Database Report

Introduction

This report explains the design, purpose, and functionality of two database systems that I created. Each database was built to illustrate an understanding of table structure, data types, relationships, and common database functioning, such as adding, storing, and retrieving records.

Real Madrid Football database

Purpose

The Football Player Database was designed to store information about Real Madrid players. The database allows users to view current players in the team and add new Real Madrid players to the system. This helps maintain an organised and up to date record of the team.

Structure

This database contains a single table called Players. It uses a simple, flat structure because no relationships are needed.

Table: Players

- PlayerID – Integer, Primary Key
- Name – Text
- Age – Integer
- Position – Text
- Nationality – Text

These fields were chosen because they provide the essential information needed to identify and describe each Real Madrid player.

Functionality

Add a Player

The user can enter details for a new Real Madrid player. The system then inserts the data into the Players table. This ensures the database can grow as new players join the team.

View All Players

Users can view all existing players stored in the database. This retrieves all records from the Players table and displays the full Real Madrid squad.

Data Accuracy

The program uses basic validation so fields must contain correct data types (e.g., age must be a number). This helps keep the database reliable and consistent.

Customer Orders Relational Database

Purpose

The purpose of the Customer–Orders Database is to store customer details and link each customer with their orders. This database demonstrates understanding of relational design, where data is stored in more than one table and connected through a relationship.

Structure

This database contains two tables: Customers and Orders.

Table 1: Customers

- Customer ID – Integer, Primary Key
- First Name – Text
- Last name – Text

Table 2: Orders

- Order ID – Integer, Primary Key
- Customer ID – Integer, Foreign Key
- Order Date – Text/Date
- Total Amount – Real

The CustomerID field in the Orders table is a Foreign Key that links each order to the customer who made it.

Relationship

The database uses a one-to-many relationship:

- One customer can have many orders
- Each order belongs to one customer

This structure prevents data duplication and allows the system to organise information more efficiently.

Functionality

The Customer Orders database allows users to:

Add Customers

The user can enter customer details, which are saved in the Customers table.

Add Orders

Users can create new orders linked to a specific customer using the CustomerID.

View all customers

Displays:

- CustomerID
- First name
- Last name

View All Orders

Displays:

- OrderID
- CustomerID
- Order Date
- Order Amount

If no orders exist, the system shows a message to the user.

Data Integrity

The use of a foreign key ensures:

- Orders cannot exist without a valid customer
- Customer information is not repeated unnecessarily

Conclusion

Both databases were successfully designed and implemented. The Football Database provides a simple but useful way to store and manage Real Madrid player information. The Customer–Orders Database demonstrates a deeper understanding of relational design by connecting customers to their orders using a one-to-many relationship.

Together, these projects show the ability to organise data, design functional database structures, and create programs that allow users to interact with stored information effectively.