



PSYLIQ

**DATA ANALYST
INTERNSHIP**

DIABETES PREDICTION ANALYSIS

BY ANKUSH VERMA



Q1. RETRIEVE THE PATIENT_ID AND AGES OF ALL PATIENTS

```
29 • SELECT Patient_id, age
30 FROM diabetes;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:







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

diabetes 3 ×

Q2. SELECT ALL FEMALE PATIENTS WHO ARE OLDER THAN 40

```
29 • SELECT *
30 FROM diabetes
31 WHERE gender = "Female" AND age > 40;
```

Result Grid  Filter Rows: <input type="text"/> 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

diabetes 4 ×






Q3. CALCULATE THE AVERAGE BMI OF PATIENTS

```
27 • SELECT AVG(bmi) AS Average_BMI
28     FROM diabetes;
29
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Average_BMI				
▶	27.32076709999422				




Q4. LIST PATIENTS IN DESCENDING ORDER OF BLOOD GLUCOSE LEVELS

```
27 • SELECT *
28 FROM diabetes
29 ORDER BY blood_glucose_level DESC;
30
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content:  Fetch rows: 											
	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
	GABRIELLA RODE...	PT29771	Male	39	0	0	never	34.54	8.8	300	1
	ELIZABETH MOSER	PT30101	Female	39	0	0	current	30.73	8.8	300	1
	MARGOT REED	PT30021	Female	80	0	0	never	34.21	6.1	300	1
	RICARDO MYERS	PT29983	Female	80	1	0	No Info	20.34	7.5	300	1
	ISSEL ALVAREZ	PT30968	Male	58	1	0	No Info	27.32	6.8	300	1
	VICKI LEE	PT28652	Female	80	0	0	never	20.67	6.6	280	1
	KAREN HA	PT29448	Female	45	1	0	never	36.18	6.5	280	1
	ENRIQUE MORA	PT28707	Female	80	0	0	never	29.39	7.5	280	1
	WINSTON LOUIE	PT28746	Female	42	0	0	never	72.89	6.8	280	1
	THERESE WALES	PT30638	Female	69	0	1	never	20.3	5.7	280	1
	HUI LIN	PT30603	Male	79	0	1	former	27.32	6	280	1
	DALE RIVERS	PT29558	Female	42	0	0	current	37.5	5.8	280	1

Q5. FIND PATIENTS WHO HAVE HYPERTENSION AND DIABETES

```
27 • SELECT *
28 FROM diabetes
29 WHERE hypertension=1 AND diabetes=1;
30
```

Result Grid											
Filter Rows: <input type="text"/>											
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
	VITANI BENJAMIN	PT727	Male	43	1	0	not current	40.86	6.6	159	1
	LANNIE ADELMAN	PT828	Female	38	1	0	not current	27.32	6.1	160	1
	JOEL DELIZONNA	PT852	Female	28	1	0	never	20.09	6.6	200	1
	KAREN KUBICK	PT861	Male	59	1	0	ever	25.94	9	140	1

diabetes 7 x

Q6. DETERMINE THE NUMBER OF PATIENTS WITH HEART DISEASE

```
27 • SELECT COUNT(*) AS Heart_Diseases_Patients
28     FROM diabetes
29     WHERE heart_disease=1;
30
31
```

Result Grid



Filter Rows:

Export:







Wrap Cell Content:



	Heart_Diseases_Patients
▶	3942

Q7. GROUP PATIENTS BY SMOKING HISTORY AND COUNT HOW MANY SMOKERS AND NON-SMOKERS THERE ARE

```
27 • SELECT smoking_history, COUNT(*) AS COUNT
28     FROM diabetes
29     WHERE smoking_history="current" OR smoking_history="never"
30     GROUP BY smoking_history;
31
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	smoking_history	COUNT			
▶	never	35095			
	current	9286			

Q8. RETRIEVE THE PATIENT_IDS OF PATIENTS WHO HAVE A BMI GREATER THAN THE AVERAGE BMI

```
27      # AVG BMI is 27.32076709999422
28 •    SELECT Patient_id, bmi
29      FROM diabetes
30      WHERE bmi > (
31          SELECT avg(bmi)
32          FROM diabetes
33      );
34
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



F

	Patient_id	bmi
▶	PT109	33.64
	PT112	54.7
	PT113	36.05
	PT117	30.36
	PT121	36.38
	PT124	27.94
	PT126	33.76

Q9. FIND THE PATIENT WITH THE HIGHEST HBA1C LEVEL AND THE PATIENT WITH THE LOWEST HBA1C LEVEL

```
27 • SELECT EmployeeName, Patient_id, HbA1c_level as MAX_HbA1c_level
28 FROM diabetes
29 ORDER BY HbA1c_level DESC
30 LIMIT 1;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch row
EmployeeName	Patient_id	MAX_HbA1c_level		
▶ MICHAEL THOMPSON	PT141	9		

```
27 • SELECT EmployeeName, Patient_id, HbA1c_level as Min_HbA1c_level
28 FROM diabetes
29 ORDER BY HbA1c_level ASC
30 LIMIT 1;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
EmployeeName	Patient_id	Min_HbA1c_level		
▶ ELLEN MOFFATT	PT120	3.5		




Q10. CALCULATE THE AGE OF PATIENTS IN YEARS (ASSUMING THE CURRENT DATE AS OF NOW)

```
27 • SELECT EmployeeName, Patient_id,  
28        YEAR(NOW()) - age AS Birth_Year,  
29        YEAR(NOW()) - YEAR(NOW()) + age AS Current_Age  
30 FROM diabetes;
```

Result Grid				
Filter Rows:		Export:	Wrap Cell Content:	Fetch rows:
	EmployeeName	Patient_id	Birth_Year	Current_Age
▶	NATHANIEL FORD	PT101	1943	80
	GARY JIMENEZ	PT102	1969	54
	ALBERT PARDINI	PT103	1995	28
	CHRISTOPHER CHONG	PT104	1987	36
	PATRICK GARDNER	PT105	1947	76
	DAVID SULLIVAN	PT106	2003	20
	ALSON LEE	PT107	1979	44
	DAVID KUSHNER	PT108	1944	79
	MICHAEL MORRIS	PT109	1981	42





Q11. RANK PATIENTS BY BLOOD GLUCOSE LEVEL WITHIN EACH GENDER GROUP

```
27 • SELECT Patient_id, gender, blood_glucose_level,  
28      RANK() OVER (PARTITION BY gender ORDER BY blood_glucose_level) AS blood_glucose_level_rank_as_per_gender  
29      FROM diabetes;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	Patient_id	gender	blood_glucose_level	blood_glucose_level_rank_as_per_gender
	PT47949	Female	80	1
	PT47157	Female	80	1
	PT47163	Female	80	1
	PT47171	Female	80	1
	PT49143	Female	80	1
	PT46271	Female	85	4199
	PT46110	Female	85	4199
	PT48066	Female	85	4199

Q12. UPDATE THE SMOKING HISTORY OF PATIENTS WHO ARE OLDER THAN 50 TO "EX-SMOKER"

```
27 • SET SQL_SAFE_UPDATES = 0;
28 • UPDATE diabetes
29   SET smoking_history = "EX-Smoker"
30   WHERE age>50;
31
32 • SELECT * FROM diabetes;
```

Result Grid  Filter Rows: <input type="text"/> 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	EX-Smoker	25.19	6.6	140	0
	GARY JIMENEZ	PT102	Female	54	0	0	EX-Smoker	27.32	6.6	80	0
	ALBERT PARDINI	PT103	Male	28	0	0	never	27.32	5.7	158	0
	CHRISTOPHER CHONG	PT104	Female	36	0	0	current	23.45	5	155	0
	PATRICK GARDNER	PT105	Male	76	1	1	EX-Smoker	20.14	4.8	155	0
	DAVID SULLIVAN	PT106	Female	20	0	0	never	27.32	6.6	85	0
	ALSON LEE	PT107	Female	44	0	0	never	19.31	6.5	200	1
	DAVID KUSHNER	PT108	Female	79	0	0	EX-Smoker	23.86	5.7	85	0
	MICHAEL MORRIS	PT109	Male	42	0	0	never	33.64	4.8	145	0
	JOANNE HAYES-WHITE	PT110	Female	32	0	0	never	27.32	5	100	0

Q13. INSERT A NEW PATIENT INTO THE DATABASE WITH SAMPLE DATA





```
34 • INSERT INTO diabetes
35 VALUES ("DAVID WARNER", "PT100101", "Male", 35, 0, 0, "No Info", 33.01, 5.1, 100, 0);
36
37 • SELECT * FROM diabetes
38 LIMIT 100102;
```

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
	Antoinette L Wells	PT100097	Female	2	0	0	No Info	17.37	6.5	100	0
	Richard D Swart	PT100098	Male	66	0	0	EX-Smoker	27.83	5.7	155	0
	Vivian Chu	PT100099	Female	24	0	0	never	35.42	4	100	0
	Savitree Satram	PT100100	Female	57	0	0	EX-Smoker	22.43	6.6	90	0
	DAVID WARNER	PT100101	Male	35	0	0	No Info	33.01	5.1	100	0

Q14. DELETE ALL PATIENTS WITH HEART DISEASE FROM THE DATABASE

```
105      # Q14. Delete all patients with heart disease from the database
106
107 •    DELETE FROM diabetes
108      WHERE heart_disease=1;
109
110 •    SELECT * FROM diabetes;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content:  Fetch rows: 											
	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
	CLEMENTINO AVILA II	PT333	Female	80	0	0	EX-Smoker	27.32	6.2	145	0
	RYAN KENNEDY	PT334	Male	11	0	0	never	17.09	6.6	126	0
	JOHN ROCCO	PT335	Male	51	0	0	EX-Smoker	27.32	4.8	145	0
	ARNOLD CHOY	PT336	Female	71	0	0	EX-Smoker	27.32	5	126	0
	REGINA GOMEZ	PT337	Male	55	0	0	EX-Smoker	27.32	6.8	159	1

diabetes 33 x

Q15. FIND PATIENTS WHO HAVE HYPERTENSION BUT NOT DIABETES USING THE EXCEPT OPERATOR

```
112      # Q15. Find patients who have hypertension but not diabetes using the EXCEPT operator
113      •  SELECT Patient_id, hypertension, diabetes
114          FROM diabetes
115          WHERE hypertension=1
116      ✖   EXCEPT
117          SELECT Patient_id, hypertension, diabetes
118          FROM diabetes WHERE diabetes=1;
119
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	Patient_id	hypertension	diabetes
	PT958	1	0
	PT960	1	0
	PT972	1	0
	PT976	1	0
	PT980	1	0

Q16. DEFINE A UNIQUE CONSTRAINT ON THE "PATIENT_ID" COLUMN TO ENSURE ITS VALUES ARE UNIQUE

```
122 • ALTER TABLE diabetes
123     ADD CONSTRAINT Patient_id UNIQUE (Patient_id);
124
125 • INSERT INTO diabetes
126     VALUES ("DAVID WARNER", "PT100101", "Male", 35, 0, 0, "No Info", 33.01, 5.1, 100, 0);
```

Output








Action Output

#	Time	Action	Message
1	12:04:19	INSERT INTO diabetes VALUES ("DAVID WARNER", "PT100101", "Male", 35, 0, 0, "No Info...	Error Code: 1062. Duplicate entry 'PT100101' for key 'diabetes.Patient_id'

Adding a UNIQUE constraint to 'Patient_id' means that attempting to insert a duplicate 'Patient_id' will result in an error.

Q17. CREATE A VIEW THAT DISPLAYS THE PATIENT_IDS, AGES, AND BMI OF PATIENTS

```
130 • CREATE VIEW Patient_Data AS (  
131     SELECT Patient_id, age, bmi  
132     FROM diabetes  
133 );  
134 • SELECT * FROM Patient_Data;  
135
```

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

Patient Data 37 x

Q18. SUGGEST IMPROVEMENTS IN THE DATABASE SCHEMA TO REDUCE DATA REDUNDANCY AND IMPROVE DATA INTEGRITY

To Reduce Data Redundancy:

- **1) Normalization:** Split 'diabetes' table into 'Patients' and 'HealthRecords' tables, linking them with a foreign key on Patient_id.
- **2) Use Enumerations:** Replace VARCHAR with ENUM for categorical data (e.g., gender, smoking_history) to enhance consistency.
- **3) Avoid Repeating Groups:** Create a separate table for health records to establish a one-to-many relationship, preventing repeated health-related information in the main 'Patients' table.

To Improve Data Integrity:

- **1) Implement Constraints:** Enforce NOT NULL, UNIQUE, and FOREIGN KEY constraints for essential fields and to prevent duplicates.
- **2) Data Validation with Check Constraints:** Use CHECK constraints for conditions like valid age range and blood_glucose_level values.
- **3) Triggers for Automation:** Implement triggers to automate actions, such as updating timestamps for health record modifications, aiding in data auditing.

Q19. EXPLAIN HOW YOU CAN OPTIMIZE THE PERFORMANCE OF SQL QUERIES ON THIS DATASET

- **1) Indexing:** Create indexes on frequently queried columns like Patient_id, age, and diabetes for faster retrieval.
- **2) Use Proper Joins:** Optimize JOIN operations, ensuring efficient linking between tables.
- **3) Limit SELECT Columns:** Retrieve only necessary columns to minimize data transfer and improve query speed.
- **4) Update Statistics:** Regularly update database statistics to help the query optimizer generate efficient execution plans.
- **5) Partitioning:** Consider partitioning large tables based on certain criteria (e.g., date) to enhance query performance.