | 30 |  |  |
| dob |  | Date |  |  |  |

###### Team entity

1. Complete the data dictionary for the Team entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| teamID | PK | Number |  | Y |  |
| name |  | Text | 20 |  |  |
| sport |  | Text | 10 |  | Restricted choice: basketball, handball, hockey, netball |
| manager |  | Text | 30 |  |  |
| coach |  | Text | 30 |  |  |
| homeTown |  | Text | 30 |  |  |

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.

The relationship is One to Many, i.e. a team has many players.

# Entity and ERD Design (5)

## Part 1 – Design

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

The details of pet owners are stored 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 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, including attributes, to represent the relationship between the Owner and Pet entities.

|  |
| --- |
|  |

## Part 4 – Relationship

1. Describe the type of relationship that exists between the Pet and Owner tables.

|  |
| --- |
|  |

##### Entity and ERD Design (5)

###### Part 1 – Design

Owner Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| ownerID | PK | Number |  | Y |  |
| firstName |  | Text | 20 |  |  |
| surname |  | Text | 30 |  |  |
| address |  | Text | 50 |  |  |
| town |  | Text | 20 |  |  |
| telephone |  | Text | 13 |  |  |

###### Part 2 – Design

Pet Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| code | PK | Text |  | Y |  |
| name |  | Text | 20 |  |  |
| type |  | Text | 8 |  | Restricted choice: Cat, Dog, Budgie, Gerbil, Tortoise |
| dob |  | Date |  |  |  |
| vaccination |  | Boolean |  |  |  |
| ownerID | FK | Number |  | Y | ownerID exists in Owner table |

###### Part 3 – ERD



# Create Tables (1)

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.

##### Create Tables (1)

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";

# Entity and ERD Design (6)

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 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.

|  |
| --- |
|  |

## Part 4 – Relationship

1. Describe the type of relationship that exists between the Manufacturer and Product tables.

|  |
| --- |
|  |

##### Entity and ERD Design (6)

###### Part 1 – Design

Product Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| name |  | Text | 30 |  |  |
| code | PK | Text | 4 | Y |  |
| stock |  | Number |  | Y | Range: >= 0 and <= 50 |
| onOrder |  | Boolean |  |  |  |
| price |  | Number |  | Y | range: > 1.00 |
| manufacturerID | FK | Number |  | Y | manufacturerID 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** |
| manufacturerID | PK | Number |  | Y |  |
| name |  | Text | 30 |  |  |
| address |  | Text | 50 |  |  |
| telephone |  | Text | 13 |  |  |

###### Part 3 – ERD



*The relationship description can be different..*

# Create Tables (2)

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.

##### Create Tables (2)

CREATE TABLE Manufacturer (

manufacturerID INT NOT NULL,

name VARCHAR(30),

address VARCHAR(50),

telephone VARCHAR(13),

PRIMARY KEY (manufacturer\_id)

);

CREATE TABLE Product (

name VARCHAR(30),

code VARCHAR(4) NOT NULL,

stock INT NOT NULL

CHECK (stock >= 0 AND stock <= 50),

onOrder BOOL,

price REAL NOT NULL

CHECK (price > 1.00),

manufacturerID INT NOT NULL,

FOREIGN KEY (manufacturerID)

REFERENCES Manufacturer (manufacturerID),

PRIMARY KEY (code)

);

PRAGMA foreign\_keys = on;

SELECT sql

FROM sqlite\_schema

WHERE tbl\_name = "Product"

OR tbl\_name = "Manufacturer";

# Entity and ERD Design (7)

## Information

An online music store uses a relational database 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

## Design

### 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** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### ERD

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

|  |
| --- |
|  |

##### Entity and ERD Design (7)

###### Part 1 – Design

CD Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Key** | **Type** | **Size** | **Req’d** | **Validation** |
| code | PK | Text | 4 | Y |  |
| title |  | Text | 30 |  |  |
| artist |  | Text | 30 |  |  |
| label | FK | Text | 30 | Y | label 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 |  |  |  |
| parentCompany |  | Text | 30 |  |  |
| country |  | Text | 7 | Y | Restricted choice: Germany, Japan, UK, USA |
| website |  | Text | 50 |  |  |

###### Part 2 – ERD



# Create Tables (3)

Open the N5 DDD Clydeview Blank.

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

##### Create Tables (3)

CREATE TABLE Label (

label VARCHAR(30) NOT NULL,

founded INT,

parentCompany VARCHAR(30),

country VARCHAR(7) NOT NULL

CHECK (country IN ("Germany", "Japan", "UK", "USA")),

website VARCHAR(50),

PRIMARY KEY (label)

);

1. Remember to enforce referential integrity between the two tables.

CREATE TABLE CD (

code VARCHAR(4) NOT NULL,

title VARCHAR(30),

artist VARCHAR(30),

label VARCHAR(30) NOT NULL,

noOfTracks INT NOT NULL

CHECK (noOfTracks >= 10 AND noOfTracks <= 60),

cost REAL NOT NULL

CHECK (cost >= 6.99 AND cost <= 15.00),

genre VARCHAR(7) NOT NULL

CHECK (genre IN ("Choral", "Country", "Garage", "Indie",

"Opera", "Pop", "R&B", "R&R", "Soul")),

FOREIGN KEY (label)

REFERENCES Label (label),

PRIMARY KEY (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 = "Label"

OR tbl\_name = "CD";

# Entity and ERD Design (8)

## Information

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** | **Published** | **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

## Design

### 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** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### ERD

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

|  |
| --- |
|  |

##### Entity and ERD Design (8)

###### Design

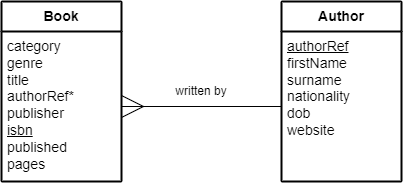
Author Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  | Text | *255* |  |  |
| authorRef | FK | Number |  | Y | Exists in Author table |
| publisher |  | Text | *30* |  |  |
| isbn | PK | Text | 10 (13) | Y |  |
| published |  | Date |  |  |  |
| pages |  | Number |  | Y | Range: >=32 and <= 950 |

###### ERD



# Create Tables (4)

###### Design

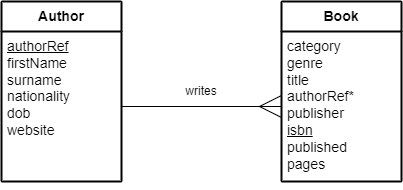
Author Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  | Text | *255* |  |  |
| authorRef | FK | Number |  | Y | Exists in Author table |
| publisher |  | Text | *30* |  |  |
| isbn | PK | Text | 10 (13) | Y |  |
| published |  | Date |  |  |  |
| pages |  | Number |  | Y | Range: >=32 and <= 950 |

###### ERD



###### Implementation

1. Open the N5 DDD Clydeview Blank database file.
2. Create the Author and Book tables, and make sure that the properties match all of the data dictionaries.
3. Remember to enforce referential integrity between the two tables.
4. Paste below the SQL used to create the tables

##### Create Tables (4) - TBC

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 (Equi-join) (1)

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 Design (1)

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 |

# Query Implementation (Equi-join) (1)

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 Implementation (1)

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 |

# Query Design (Equi-join) (2)

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) |  |
| 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 Design (2)

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 |

# Query Implementation (Equi-join) (2)

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.

|  |
| --- |
|  |

##### Query Implementation (2) – TBC

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

SELECT Product.name, price, Manufacturer.name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND Manufacturer .name = "Craft Supplies";

|  |  |  |
| --- | --- | --- |
| **Product.name** | **price** | **Manufacturer.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 Manufacturer name, address, telephone, price

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND price > 100;

|  |  |  |  |
| --- | --- | --- | --- |
| **Manufacturer.name** | **address** | **telephone** | **price** |
| 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 Manufacturer.name, Product.name, price

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND onOrder = TRUE;

|  |  |  |
| --- | --- | --- |
| **Manufacturer.name** | **Product.name** | **price** |
| 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 Product.name, Manufacturer.name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND stock >= 12;

|  |  |
| --- | --- |
| **Product.name** | **Manufacturer.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 Product.name, price, Manufacturer.name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND Manufacturer.name = "Tool Makers"

ORDER BY price ASC;

|  |  |  |
| --- | --- | --- |
| **Product.name** | **price** | **Manufacturer.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 Manufacturer.name, address, onOrder, stock

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND onOrder = FALSE

ORDER BY stock DESC;

|  |  |  |  |
| --- | --- | --- | --- |
| **Manufacturer.name** | **address** | **onOrder** | **stock** |
| 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 Product.name, stock, price, Manufacturer.name

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND price < 20

ORDER BY Manufacturer.name ASC,

price DESC;

|  |  |  |  |
| --- | --- | --- | --- |
| **Product.name** | **stock** | **price** | **Manufacturer.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.

SELECT Manufacturer.name, address, telephone, code, stock

FROM Manufacturer, Product

WHERE Manufacturer.manufacturerID = Product.manufacturerID

AND stock > 2

ORDER BY Manufacturer.name ASC,

stock DESC;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **address** | **telephone** | **code** | **stock** |
| Craft Supplies | Wishaw Industrial Estate | 01415437212 | MTF5 | 9 |
| Craft Supplies | Wishaw Industrial Estate | 01415437212 | PLS6 | 5 |
| Metal and Wood | Tyne Way Newcastle | 01542123485 | MPB2 | 34 |
| Metal and Wood | Tyne Way Newcastle | 01542123485 | MA16 | 7 |
| Metal and Wood | Tyne Way Newcastle | 01542123485 | SW22 | 4 |

# 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **customerNo** | **forename** | **surname** | **street** | **town** | **package** | **directDebit** | **paymentDue** |
| 101 | Lauren | Calder | 2 Paisley Street | Wemyss Bay | Premier | TRUE | 2025-04-11 |
| 102 | Abigail | Cameron | 16 Paisley Street | Wemyss Bay | Standard | FALSE | 2025-04-11 |
| 103 | Ryan | Collins | 17 Dunoon Drive | Gourock | Premier | TRUE | 2025-05-18 |
| 104 | Nicole | Rutherford | 5A Panama Place | Port Glasgow | Large | FALSE | 2025-04-12 |
| 105 | Justine | O'Docherty | 7 High Street | Kilmacolm | Premier | TRUE | 2025-04-17 |
| 106 | Shelby | Sweeney | 3 Paisley Road | Port Glasgow | Premier | TRUE | 2025-05-25 |
| 107 | Donald | McAndrew | 1 Big Hill Avenue | Inverkip | Standard | TRUE | 2025-07-31 |
| 108 | Rowan | Hastings | 6 Clydeview Crescent | Gourock | Large | TRUE | 2025-05-04 |
| 109 | Grant | Donaldson | 9 Dunoon Drive | Gourock | Premier | FALSE | 2025-04-06 |
| 110 | Christine | Flowers | 63 Hamilton Drive | Greenock | Large | TRUE | 2025-04-18 |
| 111 | Ross | Lambie | 12 Paisley Road | Kilmacolm | Standard | TRUE | 2025-03-26 |
| 112 | Paul | McGill | 3C Cow Lane | Kilmacolm | Premier | FALSE | 2025-04-09 |
| 113 | Jack | Shields | 4 Brookside Close | Port Glasgow | Large | FALSE | 2025-05-09 |
| 114 | Pauline | Milne | 32 High Street | Kilmacolm | Premier | TRUE | 2025-05-08 |
| 115 | Margaret | Rice | 5 Drumchapel Square | Greenock | Standard | FALSE | 2025-04-13 |

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, "Niall", "Davies", "12 Clyde View",

3             "Gourock", "Premier", TRUE, "2025-05-08");

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 paymentDue = "2025-06-01";

Query 4 – Description

|  |
| --- |
|  |

## Query 5 – SQL

1 INSERT INTO Customer (forename, surname, directDebit,   
 package, paymentDue, street,   
 town, customerNo)

2 VALUES ("Kelly", "Holmes", FALSE,   
 "Standard", "2025-05-16", "5 Lime Grove",   
 "Greenock", 126);

Query 5 – Description

|  |
| --- |
|  |

##### Modify Data – Introduction – TBC

###### Query 1 – SQL

Query 1 – Description

###### Query 2 – SQL

Query 2 – Description

###### Query 3 – SQL

Query 3 – Description

###### Query 4 – SQL

Query 4 – Description

###### Query 5 – SQL

Query 5 – Description

# Modify Data – Design (1)

ClydeVet veterinary practice uses a relational database store details pets and their owners. The ERD for Owner and Pet entities 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.

1. 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 |  |

1. 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 |  |

1. 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 |  |

1. 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 |  |

1. 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 |  |

1. Remove details of Slinky the tortoise from the database.

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

1. 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 |  |

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

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

1. 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 |  |

1. 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 |  |

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

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

##### Modify Data – Design (1) – TBC

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

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

1. 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 |  |

1. 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 |  |

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

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

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

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

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

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

1. Remove details of Slinky the tortoise from the database.

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

1. 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 |  |

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

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

1. 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 |  |

1. 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 |  |

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

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

# Modify Data – Implementation (1)

Use your query designs from A26 to help create the SQL statement for each task.

1. 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 |

1. The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.
2. The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.
3. 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 |

1. 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 |

1. Remove details of Slinky the tortoise from the database.
2. The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.
3. Remove the details of the owner with Owner ID 3510 from the database.
4. 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.
5. 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.
6. Remove the details of the pet called Usain from the database.

##### Modify Data – Implementation (1) – TBC

1. Add the details of Goldie the dog to the Pet table.
2. The pet with Pet Code P1559 has just received its vaccination. Edit the correct record of the database.
3. The owner with Owner ID 2356 has changed her contact number to 07723456789. Edit the correct record of the database.
4. Add the following details of a new owner and their pet to the database.
5. Add the following details of a new owner and their pet to the database.
6. Remove details of Slinky the tortoise from the database.
7. The owner called Sally Chan has moved house. Her new address is 64 Lochview Road, Gourock. Edit the correct record of the database.
8. Remove the details of the owner with Owner ID 3510 from the database.
9. 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.
10. 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.
11. Remove the details of the pet called Usain from the database.

# Modify Data – Design (2)

A hardware store uses a relational database to store details of the products for sale and the manufacturer of each product. The ERD for the Product and Manufacturer entities 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 |  |

##### Modify Data – Design (2) – TBC

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

|  |  |
| --- | --- |
| 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.

|  |  |
| --- | --- |
| 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.

|  |  |
| --- | --- |
| 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 (2)

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

Use your query designs from A28 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.

|  |
| --- |
|  |

##### Modify Data – Implementation (2) – TBC

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.
2. 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.
3. 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.
4. Remove the saw with Product Code SW22 form the database.
5. Add the following manufacturer and product details to the database.
6. Add the following manufacturer and product details to the database.
7. Remove all products manufactured by the manufacturer with ID 441 from the database.
8. 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.
9. Remove the details of the manufacturer called Craft Supplies from the database.
10. Increase the number in stock of all products by 2.
11. Reduce by £5 the cost of all products that are manufactured by the manufacturer with ID 627.

# Testing (1)

## Information

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.



### Table: CD

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

### Table: Label

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

## Tasks

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.

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **title** | **label** | **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.

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

##### Testing (1)

SELECT code, title, label, tracks

FROM CD

WHERE tracks < 14;

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **title** | **label** | **tracks** |
| 5J8Y | + | Atlantic Records | 13 |
| 82FK | The Power of Love | Syco Music | 11 |
| 8G9K | iDos! | Warner Bros | 13 |
| 91TU | Right Place Right Time | Epic Records | 12 |
| 942Y | Take Me Home | Syco Music | 13 |

SELECT artist, CD.label, genre

FROM Label, CD

WHERE Label.label = CD.label

AND country = "Germany";

|  |  |  |
| --- | --- | --- |
| **artist** | **label** | **genre** |
| The Rolling Stones | Polydor Records | R&R |

SELECT label, founded, country, website

FROM Label

WHERE founded < 1965

AND country = "USA";

|  |  |  |  |
| --- | --- | --- | --- |
| **label** | **founded** | **country** | **website** |
| Atlantic Records | 1947 | USA | www.atlanticrecords.com |
| Epic Records | 1953 | USA | www.epicrecords.com |

SELECT title, genre, cost

FROM Label, CD

WHERE Label.label = CD.label

AND country = "USA"

AND cost < 10;

|  |  |  |
| --- | --- | --- |
| **title** | **genre** | **cost** |
| The Best of | Pop | 9.99 |
| Our Version of Events | R&B | 8.99 |

SELECT code, label, tracks

FROM CD

WHERE label = "Syco Music"

ORDER BY tracks DESC;

|  |  |  |
| --- | --- | --- |
| **code** | **label** | **tracks** |
| 9KYX | Syco Music | 20 |
| 942Y | Syco Music | 13 |
| 82FK | Syco Music | 11 |

SELECT title, artist, CD.label, tracks

FROM Label, CD

WHERE Label.label = CD.label

AND country = "UK"

AND tracks <= 15;

|  |  |  |  |
| --- | --- | --- | --- |
| **title** | **artist** | **label** | **tracks** |
| Stronger Together | Military Wives | Decca Records | 14 |
| The Power of Love | Sam Bailey | Syco Music | 11 |
| Take Me Home | One Direction | Syco Music | 13 |

# Evaluation (1)

1. A query was required to select the CD code, title, label and number of tracks of all the CDs that have fewer than 14 tracks.

The predicted output is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **title** | **label** | **tracks** |
| 5J8Y | + | Atlantic Records | 13 |
| 82FK | The Power of Love | Syco Music | 11 |
| 8G9K | iDos! | Warner Bros | 13 |
| 91TU | Right Place Right Time | Epic Records | 12 |
| 942Y | Take Me Home | Syco Music | 13 |

The actual result of the query is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **title** | **label** | **tracks** |
| 5J8Y | + | Atlantic Records | 13 |
| 61DS | Stronger Together | Decca Records | 14 |
| 82FK | The Power of Love | Syco Music | 11 |
| 8G9K | iDos! | Warner Bros | 13 |
| 91TU | Right Place Right Time | Epic Records | 12 |
| 942Y | Take Me Home | Syco Music | 13 |

Look at the predicted output for this query and compare it with the actual output.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the artist name, label and genre of any CDs produced by a German record label.

The predicted output is shown below:

|  |  |  |
| --- | --- | --- |
| **artist** | **label** | **genre** |
| The Rolling Stones | Polydor Records | R&R |

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **artist** | **label** | **genre** |
| The Rolling Stones | Polydor Records | R&R |

Look at the predicted output for this query and compare it with actual solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the label name, year founded, country of origin and website of any record label that originated in the USA before 1965.

The predicted output is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **label** | **founded** | **country** | **website** |
| Atlantic Records | 1947 | USA | www.atlanticrecords.com |
| Epic Records | 1953 | USA | www.epicrecords.com |

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **label** | **founded** | **country** |
| Atlantic Records | 1947 | USA |
| Epic Records | 1953 | USA |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the title, genre and cost of any CDs that were produced by record labels that originated in the USA and cost less than £10.

The predicted output is shown below:

|  |  |  |
| --- | --- | --- |
| **title** | **genre** | **cost** |
| The Best of | Pop | 9.99 |
| Our Version of Events | R&B | 8.99 |

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **title** | **genre** | **cost** |
| The Best of | Pop | 9.99 |
| Our Version of Events | R&B | 8.99 |

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the CD code, label and number of tracks of any CD produced by Syco Music. These details were to be ordered so that the CD with the most tracks appears first.

The predicted output is shown below:

|  |  |  |
| --- | --- | --- |
| **code** | **label** | **tracks** |
| 9KYX | Syco Music | 20 |
| 942Y | Syco Music | 13 |
| 82FK | Syco Music | 11 |

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **code** | **label** | **tracks** |
| 82FK | Syco Music | 11 |
| 942Y | Syco Music | 13 |
| 9KYX | Syco Music | 20 |

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the title, artist, label and number of tracks of and CDs that were produced by UK record labels that have 15 or fewer tracks.

The predicted output is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **title** | **artist** | **label** | **tracks** |
| Stronger Together | Military Wives | Decca Records | 14 |
| The Power of Love | Sam Bailey | Syco Music | 11 |
| Take Me Home | One Direction | Syco Music | 13 |

The actual result of the query is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **title** | **artist** | **label** | **tracks** |
| The Power of Love | Sam Bailey | Syco Music | 11 |
| Take Me Home | One Direction | Syco Music | 13 |
| Stronger Together | Military Wives | Decca Records | 14 |

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

##### Evaluation (1) – TBC

SELECT code, title, label, tracks

FROM CD

WHERE tracks <= 14;

* The query is not fit for purpose because the answer table displays details of 5 CDs including Stronger Together which has 14 tracks.
* The output from of the query is accurate because the answer table includes all of the four required details (CD code, title, label and number of tracks) meaning that nothing is missing from the output.
* The query is fit for purpose because it displays details of the only CD produced by a German record label.
* The output from the query is accurate because only the specified fields are included and nothing is missing.
* The query is fit for purpose because it has returned the details of the two CDs produced by US record labels which were founded before 1965.
* The output from the query is not accurate because the answer table only includes three of the required details (label, year founded and country of origin) and details of the website are missing.
* The query is fit for purpose because it has returned the details of the two CDs produced by US record labels which cost less than £10.
* The output from the query is accurate because the answer table includes all three of the details required (title, genre and cost) meaning that nothing is missing from the output.

SELECT code, label, tracks

FROM CD

WHERE label = "Syco Music"

ORDER BY tracks ASC;

* The query is not fit for purpose because the details have been displayed in ascending order and the CD with the smallest number of tracks was listed first instead of last.
* The output from the query is correct because all of the required details (CD code, label and number of tracks) have been displayed and no details have been omitted.

SELECT title, artist, CD.label, tracks

FROM Label, CD

WHERE Label.label = CD.label

AND country = "UK"

AND tracks <= 15

ORDER BY tracks ASC;

* The query is fit for purpose because the answer table shows details of the only three CDs produced by UK labels that have 15 or fewer tracks.
* The output from the query is accurate because all four of the required details (title, artist, label and number of tracks) are included in the answer table and nothing is missing.

# Testing (2)

## Information

Clydeview Library uses a relational database 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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **authorRef** | **forename** | **surname** | **nationality** | **dob** | **website** |
| 1862 | Nick | Hornby | British | 1957-04-17 | www.nickhornbyofficial.com |
| 1902 | Gareth | Owen | British |  | www.garethowen.com |
| 2002 | Eric | Carle | American | 1929-06-25 | www.eric-carle.com |
| 2715 | Jon | Blake | British |  | jonblakeblog.wordpress.com |
| 2864 | Kenneth | Oppel | Canadian | 1967-08-31 | www.kennethoppel.ca |
| 3197 | Joanne | Rowling | British | 1965-07-31 | www.jkrowling.com |
| 3390 | Patricia | Cornwell | American | 1956-06-09 | www.patriciacornwell.com |
| 3507 | Nick | Hunter | British |  |  |
| 3713 | Mick | Inkpen | British |  |  |
| 4079 | Jack | Higgins | British | 1929-07-27 |  |
| 4097 | Mary | Berry | British | 1935-03-24 |  |
| 4231 | Nicholas | Evans | British | 1950-07-26 | www.nicholasevens.com |

### Book table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **category** | **genre** | **title** | **authorRef** | **publisher** | **ISBN** | **published** | **pages** |
| Adult | Fiction | About a Boy | 1862 | Indigo | 0575400951 | 2009-09-22 | 278 |
| Child | Fantasy | Galactic Snapshots | 2864 | Puffin | 0140373683 | 2010-08-03 | 96 |
| Child | Mystery | Harry Potter and the Chamber of Secrets | 3197 | Bloomsbury | 0747538492 | 1998-07-02 | 251 |
| Child | Sport | London Olympics 2012 | 3507 | Raintree | 1406223948 | 2011-07-05 | 32 |
| Adult | Thriller | Point of Origin | 3390 | Sphere | 0751544787 | 2009-08-23 | 400 |
| Adult | Autobiography | Recipe for Life | 4097 | Penguin | 1405912855 | 2014-02-27 | 432 |
| Child | Fiction | Roboskool | 2715 | Puffin | 0140363240 | 2005-03-12 | 79 |
| Child | Joke | Say Cheese | 1902 | Corgi | 0552529753 | 1996-11-07 | 64 |
| Adult | Fiction | The Casual Vacancy | 3197 | Little Brown Company | 0751552860 | 2012-09-27 | 503 |
| Adult | Fiction | The Horse Whisperer | 4231 | Sphere | 0751539368 | 2006-08-10 | 448 |
| Adult | Mystery | The President's Daughter | 4079 | Penguin | 0140269061 | 2011-11-14 | 400 |
| Child | Fiction | The Very Hungry Caterpillar | 2002 | Puffin | 2067152998 | 1990-09-29 | 26 |
| Child | Fiction | Threadbear | 3713 | Hodder and Stoughton | 0340531290 | 2007-07-05 | 32 |

## Tasks

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 Mary Berry to the database. Her website address is [www.maryberry.co.uk](http://www.maryberry.co.uk)

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.

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

##### Testing (2)

SELECT ISBN, category, genre, publisher

FROM Book

WHERE genre = "Fiction"

AND category = "Adult";

|  |  |  |  |
| --- | --- | --- | --- |
| **ISBN** | **category** | **genre** | **publisher** |
| 0575400951 | Adult | Fiction | Indigo |
| 0751552860 | Adult | Fiction | Little Brown Company |
| 0751539368 | Adult | Fiction | Sphere |

SELECT forename, surname, nationality

FROM Author

WHERE nationality = "American"

ORDER BY surname ASC;

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **nationality** |
| Eric | Carle | American |
| Patricia | Cornwell | American |

SELECT forename, website, ISBN, published

FROM Author, Book

WHERE Author.authorRef = Book.authorRef

AND surname = "Rowling";

|  |  |  |  |
| --- | --- | --- | --- |
| **forename** | **website** | **ISBN** | **published** |
| Joanne | www.jkrowling.com | 0747538492 | 1998-07-02 |
| Joanne | www.jkrowling.com | 0751552860 | 2012-09-27 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **authorRef** | **forename** | **surname** | **nationality** | **dob** | **website** |
| 4097 | Mary | Berry | British | 1935-03-24 | [www.maryberry.co.uk](http://www.maryberry.co.uk) |

SELECT surname, genre, pages

FROM Author, Book

WHERE Author.authorRef = Book.authorRef

AND genre = "Fiction"

AND pages > 200

ORDER BY pages DESC;

|  |  |  |
| --- | --- | --- |
| **surname** | **genre** | **pages** |
| Rowling | Fiction | 503 |
| Evans | Fiction | 448 |
| Hornby | Fiction | 278 |

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.

SELECT title, category, pages, surname

FROM Author, Book

WHERE Author.authorRef = Book.authorRef

AND category = "Child"

AND pages < 50;

|  |  |  |  |
| --- | --- | --- | --- |
| **title** | **category** | **pages** | **surname** |
| London Olympics 2012 | Child | 32 | Hunter |
| The Very Hungry Caterpillar | Child | 26 | Carle |
| Threadbear | Child | 32 | Inkpen |

# Evaluation (2)

1. A query was required to select the ISBN, category, genre and publisher of all fiction books suitable for adults.

The predicted output is shown below:

The actual result of the query is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **ISBN** | **category** | **genre** | **authorRef** |
| 0575400951 | Adult | Fiction | 1862 |
| 0751552860 | Adult | Fiction | 3197 |
| 0751539368 | Adult | Fiction | 4231 |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select the full name and nationality of all American authors in alphabetical order of author surname.

The predicted output is shown below:

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **forename** | **surname** | **nationality** |
| Eric | Carle | American |
| Patricia | Cornwell | American |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select 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.

The predicted output is shown below:

The actual result of the query is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **forename** | **published** | **ISBN** | **website** |
| Gareth | 1996-11-07 | 0552529753 | www.garethowen.com |
| Joanne | 1998-07-02 | 0747538492 | www.jkrowling.com |
| Joanne | 2012-09-27 | 0751552860 | www.jkrowling.com |
| Mary | 2014-02-27 | 1405912855 | www.maryberry.co.uk |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to add the website details for the author called Mary Berry to the database. Her website address is [www.maryberry.co.uk](http://www.maryberry.co.uk)

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **authorRef** | **forename** | **surname** | **nationality** | **dob** | **website** |
| 1862 | Nick | Hornby | British | 1957-04-17 | www.nickhornbyofficial.com |
| 1902 | Gareth | Owen | British |  | www.garethowen.com |
| 2002 | Eric | Carle | American | 1929-06-25 | www.eric-carle.com |
| 2715 | Jon | Blake | British |  | jonblakeblog.wordpress.com |
| 2864 | Kenneth | Oppel | Canadian | 1967-08-31 | www.kennethoppel.ca |
| 3197 | Joanne | Rowling | British | 1965-07-31 | www.jkrowling.com |
| 3390 | Patricia | Cornwell | American | 1956-06-09 | www.patriciacornwell.com |
| 3507 | Nick | Hunter | British |  |  |
| 3713 | Mick | Inkpen | British |  |  |
| 4079 | Jack | Higgins | British | 1929-07-27 | www.maryberry.co.uk |
| 4097 | Mary | Berry | British | 1935-03-24 |  |
| 4231 | Nicholas | Evans | British | 1950-07-26 | www.nicholasevens.com |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select 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.

The predicted output is shown below:

The actual result of the query is shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **forename** | **surname** | **genre** | **pages** |
| Mary | Berry | Autobiography | 432 |
| Patricia | Cornwell | Thriller | 400 |
| Jack | Higgins | Mystery | 400 |
| Joanne | Rowling | Mystery | 251 |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

1. A query was required to select 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

The predicted output is shown below:

The actual result of the query is shown below.

|  |  |  |
| --- | --- | --- |
| **title** | **category** | **pages** |
| Galactic Snapshots | Child | 96 |
| Harry Potter and the Chamber of Secrets | Child | 251 |
| Roboskool | Child | 79 |
| Say Cheese | Child | 64 |

Look back at the predicted output for this query and compare it with the solution.

Evaluate this solution in terms of:

* its fitness for purpose
* the accuracy of the output

|  |
| --- |
|  |

##### Evaluation (2) – TBC

* The query is fit for purpose because the answer table displays details the 3 records with details of fiction books written for adults.
* The output from of the query is not accurate because the answer table includes the author surname which was not required but doesn’t show the book publisher.
* The query is fit for purpose because it displays the only 2 records that have details of American authors and the details are arranged in alphabetical order of author surname.
* The output from the query is accurate because the answer table includes all three of the details required (first name, surname and nationality) meaning that nothing is missing from the output.
* The query is not fit for purpose as two extra records are shown.
* The output from the query is accurate because all the fields are included so nothing is missing.
* The query is not fit for purpose because the wrong record has been updated and the record for Nick Hunter now stores the website details for Mick Inkpen.
* The output from the query is accurate because the content of the website field has been updated with the correct website URL.
* The query is not fit for purpose as the wrong query has been used.
* The output from the query is not accurate because firstName was included and was not required.
* The query is not fit for purpose. Although the answer table only shows includes records with details of children’s books, in each case, the number of pages is over 50: he query asked for details of books that had less than 50 pages.
* The output from the query is not accurate because the answer table only shows three of the required fields (title, category and number of pages). The author surname is missing.

# Predict - 18 marks

Read the following pieces of SQL along with the corresponding table to identify what you think the result tables will be.

## Car Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **regPlate** | **make** | **model** | **ownerID** | **mileage** | **servicePrice** |
| SP77JLP | BMW | 4 Series | 0001 | 40,000 | 125.00 |
| AD88KUE | BMW | M4 | 0002 | 200 | 120.00 |
| YU21UPM | Hyundia | Veloster | 0004 | 42,145 | 75.00 |
| PR43QQS | Tesla | Model 3 | 0003 | 60,554 | 125.00 |
| FD66HFD | Tesla | Model S | 0004 | 80,000 | 500.00 |

## Owner Table

|  |  |  |  |
| --- | --- | --- | --- |
| **ownerID** | **name** | **city** | **job** |
| 0001 | James | Aberdeen | Doctor |
| 0002 | Hannah | Perth | Teacher |
| 0003 | Jennifer | Inverness | Nurse |
| 0004 | Matt | Aberdeen | Retail Manager |

1 a. [3 mark]

|  |
| --- |
| SELECT make, model, mileage  FROM Car  WHERE make = "Tesla"; |

Fill in the expected results table (the first row will be field names):

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

b. [4 marks]

|  |
| --- |
| SELECT regPlate, make, model, mileage, servicePrice  FROM Car  WHERE mileage >= 40,000  AND servicePrice > 100.00  ORDER BY make DESC,  Mileage ASC; |

Fill in the expected results table (the first row will be field names):

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

c. [3 marks]

|  |
| --- |
| SELECT make, mileage, name, city  FROM Car, Owner  WHERE Owner.OwnerID = Car.OwnerID  AND OwnerID = "0004"; |

Fill in the expected results table (the first row will be field names):

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

d. [2 marks]

|  |
| --- |
| UPDATE Owner  SET City = "Dundee"  WHERE ownerID = "0002"; |

Fill in the expected new record (the first row will be field names):

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

e. [2 marks]

|  |
| --- |
| DELETE Car  WHERE regPlate = "YU21UPM" |

Fill in the expected deleted record (the first row will be field names):

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

f. [2 marks]

|  |
| --- |
| INSERT INTO Owner  VALUES ("0005", "Pippa", "Glasgow", "Builder") |

Fill in the expected updated record (the first row will be field names):

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

##### Predict - 18 marks – TBC

1 a. [3 mark]

|  |  |  |
| --- | --- | --- |
| Make | Model | Mileage |
| Tesla | Model 3 | 60,554 |
| Tesla | Model S | 80,000 |

b. [4 marks]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RegPlate | Make | Model | Mileaage | ServicePlan |
| PR43QQS | Tesla | Model 3 | 0003 | 60,554 |
| FD66HFD | Tesla | Model S | 0004 | 80,000 |
| SP77JLP | BMW | 4 Series | 0001 | 40,000 |

c. [3 marks]

|  |  |  |  |
| --- | --- | --- | --- |
| Make | Mileage | Name | City |
| Hyundia | Veloster | Matt | Aberdeen |
| Tesla | Model S | Matt | Aberdeen |

d. [2 marks]

|  |  |  |  |
| --- | --- | --- | --- |
| OwnerID | Name | City | Job |
| 0002 | Hannah | Perth | Teacher |

e. [2 marks]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RegPlate | Make | Model | OwnerID | Mileage | ServicePrice |
| YU21UPM | Hyundia | Veloster | 0004 | 42,145 | 75.00 |

f. [2 marks]

|  |  |  |  |
| --- | --- | --- | --- |
| OwnerID | Name | City | Job |
| 0005 | Pippa | Glasgow | Builder |

# Run - 10 marks

2. Open up the Music Database that has been given to you, notice how there are 5 queries already written.

The SQL for those queries are written in the table below for each of the queries please write down what you think you will happen (e.g. I think I will see a table of all the songs by artist Ariana Grande).

Once you’ve written your expected results you can run the queries in Access and write in the Actual results column. *[10 marks]*

|  |  |  |
| --- | --- | --- |
| SQL | What do you expect tohappen? | Actual results match up? (if no then please explain why) |
| SELECT Title, Artist, Album  FROM Music  WHERE artist = "George Ezra"; |  |  |
| SELECT Title, Artist, NoOfPlays  FROM Music  WHERE Artist = "Ariana Grande" OR Artist = "Lady Gaga"  ORDER BY Artist ASC, NoOfPlays DESC; |  |  |
| SELECT Music.Title, Music.Artist, Playlist.PlaylistName, Playlist.NoOfSongs  FROM Playlist, Music  WHERE Playlist.PlaylistID = Music.PlaylistID AND Playlist.PlaylistName = "Travel" |  |  |
| UPDATE Music  SET PlaylistID = 3  WHERE SongID= 3; |  |  |
| DELETE FROM Playlist  WHERE PlaylistID = 1; |  |  |
| INSERT INTO Playlist (PlaylistID, PlaylistName, NoOfSongs, Private)  VALUES (4, "Cooking", 10, No) |  |  |

##### Run - 10 marks – TBC

|  |  |  |
| --- | --- | --- |
| SQL | What do you expect tohappen? | Actual results match up? (if no then please explain why) |
| SELECT Title, Artist, Album  FROM Music  WHERE artist = "George Ezra"; | E.g. I expect it to show all the song titles, artist name and album name for the songs by George Ezra | Yes / Any reasonable explanation for it not matching up |
| SELECT Title, Artist, NoOfPlays  FROM Music  WHERE Artist = "Ariana Grande" OR Artist = "Lady Gaga"  ORDER BY Artist ASC, NoOfPlays DESC; | E.g. I expect it to show the song titles, artists and number of plays of songs by Ariana Grande or Lady Gaga. I expect to see Ariana Grande songs first and then songs that have been played the most at the top. | Yes / Any reasonable explanation for it not matching up |
| SELECT Music.Title, Music.Artist, Playlist.PlaylistName, Playlist.NoOfSongs  FROM Playlist, Music  WHERE Playlist.PlaylistID = Music.PlaylistID AND Playlist.PlaylistName = "Travel" | I expect to see the title, artist, playlist name and number of songs for the songs on the Travel playlist | Yes / Any reasonable explanation for it not matching up |
| UPDATE Music  SET PlaylistID = 3  WHERE SongID= 3; | I expect the Playlist ID to be updated to 3 | Yes / Any reasonable explanation for it not matching up |
| DELETE FROM Playlist  WHERE PlaylistID = 1; | I expected the playlist record with the ID of 1 to be deleted from the database | Yes / Any reasonable explanation for it not matching up |
| INSERT INTO Playlist (PlaylistID, PlaylistName, NoOfSongs, Private)  VALUES (4, "Cooking", 10, No) | I expected the following information to be added as a new record into the playlist table = playlistID 4, PlaylistName Cooking, NoOfSongs 10 and Private set to no. | Yes / Any reasonable explanation for it not matching up |

# Investigate - 5 marks

When using the ORDER BY what does ASC and DESC meaning in SQL?

*[1 mark]*

|  |
| --- |
|  |

Staff Table

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Age | City |
| A001 | Terry | 24 | Aberdeen |
| A002 | Sabrina | 19 | Perth |
| A003 | Terry | 30 | Glasgow |

Terry (A001) has moved from Aberdeen to Stirling. Evaluate the effect of running the following update statement and explain any changes you could make.

UPDATE Staff

SET City= “Stirling”

WHERE Name = “Terry” *[2 marks]*

|  |
| --- |
|  |

In the SQL statement below, the SELECT statement uses the field “Staff.Name”. Explain why this was used in the query rather than just “Name”.

SELECT Staff.Name, Address, Phone Number

FROM Staff, Animal

WHERE Staff.Employee Number = Animal.EmployeeNumber AND Type=”Rabbit”

*[2 marks]*

|  |
| --- |
|  |

##### Investigate - 5 marks – TBC

*[1 mark]*

|  |
| --- |
| Ascending and Descending |

*[2 marks]*

|  |
| --- |
| There are two records where name = Terry so it will alter both records - they should use the ID to identify specific records. |

*[2 marks]*

|  |
| --- |
| This is to identify which field is being referred to as there might be a field with the same name in the other table. |

# Modify - 18 marks

Open the Music Database in Access and modify the SQL queries in the database so they create the following results:

Query 1 - Modify the SQL so it shows all the songs by Jonas Blue

(paste your SQL below) *[3 marks]*

|  |
| --- |
|  |

Query 2 - Modify the SQL to show music by artists Post Malone or Mark Ronson with the title descending and the number of plays ascending.

(paste your SQL below) *[6 marks]*

|  |
| --- |
|  |

Query 3 - Modify the query to show the songs on the playlist Gym, with the songs in ascending order by song title.

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

Query 4 - Modify the query to delete the **song** with the ID “12” from the music table.

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

Query 5 - Modify the query to update the song “Shotgun” by George Ezra to be included on the playlist “Gym” (hint: check the playlist table for the primary key/foreign key to use in the music table)

(paste your SQL below) *[3 marks]*

|  |
| --- |
|  |

Query 6 - Modify the query to add a new song into the music table with the following values (23, “Sweet But Psycho”, “Ava Max”, 3, 100) into the relevant fields.

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

##### Modify - 18 marks – TBC

*[3 marks]*

|  |
| --- |
| SELECT Title, Artist, Album  FROM Music  WHERE artist = "Jonas Blue"; |

*[6 marks]*

|  |
| --- |
| SELECT Title, Artist, NoOfPlays  FROM Music  WHERE Artist = "Post Malone" OR Artist = "Mark Ronson"  ORDER BY Title DESC, NoOfPlays ASC; |

*[2 marks]*

|  |
| --- |
| SELECT Music.Title, Music.Artist, Playlist.PlaylistName, Playlist.NoOfSongs  FROM Playlist, Music  WHERE Playlist.PlaylistID = Music.PlaylistID AND Playlist.PlaylistName = "Gym"  ORDER BY Title ASC; |

*[2 marks]*

|  |
| --- |
| DELETE \*  FROM Music  WHERE SongID = 12; |

*[3 marks]*

|  |
| --- |
| UPDATE Music  SET PlaylistID = 1  WHERE SongID= 21; |

*[2 marks]*

|  |
| --- |
| INSERT INTO Music ( PlaylistID, Title, Artist, PlaylistID, NoOfPlays)  VALUES (23, “Sweet But Psycho”, “Ava Max”, 3, 100); |

# Make - 12 marks

Open the Company Database in Access and make new SQL queries in the database so they create the following results:

Make a query to display all the First name and last name of the members of staff who work in the United States.

(paste your SQL below) *[1 mark]*

|  |
| --- |
|  |

Make a query to display the first name and last name of all the people who work in Norway and in DepartmentID 4. The last name should be sorted in descending order.

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

Make a query to display the last name and location from the staff table and using the DepartmentID display the department name and the number of staff working in that department where the location is Mexico.

(paste your SQL below) *[3 marks]*

|  |
| --- |
|  |

Make a query that will update Lacey Peck’s (staffID 18) last name to her married name of Matthews.

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

Make a query that will delete the training department as it is merging with the HR department. (Hint: remember to use the ID)

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

Make a query that will insert a new member of staff with the following values (22, Janet, Smith, Norway, 5).

(paste your SQL below) *[2 marks]*

|  |
| --- |
|  |

##### Make - 12 marks – TBC

*[1 mark]*

|  |
| --- |
| SELECT FirstName, LastName  FROM Staff  WHERE Location = “United States”; |

*[2 marks]*

|  |
| --- |
| SELECT FirstName, LastName  FROM Staff  WHERE Location = “Norway” AND DepartmentID = 4  ORDER BY LastName DESC; |

*[3 marks]*

|  |
| --- |
| SELECT Staff.LastName, Staff.Location, Department.Name, Department.NoOfStaff  FROM Staff, Department  WHERE Staff.DepartmentID = Department.DepartmentID AND Location = “Mexico”; |

*[2 marks]*

|  |
| --- |
| UPDATE Staff  SET LastName = “Matthews”  WHERE StaffID = 18; |

*[2 marks]*

|  |
| --- |
| DELETE FROM Department  WHERE DepartmentID = 5; |

*[2 marks]*

|  |
| --- |
| INSERT INTO Staff (StaffID, FirstName, LastName, Location, DepartmentID)  VALUES (22, “Janet”, “Smith”, “Norway”, 5); |