

DELIVERABLE 3

Group N

PROJECT TOPIC: HEALTHCARE MANAGEMENT SYSTEMS

Group members	CWID	Contribution
Dhanvanth Voona	A20543395	20%
Neeraj Vardhan Buneeti	A20545853	20%
Priyadarshani LNU	A20561249	20%
Sai Charan Gangili	A20543155	20%
Sai Pranay Yada	A20553636	20%

1. What is the average age of all patients, and provide the average age of patients categorised by their gender?

QUERY

SELECT

 'Total' AS Gender,

 AVG(YEAR(NOW()) - YEAR(Patient_DateofBirth)) AS AverageAge

FROM Patient

UNION

SELECT

 Patient_Gender,

 AVG(YEAR(NOW()) - YEAR(Patient_DateofBirth)) AS AverageAge

FROM Patient

GROUP BY Patient_Gender;

OUTPUT

```
328 • SELECT
329     'Total' AS Gender,
330     AVG(YEAR(NOW()) - YEAR(Patient_DateofBirth)) AS AverageAge
331 FROM Patient
332 UNION
333 SELECT
334     Patient_Gender,
335     AVG(YEAR(NOW()) - YEAR(Patient_DateofBirth)) AS AverageAge
336 FROM Patient
337 GROUP BY Patient_Gender;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Gender	AverageAge		
Total	20.6000		
M	22.7500		
F	18.1429		

EXPLANATION

This query is used to calculate average ages of patients categorised by gender. This line of the query is used to calculate the average age: **AVG(YEAR(NOW()) - YEAR(Patient_DateofBirth))**. We calculate total average age and union with average age categorised by gender.

2. Provide a list of different types of lab tests, along with the counts of positive and negative results for each test?

QUERY

```
SELECT

    Labtest_name,

    SUM(CASE WHEN Labtest_results = 'postive' THEN 1 ELSE 0 END)
AS Positive_Count,

    SUM(CASE WHEN Labtest_results = 'negative' THEN 1 ELSE 0 END)
AS Negative_Count

FROM LabTest

GROUP BY Labtest_name;
```

OUTPUT

```
341 • SELECT
342     Labtest_name,
343     SUM(CASE WHEN Labtest_results = 'postive' THEN 1 ELSE 0 END) AS Positive_Count,
344     SUM(CASE WHEN Labtest_results = 'negative' THEN 1 ELSE 0 END) AS Negative_Count
345 FROM LabTest
346 GROUP BY Labtest_name;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Labtest_name	Positive_Count	Negative_Count	
Liver Function Tests (LFTs)	0	2	
Thyroid Function Test (TFT)	2	2	
Electroencephalogram (EEG)	2	0	
Computed Tomography (CT) Scan	0	2	
X-ray	1	0	
Magnetic Resonance Imaging (MRI)	3	1	
Bone Density Scan (DEXA)	0	1	

EXPLANATION

This query represents the total number of lab tests available and number of positive and negative results in each of the following lab tests. Lab test table contains 16 entries in which there are 7 distinct lab tests. **Case** command is used to obtain the required results.

3. Give the names of patients and their id who have not undergone any lab tests yet.

QUERY

```
SELECT P.Patient_firstname, P.Patient_lastname, P.PatientID  
FROM Patient P  
LEFT JOIN LabTest LT ON P.PatientID = LT.PatientID  
WHERE LT.LabtestID IS NULL;
```

OUTPUT

```
310 • SELECT P.Patient_firstname, P.Patient_lastname, P.PatientID  
311 FROM Patient P  
312 LEFT JOIN LabTest LT ON P.PatientID = LT.PatientID  
313 WHERE LT.LabtestID IS NULL;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Patient_firstname	Patient_lastname	PatientID	
Maxwell	Longstreeth	1	
Natala	Shackell	3	
Mora	Harpham	8	
Ab	Macauley	11	
Rafaelita	Minger	12	

EXPLANATION

Left Join connects both Patient and labtest tables using patientId and this query identifies patients who have not undergone any lab tests.

4. Provide the names of the seven patients who have undergone the latest lab tests, along with the corresponding lab test names, results and date?

QUERY

```
SELECT P.Patient_firstname, P.Patient_lastname, LT.Labtest_name,  
LT.Labtest_results, LT.Labtest_date  
FROM Patient P  
JOIN LabTest LT ON P.PatientID = LT.PatientID  
ORDER BY LT.Labtest_date DESC  
LIMIT 7;
```

OUTPUT

```
317 • SELECT P.Patient_firstname, P.Patient_lastname, LT.Labtest_name, LT.Labtest_results, LT.Labtest_date  
318 FROM Patient P  
319 JOIN LabTest LT ON P.PatientID = LT.PatientID  
320 ORDER BY LT.Labtest_date DESC  
321 LIMIT 7;  
322
```

	Patient_firstname	Patient_lastname	Labtest_name	Labtest_results	Labtest_date
►	Mord	Batt	X-ray	postive	2023-09-27
	Tove	Grabham	Electroencephalogram (EEG)	postive	2023-09-13
	Johnathan	Hazlewood	Magnetic Resonance Imaging (MRI)	postive	2023-09-09
	Ysabel	Purselow	Computed Tomography (CT) Scan	negative	2023-07-30
	Ronda	Easterfield	Magnetic Resonance Imaging (MRI)	postive	2023-06-11
	Quillan	Pierrepoint	Electroencephalogram (EEG)	postive	2023-04-16
	Shantee	Mainston	Thyroid Function Test (TFT)	negative	2023-04-04

EXPLANATION

Left Join connects both Patient and lab test tables using patientId. Order by date desc gives the latest performed lab tests and **limit 7** provides the names of the seven patients who have undergone the latest lab tests.

5. What is the count of total appointments made for each medical staff member?

QUERY

```
SELECT MedicalstaffID, COUNT(*) AS TotalAppointments FROM Appointment GROUP BY MedicalstaffID;
```

OUTPUT

358 • `SELECT MedicalstaffID, COUNT(*) AS TotalAppointments FROM Appointment GROUP BY MedicalstaffID;`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	MedicalstaffID	TotalAppointments
▶	401	1
	413	1
	416	1
	425	1
	427	1
	430	1
	446	1
	451	1
	455	1
	465	1
	480	1
	481	1
	488	1
	493	1

EXPLANATION

This query is used to calculate the number of appointments made for each medical staff in total

6. Provide the list of medical staff who specialise in “General Surgery”, including their names, designation and contact information.

QUERY

```
SELECT * FROM Medicalstaff WHERE Medicalstaff_specialty = 'General surgery';
```

OUTPUT

333 • `SELECT * FROM Medicalstaff WHERE Medicalstaff_specialty = 'General surgery';`

MedicalstaffID	Medicalstaff_firstname	Medicalstaff_lastname	Medicalstaff_specialty	Medicalstaff_phonenumber	Medicalstaff_email	Medicalstaff_designation
401	Constancia	Stuchbury	General Surgery	3234516447	cstuchbury2@barnesandnoble.com	surgeons
430	Dare	Insoll	General Surgery	2416098768	dinsoll8@aol.com	surgeons
480	Micaela	Boylan	General Surgery	1685265510	mboylan7@abc.net.au	nurses
NULL	NULL	NULL	NULL	NULL	NULL	NULL

EXPLANATION

This query provides the all the medical staff whose speciality is general surgery

7. Get details of all appointments, including the information of the medical staff associated with each appointment

QUERY

```
SELECT Appointment.*, Medicalstaff.*
FROM Appointment
JOIN Medicalstaff ON Appointment.MedicalstaffID =
Medicalstaff.MedicalstaffID;
```

OUTPUT

320 • `SELECT Appointment.*, Medicalstaff.*`
 321 `FROM Appointment`
 322 `JOIN Medicalstaff ON Appointment.MedicalstaffID = Medicalstaff.MedicalstaffID;`

AppointmentID	PatientID	HospitalID	MedicalstaffID	Appointment_date	Notes	MedicalstaffID	Medicalstaff_firstname	Medicalstaff_lastname	Medicalstaff_specialty	Medicalstaff_phonenumber
503	15	313	425	2019-08-12	Maternal care for disproportion due to outlet co...	425	Bucky	MacAdam	Cardiology	4271884242
518	1	310	413	2017-06-11	Presence of (intrauterine) contraceptive device	413	Kriste	Kohnert	Emergency Medicine	7205840488
521	3	323	427	2023-07-30	Nondisplaced avulsion fracture of left ischium	427	Frederigo	Morde	Emergency Medicine	6122171984
525	10	361	465	1999-08-20	Laceration of unspecified muscles, fascia and te...	465	Eziechiele	Snell	Emergency Medicine	6157216219
532	14	344	455	1923-12-12	Sprain of interphalangeal joint of right great toe	455	Carlotta	Rushton	Cardiology	9338157717
534	8	325	430	1989-07-01	Nondisplaced fracture of posterior column of rig...	430	Dare	Insoll	General Surgery	2416098768
536	12	306	401	2013-02-24	Unilateral primary osteoarthritis of first carpine...	401	Constancia	Stuchbury	General Surgery	3234516447
547	9	357	480	1998-04-17	Corrosion of unspecified degree of unspecified...	480	Micaela	Boylan	General Surgery	1685265510
564	2	386	493	2022-06-19	Unspecified fracture of shaft of unspecified fibula	493	Rennie	Chinnick	Allergy and Immunology	9427967575
578	4	385	488	1947-08-15	Retinopathy of prematurity, stage 5	488	Leicester	Sudell	Plastic Surgery	5189280478
584	7	378	481	1965-03-23	Irritant contact dermatitis due to food in contac...	481	Matthus	Digle	Cardiology	1417577727
588	6	334	446	2020-01-23	Burn of first degree of multiple sites of unspeci...	446	Nydia	Tadman	Anesthesiology	7739657831
594	13	335	451	2021-02-15	Person boarding or alighting a three-wheeled m...	451	Dotty	Ralton	Anesthesiology	7568949499
596	11	311	416	2020-09-09	Nondisplaced osteochondral fracture of right pa...	416	Marchall	Alfonsetti	Plastic Surgery	3733228583

EXPLANATION

This query is about getting the appointment including their appointment information and patient information like appointmentID, patientsID, hospitalID, medicalstaffID, appointment_date, Notes, Meicalstaff_firstname, medicalstaff_lastname, medicalstaff_speciality, medicalstaff_phonenumber

8. Update query:

QUERY

Before update, select * from healthcareprovider;

OUTPUT

150 • `select * from healthcareprovider;`

HealthcareproviderID	Healthcareprovider_name	Healthcareprovider_address	Healthcareprovider_contactno	Healthcareprovider_email
112	Talane	Suite 18	537 560 3520	mazema5@google.es
115	Midel	PO Box 81000	686 590 2388	mharverson4@devhub.com
117	Rhynoodle	Room 1481	471 867 2827	bguillard6@nytimes.com
118	Zooveo	Room 1568	404 449 3090	elissandri0@scribd.com
124	Wikibox	Apt 93	860 256 9031	mpopov3@hatena.ne.jp
128	Katz	19th Floor	571 128 1584	jcrosstonc@usda.gov
130	Eimbee	Apt 1301	107 294 6144	estroobandd@miibeian.gov.cn
133	Aimbu	12th Floor	689 428 3300	abarthe18@prweb.com
135	Avamba	PO Box 88510	464 995 3708	chartfleet7@ucsd.edu
136	Oloo	Apt 1830	505 755 5629	cgreystoke2@livejournal.com
141	Kimia	Suite 16	315 232 3707	aedgeson1@epa.gov
145	Skiptube	Apt 618	734 356 6148	tbuntinea@nyu.edu
147	Linktype	Suite 87	223 146 8810	wgurnee9@state.gov
150	Lazz	Apt 1547	214 148 5879	sgrelakb@amazon.co.jp
NULL	NULL	NULL	NULL	NULL

Added new row row_num to the healthcareprovider table with below query:

QUERY

Alter table healthcareprovider add row_num int;

Update row_num value for all healthcareproviderID with below query:

```
update healthcareprovider set row_num = 12 where HealthcareproviderID = 150;
```

Update Healthcareprovider name as Poloo in table healthcareprovider where Healthcareplan_startdate is less than '2023-08-31'

```
UPDATE healthcareprovider SET Healthcareprovider_name = 'Poloo'
WHERE healthcareproviderID in (
  SELECT healthcareproviderID from healthcareplan where
  Healthcareplan_startdate > '2023-08-31');
```

OUTPUT


```
25 • UPDATE healthcareprovider SET Healthcareprovider_name = 'Poloo'
26 WHERE
27 healthcareproviderID in (
28     SELECT healthcareproviderID from healthcareplan where Healthcareplan_startdate > '2023-08-31'
29 );
30
```

100% 3:29 1 error found

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 258	23:53:51	SELECT healthcareproviderID from healthcareplan where Healthcareplan_startdate > '2023-08-31' LIMIT 0, 1000	1 row(s) returned	0.0047 sec / 0.00001...
✓ 259	23:56:59	UPDATE healthcareprovider SET Healthcareprovider_name = 'Poloo' WHERE healthcareproviderID in (SELECT health...	1 row(s) affected Rows matched: 1 Changed: 1 Warni...	0.016 sec

Updated successfully.

After update, select * from healthcareprovider;

OUTPUT

3

•

select * from healthcareprovider;

4

5

6

100%

32:3

Result Grid

Filter Rows:

Search

Edit:

Export/Import:

HealthcareproviderID	Healthcareprovider_name	Healthcareprovider_addr...	Healthcareprovider_contac...	Healthcareprovider_email	row_num	
112	Talane	Suite 18	537 560 3520	mazema5@google.es	1	
115	Midel	PO Box 81000	686 590 2388	mharverson4@devhub.com	2	
117	Rhynoodle	Room 1481	471 867 2827	bguillard6@nytimes.com	3	
118	Zooveo	Room 1568	404 449 3090	elissandri0@scribd.com	4	
124	Wikibox	Apt 93	860 256 9031	mpopov3@hatena.ne.jp	5	
128	Katz	19th Floor	571 128 1584	jcrosstncc@usda.gov	6	
133	Aimbu	12th Floor	689 428 3300	abarthei8@prweb.com	7	
135	Avamba	PO Box 88510	464 995 3708	chartfleet7@ucsd.edu	0	
136	Poloo	Apt 1830	505 755 5629	cgreystoke2@livejournal.com	8	
141	Kimia	Suite 16	315 232 3707	aedgeson1@epa.gov	9	
145	Skiptube	Apt 618	734 356 6148	tbuntinea@nyu.edu	10	
147	Linktype	Suite 87	223 146 8810	wgurnee9@state.gov	11	
150	Lazz	Apt 1547	214 148 5879	sgrelakb@amazon.co.jp	12	
NULL	NULL	NULL	NULL	NULL	NULL	

EXPLANATION

In this query, first add one new column row_num and insert auto increment value. Then Update Healthcareprovider name as Poloo in table healthcareprovider where Healthcareplan_startdate is less than '2023-08-31'. After successful update verify update in healthcareprovider table and its updated successfully.

9. Write a query to compute the NTILE(4) and return the entire dataset of healthcareprovider showing approximately equal groups/buckets.

QUERY

```
SELECT row_num, healthcareproviderID,  
healthcareprovider_name,healthcareprovider_address,  
healthcareprovider_contactno,  
healthcareprovider_email, NTILE(4) OVER (ORDER BY  
healthcareproviderID) as 'HC_NTILE(4)' FROM healthcareprovider;
```

OUTPUT

```
12 • SELECT row_num, healthcareproviderID, healthcareprovider_name,healthcareprovider_address, healthcareprovider_contactno,  
13 healthcareprovider_email, NTILE(4) OVER (ORDER BY healthcareproviderID) as 'HC_NTILE(4)'  
14 FROM healthcareprovider;  
15
```

100% 25:14 1 error found

Result Grid Filter Rows: Search Export:

row_num	healthcareprovider...	healthcareprovider_na...	healthcareprovider_addr...	healthcareprovider_contac...	healthcareprovider_email	HC_NTILE(4)
1	112	Talane	Suite 18	537 560 3520	mazema5@google.es	1
2	115	Midel	PO Box 81000	686 590 2388	mharverson4@devhub.com	1
3	117	Rhynoodle	Room 1481	471 867 2827	bguillard6@nytimes.com	1
4	118	Zooveo	Room 1568	404 449 3090	elissandri0@scribd.com	1
5	124	Wikibox	Apt 93	860 256 9031	mpopov3@hatena.ne.jp	2
6	128	Katz	19th Floor	571 128 1584	jcrosstonc@usda.gov	2
7	133	Aimbu	12th Floor	689 428 3300	abarthe18@prweb.com	2
0	135	Avamba	PO Box 88510	464 995 3708	chartfleet7@ucsd.edu	3
8	136	Poloo	Apt 1830	505 755 5629	cgreystoke2@livejournal.com	3
9	141	Kimia	Suite 16	315 232 3707	aedgeson1@epa.gov	3
10	145	Skiptube	Apt 618	734 356 6148	tbuntinea@nyu.edu	4
11	147	Linktype	Suite 87	223 146 8810	wgurnee9@state.gov	4
12	150	Lazz	Apt 1547	214 148 5879	sgrelakb@amazon.co.jp	4

EXPLANATION

In this query, perform NTILE() to divide all rows in the partition into the specified number of buckets as evenly as possible, and return the rank of the bucket where the entire dataset is located for the table healthcareprovider.

10. Write a query to compute the CUME_DIST() and return the entire dataset of healthcareprovider Format CUME_DIST() values to 2 decimal places.

QUERY

```
SELECT row_num, healthcareproviderID,  
healthcareprovider_name,healthcareprovider_address,  
healthcareprovider_contactno, healthcareprovider_email,  
ROUND(CUME_DIST() OVER (ORDER BY healthcareproviderID),2) as  
'HC_cume_dist' FROM healthcareprovider;
```

OUTPUT

```
15 • SELECT row_num, healthcareproviderID, healthcareprovider_name,healthcareprovider_address, healthcareprovider_contactno,  
16 healthcareprovider_email,  
17 ROUND(CUME_DIST() OVER (ORDER BY healthcareproviderID),2) as 'HC_cume_dist' FROM healthcareprovider;  
18  
19
```

row_num	healthcareprovider...	healthcareprovider_na...	healthcareprovider_addr...	healthcareprovider_contac...	healthcareprovider_email	HC_cume_dist
1	112	Talane	Suite 18	537 560 3520	mazema5@google.es	0.08
2	115	Midel	PO Box 81000	686 590 2388	mharverson4@devhub.com	0.15
3	117	Rhynoodle	Room 1481	471 867 2827	bguillard6@nytimes.com	0.23
4	118	Zoo veo	Room 1568	404 449 3090	elissandri0@scribd.com	0.31
5	124	Wikibox	Apt 93	860 256 9031	mpopov3@hatena.ne.jp	0.38
6	128	Katz	19th Floor	571 128 1584	jcrosstonc@usda.gov	0.46
7	133	Aimbu	12th Floor	689 428 3300	abarthe18@prweb.com	0.54
0	135	Avamba	PO Box 88510	464 995 3708	chartfleet7@ucsd.edu	0.62
8	136	Poloo	Apt 1830	505 755 5629	cgreystoke2@livejournal.com	0.69
9	141	Kimia	Suite 16	315 232 3707	aedgeson1@epa.gov	0.77
10	145	Skiptube	Apt 618	734 356 6148	tbuntinea@nyu.edu	0.85
11	147	Linktype	Suite 87	223 146 8810	wgurnee9@state.gov	0.92
12	150	Lazz	Apt 1547	214 148 5879	sgrelakb@amazon.co.jp	1

EXPLANATION

In this query, perform CUME_DIST() for the table healthcareprovider to return the cumulative distribution of the entire dataset. Also it should show results with 2 decimal values.

11. Retrieve the top 5 Healthcareplan name with the highest number of HealthcareproviderID.

QUERY

```
SELECT H1.Healthcareplan_name, COUNT(H2.HealthcareproviderID) AS  
total_HealthcareproviderID FROM Healthcareplan H1 LEFT JOIN  
Healthcareprovider H2 ON H1.HealthcareproviderID =  
H2.HealthcareproviderID GROUP BY H1.Healthcareplan_name ORDER BY  
total_HealthcareproviderID DESC LIMIT 5;
```

OUTPUT

```
4 • SELECT H1.Healthcareplan_name, COUNT(H2.HealthcareproviderID) AS total_HealthcareproviderID FROM Healthcareplan H1
5 LEFT JOIN Healthcareprovider H2 ON H1.HealthcareproviderID = H2.HealthcareproviderID
6 GROUP BY H1.Healthcareplan_name
7 ORDER BY total_HealthcareproviderID DESC
8 LIMIT 5;
9
```

100%	36:7	1 error found
Result Grid	Filter Rows: Search	Export: Fetch rows:
Healthcareplan_name	total_Healthcareprovide...	
Goldner Inc	2	
Harber, Kuphal	1	
Tromp Inc	1	
O'Keefe-Cassin	1	
Friesen and Sipes	1	

EXPLANATION

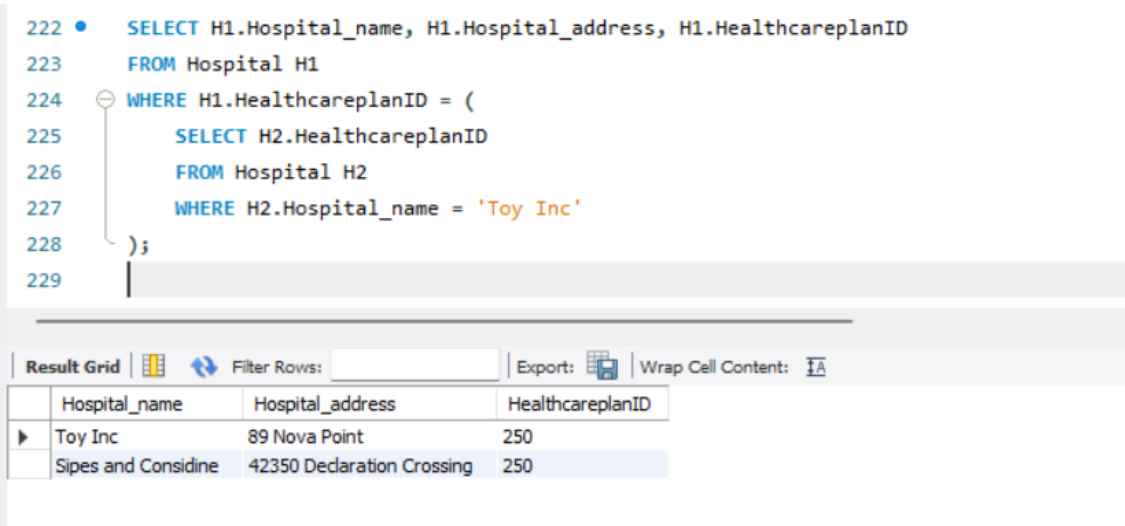
In this Query, Extracting top 5 healthcare plan names with the highest number of HealthcareproviderID.

12. Obtain the healthcare plan ID of one hospital and then use that value to filter other hospitals with the same plan.

QUERY

```
SELECT H1.Hospital_name, H1.Hospital_address, H1.HealthcareplanID
FROM Hospital H1
WHERE H1.HealthcareplanID = (
    SELECT H2.HealthcareplanID
    FROM Hospital H2
    WHERE H2.Hospital_name = 'Toy Inc'
);
```

OUTPUT



The screenshot shows a SQL query editor with the following query:

```
222 • SELECT H1.Hospital_name, H1.Hospital_address, H1.HealthcareplanID
223 FROM Hospital H1
224 WHERE H1.HealthcareplanID = (
225     SELECT H2.HealthcareplanID
226     FROM Hospital H2
227     WHERE H2.Hospital_name = 'Toy Inc'
228 );
229
```

Below the query editor, there is a 'Result Grid' section with a table showing the results of the query. The table has three columns: Hospital_name, Hospital_address, and HealthcareplanID. The results are as follows:

Hospital_name	Hospital_address	HealthcareplanID
Toy Inc	89 Nova Point	250
Sipes and Considine	42350 Declaration Crossing	250

EXPLANATION

Here we obtain the hospitals that have similar healthcare plan by selecting one hospital(Toy Inc) and use its healthcareplanid to find other hospitals which also have the same plan.

13. Selects the "Medication_name" and "Medication_description" for medications prescribed to patients with IDs 6 and 14.

QUERY

```
SELECT Medication_name, Medication_description
FROM Medication
WHERE PatientID IN (6, 14);
```

OUTPUT

```
188 • select * from Medication;
189 • SELECT Medication_name, Medication_description
190 FROM Medication
191 WHERE PatientID IN (6, 14);
192
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Medication_name	Medication_description		
Jakubowski	Octinoxate		
Schiller-Kilback	Isopropyl Alcohol		

EXPLANATION

Here we select the medication name and medication description of patients whose IDs lie between 6 and 14.

14. Give information on the medication name Schiller-Kilback from the medication table but exact spelling is unknown.




QUERY



```
Select * from medication where medication_name like '_c%ler%ck';
```


356 • `Select * from medication where medication_name like '_c%ler%ck';`

Result Grid

Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

	MedicationID	Medication_name	PatientID	MedicalstaffID	Medication_instructions	Medication_description
▶	804	Schiller-Kilback	14	493	Citalopram Hydrobro...	Isopropyl Alcohol
*	NULL	NULL	NULL	NULL	NULL	NULL

EXPLANATION

In this query like keyword is used because the exact spelling is unknown and obtained the desired output.

15.What are the total billing amounts of each patient associated with each medical staff and AppointmentID, including subtotals and the grand total?

QUERY

```
select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount) as Billing_Amount
from billing
group by PatientID, MedicalstaffID , AppointmentID with ROLLUP;
```

OUTPUT

```
306 • select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount) as Billing_Amount
307 from billing
308 group by PatientID, MedicalstaffID , AppointmentID with ROLLUP;
```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content: F6				
	PatientID	MedicalstaffID	AppointmentID	Billing_Amount
▶	1	493	594	526
	1	493	NULL	526
	1	NULL	NULL	526
	2	416	578	958
	2	416	NULL	958
	2	427	534	8
	2	427	NULL	8
	2	NULL	NULL	966
	3	451	564	231
	3	451	NULL	231
	3	488	578	758
	3	488	NULL	758
	3	NULL	NULL	989
	4	455	588	144
	4	455	NULL	144
	4	NULL	NULL	144
	7	488	532	899
	7	488	NULL	899
	7	NULL	NULL	899
	9	401	521	458
	9	401	NULL	458
	9	NULL	NULL	458
	10	465	518	237
	10	465	NULL	237
	10	481	518	440
	10	481	NULL	440
	10	NULL	NULL	677
	12	425	547	924
	12	425	NULL	924
	12	446	547	936
	12	446	NULL	936
	12	NULL	NULL	1860
	13	430	503	124
	13	430	NULL	124
	13	NULL	NULL	124
	15	480	596	308
	15	480	NULL	308
	15	NULL	NULL	308
	NULL	NULL	NULL	6951

EXPLANATION

In the 'Billing' table, the query determines the grand total for completed appointments as well as the total billing amounts for each patient linked to each medical staff member and appointment, including subtotals. Subtotals for every combination of PatientID, MedicalstaffID, and AppointmentID are added by the `WITH ROLLUP` clause, along with a grand total for all records.

16. What are the total billing amounts of each patient associated with each medical staff and appointment whose payment status is complete?

QUERY

```
select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount) as  
Billing_Amount  
from billing where billing_paymentstatus="completed"  
group by PatientID, MedicalstaffID , AppointmentID;
```

OUTPUT

```
311 • select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount) as Billing_Amount
312 from billing where billing_paymentstatus="completed"
313 group by PatientID, MedicalstaffID , AppointmentID;
```

	PatientID	MedicalstaffID	AppointmentID	Billing_Amount
▶	12	425	547	924
	12	446	547	936
	3	451	564	231
	9	401	521	458
	10	481	518	440
	3	488	578	758

EXPLANATION

When the payment status is indicated as "completed," the query retrieves the total billing amounts for each patient associated with a particular medical staff member and appointment. A summary of these billing amounts categorized by PatientID, MedicalstaffID, and AppointmentID is given by the result.

17. What are the total billing amounts by each patient associated with each Medical staff and Appointment, including all subtotals and grand totals?

QUERY

```
select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount)  
as Billing_Amount  
from billing  
group by CUBE (PatientID, MedicalstaffID , AppointmentID)  
order by PatientID, MedicalstaffID , AppointmentID;
```


OUTPUT

```

319 select PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount) as Billing_Am
320 from billing
321 group by CUBE (PatientID, MedicalstaffID , AppointmentID)
322 order by PatientID, MedicalstaffID , AppointmentID;

```

Data Output Messages Notifications

	patientid integer	medicalstaffid integer	appointmentid integer	billing_amount bigint
1	1	493	594	526
2	1	493	[null]	526
3	1	[null]	594	526
4	1	[null]	[null]	526
5	2	416	578	958
6	2	416	[null]	958
7	2	427	534	8
8	2	427	[null]	8
9	2	[null]	534	8
10	2	[null]	578	958
11	2	[null]	[null]	966
12	3	451	564	231
13	3	451	[null]	231
14	3	488	578	758
15	3	488	[null]	758
16	3	[null]	564	231
17	3	[null]	578	758
18	3	[null]	[null]	989
19	4	455	588	144
20	4	455	[null]	144
21	4	[null]	588	144
22	4	[null]	[null]	144
23	7	488	532	899
24	7	488	[null]	899
25	7	[null]	532	899
26	7	[null]	[null]	899
27	9	401	521	458
28	9	401	[null]	458
29	9	[null]	521	458
30	9	[null]	[null]	458
31	10	465	518	237
32	10	465	[null]	237
33	10	481	518	440
34	10	481	[null]	440
35	10	[null]	518	677

36	10	[null]	[null]	677
37	12	425	547	924
38	12	425	[null]	924
39	12	446	547	936
40	12	446	[null]	936
41	12	[null]	547	1860
42	12	[null]	[null]	1860
43	13	430	503	124
44	13	430	[null]	124
45	13	[null]	503	124
46	13	[null]	[null]	124
47	15	480	596	308
48	15	480	[null]	308
49	15	[null]	596	308
50	15	[null]	[null]	308
51	[null]	401	521	458
52	[null]	401	[null]	458
53	[null]	416	578	958
54	[null]	416	[null]	958
55	[null]	425	547	924
56	[null]	425	[null]	924
57	[null]	427	534	8
58	[null]	427	[null]	8
59	[null]	430	503	124
60	[null]	430	[null]	124
61	[null]	446	547	936
62	[null]	446	[null]	936
63	[null]	451	564	231
64	[null]	451	[null]	231
65	[null]	455	588	144
66	[null]	455	[null]	144
67	[null]	465	518	237
68	[null]	465	[null]	237
69	[null]	480	596	308
70	[null]	480	[null]	308
71	[null]	481	518	440
72	[null]	481	[null]	440
73	[null]	488	532	899
74	[null]	488	578	758
75	[null]	488	[null]	1657

76	[null]	493	594	526
77	[null]	493	[null]	526
78	[null]	[null]	503	124
79	[null]	[null]	518	677
80	[null]	[null]	521	458
81	[null]	[null]	532	899
82	[null]	[null]	534	8
83	[null]	[null]	547	1860
84	[null]	[null]	564	231
85	[null]	[null]	578	1716
86	[null]	[null]	588	144
87	[null]	[null]	594	526
88	[null]	[null]	596	308
89	[null]	[null]	[null]	6951

EXPLANATION

By classifying the data according to these criteria, the query determines the total billing amounts for each combination of Patient, Medical staff, and Appointment, including subtotals and grand totals. A breakdown of the billing amounts for each patient, member of the medical staff, and combination of appointments is provided in the result set.

18. Find a running total of Billing amounts.

QUERY

```
SELECT PatientID, MedicalstaffID , AppointmentID,  
sum(Billing_amount)  
OVER ( ORDER BY patientID ROWS UNBOUNDED PRECEDING)  
as "Running totals of Billing Amount" from Billing;
```

OUTPUT

```
324 SELECT PatientID, MedicalstaffID , AppointmentID, sum(Billing_amount)  
325 OVER ( ORDER BY patientid ROWS UNBOUNDED PRECEDING)  
326 as "Running totals of Billing Amount" from Billing;
```

Data Output Messages Notifications



	patientid integer	medicalstaffid integer	appointmentid integer	Running totals of Billing Amount bigint
1	1	493	594	526
2	2	427	534	534
3	2	416	578	1492
4	3	451	564	1723
5	3	488	578	2481
6	4	455	588	2625
7	7	488	532	3524
8	9	401	521	3982
9	10	465	518	4219
10	10	481	518	4659
11	12	425	547	5583
12	12	446	547	6519
13	13	430	503	6643
14	15	480	596	6951

EXPLANATION

The running total of the billing amounts, divided by PatientID and sorted by PatientID, is determined by this query for every row in the "Billing" table. Listed in the result are the following: the PatientID, MedicalstaffID, AppointmentID, and the running total with the heading "Running totals of Billing Amount."