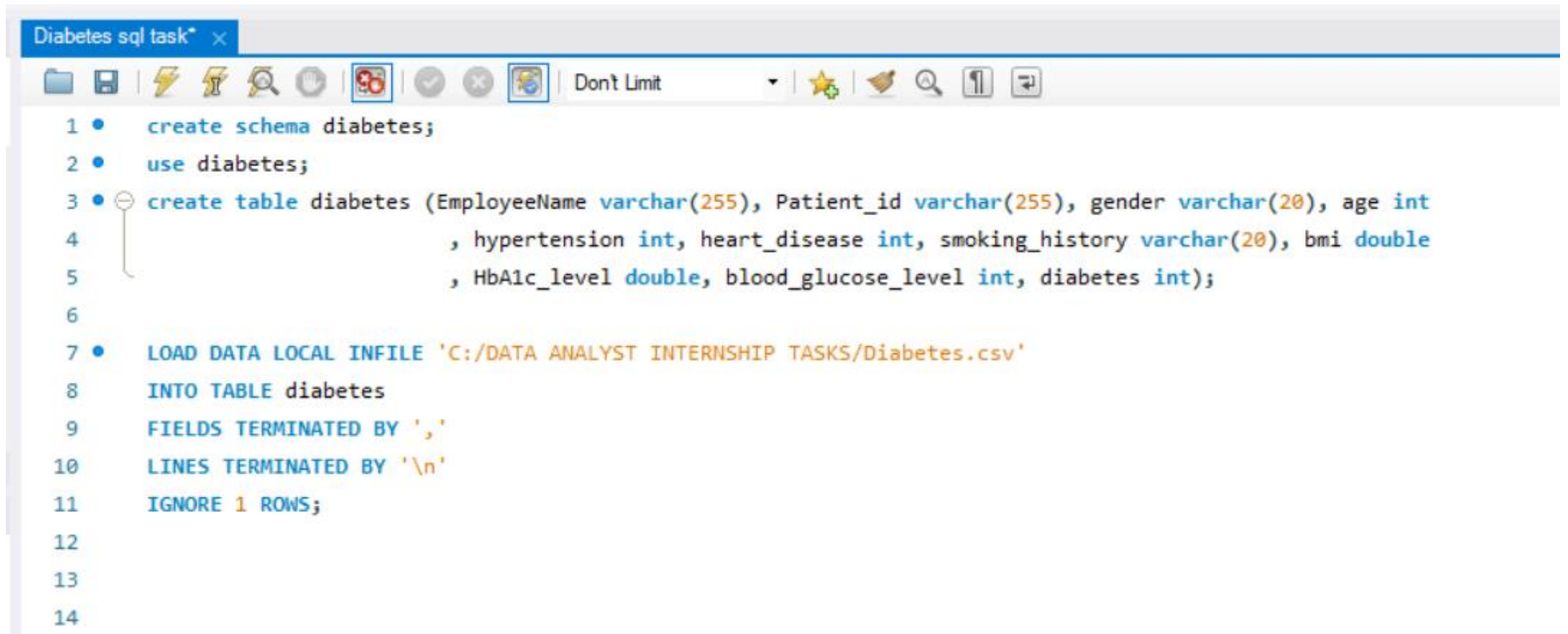




# DIABETES PREDICTION

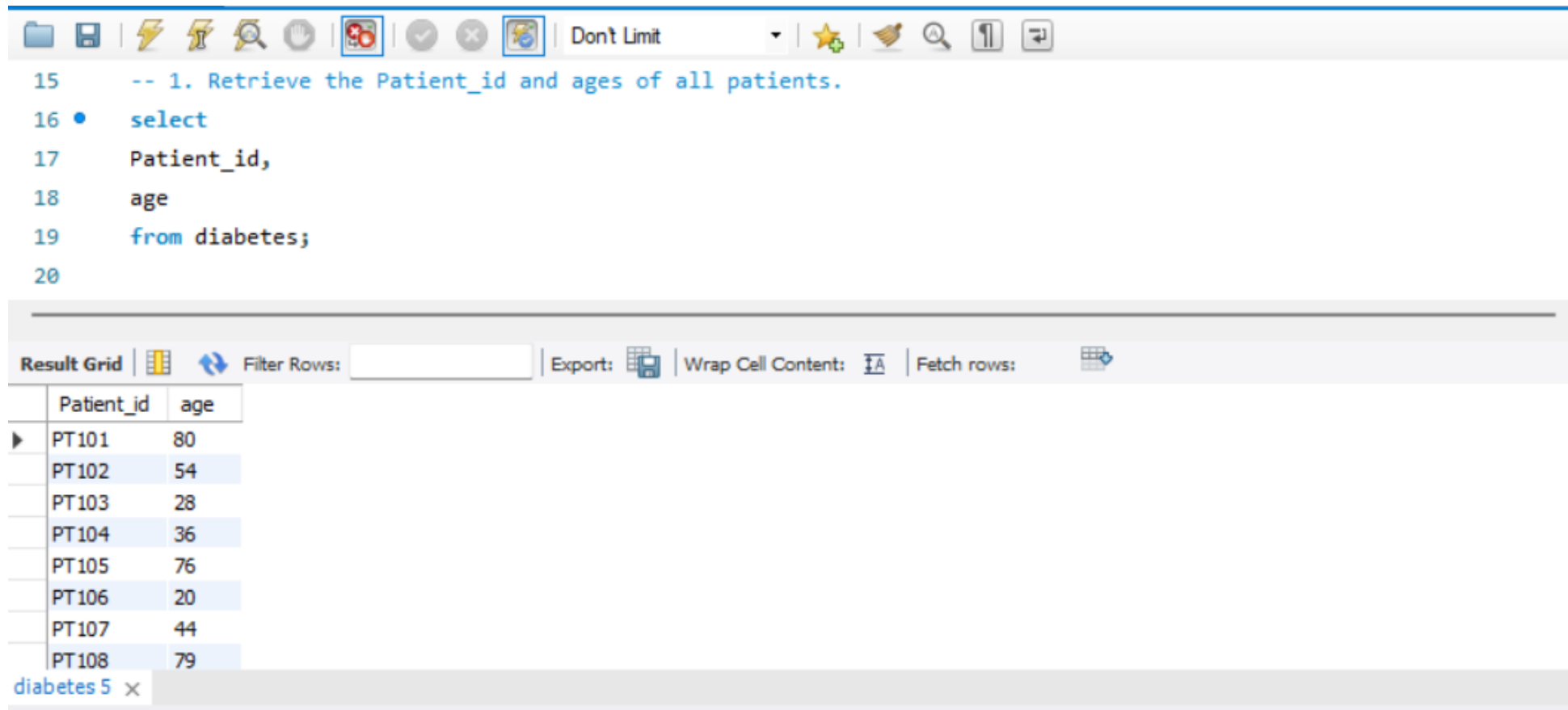
12/7/2023

# SCHEMA CREATION AND DATA IMPORT FOR DIABETES DATASET.



```
Diabetes sql task x
1 • create schema diabetes;
2 • use diabetes;
3 • create table diabetes (EmployeeName varchar(255), Patient_id varchar(255), gender varchar(20), age int
4                               , hypertension int, heart_disease int, smoking_history varchar(20), bmi double
5                               , HbA1c_level double, blood_glucose_level int, diabetes int);
6
7 • LOAD DATA LOCAL INFILE 'C:/DATA ANALYST INTERNSHIP TASKS/Diabetes.csv'
8 INTO TABLE diabetes
9 FIELDS TERMINATED BY ','
10 LINES TERMINATED BY '\n'
11 IGNORE 1 ROWS;
12
13
14
```

# QUESTION 1: RETRIEVE THE PATIENT\_ID AND AGES OF ALL PATIENTS.



The screenshot displays a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The SQL editor contains the following query:

```
15  -- 1. Retrieve the Patient_id and ages of all patients.  
16  •  select  
17      Patient_id,  
18      age  
19  from diabetes;  
20
```

Below the editor, the 'Result Grid' tab is active, showing the query results in a table. The table has two columns: 'Patient\_id' and 'age'. The results are as follows:

Patient_id	age
PT101	80
PT102	54
PT103	28
PT104	36
PT105	76
PT106	20
PT107	44
PT108	79

The bottom status bar indicates the current tab is 'diabetes 5'.

# QUESTION 2: SELECT ALL FEMALE PATIENTS WHO ARE OLDER THAN 40.

Diabetes sql task\* x

21

22 -- 2. Select all female patients who are older than 40.

23 • select \*from diabetes

24 where gender = 'female' and age >40;

25

26

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	NATHANIEL FORD	PT101	Female	80	0	1	never	25.19	6.6	140	0
	GARY JIMENEZ	PT102	Female	54	0	0	No Info	27.32	6.6	80	0
	ALSON LEE	PT107	Female	44	0	0	never	19.31	6.5	200	1
	DAVID KUSHNER	PT108	Female	79	0	0	No Info	23.86	5.7	85	0
	ARTHUR KENNEY	PT111	Female	53	0	0	never	27.32	6.1	85	0
	PATRICIA JACKSON	PT112	Female	54	0	0	former	54.7	6	100	0
	EDWARD HARRINGTON	PT113	Female	78	0	0	former	36.05	5	130	0
	JOHN MARTIN	PT114	Female	67	0	0	never	25.69	5.8	200	0

diabetes 6 x

# QUESTION 3: CALCULATE THE AVERAGE BMI OF PATIENTS.

Diabetes sql task\* x

26

27 -- 3. Calculate the average BMI of patients.

28 • select

29 avg(bmi) as Avg\_BMI

30 from diabetes;

31

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

Avg_BMI
27.32076709999422

Result 7 x

# QUESTION 4: LIST PATIENTS IN DESCENDING ORDER OF BLOOD GLUCOSE LEVELS.

Diabetes sql task\* x

31

32 -- 4. List patients in descending order of blood glucose levels.

33 • select \* from diabetes

34 order by blood\_glucose\_level desc;

35

36

---

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	Amado A Lumas Jr	PT99663	Male	56	1	0	current	28.47	6.1	300	1
	Shanice M Guidry	PT99672	Male	57	1	0	never	41.93	5.7	300	1
	Angelica J Young	PT99764	Male	80	0	0	No Info	34	9	300	1
	Flor D Roman	PT99809	Male	62	0	0	not current	27.32	6	300	1
	Clyde L Woods	PT99927	Male	63	0	1	No Info	27.32	6.6	300	1
	Josephine C Cabrera	PT99968	Male	64	1	0	former	33.12	5.7	300	1
	Marquis D Walker	PT100039	Male	55	0	1	former	30.42	6.2	300	1
	Alsifredo A Pina Figueroa	PT97141	Female	44	0	0	never	30.83	6.5	300	1

diabetes 8 x

# QUESTION 5: FIND PATIENTS WHO HAVE HYPERTENSION AND DIABETES.

Diabetes sql task\*

```
36
37  -- 5. Find patients who have hypertension and diabetes.
38 • select * from diabetes
39   where hypertension = 1 and diabetes = 1;
40
41
```

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	JONES WONG	PT139	Male	50	1	0	current	27.32	5.7	260	1
	PATRIC STEELE	PT205	Female	80	1	0	never	27.32	6.8	280	1
	ARTHUR STELLINI	PT343	Male	57	1	1	not current	27.77	6.6	160	1
	CHAD LAW	PT355	Male	63	1	0	ever	35.06	5.8	200	1
	CATHERINE JAMES	PT451	Female	52	1	0	never	50.3	6.6	155	1
	JOHN HART	PT565	Male	48	1	0	current	36.12	6.8	140	1
	JOHN BARKER	PT567	Female	79	1	0	former	27.32	6.5	159	1
	ROBERT BONNET	PT632	Female	49	1	0	not current	36.93	8.8	155	1

diabetes 9

# QUESTION 6: DETERMINE THE NUMBER OF PATIENTS WITH HEART DISEASE.

Diabetes sql task\*

```
42  -- 6.Determine the number of patients with heart disease.
43  •  select
44      count(Patient_id) as No_of_Patients_with_Heart_Disease
45      from diabetes
46      where heart_disease = 1;
47
```

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

No_of_Patients_with_Heart_Disease
3942

Result 10



# QUESTION 7: GROUP PATIENTS BY SMOKING HISTORY AND COUNT HOW MANY SMOKERS AND NON-SMOKERS THERE ARE.

Diabetes sql task\* x

48 -- 7. Group patients by smoking history and count how many smokers and non-smokers there are.  
49 • select  
50 smoking\_history,  
51 count(\*) as No\_of\_Patients  
52 from diabetes  
53 where smoking\_history in ('current','never')  
54 group by 1;  
55

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

	smoking_history	No_of_Patients
▶	never	35095
	current	9286

Result 11 x

# QUESTION 8: RETRIEVE THE PATIENT\_IDS OF PATIENTS WHO HAVE A BMI GREATER THAN THE AVERAGE BMI.

Diabetes sql task\* x

55

56 -- 8. Retrieve the Patient\_ids of patients who have a BMI greater than the average BMI.

57 • select

58 Patient\_id

59 from diabetes

60 where bmi > (select avg(bmi) from diabetes);

61

62

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

	Patient_id
▶	PT109
	PT112
	PT113
	PT117
	PT121
	PT124

diabetes 12 x

# QUESTION 9: FIND THE PATIENT WITH THE HIGHEST HBA1C LEVEL AND THE PATIENT WITH THE LOWEST HBA1C LEVEL.

Diabetes sql task\*

```
-- 9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1c level.  
select * from diabetes  
where HbA1c_level = (select max(HbA1c_level) from diabetes)  
union  
select * from diabetes  
where HbA1c_level = (select min(HbA1c_level) from diabetes);
```

Result Grid

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	MICHAEL THOMPSON	PT141	Male	73	0	0	former	25.91	9	160	1
	KEVIN CASHMAN	PT156	Male	50	0	0	former	37.16	9	159	1
	MARK CASTAGNOLA	PT236	Male	80	0	0	never	22.06	9	155	1
	WILLIAM SCOTT	PT270	Female	61	0	0	not current	39.36	9	140	1
	JOANNE HOEPER	PT400	Female	42	0	0	never	24.81	9	159	1
	VINCENT PAMPANIN	PT519	Female	52	0	0	No Info	27.32	9	140	1

Result 13

# QUESTION 10: CALCULATE THE AGE OF PATIENTS IN YEARS (ASSUMING THE CURRENT DATE AS OF NOW).

Diabetes sql task\*

```
72  -- 10. Calculate the age of patients in years (assuming the current date as of now).
73  • select
74  Patient_id,
75  age,
76  year(current_date()) - age as 'Birth_Year'
77  from diabetes;
78
79
```

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

	Patient_id	age	Birth_Year
▶	PT101	80	1943
	PT102	54	1969
	PT103	28	1995
	PT104	36	1987
	PT105	76	1947
	PT106	20	2003

Result 15 x

# QUESTION 11: RANK PATIENTS BY BLOOD GLUCOSE LEVEL WITHIN EACH GENDER GROUP.

Diabetes sql task\*

```
-- 11.Rank patients by blood glucose level within each gender group.
select
  EmployeeName as Patient_Name,
  Patient_id,
  gender,
  blood_glucose_level,
  dense_rank () over (partition by gender order by blood_glucose_level desc) as Glucose_Rank
from diabetes;
```

Result Grid

	Patient_Name	Patient_id	gender	blood_glucose_level	Glucose_Rank
▶	Gilbert J Fragoso	PT99638	Female	300	1
	Rahman A Jhinu	PT96815	Female	300	1
	Tualatai Auimatagi	PT98538	Female	300	1
	Nadine R Gordon	PT96371	Female	300	1
	Idalia R Farina	PT97708	Female	300	1
	Maria D Costa	PT97708	Female	300	1

Result 16

# QUESTION 12: UPDATE THE SMOKING HISTORY OF PATIENTS WHO ARE OLDER THAN 50 TO 'EX-SMOKER.'

Diabetes sql task\*

88

89 -- 12. Update the smoking history of patients who are older than 50 to "Ex-smoker."

90 • set sql\_safe\_updates = 0;

91

92 • update diabetes

93 set smoking\_history = 'Ex-smoker' where age > 50;

94

95 • select Patient\_id, smoking\_history, age from diabetes;

96

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

	Patient_id	smoking_history	age
▶	PT101	Ex-smoker	80
	PT102	Ex-smoker	54
	PT103	never	28
	PT104	current	36
	PT105	Ex-smoker	76
	PT106	never	28

diabetes 17 x

# QUESTION 13: INSERT A NEW PATIENT INTO THE DATABASE WITH SAMPLE DATA

Diabetes sql task\* x

Don't Limit

```
97 -- 13. Insert a new patient into the database with sample data
98 • insert into diabetes (EmployeeName, Patient_id, gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level,
99 values ('ABCD', 'PT0154235', 'Male', 54, 1, 1, 'Ex-Smoker', 26.12, 6.2, 187, 1 );
100
101 • select * from diabetes
102 where EmployeeName = 'ABCD';
103
104
105
```

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	ABCD	PT0154235	Male	54	1	1	Ex-Smoker	26.12	6.2	187	1

diabetes 18 x

# QUESTION 14: DELETE ALL PATIENTS WITH HEART DISEASE FROM THE DATABASE.

Diabetes sql task\*

105

106 -- 14. Delete all patients with heart disease from the database.

107 • delete from diabetes

108 where heart\_disease = 1;

109

110 • select \* from diabetes

111 where heart\_disease = 1;

112

113

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
--	--------------	------------	--------	-----	--------------	---------------	-----------------	-----	-------------	---------------------	----------

diabetes 20



# QUESTION 15: FIND PATIENTS WHO HAVE HYPERTENSION BUT NOT DIABETES USING THE EXCEPT OPERATOR.

Diabetes sql task" x

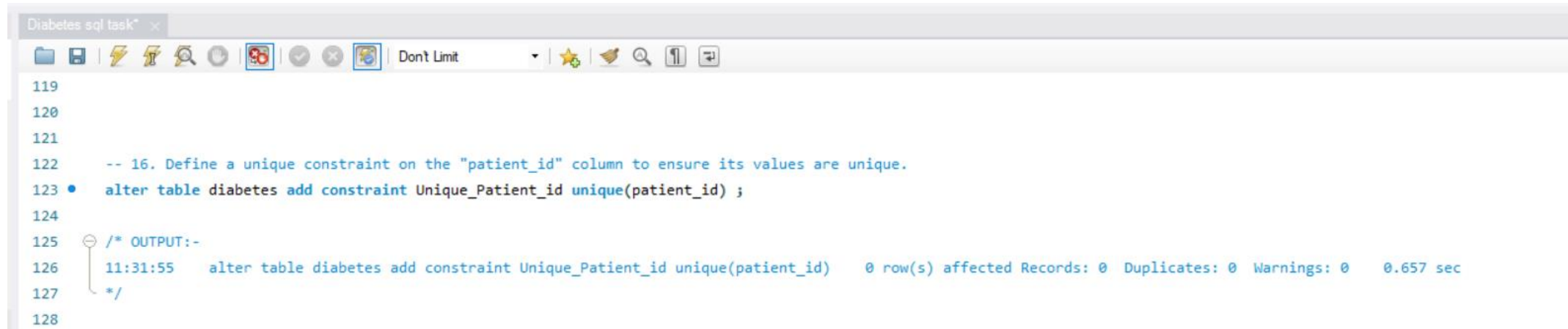
113  
114  
115 -- 15. Find patients who have hypertension but not diabetes using the EXCEPT operator.  
116 • select EmployeeName, patient\_id, hypertension, diabetes from diabetes where hypertension = 1  
117 ✖ except  
118 select EmployeeName, patient\_id, hypertension, diabetes from diabetes where diabetes = 1;  
119  
120  
121

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

	EmployeeName	patient_id	hypertension	diabetes
▶	DENISE SCHMITT	PT129	1	0
	RAY CRAWFORD	PT155	1	0
	KENNETH SMITH	PT161	1	0
	CHARLES SCOTT	PT215	1	0
	SHANNON SAKOWSKI	PT227	1	0
	MARICA MORSE	PT244	1	0

Result 21 x

# QUESTION 16: DEFINE A UNIQUE CONSTRAINT ON THE 'PATIENT\_ID' COLUMN TO ENSURE ITS VALUES ARE UNIQUE.



The screenshot shows a SQL IDE window titled "Diabetes sql task". The interface includes a toolbar with icons for file operations, execution, and search. The main area displays SQL code and its output. The code defines a unique constraint on the 'patient\_id' column of the 'diabetes' table. The output shows the successful execution of the command, with 0 rows affected, 0 duplicates, 0 warnings, and a duration of 0.657 seconds.

```
119
120
121
122  -- 16. Define a unique constraint on the "patient_id" column to ensure its values are unique.
123  • alter table diabetes add constraint Unique_Patient_id unique(patient_id) ;
124
125  /* OUTPUT:-
126  11:31:55      alter table diabetes add constraint Unique_Patient_id unique(patient_id)      0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0      0.657 sec
127  */
128
```

# QUESTION 17: CREATE A VIEW THAT DISPLAYS THE PATIENT\_IDS, AGES, AND BMI OF PATIENTS.

Diabetes sql task\* x

Don't Limit

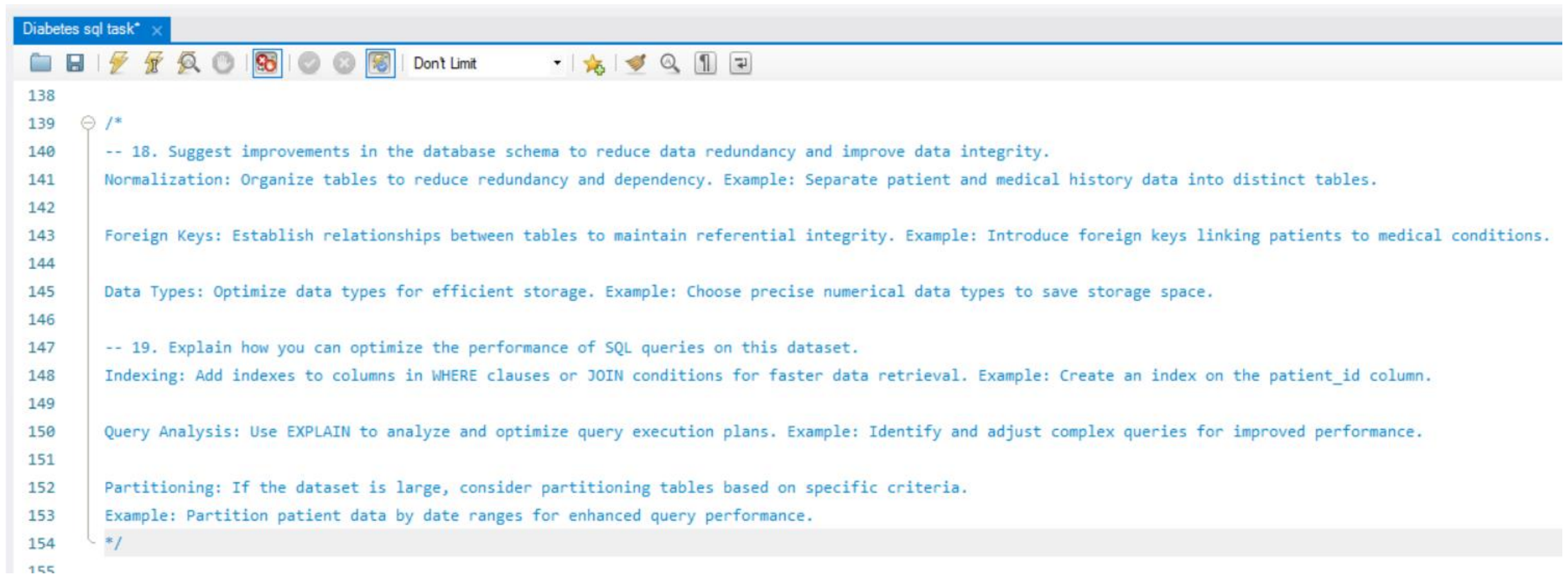
```
131 -- 17. Create a view that displays the Patient_ids, ages, and BMI of patients.  
132 • create view Patient_Details as  
133 select Patient_id,age,bmi from diabetes;  
134  
135 • select * from Patient_Details;  
136  
137  
138
```

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

	Patient_id	age	bmi
▶	PT102	54	27.32
	PT103	28	27.32
	PT104	36	23.45
	PT106	20	27.32
	PT107	44	19.31
	PT108	79	23.86
	PT109	42	33.64
	PT110	32	27.32
	PT111	53	27.32

Patient\_Details 22 x

# QUESTION 18,19:



```
138
139  /*
140  -- 18. Suggest improvements in the database schema to reduce data redundancy and improve data integrity.
141  Normalization: Organize tables to reduce redundancy and dependency. Example: Separate patient and medical history data into distinct tables.
142
143  Foreign Keys: Establish relationships between tables to maintain referential integrity. Example: Introduce foreign keys linking patients to medical conditions.
144
145  Data Types: Optimize data types for efficient storage. Example: Choose precise numerical data types to save storage space.
146
147  -- 19. Explain how you can optimize the performance of SQL queries on this dataset.
148  Indexing: Add indexes to columns in WHERE clauses or JOIN conditions for faster data retrieval. Example: Create an index on the patient_id column.
149
150  Query Analysis: Use EXPLAIN to analyze and optimize query execution plans. Example: Identify and adjust complex queries for improved performance.
151
152  Partitioning: If the dataset is large, consider partitioning tables based on specific criteria.
153  Example: Partition patient data by date ranges for enhanced query performance.
154  */
155
```