SUMMARY

1. APPOINTMENTS.

1.1 - Select appointments on a specific date.

1.2 - Select appointments canceled.

1.3 - Select all Patient Name with appointments on a specific date.

1.4 - Select Appointments in a specific data and time.

1.5 - Select all Appointments for a specific patient.

1.6 - Select appointment related to specialist name.

1.7 - Select appointments for an specialist ordered by time.

1.8 - Select Number of appointments per specialist.

2. BILLS and BILL in array.

2.1 - Select Bill by Patient' Name.

2.2 - See Bills in Array.

2.3 - See Bills in Array older than date given.

2.4 - See Bills OutStanding group by name.

3. CREATE NEW.

3.1 - Create new Patient.

3.2 - Create new appointment.

3.3 - Create new BILL.

3.4 - Create new PAYMENT.

3.5 - Create new TREATMENT.

4. UPDATE.

4.1 - Update tables.

4.2 - Update view.

5. PAYMENTS.

5.1 - Select payments by patient.

5.2 - See All Payment made by a Patient

5.3 - Select payments by bill ID.

5.4 - Select all payments by Patient name.

6. VIEWS

6.1 - Select all patient bills and payments.

6.2 - Select all patient appointments.

7. INSTRUCTIONS.

1. QUERIES RELATED TO APPOINTMENTS.

1.1 - Select appointments on a specific date.

```
Select * from appointment where Appointment_date = '2018-05-30'
```

1.2 - Select appointments canceled

```
Select * from APPOINTMENT where Treatment_code = 'CC0001'
```

1.3 - Select all Patient Name with appointments on a specific date:

```
Select PatientAppointments.First_name, PatientAppointments.Surname,
PatientAppointments.Appointment_Date, PatientAppointments.Appointment_time
        from PatientAppointments
                where Appointment_Date =  '2018-05-30'
```

1.4 - Select Appointments in a specific data and time.

```
Select PatientAppointments.First_name, PatientAppointments.Surname,
PatientAppointments.Appointment_Date, PatientAppointments.Appointment_time

        from PatientAppointments

                where Appointment_Date =  '2018-05-30' and Appointment_time = '13:00:00'
```

1.5 - Select all Appointments for a specific patient.

```
Select PatientAppointments.First_name, PatientAppointments.Surname,
PatientAppointments.Appointment_Date, PatientAppointments.Appointment_time

        from PatientAppointments

                where First_name = 'Thiago' and Surname = 'Dias'
```

1.6 - Select appointment related to specialist name.

```
Select APPOINTMENT.Appointment_ID, APPOINTMENT.Appointment_Date,
APPOINTMENT.Appointment_time, APPOINTMENT.Treatment_code, SPECIALIST.Specialist_Name

        from APPOINTMENT

                right join SPECIALIST on APPOINTMENT.Specialist_ID = SPECIALIST.Specialist_ID;
```

1.7 - Select appointments for an specialist ordered by time.

```
Select APPOINTMENT.Appointment_ID, APPOINTMENT.Appointment_Date,
APPOINTMENT.Appointment_time, APPOINTMENT.Treatment_code, SPECIALIST.Specialist_Name

        from APPOINTMENT

                right join SPECIALIST on APPOINTMENT.Specialist_ID = SPECIALIST.Specialist_ID

                        Where SPECIALIST.Specialist_Name = 'Urologist'

                                Order by Appointment_time;
```

1.8 - Select quantity of procedure per specialist.

```
Select APPOINTMENT.Appointment_ID, APPOINTMENT.Specialist_ID, SPECIALIST.Specialist_Name, APPOINTMENT.Treatment_code, COUNT(APPOINTMENT.Treatment_code) as Quantity, TREATMENT.Treament_description

        from APPOINTMENT

                right join SPECIALIST on APPOINTMENT.Specialist_ID = SPECIALIST.Specialist_ID

                        inner join TREATMENT on APPOINTMENT.Treatment_code = TREATMENT.Treatment_code

                        WHERE APPOINTMENT.Treatment_code <> 'CC0001'

                        Group by Specialist_Name
```

2.1 - Select Bill by Patient' Name

```
SELECT BILL_ID, Bill_Date, Total_due, SUM(amount_paid) FROM `patientbills` Where First_name = 'Thiago' and Surname = 'Dias'
```

2.2 - See Bills in Array

```
Select First_name, Surname, BILL_ID, Bill_Date, Total_due - SUM(amount_paid) as OutStanding from PatientBills

group by Bill_ID
```

2.3 - See Bills in Array older than date given.

```
Select First_name, Surname, BILL_ID, Bill_Date, Total_due - SUM(amount_paid) as OutStanding from PatientBills

WHERE (Bill_Date < Date '2018-05-01') and Patientbills.Total_due - Patientbills.Amount_paid >0 group by Bill_ID
```

2.4 - See Bills OutStanding group by name.

```
Select First_name, Surname, BILL_ID, Bill_Date, Total_due - SUM(amount_paid) as OutStanding from PatientBills

WHERE Patientbills.Total_due - Patientbills.Amount_paid >0

group by first_name
```

3.1 - Create new Patient.

//Auto increment used for Patient_ID.

Enter all information for all the attribute as example below on the SQL command line.

```
INSERT INTO PATIENT (First_name, Surname , Gender ,Data_of_birth, Address, County, Phone_Number)

Values

('Theo','Marquezin','M', '26-03-2016','Ballybrack road', 'Kildare', '089993726');
```

3.2 - Create new appointment.

//Auto increment used for Appointment_ID.

Press Ctrl+F and enter 'FIRST()' to find instructions.

```
INSERT INTO PATIENT (Appointment_date, Appointment_time, Patient_ID, Appointment_made_by, Treatment_code, Specialist_ID)

Values

('2018-05-04', '09:00:00', '00002','Drop in', 'UL6066', '5555');
```

3.3 - Create new BILL.

//Auto increment used for Bill_ID.

Press Ctrl+F and enter 'SECOND()' to find instructions.

INSERT INTO BILL (Appointment_ID, Patient_ID, Treatment_code, Fees, Total_due)

Values

('3','3', 'CI0120', '120','120');

3.4 - Create new PAYMENT.

//Auto increment used for Payment_ID.

Press Ctrl+F and enter 'THIRD()' to find instructions.

INSERT INTO PAYMENT (Bill_ID, Patient_ID, Payment_Date, Amount_paid, Payment_type)

Values

('1', '1','2018-05-02', '30', 'VISA');

3.5 - Create new TREATMENT.

//Enter all information for all the attribute as example below on the SQL command line.

//Treatment_code is a PRIMARY KEY. Format use is; initial letter of the procedure+0+Fee, but mandatory.

INSERT INTO TREATMENT (Treatment_code, Treament_description, Fees)

Values

('FL0050', 'FLUORIDE', '50');

4.1 - Update tables.

UPDATE 'table you wish to update'

SET 'Attribute name that you wish to update' = 'tuple value that you wish to update'

```
WHERE 'Attribute_ID of the table' = 'Value of the tuple you wish to update';
```

EXAMPLE

```
UPDATE PATIENT

SET First_name = 'Pedro'

WHERE Patient_ID = 1;
```

5.1 - Select payments by patient ID.

```
SELECT PatientBills.First_name, PatientBills.Surname, PatientBills.BILL_ID, PatientBills.Bill_Date,
PatientBills.Total_due, PatientBills.Payment_ID, PatientBills.Payment_Date,
PatientBills.Amount_paid FROM PatientBills

WHERE Patient_ID = '1';
```

5.2 - See All Payment made by Patient name

```
SELECT First_name, Surname, BILL_ID, Bill_Date, Total_due, amount_paid

FROM `patientbills` Where First_name = 'Grainne' and Surname = 'OConnor'
```

5.3 - Select payments by bill ID.

```
SELECT PatientBills.First_name, PatientBills.Surname, PatientBills.Bill_Date, PatientBills.Total_due,
PatientBills.Payment_ID, PatientBills.Payment_Date, PatientBills.Amount_paid FROM PatientBills

WHERE BILL_ID = '1'
```

5.4 - Select all payments by Patient name.

```
SELECT PatientBills. Patient_ID, PatientBills.First_name, PatientBills.Surname, PatientBills.BILL_ID,
PatientBills.Bill_Date, PatientBills.Total_due, PatientBills.Payment_ID, PatientBills.Payment_Date,
PatientBills.Amount_paid FROM PatientBills

Order by First_name;
```

6.1 - Select all patient bills and payments.

```
SELECT * FROM PatientBills
```

6.2 - Select all patient appointments.

```
SELECT * FROM PatientAppointments
```

7. INSTRUCTIONS.

Open file project_Database.txt copy all and paste into the SQL command line.

NOTE THAT INSERT INTO FOR TABLES APPOINTMENT, BILL AND PAYMENTS HAS TO BE INSERTED AFTER TABLES PATIENT, TREATMENT AND SPECIALIST AS PRIMARY KEY VALUES WILL BE AUTO INCREMENTED BY THE SYSTEM. THIS VALUES HAVE TO BE ENTERED IN THE INSERTED MANUALLY AS BELOW TO ENFORCE CONSISTENCY.

Insert Patient_ID, Appointment_ID and BILL_ID as per tables PATIENT, APPOINTMENT and BILL accordingly. SEE INSTRUCTIONS BELOW.

FIRST(), Select*from PATIENT and use the Patient_ID to fill the third attribute on the correspond tuples.

After Patient_ID is correctly entered on the tuples, copy INSERT INTO APPOINTMENT including all the tuples on the SQL command line and press Ctrl+Enter.

SEE EXAMPLE BELOW:

```
INSERT INTO APPOINTMENT (Appointment_date, Appointment_time, Patient_ID,
Appointment_made_via, Treatment_code, Specialist_ID)

Values

('2018-03-30', '13:00:00', '1','Drop in', 'CC0001', '9999'),

('2018-03-29', '13:30:00', '2','Phone', 'PR0100', '6666'),
('2018-02-28', '14:30:00', '3','Post', 'PE0120', '8888'),

('2018-04-24', '17:30:00', '4','Post', 'RC0200', '5555'),

('2018-04-31', '09:30:00', '5', 'Phone', 'FI0080', '9999'),

('2018-04-31', '13:00:00', '5', 'Phone','WE0300', '7777'),

('2018-04-15', '09:00:00', '6', 'Drop in','XR0050','9999');
```

SECOND() Repeat FIRST() procedure for the table BILL, however, this time Select*from APPOINTMENT use the Patient_ID and Treatment_code to fill out the third and fourth attribute on the correspond tuples.

In case there is a discount Total_due should be less than Fees.

```
INSERT INTO BILL (Bill_Date, Appointment_ID, Patient_ID, Treatment_code, Fees, Total_due)

Values

('2018-03-30','1', '1', 'CC0001', '10','10'),

('2018-03-29','2', '2', 'PR0100', '100','100'),

('2018-02-28','3', '3', 'PE0120', '120','120'),

('2018-04-24','4', '4', 'RC0200', '200','200'),

('2018-04-31','5', '5', 'FI0080', '080','080'),

('2018-04-15','6', '5', 'WE0300', '300','300');
```

THIRD ()Repeat FIRST() again for table PAYMENT, however, using Select*from BILL so that you can use the BILL_ID and Patient_ID to fill out the first and second attribute on the correspond tuples.

Patients can make more than one payment for the same bill.

```
INSERT INTO PAYMENT (Bill_ID, Patient_ID, Payment_Date, Amount_paid, Payment_type)

Values

('1', '1','2018-05-02', '30', 'VISA'),

('2', '2','2018-05-01', '100', 'VISA'),

('3', '3','2018-04-30', '60', 'CASH'),
('4', '4','2018-04-01', '200', 'MASTER'),
('5', '5','2018-05-30', '50', 'MASTER'),
('6', '6','2018-04-20', '100', 'CHEQUE'),
('6', '6','2018-04-30', '100', 'CHEQUE'),
('1', '1','2018-05-01', '20', 'VISA'),
('2', '2','2018-05-01', '60','CHEQUE');
```