F28DM Database Management System

Coursework 1: Database Design and Implementation

Group 14

Members:

Leelian Alhadhoud
Avoor Minha Fathima
Eman Iftikhar
Kanana Muchiri
Nur Shahira Sharizan
Sibila Shihab

Lecturers:

Professor Talal Sheikh
Professor Md Azher Uddin

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM (Databases Management Systems)
Type of assessment:	**Group**
Coursework Title:	CW1 – Database Design and Implementation
Student Name:	Leelian Waleed Alhadhoud
Student ID Number:	H00377549

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Leelian Waleed Alhadhoud*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM – Database Management Systems
Type of assessment:	**Group**
Coursework Title:	CW1 – Database Design and Implementation
Student Name:	Avoor Minha Fathima
Student ID Number:	H00376380

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Avoor Minha Fathima*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM Database Management Systems
Type of assessment:	**Group**
Coursework Title:	CW1 – Database Design and Implementation
Student Name:	Sibila Shihab
Student ID Number:	H00413308

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Sibila*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM Database Management
Type of assessment:	**Group**
Coursework Title:	Coursework 1: Database Design and Implementation
Student Name:	Kanana Gathoni Muchiri
Student ID Number:	H00375728

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Kanana Muchiri*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM-Database Management System
Type of assessment:	**Group**
Coursework Title:	CW1 Database Design and Implementation
Student Name:	Nur Shahira Sharizan
Student ID Number:	H00379248

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Nur Shahira Sharizan*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F28DM-Database Management System
Type of assessment:	**Group**
Coursework Title:	CW1 Database Design and Implementation
Student Name:	Eman Iftikhar
Student ID Number:	H00362343

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Eman Iftikhar*

Date: *16/02/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

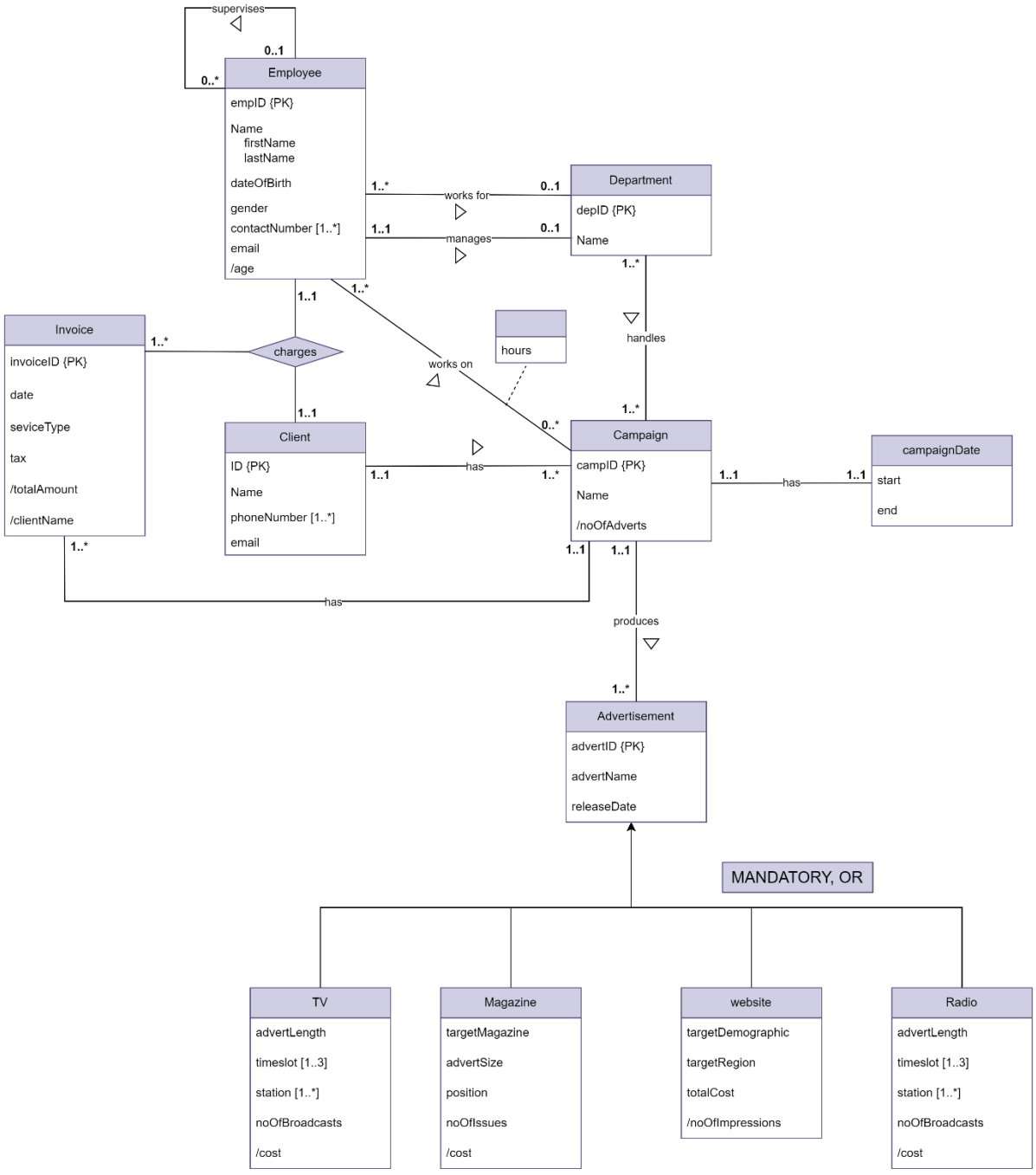
Member's Contribution

1. **Avoor Minha Fathima:** I contributed to developing the ER diagram and converting the ER to its respective relational schema and data dictionary. I helped in writing the MariaDB implementation for few tables, indexing and reviewing the final report.
2. **Nur Shahira Sharizan:** I contributed to developing the ER diagram, writing additional comments for ER, and commenting on the MariaDB.
3. **Leelian Waleed Alhadoud:** I contributed to developing the ER diagram and converting the ER to its respective relational schema. I also did the commenting for the MariaDB implementation
4. **Eman Iftikhar:** I contributed to developing the ER diagram and converting the ER to its respective relational schema. I also added descriptions for relational schema and ER diagram. I also contributed to indexing and the formatting of the report
5. **Sibila Shihab:** I contributed to the implementation of the majority of the MariaDB code. I also did the reviewing and editing of the data dictionary and relational schema
6. **Kanana Muchiri:** I contributed to the development of the ER diagram with its corresponding description and the relational schema and data dictionary that follow. Additionally, I contributed to the indexing.

Table of Contents

1 Entity Relationship Modelling

1.1 ER Diagram



1.2 ER Diagram Description

In the ER diagram for generalization/specialization, the entity Mandatory Or was chosen because each advertisement can be of only one type at a time. A campaign, however, can have many advertisements which is represented by their relationship.

The entities employee, client and invoice are in a ternary relationship because an employee can charge many clients with many invoices.

An employee has a recursive relationship because an employee can supervise many employees and each employee will have a supervisor.

Campaign has a weak entity named date referring to the defined start and end dates of a campaign.

The ER diagram includes derived attributes including age, noOfAdverts, cost, noOfImpressions and total cost.

Relationship Description:

- An employees can work for zero or one department
- A department can have multiple employees
- An employee manages zero or one department
- A department can be managed by a single employee
- An employees can work on multiple campaigns
- A campaign can have multiple employees
- A department handles multiple campaigns
- A campaign can be handled by only one department
- A campaign produces multiple advertisements
- Each advertisement belongs to a single campaign
- Each campaign has a defined start and end date
- An employee can charge multiple invoices to a client
- A client has multiple campaigns
- A campaign belongs to a single client
- A campaign can have many invoices
- Each invoice belongs to only to one campaign
- An employee can supervise many employees
- Each employee can only have one supervisor

2 Relational Model

2.1 Relational Schema

Employee (empID: integer(6), firstName: string(30), lastName: string(20), dateOfBirth: date, gender: [M|F|N], email: string(200) , *empDepID: integer(6), supEmpID: integer(6)*)

Primary Key: empId

Foreign Key: *empDepID, supEmpID*

Department (depID: integer, depName: string(20), *managerEmpID: integer(6)*)

Primary Key: depId

Foreign Key: *managerEmpID*

Campaign (campID: integer(6), campName: string(30)*, campClientID: integer(6)*)

Primary Key: campId

Foreign Key: *campClientID*

CampaignDate (dcampID: integer(6), dStart: date, dEnd: date)

Primary Key: dcampID, dStart, dEnd

Composite Key: dStart, dEnd

Note: CampaignDate is a weak entity that is supported by it's parent entity Campaign. A defined start and end date for a campaign will only exist if a campaign exists.

Client (clientID: integer(6), clientName: string(30), cEmail: string(200))

Primary Key: clientID

Invoice (invoiceID: integer(6), invoiceDate: date, serviceType: string(50), tax: integer(1..100), *iCampID:* integer(6))

Primary Key: invoiceID

TV (advertID: integer(6), advertName: string(20), releaseDate: date, advertLength: time, noOfBroadcast: integer*, advertCampID: integer(6)*)

Primary Key: advertID

Foreign Key: *advertCampI*

Magazine (advertID: integer(6), advertName: string(20), releaseDate: date, targetMagazine: string(20), advertSize: integer, position: string(15), noOfIssues: integer, *advertCampID: integer(6)*)

Primary Key: advertID

Foreign Key: *advertCampID*

Website (advertID: integer(6), advertName: string(20), releaseDate: date, targetDemographic: string(20), targetRegion: string(20), totalCost: integer, *advertCampID: integer(6)*)

Primary Key: advertID

Foreign Key: *advertCampID*

Radio (advertID: integer(6), advertName: string(20), releaseDate: date, advertLength: time, noOfBroadcast: integer, *advertCampID: integer(6)*)

Primary Key: advertID

Foreign Key: *advertCampID*

Handles (hCampID: integer(6), hDepID: integer)

Primary Key: hCampID, hDepID

Composite Key: hCampID, hDepID

Note: Handles is an additional table for many to many relationship between employee and campaign. Handles captures the information consisting of the specefic department handling a campaign

WorksOn (wEmpID: integer(6), wCampID: integer(6), hours: integer)

Primary Key: wEmpID, wCampID

Composite Key: wEmpID, wCampID

Note: WorksOn is an additional table for many to many relationship between employee and campaign. WorksOn captures the time in hours for an employee working on a campaign

Charges (cEmpID: integer(6), cInvoiceID: integer(6), cClientID: integer(6))

Primary Key: cEmpID, cInvoiceID, cClientID

Composite Key: cEmpID, cInvoiceID, cClientID

Note: Charges is an additional table for many to many relationship between employee and campaign. WorksOn captures the ternary relationsip between employee, invoice and client

TvTimeslot (tAdvertID: integer(6), tTimeslot: string(20))

Primary Key: tAdvertID, tTimeslot

Note: TvTimeslot is an additional table representing multivalued attribute from the table TV

RadioTimeslot (rAdvertID: integer(6), rTimeslot: string(20))

Primary Key: rAdvertID, rTimeslot

Note: RadioTimeslot is an additional table representing multivalued attribute from the table Radio

TvStation (tAdvertID: integer(6), station: string(20))

Primary Key: tAdvertID, station

Note: TvStation is an additional table representing multivalued attribute from the table TV

RadioStation (rAdvertID: integer(6), station: string(20))

Primary Key: rAdvertID, station

Note: RadioStation is an additional table representing multivalued attribute from the table Radio

EmpContactNo (empID: integer(6), contactNo: integer(10))

Primary Key: empID, contactNo

Note: EmpContactNo is an additional table representing the multivalued attribute from the table Employee. The contactNo is set to the limit of 10 based on the format (05XXXXXXXX)

ClientContactNo (clientID: integer(6), contactNo: integer(10))

Primary Key: clientID, contactNo

Note: ClientContactNo is an additional table representing the multivalued attribute from the table Client. The contactNo is set to the limit of 10 based on the format (05XXXXXXXX)

2.2 Data Dictionary

EMPLOYEE RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
empID	Employment Identification number	6 Digit	N	Y	
firstName	First names	Text(30)	N	N	
lastName	Last name	Text(20)	N	N	
dateOfBirth	Date of birth	Date	N	N	
email	Email of the employee	Text(200)	N	N	
gender	Whether the employee is male or female	Char[M|F|N]	N	N	
empDepID	Identification number of the department the employee works at	Integer(1..100)	N	N	Department.depID
supEmpID	Employee Identification number of their supervisor	6 Digit	Y	N	Employee.EmpID

DEPARTMENT RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
depID	Department Identification number	Integer(1..100)	N	Y	
depName	Department name	Text(20)	N	N	
managerEmpID	Employee Identification number of the department manager	6 Digit	N	N	Employee.empID

CAMPAIGN RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
campID	Campaign Identification number	6 Digit	N	Y	
campName	Campaign name	Text(20)	N	N	
campDepID	Identification number of the department the campaign belongs to.	Integer(1..100)	N	N	Department.depID
campClientID	Client Identification Number for the campaign	6 Digit	N	N	Client.clientID

CLIENT RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
clientID	Client Identification number	6 Digit	N	Y	
clientName	Name of Client	Text(30)	N	N	
email	Email address of the client	Text(200)	N	N	

INVOICE RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
invoiceID	Invoice Identification number	6 Digit	N	Y	
invoiceDate	Issue date of invoice	Date	N	N	
serviceType	Types of services done	Integer	N	N	
tax	Percentage of tax charged	Integer(1..100)	N	N	
icampID	Campaign Identification Number associated with the invoice	6 Digit	N	N	Campaign.campID

CAMPAIGNDATE RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
dCampID	Campaign Identification number associated with the dates	6 Digit	N	Y	Campaign.campID
dStart	Start date of the campaign	Date	N	N	
dEnd	End date of the campaign	Date	N	N	

TV RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
advertID	Advertisement Identification number	6 Digit	N	Y	
advertName	Advertisement name	Text(20)	N	N	
releaseDate	Advertisement Release date	Date	N	N	
advertLength	Length (duration) of the advertisement	Time	N	N	
noOfBroadcasts	Number of the broadcasts	Integer	N	N	
advertCampID	Identification number of the campaign the advertisement belongs to	6 Digit	N	N	Campaign.campID

MAGAZINE RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
advertID	Advertisement Identification number	6 Digit	N	Y	
advertName	Advertisement name	Text(20)	N	N	
releaseDate	Advertisement Release date	Date	N	N	
targetMagazine	Magazines targeted	Text(20)	N	N	
advertSize	The size of the advertisement in the magazine (percentage of page it occupies)	Integer(1..100)	N	N	
position	The position the advertisement is placed in the magazine	Text(20)	N	N	
noOfIssues	The number of issues of the magazine the advertisement appears in	Integer(1..100)	N	N	
advertCampID	Identification number of the campaign the advertisement belongs to	6 Digit	N	N	Campaign.campID

WEBSITE RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
advertID	Advertisement Identification number	6 Digit	N	Y	
advertName	Advertisement name	Text(20)	N	N	
releaseDate	Advertisement Release date	Date	N	N	
targetDemographic	A particular sector of an audience (demographic) targeted	Text(20)	Y	N	
targetRegion	A particular area (region) targeted	Text(20)	Y	N	
totalCost	Total cost specified by the client	Integer	N	N	
advertCampID	Campaign Identification number of the campaign the advertisement belongs to	6 Digit	N	N	Campaign.campID

RADIO RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
advertID	Advertisement Identification number	6 Digit	N	Y	
advertName	Advertisement name	Text(20)	N	N	
releaseDate	Advertisement Release date	Date	N	N	
advertLength	Length (duration) of the advertisement	Time	N	N	
noOfBroadcasts	Number of the broadcasts	Integer	N	N	
advertCampID	Identification number of the campaign the advertisement belongs to	6 Digit	N	N	Campaign.campID

HANDLES RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
hCampID	Identification number of the campaign handled by the department	6 Digit	N	Y	Campaign.campID
hDepID	Identification number of the department handling the campaign	6 Digit	N	Y	Department.depID

WORKSON RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
wEmpID	Identification number of the employee working on the campaign	6 Digit	N	Y	Employee.empID
wCampID	Identification number of the Campaign being worked on	6 Digit	N	Y	Campaign.campID
hours	The duration spent working on the campaign	Integer	N	N	

CHARGES RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
cEmpID	Employment Identification number of the employee that charges the employee	6 Digit	N	Y	Employee.empID
cInvoiceID	Invoice Identification number of the invoice being charged to the client	6 Digit	N	Y	Invoice.invoiceID
cClientID	Client Identification number of the client being charged	6 Digit	N	Y	Client.clientID

TVTIMESLOT RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
tAdvertID	Advertisement Identification number	6 Digit	N	Y	TV.advertID
timeslot	Time at which the advert is broadcasted	Text(20)	N	Y	

RADIOTIMESLOT RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
rAdvertID	Advertisement Identification number	6 Digit	N	Y	Radio.advertID
timeslot	Time at which the advert is broadcasted	Text(20)	N	Y	

TVSTATION RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
tAdvertID	Advertisement Identification number	6 Digit	N	Y	TV.advertID
station	Name of the station	Text(20)	N	Y	

RADIOSTATION RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
rAdvertID	Advertisement Identification number	6 Digit	N	Y	Radio.advertID
station	Name of the station	Text(20)	N	Y	

empContactNo RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
empID	Employement Identification number	6 Digit	N	Y	Employee.empID
contactNo	Contact number of the employee	10 Digit	N	Y	

clientContactNo RELATION					
Attributes	Description	Domain	Null?	Primary Key	Foreign Key
clientID	client Identification number	6 Digit	N	Y	Client.clientID
contactNo	Contact number of the client	10 Digit	N	Y	

3 Implementation of Schema

3.1 MariaDB Implementation

NOTE: UNCOMMENTED MARIADB FILE IS ATTACHED, PLEASE REFER TO THAT FILE FOR RUNNING THE CODE

```
CREATE TABLE Department (

/*a table was created with the name Department that has the following attributes */

        depID INT PRIMARY KEY NOT NULL,

/*an attribute for depID in the table (Department) is created as a primary key where the record can not be empty */

        depName VARCHAR(20) UNIQUE NOT NULL,

/* an attribute for depName in the table (Department) is created as unique to ensure that all the values are not the same. The advertName value can not be empty and can hold up to 20 characters */

        managerEmpID INT(6) NOT NULL

/* an attribute managerEmpID in the table (Department) is created where the record can not be empty and can hold up to 6 integers */

) ENGINE=InnoDB;


CREATE TABLE Employee (

/* a table was created with the name Employee that has the following attributes */

        empID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,

/* an attribute for empID in the table (Employee) is created as a primary key where the record can not be empty and can hold up to 6 integers. With the auto-increment it will generate a new empID that will be inserted into the table */

        firstName VARCHAR(30) NOT NULL,

/* an attribute for firstName in the table (Employee) is created where the record can not be empty and can hold up to 30 characters */

        lastName VARCHAR (20) NOT NULL,

/* an attribute for lastName in the table (Employee) is created where the record can not be empty and can hold up to 20 characters */

        dateOfBirth DATE NOT NULL,

/* an attribute for dateOfBirth in the table (Employee) is created where record can not be empty and the data type is Date */
```

```
        email VARCHAR(200) NOT NULL,
        empDepID INT NOT NULL,
/* an attribute for empDepID in the table (Employee) is created where the record can not be
empty */
        FOREIGN KEY (empDepID) REFERENCES Department(depID)
/* empDepID is a foreign key that references the Department table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (empDepID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        supEmpID INT(6),
/* an attribute for supEmpID in the table (Employee) is created and can hold up to 6
integers */
        FOREIGN KEY (supEmpID) REFERENCES Employee(empID)
/* supEmpID is a foreign key that references the Employee table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (supEmpID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        gender ENUM('F' ,'M' ,'N') NOT NULL
/* an attribute for gender in the table (Employee) is created and the entry can only be F,
M or N and the record can not be empty */
) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table employee */
ALTER TABLE Employee AUTO_INCREMENT=100000;


ALTER TABLE Department
/* to add, delete or modify an existing table (Department) */
ADD FOREIGN KEY(managerEmpID) REFERENCES Employee(empID)
/* adding a new foreign key managerEmpID that references the Employee table */
 ON UPDATE CASCADE ON DELETE CASCADE;
/* when the foreign key (managerEmpID) is updated the value will automatically reflect the
changes on the parent table */
```

```
CREATE TABLE Client (
/* a table was created with the name Client that has the following attributes */
        clientID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,
/* an attribute for clientID in the table (Client) is created as a primary key where the
record can not be empty and can hold up to 6 integers */
        clientName VARCHAR(30),
/* an attribute for clientName in the table (Client) is created where it can hold up to 30
characters */
        email VARCHAR(200) NOT NULL
) ENGINE=InnoDB;

/* setting the auto increment start value to 100000 for table client*/
ALTER TABLE Client AUTO_INCREMENT=100000;

CREATE TABLE Campaign (
/* a table was created with the name Campaign that has the following attributes */
        campID INT(6) PRIMARY KEY NOT NULL  AUTO_INCREMENT,
/* an attribute for campID in the table (Campaign) is created as a primary key where the
record can not be empty and can hold up to 6 integers. With the auto-increment it will
generate a new campID that will be inserted into the table */
        campName VARCHAR(20) NOT NULL,
/* an attribute for campName in the table (Campaign) is created where the record can not be
empty and can hold up to 20 characters */
        campClientID INT(6) NOT NULL,
/* an attribute for campClientID in the table (Campaign) is created where the record can
not be empty and can hold up to 6 integers */
        FOREIGN KEY (campClientID) REFERENCES Client(clientID)
/* campClientID is a foreign key that references the Client Table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (campClientID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        campDepID INT NOT NULL,
```

```
/* an attribute for campDepID in the table (Campign) is created where the record can not be
empty and the data type is an integer */


        FOREIGN KEY (campDepID) REFERENCES Department(depID)

/* campDepID is a foreign key that references the Department table */

        ON UPDATE CASCADE ON DELETE CASCADE

/* when the foreign key (campDepID) is deleted or updated the value will automatically
reflect the changes on the parent table */

) ENGINE=InnoDB;


 /* setting the auto increment start value to 100000 for table campaign */

ALTER TABLE Campaign AUTO_INCREMENT=100000;


CREATE TABLE Invoice (

/* a table was created with the name Invoice that has the following attribute */

        invoiceID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,

/* an attribute for invoiceID in the table (Invoice) is created as a primary key where the
record can not be empty and can hold up to 6 integers. With the auto-increment it will
generate a new invoiceID that will be inserted into the table */

        invoiceDate DATE NOT NULL,

/* an attribute for invoiceDate in the table (Invoice) is created where record can not be
empty and the data type is date */

        serviceType VARCHAR(50),

/* an attribute for serviceType in the table (Invoice) is created and it can hold up to 50
characters */

        tax INT NOT NULL,

/* an attribute for tax in the table (Invoice) is created where the record can not be empty
and the data type is an integer */

        iCampID INT(6) NOT NULL,

/* an attribute for iCampID in the table (Invoice) is created where the record can not be
empty and can hold up to 6 integers */

        FOREIGN KEY (iCampID) REFERENCES Campaign (campID)

/* iCampID is a foreign key that references the Campaign Table */

        ON UPDATE CASCADE ON DELETE CASCADE

/* when the foreign key (iCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */
```

```
) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table invoice */

ALTER TABLE Invoice AUTO_INCREMENT=100000;


CREATE TABLE CampaignDate (

/* a table was created with the name CampaignDate that has the following attributes */

        dCampID INT(6) PRIMARY KEY NOT NULL,

/* an attribute for dCampID in the table (CampaignDate) is created as a primary key where
the record can not be empty */

        FOREIGN KEY(dcampID) REFERENCES Campaign(campID)

/* dCampID is a foreign key that references the Campaign table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (dCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */

        dStart DATE NOT NULL,

/* an attribute for dStart in the table (CampaignDate) is created where the record can not
be empty and the data type is date */

        dEnd DATE NOT NULL

/* an attribute for dEnd in the table (CampaignDate) is created where the record can not be
empty and the data type is date */

) ENGINE=InnoDB;


CREATE TABLE TV (

/* a table was created with the name TV that has the following attribute */

        advertID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,

/* an attribute for advertID in the table (TV) is created as a primary key where the record
can not be empty and can hold up to 6 integers. With the auto-increment it will generate a
new advertID that will be inserted into the table */

        advertName VARCHAR(20) UNIQUE NOT NULL,

/* an attribute for advertName in the table (TV) is created as unique to ensure that all
the values are unique. The advertName value can not be empty and can hold up to 20
characters */

        releaseDate DATE NOT NULL,

/* an attribute for releaseDate in the table (TV) is created where the record can not be
empty and the data type is Date */
```

```
        advertLength TIME NOT NULL,
/* an attribute for advertLength in the table (TV) is created where the record can not be
empty and the data type is Time */
        noOfBrodcasts INT NOT NULL,
/* an attribute for noOfBrodcasts in the table (TV) is created where the record can not be
empty and the data type is an integer */
        advertCampID INT(6) NOT NULL,
/* an attribute for advertCampID in the table (TV) is created where the record can not be
empty and can hold up to 6 integers */
        FOREIGN KEY (advertCampID) REFERENCES Campaign(CampID)
/* advertCampID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE
/* when the foreign key (advertCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */
) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table TV */
ALTER TABLE TV AUTO_INCREMENT=100000;


CREATE TABLE Magazine (
/*a table was created with the name Magazine that has the following attribute */
        advertID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,
/* an attribute for advertID in the table (Magazine) is created as a primary key where the
record can not be empty and can hold up to 6 integers. With the auto-increment it will
generate a new advertID that will be inserted into the table */
        advertName VARCHAR(20) UNIQUE NOT NULL,
/* an attribute for advertName in the table (Magazine) is created as unique to ensure that
all the values are unique. The advertName value can not be empty and can hold up to 20
characters */
        releaseDate DATE NOT NULL,
/* an attribute for releaseDate in the table (Magazine) is created where the record can not
be empty and the data type is Date */
        targetMagazine VARCHAR(20) NOT NULL,
```

```
/* an attribute for targetMagazine in the table (Magazine) is created where the record can
not be empty and it can hold up to 20 characters */


        position VARCHAR(20) NOT NULL,

/* an attribute for position in the table (Magazine) is created where the record can not be
empty and it can hold up to 20 characters */

        advertSize INT NOT NULL DEFAULT 50,

/* an attribute for advertSize in the table (Magazine) is created where the record can not
be empty and the data type is an integers. If there are no given values, it will give a
default of 50 */

        noOfIssues INT NOT NULL,

/* an attribute for noOfIssues in the table (Magazine) is created where the record can not
be empty and the data type is an integers */

        advertCampID INT(6) NOT NULL,

/* an attribute for adverCampID in the table (Magazine) is created where the record can not
be empty and can hold up to 6 integers */

        FOREIGN KEY (advertCampID) REFERENCES Campaign(CampID)

/* advertCampID is a foreign key that references the Campaign table */

        ON UPDATE CASCADE ON DELETE CASCADE

/* when the foreign key (advertCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */

) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table magazine */

ALTER TABLE Magazine AUTO_INCREMENT=100000;


CREATE TABLE Website (

/* a table was created with the name Website that has the following attribute */

        advertID INT(6) PRIMARY KEY NOT NULL AUTO_INCREMENT,

/* an attribute for advertID in the table (Website) is created as a primary key where the
record can not be empty and can hold up to 6 integers. With the auto-increment it will
generate a new advertID that will be inserted into the table */

        advertName VARCHAR(20) UNIQUE NOT NULL,

/* an attribute for advertName in the table (Website) is created as unique to ensure that
all the values are unique. The advertName value can not be empty and can hold up to 20
characters */

        releaseDate DATE NOT NULL,
```

```
/* an attribute for releaseDate in the table (Website) is created where the record can not
be empty and the data type is Date */


        targetDemographic VARCHAR(20) ,

/* an attribute for targetDemographic in the table (Website) is created and can hold up to
20 characters */

        targetRegion VARCHAR(20),

/* an attribute for targetRegion in the table (Website) is created and can hold up to 20
characters */

        totalCost INT NOT NULL,

/* an attribute for totalCost in the table (Website) is created where the record can not be
empty and the data type is an integers */

        advertCampID INT(6) NOT NULL,

/* an attribute for advertCampID in the table (Website) is created where the record can not
be empty and can hold up to 6 integers */

        FOREIGN KEY (advertCampID) REFERENCES Campaign(campID)

/* advertCampID is a foreign key that references the Campaign table */

        ON UPDATE CASCADE ON DELETE CASCADE

/* when the foreign key (advertCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */

) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table website */

 ALTER TABLE Website AUTO_INCREMENT=100000;


CREATE TABLE Radio (

/* a table was created with the name Radio that has the following attribute */

        advertID INT(6) PRIMARY KEY AUTO_INCREMENT,

/* an attribute for advertID in the table (Radio) is created as a primary key where the
record can not be empty and can hold up to 6 integers. With the auto-increment it will
generate a new advertID that will be inserted into the table */

        advertName VARCHAR(20) UNIQUE NOT NULL,

/* an attribute for advertName in the table (Website) is created as unique to ensure that
all the values are unique. The advertName value can not be empty and can hold up to 20
characters */
```

```
releaseDate DATE NOT NULL,
/* an attribute for releaseDate in the table (Radio) is created where the record can not be
empty and the data type is Date  */

advertLength TIME NOT NULL DEFAULT '00:00:30',
/* an attribute for advertLength in the table (Radio) is created where the record can not
be empty and the data type is Time with a default of 30 secounds */

noOfBrodcasts INT NOT NULL DEFAULT 10,
/* an attribute for noOfBrodcasts in the table (Radio) is created where the record can not
be empty and the data type is int with a default of 1010 numbers */

advertCampID INT(6) NOT NULL,
/* an attribute for advertCampID in the table (Radio) is created where the record can not
be empty and can hold up to 6 integers */
/* an attribute for adverCampID in the table (Radio) is created where the record can not be
empty and can hold up to 6 numbers */

FOREIGN KEY (advertCampID) REFERENCES Campaign(CampID)
/* advertCampID is a foreign key that references the Campaign table */

ON UPDATE CASCADE ON DELETE CASCADE
/* when the foreign key (advertCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */

) ENGINE=InnoDB;


/* setting the auto increment start value to 100000 for table radio*/
ALTER TABLE Radio AUTO_INCREMENT=100000;


CREATE TABLE Handles (
/* a table was created with the name Handles that has the following attribute */

hCampID INT(6) NOT NULL,
/* an attribute for hCampID in the table (Handles) is created where the record can not be
empty and can hold up to 6 integers */

FOREIGN KEY (hCampID) REFERENCES Campaign (campID)
/* hCampID is a foreign key that references the Campaign table */

ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (hCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */
```

```
        hDepID INT NOT NULL,
/* an attribute for hDepID in the table (Handles) is created where the record can not be
empty and the data type is an integer */
        FOREIGN KEY (hDepID) REFERENCES Department (depID)
/* hDepID is a foreign key that references the Department table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (hDepID) is deleted or updated the value will automatically reflect
the changes on the parent table */
        PRIMARY KEY(hCampID, hDepID)
/* the two attributes (hCampID) and (hDepID) collectively form the primary key since they
are non-volatile values */
) ENGINE=InnoDB;


CREATE TABLE WorksOn (
/* a table was created with the name WorksOn that has the following attribute */
        wEmpID INT(6) NOT NULL,
/* an attribute for wCampID in the table (WorksOn) is created where the record can not be
empty and can hold up to 6 numbers */
        FOREIGN KEY (wEmpID) REFERENCES Employee(empID)
/* wEmpID is a foreign key that references the Employee table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (wEmpID) is deleted or updated the value will automatically reflect
the changes on the parent table */
        wCampID INT(6) NOT NULL,
/* an attribute for wCampID in the table (WorksOn) is created where the record can not be
empty and can hold up to 6 integers */
        FOREIGN KEY (wCampID) REFERENCES Campaign(campID)
/* wCampID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (wCampID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        hours INT NOT NULL,
```

```
/* an attribute for hours in the table (WorksOn) is created where the record can not be
empty and the data type is time */

        PRIMARY KEY(wEmpID, wCampID)

/* the two attributes (wEmpID) and (hCampID) collectively form the primary key since they
are none volatile values */

) ENGINE=InnoDB;


CREATE TABLE Charges(

/* a table was created with the name Charges that has the following attribute  */

        cInvoiceID INT(6) NOT NULL,

/* an attribute for cInvoice in the table (Charges) is created where the record can not be
empty and can hold up to 6 integers */

        FOREIGN KEY(cInvoiceID) REFERENCES Invoice(invoiceID)

/* cInoviceID is a foreign key that references the Invoice table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (cInvoiceID) is deleted or updated the value will automatically
reflect the changes on the parent table */

        cEmpID INT(6) NOT NULL,

/* an attribute for cEmpID in the table (Charges) is created where the record can not be
empty and can hold up to 6 integers */

        FOREIGN KEY(cEmpID) REFERENCES Employee(empID)

/* cEmpID is a foreign key that references the Employee table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (cEmpID) is deleted or updated the value will automatically reflect
the changes on the parent table */

        cClientID INT(6) NOT NULL,

/* an attribute for cClientID in the table (Charges) is created where the record can not be
empty and can hold up to 6 integers */

        FOREIGN KEY(cClientID) REFERENCES Client(clientID)

/* cClientID is a foreign key that references the Client table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (cClientID) is deleted or updated the value will automatically
reflect the changes on the parent table */

        PRIMARY KEY (cInvoiceID, cEmpID, cClientID)

/* the three attributes (cInvoice), (cEmpID), and (cClientID) collectively form the primary
key since they are none volatile values */
```

```
) ENGINE=InnoDB;


CREATE TABLE TvTimeslot(
/* a table was created with the name TvTimeslot that has the following attribute  */
        tAdvertID INT(6) NOT NULL,
/* an attribute for tAdvertID in the table (TvTimeslot) is created where the record can not
be empty and can hold up to 6 integers */
        FOREIGN KEY(tAdvertID) REFERENCES TV(advertID)
/* tAdvertID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (tAdvertID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        timeslot VARCHAR(20) NOT NULL
/* an attribute for timeslot in the table (TvTimeslot) is created where the record can not
be empty and can hold up to 20 characters  */
) ENGINE=InnoDB;


CREATE TABLE RadioTimeslot(
/* a table was created with the name RadioTimeslot that has the following attribute  */
        rAdvertID INT(6) NOT NULL,
/* an attribute for rAdvertID in the table (RadioTimeslot) is created where the record can
not be empty and can hold up to 6 integers */
        FOREIGN KEY(rAdvertID) REFERENCES radio(advertID)
/* rAdvertID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (rAdvertID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        timeslot VARCHAR(20) NOT NULL
/* an attribute for timeslot in the table (RadioTimeslot) is created where the record can
not be empty and can hold up to 20 characters  */
) ENGINE=InnoDB;
```

```
CREATE TABLE TvStation(
/* a table was created with the name TvStation that has the following attribute  */
        tAdvertID INT(6) NOT NULL,
/* an attribute for tAdvertID in the table (TvStation) is created where the record can not
be empty and can hold up to 6 integers */
        FOREIGN KEY(tAdvertID) REFERENCES TV(advertID)
/* tAdvertID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (tAdvertID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        station VARCHAR(20) NOT NULL
/* an attribute for station in the table (TvStation) is created where the record can not be
empty and can hold up to 20 characters  */
) ENGINE=InnoDB;

CREATE TABLE RadioStation(
/* a table was created with the name RadioStation that has the following attribute */
        rAdvertID INT(6) NOT NULL,
/* an attribute for rAdvertID in the table (RadioStation) is created where the record can
not be empty and can hold up to 6 numbers */
        FOREIGN KEY(rAdvertID) REFERENCES Radio(advertID)
/* rAdvertID is a foreign key that references the Campaign table */
        ON UPDATE CASCADE ON DELETE CASCADE,
/* when the foreign key (rAdvertID) is deleted or updated the value will automatically
reflect the changes on the parent table */
        station VARCHAR(20) NOT NULL
/* an attribute for station in the table (RadioStation) is created where the record can not
be empty and can hold up to 20 characters */
)ENGINE=InnoDB;

CREATE TABLE EmpContactNo(
/* a table was created with the name EmpContactNo that has the following attribute  */
        empID INT(6) NOT NULL,
```

```
/* an attribute for empID in the table (EmpContactNo) is created where the record can not
be empty and can hold up to 6 numbers */

        FOREIGN KEY(empID) REFERENCES Employee(empID)

/* empID is a foreign key that references the Campaign table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (empID) is deleted or updated the value will automatically reflect
the changes on the parent table */

        contactNo INT(10) NOT NULL,

/* an attribute for contactNo in the table (EmpContactNo) is created where the record can
not be empty and can hold up to 10 numbers */

        PRIMARY KEY(empID, contactNo)

/* the two attributes (empID) and (contactNo) collectively form the primary key since they
are none volatile values */

        ) ENGINE=InnoDB;


CREATE TABLE clientContactNo(

/* a table was created with the name clientContactNo that has the following attribute  */

        clientID INT(6) NOT NULL,

/* an attribute for clientID in the table (clientContactNo) is created where the record can
not be empty and can hold up to 6 integers */

        FOREIGN KEY(clientID) REFERENCES Client(clientID)

/* clientID is a foreign key that references the Campaign table */

        ON UPDATE CASCADE ON DELETE CASCADE,

/* when the foreign key (clientID) is deleted or updated the value will automatically
reflect the changes on the parent table */

        contactNo INT(10) NOT NULL,

/* an attribute for contactNo in the table (clientContactNo) is created where the record
can not be empty and can hold up to 10 integers */

        PRIMARY KEY(clientID, contactNo)

/* the two attributes (clientID) and (contactNo) collectively form the primary key since
they are none volatile values */

        ) ENGINE=InnoDB;


SET foreign_key_checks = 0;

INSERT INTO `department` VALUES ('1', 'Sales Department', '895496');
```

SET foreign_key_checks = 1;

3.2 Additional Notes

When inserting values into the first table 'Department' there is a foreign key constraint due to the referencing of Employee table in it. To overcome this, we set foreign_key_checks to 0 which enables us to insert values into the 'Department'.

4 Indexing

For Employee:

CREATE INDEX indx_empname ON Campaign(firstName,lastName);

Explanation: An employee is most likely to be referenced by their name. Therefore, indexing the names of the employees would optimize the retrieval of the employee details.

For Department:

CREATE UNIQUE INDEX indx_depname ON Department(depName);

Explanation: A department is most likely to be referenced by their name. Therefore, indexing the names of the departments would optimize the retrieval of the department details. Moreover, the unique keyword will ensure that there are no duplicate values for the department name.

For Client:

CREATE INDEX indx_clientname ON Client(clientName);

Explanation: A client is most likely to be referenced by their name. Therefore, indexing the names of the clients would optimize the retrieval of the client details.

For Campaign:

CREATE UNIQUE INDEX indx_campname ON Campaign(campName);

Explanation: A campaign is most likely to be referenced by their name. Therefore, indexing the names of the campaign would optimize the retrieval of the campaign details. Moreover, the unique keyword will ensure that there are no duplicate values for the campaign name.

For Invoice:

CREATE INDEX indx_icampname ON Invoice(iCampID);

Explanation: When referring to an invoice, it is likely that the campaign the invoice is a part of is searched. Therefore, indexing the campaign ID in the invoice table would optimize the retrieval of the campaign details associated with the particular invoice.