INM370 – Advanced Databases

Tutorial 1 – ER-Diagram, Relational DB Modelling and SQL

1. Consider the following requirements for a car factory database.

The factory is 30 years old and is organized into departments based on different areas. Each department is identified by a unique code, has a name related to the area, a location, and a manager responsible for the department. The department is composed of employees, whose salaries start at £25K per year. Each employee is described by a social security number, has name, address, telephone number, and function. A manager of the department is an employee with salary greater than £120K per year. An employee works for a particular department.

The factory produces some items needed for assembling cars, and purchases other items from specific suppliers. However, no items have been purchased before 1995. Each produced item is developed by a different department, has a number, model, and a production date. Each department can produce different items, but an item can only be produced by one department. The suppliers of the purchased items are identified by a name, address, telephone number, contact person, and are located outside the UK. Each purchased item has a number and model, and can be supplied by one supplier many times. Each supplier can supply different items. The quantity of an item supplied and the related date have to be recorded. An invoice is generated for each request of an item. The invoices are identified by a number, have a payment date, and total amount due. A car produced in the factory is composed of different items and is identified by a number, colour, model, and date.

a) Design an ER-diagram for the above database. Your diagram should specify entities, relationships, attributes, keys, cardinality and participation constraints, and existential dependencies.

b) Translate the ER-diagram from a) into a set of relations (for example, following the guidelines discussed in the lectures; e.g. slides 25-26). Make sure that you identify all the primary and foreign keys.

2. The following tables form a part of a database held in a relational DBMS

Hotel	(hotelNo, hotelName, city)
Room	(roomNo, hotelNo, type, price)
Booking	(hotelNo, guestNo, dateFrom, dateTo, roomNo)
Guest	(guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;
Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;
Booking contains details of the bookings and (hotelNo, guestNo, dateFrom) forms the primary key;
and Guest contains guest details and guestNo is the primary key.[1]

a) List all double or family rooms with a price below £40.00 per night, in ascending order of price.
b) List the bookings for which no dateTo has been specified.
c) List the number of rooms in each hotel.
d) Create a view containing all the information about the guests who are from the United Kingdom.

[1] The queries that follow cover only a part of SQL; you ought to know a much wider part of the language (see the relevant Lecture 1 slides, for example), and thus must practice in your own time!

3. Consider the database of a theatre described below (the primary keys are underlined, the foreign keys are in *italics*):

   Play (PlayNum, Title, Period, Type, Writer)
   Actor (ActNum, Name, Address, Type)
   Participate (*PlayNum*, *ActNum*, PlayDate, StartTime)
   Ticket (SeatNum, PlayDate, *PlayCode* StartTime, Price,)

a) Create the above schema in SQL. Use appropriate data types for the attributes; base your choice on the datatypes indicated in the tables below (some of which might not be natively available in SQL!). Make sure that you specify referential integrity constraints.

b) Populate the database with the provided data for each table below.

c) Specify the following queries in SQL:

i) Find the plays that have the word ‘Family’ in the respective title.

ii) Give the total value of tickets sold for play number ‘P12’ performed on 20 June 2019.

iii) Find the name of the actors performing in the play written by Shakespeare on 20 June 2019.

iv) An SQL view named GoodSeats, which retrieves the seats that cost at least £25.00 and that have been sold for the play entitled Madame Butterfly.

Table Play				
PlayNum	**Title**	**Period**	**Type**	**Writer**
String - 5	*String - 30*	*String - 15*	*String - 15*	*String - 25*
P11	Madamme Butterfly	Spring 2019	Opera	G. Puccini
P12	A Midsummer Night’s Dream	Spring 2019	Classic	William Shakespeare
P13	The Lion King	Summer 2019	Musical	Irene Mecchi
P14	The Sad Family	Winter 2019	Drama	Paul Charles

Table Actor			
ActNum	**Name**	**Address**	**Type**
String - 5	*String - 20*	*String - 45*	*String - 10*
A1	Mary White	15 High Street, London, SW1	Opera
A2	John Black	20 Main Road, London, N15	Dance
A3	Lucy Smith	18 White Street, London, N13	Drama

Table Participate			
PlayNum	**ActNum**	**PlayDate**	**StartTime**
		Date	*String - 5*
P11	A1	10-Jun-2019	7:30
P11	A2	10-Jun-2019	7:30
P12	A3	20-Jun-2019	7:30
P12	A2	20-Jun-2019	7:30

P12	A1	20-Jun-2019	7:30
P13	A2	15-Jul-2019	2:30

Table Ticket				
SeatNum	**PlayDate**	**StartTime**	**Price**	**PlayCode**
String - 4	*Date*	*String - 5*	*Number - 4*	*String - 5*
A12	10-Jun-2019	7:30	50	P11
A13	10-Jun-2019	7:30	50	P11
B12	10-Jun-2019	7:30	45	P11
C11	15-Jul-2019	2:30	55	P13
C12	15-Jul-2019	2:30	55	P13
D20	20-Jun-2019	7:30	35	P12

4. Consider a database for a zoo including the following schema (the primary keys are underlined):[2]

    Animal (AniCode, Type, Location, Age)
    Staff (StaffCode, Name, Function, Address, Date_of_Birth)
    History (HistCode, HistDate, Description, AniCode)
    Takes_Care (*AniCode*, *StaffCode*, CareDate, Time, Procedure)

a) Define the above schema in SQL. Use appropriate data types for the attributes. In this case, you will decide on the data types of the attributes. Make sure that you specify referential integrity constraints.

b) Populate the database. In this case, you will decide on the data to use to populate the database.

c) Specify the following queries in SQL:

i. Find information about the history for all animals of type ‘giraffe’.

ii. Find the total number of times that the staff named ‘John White’ has taken care of the animal with code ‘AA12’.

iii. List the names of the staff and the codes of the animals that have been taken care by the respective staff, sorted by staff name.

[2] Please attempt this question only after you have done all others.