GST Filing Tracker

Submitted by

Adarsh Sharma (RA2211003011653)
Tanisa De (RA2211003011649)
Raghav Gupta (RA2211003011681)

Under the Guidance of

Mr. ANAND M

(Assistant Professor , Department of Computing Technologies)

In partial fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY
of
COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR– 603 203

APRIL - 2024

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTING TECHNOLOGIES
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Project Report titled "GST Filing Tracker" is the bonafide work done by Raghav Gupta (RA2211003011681) , Tanisa De (RA2211003011649) and Adarsh Sharma (RA2211003011653) who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Mr. ANAND M
DBMS– Course Faculty
Assistant Professor
Department of Computing Technologies

SIGNATURE
Dr. M. PUSHPALATHA
Head of the Department
Department of Computing Technologies

GST Filing Tracker

Abstract:

The GST Filing Tracker is a comprehensive web-based application designed to streamline the process of Goods and Services Tax (GST) compliance for businesses. In today's complex regulatory environment, timely and accurate GST filing is crucial for organizations to maintain compliance and avoid penalties. This project aims to address the challenges faced by businesses in managing their GST obligations by providing a centralized platform for tracking, managing, and reporting GST filings.

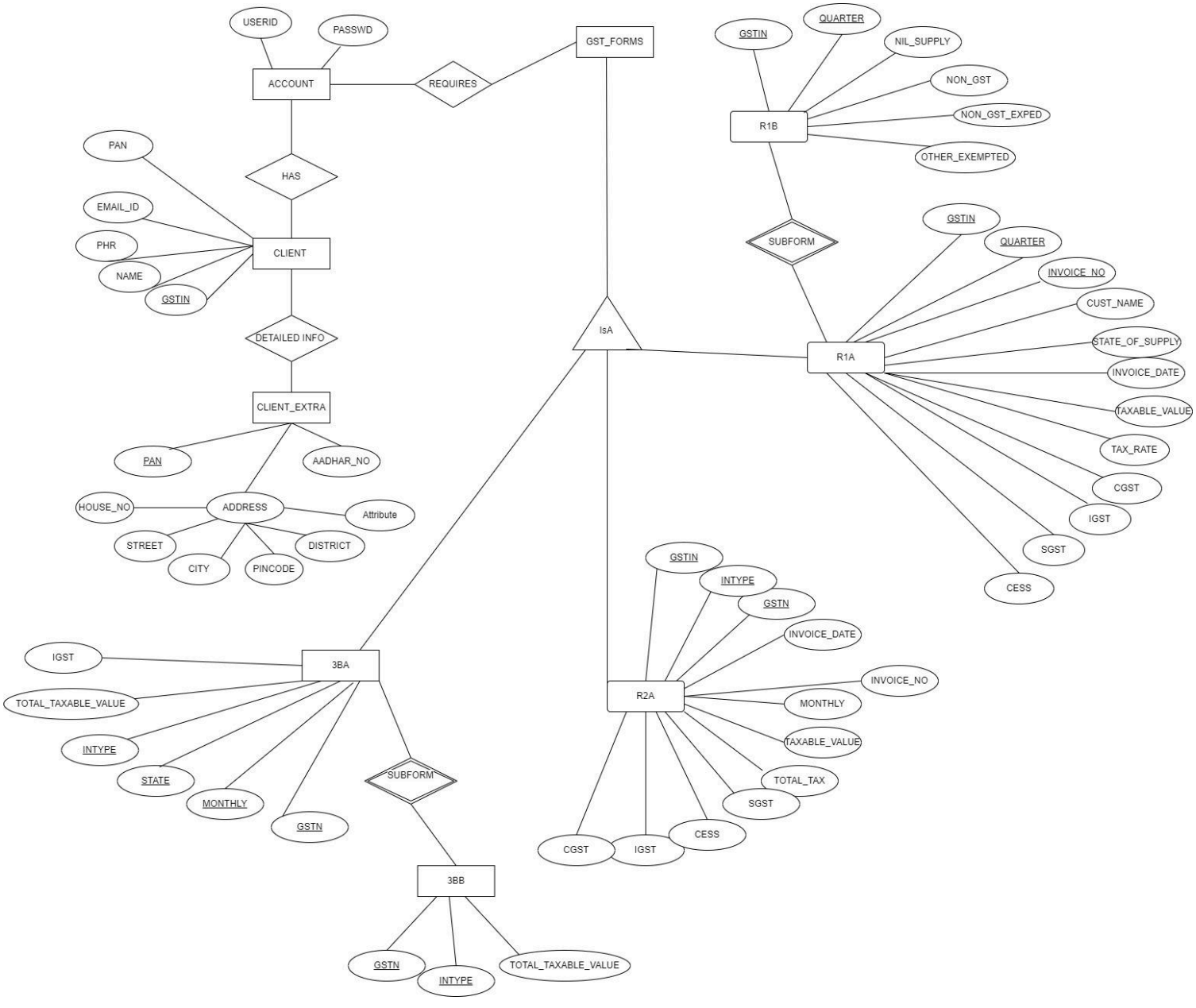
The system offers a user-friendly interface accessible to registered users, facilitating secure authentication and personalized dashboards. Users can input details of their GST filings, including relevant tax information, filing periods, and supporting documentation. Deadline notifications ensure that users stay informed about upcoming filing deadlines, reducing the risk of missed submissions.

Key features of the GST Filing Tracker include robust data validation to ensure the accuracy and integrity of information entered by users. Additionally, the system maintains an audit trail of all user activities, ensuring transparency and accountability in the filing process.

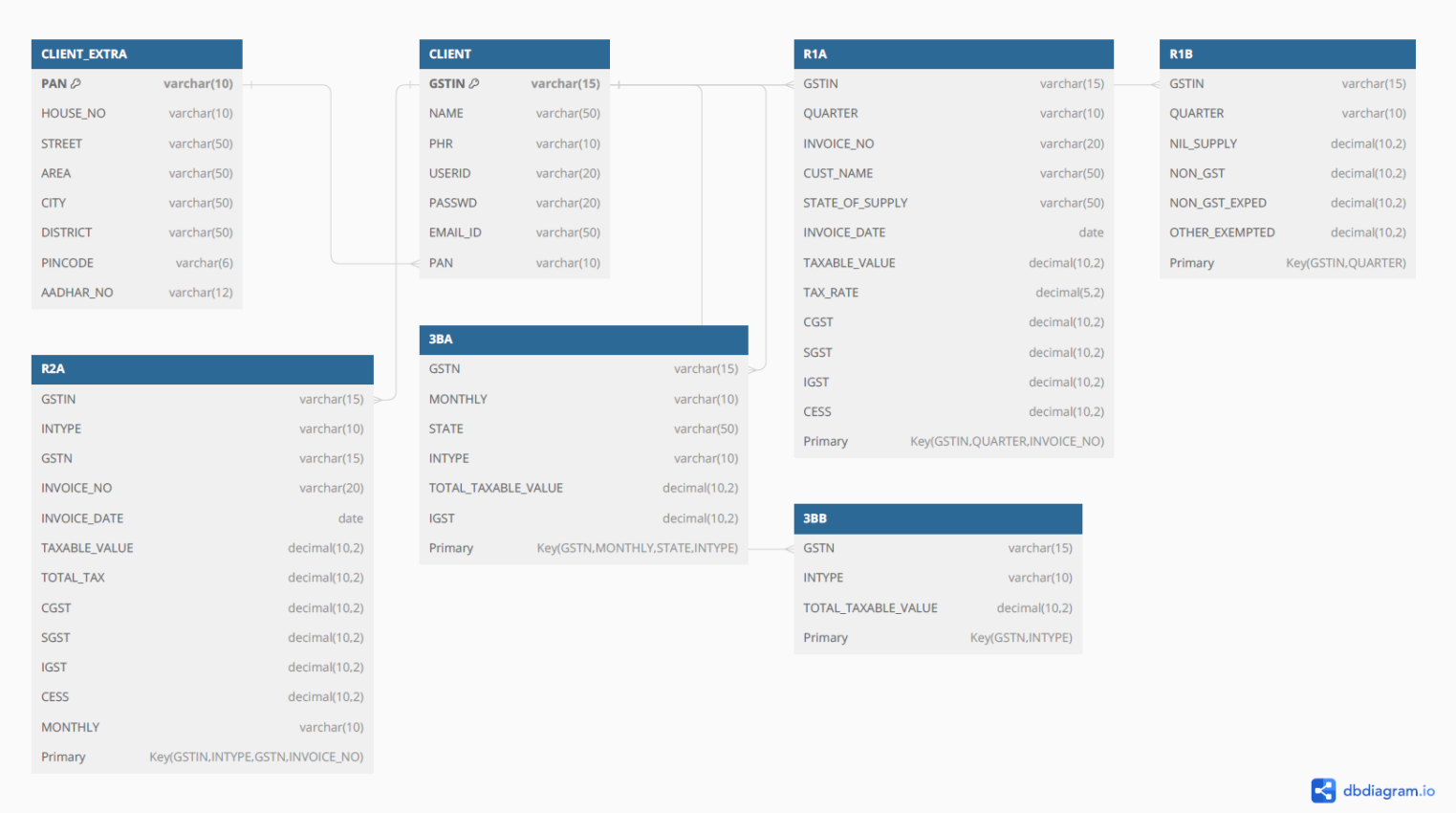
The project also emphasizes the importance of data security and compliance with regulatory requirements. Measures are implemented to safeguard sensitive information and protect user privacy.

Overall, the GST Filing Tracker project aims to empower businesses with a reliable and efficient solution for managing their GST compliance obligations. By providing a centralized platform for tracking and managing GST filings, this system helps businesses streamline their tax processes and maintain regulatory compliance.

ER DIAGRAM:


USERID
PASSWD
ACCOUNT
REQUIRES
GST_FORMS
PAN
HAS
EMAIL_ID
PHR
NAME
GSTIN
CLIENT
DETAILED INFO
CLIENT_EXTRA
PAN
AADHAR_NO
HOUSE_NO
ADDRESS
Attribute
STREET
DISTRICT
CITY
PINCODE
GSTIN
QUARTER
NIL_SUPPLY
NON_GST
NON_GST_EXPED
OTHER_EXEMPTED
R1B
SUBFORM
GSTIN
QUARTER
INVOICE_NO
CUST_NAME
STATE_OF_SUPPLY
INVOICE_DATE
TAXABLE_VALUE
TAX_RATE
CGST
IGST
SGST
CESS
R1A
IsA
GSTIN
INTYPE
GSTN
INVOICE_DATE
INVOICE_NO
MONTHLY
TAXABLE_VALUE
TOTAL_TAX
SGST
CESS
IGST
CGST
R2A
IGST
TOTAL_TAXABLE_VALUE
INTYPE
STATE
MONTHLY
GSTN
3BA
SUBFORM
3BB
GSTN
INTYPE
TOTAL_TAXABLE_VALUE

Relational Table:



Relational schema Algebra

1. CLIENT_EXTRA table:

CLIENT_EXTRA(PAN, HOUSE_NO, STREET, AREA, CITY, DISTRICT, PINCODE, AADHAR_NO)

2. CLIENT table:

CLIENT(GSTIN, NAME, PHR, USERID, PASSWD, EMAIL_ID, PAN)

3. R1A table:

R1A(GSTIN, QUARTER, INVOICE_NO, CUST_NAME, STATE_OF_SUPPLY, INVOICE_DATE, TAXABLE_VALUE, TAX_RATE, CGST, SGST, IGST, CESS)

4. R1B table:

R1B(GSTIN, QUARTER, NIL_SUPPLY, NON_GST, NON_GST_EXPED, OTHER_EXEMPTED)

5. R2A table:

R2A(GSTIN, INTYPE, GSTN, INVOICE_NO, INVOICE_DATE, TAXABLE_VALUE, TOTAL_TAX, CGST, SGST, IGST, CESS, MONTHLY)

6. 3BA table:

3BA(GSTN, MONTHLY, STATE, INTYPE, TOTAL_TAXABLE_VALUE, IGST)

7. 3BB table:

3BB(GSTN, INTYPE, TOTAL_TAXABLE_VALUE)

Query:

```
-- Create table CLIENT_EXTRA
CREATE TABLE CLIENT_EXTRA (
  PAN VARCHAR(10) PRIMARY KEY,
  HOUSE_NO VARCHAR(10),
  STREET VARCHAR(50),
  AREA VARCHAR(50),
  CITY VARCHAR(50),
  DISTRICT VARCHAR(50),
  PINCODE VARCHAR(6),
  AADHAR_NO VARCHAR(12)
);

-- Create table CLIENT
CREATE TABLE CLIENT (
  GSTIN VARCHAR(15) PRIMARY KEY,
  NAME VARCHAR(50),
  PHR VARCHAR(10),
  USERID VARCHAR(20),
  PASSWD VARCHAR(20),
  EMAIL_ID VARCHAR(50),
  PAN VARCHAR(10),
  FOREIGN KEY (PAN) REFERENCES CLIENT_EXTRA (PAN)
);

-- Create table R1A
CREATE TABLE R1A (
  GSTIN VARCHAR(15),
  QUARTER VARCHAR(10),
  INVOICE_NO VARCHAR(20),
  CUST_NAME VARCHAR(50),
  STATE_OF_SUPPLY VARCHAR(50),
  INVOICE_DATE DATE,
  TAXABLE_VALUE DECIMAL(10,2),
  TAX_RATE DECIMAL(5,2),
  CGST DECIMAL(10,2),
  SGST DECIMAL(10,2),
  IGST DECIMAL(10,2),
  CESS DECIMAL(10,2),
```

```
  PRIMARY KEY (GSTIN, QUARTER, INVOICE_NO),
  FOREIGN KEY (GSTIN) REFERENCES CLIENT (GSTIN)
);

-- Create table R1B
CREATE TABLE R1B (
  GSTIN VARCHAR(15),
  QUARTER VARCHAR(10),
  NIL_SUPPLY DECIMAL(10,2),
  NON_GST DECIMAL(10,2),
  NON_GST_EXPED DECIMAL(10,2),
  OTHER_EXEMPTED DECIMAL(10,2),
  PRIMARY KEY (GSTIN, QUARTER),
  FOREIGN KEY (GSTIN) REFERENCES CLIENT (GSTIN)
);

-- Create table R2A
CREATE TABLE R2A (
  GSTIN VARCHAR(15),
  INTYPE VARCHAR(10),
  GSTN VARCHAR(15),
  INVOICE_NO VARCHAR(20),
  INVOICE_DATE DATE,
  TAXABLE_VALUE DECIMAL(10,2),
  TOTAL_TAX DECIMAL(10,2),
  CGST DECIMAL(10,2),
  SGST DECIMAL(10,2),
  IGST DECIMAL(10,2),
  CESS DECIMAL(10,2),
  MONTHLY VARCHAR(10),
  PRIMARY KEY (GSTIN, INTYPE, GSTN, INVOICE_NO),
  FOREIGN KEY (GSTIN) REFERENCES CLIENT (GSTIN)
);

-- Create table 3BA
CREATE TABLE form_3BA (
  GSTN VARCHAR(15),
  MONTHLY VARCHAR(10),
  STATE VARCHAR(50),
  INTYPE VARCHAR(10),
```

```
  TOTAL_TAXABLE_VALUE DECIMAL(10,2),
  IGST DECIMAL(10,2),
  PRIMARY KEY (GSTN, MONTHLY, STATE, INTYPE),
  FOREIGN KEY (GSTN) REFERENCES CLIENT (GSTIN)
);


-- Create table 3BB
CREATE TABLE form_3BB (
  GSTN VARCHAR(15),
  INTYPE VARCHAR(10),
  TOTAL_TAXABLE_VALUE DECIMAL(10,2),
  PRIMARY KEY (GSTN, INTYPE),
  FOREIGN KEY (GSTN) REFERENCES CLIENT (GSTIN)
);
```

WORKBOOK practice :

1)Creating Database Table

```
CREATE TABLE CLIENTEXTRA (
  PAN VARCHAR(10) PRIMARY KEY,
  HOUSE_NO VARCHAR(10),
  STREET VARCHAR(50),
  AREA VARCHAR(50),
  CITY VARCHAR(50),
  DISTRICT VARCHAR(50),
  PINCODE VARCHAR(6),
  AADHAR_NO VARCHAR(12)
);
```

```
Table CLIENTEXTRA created.
```

2)Working with Data Manipulation Commands

```
INSERT INTO CLIENTEXTRA (PAN, HOUSE_NO, STREET, AREA, CITY, DISTRICT, PINCODE,
AADHAR_NO)
VALUES ('ABCDE1234F', '123', 'Main Street', 'Downtown', 'City1', 'District1', '123456', '123456789012');
```

```
1 row inserted.
```

3)Integrity Constraints

```
CREATE TABLE CLIENT1 (
  GSTIN VARCHAR(15) PRIMARY KEY,
  NAME VARCHAR(50),
  PHR VARCHAR(10),
  USERID VARCHAR(20),
  PASSWD VARCHAR(20),
  EMAIL_ID VARCHAR(50),
  PAN VARCHAR(10),
  FOREIGN KEY (PAN) REFERENCES CLIENT_EXTRA (PAN)
);
```

```
Table CLIENT1 created.
```

4)Basic Select Statements

SELECT * FROM form_3BA WHERE GSTN = 'GSTIN123456789';

SQL | All Rows Fetched: 1 in 0.296 seconds

	GSTN	MONTHLY	STATE	INTYPE	TOTAL_TAXABLE_VALUE	IGST
1	GSTIN123456789	January 2023	State1	Type1	10000	1800

5)SQL Functions

SELECT MIN(TOTAL_TAXABLE_VALUE) FROM form_3BA;

SQL | All Rows Fetched: 1 in 0.269 seconds

	MIN(TOTAL_TAXABLE_VALUE)
1	10000

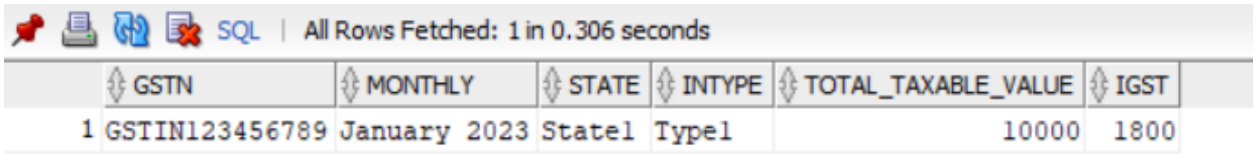
6)Joining Tables

SQL | All Rows Fetched: 18 in 0.355 seconds

	GSTN	MONTHLY	STATE	INTYPE	TOTAL_TAXABLE_VALUE	IGST	GSTN_1	INTYPE_1	TOTAL_TAXABLE_VALUE_1
1	GSTIN123456789	January 2023	State1	Type1	10000	1800	GSTIN123456789	Type1	5000
2	GSTIN987654321	February 2023	State2	Type2	15000	2700	GSTIN987654321	Type2	7000
3	GSTIN345678901	March 2023	State3	Type3	12000	2160	GSTIN345678901	Type3	6000
4	GSTIN345678901	November 2023	State11	Type11	23000	4140	GSTIN345678901	Type3	6000
5	GSTIN234567890	April 2023	State4	Type4	18000	3240	GSTIN234567890	Type4	8000
6	GSTIN234567890	October 2023	State10	Type10	21000	3780	GSTIN234567890	Type4	8000
7	GSTIN456789012	May 2023	State5	Type5	20000	3600	GSTIN456789012	Type5	9000
8	GSTIN789012345	June 2023	State6	Type6	25000	4500	GSTIN789012345	Type6	10000
9	GSTIN789012345	December 2023	State12	Type12	27000	4860	GSTIN789012345	Type6	10000
10	GSTIN901234567	July 2023	State7	Type7	22000	3960	GSTIN901234567	Type7	12000
11	GSTIN012345678	August 2023	State8	Type8	17000	3060	GSTIN012345678	Type8	11000
12	GSTIN567890123	September 2023	State9	Type9	19000	3420	GSTIN567890123	Type9	13000
13	GSTIN234567890	April 2023	State4	Type4	18000	3240	GSTIN234567890	Type10	15000
14	GSTIN234567890	October 2023	State10	Type10	21000	3780	GSTIN234567890	Type10	15000
15	GSTIN345678901	March 2023	State3	Type3	12000	2160	GSTIN345678901	Type11	14000
16	GSTIN345678901	November 2023	State11	Type11	23000	4140	GSTIN345678901	Type11	14000
17	GSTIN789012345	June 2023	State6	Type6	25000	4500	GSTIN789012345	Type12	16000
18	GSTIN789012345	December 2023	State12	Type12	27000	4860	GSTIN789012345	Type12	16000

7)Sub Queries

```
SELECT *
FROM form_3BA
WHERE GSTN IN (SELECT GSTN FROM form_3BB WHERE INTYPE = 'Type1');
```


SQL | All Rows Fetched: 1 in 0.306 seconds

	GSTN	MONTHLY	STATE	INTYPE	TOTAL_TAXABLE_VALUE	IGST
1	GSTIN123456789	January 2023	State1	Type1	10000	1800

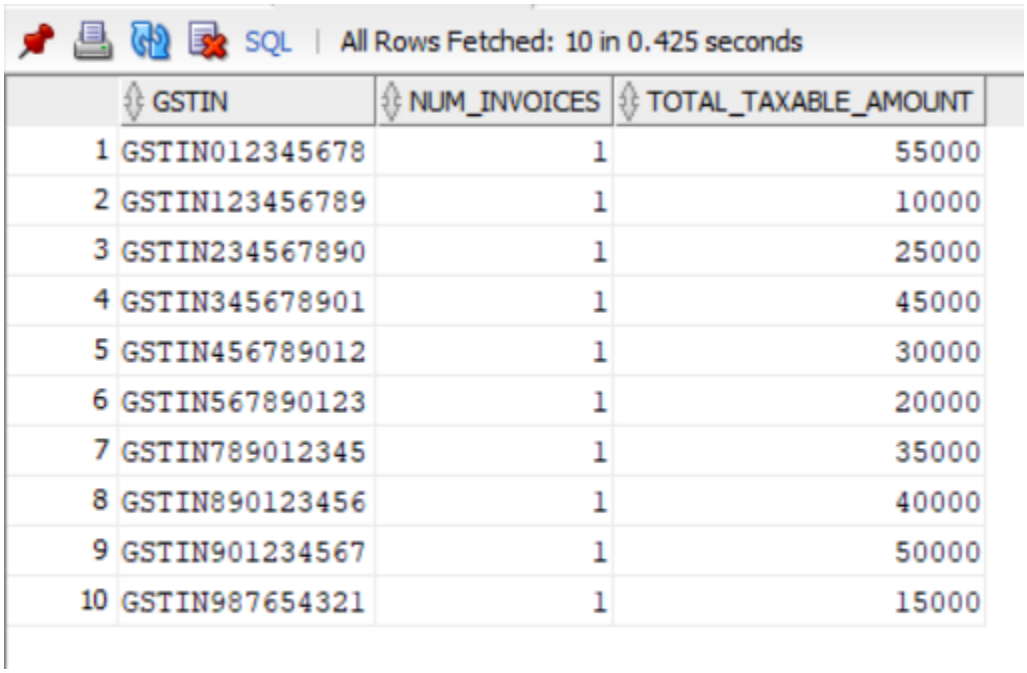
8)Views

```
CREATE VIEW form_3BB_read_only_view AS
SELECT * FROM form_3BB
WITH READ ONLY;

select * from form_3BB_read_only_view;
```

	GSTN	INTYPE	TOTAL_TAXABLE_VALUE
1	GSTIN123456789	Type1	5000
2	GSTIN987654321	Type2	7000
3	GSTIN345678901	Type3	6000
4	GSTIN234567890	Type4	8000
5	GSTIN456789012	Type5	9000
6	GSTIN789012345	Type6	10000
7	GSTIN901234567	Type7	12000
8	GSTIN012345678	Type8	11000
9	GSTIN567890123	Type9	13000
10	GSTIN234567890	Type10	15000
11	GSTIN345678901	Type11	14000
12	GSTIN789012345	Type12	16000

9)Advanced Select Statements

```
SELECT CLIENT.GSTIN, COUNT(*) AS NUM_INVOICES, SUM(R1A.TAXABLE_VALUE) AS
TOTAL_TAXABLE_AMOUNT
FROM R1A
JOIN CLIENT ON R1A.GSTIN = CLIENT.GSTIN
GROUP BY CLIENT.GSTIN;
```

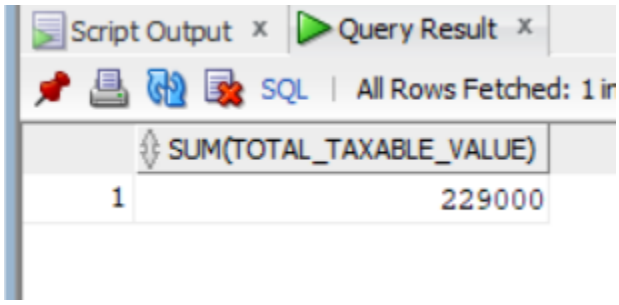


SQL | All Rows Fetched: 10 in 0.425 seconds

	GSTIN	NUM_INVOICES	TOTAL_TAXABLE_AMOUNT
1	GSTIN012345678	1	55000
2	GSTIN123456789	1	10000
3	GSTIN234567890	1	25000
4	GSTIN345678901	1	45000
5	GSTIN456789012	1	30000
6	GSTIN567890123	1	20000
7	GSTIN789012345	1	35000
8	GSTIN890123456	1	40000
9	GSTIN901234567	1	50000
10	GSTIN987654321	1	15000

10)Additional Queries

```
SELECT SUM(TOTAL_TAXABLE_VALUE) FROM form_3BA;
```



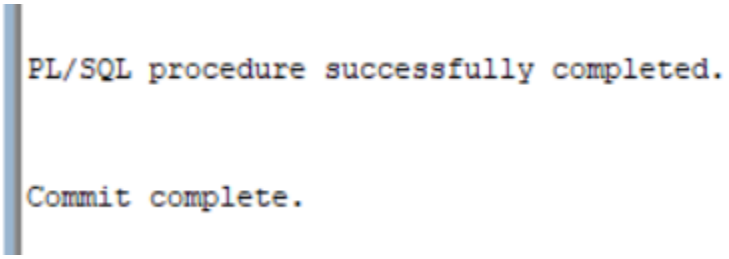
Script Output | Query Result

SQL | All Rows Fetched: 1 in

	SUM(TOTAL_TAXABLE_VALUE)
1	229000

11)Basic of PL/SQL

```
DECLARE
  v_total NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_total FROM form_3BA;
  IF v_total > 100 THEN
    RAISE_APPLICATION_ERROR(-20001, 'The number of records exceeds the limit.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('The number of records is within the limit.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;
/
commit;
```



Query practiced for workbook :

```
--workbook

--select
SELECT * FROM form_3BA WHERE GSTN = 'GSTIN123456789';
SELECT * FROM form_3BA ORDER BY MONTHLY DESC;
SELECT DISTINCT INTYPE FROM form_3BA;
SELECT MONTHLY, SUM(TOTAL_TAXABLE_VALUE) AS TOTAL_AMOUNT FROM form_3BA
GROUP BY MONTHLY;
SELECT a.GSTN, b.INTYPE, b.TOTAL_TAXABLE_VALUE FROM form_3BA a INNER JOIN
form_3BB b ON a.GSTN = b.GSTN;

--SQL function
SELECT COUNT(*) FROM form_3BA;
SELECT SUM(TOTAL_TAXABLE_VALUE) FROM form_3BA;
SELECT AVG(TOTAL_TAXABLE_VALUE) FROM form_3BA;
SELECT MAX(TOTAL_TAXABLE_VALUE) FROM form_3BA;
SELECT MIN(TOTAL_TAXABLE_VALUE) FROM form_3BA;
SELECT UPPER(INTYPE) FROM form_3BA;
SELECT LOWER(GSTN) FROM form_3BB;
SELECT CONCAT(GSTN, '-', INTYPE) FROM form_3BA;
SELECT DATE_FORMAT(INVOICE_DATE, '%Y-%m-%d') FROM form_3BA;

--join
SELECT *
FROM form_3BA
INNER JOIN form_3BB ON form_3BA.GSTN = form_3BB.GSTN;

SELECT *
FROM form_3BA
LEFT JOIN form_3BB ON form_3BA.GSTN = form_3BB.GSTN;

SELECT *
FROM form_3BA
RIGHT JOIN form_3BB ON form_3BA.GSTN = form_3BB.GSTN;

SELECT *
FROM form_3BA
FULL JOIN form_3BB ON form_3BA.GSTN = form_3BB.GSTN;
```

```
--sub queries
SELECT *
FROM form_3BA
WHERE GSTN IN (SELECT GSTN FROM form_3BB WHERE INTYPE = 'Type1');

SELECT *,
    (SELECT COUNT(*) FROM form_3BB WHERE GSTN = form_3BA.GSTN) AS BB_COUNT
FROM form_3BA;

SELECT *
FROM (SELECT DISTINCT GSTN FROM form_3BA) AS SubqueryTable;

SELECT *,
    (SELECT MAX(TOTAL_TAXABLE_VALUE) FROM form_3BB WHERE GSTN =
form_3BA.GSTN) AS MAX_TAXABLE_VALUE
FROM form_3BA;


-- views
CREATE MATERIALIZED VIEW form_3BB_materialized_view
BUILD IMMEDIATE
REFRESH COMPLETE
AS
SELECT * FROM form_3BB;

CREATE VIEW form_3BB_read_only_view AS
SELECT * FROM form_3BB
WITH READ ONLY;

select * from form_3BB_read_only_view;

CREATE VIEW form_3BA_updatable_view AS
SELECT * FROM form_3BA
WITH CHECK OPTION;
```

```
-- advanced select
SELECT CLIENT.GSTIN, COUNT(*) AS NUM_INVOICES, SUM(R1A.TAXABLE_VALUE) AS
TOTAL_TAXABLE_AMOUNT
FROM R1A
JOIN CLIENT ON R1A.GSTIN = CLIENT.GSTIN
GROUP BY CLIENT.GSTIN;

SELECT CLIENT.GSTIN, CLIENT.NAME,
     (SELECT COUNT(*) FROM R1A WHERE R1A.GSTIN = CLIENT.GSTIN) AS NUM_INVOICES
FROM CLIENT
WHERE CLIENT.GSTIN IN (SELECT DISTINCT GSTIN FROM R1A);

SELECT GSTIN, INVOICE_NO, TAXABLE_VALUE,
     SUM(TAXABLE_VALUE) OVER (PARTITION BY GSTIN ORDER BY INVOICE_DATE) AS
CUMULATIVE_TOTAL
FROM R1A;


-- pl sql
DECLARE
  CURSOR c_data IS
    SELECT * FROM form_3BA;
BEGIN
  FOR rec IN c_data LOOP
    DBMS_OUTPUT.PUT_LINE('GSTN: ' || rec.GSTN || ', Monthly: ' || rec.MONTHLY);
  END LOOP;
END;/

DECLARE
  v_total NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_total FROM form_3BB;
  IF v_total > 0 THEN
    DBMS_OUTPUT.PUT_LINE('There are ' || v_total || ' records in form_3BB table.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('form_3BB table is empty.');
  END IF;
END;
```

```
DECLARE
  v_total NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_total FROM form_3BA;
  IF v_total > 100 THEN
    RAISE_APPLICATION_ERROR(-20001, 'The number of records exceeds the limit.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('The number of records is within the limit.');
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;

commit;

SELECT COUNT(*) INTO v_total FROM form_3BA;
```

EMPATHY MAPPING :

IMPLEMENTATION OF NORMALIZATION

As we did had all table in proper normal form

So we created a sample view table such that if we had a table that violates the normal form

1NF:

BEFORE

	GSTIN	NAME	PHR	USERID	PASSWD	EMAIL_ID	HOUSE_NO	STREET	AREA	CITY	DISTRICT	PINCODE	AADHAR_NO
1	GSTIN123456789	Client1	9876543210	client1_user	client1_pass	client1@example.com	123	Main Street	Downtown	City1	District1	123456	123456789012
2	GSTIN789012345	Client6	2345678901	client6_user	client6_pass	client6@example.com	123	Main Street	Downtown	City1	District1	123456	123456789012
3	GSTIN987654321	Client2	1234567890	client2_user	client2_pass	client2@example.com	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
4	GSTIN890123456	Client7	5678901234	client7_user	client7_pass	client7@example.com	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
5	GSTIN567890123	Client3	4567890123	client3_user	client3_pass	client3@example.com	789	Elm Street	Uptown	City3	District3	789012	345678901234
6	GSTIN345678901	Client8	8901234567	client8_user	client8_pass	client8@example.com	789	Elm Street	Uptown	City3	District3	789012	345678901234
7	GSTIN901234567	Client9	9012345678	client9_user	client9_pass	client9@example.com	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
8	GSTIN234567890	Client4	7890123456	client4_user	client4_pass	client4@example.com	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
9	GSTIN012345678	Client10	1234567890	client10_user	client10_pass	client10@example.com	1314	Cedar Street	Downtown	City5	District5	131415	567890123456
10	GSTIN456789012	Client5	0123456789	client5_user	client5_pass	client5@example.com	1314	Cedar Street	Downtown	City5	District5	131415	567890123456

AFTER

	GSTIN	NAME	PHR	USERID	PASSWD	EMAIL_ID	PAN
1	GSTIN123456789	Client1	9876543210	client1_user	client1_pass	client1@example.com	ABCDE1234F
2	GSTIN987654321	Client2	1234567890	client2_user	client2_pass	client2@example.com	FGHIJ5678K
3	GSTIN567890123	Client3	4567890123	client3_user	client3_pass	client3@example.com	LMNOP9012G
4	GSTIN234567890	Client4	7890123456	client4_user	client4_pass	client4@example.com	QRSTU3456H
5	GSTIN456789012	Client5	0123456789	client5_user	client5_pass	client5@example.com	VWXYZ6789I
6	GSTIN789012345	Client6	2345678901	client6_user	client6_pass	client6@example.com	ABCDE1234F
7	GSTIN890123456	Client7	5678901234	client7_user	client7_pass	client7@example.com	FGHIJ5678K
8	GSTIN345678901	Client8	8901234567	client8_user	client8_pass	client8@example.com	LMNOP9012G
9	GSTIN901234567	Client9	9012345678	client9_user	client9_pass	client9@example.com	QRSTU3456H
10	GSTIN012345678	Client10	1234567890	client10_user	client10_pass	client10@example.com	VWXYZ6789I

	PAN	HOUSE_NO	STREET	AREA	CITY	DISTRICT	PINCODE	AADHAR_NO
1	ABCDE1234F	123	Main Street	Downtown	City1	District1	123456	123456789012
2	FGHIJ5678K	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
3	LMNOP9012G	789	Elm Street	Uptown	City3	District3	789012	345678901234
4	QRSTU3456H	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
5	VWXYZ6789I	1314	Cedar Street	Downtown	City5	District5	131415	567890123456

2NF:

BEFORE

	GSTIN	QUARTER	INVOICE_NO	CUST_NAME	STATE_OF_SUPPLY	INVOICE_DATE	TAXABLE_VALUE	TAX_RATE	CGST	SGST	IGST	CESS	NAME	PHR
1	GSTIN012345678	Q2	INV012345	Customer10	State10	15-01-23	55000	12	6600	6600	0	700	Client10	1234567890
2	GSTIN123456789	Q1	INV123456	Customer1	State1	15-01-23	10000	18	900	900	0	100	Client1	9876543210
3	GSTIN234567890	Q4	INV456789	Customer4	State4	15-01-23	25000	12	3000	3000	0	300	Client4	7890123456
4	GSTIN345678901	Q4	INV890123	Customer8	State8	15-01-23	45000	12	5400	5400	0	600	Client8	8901234567
5	GSTIN456789012	Q1	INV567890	Customer5	State5	15-01-23	30000	18	5400	5400	0	600	Client5	0123456789
6	GSTIN567890123	Q3	INV345678	Customer3	State3	15-01-23	20000	18	3600	3600	0	400	Client3	4567890123
7	GSTIN789012345	Q2	INV678901	Customer6	State6	15-01-23	35000	12	4200	4200	0	500	Client6	2345678901
8	GSTIN890123456	Q3	INV789012	Customer7	State7	15-01-23	40000	18	7200	7200	0	800	Client7	5678901234
9	GSTIN901234567	Q1	INV901234	Customer9	State9	15-01-23	50000	18	9000	9000	0	1000	Client9	9012345678
10	GSTIN987654321	Q2	INV234567	Customer2	State2	15-01-23	15000	12	1800	1800	0	200	Client2	1234567890

AFTER

	GSTIN	NAME	PHR	USERID	PASSWD	EMAIL_ID	PAN
1	GSTIN123456789	Client1	9876543210	client1_user	client1_pass	client1@example.com	ABCDE1234F
2	GSTIN987654321	Client2	1234567890	client2_user	client2_pass	client2@example.com	FGHIJ5678K
3	GSTIN567890123	Client3	4567890123	client3_user	client3_pass	client3@example.com	LMNOP9012G
4	GSTIN234567890	Client4	7890123456	client4_user	client4_pass	client4@example.com	QRSTU3456H
5	GSTIN456789012	Client5	0123456789	client5_user	client5_pass	client5@example.com	VWXYZ6789I
6	GSTIN789012345	Client6	2345678901	client6_user	client6_pass	client6@example.com	ABCDE1234F
7	GSTIN890123456	Client7	5678901234	client7_user	client7_pass	client7@example.com	FGHIJ5678K
8	GSTIN345678901	Client8	8901234567	client8_user	client8_pass	client8@example.com	LMNOP9012G
9	GSTIN901234567	Client9	9012345678	client9_user	client9_pass	client9@example.com	QRSTU3456H
10	GSTIN012345678	Client10	1234567890	client10_user	client10_pass	client10@example.com	VWXYZ6789I

	GSTIN	QUARTER	INVOICE_NO	CUST_NAME	STATE_OF_SUPPLY	INVOICE_DATE	TAXABLE_VALUE	TAX_RATE	CGST	SGST	IGST	CESS
1	GSTIN123456789	Q1	INV123456	Customer1	State1	15-01-23	10000	18	900	900	0	100
2	GSTIN987654321	Q2	INV234567	Customer2	State2	15-01-23	15000	12	1800	1800	0	200
3	GSTIN567890123	Q3	INV345678	Customer3	State3	15-01-23	20000	18	3600	3600	0	400
4	GSTIN234567890	Q4	INV456789	Customer4	State4	15-01-23	25000	12	3000	3000	0	300
5	GSTIN456789012	Q1	INV567890	Customer5	State5	15-01-23	30000	18	5400	5400	0	600
6	GSTIN789012345	Q2	INV678901	Customer6	State6	15-01-23	35000	12	4200	4200	0	500
7	GSTIN890123456	Q3	INV789012	Customer7	State7	15-01-23	40000	18	7200	7200	0	800
8	GSTIN345678901	Q4	INV890123	Customer8	State8	15-01-23	45000	12	5400	5400	0	600
9	GSTIN901234567	Q1	INV901234	Customer9	State9	15-01-23	50000	18	9000	9000	0	1000
10	GSTIN012345678	Q2	INV012345	Customer10	State10	15-01-23	55000	12	6600	6600	0	700

3NF:

BEFORE

	GSTIN	QUARTER	NIL_SUPPLY	NON_GST	NON_GST_EXPED	OTHER_EXEMPTED	NAME	PHR	HOUSE_NO	STREET	AREA	CITY	DISTRICT	PINCODE	AADHAR_NO
1	GSTIN123456789	Q1 2023	1000	500	200	100	Client1	9876543210	123	Main Street	Downtown	City1	District1	123456	123456789012
2	GSTIN987654321	Q2 2023	1500	600	250	150	Client2	1234567890	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
3	GSTIN345678901	Q3 2023	1200	700	300	200	Client8	8901234567	789	Elm Street	Uptown	City3	District3	789012	345678901234
4	GSTIN234567890	Q4 2023	1300	800	350	250	Client4	7890123456	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
5	GSTIN456789012	Q1 2024	1400	900	400	300	Client5	0123456789	1314	Cedar Street	Downtown	City5	District5	131415	567890123456
6	GSTIN789012345	Q2 2024	1500	1000	450	350	Client6	2345678901	123	Main Street	Downtown	City1	District1	123456	123456789012
7	GSTIN901234567	Q3 2024	1600	1100	500	400	Client9	9012345678	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
8	GSTIN012345678	Q4 2024	1700	1200	550	450	Client10	1234567890	1314	Cedar Street	Downtown	City5	District5	131415	567890123456
9	GSTIN567890123	Q1 2025	1800	1300	600	500	Client3	4567890123	789	Elm Street	Uptown	City3	District3	789012	345678901234
10	GSTIN234567890	Q2 2025	1900	1400	650	550	Client4	7890123456	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
11	GSTIN345678901	Q3 2025	2000	1500	700	600	Client8	8901234567	789	Elm Street	Uptown	City3	District3	789012	345678901234
12	GSTIN789012345	Q4 2025	2100	1600	750	650	Client6	2345678901	123	Main Street	Downtown	City1	District1	123456	123456789012

AFTER

	GSTIN	QUARTER	NIL_SUPPLY	NON_GST	NON_GST_EXPED	OTHER_EXEMPTED
1	GSTIN123456789	Q1 2023	1000	500	200	100
2	GSTIN987654321	Q2 2023	1500	600	250	150
3	GSTIN345678901	Q3 2023	1200	700	300	200
4	GSTIN234567890	Q4 2023	1300	800	350	250
5	GSTIN456789012	Q1 2024	1400	900	400	300
6	GSTIN789012345	Q2 2024	1500	1000	450	350
7	GSTIN901234567	Q3 2024	1600	1100	500	400
8	GSTIN012345678	Q4 2024	1700	1200	550	450
9	GSTIN567890123	Q1 2025	1800	1300	600	500
10	GSTIN234567890	Q2 2025	1900	1400	650	550
11	GSTIN345678901	Q3 2025	2000	1500	700	600
12	GSTIN789012345	Q4 2025	2100	1600	750	650

	GSTIN	NAME	PHR	USERID	PASSWD	EMAIL_ID	PAN
1	GSTIN123456789	Client1	9876543210	client1_user	client1_pass	client1@example.com	ABCDE1234F
2	GSTIN987654321	Client2	1234567890	client2_user	client2_pass	client2@example.com	FGHIJ5678K
3	GSTIN567890123	Client3	4567890123	client3_user	client3_pass	client3@example.com	LMNOP9012G
4	GSTIN234567890	Client4	7890123456	client4_user	client4_pass	client4@example.com	QRSTU3456H
5	GSTIN456789012	Client5	0123456789	client5_user	client5_pass	client5@example.com	VWXYZ6789I
6	GSTIN789012345	Client6	2345678901	client6_user	client6_pass	client6@example.com	ABCDE1234F
7	GSTIN890123456	Client7	5678901234	client7_user	client7_pass	client7@example.com	FGHIJ5678K
8	GSTIN345678901	Client8	8901234567	client8_user	client8_pass	client8@example.com	LMNOP9012G
9	GSTIN901234567	Client9	9012345678	client9_user	client9_pass	client9@example.com	QRSTU3456H
10	GSTIN012345678	Client10	1234567890	client10_user	client10_pass	client10@example.com	VWXYZ6789I

Implementation of concurrency control and recovery mechanisms

Concurrency Control

Concurrency control ensures that database transactions are performed concurrently without leading to data inconsistency. It maintains the accuracy and integrity of the database when multiple users access and manipulate the data simultaneously.

1.Locking Mechanisms

Row-Level Locking: Implement row-level locking where a transaction locks only the specific row it is accessing. For instance, when a ticket is being booked, lock only that particular ticket entry, not the entire table.

Read and Write Locks (Shared and Exclusive Locks):

Shared Locks for read-only operations, allowing multiple users to read the data simultaneously without modifying it.

Exclusive Locks for write operations, preventing other operations from accessing the locked data.

2. Optimistic Concurrency Control

Use optimistic concurrency for operations where conflicts are less likely but do need protection against anomalies. This typically involves:

Reading a record,

Taking note of a version number or timestamp,

Updating the record,

Checking the version or timestamp before committing to ensure no other transaction has modified the record.

3. Transaction Management

Ensure that all database transactions are atomic, consistent, isolated, and durable (ACID properties).

Use transaction logs to ensure that operations can be rolled back if a transaction is incomplete (e.g., a user books a ticket but doesn't complete payment).

Recovery Mechanisms

Recovery mechanisms ensure that the system can recover from hardware or software failures and restore its state to the last consistent state.

1. Database Backups

Regular Backups: Implement regular full and incremental backups of the database. Full backups capture the entire database at a point in time, while incremental backups only record changes since the last backup.

Redundancy: Use database replication to maintain real-time backups on different servers.

2. Transaction Logs

Maintain a detailed transaction log that records every change made to the database. In case of a system failure, these logs can be used to redo or undo transactions to restore the database to its last consistent state.

3. Checkpointing

Implement checkpointing in your system. A checkpoint is a point in the transaction log where all prior transactions have been committed to the database. In case of a crash, recovery processes only need to start from the last checkpoint.

4. Failover Mechanisms

Set up failover mechanisms such as database clustering or master-slave replication to ensure high availability and continuity in case the primary server fails.

Implementation in SQL

START TRANSACTION;

-- Set the isolation level to SERIALIZABLE to ensure serializability
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

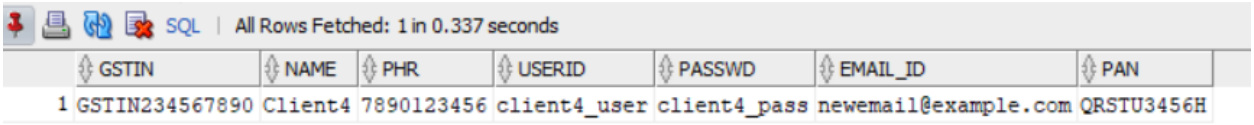
-- SQL queries within the transaction
-- For example, update client information
UPDATE CLIENT SET EMAIL_ID = 'newemail@example.com' WHERE GSTIN = 'GSTIN234567890';

-- Commit the transaction
COMMIT;

```
Transaction ISOLATION succeeded.


1 row updated.


Commit complete.
```

explicit locking using SELECT FOR UPDATE
START TRANSACTION;
SELECT * FROM CLIENT WHERE GSTIN = 'GSTIN234567890' FOR UPDATE;
-- Perform operations on the selected data
COMMIT;

```
>>Query Run In:Query Result

Commit complete.
```

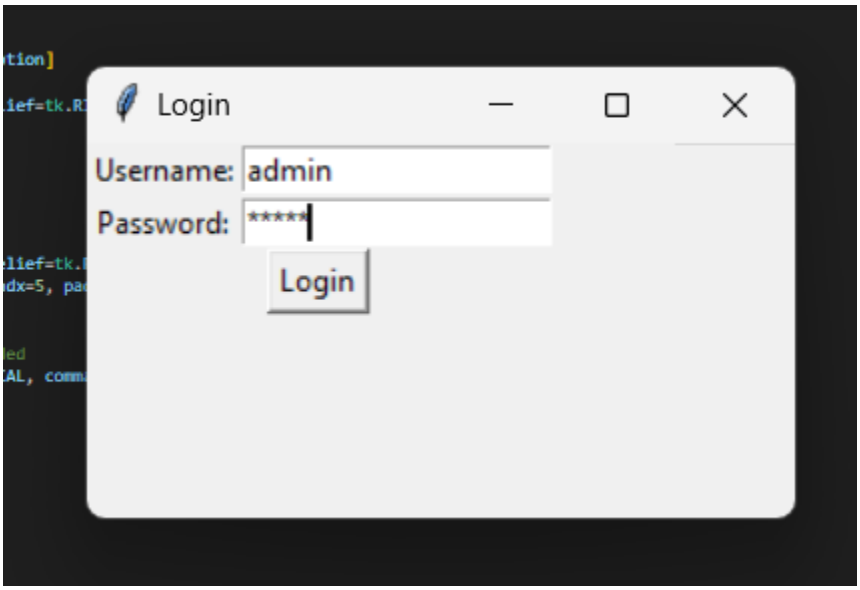


SQL | All Rows Fetched: 1 in 0.337 seconds

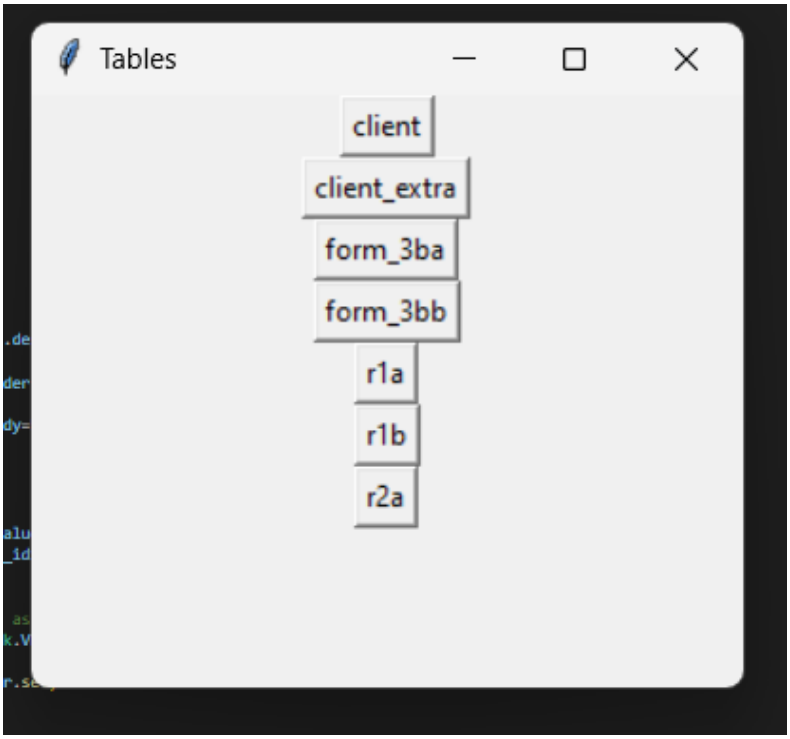
	GSTIN	NAME	PHR	USERID	PASSWD	EMAIL_ID	PAN
1	GSTIN234567890	Client4	7890123456	client4_user	client4_pass	newemail@example.com	QRSTU3456H

Frontend

Login page :

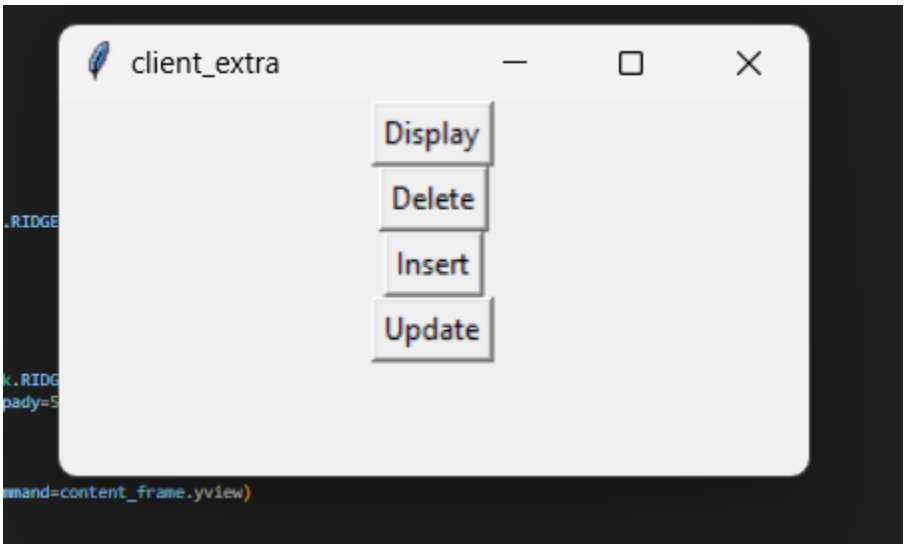


Display of tables:

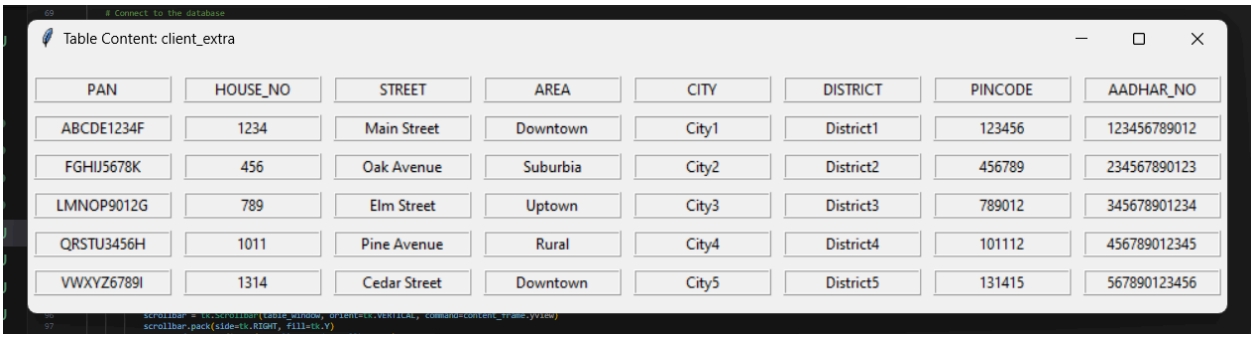


Here we clicked on client_extra:

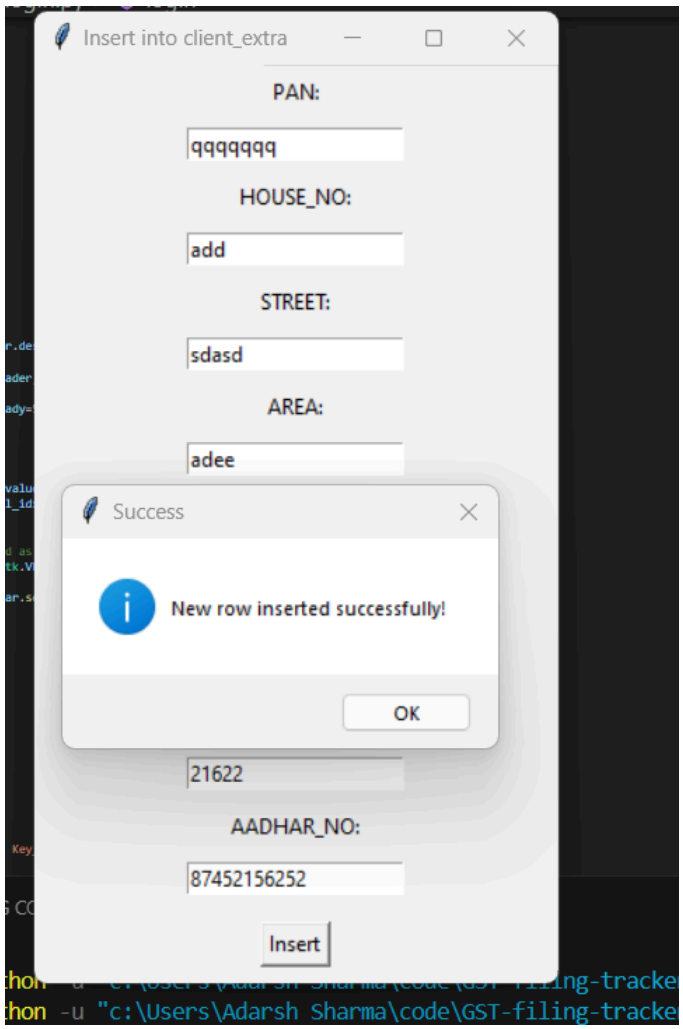
We get option to edit the table:



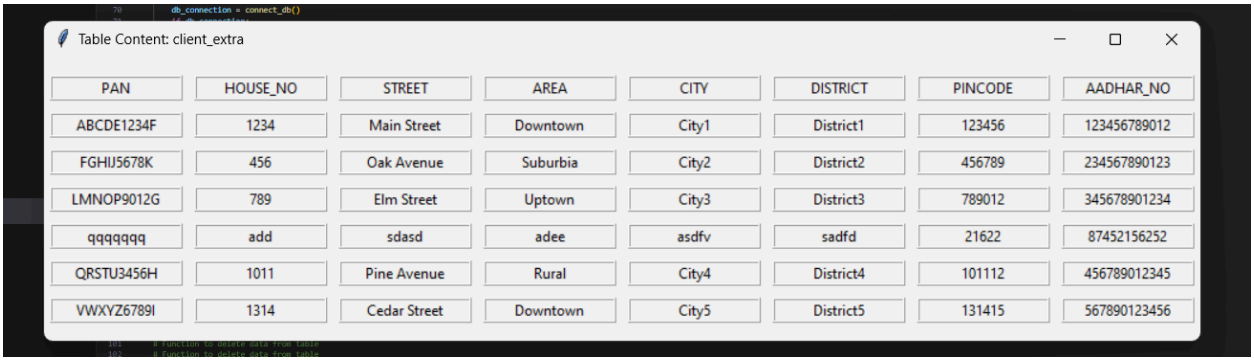
Displaying client_extra data :



Adding data into client_extra data:

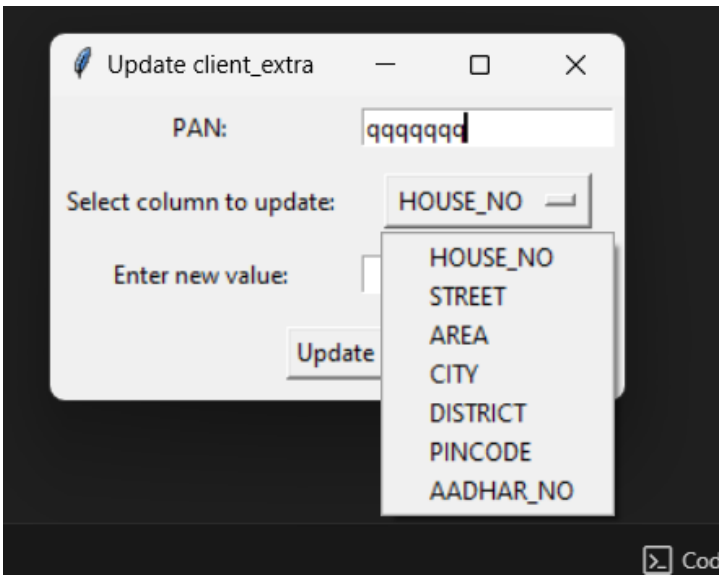


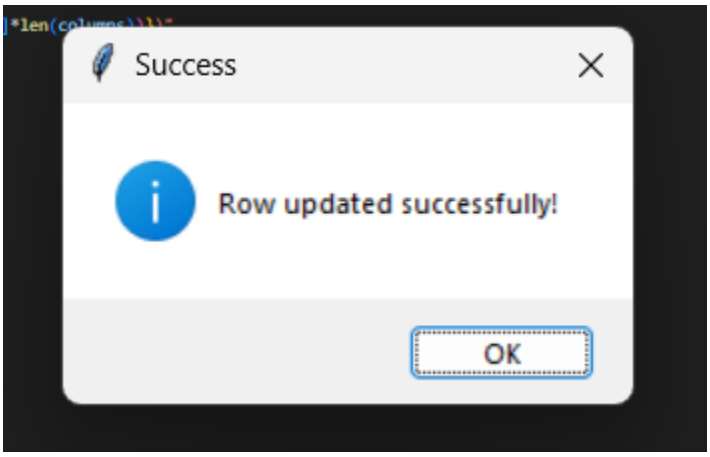
Display of inserted data:



PAN	HOUSE_NO	STREET	AREA	CITY	DISTRICT	PINCODE	AADHAR_NO
ABCDE1234F	1234	Main Street	Downtown	City1	District1	123456	123456789012
FGHIJ5678K	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
LMNOP9012G	789	Elm Street	Uptown	City3	District3	789012	345678901234
qqqqqqq	add	sdasd	adee	asdfv	sadfd	21622	87452156252
QRSTU3456H	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
VWXYZ6789I	1314	Cedar Street	Downtown	City5	District5	131415	567890123456

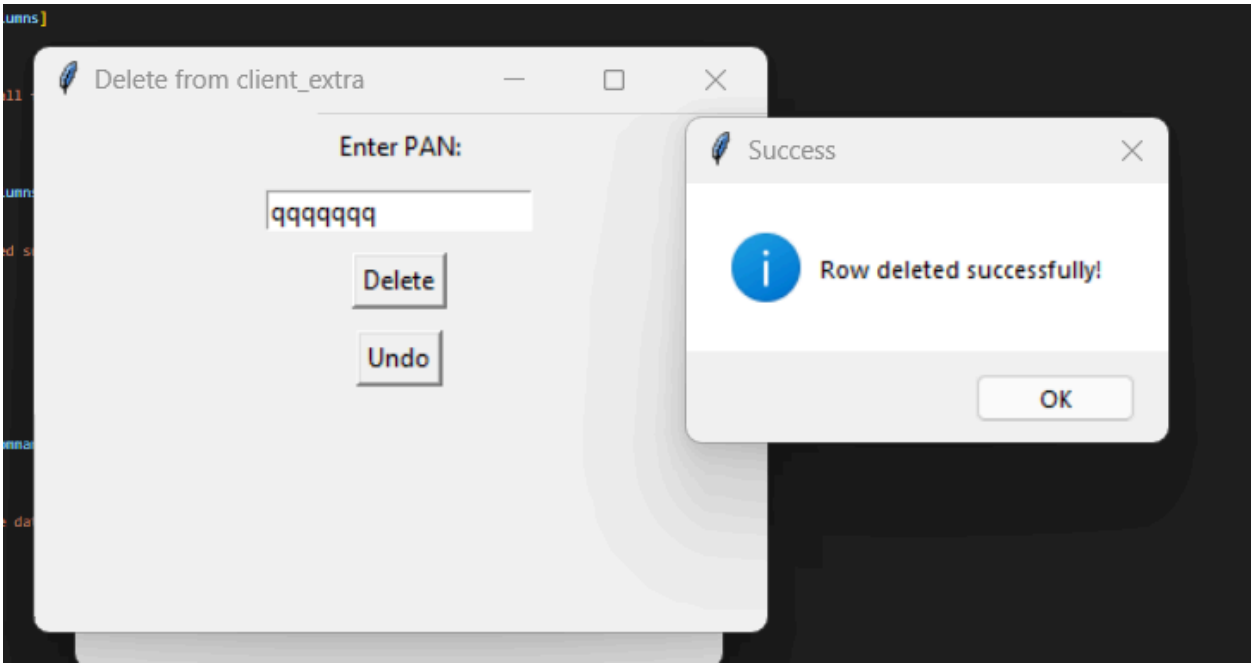
Using update option



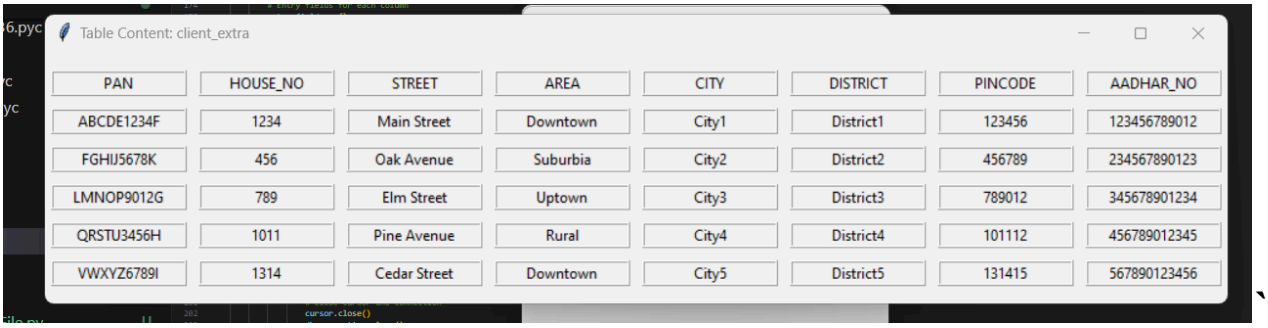


After updating , displaying the data:

Deleting the data



Display of data after deleting :



PAN	HOUSE_NO	STREET	AREA	CITY	DISTRICT	PINCODE	AADHAR_NO
ABCDE1234F	1234	Main Street	Downtown	City1	District1	123456	123456789012
FGHIJ5678K	456	Oak Avenue	Suburbia	City2	District2	456789	234567890123
LMNOP9012G	789	Elm Street	Uptown	City3	District3	789012	345678901234
QRSTU3456H	1011	Pine Avenue	Rural	City4	District4	101112	456789012345
VWXYZ6789I	1314	Cedar Street	Downtown	City5	District5	131415	567890123456

Conclusion

In conclusion, the features of NoSQL databases represent a significant advancement in data management technology, offering a compelling alternative to traditional relational databases. The flexibility, scalability, and performance of NoSQL databases make them well-suited for modern applications that handle large volumes of diverse and rapidly changing data. With features such as schema less design, horizontal scalability, high availability, and support for various data models, NoSQL databases empower developers to build robust, scalable, and efficient systems to meet the demands of today's data-driven world.

By embracing NoSQL technology, organizations can unlock new possibilities for handling big data, real-time analytics, and complex data processing tasks. Whether it's powering web applications, IoT platforms, content management systems, or mobile apps, NoSQL databases offer the agility and performance required to thrive in today's competitive digital landscape.

As the demand for scalable, flexible, and high-performance data storage solutions continues to grow, NoSQL databases are poised to play a pivotal role in shaping the future of data management. With ongoing advancements and innovations in the NoSQL ecosystem, the potential for leveraging these databases to drive business growth and innovation is boundless. In summary, the features of NoSQL databases represent a transformative force in the realm of data management, offering a foundation upon which organizations can build the next generation of data-driven applications and services.