Database Design & Development Project.

All data contained within this Database (DB) is fictitious with the author not having thorough knowledge of the inner workings of a veterinary clinic, purely for the purpose of showing a potential database which may be deployed in a commercial database management system.

The DB has is composed of nine different tables for management of the data (a more detailed explanation given in the ***RelationalSchema.pdf***):

1. Animal
2. Appointment
3. Billing
4. Food
5. Medication
6. Payments
7. Pet Owner
8. Staff
9. Treatment

Two of tables are extra to the tables outlined in the project requirements. Payments was to allow multiple payments to made against one bill as included in the narrative. The treatment table was included as Animals may be treated in the clinic multiple times. A single appointment may contain multiple treatments to multiple animals/aliments or conditions.

An extra table that could be included would be a pricing table outlining the different costs of treatments, including consultations, check-ups, animal chipping etc. this would be a highly variable topic and without supporting normal pricing would be hard to portray accurately. The Staff table could be expanded with wages, contact information etc. These felt out of the scope of this project based on the narrative given as the focus appears more the treatments, customers and billing but given more time would be implemented also.

Many tables included a comments column as given the nature of a veterinary clinic significant notable details may apply to a treatment for example but would not conform to the data expected in other columns but should also be recorded.

Constructing the Database.

The narrative supplied in the project requirements was studied, additional tables were included, and any information supplied was linked to different tables to ascertain the columns they would contain. Additional columns and data were added to make the DB more whole.

Using XAMPP and SQL each table was created with column name, constraints, primary keys, and foreign keys (where relevant). Once the structure was created, the fictitious data populating the DB was entered into an Excel file, then saved as comma separated values (CSV) file and imported into each table individually. This was to preserve the data in case of any errors occurring or corruption of the DB. An export file was also routinely created as a back-up, alternatively the DB table structure could be recreated, and the data quickly imported again conforming to the same or changed structure.

The Entity Relation Diagrams (ERD) were only created once the DB was completed and through multiple SQL queries was confirmed to be operating as expected so that no irregularities were represented.

Implementing the Database.

This project was developed using XAMPP, the ***EXPORT.sql*** file should be imported into XAMPP or a similar environment of your choosing. ***Queries.sql*** contains a variation of queries that can be run against the DB. There are Select, Insert, Update and Delete queries separated into sections with supporting comments.

NB: The final Delete queries remove test data inserted in the Queries script. These are intended to showcase multiple deletes with single statements and return the DB near to its original imported state.

Testdata.sql is intended to showcase the test data included within ***EXPORT.sql***. There is no need to run these Insert statements as well. Can be used as a reference for the original values if for example the Updates made in ***Queries.sql*** wished to be reversed.

Supporting Files in Submission.

RelationalSchema.pdf:

This contains an overview of the DB, its tables, relationships, keys, and constraints. Two distinctive styles of Entity Relationship Diagrams (ERD) have been used here to best relay this information. The two PNG files included ***ERD.png*** and ***chenERD.png*** are from this document, if better clarity is required.

CoddsRules.pdf:

Each of Codd's Rules separated with a narrative and SQL commands where appropriate in reference to this DB. All SQL commands can be run within the DB.