

TASK 2: DIABETES PREDICTION

MITALI NITIN KUBAL

[mitali.kubal2001@gmail.com](mailto:mitali.kubal2001@gmail.com)

Greetings,

I am embarking on a virtual internship tasked by Psyliq to delve into SQL analysis using a diabetes dataset. Through this endeavor, I aim to not only predict diabetes but also unearth the underlying factors influencing its prevalence. As a student pursuing MSc in Statistics, this internship presents a valuable opportunity to apply my skills and knowledge in a practical setting, enriching my understanding of complex healthcare data.

The dataset at hand encompasses essential variables such as Patient id, gender, date of birth (D.O.B), hypertension, heart disease, smoking history, body mass index (BMI), HbA1c level, blood glucose level, and diabetes status. Reflecting the intricacies of real-world healthcare data, it provides a fertile ground for honing my SQL skills and gaining insights into the dynamics of diabetes. By embarking on these tasks, I aim to not only fulfill the requirements of my internship but also contribute meaningfully to the understanding of diabetes dynamics through data-driven insights.

**Aim:**

To analyze the given dataset ‘Diabetes\_prediction.xlsx’ and perform the tasks provided.

1. Retrieve the Patient\_id and ages of all patients.

* SELECT Patient\_id, age FROM dp;

1. Select all female patients who are older than 40.

* SELECT \* FROM dp WHERE gender = 'Female' AND age > 40;

1. Calculate the average BMI of patients.

* SELECT AVG(bmi) AS average\_bmi FROM dp;

1. List patients in descending order of blood glucose levels.

* SELECT \* FROM dp ORDER BY blood\_glucose\_level DESC;

1. Find patients who have hypertension and diabetes.

* SELECT \* FROM dp WHERE hypertension = 1 AND diabetes = 1;

1. Determine the number of patients with heart disease.

* SELECT COUNT(\*) AS number\_of\_patients\_with\_heart\_disease FROM dp WHERE heart\_disease = 1;

1. Group patients by smoking history and count how many smokers and nonsmokers there are.

* SELECT smoking\_history, COUNT(\*) AS number\_of\_patients FROM dp GROUP BY smoking\_history;

1. Retrieve the Patient\_ids of patients who have a BMI greater than the average BMI.

* SELECT Patient\_id FROM dp WHERE bmi > (SELECT AVG(bmi) FROM dp);

1. Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.

* Patient with the highest HbA1c level

SELECT \* FROM dp WHERE HbA1\_level = (SELECT MAX(HbA1\_level) FROM dp);

* Patient with the lowest HbA1c level

SELECT \* FROM dp WHERE HbA1\_level = (SELECT MIN(HbA1\_level) FROM dp);

1. Calculate the age of patients in years (assuming the current date as of now).

* SELECT Patient\_id, age, DATEDIFF(CURDATE(), STR\_TO\_DATE(age, '%Y-%m-%d')) / 365 AS calculated\_age FROM dp;

1. Rank patients by blood glucose level within each gender group

* SELECT Patient\_id, gender, blood\_glucose\_level, RANK() OVER (PARTITION BY gender ORDER BY blood\_glucose\_level DESC) AS glucose\_level\_rank FROM dp;

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

* UPDATE dp SET smoking\_history = 'Ex-smoker' WHERE age > 50;

1. Insert a new patient into the database with sample data.

* INSERT INTO dp (EmployeeName, Patiend\_id, gender, age, hypertension, heart\_disease, smoking\_history, bmi, HbA1\_level, blood\_glucose\_level, diabetes) VALUES (SHAWNA CABELLO', 'P654321', 'female', 32, 'yes', 'yes', 'smoker', 23.9, 6.7,140, 'yes');

1. Delete all patients with heart disease from the database.

* DELETE FROM dp WHERE heart\_disease = 1;

1. Find patients who have hypertension but not diabetes using the EXCEPT operator.

* SELECT Patient\_id FROM dp WHERE hypertension = 1 EXCEPT SELECT Patient\_id FROM dp WHERE diabetes=1;

1. Define a unique constraint on the "patient\_id" column to ensure its values are unique.

* ALTER TABLE dp ADD UNIQUE (patient\_id);

1. Create a view that displays the Patient\_ids, ages, and BMI of patients.

* CREATE VIEW patient\_info AS SELECT Patient\_id, age, bmi FROM dp;

1. Suggest improvements in the database schema to reduce data redundancy and improve data integrity.

* To reduce data redundancy and improve data integrity in our database schema, we can consider the following suggestions:
* Normalization: Break down tables to eliminate redundancy and minimize data duplication.
* Foreign Keys: Establish relationships between tables using foreign keys to improve data integrity and maintain consistency.
* Demographics and Medical History Separation: Split patient information into separate tables for demographics and medical history to streamline queries and reduce redundancy.
* Normalization of Repeating Groups: Normalize repeating groups by creating separate tables for repeated data and linking them to the main table using foreign keys.
* Data Validation Constraints: Enforce constraints to ensure only valid data is entered, such as range constraints for numerical fields and format constraints for date fields.

19) Explain how you can optimize the performance of SQL queries on this dataset.

* Indexing: Create indexes on columns frequently used in WHERE, JOIN, and ORDER BY clauses to speed up data retrieval.
* Query Optimization: Write efficient queries by avoiding SELECT \* and using JOINs instead of subqueries where possible.
* Limiting Result Sets: Use the LIMIT keyword to restrict the number of rows returned, reducing data transfer overhead.
* Partitioning: Partition large tables based on criteria like date ranges to improve query performance by accessing smaller data subsets.
* Data Denormalization: Consider denormalizing tables to simplify complex JOIN operations and improve query performance in some cases.
* Optimizing Data Types: Use appropriate data types to minimize storage space and improve query execution speed.