SELECT EXAMPLES

```
SELECT PATIENT.firstName, patient.surName, patient.contactNumber,
appoinmentdiary.appointmentID,
appoinmentdiary.paymentReceived,
appoinmentdiary.appointmentDateTime,
appoinmentdiary.cancellation,
appointmentdiary.latefee
FROM patient left join appoinmentdiary
ON patient.patient_ID=appoinmentdiary.patient_ID
WHERE appoinmentdiary.paymentReceived = 'No'
ORDER BY appoinmentdiary.appointmentDateTime DESC
```

This command allows for the user to list the patient details and contact number in order to contact patients who have not payed for previous appointments. The user can also see if the appointment was cancelled - in which case no payment would be expected(aside from a possible late fee). In addition, the list is ordered by descending dates so that future appointments that are not yet expecting payments are distinguishable from past appointments.

```
SELECT patient_ID, appointmentID, lateFee from appoinmentdiary WHERE lateFee > 0
```

This command could be used to keep track of guilty culprits who have consistently late cancelled on their appointments. This could be eventually used to restrict patient visitation if persistant cancellations occur.

```
SELECT visitCardFiled, patient_ID, cancellation
FROM appoinmentdiary WHERE visitCardFiled = 'No'
AND appointmentDateTime <= NOW()
```

This command would allow the user to check if visit cards up to the previous day have been filed.
Further investigation as to why can then be instigated.

INSERT EXAMPLE

```
INSERT INTO `patient`(`firstName`, `surName`, `contactNumber`, `address`,
`overduePaymentSpecify`, `treatmentIDHistory`)
VALUES ('Daniel', 'Murray', '+353877707678', 'Rathdown Road, Dublin 7', 'N/A', 'N/A');
```

To add a new patient the database. Part two includes more examples of this command.

DELETE EXAMPLE
DELETE FROM patient WHERE patient_ID = 10006
This deletes a patient who may no longer be with the practice etc.

UPDATE EXAMPLE
UPDATE treatmentandfees
SET treatmentFee = treatmentFee + 10
WHERE treatmentTypeID = 100

This increases the price of the treatmentID 100 by 10.

CREATE EXAMPLE
CREATE view newView AS SELECT treatmentandfees.treatmentTypeID, treatmentandfees.treatmentType, specialist.specialistID, specialist.specialistName FROM treatmentandfees left JOIN specialist ON treatmentandfees.treatmentTypeID=specialist.treatmentTypeID WHERE treatmentandfees.specialistRequired = 'Yes'

This creates a view that lists the treatments and associated specialist. If future specialists are added, this table will have a list of useful info for dental staff to know