DATASET IMPORTANCE OF HAVING SCHEMA IN A DATASET (PRIMARY KEY)

SUBMITTED BY:-**NANDINI RATHORE**

TABLES OF CONTENT:-

INTRODUCTION :->**Structured Data**

Structured data means data that is well-organized in rows and columns, like in a table. It is stored in databases such as Microsoft SQL or Excel. Each row is one record (like one student or one patient), and each column is a specific detail (like name, age, or address).

Examples of structured data:

- Student records in a school
- Patient details in a hospital
- Booking details of airline.

Structured data is easy to search, filter, and analyze.

Importance of schema **in Data Management**

A **schema** is like a plan or a map of how data is arranged in a database. It tells us:

- What kind of data is stored (like numbers or text)
- How different tables are connected
- Which column is the **Primary Key** (unique for each row,it didn't contain any null value and duplicate values.)
- How data in one table relates to another table using **Foreign Keys.**

Why schema is important:

- It keeps the data clean and organized.
- It avoids mistakes like storing wrong types of data.
- It helps us understand how different tables are connected.
- It makes searching and analyzing data faster and easier .

Without a schema, the data can become messy, hard to use, and full of errors.

ADVANTAGES OF PRIMARY KEY:-

1. Uniquely Identifies Each Record
    - No two rows will have the same primary key
    - Helps avoid duplicate data
2. Ensures Data Integrity
    - Guarantees that every row has a unique and non-null value
3. Helps in Indexing and Fast Searching
    - Speeds up data retrieval in queries
4. Foundation for Table Relationships
    - Used to link with other tables via foreign keys

DISADVANTAGES OF PRIMARY KEY:- **Cannot Be Null**:--Every record must have a value — sometimes a problem if data is incomplete.

Needs Proper Design:--Choosing a wrong primary key (like a name) can cause issues later.

Harder to Change:--If a primary key value needs to be updated, it may affect linked data.

ADVANTAGE OF FOREIGN KEY:--1. Connects records across different tables (e.g., patient → doctor).

2. Prevents orphan records (e.g., prescription linked to a non-existent patient).

3. Allows you to split large datasets into logical smaller parts.

4.NULL values are allowed unless a NOT NULL constraint is applied.Cannot be the same column as the primary key in the same table.

DISADVANTAGES OF FOREIGN KEY:--1. Database checks relationships before making changes, which can slow down performance.means slow insert,delete and update of statements.

2. Dependency on Parent Table:--cannot insert a value unless the referenced primary key exists.

3. Complexity in Large Schemas:--Managing many foreign keys can make design and debugging harder.

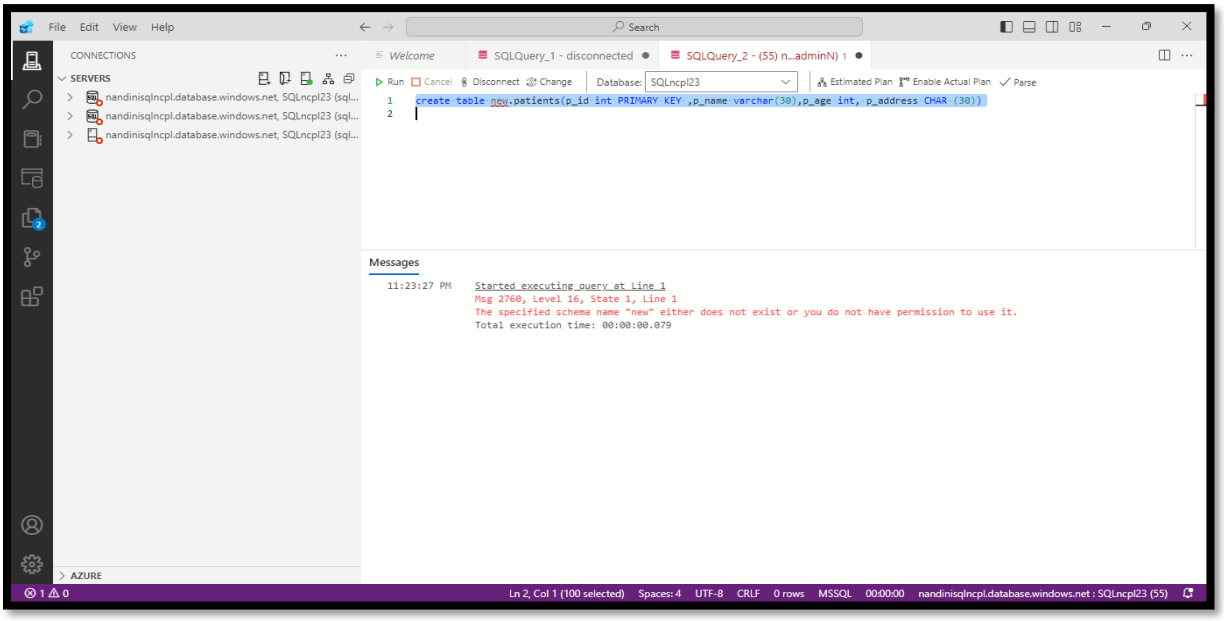
DATASET ANALYSIS

FIRST DATASET

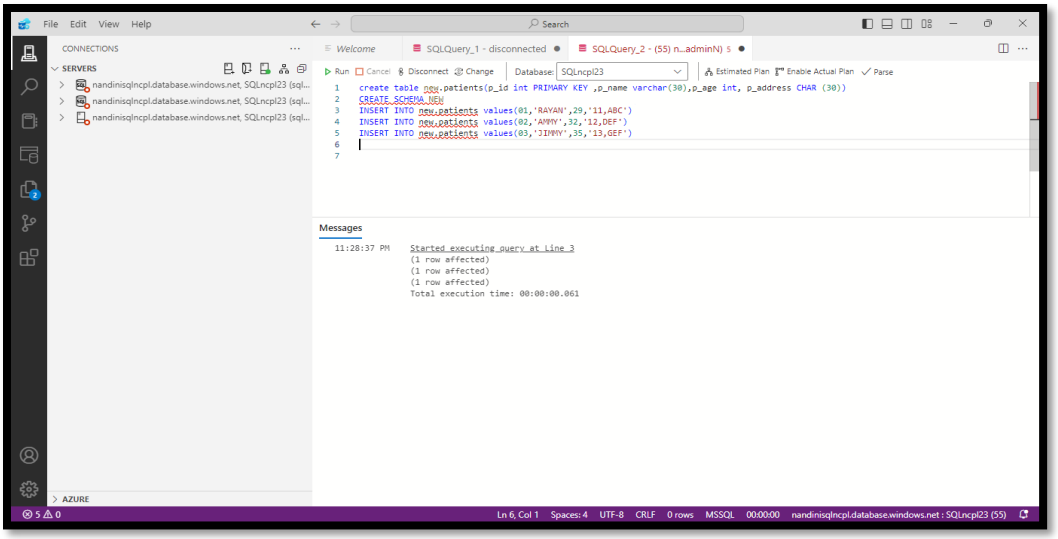
- **NEW.HOSPITAL:->**

  SCHEMA=NEW

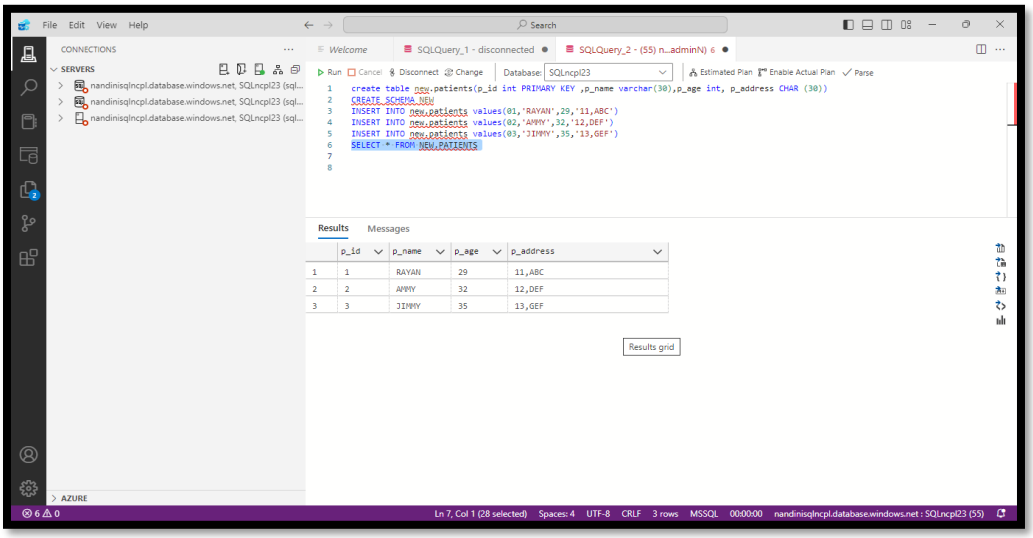
  TABLENAME=HOSPITAL



1.CREATE TABLE NEW.PATIENTS.IT SHOWS AN ERROR BECAUSE I DIDN'T CREATE SCHEMA FIRST.

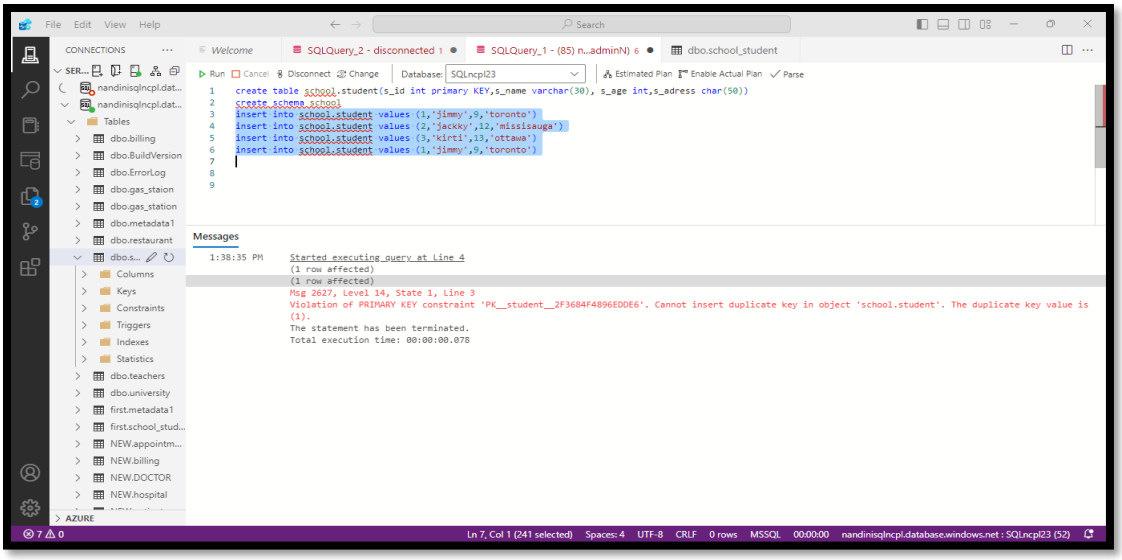


2.THEN I CREATE SCHEMA ,THEN CREATE TABLE AND INSERT VALUES INTO IT.

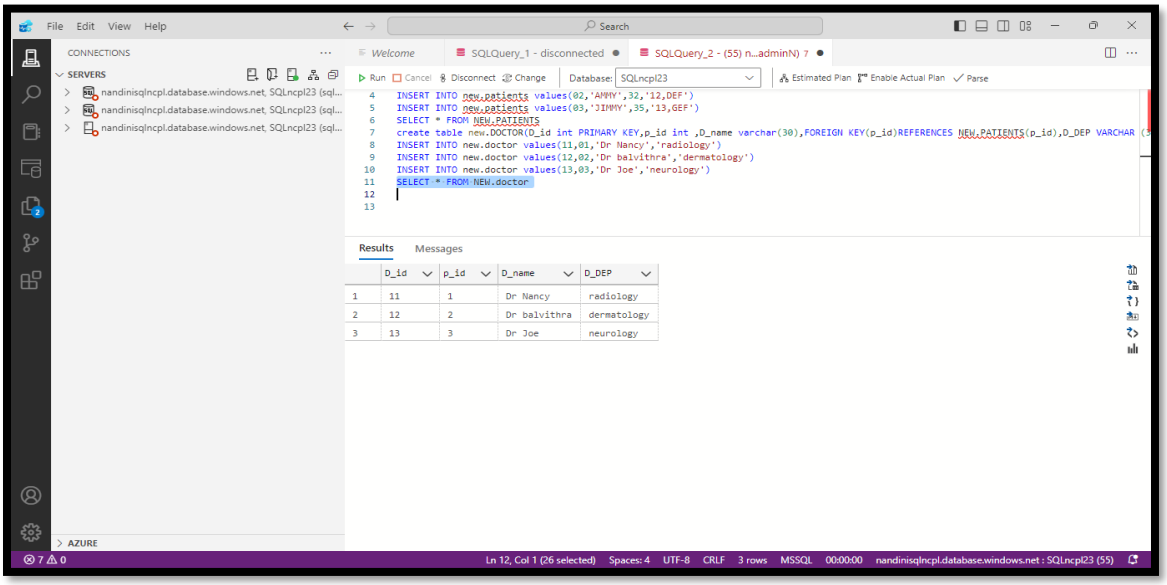


3.ENTITIES AND ATTRIBUTES OF TABLE :-NEW.PATIENT

- **Attributes:** `p_id` (Primary Key), `p_name`, `p_age`, `p_address`
- **Primary Key:** `p_id`
- **Description:** Stores basic patient details.



4.This shows error of duplicate key because I wrote the same s_id twice. As primary key didn't contain duplicate values and null values.

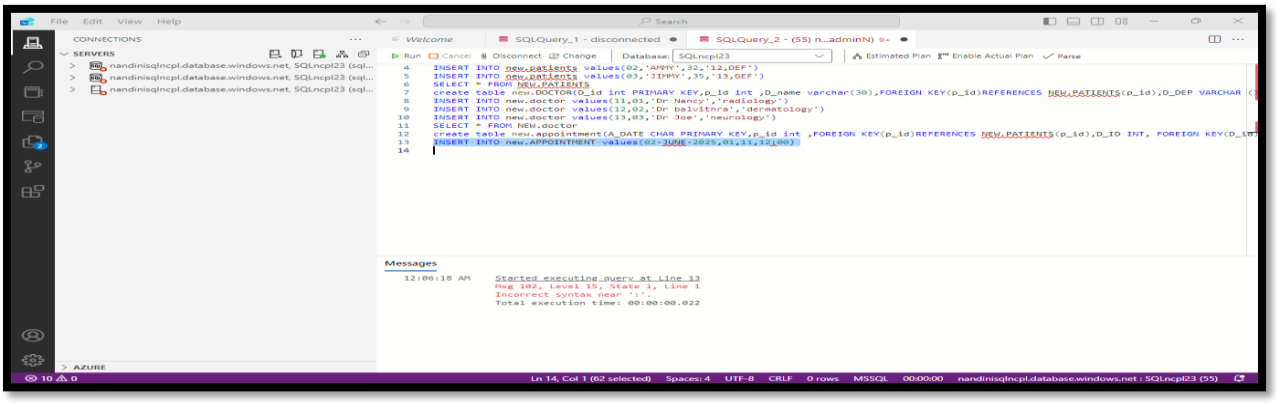


5. **Attributes:** D_id (Primary Key), p_id (Foreign Key), D_name, D_DEP

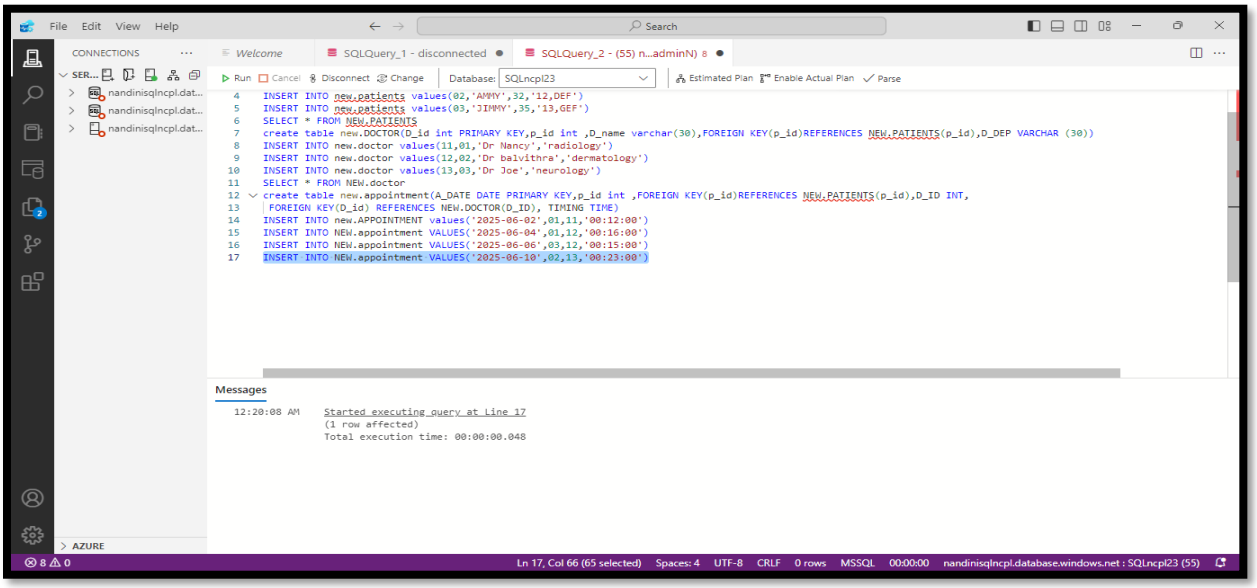
Primary Key: D_id

Foreign Key: p_id → patients(p_id)

Description: Stores doctor details, and each doctor is related to a patient.

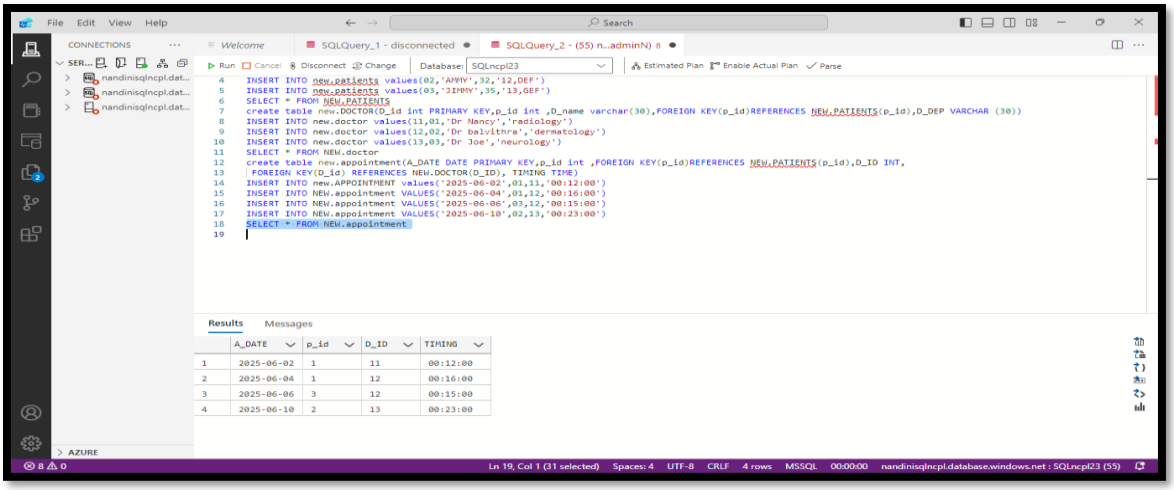


6.This error is because of time data datatype which require comma ’12:00’, in the screenshot.

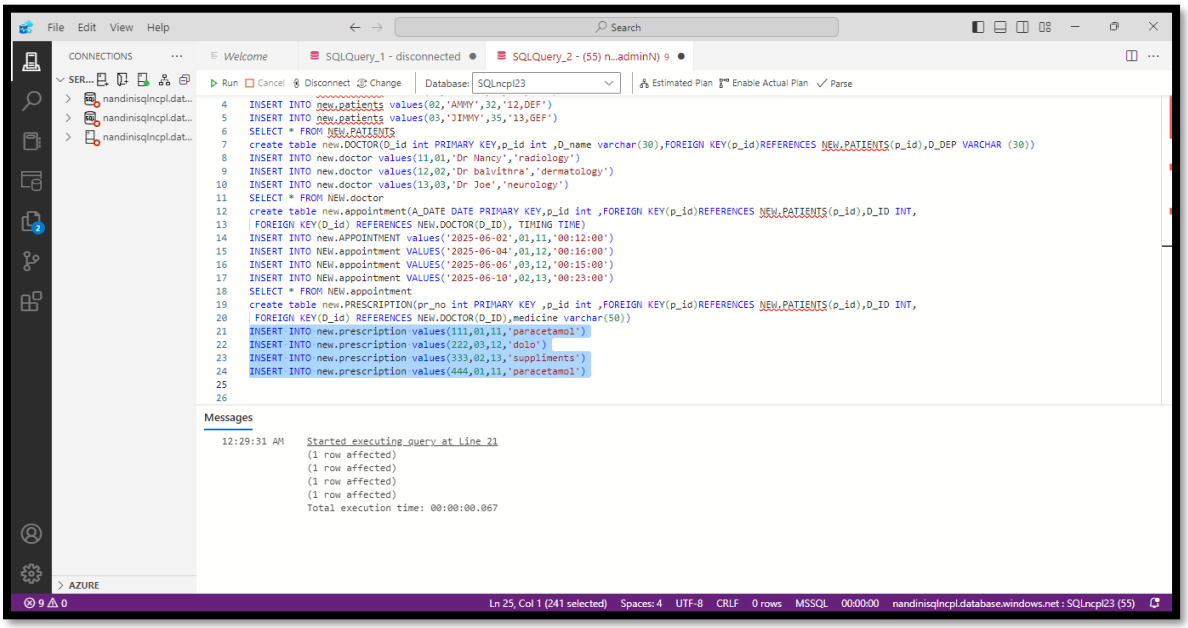


7. insert statement into table appointment.

- **Attributes:** A_DATE (Primary Key), p_id, D_ID, TIMING
- **Primary Key:** A_DATE
- **Foreign Keys:**
  - p_id → patients(p_id)
  - D_ID → doctor(D_ID)
- **Description:** Stores appointment info between patients and doctors.



9. using select statement , appointment table appears.



10. Insert values into prescription table.

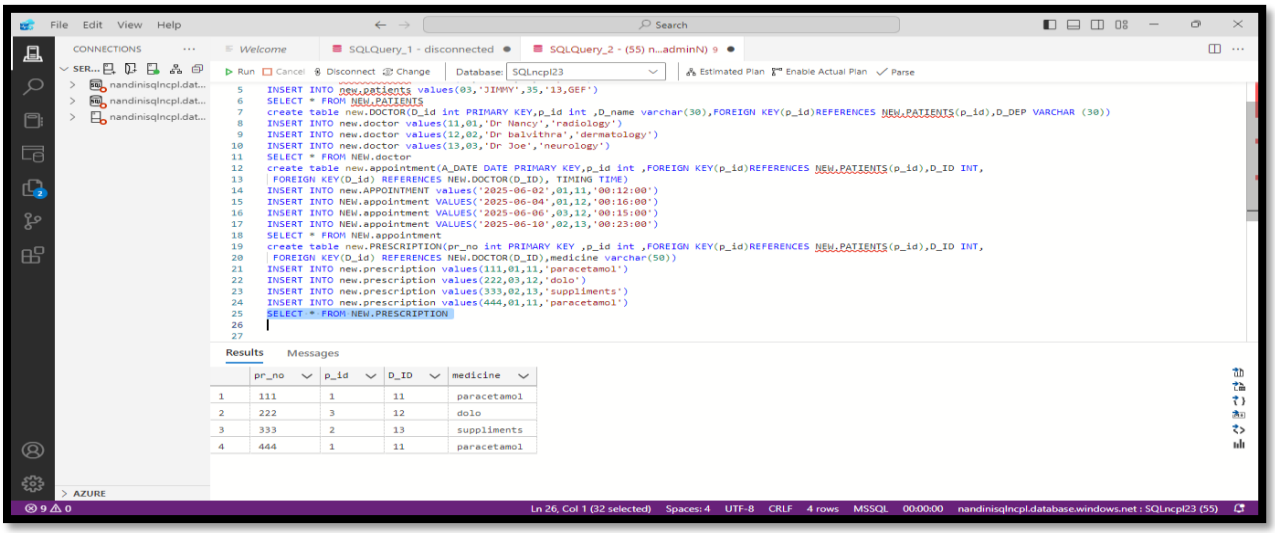
Attributes: pr_no (Primary Key), p_id, D_id, medicine

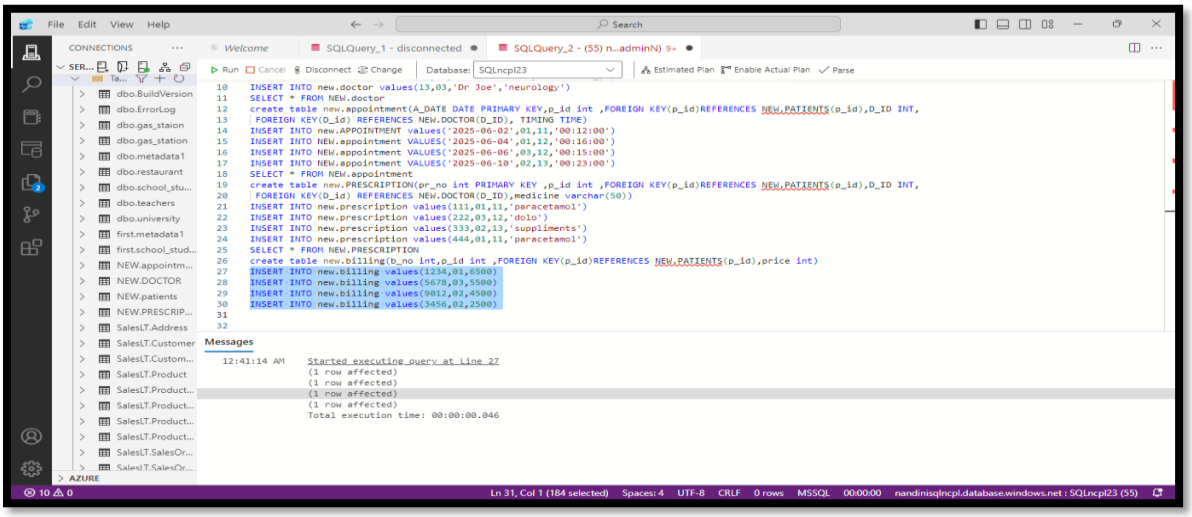
Primary Key: pr_no

Foreign Keys:

- p_id → patients(p_id)
- D_id → doctor(D_ID)

Description: Medicines prescribed to patients by doctors.





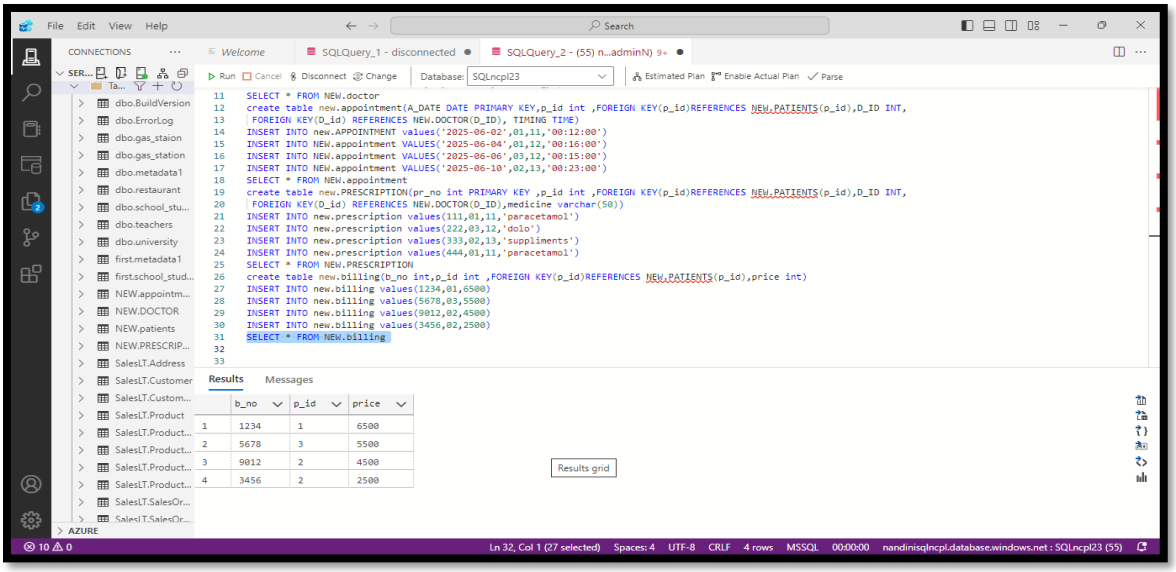
11.create table new.billing. Insert values in it and use select statement.

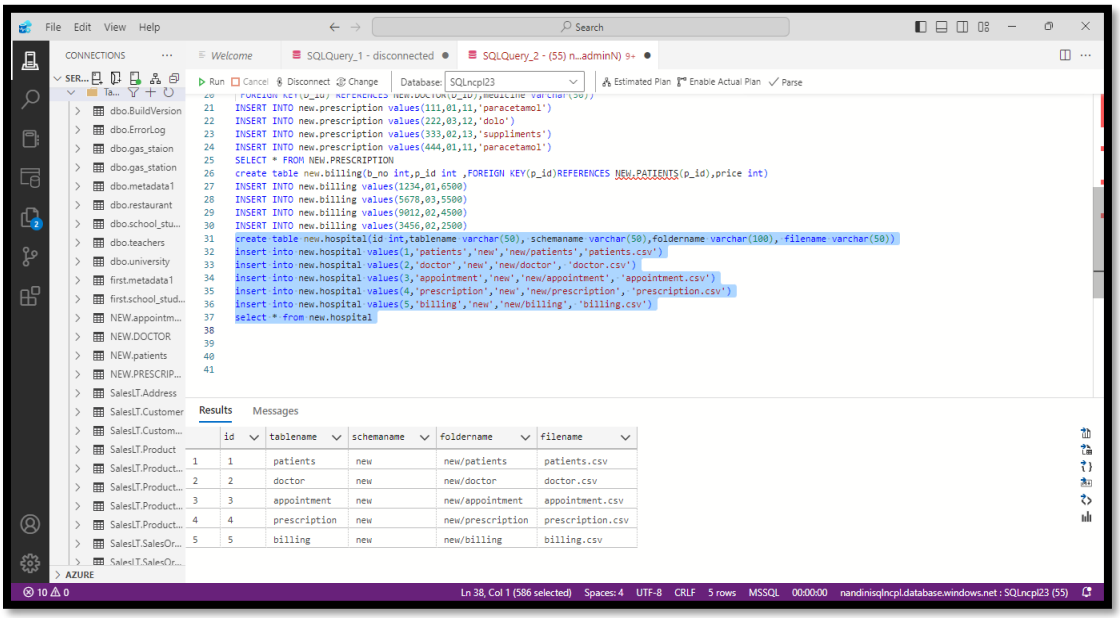
Attributes: b_no (Primary Key), p_id, price

Primary Key: b_no

Foreign Key: p_id → patients(p_id)

Description: Billing information for each patient.





12. metadata table of hospital dataset having 5 tables in it.

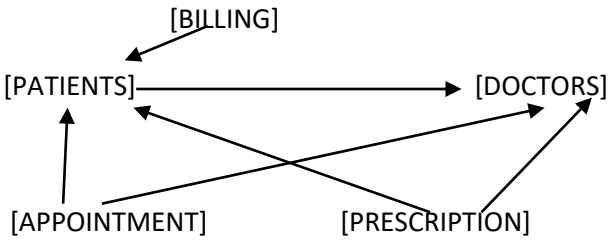
Attributes: id (Primary Key), tablename, schemaname, foldername, filename

Primary Key: id

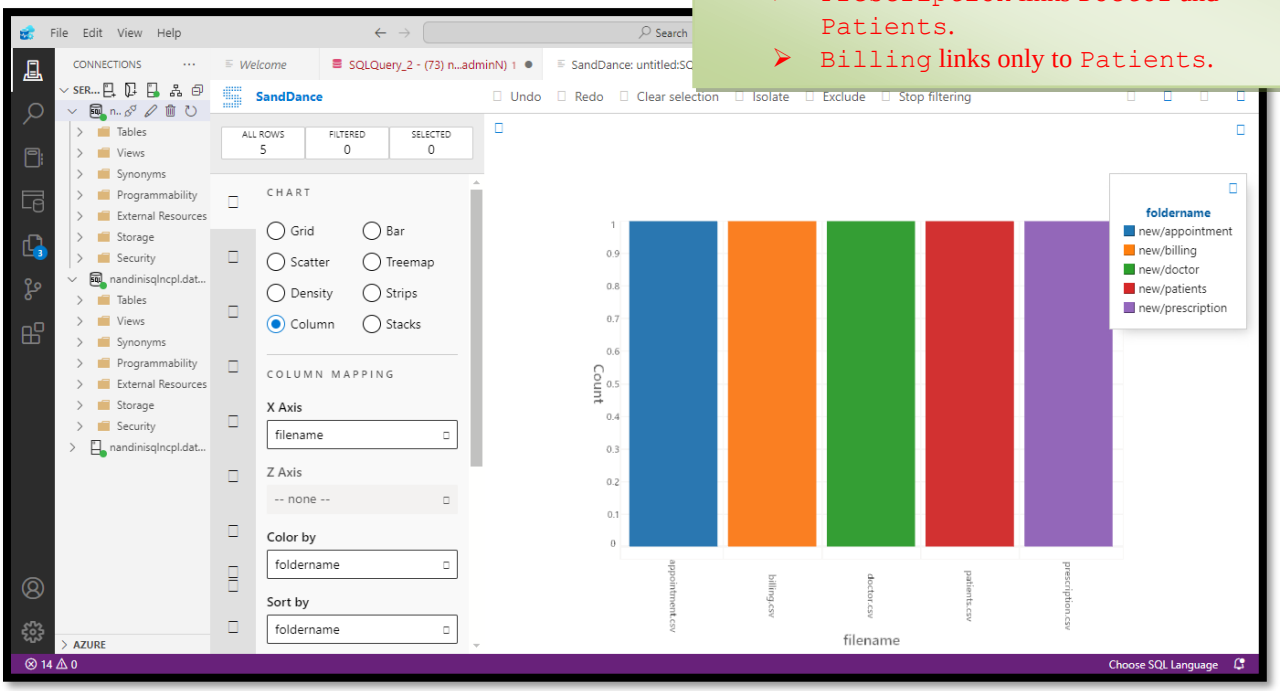
Description: Metadata table – stores info about dataset file locations.

RELATIONSHIP BETWEEN THE TABLES:->

Table	Related Table	Relationship Type
doctor	patients	Many-to-One (Each doctor is linked to a patient)
appointment	patients, doctor	Many-to-One (Appointment is linked to both)
prescription	patients, doctor	Many-to-One
billing	patients	Many-to-One
hospital	(Metadata only)	No FK relationships



- `Patients` is the **central table** (all others link to it).
- `Doctor` has a **foreign key to Patients**.
- `Appointment` connects both `Doctor` and `Patients`.
- `Prescription` links `Doctor` and `Patients`.
- `Billing` links only to `Patients`.



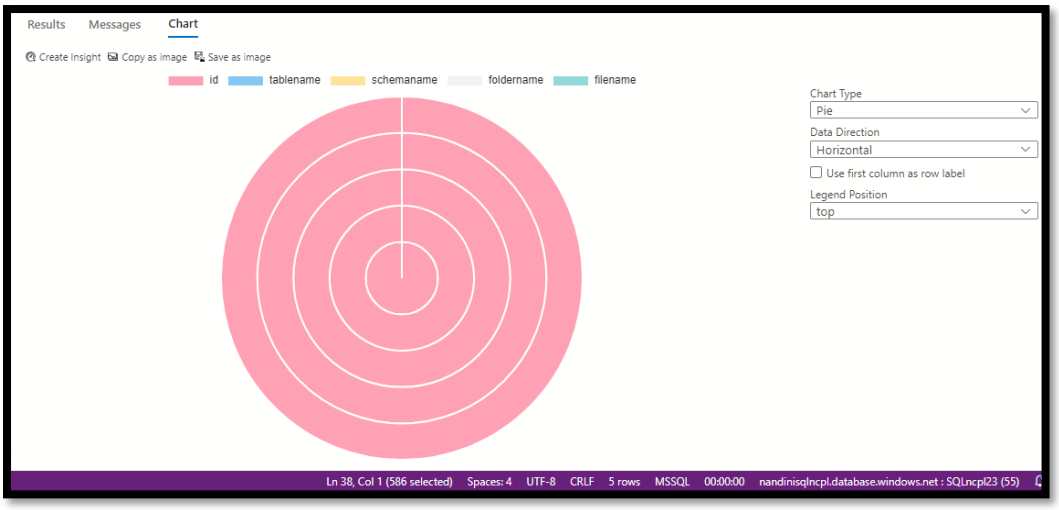
I. CHART:-> X-axis shows the **filename** (like patients.csv, doctor.csv, appointment.csv,billing.csv,prescription.csv).

II. On y-axis, It counts how many times each file is used.

III. All 5 bars are equal and count is 1 for each — that means:

- No duplicates
- No missing files
- Each file is correctly mapped to one tab

- The column chart shows a one-to-one mapping between database tables and CSV source files. Each dataset is uniquely stored under its respective folder, demonstrating organized file management and supporting schema-based data loading.



I. PIE CHART:-> It uses columns from new.hospital table: id, tablename, schemaname, foldername, filename.

II. The chart shows that each attribute has data.

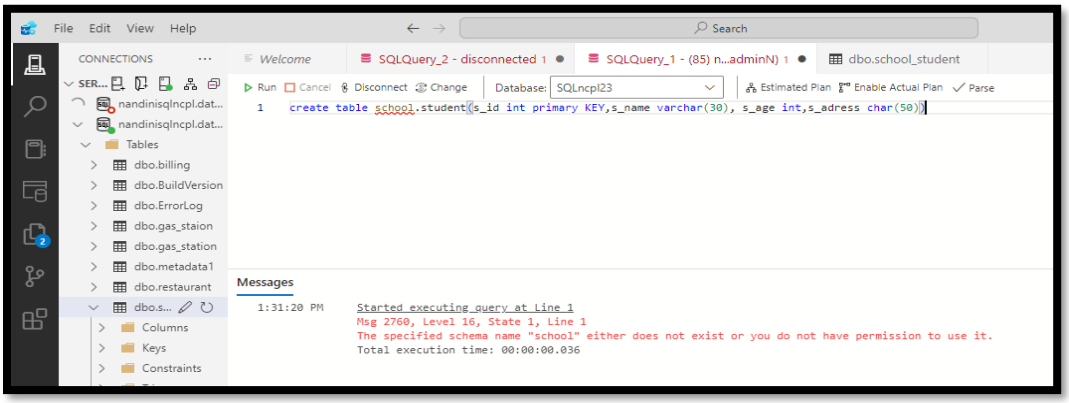
III. Each segment (color) represents one column. Since it forms a full circle, it means all values are present in all 5 rows.

IV. Metadata is **complete** — no missing values. Every table is properly linked to a folder and a `.csv` file

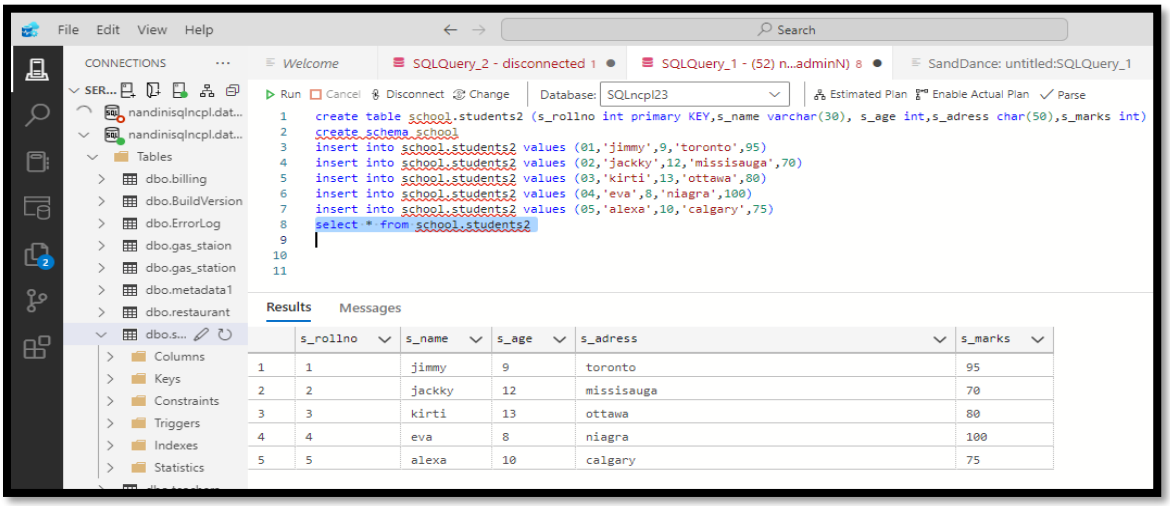
2.DATASET:->SCHOOL.METADATA

SCHEMA= SCHOOL

TABLENAME =METADATA

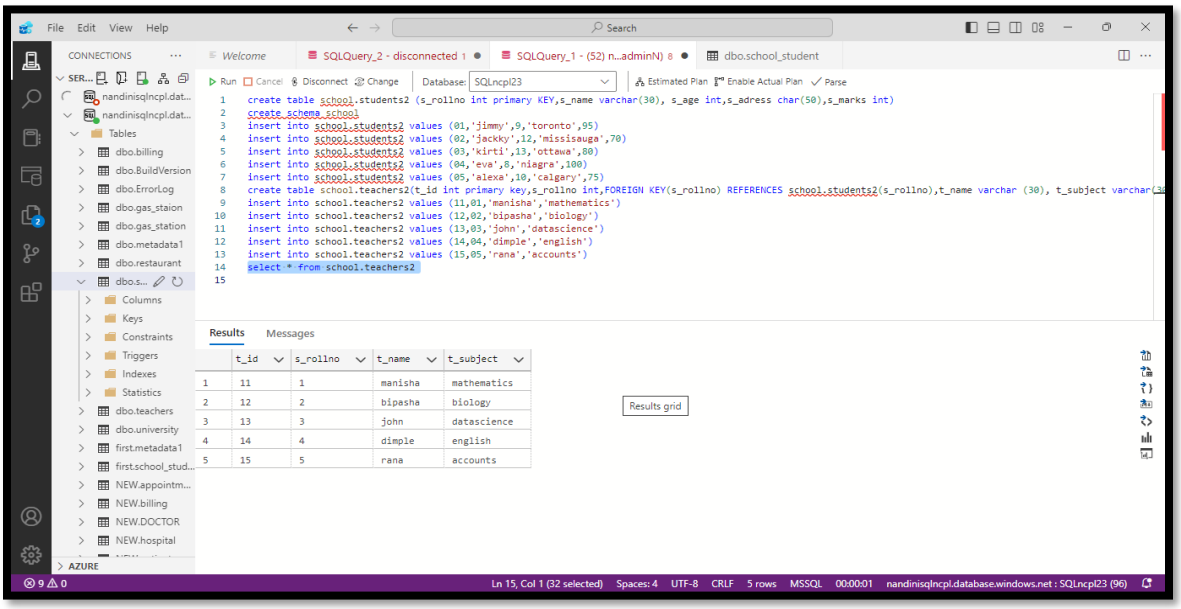


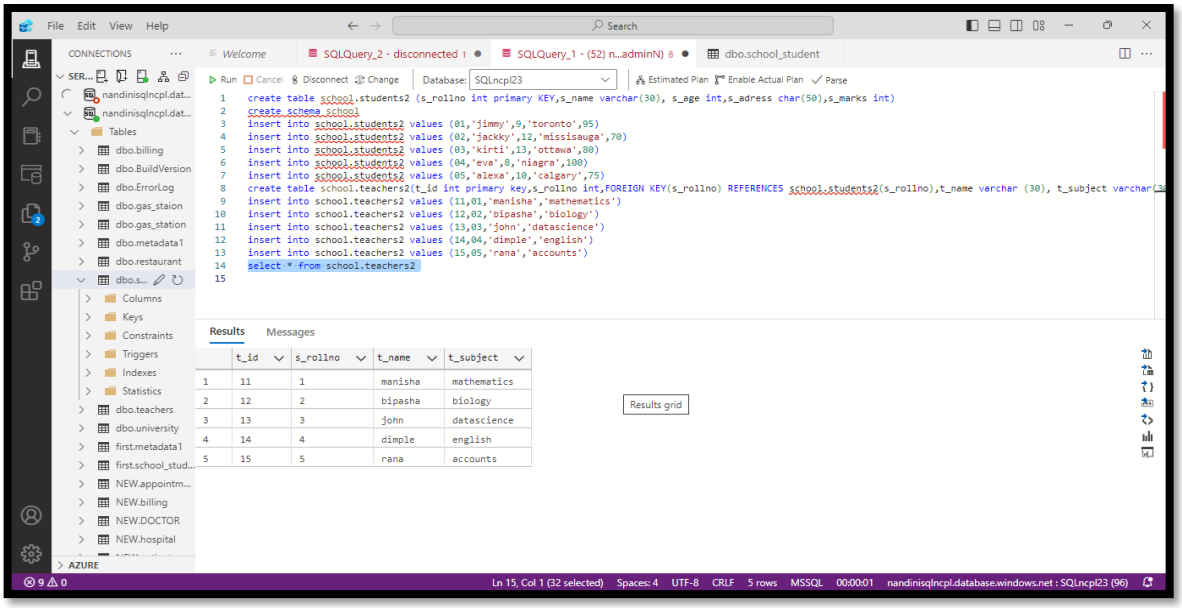
1.This error appears because first I have to create schema for table school.



2.now I create schema by using,'create schema school'.insert values into table and then use select statement.

Description: Stores appointment info about school students their rollno,name,age ,address and marks.





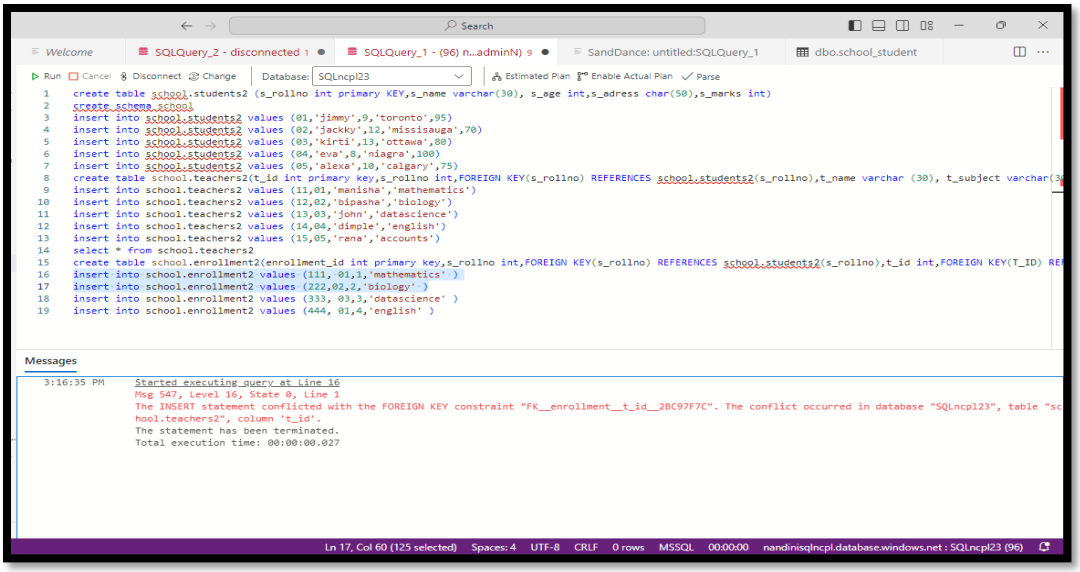
3.for creating school.teachers table by using create,insert and select statement ,contains information about teachers.

Attributes: t_id (Primary Key), s_rollno (Foreign Key), t_name, t_subject

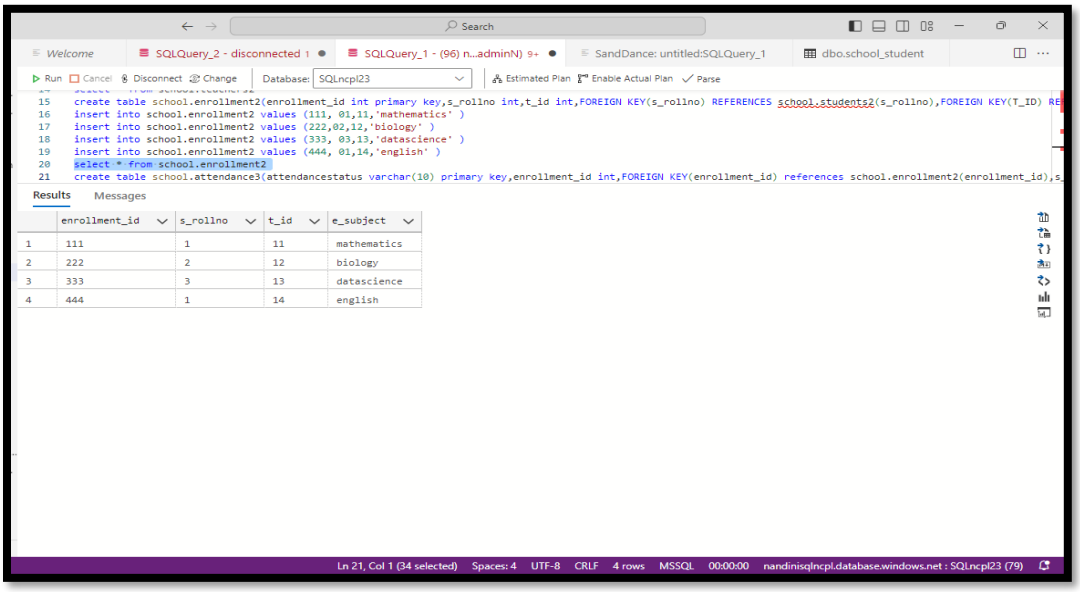
Primary Key: t_id

Foreign Key: s_rollno → students(s_rollno)

Description: Stores teachers details, and each teacher is related to a student.



4.In this I wrote wrong t_id . that's why it shows foreigh key constraints(t_id).



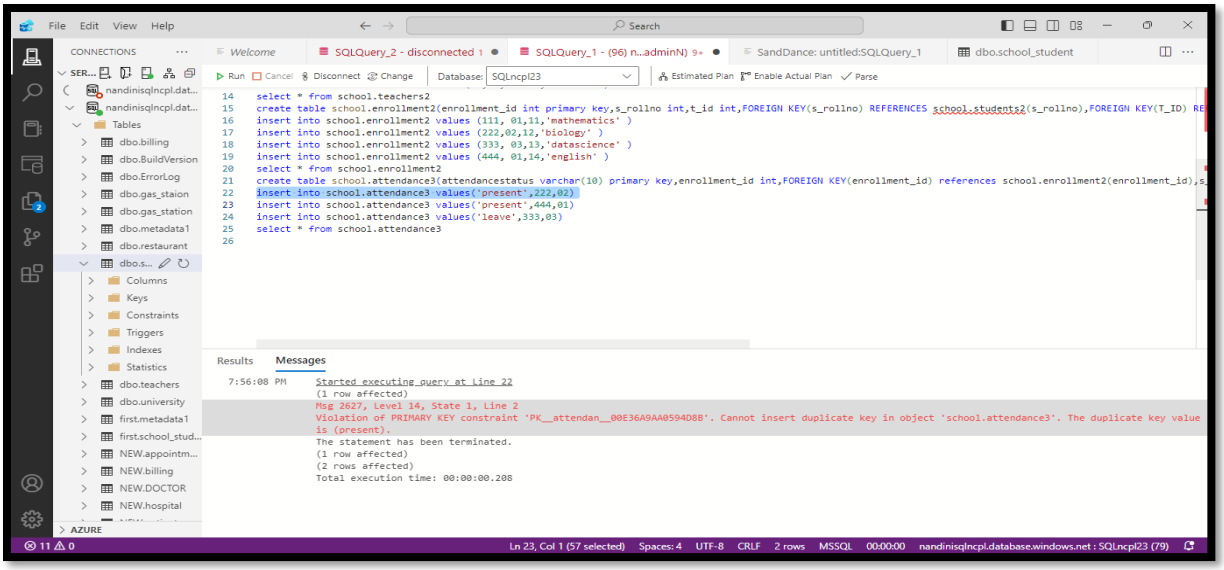
5.School.enrollment2 table.

Attributes: enrollment_id (Primary Key), t_id (Foreign Key), s_rollno,e_subject

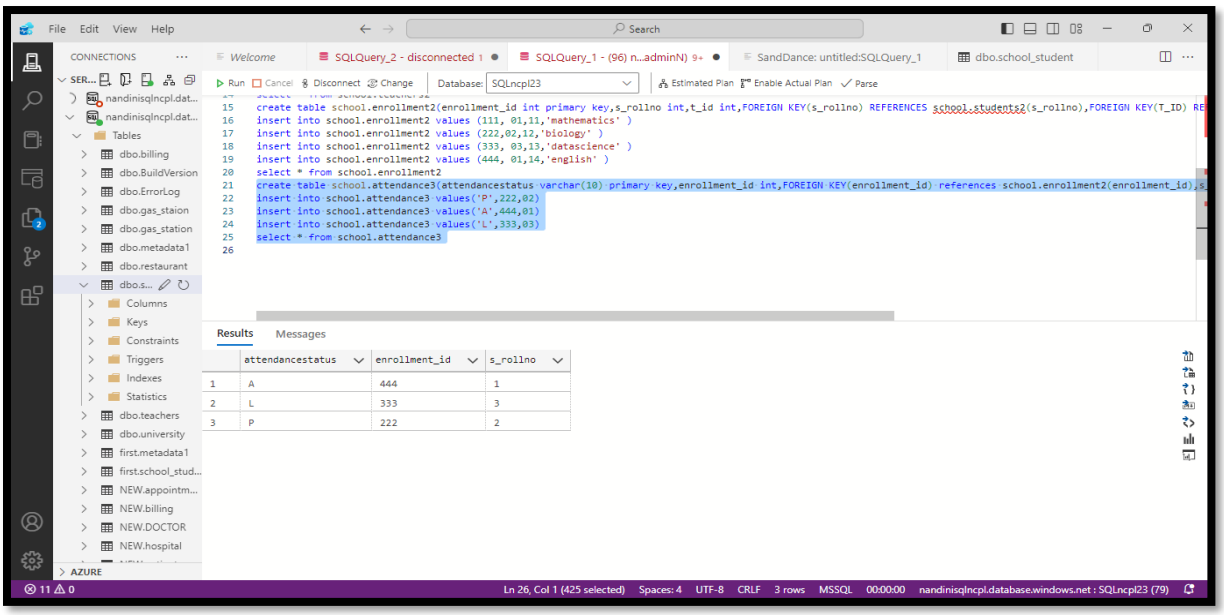
Primary Key: enrollment_id

Foreign Key: t_id → teachers(t_id),s_rollno→students(s_rollno)

Description: Stores details about ,in which teacher's subject student enroll.



6. This error shows primary key constraints, as primary key didn't contain any duplicate values. Here I put 'absent ' two times this shows an error because we can't put same values in primary key.



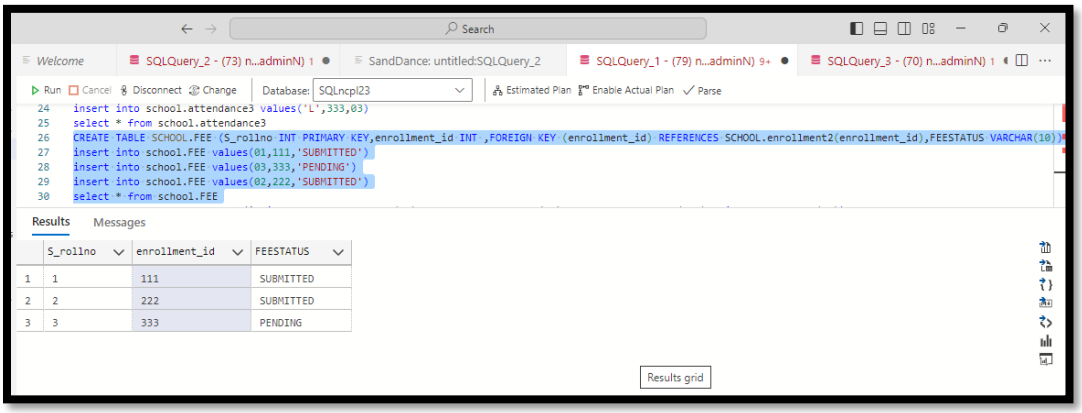
7.school.attendance3 table.

Attributes: attendancestatus(Primary Key), enrollment_id (Foreign Key), s_rollno(foreign key)

Primary Key: attendancestatus

Foreign Key: enrollment_id → enrollments(enrollment_id),s_rollno→students(s_rollno)

Description: Stores information about attendance of students which are enrolled in subjects.



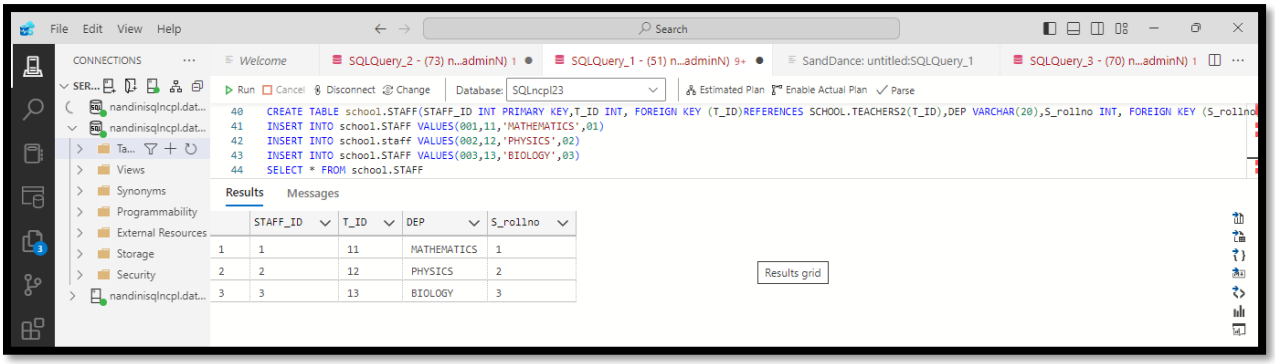
8. school.fee table

Attributes: s_rollno(Primary Key), enrollment_id (Foreign Key), feestatus.

Primary Key: s_rollno

Foreign Key: enrollment_id → enrollments(enrollment_id).

Description: Stores information about fee status of enrolled students.



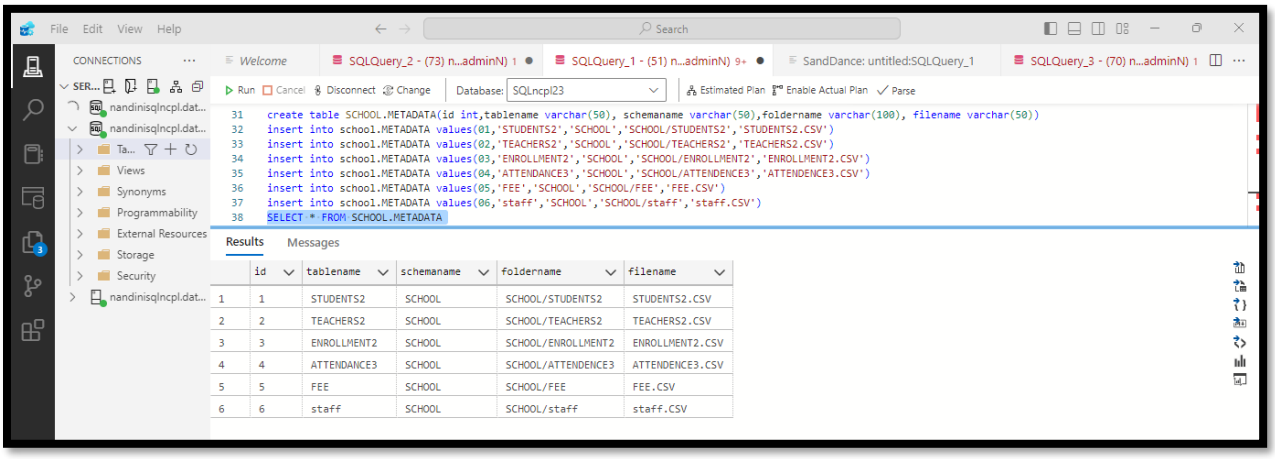
9.STAFF TABLE

Attributes:STAFF_ID (PRIMARY KEY),T_ID,DEPARTMENT,S_ROLLNO.

Primary Key: STAFF_ID

Foreign Key: T_id → TEACHERS(T_id),S_ROLLNO→STUDENTS(S_ROLLNO).

Description: Stores information about STAFF in the school including teachers who teach students for particular subject.



9. . metadata table of school dataset having 5 tables in it.

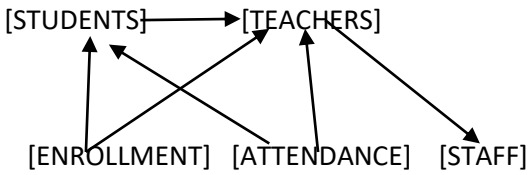
Attributes: id (Primary Key), tablename, schemaname, foldername, filename

Primary Key: id

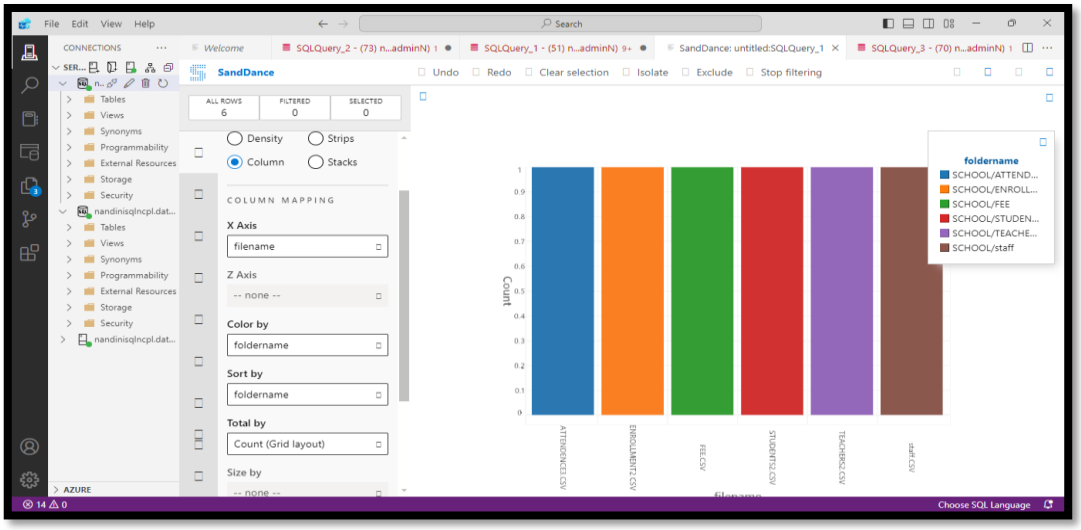
Description: Metadata table – stores info about dataset file locations.

RELATIONSHIP BETWEEN TABLES:-

Table	Related Table	Relationship Type
TEACHERS	STUDENTS	Many-to-One (Each teacher is linked to a student)
ENROLLMENT	STUDENTS, TEACHERS	Many-to-One (enrollment is linked to both)
FEE	STUDENTS, TEACHERS	Many-to-One
ATTENDANCE	STUDENTS	Many-to-One
STAFF	STUDENTS,TEACHERS	Many-to-One
METADATA	ALL THE TABLES	No FK relationships



- STUDENTS is the **central table** (all others link to it).
- TEACHERS has a **foreign key to STUDENTS**.
- ENROLLMENT connects both TEACHERS and STUDENTS.
- ATTENDANCE links TEACHERS and STUDENTS.
- STAFF links only to TEACHERS.

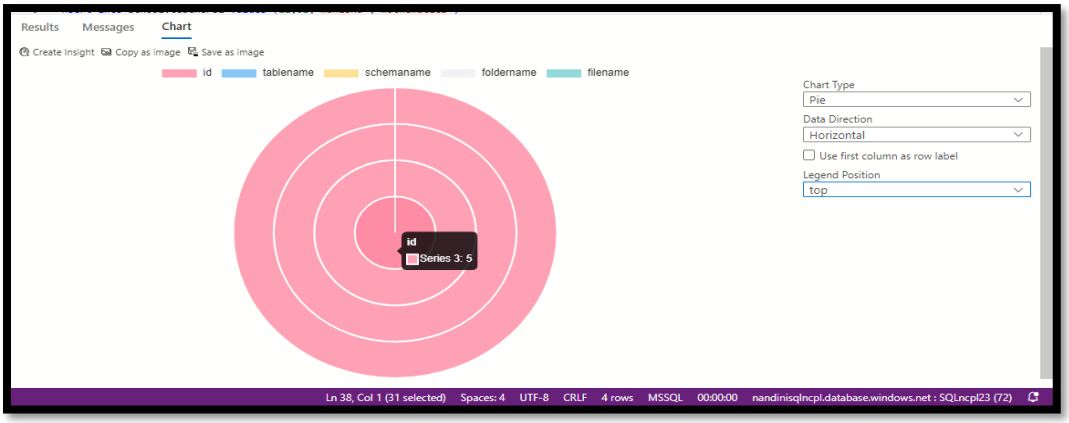


10.

CHART:-> X-axis shows the **filename** (like students.csv, teachers.csv, enrollment.csv,fee.csv,staff.csv).

- On y-axis, It counts how many times each file is used.
- All 5 bars are equal and count is 1 for each — that means:

- No duplicates
- No missing files
- Each file is correctly mapped to one tab
- The column chart shows a one-to-one mapping between database tables and CSV source files. Each dataset is uniquely stored under its respective folder, demonstrating organized file management and supporting schema-based data loading.



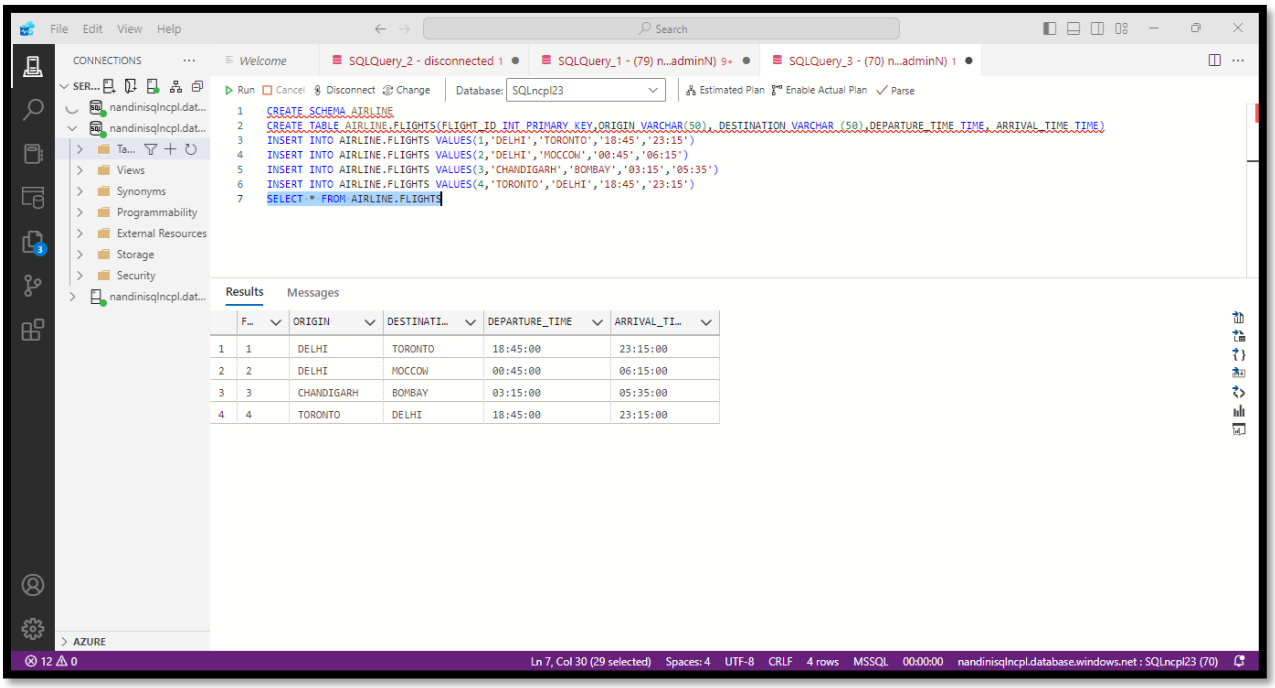
I. PIE CHART:-> It uses columns from new.hospital table: id, tablename, schemaname, foldername, filename.

II. The chart shows that each attribute has data.

III. Each segment (color) represents one column. Since it forms a full circle, it means all values are present in all 5 rows.

IV. Metadata is **complete** — no missing values. Every table is properly linked to a folder and a `.csv` file

3 DATASET:->AIRLINE.METADATA6

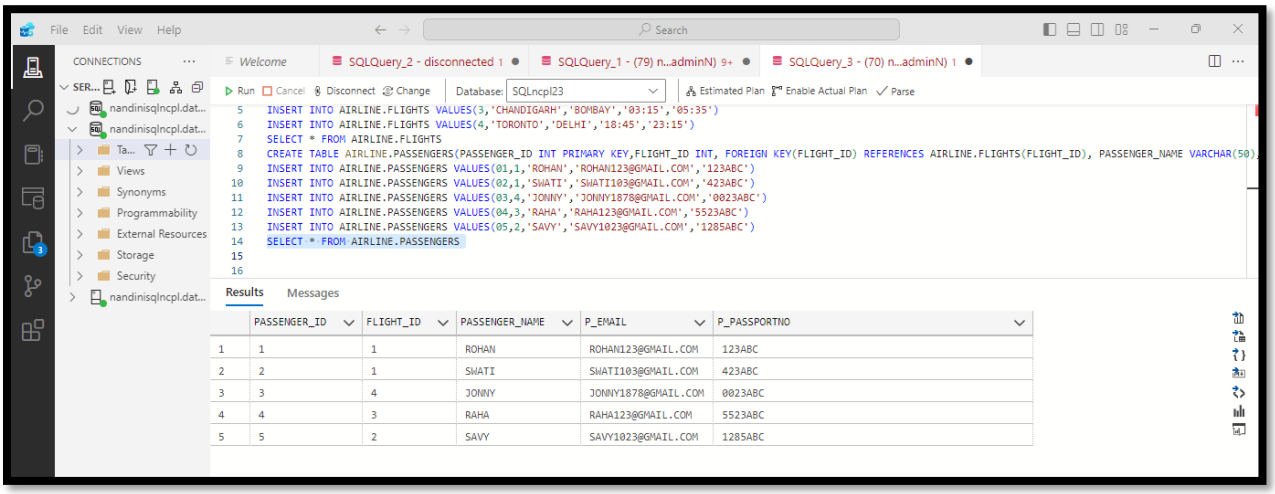


1.creating schema airline first. Then insert values in table airline.flights.

Attributes:flight_ID (PRIMARY KEY),origin,departure time,arrival time..

Primary Key: flight_id

Description: Stores information about flights.

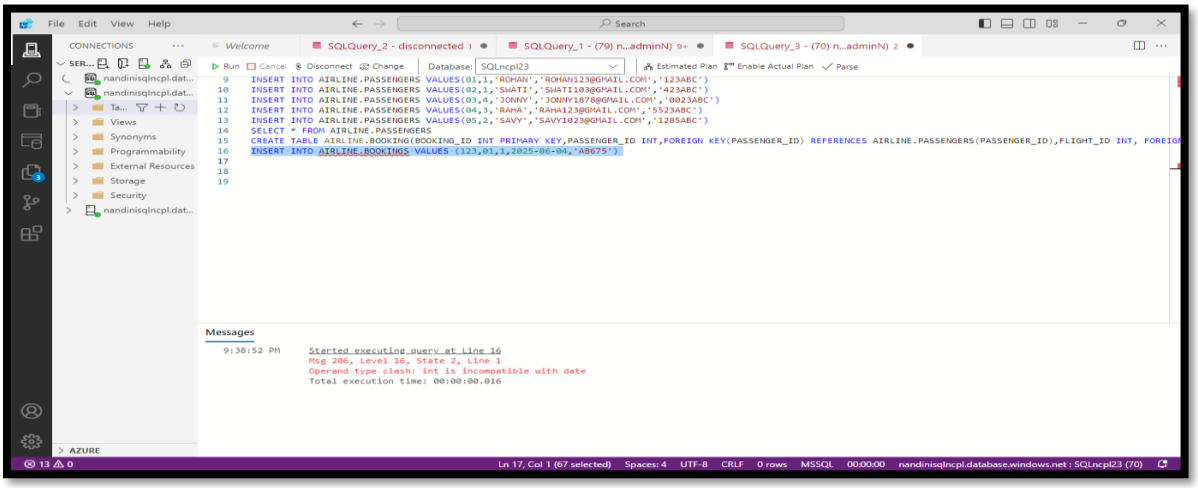


’**Attributes:**passenger_ID (PRIMARY KEY),flight_ID,p_email,p_passportno

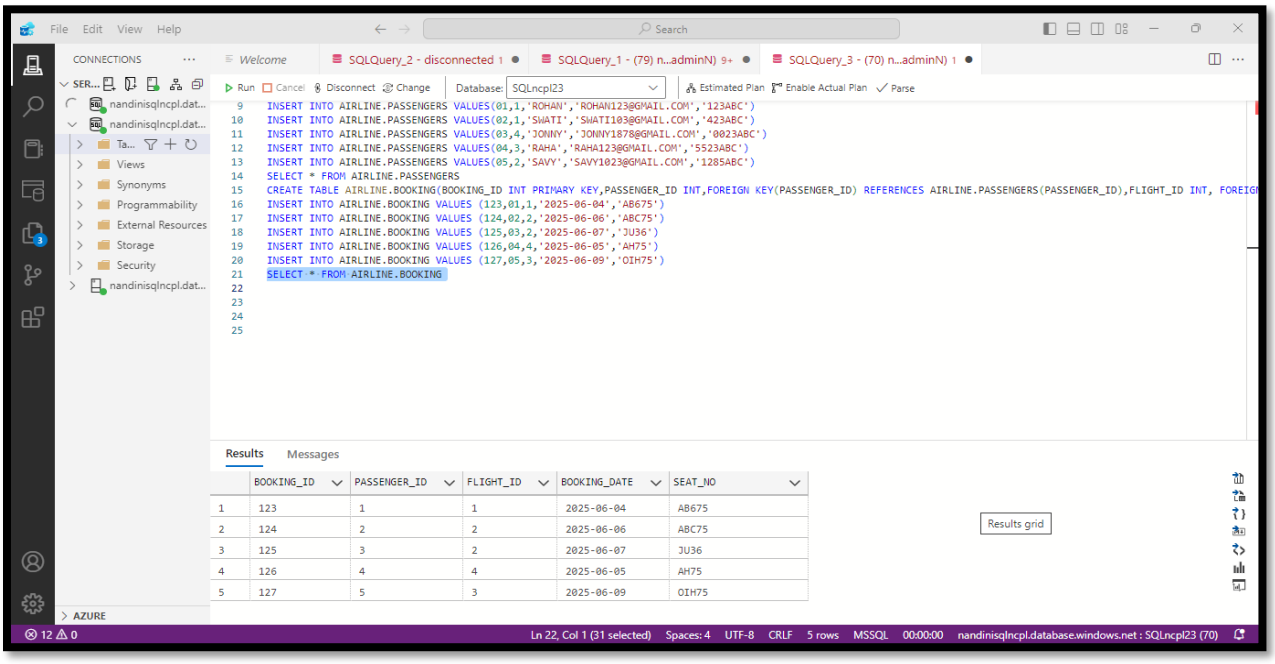
Primary Key: PASSENGER_ID

Foreign Key: FLIGHT_id →FLIGHT(FLIGHT_ID)

Description: Stores information about PASSENGERS , WHICH FLIGHTTHEY BOOK FOR TRAVELLING AT WHAT TIME.



HERE DATE SHOULD BE IN ‘ ‘.LIKE’2025-06-04’.



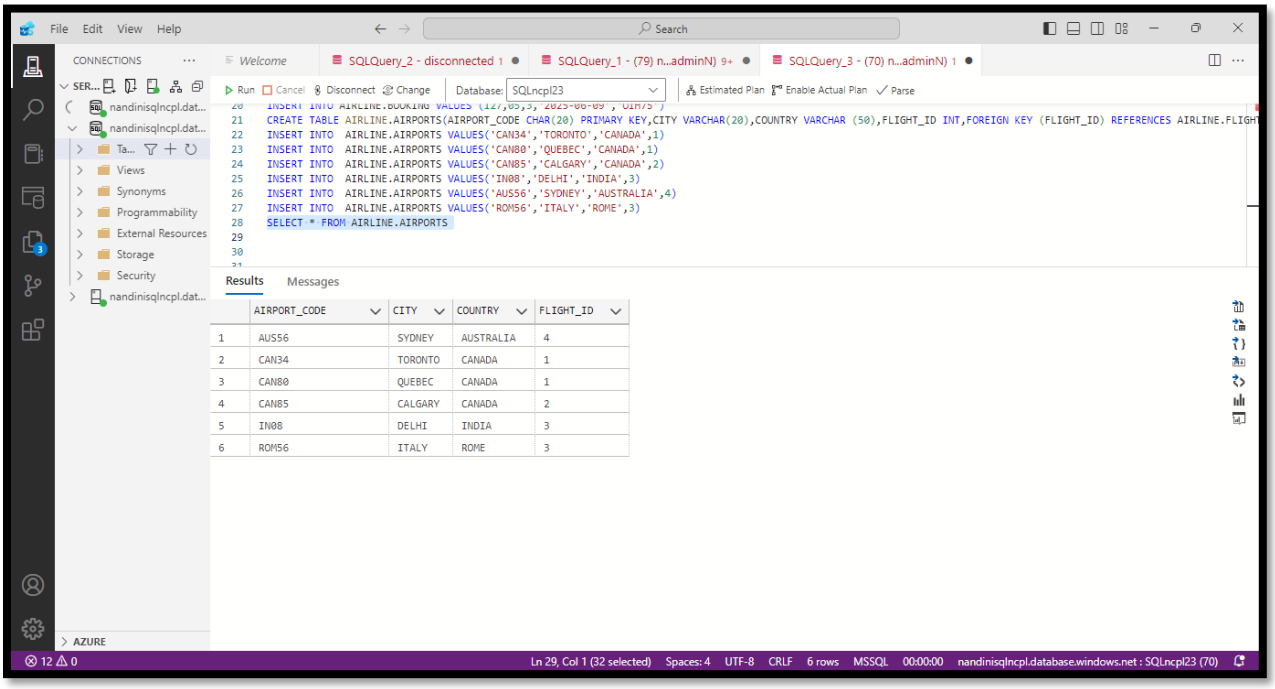
3. BOOKING TABLE:→

Attributes: BOOKING _ID (PRIMARY KEY),PASSENGER_IDFLIGHT_ID,BOOKING_DATE,SEAT_NO.

Primary Key: BOOKING**_ID**

Foreign Key: PASSENGER_ID,FLIGHT_ID.

Description: Stores information about BOOKING OF FLIGHT BY PASSENGER ,DETAILS ABOUT SEAT NO. AND DATE.



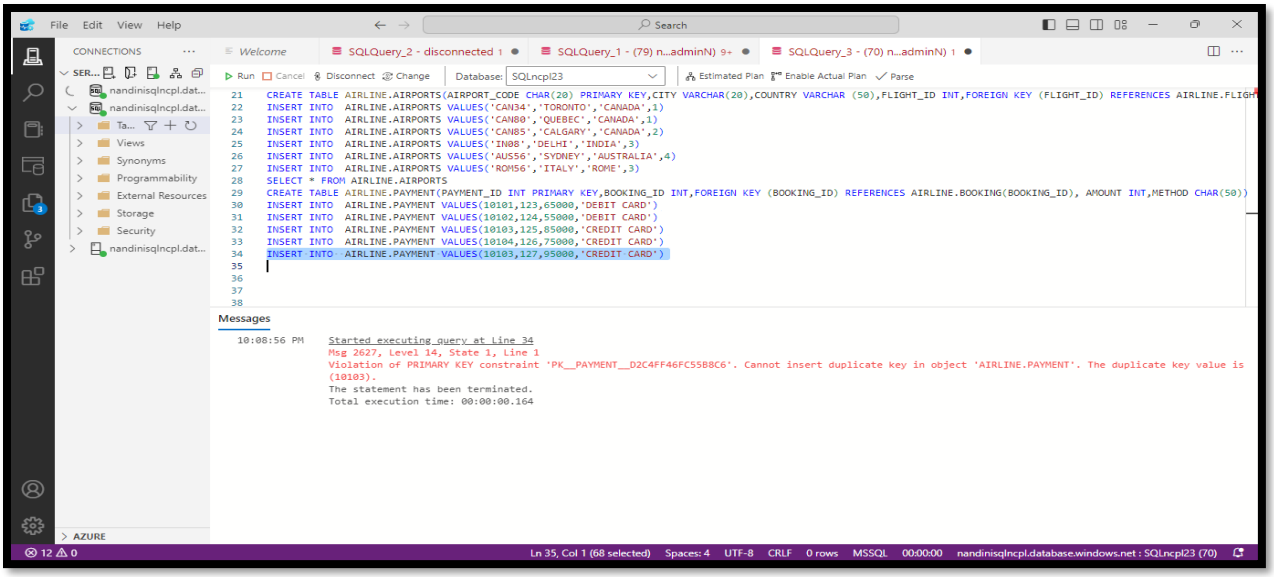
4.AIRPORTS TABLE:->

Attributes: AIRPORT_CODE (PRIMARY KEY),CITY,COUNTRY,FLIGHT_ID

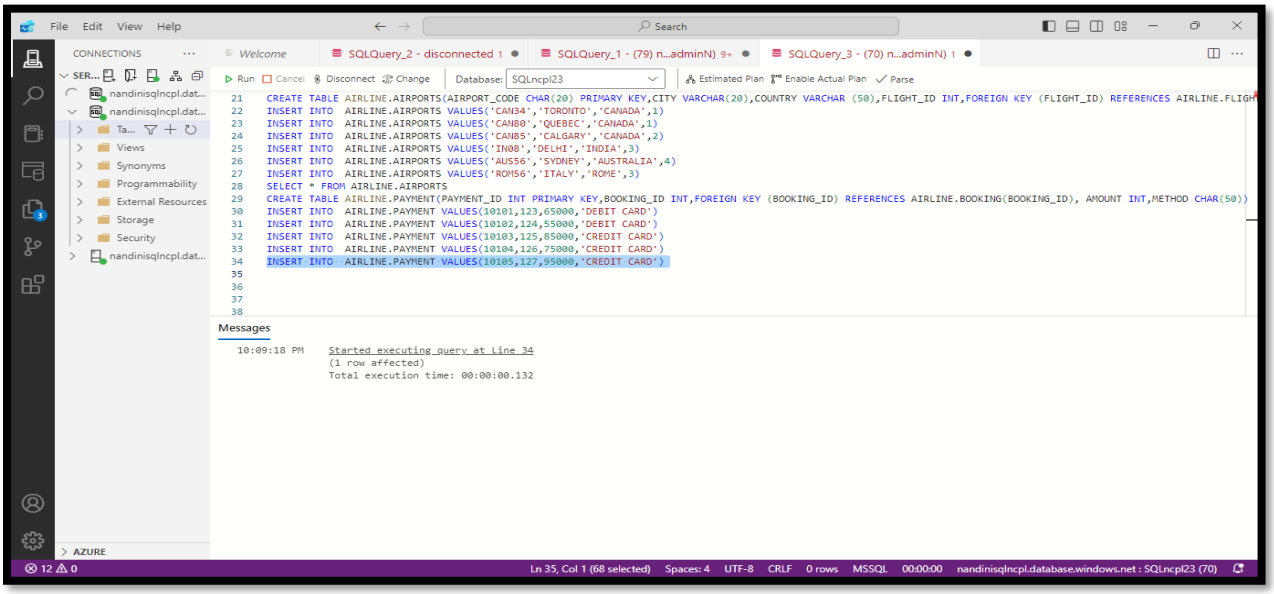
Primary Key: AIRPORT_CODE

Foreign Key:->FLIGHT_ID.

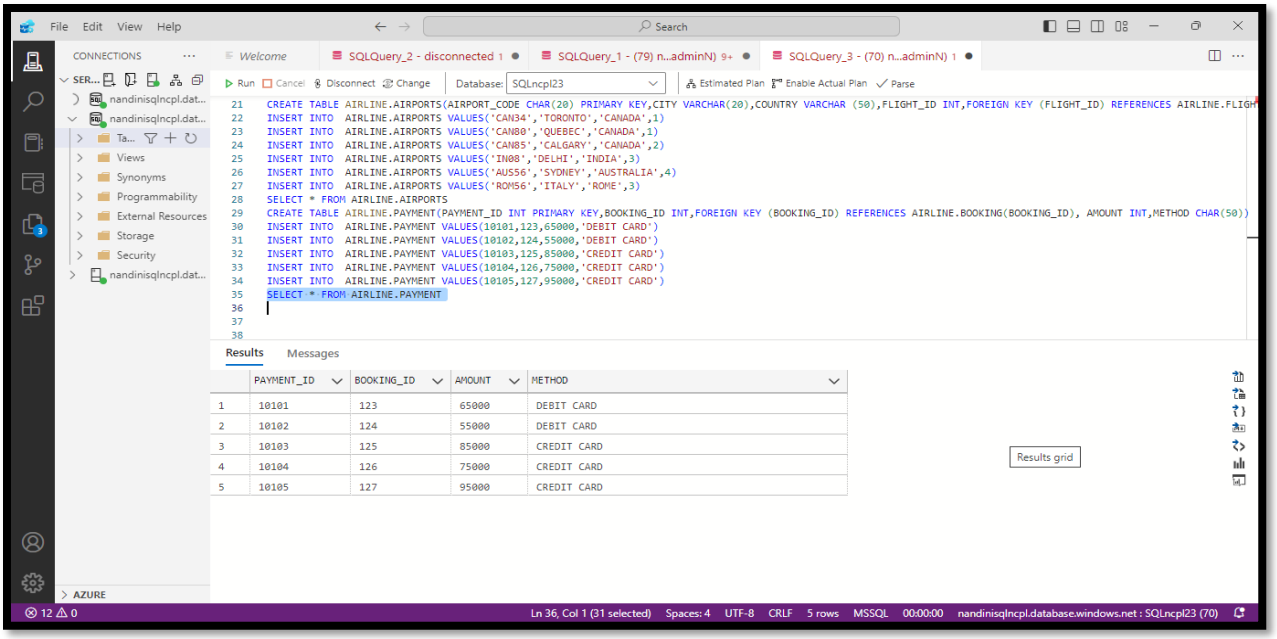
Description: Stores information about AIRPORTS OF COUNTRY HAVING FLIGHT_ID.



HERE ERROR BECAUSE I WROTE SAME PAYMENT_ID WHICH IS A PRIMARY KEY.



5.CREATE,INSERT STATEMENT OF TABLE AIRLINE.PAYMENT.

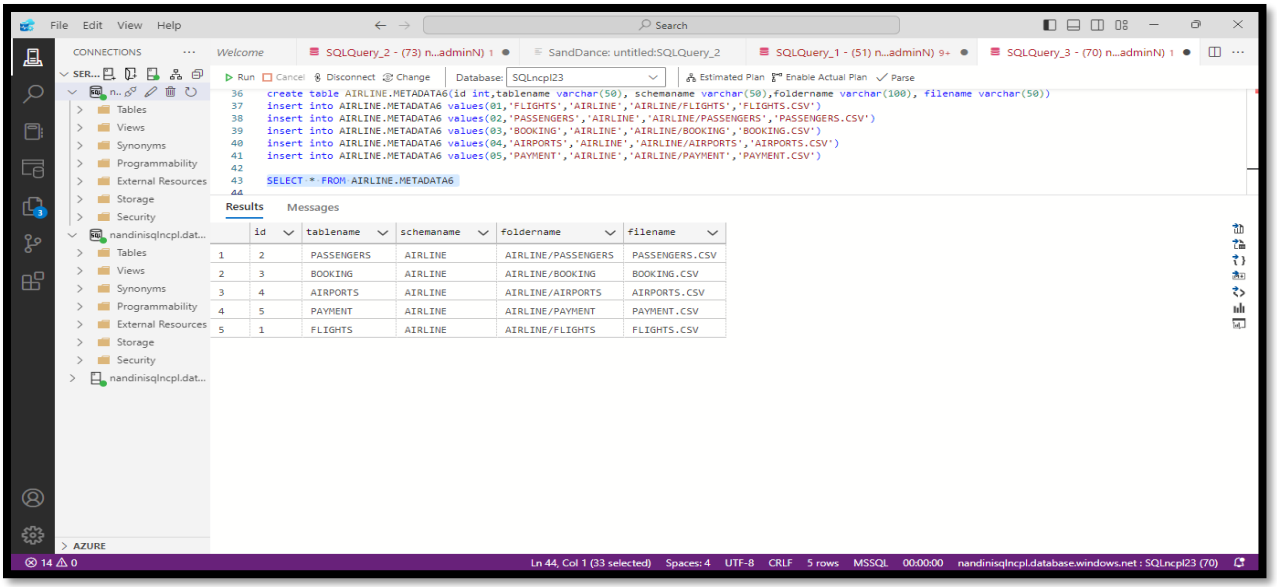


Attributes: PAYMENT_ID PRIMARY KEY,BOOKING_ID,AMOUNT,METHOD.

Primary Key: PAYMENT_ID

Foreign Key:->BOOKING_ID

Description: Stores information about PAYMENT METHOD FOR BOOKING FLIGHT.



6. . metadata table of school dataset having 5 tables in it.

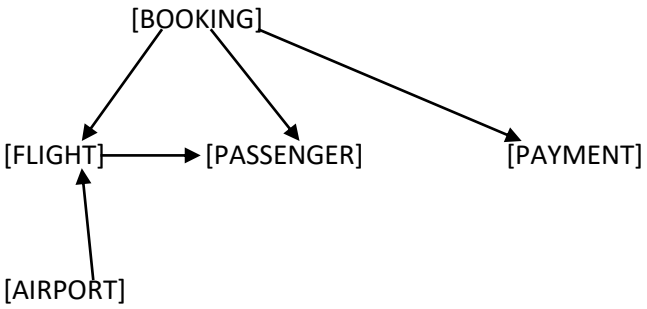
Attributes: id (Primary Key), tablename, schemaname, foldername, filename

Primary Key: id

Description: Metadata table – stores info about dataset file locations.

RELATIONSHIP BETWEEN TABLES:->

TABLES	Related Table	Relationship Type
FLIGHT		ALL tables link ro flight table.
PASSENGER	FLIGHT	Many-to-One (Each PASSENGER is linked to a FLIGHT)
AIRPORT	FLIGHT	Many-to-One
BOOKING	FLIGHT, PASSENGER	Many-to-One
PAYMENT	BOOKING	Many-to-One
METADATA	ALL THE TABLES	No FK relationships



- FLIGHT is the **central table** (all others link to it).
- PASSENGER has a **foreign key to FLIGHT**.
- BOOKING connects both FLIGHT and PASSENGER.
- PAYMENT LINKS WITH BOOKING.
- AIRPORT links only to FLIGHT.

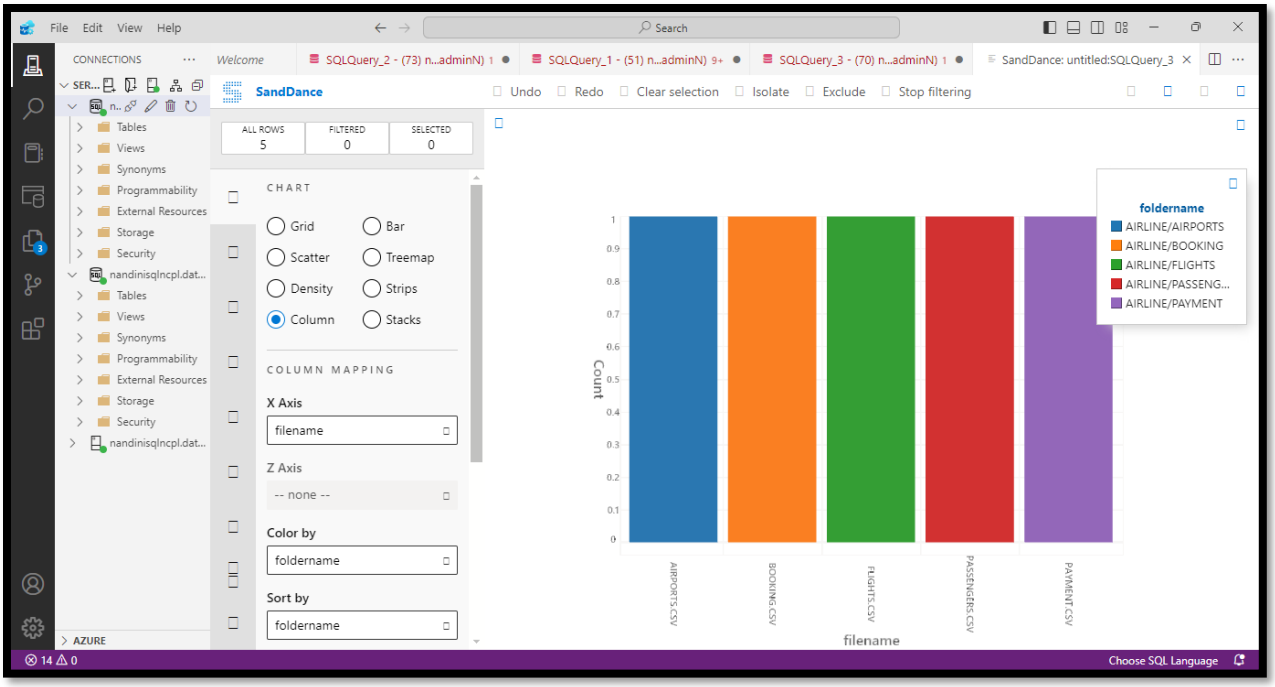
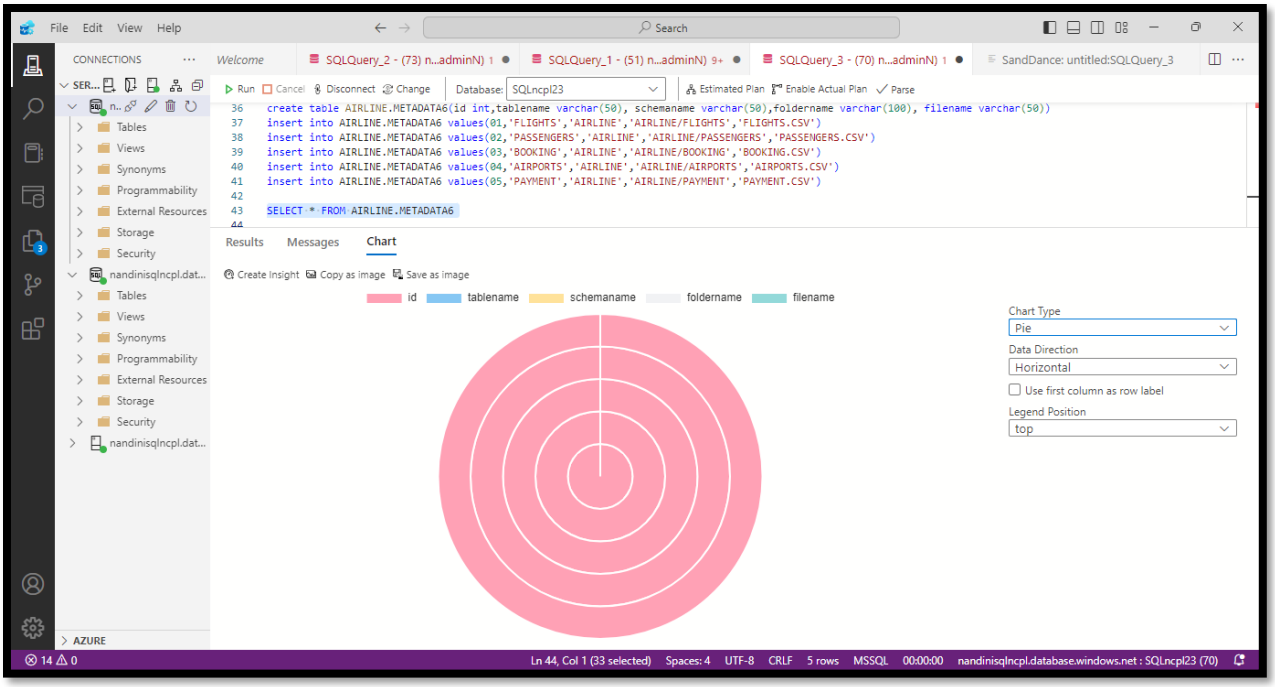


CHART:-> X-axis shows the filename (like flights.csv, passengers.csv, airports.csv, payments.csv, booking.csv).

On y-axis, It counts how many times each file is used.

All 5 bars are equal and count is 1 for each — that means:

- No duplicates
- No missing files
- Each file is correctly mapped to one tab
- The column chart shows a one-to-one mapping between database tables and CSV source files. Each dataset is uniquely stored under its respective folder, demonstrating organized file management and supporting schema-based data loading.



I. PIE CHART:-> It uses columns from new.hospital table: id, tablename, schemaname, foldername, filename.

II. The chart shows that each attribute has data.

III. Each segment (color) represents one column. Since it forms a full circle, it means all values are present in all 5 rows.

IV. Metadata is **complete** — no missing values. Every table is properly linked to a folder and a `.csv` file.

COMPARATIVE ANALYSIS

ADVATAGES AND DISADVANTAGES OF HAVING SCHEMA

ADVANTAGES:-

1.ORGANISED DATA

- Tables have a fixed structure (columns, data types).
- Easy to understand and manage.

2.DATA QUALITY

- Only correct type of data is allowed.
- Reduces errors in data entry.

3. Supports Relationships

- Can link tables using Primary and Foreign Keys.
- Helps in joining data for deeper analysis.

4.Improves Performance

- Faster querying and indexing.
- Efficient data storage and retrieval.

5.Easier for Analysis

- Clear structure helps tools read and analyze data.
- Useful for reports and dashboards.

6.Enforces Security

- Access control and validation are easier with schema.

DISADVANTAGES OF NOT HAVING SCHEMA

1. **Unorganized Data**
   - Data can be messy and unstructured
   - No fixed format, making it hard to understand
2. **Low Data Quality**
   - No rules for data types (e.g., age can be text)
   - More chances of errors and duplicates
3. **Difficult to Analyze**
   - Tools cannot easily read or process unstructured data
   - Data cleaning takes more time
4. **No Table Relationships**
   - Cannot link data from different sources
   - Makes it hard to do joins or complex queries
5. **Slower Performance**
   - Searching or filtering takes longer

- No indexes or optimization possible

6. **Security Risks**
    - No validation or control over what data is entered
    - Harder to manage access and permissions
7. **Hard to Scale**
    - As data grows, it becomes more difficult to manage without structure.

Recommendations based on analysis:-

1.Always use schema for structured data. It helps keep data clean, organized, and easy to understand.Makes data analysis and reporting much faster and more accurate.

2. Design the Schema Before Loading Data.Plan tables, primary keys, and relationships in advance.This avoids errors and saves time later.

3. Use Metadata Tables.Store information like file names, folder paths, and table names.Helps manage and track data sources easily.metadata tables makes work easy.

4. Apply Primary and Foreign Keys.Link related tables using keys to support data relationships. Useful for joining data in reports and dashboards.

5. Use Schema Validation Tools.Use tools in SQL or data platforms to check if data matches the schema.Prevents wrong data from entering the system.

Impact on Data Processing and Analysis

- **With Schema**
  Fast and structured processing.
  Clean joins and groupings.
  Works well with BI tools (Power BI, Tableau).
- **Without Schema**
  Slower processing.
  Hard to clean and analyze.
  Risk of incorrect or incomplete results.

REFERENCES:--

https://www.geeksforgeeks.org/foreign-key-constraint-in-sql/

https://www.geeksforgeeks.org/dbms-integrity-constraints/

https://www.geeksforgeeks.org/create-schema-in-sql-server/

THANKYOU!