

The Sample Database

The sample database has two entities, APPOINTMENT and STYLIST. However, it must also store some reference data, which is a list of those times during the day at which appointments can be made.

You will see from the relationship diagram that STYLIST has only one attribute *StylistName*. Since this single attribute must become the primary key, this assumes that no two stylists have the same name. In a larger organisation two attributes would be needed – *StylistID* and *StylistName*.

In this sample solution, customer data is not stored separately. In many situations customers or clients would be 'regulars' and CUSTOMER would be required as an entity. In this case a customer identifier such as *CustomerID* or *AccountNo* would replace Customer in APPOINTMENT and there would be a one-to-many relationship between CUSTOMER and APPOINTMENT.

The database has been implemented as *dbSolution22.accdb* with the entities realised as *tblAppointments* and *tblStylists*.

The reference data is stored in *tblTimes*. This table, as you may see from the screenshot, is simply a list of times during the day at which appointments may be made.

The database contains a form, *frmMakeAppointment*, which is used to make new appointments.

Making Appointments

Once a date and stylist have been selected, we want the drop down list of times that shows only available appointment times for that stylist on the chosen date.

When a new appointment is being made the user may need to see a timetable for the day showing appointments already made and 'slots' that are still available.

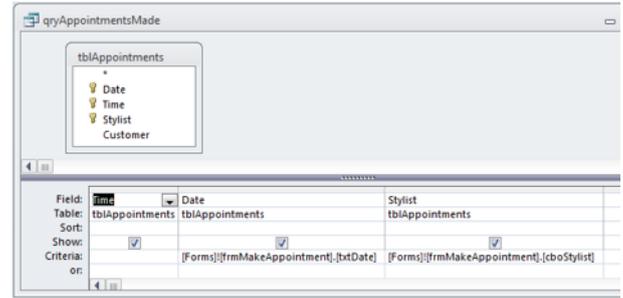
Clicking the Appointments button brings up a list of all the appointments for the date concerned.

When a date and stylist have been selected the drop-down box of times gives the available appointment times for the stylist on that date.

Another stylist has been chosen on the same date. This produces a revised list of available appointments showing the times that this stylist is available on the chosen date.

The Method

Two queries are used to solve the problem of putting the list of times in the combo box. The first, *qryAppointmentsMade*, provides a list of appointment times for the date and time that match the date and time that have been entered on *frmMakeAppointment* (the form used to enter new appointments).



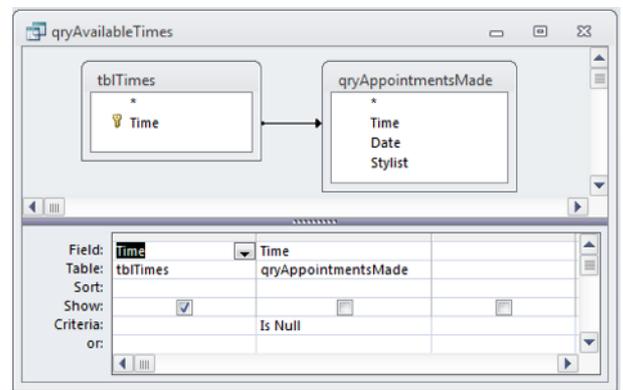
The criterion for the *Date* field is  
Forms!frmMakeAppointment.txtDate

Similarly the criterion for *Stylist* is:  
Forms!frmMakeAppointment.txtStylist

ensuring that the records displayed are further limited to those where the stylist matches the stylist entered on *frmMakeAppointment*.

This ensures that the query shows data only where the date and stylist match the values that have been entered on the form.

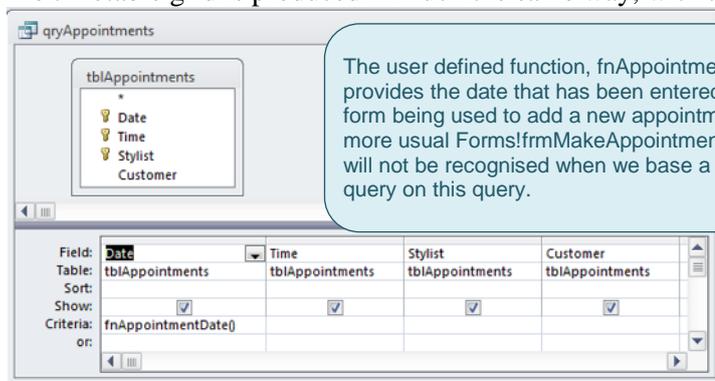
The second query, *qryAvailableTimes* combines data from *tblTimes* and *qryAppointmentsMade* to produce a list of times that are in the table of possible appointment times but not in *qryAppointmentsMade*. In other words *qryAvailableTimes* produces a list of times for which no appointment has been made with the selected stylist on the particular date.



The join type has been changed so that the query shows all the records from *tblTimes* and any matching records from *qryAppointments*. This means that there would be a row for each time, irrespective of whether or not there was a matching appointment. The *Is Null* criterion then reduces the rows to show only the times where there is no matching record in *qryAppointments* so we are left with a set of times for which there is currently no appointment and for which appointment could therefore be made.

This second query, *qryAvailableTimes*, is used to provide the list of times in the combo box. The combo box must be disabled unless a date and stylist have been entered since, without this information, a list of available times would be meaningless. If the date or stylist is changed then the list of times will have to be updated too.

The timetable grid is produced in much the same way, with two queries.



The first query, *qryAppointments* (shown on the left), produces the list of appointments. The second query *qryAppointmentGrid* is a crosstab query to produce the list of possible times and to show, for each time, which customer is booked for which stylist.

There is however a problem. We cannot use the expression Forms!frmMakeAppointment.txtDate in a crosstab query or in any other query that will be used as the basis of the crosstab query. If we do then we will get an error reporting that the expression cannot be evaluated – even though it works perfectly well for a select query.