Higher Diploma in Business Systems Analysis

Fitness Studio Database Design

WIT No: 20081437 December 12th 2017

Damien Hogan

Table of Contents

Introduction

The client owns a fitness studio which provides Classes in Pilates, Cardio Pilates, Yoga, Prenatal Yoga, Barre and intends to also incorporate Kids Classes in both Pilates and Yoga.

Ideally the database will include Customers & Instructors details incorporating their name, address, phone number, email address so they can be contacted easily.

I will set up this database using the **CRUD** method which will allow the administrator (client) to **Create** data by adding new information, **Read** by being able to view any data in the database, **Update** by have the knowledge and permissions to adding or amending information in the database and finally **Delete** by having the ability to delete information that has become obsolete,

Purpose

This document is intended to act as guide to develop a new database for the client so that she can access all information in one place. It is hoped that the customer and the other users will gain an understanding of how the database is setup and how they can easily update the database.

It is expected that through consultation that the purpose and structure will be understood by the system programmer.

Present

At the moment all contact details for customers are input to Microsoft Excel worksheet and all are encouraged to like or follow the client on Facebook. When a new class term starts four to five times per year the client launches a Facebook campaign to raise awareness. The client also pays for a bulk mail service so send out notifications of upcoming classes and workshops.

The client also manually sends out mobile phone texts intermittently to those customers who are not on the database, due to time constraints and a lack of a formal administration and data collection system.

When the client wants to contact customers enmasse she has to spend many hours of looking up contact details and trying to manually update her informal database (excel file). She ends up with duplication customer contact details and sometimes different details for the same customer as people may move house, change email address or change mobile phone providers.

I decided to present this assignment as a set of logical steps so that it is easily read and understood by the client and the database programmer.

Project Scope

- It is proposed to develop a relational database that will record and then retrieve information relating to customers, instructors, class types, class prices, class calendar etc.
- It is proposed to develop a specific query to assist the Client retrieve information and contact details regarding instructors.
- It is proposed to develop a specific query to assist Client retrieve information and contact and personal details regarding customers so that a proper assessment of customer can be done.
- The contact details will be used to contact customers via email and or text so that the client can send promotional material and class timetables.
- It will also aid the client in finding out who has paid and who is due to pay. So that reminders can be sent to those owing. If reminders result in the issuing of invoices, this is deemed to be outside the scope of this database design (an email can be sent as a reminder).
- The client would like to be able to calculate earnings at the end of given time period.
- In phase two (in the future) the customer would like to link to her website to the database so that a customer can view class timetables and be able to see availability and and be able to book a class and pay in advance (at the moment this is beyond the scope of the project).
- Moving to this system can acquire advantages such as saving time, man hours and space wastage. This will increase the efficiency of class availability.

First Consultation Information review

I had access to the clients excel files and was able to see what information she had on file already and it gave me a good idea of what kind of information was needed to complete the database. The client had lists of phone numbers which I was able to add to the database and the client went to the customers to fill in missing data. It was decided that only customers with a complete set of data was to be input to the new database.
After initial consultation with the client and finding out what she would like to achieve. I made a list of possible areas to make queries and made suggestions to the client after consulted her about what would be desirable and what would be possible to be able to benefit her.

Second Consultation Ideas for System

Following on from second consultation with the Client and deciding what's achievable.
The following has been agreed and signed off by the client.

List of tables to be created for Imelda Studio Database:

Studioinstructor	List of Instructors with contact details and class speciality.
StudioCustomer	List of Customers with contact details and personal information.
studiobooking	List of bookings with time and date.
Studioclasstype	List of class types in the studio.
StudioClassEquipment	Type of equipment needed for each class type.
StudioClassPayment	The cost of the class and who has paid or not.
StudioRoom	Which studio the class is carried out in.

Main database features:

Customer account status
Edit/Delete/View Member details
Unpaid/advanced/new Payments
Customer history
Instructor history
Create/Edit/Delete/View class type details

Constraints

Studio has a name, a phone, a website and an address.
Instructors have a name, an address, phone, a contact email, and Specialty
Customers have a name, an address, phone, a contact email, and Specialty
Customers can be taught by the client and by instructor. The client is also an instructor.
A class has a date, a time, a fee and a class type.
Every class takes place at exactly one studio, and every class has exactly one teacher
A studio will have many classes. An instructor can teach many classes.

Entities and Attributes

The following list of Entities and Attributes has been agreed and signed off by the client.

Booking Queries---

Who has booked a Class
What Type of Class have they booked
What Date have they Booked
What Time Have they Booked
What Instructor is Assigned to that Class

Booking ID	Customer ID	Date	Time	Class Type ID	Payment ID	Instructor ID	Room ID

Primary Key is Booking ID and this is also the candidate Key due to it being unique.
Foreign Key(s) will be Customer ID and Class Type ID.

Equipment Queries---

Which Class Type equipment is needed for

Equipment ID	Class Type

Primary Key is Equipment ID and this is also the candidate Key due to it being unique.

Instructor Queries---

Name of Instructor
Address of Instructor
Phone Number of Instructor
Email of Instructor
Which Class Type do they teach

Instructor ID	Name	Address	Phone	Email	Class Type ID

Primary Key is Instructor ID and this is also the candidate Key due to it being unique.
Foreign Key(s) will be Class Type ID.

Payment Queries---

Who has Paid for Classes
What Date is the Class
What Time is the Class
What Class have they Paid for
What is the Cost of the class

Payment ID	Customer ID	Date	Time	Class ID	Cost of Class

Primary Key is Payment ID and this is also the candidate Key due to it being unique.
Foreign Key(s) will be Customer ID.

Customer Queries---

Name of Customer
Address of Customer
Phone Number of Customer
Email of Customer
What is their Booking ID
What Class(es) have they booked

Customer ID	Name	Address	Phone	Email	Booking ID	Class ID

Primary Key is Customer ID and this is also the candidate Key due to it being unique.

Studio Queries---

Which Class(es) is/are assigned

Studio ID	Class Assigned

Primary Key is Studio ID and this is also the candidate Key due to it being unique.
Foreign Key(s) will be

Class Type Queries---

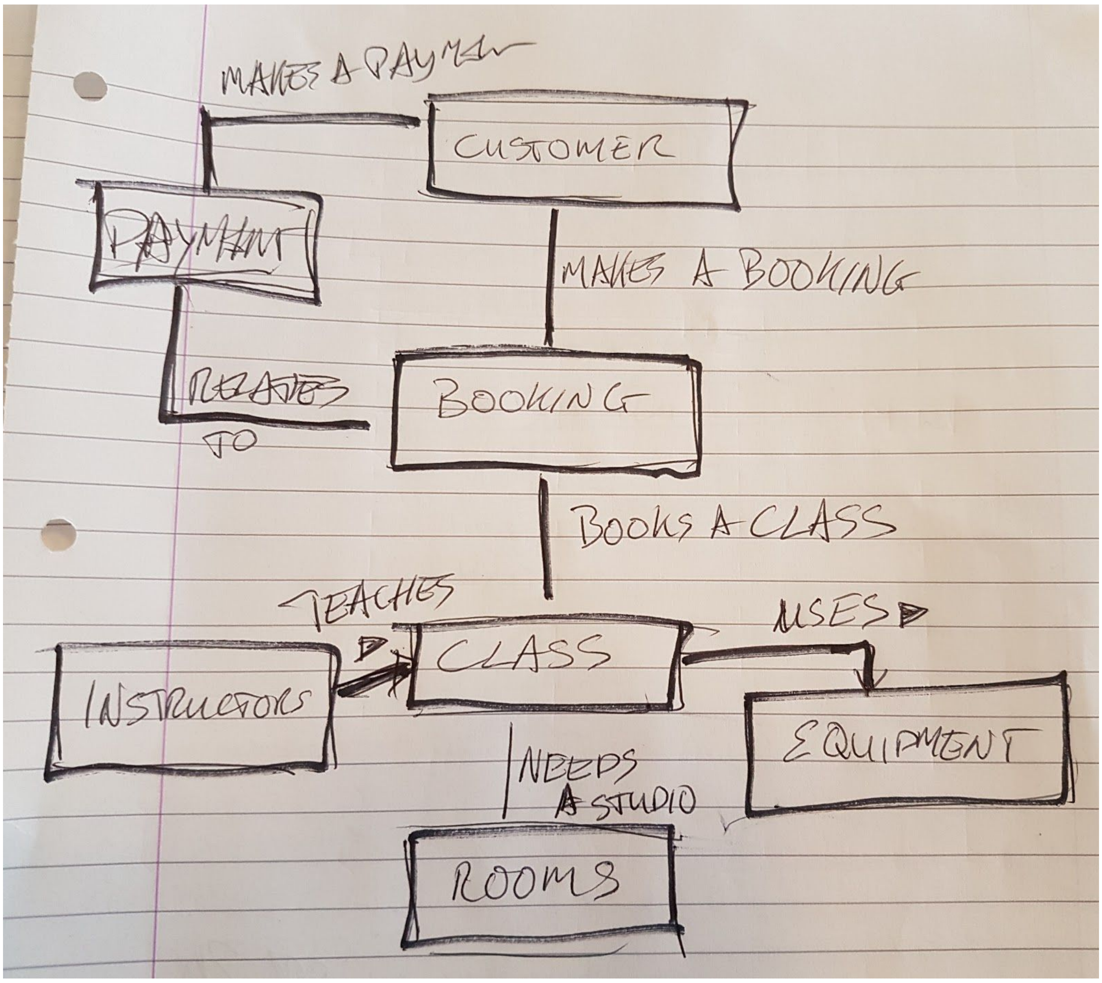
What types of class are available
Which Instructor is teaches a Which Class Type
Which Studio is used
What is the cost of the class
What type of equipment is need for each class

Class ID	Class Type	Instructor ID	StudioID	Class Cost	EquipmentID

Primary Key is Class ID and this is also the candidate Key due to it being unique.
Foreign Key(s) will be Equipment ID.

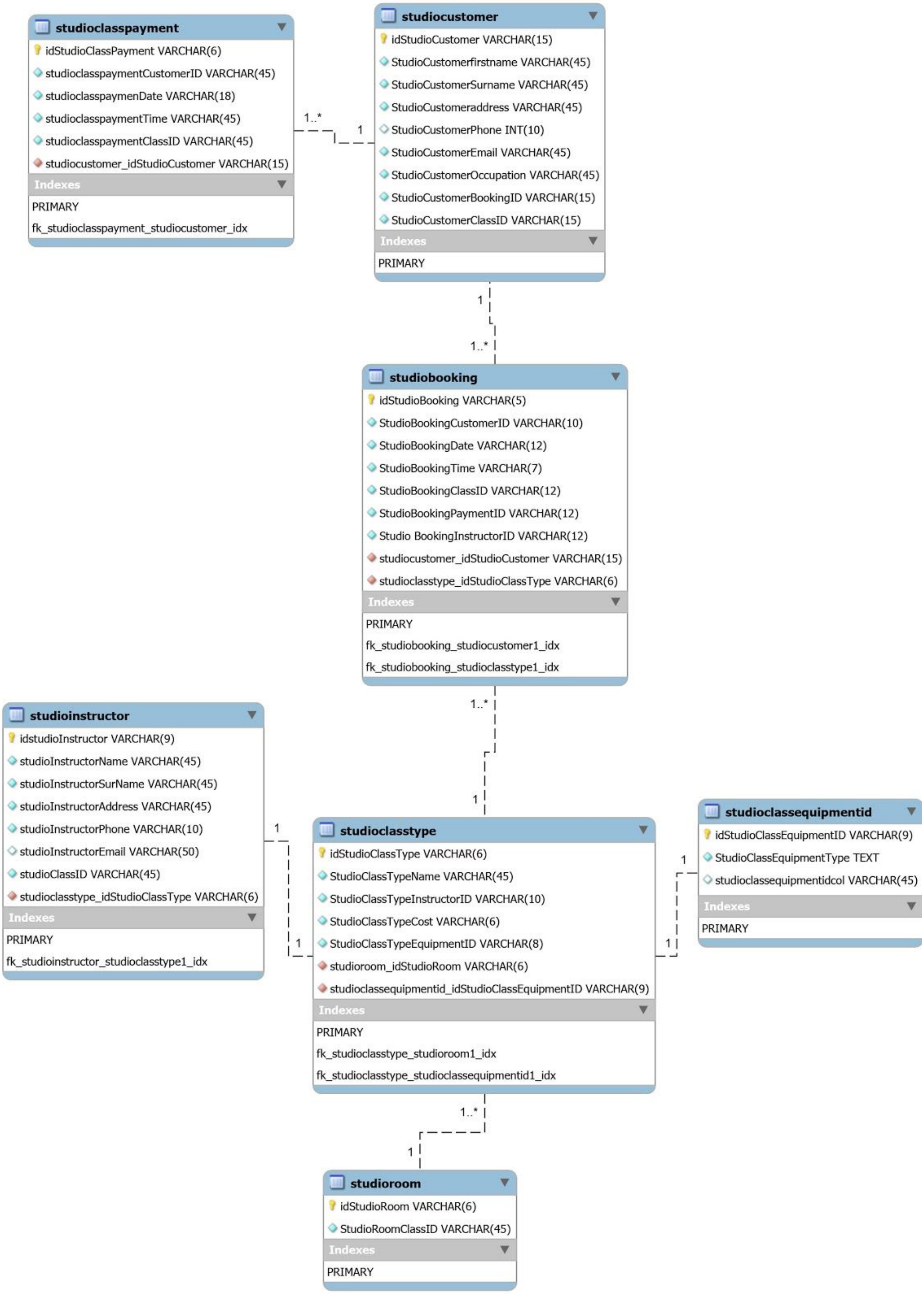
Initial Entity Relationships (ER) Diagram

From an initial appraisal of the data available and using the top down ER modelling technique I produced a first draft ER diagram which was drawn by hand. This gave me a good idea what information and what the relationships should be between entities.



Final Database Design

Entity Relationship Diagram



studioclasspayment
- idStudioClassPayment VARCHAR(6)
- studioclasspaymentCustomerID VARCHAR(45)
- studioclasspaymenDate VARCHAR(18)
- studioclasspaymentTime VARCHAR(45)
- studioclasspaymentClassID VARCHAR(45)
- studiocustomer_idStudioCustomer VARCHAR(15)

Indexes
- PRIMARY
- fk_studioclasspayment_studiocustomer_idx

studiocustomer
- idStudioCustomer VARCHAR(15)
- StudioCustomerfirstname VARCHAR(45)
- StudioCustomerSurname VARCHAR(45)
- StudioCustomeraddress VARCHAR(45)
- StudioCustomerPhone INT(10)
- StudioCustomerEmail VARCHAR(45)
- StudioCustomerOccupation VARCHAR(45)
- StudioCustomerBookingID VARCHAR(15)
- StudioCustomerClassID VARCHAR(15)

Indexes
- PRIMARY

studiobooking
- idStudioBooking VARCHAR(5)
- StudioBookingCustomerID VARCHAR(10)
- StudioBookingDate VARCHAR(12)
- StudioBookingTime VARCHAR(7)
- StudioBookingClassID VARCHAR(12)
- StudioBookingPaymentID VARCHAR(12)
- Studio BookingInstructorID VARCHAR(12)
- studiocustomer_idStudioCustomer VARCHAR(15)
- studioclasstype_idStudioClassType VARCHAR(6)

Indexes
- PRIMARY
- fk_studiobooking_studiocustomer1_idx
- fk_studiobooking_studioclasstype1_idx

studioinstructor
- idstudioInstructor VARCHAR(9)
- studioInstructorName VARCHAR(45)
- studioInstructorSurName VARCHAR(45)
- studioInstructorAddress VARCHAR(45)
- studioInstructorPhone VARCHAR(10)
- studioInstructorEmail VARCHAR(50)
- studioClassID VARCHAR(45)
- studioclasstype_idStudioClassType VARCHAR(6)

Indexes
- PRIMARY
- fk_studioinstructor_studioclasstype1_idx

studioclasstype
- idStudioClassType VARCHAR(6)
- StudioClassTypeName VARCHAR(45)
- StudioClassTypeInstructorID VARCHAR(10)
- StudioClassTypeCost VARCHAR(6)
- StudioClassTypeEquipmentID VARCHAR(8)
- studioroom_idStudioRoom VARCHAR(6)
- studioclassequipmentid_idStudioClassEquipmentID VARCHAR(9)

Indexes
- PRIMARY
- fk_studioclasstype_studioroom1_idx
- fk_studioclasstype_studioclassequipmentid1_idx

studioclassequipmentid
- idStudioClassEquipmentID VARCHAR(9)
- StudioClassEquipmentType TEXT
- studioclassequipmentidcol VARCHAR(45)

Indexes
- PRIMARY

studioroom
- idStudioRoom VARCHAR(6)
- StudioRoomClassID VARCHAR(45)

Indexes
- PRIMARY

Table Design Specification

```
-- -----------------------------------------------------------
-- Table `imeldastudiodb`.`studioequipment`
-- -----------------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioClassEquipmentID	VARCHAR	(9)	NOT NULL	This is a unique code for each piece of equipment – Primary Key
idStudioClassEquipmenttype	VARCHAR	(45)	NOT NULL	This is a description of equipment type

```
-- -----------------------------------------------------------
-- -----------------------------------------------------------
-- Table `imeldastudiodb`.`studiopayment
-- -----------------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioClassPayment	INT		NOT NULL	This is a unique code for each piece of equipment – Primary Key
StudioClassPaymentCost	VARCHAR	(45)	NOT NULL	This is the price of selected class

StudioCustomerID = Foreign Key

```
-- ---------------------------------------------------------
-- Table `imeldastudiodb`.`studioclasstype`
-- ---------------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioClassType	VARCHAR	(9)	NOT NULL	This is a unique code for each Class Type – Primary Key
idStudioClassTypeName	VARCHAR	(45)	NOT NULL	This is a unique name for each Class Type
idStudioClassTypeInstructorID				This is a unique code for each Instructor
StudioClassCost	VARCHAR	(45)	NOT NULL	This is the price of selected class
StudioClassEquipmentID	VARCHAR	(45)	NOT NULL	This is a unique code for each piece of equipment

studioRoomID= Foreign Key

studioClassEquipmentID= Foreign Key

```
-- -----------------------------------------------------
-- Table `imeldastudiodb`.`studioinstructor`
-- -----------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idstudioInstructor	**VARCHAR**	(9)	NOT NULL	**This is a unique code for each customer – Primary Key**
studioInstructorName	**VARCHAR**	**(45)**	**NOT NULL**	This is the first name of customer it is not unique
studioInstructorSurName	**VARCHAR**	**(45)**	**NOT NULL**	This is the Surname of a customer it is not unique
studioInstructorAddress	**VARCHAR**	**(45)**	**NOT NULL**	This is the Address of the instructor it may not be unique, so will be treated as not unique
studioInstructorPhone	**VARCHAR**	**(45)**	**NOT NULL**	This is the instructor phone number
studioInstructorEmail	**VARCHAR**	**(45)**	**NOT NULL**	This is the instructor phone email
studioClassID	**VARCHAR**	**(45)**	**NOT NULL**	This is a unique number assigned to each type of class

StudioClassType = Foreign Key

```
-- -----------------------------------------------------
-- Table `imeldastudiodb`.`studiocustomer`
-- -----------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioCustomer	**VARCHAR**	(15)	NOT NULL	**This is a unique code for each customer- Primary Key**
StudioCustomerfirstname	**VARCHAR**	**(45)**	**NOT NULL**	This is the first name of customer it is not unique
StudioCustomerSurname	**VARCHAR**	**(45)**	**NOT NULL**	This is the Surname of customer it is not unique
StudioCustomeraddress	**VARCHAR**	**(45)**	**NOT NULL**	This is the Address of the customer it may not unique to will be treated as not unique
StudioCustomerPhone	**INT NULL**	**(9)**	**NOT NULL**	This is the customer phone number
StudioCustomerEmail	**VARCHAR**	**(45)**	**NOTNUL**	This is the customer phone email
StudioCustomerOccupation	**VARCHAR**	**(45)**	**NOT NULL**	This is the customer occupation
StudioCustomerBookingID	**VARCHAR**	**(15)**	**NOT NULL**	This is a unique number assigned to a booking
StudioCustomerClassID	**VARCHAR**	**(15)**	**NOT NULL**	This is a unique number assigned to each type of class

```
-- -----------------------------------------------------
-- Table `imeldastudiodb`.`studiobookingclass`
-- -----------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioBooking	VARCHAR	(9)	NOT NULL	This is a unique code for each Class – Primary Key
StudioBookingCustomerID	VARCHAR	(9)	NOT NULL	This is a unique number assigned to each type of class
studioBookingDate	DATE		NOT NULL	This is the date of given classes customer it is not unique
studioBookingTime	VARCHAR	(9)	NOT NULL	This is the time of given classes customer it is not unique
studioBookingPaymentID	VARCHAR	(12)	NOT NULL	This is a unique code for each Payment
studioBookingClassID	VARCHAR	(45)	NOT NULL	This is a unique code for each piece of equipment
studioBookingInstructorID	VARCHAR	(45)	NOT NULL	This is a unique code for each instructor

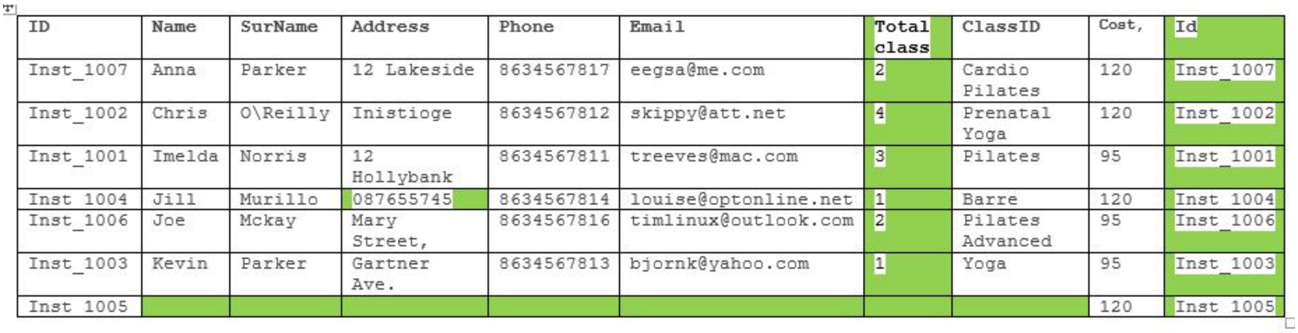
studioBookingCustomerID = Foreign Key
StudioClassTypeID = Foreign Key

```
-- -----------------------------------------------------
-- Table `imeldastudiodb`.`studioroom’
-- -----------------------------------------------------
```

Field	Type	Size	Null/NOT Null	Constraints
idStudioroom	VARCHAR	(9)	NOT NULL	This is a unique code for each Room in the studio – Primary Key
StudioRoomClassID	VARCHAR	(45)	NOT NULL	This is a unique code for each Class Type

Normalisation

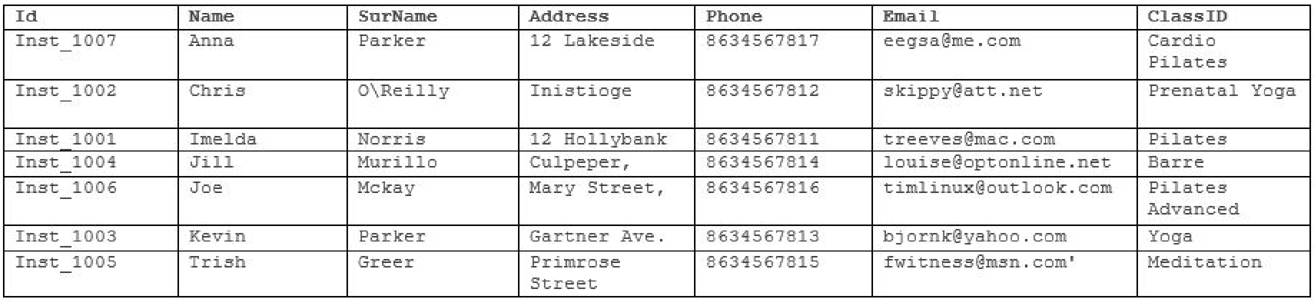
Instructor Table Unnormalised



ID	Name	SurName	Address	Phone	Email	Total class	ClassID	Cost,	Id
Inst_1007	Anna	Parker	12 Lakeside	8634567817	eegsa@me.com	2	Cardio Pilates	120	Inst_1007
Inst_1002	Chris	O\Reilly	Inistioge	8634567812	skippy@att.net	4	Prenatal Yoga	120	Inst_1002
Inst_1001	Imelda	Norris	12 Hollybank	8634567811	treeves@mac.com	3	Pilates	95	Inst_1001
Inst_1004	Jill	Murillo	087655745	8634567814	louise@optonline.net	1	Barre	120	Inst_1004
Inst_1006	Joe	Mckay	Mary Street,	8634567816	timlinux@outlook.com	2	Pilates Advanced	95	Inst_1006
Inst_1003	Kevin	Parker	Gartner Ave.	8634567813	bjornk@yahoo.com	1	Yoga	95	Inst_1003
Inst_1005								120	Inst_1005

Unnormalised version of Instructor table: I initially introduced a column of total classes taught by an instructor but felt after consulting with the client that it was not needed. There was also a duplicate of Instructor ID in the last Column so this has also been removed. In the address column there was a phone number on stead of an address, this has also been corrected in the normalized version. I the final row there are fields that had not been populated, this has also been rectified in first normal form.

Instructor Table First Normal Form



Id	Name	SurName	Address	Phone	Email	ClassID
Inst_1007	Anna	Parker	12 Lakeside	8634567817	eegsa@me.com	Cardio Pilates
Inst_1002	Chris	O\Reilly	Inistioge	8634567812	skippy@att.net	Prenatal Yoga
Inst_1001	Imelda	Norris	12 Hollybank	8634567811	treeves@mac.com	Pilates
Inst_1004	Jill	Murillo	Culpeper,	8634567814	louise@optonline.net	Barre
Inst_1006	Joe	Mckay	Mary Street,	8634567816	timlinux@outlook.com	Pilates Advanced
Inst_1003	Kevin	Parker	Gartner Ave.	8634567813	bjornk@yahoo.com	Yoga
Inst_1005	Trish	Greer	Primrose Street	8634567815	fwitness@msn.com'	Meditation

Instructor Second Normal Form

StudioID (IDStudioInstructort)
Instructor (StudioInstructorfirstname, StudioInstructorSurname, StudioInstructoraddress, StudioInstructorPhone, StudioInstructorEmail)

Database Security

- Having consulted the client I have suggested certain security limit and constraints be placed on users of the database.

- Customers will not have access to the database.

- Access is limited to the Owner (client) and the Instructors so that they to be able to look up a individual customers for the purpose of background information regarding occupation. This will be limited to the class(es) that the Customer attends the frequency of attendance.

- The instructor is limited to customer information, class timetable and class booking information and all queries that are linked to this information.

- An instructor will not be able look up financial matters.

- An instructor will not be able look up personal information on other instructors.

- An instructor will not be able look up personal information on the client.

- The owner will have sole access to the Instructor information.

- The owner will have sole access to how much the customer spends, payment status and any private email address and phone number information.

SQL Scipts

All Scripts as listed below along with the name of the script and what it does, These scripts have been saved and named .
I have also included Data and structure dump scripts:
Dumpdataonly.sql
DumpStructureanddata.sql
Dumpstructureonly.sql

studioinstructorall instructors.sql

```
SELECT * FROM imeldastudiodb.studioinstructor
```

Allows Client and Instructors retrieve all the data in the tbl_studioinstructor

---

studiocustomerAll customers.sql

```
SELECT * FROM imeldastudiodb.studiocustomer;
```

Allows Client and Instructors retrieve all the data in the tbl_studiocustomer so that all information on each customer can be accessed quickly.

---

instructorswithkilkennyaddress.sql

```
SELECT * FROM imeldastudiodb.studioinstructor
where studioInstructorAddress like '%kilkenny'
```

Retrieve all the data that relates to instructors with a Kilkenny address In the tbl_studioinstructor.

---

instructorswithkilkennyaddresswithmac address.sql

```
SELECT * FROM imeldastudiodb.studioinstructor
where studioInstructorAddress like '%kilkenny'
and studioinstructoremail like '%mac%'
```

Retrieve all the data that relates to instructors with a Kilkenny address In the tbl_studioinstructor but only one with mac address

---

studioclassequipmentlist.sql

```
SELECT * FROM imeldastudiodb.studioclassequipmentid;
```

Lets the client and the instructors know which type of equipment is need for each class.

ClasspaymentID

```
SELECT * FROM imeldastudiodb.studioclass
use imeldastudiodb;
SELECT * FROM studioclasspayment;
```

Lets the client and the instructors know the payment id for each class and if there is no payment id they know that the customer has not yet paid.

Inner join

```
SELECT studioclasstype.studioclasstypecost, studioinstructor.idstudioInstructor from
studioinstructor
inner join studioclasstype on studioclasstype.studioClasstypename =
studioinstructor.studioClassID;
```

This an example of an inner join selects re
cords that have matching values in both tables.

---

customerOccupation.sql

```
SELECT * FROM imeldastudiodb.studiocustomer.StudioCustomerOccupation;
SELECT * FROM imeldastudiodb.studiocustomer
where studiocustomeroccupation like '%officer'
```

This allows the instructor to look up the occupation of customers which can give an indication of the physicality of their occupation (with overall assessment of a sedentary person)

---

studiobooking.sql

```
SELECT * FROM imeldastudiodb.studiobooking;
```

Both Client and Instructor can see what classes have been booked by who.

---

distinct studiobookingdate.sql

```
SELECT studiobookingdate FROM studiobooking;
```

The result of this query shows duplicate values.

```
SELECT distinct studiobookingdate FROM studiobooking;
```

I have edited the above query and added DISTINCT which eliminates any duplicate values.

Minimum Class Cost.sql

```
SELECT MIN(StudioClassTypeCost)
FROM StudioClassType;
```

This Query gives the user the minimum cost of a class at the studio

---

Maximum Class Cost.sql

```
SELECT max(StudioClassTypeCost)
FROM StudioClassType;
```

This Query gives the user the maximum cost of a class at the studio

---

Conclusion

I learned a lot from doing this assignment as I can see from my end results I made some mistakes early on that I did not realise would affect me until I was near the end. For instance in my attempt to label everything in a very precises way I ended up elongating attributes for the entities. For example as part of the Customer Entity I named the attributes as StudioCustomerfirstname which made doing the scripts harder as I had increased my chances of a spelling error. It was a good lesson but my database still does what I want it to do.

The database examples given in this project were for a client that runs a Fitness Studio but she works alone therefore performs all of the Instructor roles herself. In the future she hopes to expand her business and take on other instructors thus making use of the new database as proposed in this document. The client does in the future want to develop her business and add classes, instructors, more studios, and the database can be changed as desired. The client will be able to amend the database which will enable her to keep her workload manageable when taking on a new project. Using the CRUD system in the design allows the client to avoid duplication of information and the use of obsolete data therefore ensuring qualitative as well as quantitative results.

I have zipped my assignment which contains the database and scripts and all Excel files which I used to import the data from. I used this method as I was originally given some excel files by the client.