4.10.3 Database Normalisation

1) A garage that services and repairs cars uses a relational database to keep track of the jobs that customers have booked. The database includes jobs that have been completed and jobs that are waiting to be done.

The details of the jobs that the garage does, together with the parts that it stocks and uses are stored in the database using the four relations shown in **Figure 1**.

Figure 1

Job (<u>JobID</u>, CarRegNo, JobDate, InGarage, JobDuration)

Car (<u>CarRegNo</u>, Make, Model, OwnerName, OwnerEmail, OwnerTelNo)

Part (<u>PartID</u>, Description, Price, QuantityInStock)

PartUsedForJob (<u>JobID, PartID</u>, QuantityUsed)

- A type of car can be uniquely identified by the combination of its Make and Model. Different Makes may use the same Model name and a particular manufacturer (Make) will produce several different car Models.
- A booking make for a car on a particular date counts as one job, regardless of how many different tasks are completed upon it.
- A job might require the use of any number of parts, including zero.

It has been suggested that the owner details (OwnerName, OwnerEmail, OwnerTelNo) should not be stored in the Car relation and that a new relation should be created to store owner details separately from car details.

a) Explain why storing the owner details in a separate relation would improve the design of the database.

[2 marks]

1 mark from:

- **A person may own more than one car // a person may bring different cars to the garage**
- **The garage might want to store details of an owner before details of their car are known**

1 mark from:

- **Avoids inputting/storing owner details multiple times for each car they own**
- **Transferring cars between owners would only require one attribute change in the Car relation**
- **Reduce data redundancy**
- **Improve data consistency**
- **Eliminate update/insertion anomalies**

There are restrictions on which parts can be fitted to which cars. For example:

- The driver's door mirror with PartID 108 can only be fitted to one make and model of car.
- The ignition switch with PartID 57 can be fitted to any model of car for one make as the maker uses the same ignition switch in all models.
- The tyre with PartID 91 can be fitted to a wide range of cars of different makes and models as it is a standard size.

b) Explain how the database could be modified to record which makes and models of car each part can be fitted to.

[3 marks]

- **Create a new relation to identify which make/model of car each part can be fitted to**
- **Store the attributes PartID, Make and Model in the new relation**
- **Make the PartID, Make and Model the entity identifier**

---

2) State **two** reasons why database designs are usually normalised.

[2 marks]

- **Minimise data duplication**
- **Eliminate data redundancy**
- **Eliminate data inconsistency**
- **Eliminate update/insertion/deletion anomalies**

---

3) A veterinary practice with four different surgeries intends to use a relational database to store the data that it needs to manage its business.

Customers of the practice are pet owners who bring their pets to one of the surgeries for appointments. The surgeries are staffed by vets.

- Each customer is identified by a unique identity number and the customer's forename, surname and telephone number are recorded.
- Each pet is identified by a unique identity number and the pet's name, type and the date of birth are recorded.
- Each surgery is identified uniquely by its name. The town in which it is located and the surgery's telephone number are recorded.
- Each vet is identified by a unique identity number and the vet's forename and surname are recorded.

A pet is owned by one or more customers and each customer may own any number of pets. Over their lifetimes, pets may attend many appointments.

To make an appointment for a pet, a customer contacts a surgery. The appointment is made for the pet to take place on a particular date and time at a specific surgery.

Each vet is associated with one surgery which they work at; each surgery is staffed by several vets.

Develop a **fully normalised design** for a relational database to store the information required by the veterinary practice. The Pet, Surgery and Vet relations have already been done for you.

Pet(PetID, PetName, Type, DateOfBirth)
Surgery(SurgeryName, Town TelephoneNumber)
Vet(VetID, VetForename, VetSurname, SurgeryName)

[4 marks]

- **Customer(CustomerID, Forename, Surname, TelephoneNumber)**
- **Appointment(PetID, Date, Time, SurgeryName)**
- **PetOwner(CustomerID, PetID)**

4) A sport centre uses a relational database to store information about its facilities (such as the swimming pool) and the bookings that have been made to use them.

Figure 2 shows the structure of the relations in the database.

Figure 2

Facility (<u>FacilityID</u>, Description, MaxPeople, PricePerHour)

FacilityForSport (<u>Sport, FacilityID</u>)

Booking (<u>FacilityID, BookingDate, StartTime</u>, EndTime, CustomerID)

Customer (<u>CustomerID</u>, Forename, Surname, EmailAddress)

- The Facility relation stores the different facilities available at the sports centre. Each one is identified by a unique number and has a brief description. For example, the facility with FacilityID 1 has a description 'Outdoor Pitch A'. PricePerHour is the price of hiring a facility for 1 hour. For example, the facility with FacilityID 1has a price of £17.50.
- The FacilityForSport relation identifies which facilities are suitable for which sports. For example, the facility with FacilityID 1 is suitable for football, rugby and hockey and would therefore require three separate records in this relation.
- The Booking relation stores the bookings that have been made. Bookings must start and end either on the hour, at quarter past, half past or quarter to the hour. A customer can make bookings for more than one facility for the same time. For example, a badminton club secretary might book both of the indoor sports halls for the same time. However, each facility can only be booked by one customer at any one time.
- The Customer relation stores the details of customers who have make bookings.

A different design was originally proposed for the database. This design did not have the Customer relation and had the following design for the Booking relation:

Booking(<u>FacilityID, BookingDate, StartTime</u>, EndTime, Forename, Surname, EmailAddress)

Give **two reasons** why the alternative database design would be less suitable than the design shown in Figure 2.

[2 marks]

- **The design is not normalised // there is data redundancy // inconsistent data could occur // There are attributes that do not rely on the primary key**
- **If a customer made more than one booking then their details would need to be entered more than once**
- **If the customer details are entered more then once this could lead to data inconsistency**

- **Deleting all of the bookings that a customer made would also delete the customer information**
- **The customers details could not be stored before they make a booking**
- **It would be difficult to identify all the bookings for one customer as they do not have a unique identifier.**

---

5) A network of zoos uses a relational database system to store information about the animals that they have so that they can be matched up with animals at other zoos in a breeding programme.

Figure 3 shows the structure of the relations in the database.

Figure 3

Zoo (<u>ZooName</u>, Town, Country)

AnimalLocation (<u>AnimalID, ZooName, DateArrived</u>, DateLeft)

Animal (<u>AnimalID</u>, IndividualName, Species, DataOfBirth, Sex)

Match (<u>AnimalFemaleID, AnimalMaleID, DateOfMatch</u>, Successful)

- The AnimalLocation relation identifies which zoos each animal has lived at. The zoo that the animal is currently at can be identified because the DateLeft attribute is set to 01/01/0001 to indicate that the animal has not left.
- The Animal relation stores details of the individual animals that are available to be matched with other animals for breeding.
- The Match relation stores details of matches that have been made. The attributes AnimalFemaleID and AnimalMaleID refer to the AnimalID values of the two matched animals in the Animal relation.

a) Which statement below (A-D) **does not have to be true** for a fully normalised database.

A	Each attribute in a relation is dependent on the primary key.
B	Each attribute in a relation is dependent only on the primary key; it is not also dependent on any other attribute in the relation.
C	The primary key in each relation consists of only one attribute.
D	There are no repeating groups (or equivalently each attribute is atomic.

[1 mark]

- **C**

b) An additional attribute, ZooName, is added to the Animal relation to store the animal's current location. Describe one advantage and one disadvantage of this change.

Describe **one advantage** and **one disadvantage** of adding this new attribute to the relation.

[2 marks]

- **Advantage**
- **It will be quicker to lookup an animal's current location**
- **The current location of an animal can be identified without having to query the AnimalLocation relation**

- **Disadvantage**
- **Additional storage space will be required**
- **This will introduce data redundancy (as the information can already be found from the AnimalLocation relation)**
- **Data inconsistency could occur as the current location in the Animal relation might not match the current location in the AnimalLocation relation**
- **More updates will be required when an animal is move between zoos**

6) A company operates a cinema which has three different screens. Each screen has a capacity determined by the number of seats the screen has.

Each seat is identified by a unique seat number for the screen it is in, but two seats in different screens can have the same number. A specific seat is classified as being of one of two types: standard or deluxe.

Each screen can have multiple showings per day, and a film can be shown multiple times.

Customers make bookings to go to the cinema. Each booking is for one specific showing of a film. A booking is for one or more seats, and the customer can select the individual seats that they want to book when they make the booking.

When a booking is made, if the customer has not previously made a booking, the customer's first name, last name and telephone number are recorded. A booking records the customer who made it, but not the individual people sitting in each seat. If a customer has made a booking previously then the details that were stored about them when the previous booking was made are re-used.

a) Develop a **fully normalised design** for a relational database to store the information required by the cinema. The Screen, Seat, Film and Showing relations have already been defined below.

Using the same format, list the other **three** relations together with the attributes that each relation will contain. Underline the primary key.

Screen (ScreenNumber, Capacity)

Seat (SeatNumber, ScreenNumber, SeatType)

Film (FilmID, FilmName, Duration, Certificate)

Showing (ShowingID, ScreenNumber, FilmID, ShowTime, ShowDate)

[5 marks]

- **Customer(CustomerID, FirstName, LastName, TelephoneNumber)**
- **Booking(BookingID, ShowingID, CustomerID)**
- **AssignedSeat(BookingID, SeatNumber)**

7) A shop that sells items through a website uses a relational database to store information about the products that it sells and the sales that it has made.

Figure 4 shows the structure of the relations in the database.

Figure 4

Product(<u>ProductID</u>, Description, QuantityInStock, SupplierID)

Sale(<u>SaleID</u>, CustomerID, SaleDate)

SaleLine(<u>SaleID, ProductID</u>, QuantitySold)

Customer(<u>CustomerID</u>, Forename, Surname, EmailAddress)

Supplier(<u>SupplierID</u>, SupplierName, SupplierEmail)

- The Product relation stores information about the products that the shop sells and who supplies them. Each type of product is identified by a unique number and has a brief description.
- The Sale and SaleLine relations are used to record the details of the sale of products to a customer. Each sale is identified by a unique SaleID, which is a number.
- The Customer relation stores the details of customers who have registered on the website so that they can purchase products. Each customer is identified by a unique CustomerID, which is a number.

a) Which statement below (**A**-**E**) indicates an assumption that has been made when the database was designed.

A	A customer cannot be added to the database until a sale has been made to them.
B	Each product is only supplied by one supplier.
C	Each supplier only supplies one product.
D	Only one sale can be made to a customer on a particular date.
E	Two different products cannot be purchased as part of the same sale.

[1 mark]

- **B**

b) The database described in **Figure 4** is fully normalised.

Describe **two** problems that can occur with databases that are **not** fully normalised.

[2 marks]

- **Redundant/duplicated data may waste storage space**
- **If data is stored more than once then it could be inconsistent // two copies of the 'same' data item might store different values;**
- **If data is stored more than once then each copy of the data would need to be updated if it changed**
- **It might not be possible to store data about one type of entity without creating a record for another type of entity // if a record for one type of entity does not exist then it might not be possible to store data about another type of entity**
- **When a record for one type of entity is deleted it might delete the data about another type of entity // it might not be possible to delete a record for one type of entity without deleting the data about another type of entity**
- **May be difficult to select/edit data if it is not atomic // if there are repeating groups;**

---

7. Relations in a database should usually be fully normalised.

Define what it means for a database to be fully normalised.

[2 marks]

- **Data is atomic // no repeating groups of attributes**
- **No partial (key) dependencies**
- **No non-key dependencies**
