Shahriyar Ahmed

Big data and database systems

3908891

Contents

[Task 1: 1](#_Toc169965997)

[Strengths vs weaknesses: 5](#_Toc169965998)

[Task 2 7](#_Toc169965999)

[Strengths vs weaknesses: 9](#_Toc169966000)

[Task 3: 9](#_Toc169966001)

[Strengths vs weaknesses: 12](#_Toc169966002)

[Task 4 16](#_Toc169966003)

[Strengths vs weaknesses: 17](#_Toc169966004)

# Task 1:

A diagram of a computer network

Description automatically generated with medium confidence

Each table is an entity which is the object in which I stored the data, this contains attributes which are the characteristics of the entity.

I have created my ERD diagram based on the case study. The tables I have used include:

**Employee, Seaport, Cabin, Ship, Voyage, Reservation, Passenger, Entertainment, Schedule, Ticket, payment**

**Employees who live in various locations all over the EU because of seaports that sail all over world and the data of these is stored in this table and is linked to a seaport in which they work at.**

**The attributes in this table are EmployeeID, Firstname, Lastname, City, country and phone number. EmployeeID is primary key and identifier.**

**Relationships:**

**One – many: employee to reservations**

**Many-to-one: seaports to employee**

**Another table required from the case study is a seaport table, this table stores all information about seaports that sails out of the European Union and details about the, the attributes required for this are listed in case study. This includes: Seaport code, city, country and phone number. Seaport code is primary key and main identifier.**

Relationships:

One-to-many: one seaport to many employees.

One-to-many: One seaport to many voyages.

**In the case study, another table required is the Ship table which is used to store data on cruise ships. The attributes for this table include ship name, ship type, number of cabins, ship full displacement, fuel capacity and maximum speed.**

Relationships:

One-to-many: one ship to many voyages

**In addition to this, another table required from case study is a voyage table which includes all details of voyage. This includes attributes such as a unique voyage number which is the primary key, the main identifier of this table. Origin, destination departure date, arrival date and Ship name which is used in another table in that case it’s a foreign key.**

Relationships:

One schedule to many voyages

One voyage to many reservations

One cabin to many reservations

**Additional tables I have identified are:**

**Passenger table which includes all details of passengers. Attributes for this include: PassengerID, FirstName, lastname, Date of Birth, address, phone number and email. PassengerID being a primary key and identifier to identify passengers.**

Relationships:

Many-to-many: Many passengers to many reservations

One-to-many: Passenger to entertainment

One-to-many: One passenger to many tickets

**Cabin table, which include details of ship and where on ship passengers have booked and information related to booking of Cabin. Attributes for this table: Cabin number, ship name, cabin type, capacity and price per night.**

Relationships:

Many-to-one: Many cabins to one ship

One cabin to many tickets

A cabin to many passengers

One cabin to many reservtions

**Entertainment table, which includes the names of different types of entertainment stated in case study e.g. live music, swimming pool, fitness activities and when it’s run. Attributes include: EntertaimentID, Type, schedule. EntertaimentID being the main primary key, which is when different activities on offer**

Relationships:

Many-to-one: Many entertainment to one passenger

One -to-one: One entertainment to one schedule

**To link into the entertainment table, I created a schedule table which stated when each passenger had each entertainment booked. The primary key for this is the scheduleID, this is unique identifier. Attributes for this include: EntertainmentID is, voyage number, date, time, location, PassengerID . EntertainmentID is foreign key because it’s primary key in entertainment table. PassengerID which is primary key, this identifies each person’s schedule according to passenger.**

Relationships:

Many-to-one: Many schedule to one voyage

One-to-one: One schedule to one passenger

Many-to-one: Many entertainment can have one schedule

**Payment table a way in which to record each passenger’s payment. Attributes for this include: paymentID, reservationID, payment date, amount and payment method. The primary key is the paymentID, the foreign key is reservationID.**

**Relationships:**

**Many payment to one reservation**

**Many payment to one ticket**

**Finally I created a ticket table, which contains all information about the ticket because the case study states that passengers have to pay by cash or card. Also how they want it delivered e.g. by post or regional office.**

**Attributes for this include: TicketID, reservationID, issue date, delivery method.**

**TicketId being the primary key and reservationID being the foreign key**

**Relationships:**

**Many tickets have one reservation**

**Many tickets have one voyage**

**Many tickets have one cabin**

**Strengths vs Weaknesses**

## Strengths vs weaknesses:

Strengths:

Creating an ERD Diagram allowed me to visualize the stakeholders involved in the development of a database to see the entities and attributes of each table required in the database; it can be used as a communication tool so everyone has a shared perspective on the database.

This allowed me to identify requirements in how each entity relates to each other and what attributes they have and in a diagram where all data is on one page.

It’s a reference for any modifications in the future

Weaknesses:

In the scenario, in order to create the database, a lot of entities and attributes were required due to the case study. Because of this there was a lot of repetition this led to the difficulty of reading the diagram as it includes more data and tables.

It was hard to make changes overtime, and the need to frequently update the diagram, once I commit changes it’s hard to do a lot of steps

Finally, stakeholders, other people viewing my work involved in the creation of this will find it difficult to read as there is a lot of lines for the relationships, without explicitly stating what each relationship meant, it was hard for me to identify anything from looking at it

# Task 2

Table creation

Here I create all the tables necessary in SQL, I have created all the tables I have described in my ERD diagram, I have created these along with data types.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Strengths vs weaknesses:

Strengths:

Creating tables in SQL allowed me to structure my data and organize it using data structures, this allowed me to run queries relating to objects in table

It allowed me to create tables and specify data types for columns, and set constraints sch as a primary key and foreign key, and define relationships between tables

Weaknesses:

It was complex, especially when I had to include a lot of tables with a lot of constraints. I had to learn a whole new language for data manipulation .

It was time-consuming and lower performance because what happens if I choose the wrong data type and lead to slower execution of the query.

If there is a lot of tables, it’s hard to manage the size of growth in the database

# Task 3:

I used SQL server management studio to create my database as the syntax is straightforward.

Below is the data I inserted:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Strengths vs weaknesses:

Strengths:

Accurate and reliable:

When inserting data, I had to specify the constraints e.g. when creating a foreign key and data types.

This allowed me to create foreign key relationships, which is important to maintain referential integrity, this is that all data in tables is to be remained consistent and valid throughout all tables An example of this is where I referenced a primary key for one table, this is a foreign key for another table, the data that is in the foreign key table has to match the primary key table otherwise error will throw up. This allows for data integrity to make sure the incorrect data type isn’t entered.

In addition, this allowed precise control of what data I want to and can enter in

Weakness:

I had to learn a whole new language for this module and for operations such as creating and inserting data

Inserting data into SQL server management is complex due to the syntax you have to learn. You also have to manually insert and manipulate data into SSMS. This can lead to errors when inserting data if I’m not careful. Once I have created the tables and specified the constraints I would often have to drop the table to modify data as it had a foreign key attached to the table and relied on another table for it to execute

Using SSMS was time-consuming as I had to build a lot of databases for the case study; this often left me with no or little time for inserting data and running queries. also, this was inefficient because of the repetitiveness when inserting data and creating databases.

Below is output of data I inserted:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

3a) Write a query to show details of Reservation Clerks, including their location and the number of reservations taken, specifically for those employees who have taken the highest and the lowest total number of reservation

A screenshot of a computer

Description automatically generated

3b) Write a query to calculate individual totals for the number of: complete, incomplete and cancelled reservations.

A screenshot of a computer

Description automatically generated

# Task 4

This dashboard visualizes the first query from task 3 which was:

Write a query to show details of Reservation Clerks, including their location and the number of reservations taken, specifically for those employees who have taken the highest and the lowest total number of reservations.

A screenshot of a computer

Description automatically generated

The visualization shows, the list of employees with their first name and last names and the number of reservations easily visualized in a bar chart, the different colours denotes the names of different countries. All of this is easily seen for the user and displayed in one dashboard.

## Strengths vs weaknesses:

Strengths:

In tableau I could easily visualize the data when I imported the .csv file, I had many options to customize the colour and size and adjust what type of graph I want to visualize the data in a click of a button

It’s easy to integrate with other sources of data and pull in data from other sources including mySQL, SQL server management server and from excel files. This makes the software user friendly

As I was a student, the software was free with the licence otherwise, I would have to pay for software

Weaknesses:

I found it time-consuming to learn at first, as there are a lot of buttons and settings when creating graphs. Unless you know how to use the software, I struggled in terms of learning and getting to grips with software. There was no guidance as to how to use the software to visualize data into graphs.