

The implementation of the database is quite straightforward. Simply create up a new database, import the dentist.sql file and the table, views and data should all be set up. The database consists of seven tables: appointment, bill, charge, patient, payment, referral, and specialist. The intersection of each row and column within the database contains one scalar value only, every non-primary key attribute is fully functionally dependent on the primary key, and no non-primary key attribute is transitively dependent on the primary key, thus satisfying the requirements for the database's relations to be in first, second, and third normal form.

The appointment table contains six attributes: appointment\_ID (the appointment's unique ID number), patient\_ID (the patient attending the appointment), time (the time of the appointment (displayed in hour:minute:second format)), date (the date of the appointment (displayed in year:month:date format)), charge\_name (the treatment which will occur during the appointment), and mode\_of\_booking (how the appointment was made (by phone, in person, etc.)). The primary key is appointment\_ID, and patient\_ID and charge\_name are foreign keys.

The bill table contains four attributes: bill\_ID (the bill's unique ID number), patient\_ID (the patient associated with the bill), charge\_name (the treatment for which the patient is being billed), and payment\_due\_date (the date by which the bill ought to be paid). The primary key is bill\_ID, and patient\_ID and charge\_name are foreign keys.

The charge table contains two attributes: charge\_name and fee. All charges which a patient might face are included in the charge\_name column. Almost all of these charges refer to treatment services offered at the dentist. The only exception to this is the late cancellation charge. As late cancellations are accompanied by a late cancellation fee, it also belongs in the charge table. The primary key is charge\_name. There are no foreign keys.

The patient table contains five attributes: patient\_ID, first\_name, last\_name, address, and phone\_number. Respectively, these refer to the patient's unique ID number, their first name, last name, address, and phone number. The primary key is patient\_ID. There are no foreign keys.

The payment table contains ten attributes: payment\_ID (the unique ID number for each payment transaction made), patient\_ID (the patient making the payment), bill\_ID (the bill being paid), amount\_paid (the value of the payment), outstanding\_fee (how much of the bill remains to be paid. A value of 0 (zero) indicates a fully paid bill), date\_of\_payment (the date when the payment was made), instalment\_number (the number of the pay instalment), total\_number\_of\_instalments (the total number of instalments the bill is being paid in), method\_of\_payment (whether the patient is paying with cash, credit card, etc.), and mode\_of\_payment (whether the patient is paying in person, via post, etc.). The primary key is payment\_ID, and patient\_ID and bill\_ID are foreign keys. It is possible for payments to be made in instalments if the overall cost is €200 or more, otherwise payment is expected to be paid in full via a single transaction.

The referral table contains six attributes: referral\_ID (the unique referral ID number), patient\_ID (the patient being referred), appointment\_ID (the appointment in which the referral is made), reason\_for\_referral (why the patient is being referred), specialist\_practice\_name (to whom the patient is being referred to), and specialist\_report (the specialist's dental report received following the specialist's treatment). After a referral is made, entries into the specialist\_report column would be NULL until a dental report is later received from the specialist. The primary key is referral\_ID, and patient\_ID and appointment\_ID are foreign keys.

The specialist table contains four attributes: specialist\_practice\_name, address, phone\_number, and area\_of\_specialisation. Collectively, these refer to the name of the specialist

dental practice, its address, phone number and area of specialisation. The primary key is `specialist_practice_name`. There are no foreign keys.

The date of January 1<sup>st</sup>, 2023 was chosen to represent the time around which the events represented in the database occur. So upcoming appointments would be from January 1<sup>st</sup> and beyond, payments which have been made are dated before this date, and unpaid bills would be due on or around this date.

Appointments can be entered into the appointment table as they come in, with the secretary able to refer to the patient table to see if they are making their first visit. If a potential patient's details aren't in the patient table, it's safe to assume that they are making their first visit to this dentist and should have their details entered into the patient table. The appointment table should then be consulted to see which appointment slots might be available before entering in their appointment to the appointment table. Similarly, if a patient needs to arrange a follow-up treatment after their appointment, the appointment table can again be consulted for available slots and then entered into the table.

Should a patient wish to rearrange an appointment time and/or date, the appointment table can be consulted to identify available slots. Patients are also free to cancel their appointment free of charge as long as sufficient notice is given (48 hours or more before the appointment). Otherwise, a late cancellation fee is applied and an entry is made in the bill table documenting the late cancellation fee. In either case, the cancelled appointment can simply be deleted from the appointment table.

The charge table can be updated at any time should the dentist wish to make any amendments to the treatments offered or alter treatment prices.

A number of views have also been created to aid in the running of the business. The `excessive_outstanding_fee` view allows the secretary to see the ID, name and amount owed by patients owing €200 or more in unpaid treatment bills. These patients should not be given further appointments until they no longer owe over this amount.

Similarly, the `overdue_payment` view allows the secretary to see which patients have had unpaid bills for too long, regardless of amount owed. Again, these patients will not be allowed to make further appointments until their bills are paid. This view makes use of the `CURDATE` function to compare the current date to the bill's payment due date. As the `CURDATE` function always reflects the actual current date, it will consistently compare the current date with the payment due dates without any need for the secretary to update the view.

The `unpaid_bill` view is there to assist the secretary when sending out the weekly bills. This view contains patient details, bill details and the amount of money owed by any patient who has an unpaid or partially unpaid bill. It should be noted that the column displaying the money owed by the patient depends on whether they've made a payment towards their bill or not. The `outstanding_fee` column will display the amount owed if the patient has a partially paid bill. If the patient has a wholly unpaid bill, the `outstanding_fee` column will display `NULL` and the amount owed should be derived from the `fee` column. The criteria used when creating the view ensure that patients with fully paid bills will not appear in this view.

Finally, the `weekly_appointments` view allows the secretary to see all the upcoming appointments for the coming week. The view contains the patient, appointment, and treatment details necessary so that the secretary has everything at hand for compiling and sending out the weekly reminders and appointment cards to all the patients for the next week.