**Assignment Fulfilment**

Kevin Del Rosario

Diploma in Information Technology, Kaplan Higher Education Institute

IT46 IDC: Introduction to Database Design and Development

Ms. Vaithilingam Malarkodi

December 8, 2023

**Part 1**

**ER Diagram for**

**Blue Garage Private Limited: Car Services**

A diagram of a service

Description automatically generated

**Assumptions**

1. *Contact Number on Client Entity Type*
   1. Contact Numbers is assumed for communication and coordination purposes regarding schedules, billing, and other related matters.
2. *Car ID on Car Entity Type*
   1. Car ID is assumed to uniquely identify every car present in the data base.
3. *Plate Number on Car Entity Type*
   1. Every car has a unique plate number. This can be useful when tracking maintenance services. This can be used as a primary key, but it will be a weak one as plate numbers can change due to some scenarios. Therefore, a Car ID was assumed.
4. *Brand on Car Entity Type*
   1. Car Type is assumed because it is essential in identifying necessary parts needed when replacing and/or maintaining vehicle parts.
5. *Model Year on Car Entity Type*
   1. This is also assumed for the same reason why brand is needed in a database – maintenance/replacement.
6. *Hourly Rate on Freelance Mechanic Entity Type*
   1. This is assumed because as a freelancer, most of them will charge based on their hourly rate. This is also good for the company because they will be able to choose a mechanic based on their rate.
7. *Hours Worked on Freelance Mechanic Entity Type*
   1. Since Hourly Rate is assumed, we also must assume the total hours they have worked to finalize and compute their salary.

**Part 2**

**Figure 1. Client Tax Invoice**

**A close-up of a receipt

Description automatically generated**

**UNF – 1NF – 2NF – 3NF Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unnormalized Form** | **First Normalized Form** | **Second Normalized Form** | **Third Normalized Form** |
| **Client Invoice** | **Client Invoice** | **Client Invoice** | **Client Invoice** |
| TaxInvoiceNumber  TransactionDate  ClientName ClientAddress  ClientNumber Gender CreditLimit PetName Treatment Price | TaxInvoiceNumber TransactionDate ClientName ClientAddress ClientNumber Gender CreditLimit | TaxInvoiceNumber TransactionDate ClientName ClientAddress ClientNumber Gender CreditLimit | TaxInvoiceNumber TransactionDate ClientName |
| **Client** |
| ClientName ClientAddress ClientNumber Gender CreditLimit |
| **Pet Treatment** | **Pet Treatment** | **Pet Treatment** |
| TaxInvoiceNumber Pet Name Treatment Price | TaxInvoiceNumber Treatment  PetName | TaxInvoiceNumber Treatment  PetName |
| **Treatment** | **Treatment** |
| Treatment Price | Treatment Price |

**2. Identify the repeating group of attributes and transform part (i) into tables that are in 1st Normal form. (Explain the rationale for your decision).**

First Normal Form addresses the issue of atomicity, ensuring that each column in the dataset holds only indivisible and single values. In the transition from UNF to 1NF, the original flat structure is divided into two tables: "Client Invoice" and "Pet Treatment." Each table represents a distinct entity, and the data is organized to eliminate repeating groups. This promotes better data integrity and lays the groundwork for further normalization.

**3. Identify any partial dependencies in part (ii) and transform into tables that are in 2nd Normal Form. (Explain the rationale for your decision).**

Second Normal Form focuses on eliminating partial dependencies within a dataset. In the context of the "Client Invoice" and "Pet Treatment" tables, 2NF ensures that each non-prime attribute is fully functionally dependent on the primary key. The modification involves rearranging and refining the data structure to better represent the relationships between different attributes. In this stage, the goal is to reduce redundancy and ensure that the dataset is more logically organized.

**4. Identify any transitive dependencies in part (iii) and transform into tables that are in 3rd Normal Form. (Explain the rationale for your decision).**

Third Normal Form aims to remove transitive dependencies, ensuring that no non-prime attribute is transitively dependent on the primary key. In the transition to 3NF, the "Client Invoice" table is further split into "Clients" and "Client Invoice" tables. This separation addresses transitive dependencies related to client-specific information and results in a more refined and normalized structure. The "Pet Treatment" table remains unchanged, maintaining its 2NF status. This level of normalization enhances data integrity, minimizes redundancy, and creates a more scalable and maintainable database schema.

**Part 3**

1. **Entity Relationship Diagram for Pet Store**

**A diagram of a client

Description automatically generated**

1. **Using the CREATE TABLE in any dialect of SQL, create the corresponding SQL tables. Indicate the primary key, foreign keys, column names, constraints etc.**

create database idd\_assignment;

use idd\_assignment;

CREATE TABLE Clients (

ClientName VARCHAR(255) PRIMARY KEY,

ClientNumber VARCHAR(10),

ClientAddress VARCHAR(255),

Gender VARCHAR(10),

CreditLimit INT

);

create table clientInvoice (

TaxInvoiceNumber Int Primary Key,

ClientName varchar(255),

TransactionDate Date,

Foreign Key (ClientName) references Clients(ClientName)

);

create table Treatment (

Treatment varchar(255) primary key,

Price Int

);

create table PetTreatment (

TaxInvoiceNumber int,

Treatment varchar(255),

PetName varchar(255),

Primary Key (TaxInvoiceNumber, Treatment),

Foreign Key (TaxInvoiceNumber) references clientinvoice(TaxInvoiceNumber),

Foreign Key (Treatment) references treatment(treatment)

);

Show tables;  
  
A screenshot of a computer

Description automatically generated

1. **Insert (at least) 10 sample rows for each identified table.**

**CLIENTS TABLE**

INSERT INTO Clients (ClientName, ClientAddress, ClientNumber, Gender, CreditLimit)

VALUES

('John Doe', '123 Main St', '123-4567', 'Male', 5000),

('Jane Smith', '456 Oak St', '987-6543', 'Female', 7000),

('Bob Johnson', '789 Pine St', '555-1234', 'Male', 6000),

('Alice Williams', '321 Maple St', '111-2223', 'Female', 8000),

('Sam Brown', '555 Elm St', '999-8887', 'Male', 5500),

('Emily Davis', '888 Birch St', '444-3332', 'Female', 6500),

('Tom Wilson', '777 Cedar St', '666-7778', 'Male', 7500),

('Lisa Turner', '222 Spruce St', '333-4445', 'Female', 9000),

('Mark Miller', '444 Fir St', '777-9991', 'Male', 4800),

('Sara Jackson', '666 Redwood St', '222-5556', 'Female', 6200);

Select \* from clients;

A screenshot of a computer

Description automatically generated

**CLIENTS INVOICE TABLE**

INSERT INTO ClientInvoice (TaxInvoiceNumber, TransactionDate, ClientName)

VALUES

(1, '2023-01-01', 'John Doe'),

(2, '2023-01-02', 'Jane Smith'),

(3, '2023-01-03', 'Bob Johnson'),

(4, '2023-01-04', 'Alice Williams'),

(5, '2023-01-05', 'Sam Brown'),

(6, '2023-01-06', 'Emily Davis'),

(7, '2023-01-07', 'Tom Wilson'),

(8, '2023-01-08', 'Lisa Turner'),

(9, '2023-01-09', 'Mark Miller'),

(10, '2023-01-10', 'Sara Jackson');

Select \* from clientsinvoice;

A screenshot of a computer

Description automatically generated

**TREATMENT TABLE**

INSERT INTO Treatment (Treatment, Price)

VALUES

('Vaccination', 50.00),

('Dental Cleaning', 75.00),

('Flea Treatment', 30.00),

('Spaying', 150.00),

('Neutering', 120.00),

('X-ray', 80.00),

('Surgery', 200.00),

('Check-up', 40.00),

('Grooming', 60.00),

('Deworming', 25.00);

A screenshot of a computer

Description automatically generated

**PET TREATMENT TABLE**

INSERT INTO PetTreatment (TaxInvoiceNumber, Treatment, PetName)

VALUES

(1, 'Vaccination', 'Buddy'),

(2, 'Dental Cleaning', 'Max'),

(3, 'Flea Treatment', 'Lucy'),

(4, 'Spaying', 'Charlie'),

(5, 'Neutering', 'Molly'),

(6, 'X-ray', 'Leo'),

(7, 'Surgery', 'Luna'),

(8, 'Check-up', 'Rocky'),

(9, 'Grooming', 'Coco'),

(10, 'Deworming', 'Milo');

Select \* from pettreatment  
A screenshot of a computer

Description automatically generated

1. **List the names and phone numbers of all clients who have a credit limit > $1,000 in**

**alphabetical order**

select clientName, clientNumber, creditlimit

from clients

where creditlimit > 1000

order by clientname ASC;  
  
A screenshot of a computer

Description automatically generated

1. **Find the total price (before tax) for all the treatments for a particular client.**

select clients.clientName, sum(treatment.price) as "Total Price Before Tax"

FROM clients

JOIN clientInvoice ON clients.clientname = clientinvoice.clientname

JOIN pettreatment ON clientinvoice.taxinvoicenumber = pettreatment.taxinvoicenumber

JOIN treatment ON pettreatment.treatment = treatment.treatment

WHERE clients.clientName = "John Doe"

GROUP BY clients.clientName;  
  
A screenshot of a computer

Description automatically generated

1. **List the cheapest treatment price for a particular client.**

select clients.clientName, min(treatment.price) as "Cheapest Treatment Price"

FROM clients

JOIN clientInvoice ON clients.clientname = clientinvoice.clientname

JOIN pettreatment ON clientinvoice.taxinvoicenumber = pettreatment.taxinvoicenumber

JOIN treatment ON pettreatment.treatment = treatment.treatment

WHERE clients.clientName = "Alice Williams"

GROUP BY clients.clientName;  
A screenshot of a computer

Description automatically generated

1. **List the clients and their pet(s) sorted by client’s gender**

select clients.clientName, clients.gender, pettreatment.petname

FROM clients

JOIN clientInvoice ON clients.clientname = clientinvoice.clientname

JOIN pettreatment ON clientinvoice.taxinvoicenumber = pettreatment.taxinvoicenumber

ORDER BY gender;

A screenshot of a computer

Description automatically generated