DATABASE DESIGN:
Normalization – Exercises & Answers

staffNo	dentistName	patientNo	patientName	appointment		surgeryNo
				date	time	
S1011	Tony Smith	P100	Gillian White	12-Aug-03	10.00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03	12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sept-03	10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sept-03	10.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03	16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03	18.00	S13

Figure 1: Details of patient dental appointments.

(a) The table shown in Figure 1 is susceptible to update anomalies. Provide examples of insertion, deletion, and modification anomalies.

Answers:

This table is not well structured, un-normalized containing redundant data. By using a **bottom-up** approach we analyzing the given table for anomalies. First observation, we see multiple values in an appointment column and this of course violate the 1NF. By assuming the staffNo and patientNo as candidate keys, there are many anomalies exist.

Insertion anomalies:

To insert a new patient particular that makes an appointment with the designated Doctor, we need to enter the correct detail for the staff. For example, to insert the details of new patient in patientNo, patientName and an appointment, we must enter the correct details of the doctor (staffNo, dentistName) so that the patient details are consistent with values for the designated Doctor for example, S1011.
To enter new patient data that doesn't have Doctor to be assigned we can't insert `NULL` values for the primary key.

Deletion anomalies:

If we want to delete a patient named Ian MacKay for example, two records need to be deleted as in row 3 and 4. This anomaly also obvious when we want to delete the dentistName, multiple records needs to be deleted to maintain the data integrity.
When we delete a Dentist record, for example Tony Smith, the details about his patients also lost from the database.

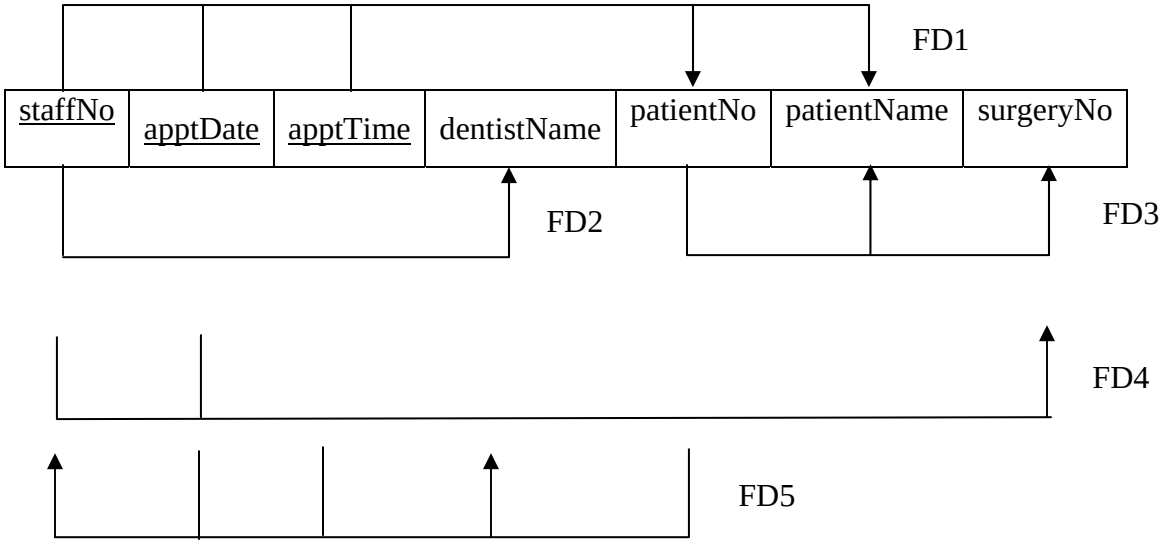
Modification anomalies:

With redundant data, when we want to change the value of one columns of a particular Dentist, for example the dentistName, we must update all the Dentist records that assigned to the particular patient otherwise the database will become inconsistent. We also need to modify the appointment schedules because different Dentist has different schedules.

(b) Describe and illustrate the process of normalizing the table shown in Figure 1 to 3NF. State any assumptions you make about the data shown in this table.

Assumptions made include that a patient is registered at only one surgery and he/she may have more than one appointment on a given day. All the schedules have been fixed for the whole days and week.

In the 1NF we remove all the repeating groups (appointment), assigning new column (apptDate and apptTime) and assigned primary keys (candidate keys). Then we figure out the functional dependencies (FDs). By using dependency diagram we represent the table as shown below. (NF – stand for Normal Form)
Note: How to find the FDs is subjective!!! However, the rule is, it must reflect the real word situation.



FD1 is already in 2NF. In this case, we can see that FD2 (just depend on staffNo) and FD4 (just depend on staffNo and apptDate) violate the 2NF. These two NFs are partially dependent on the candidate keys not the whole keys. FD2 can stand on its own by depending on the staffNo and meanwhile FD4 also can stand on its own by depending on the staffNo.
The FD3 violates the 3NF showing the transitive dependency where surgeryNo and patientName depend on patientNo while patientNo depend on the staffNo that is the non-key is depending on another non-key.

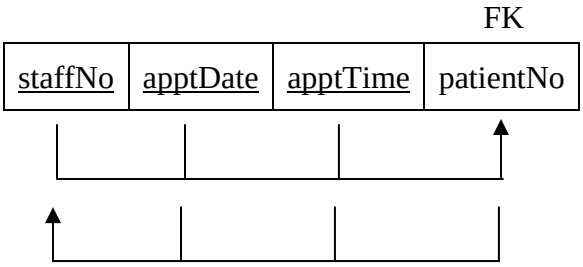
The 2NF, it is already in 1NF and there is no **partial dependency**. So we need to remove the FD2 and FD4 by splitting into new tables and at the same time creating foreign keys. The new tables that are in 2NF are shown below.

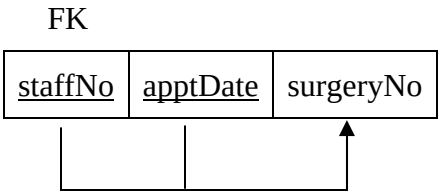
staffNo	apptDate	apptTime	patientNo	patientName

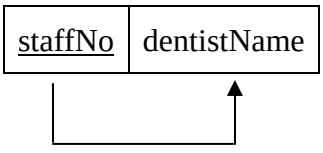
staffNo	apptDate	surgeryNo

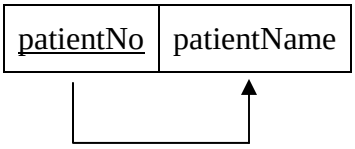
staffNo	dentistName

Finally in 3NF we must remove the transitive dependency. In this case we remove the FD3 by splitting into a new table. The transitive dependency left is the patientName that depend on the patientNo, so we split this into new table while creating a foreign key.









Let re-arrange and giving names to the tables.

staffNo	dentistName

Dentist(staffNo, dentistName)

FK

staffNo	apptDate	surgeryNo

Surgery(staffNo, apptDate, surgeryNo)

patientNo	patientName

Patient(patientNo, patientName)

FK

staffNo	apptDate	apptTime	patientNo

Appointment(staffNo, apptDate, apptTime, patientNo)

An agency called *InstantCover* supplies part-time/temporary staff to hotels throughout Scotland. The table shown in Figure 2 lists the time spent by agency staff working at two hotels. The National Insurance Number (NIN) is unique for employee.

NIN	contractNo	hoursPerWeek	eName	hotelNo	hotelLocation
113567WD	C1024	16	John Smith	H25	Edinburgh
234111XA	C1024	24	Diane Hocine	H25	Edinburgh
712670YD	C1025	28	Sarah White	H4	Glasgow
113567WD	C1025	16	John Smith	H4	Glasgow

Figure 2: Employees of *InstantCover* and their contracts to work at hotels.

(1) The table shown in Figure 2 is susceptible to update anomalies. Provide examples of insertion, deletion, and modification anomalies.

Answers:

It is obvious that the NIN and contractNo can be candidate keys. There are data redundancies in term of repeating groups.

Insertion anomalies:

To insert the details of new hotel we must know the NIN of the staff because this is primary key and we cannot assign NULL value for primary key. For example if we want to insert the details of new hotel, we must also insert the NIN and contractNo.
To recruit a new staff we need to know the details of hotel. For example if we want to recruit and assign new staff at H4 we need to enter the correct details of H4 so that the hotel details are consistent with values for hotel H4.

Deletion anomalies:

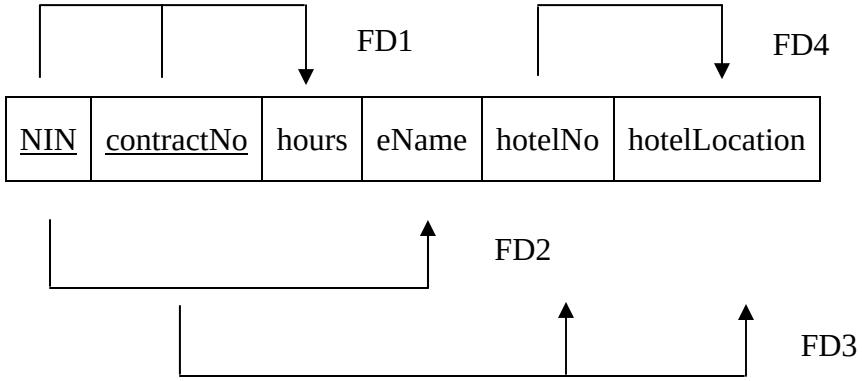
If we delete a record from the table that represent the staff name for example, John Smith, the details about the contract also will be lost from the database and will affect other record as seen in the second row of the contractNo (C1024).
Oppositely, when we delete a contractNo, other staff details will also be lost. For example if we delete C1025, both Sarah White and John Smith details will be lost.

Modification anomalies:

If we change the value of the contractNo, several records need to changed, for example row 1 and 2, also row 3 and 4 if we edit either contractNo.
If we edit the staff details, for example updating John Smith, several records need to be updated. This also applies to the hotelNo column.

(2) Describe and illustrate the process of normalizing the table shown in Figure 2 to 3NF. State any assumptions you make about the data shown in this table.

In the 1NF we remove all the repeating groups if any and assigned primary keys (candidate keys). Then we figure out the functional dependencies (FDs). By using dependency diagram we represent the table as shown below. (NF –stand for Normal Form)



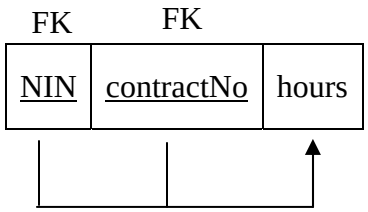
Obviously FD2 and FD3 violate the 2NF (partial dependency) and FD4 violate 3NF (transitive dependency). In the 2NF we remove the partial dependency of the FD2 and FD3 by splitting it into new tables as shown below.

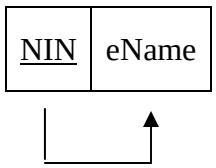
NIN	contractNo	hours

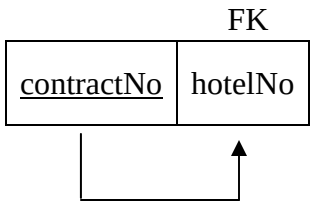
NIN	eName

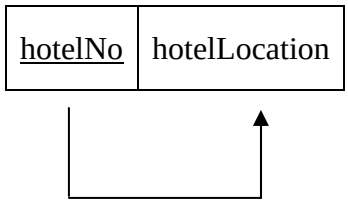
contractNo	hotelNo	hotelLocation

In 3NF we remove the transitive dependency of the FD4 by splitting it into new table as shown below.

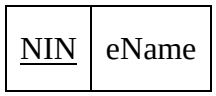




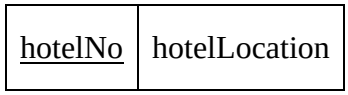




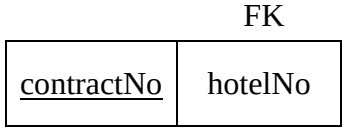
Finally, let re-arrange the table and assign names.



Staff(NIN, eName)



Hotel(hotelNo, hotelLocation)



Contract(contractNo, hotelNo)

FK	FK	
NIN	contractNo	hours

WorkHours(NIN, contractNo, hours)