

INFO503 ASSESSMENT

Part B

Melissa Wood 18478786

Question 1: Write a query (or queries) to show the number of entries in each table in the database

/* Comment: For this query it used the count clause to count the number of entries in the selected table, this was repeated for each table. */

```
SELECT COUNT(*)  
FROM Staff
```

```
SELECT COUNT(*)  
FROM Patient
```

```
SELECT COUNT(*)  
FROM Speciality
```

```
SELECT COUNT(*)  
FROM Charges
```

```
SELECT COUNT(*)  
FROM Consultation
```

```
SELECT COUNT(*)  
FROM Staff_Speciality
```

Question 2: Write a query to show all information of staff in the STAFF table that started before 23-Jan-2017

/* Comment: In this query it used the Select all clause to select all entries from the staff table. Using the where clause to specify the column DateJoined before (<) the specified date */

```
SELECT*  
FROM Staff
```

```
WHERE DateJoined < '2017-01-23'
```

Question 3: Write a query to show the staff in the STAFF table that have left.
Show all columns

```
/* Comment: The query used the select all clause to show all columns, retrieved the information from  
the staff table and used the Where clause to specify the DateLeft is not null (this means it has an entry in  
the dateleft column) */
```

```
SELECT *
```

```
FROM Staff s
```

```
WHERE s.DateLeft is NOT NULL
```

Question 4: Write a query to show all details of female staff in the STAFF table
that have left

```
/* Comment: For this query it used the select all statement to show all columns, retrieved the  
information from the staff table and used the Where clause to specify that gender is female AND the  
DateLeft is not null (this means it has an entry in the dateleft column) */
```

```
SELECT *
```

```
FROM Staff
```

```
WHERE Gender = 'F' AND DateLeft is NOT NULL
```

Question 5: Write a query to show the charge code, description and hourly rate of
all charges for casual patients (in the description).

```
/* Comment: For this query it used the select statement to choose the required fields and renamed  
them, these fields were all taken from the charges table. Used the "where like" clause to specify that  
only the charges with the word "Casual" appear to show all the charges specifically for casual patients.  
*/
```

```
SELECT ch.ChargeID AS "Charge Code", ch.ChargeDesc AS "Description", ch.HourlyRate AS "Hourly Rate"
```

```
FROM Charges ch
```

```
WHERE ChargeDesc LIKE '%casual%'
```

Question 6: Write a query to show the charge code, description and hourly rate of all charges for all charges that does NOT have casual patients (in description)

/* Comment: For this query it used the select statement to choose the required fields and renamed them, these fields were all taken from the charges table. Used the "where not like" clause to specify that only the charges with the word "Casual" do not appear to show all the charges specifically for not casual patients. */

```
SELECT ch.ChargeID AS "Charge Code", ch.ChargeDesc AS "Description", ch.HourlyRate AS "Hourly Rate"
FROM Charges ch
WHERE ChargeDesc NOT LIKE '%casual%'
```

Question 7: Write a query to show the highest hourly rate, the lowest hourly rate, and the average hourly rate for the charges applicable. Name the columns highest rate, lowest rate, average rate respectively. Format all the columns to show the dollar sign and 2 decimal places (\$99.99).

/* Comment: Using the select statement to choose the required fields and formatted them using the aggregate functions Max, Min and avg to get the max, min and avg of the charges hourly rate, this is formatted as US currency. These fields are taken from the charges table. */

```
SELECT
FORMAT(MAX(HourlyRate),'C','en-us') AS "Highest Rate",
FORMAT(MIN(HourlyRate),'C','en-us') AS "Lowest Rate",
FORMAT(AVG(HourlyRate),'C','en-us') AS "Average Rate"
FROM Charges
```

Question 8: Write a query to show the duration rate for each charge. Show the charge code, hourly rate, duration, and duration rate. Duration rate will be calculated ($\text{Duration Rate} = \text{Hourly Rate} * \text{Duration} / 60$). Format the Duration rate column to show the dollar sign and 2 decimal places.

/* Comment: For this query it selects the required fields and renames them to look better, it then formats the duration rate by multiplying the hourly rate by the duration and dividing it by 60 (minutes), this is formatted as US currency. All this information is retrieved from the charges table. */

```

SELECT ch.ChargeID AS "Charge Code", ch.HourlyRate AS "Hourly Rate", ch.Duration AS "Duration",
FORMAT((ch.HourlyRate * ch.Duration/60), 'C', 'en-us') AS "Duration Rate"
FROM Charges ch

```

Question 9: Write a query to show the StaffID, SpecialityID, Date Qualified, Valid Till Date and Days Valid. Days Valid will be calculated using the Date Qualified and Valid Till Date.

Comment: For this query it selects the required fields and renames them to look better, it then uses DATEDIFF function by day to calculate the number of days valid between date qualified and valid till date. All this is taken from the staff speciality table.

```

SELECT ss.StaffID AS "Staff ID", ss.SpecialityID AS "Speciality ID", ss.DateQualified AS "Date Qualified",
ss.ValidTillDate AS "Valid Till Date",
DATEDIFF (day, DateQualified, ValidTillDate) AS "Days Valid"
FROM Staff_Speciality ss

```

Question 10: All charges are due within 21 days after the patient has had the consultation (Date Consulted). Write a query to show the StaffID, Charge Code, Date Consulted, Start Time and Hourly Rate for all consultations.

/* Comment: For this query it selects the required fields and renames them to look better, all this information was taken from the consultation and charges table. It uses a where clause to join the two tables together */

```

SELECT c.StaffID AS "Staff ID", c.ChargeID AS "Charge Code", c.DateConsulted AS "Date Consulted",
c.StartTime AS "Start Time", ch.HourlyRate AS "Hourly Rate"
FROM Consultation c, Charges ch
WHERE ch.ChargeID = c.ChargeID

```

Question 11: Write a query to show the StaffID, SpecialityID, Speciality Name, Date Qualified, and Valid Till Date for all consultations.

/* Comment: For this query it selects the required fields and renames them to look better, it gets all this information from the staff specialty and speciality table. It specifies in the where clause that the two tables are joined. */

```

SELECT ss.StaffID AS "Staff ID", sp.SpecialityID AS "Speciality ID", sp.SpecName AS "Speciality Name",
ss.DateQualified AS "Date Qualified", ss.ValidTillDate AS "Valid Till Date"

FROM Staff_Speciality ss, Speciality sp

WHERE ss.SpecialityID = sp.SpecialityID

```

Question 12: Write a query to show the StaffID, Patient Num, Date Consulted, Start Time, and Hourly Rate for all consultations.

/ Comment: For this query it selects the required fields and renames them to look better, it gets all this information from the consultation and charges table. The where clause joins these two tables together. */*

```

SELECT c.StaffID AS "Staff ID", c.PatientID AS "Patient Num", c.DateConsulted AS "Date Consulted",
c.StartTime AS "Start Time", ch.HourlyRate AS "Hourly Rate"

FROM Consultation c, Charges ch

WHERE ch.ChargeID = c.ChargeID

```

Question 13: Write a query to show the StaffID, Staff Full Name and SpecialityID of all male staff.

/ Comment: For this query it selects the required fields and renames them to look better, including joining the staff first name and last name together in one column. This information is taken from the staff and Staff specialty table. The where clause joins the two tables together AND specifies that the gender is Male */*

```

SELECT s.StaffID AS "Staff ID", s.Staff_FirstName + ' ' + s.Staff_LastName AS "Staff Full Name",
ss.SpecialityID AS "Speciality ID"

FROM Staff s, Staff_Speciality ss

WHERE s.StaffID = ss.StaffID

AND s.Gender = 'M'

```

Question 14: Write a query to show the StaffID, Speciality Name, Speciality Notes of all staff which relate to surgery or surgeon. Name this column Details. (Hint: look up for Surge or surge)

/* Comment: For this query it selects the required fields and renames them to look better, all this information is from the staff speciality and speciality table. The where clause joins these tables together AND specifies that the specnotes include the word Surge OR surge. */

```
SELECT ss.StaffID AS "Staff ID", sp.SpecName AS "Speciality Name", sp.SpecNotes AS "Details"
FROM Staff_Speciality ss, Speciality sp
WHERE ss.SpecialityID= sp.SpecialityID and sp.SpecNotes LIKE '%Surge%'
OR sp.SpecNotes LIKE '%surge%'
```

Question 15: Write a query to show the StaffID, Patient Num, Charge Description, Time for all consultations after 9.45, in descending order of the StaffID.

/* Comment: For this query it selects the required fields and renames them to look better, it gets all this information from the consultation and charges table. The where clause joins the two tables AND specifies that the start time is later than 9.45, it is ordered by the staff ID in descending order. */

```
SELECT c.StaffID AS "Staff ID", c.PatientID AS "Patient Num", ch.ChargeDesc AS "Charge Description",
c.StartTime AS "Time"
FROM Consultation c, Charges ch
WHERE ch.ChargeID = c.ChargeID AND c.StartTime > '09.45'
ORDER BY c.StaffID DESC
```

Question 16: Write a query to show the StaffID, Number of Consultations. Name the new column Num of Consults.

/* Comment: For this query it selects the required fields, renames them to look better and counts the number of consults, it gets all this information from the consultation table and is grouped by staff id to show the total amount for each staff. */

```
SELECT StaffID AS "Staff ID", COUNT(*) AS "Num of Consults"
FROM Consultation
GROUP BY StaffID
```

Question 17: Write a query to show the StaffID, Num of Consults only for staff that have more than 2 consults.

/* Comment: For this query it selects the required fields, renames them to look better and counts the number of consults, it gets all this information from the consultation table. It is grouped by staff id to

show the total amount for each staff and the having clause so it only shows the rows that meet the condition (greater than 2). */

```
SELECT c.StaffID AS "Staff ID", COUNT(*) AS "Num of Consults"
FROM Consultation c
GROUP BY c.StaffID
HAVING COUNT(*) > 2
```

Question 18: Write a query to show the SpecialityID, Num With Speciality (i.e. the number of staff with that speciality). Display in descending order of num with speciality.

/* Comment: For this query it selects the required fields, renames them to look better and counts the number of specialties, it gets all this information from the staff speciality table. It is grouped by speciality id to show the total amount for each staff and is ordered by number with specialties in descending order. */

```
SELECT ss.SpecialityID AS "Specialty ID", COUNT(*) AS "Num With Speciality"
FROM Staff_Speciality ss
GROUP BY ss.SpecialityID
ORDER BY "Num With Speciality" DESC
```

Question 19: Write a query to show the SpecialityID, Num with Speciality with specialties having more than two staff. Display in descending order of num with speciality.

/* Comment: For this query it selects the required fields, renames them to look better and counts the amount of specialties, it gets all this information from the staff speciality table. It is grouped by speciality id to show the total amount for each speciality and the having clause so it only shows the rows that meet the condition (greater than 2). It is ordered by num with speciality in descending order. */

```
SELECT ss.SpecialityID AS "Specialty ID", COUNT(*) AS "Num With Speciality"
FROM Staff_Speciality ss
GROUP BY ss.SpecialityID
HAVING COUNT(*) > 2
```


ORDER BY "Num With Speciality" DESC

Question 20: Write a query to show the Staff Full Name, Num of Specialities. Sort in ascending order by Staff Full Name.

/* Comment: For this query it selects the first name and last name and joins them into one row as Staff full name and counts the number of specialities, this information is from the staff and staff speciality tables. The where clause joins the two tables together and groups them by staff full name, it displays it in order by full name ascending. */

```
SELECT s.Staff_FirstName+' '+s.Staff_LastName AS "Staff Full Name", COUNT(*) AS "Num of Specialities"
FROM Staff s, Staff_Speciality ss
WHERE s.StaffID = ss.StaffID
GROUP BY s.Staff_FirstName+' '+ s.Staff_LastName
ORDER BY "Staff Full Name" ASC
```

Question 21: Write a query to show the Staff Full Name, Num. Specialities of all male staff with more than one speciality.

/* Comment: For this query it selects the first name and last name and joins them into one row as Staff full name and counts the number of specialities, this information is from the staff and staff speciality tables. The where clause joins the two tables together and specifies the gender that is to show, the having clause means it will only return rows greater than 1. It groups them by staff full name, it displays it in order by full name ascending. */

```
SELECT s.Staff_FirstName+' '+s.Staff_LastName AS "Staff Full Name", COUNT(*) AS "Num. Specialities"
FROM Staff s, Staff_Speciality ss
WHERE s.StaffID = ss.StaffID AND s.Gender = 'M'
GROUP BY s.Staff_FirstName+' '+ s.Staff_LastName
HAVING COUNT(*) > 1
ORDER BY "Staff Full Name" ASC
```

Question 22: Write a query to show the StaffID, Staff Full Name, SpecialityID, Speciality Name for all staff.

/* Comment: For this query it selects the required fields and renames them to look better, including joining the staff first name and last name together in one column. This information is taken from the staff, speciality and staff speciality tables, and uses the where clause to join staff to speciality and speciality to staff speciality. */

```
SELECT s.StaffID AS "Staff ID", s.Staff_FirstName + ' ' + s.Staff_LastName AS "Staff Full Name",  
sp.SpecialityID AS "Speciality ID", sp.SpecName AS "Speciality Name"
```

```
FROM Staff s, Speciality sp, Staff_Speciality ss
```

```
WHERE s.StaffID = ss.StaffID and sp.SpecialityID = ss.SpecialityID
```

REFERENCES:

Brian Wood (my dad) helped explain any of the queries that I didn't understand and taught me what joining the tables meant.