

ECO BIKE HUB CASE STUDY

Ewan James Richards (1802187)

UNIVERSITY OF WALES TRINITY ST DAVID Database Systems, MSc Software Engineering

Abstract

A working database has been created to solve the problems addressed by Eco Bikes Hub. The differences of SQL and NoSQL has been debated as well as suitable fragmentation strategies.

Contents

Abstract	1
Table of Figures.....	2
Task 1: Database Creation.....	4
Coursework Practical Element 1	4
Data Dictionary	4
ERD Diagram	7
Rationale	8
Task 2: Performance Tuning Enhancement.....	10
Coursework Practical Element 2	10
File 1	10
File 2	13
Task 3: Report.....	19
Written Element 1.....	19
SQL vs. NoSQL.....	19
Written Element 2.....	22
Performance Tuning	22
Transaction justification	22
Distributed Database Solution	23
Conclusion.....	25
References	26

Table of Figures

Figure 1: Store table	4
Figure 2: Staff table	4
Figure 3: Customer table	5

Figure 4: Bike table	5
Figure 5: Parts table.....	5
Figure 6: Booking table	6
Figure 7: Appointment Types table.....	6
Figure 8: ERD Diagram.....	7
Figure 9: Relationships	8
Figure 10: Constraints	9
Figure 11: Create Database	10
Figure 12: Creating the Store/Staff tables.....	10
Figure 13: Creating the Customer/Bikes/Parts tables	11
Figure 14: Creating the Booking/Appointment Types tables.	11
Figure 15: Test Data: Appointment Types/Bikes/Parts	13
Figure 16: Test Data: Store/Staff tables	13
Figure 17: Test Data: Customer/Booking Tables	14
Figure 18: Booked date constraint.....	14
Figure 19: Cancelled date/time constraint.....	15
Figure 20: Customer Date of Birth Constraint.....	15
Figure 21: Staff Leaving Date Constraint	15
Figure 22: Postcode length constraint.....	16
Figure 23: Staff bonus range constraint.....	16
Figure 24: Staff position constraint.....	16
Figure 25: Discount range constraint.....	17
Figure 26: Transaction/Commit.....	17
Figure 27: SQL types.....	19
Figure 28: SQL Vs. NoSQL	20
Figure 29: Advantages and Disadvantages of SQL and NoSQL	21
Figure 30: Indexes.....	22

Task 1: Database Creation

Coursework Practical Element 1

Data Dictionary

Store

Column Name	Data Type and Size	Constraints	Comments
<u>BranchNo</u>	Char (5)	Primary Key Check: values must be 1 to 9.	Not null
Street Address	Varchar (30)		Not null
City	Varchar (20)		Not null
Postcode	Varchar (8)	Length NOT >=6	Not null
<u>PhoneNo</u>	Varchar (12)		Not null

Figure 1: Store table

Staff

Column Name	Data Type and Size	Constraints	Comments
<u>StaffNo</u>	Char (6)	Primary Key	Not null
FirstName	Varchar (30)		Not null
Surname	Varchar (30)		Not null
DOB	Date		Not null
Gender	Varchar (30)		Not null
Position	Varchar (20)		Not null
Salary	Decimal (12, 2)		Not null
Bonus	Decimal (4, 2)		Not null
Email Address	Varchar (100)		Not null
Contact number	Varchar (12)		Not null
Start date	Date		Not null
Leaving Date	Date	Check constraint null or leaving date must be after start date	
Branch No	Char (5)	Foreign Key	Not null
Line Manager	Char (6)	Foreign Key	

Figure 2: Staff table

Customer

Column Name	Data Type and Size	Constraints	Comments
Customer number	Char (7)	Primary Key	Not null
First Name	Varchar (30)		Not null
Surname	Varchar (30)		Not null
Contact number	Varchar (12)		Not null. Telephone number
Email Address	Varchar (100)		Not null
Date of Birth	Date	Customer must be over 18 to make an account	Not null
Registration Date	Date		Not null. Date registered at store.
Password	Varchar (16)	Account password	

Figure 3: Customer table

Bikes

Column Name	Data Type and Size	Constraints	Comments
Bike ID	Char (7)	Primary Key	Not null
Make	Varchar (10)		Not null
Model	Varchar (10)		

Figure 4: Bike table

Parts

Column Name	Data Type and Size	Constraints	Comments
Part ID	Char (5)	Primary Key	Not null. Part ID Number.
Part Name	Varchar (10)		Not null
Bike ID	Varchar (10)		Not null

Figure 5: Parts table

Booking

Column Name	Data Type and Size	Constraints	Comments
Booking number	Int Identity (1, 1)	Primary Key	Not null
Customer number	Char (7)		Not null
Booking Date and Time	Date Time	Constraint, Get Date	Not null
Bike ID	Char (7)	Bike ID Number	Not null
Appointment Type	Varchar (15)	Type of Appointment	Not null
Booked Date	Date		Not null
Price	Decimal (8, 2)	Booking cost	Not null
Discount	Decimal (4, 2)	Check constraint must be between 0 and 1	
Commission	As	Price multiplied by 0.15	
Cancellation Date and Time	Date and Time		
Comments	Varchar (1000)		

Figure 6: Booking table

Appointment Types

Column Name	Data Type and Size	Constraints	Comments
Appointment ID	Varchar (15)	Primary Key	Not null. Appointment ID number
Type name	Varchar (15)		Not null. Type of appointment
Duration Hours	Int		Not null. Length of appointment in hours

Figure 7: Appointment Types table

ERD Diagram

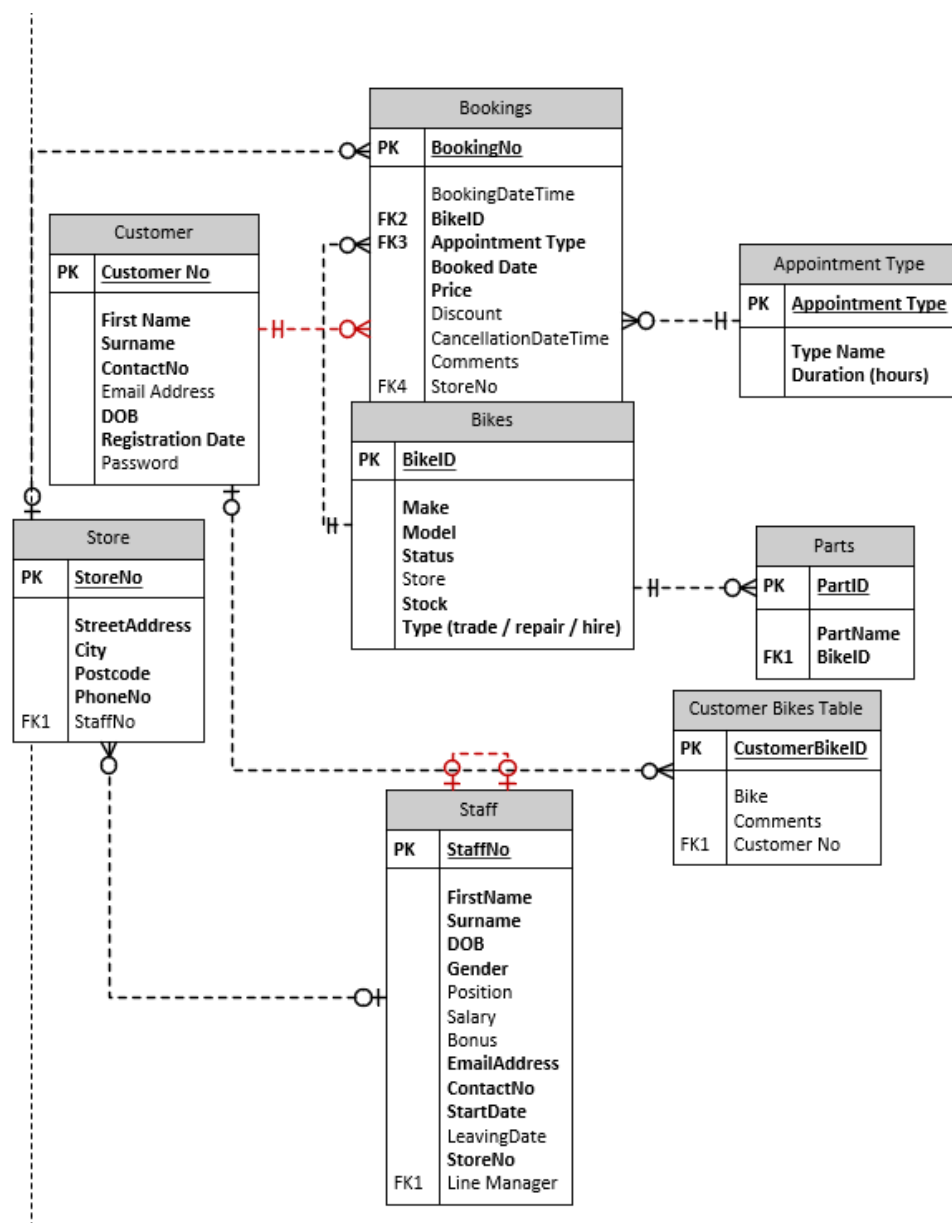


Figure 8: ERD Diagram

There will be eight tables with seven relationships as well as a unique primary key on every table with foreign keys in most tables which are connected to the relevant relationships, as shown in Figure 9: Relationships.

Relationships

From Table	From Column	To Table	To Column
Staff	<u>StaffNo</u>	Store	<u>StaffNo</u>
Customer	Customer No	Customer Bikes Table	Customer No
Bikes	<u>BikeID</u>	Parts	<u>BikeID</u>
Bikes	<u>BikeID</u>	Bookings	Bike ID
Store	<u>StoreNo</u>	Bookings	<u>StoreNo</u>
Appointment Type	Appointment Type	Bookings	Appointment Type
Staff	Line Manager	Staff	<u>StaffNo</u>

Figure 9: Relationships

StaffNo (Staff) – StaffNo (Store): The StaffNo from the Staff table would be related to the StaffNo to the Store table as there would be 1 member of staff to many stores.

CustomerNo (Customer) – CustomerNo (Customer Bikes Table): the CustomerNo from the Customer table would be related to the CustomerNo on the Customer Bikes Table as there would be one customer for many bikes.

BikeID (Bikes) – BikeID (Parts): The BikeID from the Bikes table would be related to the Bike ID on the Parts table as there would be one bike for many parts.

BikeID (Bikes) – BikeID (Bookings): The BikeID from the Bikes table would be related to the BikeID on the Bookings table as there would be one bike for many bookings.

StoreNo (Store) – StoreNo (Bookings): The StoreNo from the Store table would be related to the StoreNo on the Bookings table as there would be one store for many bookings.

Appointment Type (Appointment Type) – Appointment Type (Bookings): The Appointment Type from the Appointment Type table would be related to the Appointment Type on the Bookings table as there would be one appointment type for many bookings.

Rationale

Normalisation

Normalization is the method of sorting data in a database that take on a set of conditions which are there is no data repetition, and the data dependencies are logical

[1]. Normalisation would be used because it can reduce data repetition and allow databases to function effectively and have the data organised.

Foreign Keys

A foreign key are columns in a relational database that relates data together in two separate tables within the database [2]. Foreign keys would be used because they can be used for relationships and once the foreign keys have been implemented alongside with the relationship, it avoids data duplication as, although some data may be duplicated, it would all be related together and appear on the main table it is related to.

Constraints

A check constraint ensures the authenticity of data being inserted into a database and provides data reliability as well as limiting the value range that can be identified in a column [3]. Check constraints would be used because it would discover any errors and ensure that, using the rollback, that the statement is terminated and does not enter the database.

The table below identifies the table in which check constraints will be used as well as how the constraint will be used in the database. The below table does not include the primary key or foreign keys as these are defined in Figure 8: ERD Diagram.

Table	Column	Constraint
Store	Postcode	Postcode must NOT be >= 6.
Staff	Position	Must be Branch Manager, Manager, Supervisor, Assistant.
Staff	Bonus	Between 0 and 0.5.
Staff	Leaving Date	Leaving date must be after the starting date.
Customer	Date of Birth	Customer must be over 18.
Booking	Discount	Between 0 and 1.
Booking	Cancellation Date Time	Cancellation Date Time must be before booked date.

Figure 10: Constraints

Task 2: Performance Tuning Enhancement

Coursework Practical Element 2

File 1

Create Database

```
CREATE DATABASE EcoBikes
```

Figure 11: Create Database

Table creation

```
CREATE TABLE Store ( --Create Store table
  BranchNo CHAR(5) NOT NULL CONSTRAINT PK_Branch PRIMARY KEY CONSTRAINT CHK_Branch_BranchNo CHECK(BranchNo LIKE 'B[0-9][0-9][0-9]'), --Branch Number / Primary Key
  StreetAddress VARCHAR(30) NOT NULL,
  City VARCHAR(20) NOT NULL,
  Postcode VARCHAR(8) NOT NULL
  CONSTRAINT CHK_Branch_Postcode CHECK(Len(Postcode) >= 6),
  PhoneNo VARCHAR(12) NOT NULL, --Telephone Number
);
GO

CREATE TABLE Staff ( --Create Staff Table
  StaffNo CHAR(6) NOT NULL CONSTRAINT PK_Staff PRIMARY KEY, --Staff ID Number / Primary Key
  FirstName VARCHAR(30) NOT NULL,
  Surname VARCHAR(30) NOT NULL,
  DOB DATE NOT NULL, --Date of Birth
  Gender VARCHAR(30) NOT NULL,
  Position VARCHAR(20)
  CONSTRAINT CHK_Staff_Position CHECK (Position IN ('Branch Manager', 'Manager', 'Supervisor', 'Assistant')),
  Salary DECIMAL(12,2),
  Bonus DECIMAL(4,2) --Bonus between 0 or 0.5
  CONSTRAINT CHK_Staff_Bonus CHECK (Bonus BETWEEN 0 AND 0.5),
  EmailAddress VARCHAR(100) NOT NULL,
  ContactNo VARCHAR(12) NOT NULL,
  StartDate DATE NOT NULL,
  LeavingDate DATE,
  BranchNo CHAR(5) NOT NULL CONSTRAINT FK_Staff_Branch FOREIGN KEY REFERENCES Store(BranchNo),
  LineManager CHAR(6) CONSTRAINT FK_Staff_LineManager FOREIGN KEY REFERENCES Staff(StaffNo), --Foreign Key
  CONSTRAINT CHK_Staff_LeavingDate CHECK(LeavingDate IS NULL OR LeavingDate > StartDate) --calculation / difference between leaving date and start date
);
```

Figure 12: Creating the Store/Staff tables

Store: The BranchNo column is the primary key for the table. If the Postcode is higher than 6, it will not be accepted into the database.

Staff: The StaffNo column is the primary key for the table with the position being three possible options – Branch Manager, Manager, Supervisor and Assistant. In terms of the Bonus column, this would be for the staff bonus – and would be a decimal number between 0 and 0.5. The BranchNo is a foreign key because it is related with the BranchNo primary key column on the Store table. The Line Manager is a foreign key with the StaffNo on the Staff table as each member of staff is assigned a position, if there is a line manager at a specific staff ID number it will be on the LineManager column.

```

CREATE TABLE Customer (
    CustomerNo CHAR(7) NOT NULL CONSTRAINT PK_Customer PRIMARY KEY, -- Customer ID Number / PRIMARY KEY
    FirstName VARCHAR(30) NOT NULL,
    Surname VARCHAR(30) NOT NULL,
    ContactNo VARCHAR(12) NOT NULL, --Telephone Number
    EmailAddress VARCHAR(100) NOT NULL,
    DOB DATE NOT NULL CONSTRAINT CHK_Customer_DOB CHECK(DATEDIFF(mm, DOB, GETDATE()) / 12 >= 18), --Date of Birth / customers must be 18 or over
    RegistrationDate DATE NOT NULL, -- date registered at store / account
    [Password] VARCHAR(16), -- customer account password
);
GO

CREATE TABLE Bikes (
    BikeID CHAR (7) NOT NULL PRIMARY KEY, --Bike ID Number / Primary Key
    Make VARCHAR (10) NOT NULL,
    Model VARCHAR (10),
);

CREATE TABLE Parts(
    PartID VARCHAR (5) NOT NULL PRIMARY KEY, -- Part ID Number / Primary Key
    PartName VARCHAR (15) NOT NULL,
    BikeID CHAR (7) NOT NULL,
);

```

Figure 13: Creating the Customer/Bikes/Parts tables

Customer: The CustomerNo attribute is a primary key. The DOB attribute is the column for the date of birth, to allow the customer to make an account, they must be 18 or over, if they are under 18, they will not be able to create an account. The RegistrationDate is the date that the customer registered their account with the store. With the exception to the passport attribute, all attributes are NOT NULL – data is required for these fields. This was done because the customer may not want to have an online account, they would not need a password.

Bikes: The Bikes ID attribute is the primary key. With the exception the model, all attributes are NOT NULL – data is required for these fields.

Parts: The PartID attribute is the primary key of the Parts table. All attributes are NOT NULL, data is required for all the fields on the table

```

CREATE TABLE Booking ( -- Creating the Booking table
    BookingNo INT IDENTITY(1, 1) NOT NULL CONSTRAINT PK_Booking PRIMARY KEY, --Booking ID Number / Primary Key
    CustomerNo CHAR(7) NOT NULL CONSTRAINT FK_Booking_Customer FOREIGN KEY REFERENCES Customer(CustomerNo), --Foreign Key / Relationship with Customer No on ...
    BookingDateTime DATETIME2 NOT NULL CONSTRAINT DF_Booking_BookingDateTime DEFAULT GETDATE(), --date / time booked on
    BikeID CHAR(7) NOT NULL, --Bike ID Number
    AppointmentType VARCHAR(15) NOT NULL, -- Type of Appointment
    BookedDate DATE NOT NULL, --Appointment Date
    Price DECIMAL(8,2) NOT NULL, --Booking cost
    Discount DECIMAL(4,2) CONSTRAINT CHK_Booking_Discount CHECK (Discount BETWEEN 0 AND 1), --discount
    Commission AS Price * 0.15,
    CancellationDateTime DATETIME2, --Cancellation date and time, if needed. Not required.
    Comments VARCHAR(1000),
    CONSTRAINT CHK_Booking_CancellationDateTime CHECK(CancellationDateTime < BookedDate) --Dates between the booked date and when it was cancelled
);
GO

CREATE TABLE AppointmentTypes( -- creating the appointment types table
    AppointmentID VARCHAR (15) NOT NULL, --appointment ID number
    TypeName VARCHAR (15) NOT NULL, -- type of appointment
    DurationHours INT NOT NULL, --length of appointment in hours
);
GO

```

Figure 14: Creating the Booking/Appointment Types tables.

Booking: The BookingNo is the primary key attribute for the booking table. The CustomerNo is the foreign key, it is related to the CustomerNo primary key attribute

on the Customer table. The BoookedDateTime attribute is the date and time that the appointment was booked with the Price attribute is for the cost of the appointment. There is a check constraint on the Discount attribute, it is a decimal value of between 0 and 1. There is another check constraint for the CancallationDateTime and Booked date, The CancellationDateTime must be before the appointment date, not after it. Apart from Discount, CancellationDateTime and Comments, all attributes are NOT NULL – data is required for these fields. This was done because some bookings may not have a discount nor be cancelled. For the specific return dates and repair tracking, this would be mentioned in the comments section of the table. Some bookings may not require having any comments, hence why this can be a NULL value.

Appointment Types: The AppointmentID is the primary key of the appointment types table. All attributes are NOT NULL, data is required for all fields in this table.

Implementing the Performance Tuning enhancements

Refer to Figure 30: Indexes to view the performance tuning enhancements implemented for Eco Bikes.

File 2

Test Data

```
use EcoBikes

-- inserting data into AppointmentTypes table
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT1', 'BikeService', 1)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT2', 'Repair', 2)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT3', 'Sales', 1)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT4', 'Return', 1)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT5', 'Enquiry', 0.5)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT6', 'Hiring Bike', 168)
INSERT INTO AppointmentTypes (AppointmentID, TypeName, DurationHours) VALUES ('AT7', 'Return Hire', 1)

Go

-- inserting data into Bikes table
INSERT INTO Bikes (BikeID, Make, Model) VALUES ('BK1', 'Carrera', 'Virtuoso')
INSERT INTO Bikes (BikeID, Make, Model) VALUES ('BK2', 'Carrera', 'Vanquish')
INSERT INTO Bikes (BikeID, Make, Model) VALUES ('BK3', 'Pendleton', 'Somerby')
INSERT INTO Bikes (BikeID, Make, Model) VALUES ('BK4', 'Apollo', 'Cafe')

Go

-- inserting data into Parts table
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT1', 'Brake', 'BK1')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT2', 'Wheels', 'BK1')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT3', 'Handles', 'BK1')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT4', 'Brake', 'BK2')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT5', 'Wheels', 'BK2')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT6', 'Handles', 'BK2')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT7', 'Brake', 'BK3')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT8', 'Wheels', 'BK3')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT9', 'Handles', 'BK3')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT10', 'Brake', 'BK4')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT11', 'Wheels', 'BK4')
INSERT INTO Parts (PartID, PartName, BikeID) VALUES ('PT12', 'Handles', 'BK4')

Go
```

Figure 15: Test Data: Appointment Types/Bikes/Parts

```
-- inserting data into Store table
INSERT INTO Store (BranchNo, StreetAddress, City, Postcode, PhoneNo)
VALUES ('B0001', '45 The Crescent', 'Swansea', 'SA1 6ED', '01792400000')
INSERT INTO Store (BranchNo, StreetAddress, City, Postcode, PhoneNo)
VALUES ('B0002', '10 Station Road', 'Llanelli', 'SA14 8ND', '01792400000')
INSERT INTO Store (BranchNo, StreetAddress, City, Postcode, PhoneNo)
VALUES ('B0003', '1 Horton Road', 'Swansea', 'SA1 7AD', '01792123321')
INSERT INTO Store (BranchNo, StreetAddress, City, Postcode, PhoneNo)
VALUES ('B0004', '2 Gower Street', 'Swansea', 'SA2 9GR', '01792321123')
INSERT INTO Store (BranchNo, StreetAddress, City, Postcode, PhoneNo)
VALUES ('B0005', '45 Principality Road', 'Cardiff', 'CF1 6ED', '013545285732')

-- inserting data into Staff table
INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
VALUES ('S001', 'Ewan', 'Richards', '21-Feb-2000', 'Male', 'Assistant', '18000', '0.25', '1802187@student.uwtsd.ac.uk', '01554 123123', '10-Jan-2022', NULL, 'B0001', NULL)
INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
VALUES ('S002', 'Bev', 'Holland', '21-Oct-1990', 'Prefer not to say', 'Manager', '25000', '0.25', 'beverleyh@yahoo.com', '01792 321321', '11-Jan-2022', NULL, 'B0002', NULL)
INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
VALUES ('S003', 'Michael', 'Dacey', '30-May-1980', 'Male', 'Manager', '25000', '0.5', 'mdacey@gmail.com', '01554 123123', '23-Dec-2021', NULL, 'B0003', NULL)
INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
VALUES ('S004', 'John', 'Davies', '1-Jan-1999', 'Male', 'Assistant', '18000', '0.5', 'j.a.davies@googlemail.com', '01792 567890', '24-Dec-2021', NULL, 'B0004', NULL)
INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
VALUES ('S005', 'Ewan', 'Owen', '28-Feb-2001', 'Male', 'Manager', '25000', '0.5', 'ewan.owen@colegsingar.ac.uk', '01554 129856', '23-Dec-2021', NULL, 'B0005', NULL)

Go
```

Figure 16: Test Data: Store/Staff tables

```

-- inserting data into Customer table
INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
VALUES ('01', 'Harriett', 'Shoubridge', '01554 287973', 'hshoubridge@about.me', '21-Jan-1995', '24-Dec-2021', 'QgQHA2ieEj', '5 Mayer Court Llanelli', 'SA153YE')
INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
VALUES ('02', 'Monro', 'Beswell', '01792 671882', 'mbeswell1@google.co.uk', '10-Apr-2000', '23-Dec-2021', 'ZfSC6uW', '2 American Ash Junction Swansea', 'SA23NM')
INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
VALUES ('03', 'Othilia', 'Maharg', '01554 819417', 'omaharg2@hexun.com', '18-May-1980', '24-Dec-2021', 'X7P4YteC', '4 Victoria Circle Llanelli', NULL);
INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
VALUES ('04', 'Thorny', 'Roote', '01792 521768', 'troote3@businessweek.com', '6-Feb-1990', '23-Dec-2021', 'jKMQGwSc42ak', '12 Alpine Court Swansea', NULL);
INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
VALUES ('05', 'Bartel', 'Fender', '01554 865112', 'bfender4@etsy.com', '2-Dec-2001', '23-Dec-2021', 'cq10aTuR', '25 Shasta Plaza Llanelli', 'SA148LL')
Go

-- inserting data into Booking table
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('01', '24-Dec-2021', 'BK1', 'AT1', '27-Dec-2021', 50, 0, NULL, 'Need it back ASAP')
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('02', '23-Dec-2021', 'BK2', 'AT2', '28-Dec-2021', 75, 0, NULL, NULL)
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('03', '24-Dec-2021', 'BK3', 'AT1', '29-Dec-2021', 50, 0, NULL, NULL)
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('04', '23-Dec-2021', 'BK4', 'AT2', '29-Dec-2021', 75, 0, NULL, NULL)
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('05', '24-Dec-2021', 'BK2', 'AT2', '30-Dec-2021', 50, 0, NULL, NULL)
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('06', '27-Dec-2021', 'BK1', 'AT2', '28-Dec-2021', 75, 0, NULL, 'Need it repaired urgently')
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('02', '13-Jan-2021', 'BK1', 'AT5', '13-Jan-2021', 0, 0, NULL, 'Enquiry regarding specialised part for bike, parts recommended by staff')
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('03', '12-Jan-2021', 'BK2', 'AT6', '13-Jan-2021', 50, 0, NULL, 'Bike hire for 1 week, return 20th January 2022')
INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
VALUES ('03', '13-Jan-2021', 'BK2', 'AT7', '20-Jan-2021', 50, 0, NULL, 'Price paid for on the 13th January 2022.')
Go

```

Figure 17: Test Data: Customer/Booking Tables

Testing the constraints

Booked date constraint

```

1 use EcoBikes
2
3 INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
4 VALUES ('03', '24-Dec-2021', 'BK1', 'AT1', '12-Jan-2022', 50, 0, NULL, NULL)
5

```

Messages

(1 row affected)

Completion time: 2022-01-13T14:56:51.4765629+00:00

Figure 18: Booked date constraint

This constraint is for the BookedDate column on the booking table, the database can read today's date and any appointment before today's date should not be able to be added to the bookings table. When this constraint was tested, the date was the 13th January 2022, so the booked date entered was the 12th January 2022, the row was successfully entered which means the date has been inserted onto the table. This constraint was unsuccessful.

Cancelled Date/Time

```
7  --testing the cancelled date constraint
8  INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
9  VALUES ('03', '24-Dec-2021', 'BK1', 'AT1', '13-Jan-2022', 50, 0, '30-Jan-2022', NULL)
10
```

1% Messages

Msg 547, Level 16, State 0, Line 8
The INSERT statement conflicted with the CHECK constraint "CHK_Booking_CancellationDateTime". The conflict occurred in database "EcoBikes", table "dbo.Booking".
The statement has been terminated.

Completion time: 2022-01-13T15:10:50.6931769+00:00

Figure 19: Cancelled date/time constraint

A new record was entered, and the date was after appointment time. As shown in the error message at the bottom of the figure, the statement has been terminated because the cancelled date is after the appointment date.

Customer Date of Birth Constraint

```
7  -- testing the cancelled date constraint
8  INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
9  VALUES ('03', '24-Dec-2021', 'BK1', 'AT3', '13-Jan-2022', 50, 0, '30-Jan-2022', NULL)
10
11 -- testing the Customer Date of Birth constraint
12 Insert into Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
13 VALUES ('06', 'Janet', 'Thomas', '01554 593759', '759.janett@me.com', '13-Jan-2012', '24-Dec-2021', 'A2ieEj', '5 Main Road Llar
14
15 INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
16 VALUES ('S007', 'James', 'McJohn', '27-Feb-2001', 'Male', 'Assistant', '25000', '0.5', 'james.mcjohn@mail.com', '01554 397275', '13-Jan-2022', '10-Jan-2022', 'B0001', NULL)
17
```

110% Messages

Msg 547, Level 16, State 0, Line 8
The INSERT statement conflicted with the CHECK constraint "CHK_Booking_CancellationDateTime". The conflict occurred in database "EcoBikes", table "dbo.Booking".
The statement has been terminated.

Completion time: 2022-01-13T17:14:05.2992334+00:00

Figure 20: Customer Date of Birth Constraint

The constraint is that the customer must be 18 or over to create an account at Eco Bikes. The date of birth entered was the 13th January 2022, meaning that the age of the customer is 11, so their record should not be able to be entered into the database. The constraint has worked.

Staff Leaving Date Constraint

```
15 INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
16 VALUES ('S007', 'James', 'McJohn', '27-Feb-2001', 'Male', 'Assistant', '25000', '0.5', 'james.mcjohn@mail.com', '01554 397275', '13-Jan-2022', '10-Jan-2022', 'B0001', NULL)
```

3% Messages

Msg 547, Level 16, State 0, Line 16
The INSERT statement conflicted with the CHECK constraint "CHK_Staff_LeavingDate". The conflict occurred in database "EcoBikes", table "dbo.Staff".
The statement has been terminated.

Completion time: 2022-01-13T15:55:19.2366650+00:00

Figure 21: Staff Leaving Date Constraint

The constraint is that the leaving date must not before the start date. For this test, the start date entered was the 13th January 2022 and the leaving date entered was

10th January 2022, the constraint worked, and the record has not been entered into the Staff table.

Postcode length constraint

```
18 -- postcode length constraint
19 INSERT INTO Customer (CustomerNo, FirstName, Surname, ContactNo, EmailAddress, DOB, RegistrationDate, [Password], [Address], Postcode)
20     VALUES ('07', 'Tom', 'Jones', '01972 792207', 'tomj.singer@thevoice.com', '10-Nov-1975', '13-Jan-2022', 'djY409m', '5 Voice Street Cardiff', 'CF10 1TV');
21
```

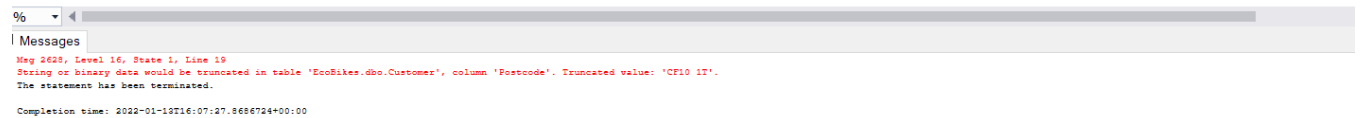


Figure 22: Postcode length constraint

The constraint is that the postcode value cannot be over 6, this postcode has 7 digits in it so it should not be permitted into the table. The constraint has worked, and the statement has been terminated.

Staff bonus range constraint

```
22 -- staff bonus range constraint
23 INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
24     VALUES ('08', 'Ewan', 'James', '21-Feb-2000', 'Male', 'Manager', '25000', '0', 'ewan.james@student.uwtsd.ac.uk', '01554 982758', '9-Jan-2021', NULL, 'B0001', NULL)
25 INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
26     VALUES ('09', 'David', 'Mitchel', '20-Feb-1990', 'Male', 'Manager', '25000', '0.90', 'd.mitchell@scelebrity.com', '01792 152142', '10-Jan-2021', NULL, 'B0002', NULL)
```

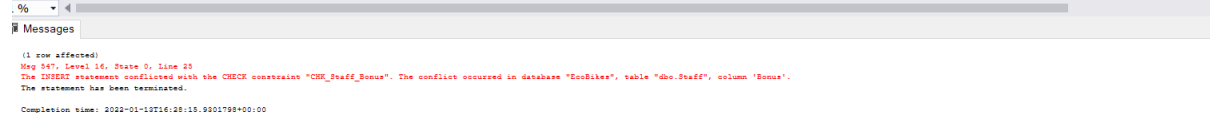


Figure 23: Staff bonus range constraint

The constraint is that the staff bonus must be between 0 or 0.5. One record has 0 bonus, the other having 0.90. the 0 bonus should be committed into the database but the 0.90 should not be committed as it is outside of the range. The 0 bonus record has been committed but the 0.90 bonus record has been terminated; the constraint has worked.

Staff position constraint

```
28 -- staff position constraint
29 INSERT INTO Staff (StaffNo, FirstName, Surname, DOB, Gender, Position, Salary, Bonus, EmailAddress, ContactNo, StartDate, LeavingDate, BranchNo, LineManager)
30     VALUES ('10', 'Charlie', 'Chaplin', '20-Jan-1950', 'Male', 'Branch Supervisor', '25000', '0.50', 'cc@entertainer.com', '01792 374854', '13-Jan-2021', NULL, 'B0004', NULL)
```

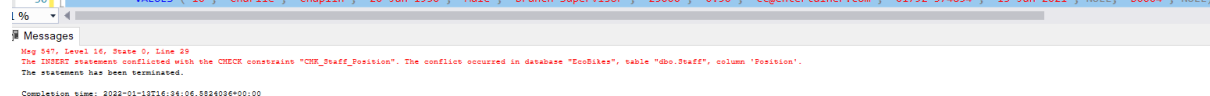


Figure 24: Staff position constraint

The constraint is that the data in this column can only be Branch Manager, Manager, Supervisor, Assistant. For this test, Branch Supervisory was used and the constraint worked as the statement was terminated.

Discount range constraint (Booking table)

```
32 | -- discount range constraint
33 | INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
34 | VALUES ('03', '14-Jan-2022', 'BK1', 'AT4', '15-Jan-2022', 50, 2, NULL, 'Need it repaired urgently')
```

0 %

Messages

Msg 547, Level 16, State 0, Line 33
The INSERT statement conflicted with the CHECK constraint "CHK_Booking_Discount". The conflict occurred in database "EcoBikes", table "dbo.Booking", column 'Discount'.
The statement has been terminated.

Completion time: 2022-01-13T17:11:39.4115826+00:00

Figure 25: Discount range constraint

The constraint is that the discount must be between 0 and 1. For the test, the discount was 1.5 and the constraint worked as the statement was terminated.

Transaction

```
1 | BEGIN TRANSACTION --WORKING VERSION
2 | --Add Booking 1
3 | BEGIN TRY
4 | INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
5 | VALUES ('01', '13-Jan-2022', 'BK2', 'AT1', '16-Jan-2022', 50, 0, NULL, NULL)
6 |
7 | --Add Booking 2
8 | INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
9 | VALUES ('03', '13-Jan-2022', 'BK4', 'AT2', '17-Jan-2022', 75, 0, NULL, 'Urgent repairs needed')
10 | COMMIT TRANSACTION
11 |
12 | END TRY
13 | BEGIN CATCH
14 |
15 | ROLLBACK TRANSACTION
16 |
17 | END CATCH
18 |
19 | BEGIN TRANSACTION --ROLLBACK VERSION
20 | --Add Booking 1
21 | BEGIN TRY
22 | INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
23 | VALUES (02, '13-Jan-2022', 'BK2', 'AT1', '16-Jan-2022', 50, 0, NULL, 'No rush to have it back, take your time.')
24 |
25 | --Add Booking 2
26 | INSERT INTO Booking (CustomerNo, BookingDateTime, BikeID, AppointmentType, BookedDate, Price, Discount, CancellationDateTime, Comments)
27 | VALUES (04, '13-Jan-2022', 'BK4', 'AT2', '17-Jan-2022', 75, 0, NULL, 'Very important')
28 | COMMIT TRANSACTION
29 |
30 | END TRY
31 | BEGIN CATCH
32 | PRINT 'Error';
33 | ROLLBACK TRANSACTION
34 |
35 | END CATCH
```

: %

Messages

(0 rows affected)
Error

Completion time: 2022-01-13T10:56:14.3808777+00:00

Figure 26: Transaction/Commit

The transaction at Figure 26: Transaction/Commit worked and the data was successfully added to the database however the transaction at the bottom of Figure 26: Transaction/Commit did not commit into the database and therefore failed. An error message is printed alongside the number of rows affected, because it failed to commit no rows have been affected by this transaction. This is because the dates in the second commit was rolled back as the booked dates had already been taken.

Task 3: Report

Written Element 1

SQL vs. NoSQL

Harrington states that “SQL is a database manipulation language that has been implemented by virtually every relational database management system (DMBS) intended for multiple users, partly because it has been accepted by the American National Standards Institute (ANSI) and International Standards Organization (ISO) as a standard query language for relational databases” [4]. In terms of flexibility, it depends on what type of what SQL developers use to implement their databases. The two types of Structured Query Language (SQL) are dynamic and static, as shown below in

Type	Description
Dynamix SQL	Dynamic SQL is more flexible than static SQL mainly because dynamic SQL programs can handle change in data definition values, as SQL statements could be changed anytime.
Static SQL	With static SQL, as it is unable to handle these changes – this means that it is not flexible. Most databases that are implemented using SQL are vertically scalable – this signifies those developers can increase the capacity on just a lone server by expanding key elements such as RAM.

Figure 27: SQL types [5] [6].

Type	Description
Dynamix SQL	Dynamic SQL is more flexible than static SQL mainly because dynamic SQL programs can handle change in data definition values, as SQL statements could be changed anytime.
Static SQL	With static SQL, as it is unable to handle these changes – this means that it is not flexible. Most databases that are implemented using SQL are vertically scalable – this signifies those developers can increase the capacity on just a lone server by expanding key elements such as RAM.

Figure 27: SQL types

According to Mullins, Vaughan, and Beal, “NoSQL is an approach to database management that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats. A NoSQL database generally means that is a non-relational, distributed, flexible and scalable” [7].

Both sources explicitly describe how both SQL and NoSQL are useful for DMBS as well as their connection with relational databases. One of the most significant changes is the scalability with SQL being vertical and NoSQL being horizontal – NoSQL performs better than SQL because of it having more servers available, which means it can handle increased traffic more effectively.

The table below states the difference between SQL and NoSQL databases [8].

	SQL databases	NoSQL databases
Data Storage Model	Table with fixed rows and columns	Table with dynamic rows and columns
Development History	Developed in 1970s to reduce data repetition.	Developed in 2000s to allow for quick development by Agile and DevOps practices.
Examples (companies used it)	Oracle, MySQL, Microsoft SQL Server, and PostgreSQL.	MongoDB, CouchDB, HBase, Graph, Neo4j and Amazon Neptune
Primary Purpose	General purpose	Holds a lot of data, amazing relationships between data.
Schemas	Inflexible	Flexible
Scaling	Vertical	Horizontal
Joins	Required	Not required

Figure 28: SQL Vs. NoSQL

Advantages and Disadvantages

The advantages and disadvantages of SQL and NoSQL are outlined below in Figure 29: Advantages and Disadvantages of SQL and NoSQL [8], [9], [10], [11], [12], [13].

SQL	Advantages	Disadvantages
SQL	The advantages of SQL databases are that it is easy to manage the database as not a lot of coding is required to control it. SQL can also be used on servers and mobile devices, not just laptops and Computer PCs and, users can quickly recuperate data from the database. SQL has been around for many years; it was initially used by the American National Standards Institute (ANSI) in 1986 and by the International Standards Organization (ISO) in 1987. SQL databases do not cost that much, and it is used to develop DBMS by IBM, Oracle and Microsoft. It is easy to learn and, although SQL can create databases, it can also manage the database security communicating data with many system users. It can also connect clients and servers, it supports the client-server architecture. Many IT support roles now require SQL skills.	that the User Interface (UI) is generally quite poorly designed, and developers find it hard use, there is quite a high cost with them meaning that some users cannot use it because they cannot afford it, the security of SQL is quite low as the threat level is high due to the amount of important information it holds and it does not allow full control to users
NoSQL	The advantages of NoSQL databases are that it works well with object-oriented programming, it works well with modern software development methodologies that involve agile sprints, NoSQL can handle a lot of structured and unstructured data	The disadvantages of NoSQL is that it is not enhanced for deducing data repetition, and it does not support Atomicity, Consistency, Isolation and Durability (ACID) transactions.

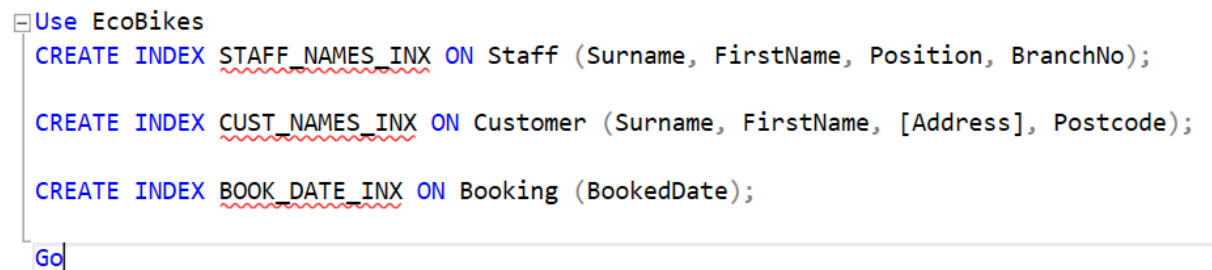
Figure 29: Advantages and Disadvantages of SQL and NoSQL

Based on the research compiled, SQL would be the more efficient language to use because of its connections with major IT communities, its data security models, and development history are more developed, and many major companies have used it since its launch in 1986. The general advantages also outweigh the disadvantages as well as how it's generally better than NoSQL. With SQL, it assists with reducing data repetition and it is enhanced just for queries.

Written Element 2

Performance Tuning

Secondary Indexes



```
Use EcoBikes
CREATE INDEX STAFF_NAMES_INX ON Staff (Surname, FirstName, Position, BranchNo);

CREATE INDEX CUST_NAMES_INX ON Customer (Surname, FirstName, [Address], Postcode);

CREATE INDEX BOOK_DATE_INX ON Booking (BookedDate);

Go
```

Figure 30: Indexes

The created indexes on the staff, customer and booking tables have been created to make searching more efficient in the database.

STAFF_NAMES_INX. The index was created on the staff table for the surname, first name, position, and Branch number columns. This index was created because management can find staff members surname and first name and view their details. If they want to see all staff working at a particular branch, they can enter the branch number and all staff would appear.

CUST_NAMES_INX. The index was created on the customer table for the surname, first name, address, and postcode columns. The index was created because when customers visit a store, staff members can search the customers surname and first name, and their account would appear. If there is a customer with the same name, the customer would be asked to confirm their address and postcode, which would be visible to the staff member.

BOOK_DATE_INX. The index was created because when customers ask whether there is any date available for a booking, the staff member can search for that specific date and their query could be answered.

Transaction justification

Figure 26: Transaction/Commit are the transaction created. The transaction that was implanted can be used to address the problems raised by Eco Bike Hubs as if the wrong data is entered or if the date has already been entered then an error message

shall show stating that the transaction would not work however if the transaction has been successful, the record would be entered into the database.

Distributed Database Solution

Distributed Database Management Systems (DDMS)

A distributed database is a compilation of many connected databases, which are distributed substantially across many locations that talks on a computing device. A DDMS is an integrated structure that controls a distributed database in a way as if it was stored in a single location. It has many features such as the following: [14]

- Create, save, revise, and remove distributed databases [14].
- Data can be updated anywhere [14].
- Data integrity [14].

DDMS has many benefits, such as a smaller transmission costs, the data within the database is delivered in an effective method and is dependable. However, there are some disadvantages of DDMS such as it could be complicated, and the sensitivity of queries is largely dependent upon appropriate data distribution. [14]

Horizontal

For Eco Bikes, the rows would be stored on the Branch table as well as the Booking as the appointments would then show on the Branch table. This also applied with the Staff, the Staff table would appear on the Branch table so that, when searching on the Branch table, the staff members would know where each other is assigned.

Vertical

For Eco Bikes, the address and contacts columns could be transferred and put into a separate store contact table and have the branch manager in the main branch table. Also, for the staff table, the salary, bonus, and start / leave dates could be placed into a Human Resources (HR) table and the remaining columns remain.

For the Customers table, the RegistrationDate and Password columns would be separated into an online account table with the other columns remaining in the Customer table. The Bikes, Parts, Booking and Appointment Types tables would remain as they were initially created.

Mixed

For Eco Bikes, all the locations would have its own table, such as Cardiff would have its own table, Llanelli would have its own table and Swansea would have its own table. The staff and stores based at these locations would be related to these tables.

Derived

Both the Branch and Customer are primary tables which are then related to other tables. The Customer table would be joined with the Branch table as well as the Booking table as the Booking number and relevant data can be saved to the customer table which shows when they had appointments and when it's related to the Branch table shows which appointments are taking place at stores each day as well as which customers would be in the stores, and which stores the customer visits. This can identify which store are customers visiting more and whether this could assist expand the company further.

Reconstruction of Global Relations

By using these relevant fragments, it would remain as one big database, but it would be stored in many tables to make it more efficient.

Conclusion

This report set out to analyse the requirements of Eco Bike Hubs and propose a database design. It also addressed problems which had to be demonstrated by using SQL commented code by using testing rollback and commit operations. The report also addressed the debate of using a SQL database versus a NoSQL database. Transaction needed to be explained on how they address the problems raised by Eco Bike Hubs as well as create a suitable fragmentation scheme for the system.

The requirements of Eco Bikes Hubs have been analysed and a database design has been proposed by creating a database schema prototype to show the company. A Microsoft SQL Server database, using SQL statements, has been created that will enable Eco Bikes to manage customer requirements. It meets the features of being normalised to 3 normal forms, the design structure has been demonstrated with a primary key being created for each table with referential integrity constraints and check constraints being implemented. Realistic records have been populated on each table using Insert statements. The data has been used to test the functionality of the primary key, foreign key, and check constraints.

Sample code has been provided to demonstrate how the use of a transaction may be used to address the problems alongside with testing rollback and commit operations for the actions needed.

SQL and NoSQL have been debated, and SQL has been discussed as being more effective than NoSQL. The database structure has been evaluated and how it could be improved and enhanced further with fragmentations, such as horizontal and vertical. DDMS has been researched and has assisted with writing these enhancements.

References

- [1] Techioedia, "Normalization," Techioedia, 2022. [Online]. Available: <https://www.techopedia.com/definition/1221/normalization>. [Accessed 13 January 2022].
- [2] J. Rabelo, "Foreign Key," Techopedia, 14 August 2020. [Online]. Available: <https://www.techopedia.com/definition/7272/foreign-key>. [Accessed 13 January 2022].
- [3] W3Schools, "SQL CHECK Constraint," w3schools, 2022. [Online]. Available: https://www.w3schools.com/sql/sql_check.asp. [Accessed 13 January 2022].
- [4] J. L. Harrington, SQL Clarly Explained, Third Edition, Burlington, MA, USA: Morgan Kaufman, 2010.
- [5] J. Russell, Application Developer's Guide - Fundamentals, Oracle Corporation, 1999.
- [6] L. Shiff and W. Rowe, "SQL vs NoSQL Databases: What's The Difference?," BMC Blogs, 5 March 2018. [Online]. Available: <https://www.bmc.com/blogs/sql-vs-nosql/>. [Accessed 6 December 2021].
- [7] C. S. Mullins, J. Vaughan and B. Beal, "NoSQL (Not Only SQL database)," SearchDataManagement, April 2021. [Online]. Available: <https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>. [Accessed 6 December 2021].
- [8] L. Schaefer, "NoSQL vs SQL Databases," MongoDB, 2022. [Online]. Available: <https://www.mongodb.com/nosql-explained/nosql-vs-sql>. [Accessed 12 January 2022].

- [9] Java Point, "https://www.javatpoint.com/dbms-advantage-of-sql," Java Point, 2022. [Online]. Available: <https://www.javatpoint.com/dbms-advantage-of-sql>. [Accessed 12 January 2022].
- [10] S. Thakur, "What is SQL, its Applications, Advantages and Disadvantages?," What is DBMS, 2 June 2017. [Online]. Available: <https://whatisdbms.com/what-is-sql-applications-advantages-and-disadvantages/>. [Accessed 12 January 2022].
- [11] "12 Benefits of SQL (Structured Query Language)," Systematix, 2022. [Online]. Available: <https://www.systematix.co.uk/tips-tricks/12-benefits-of-sql-structured-query-language>. [Accessed 12 January 2022].
- [12] MongoDB, "Benefits Of NoSQL," MongoDB, 2022. [Online]. Available: <https://www.mongodb.com/scale/benefits-of-nosql>. [Accessed 12 January 2022].
- [13] S. Vatsal, "Advantages And Disadvantages Of SQL: A Popular Choice For Databases!," Dare2Compete, 12 January 2022. [Online]. Available: <https://dare2compete.com/blog/advantages-and-disadvantages-of-sql>. [Accessed 12 January 2022].
- [14] Tutorials Point, "Distributed DBMS - Distributed Databases," Tutorials Point, 2022. [Online]. Available: https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_databases.htm. [Accessed 13 January 2022].