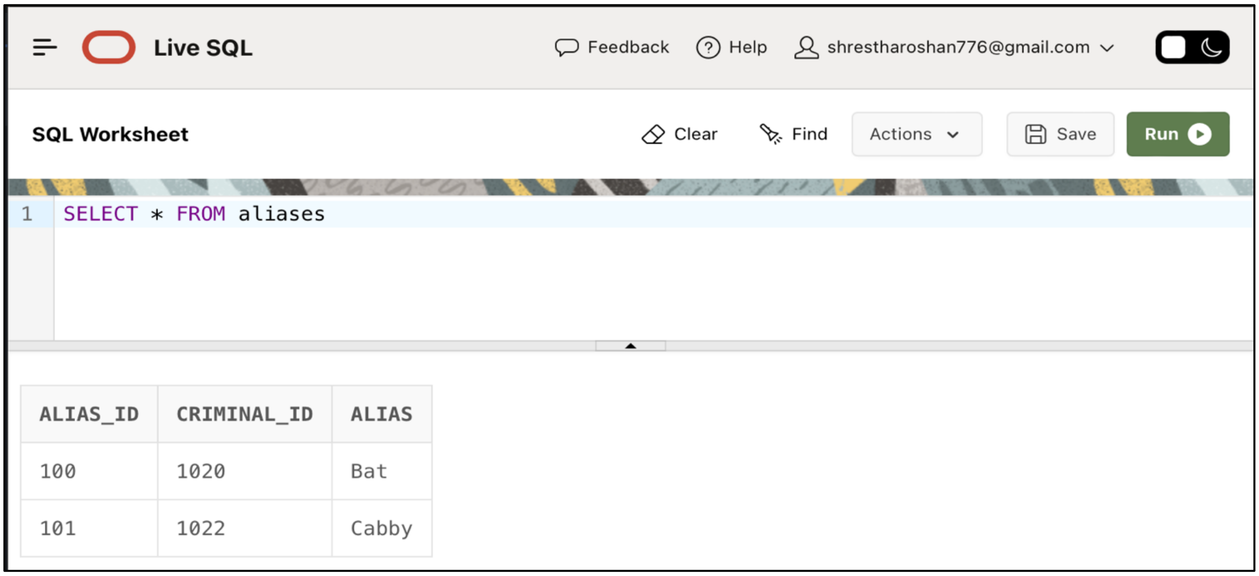
Database Design – 2023W
Student ID: 901142
Student Name: Roshan Shrestha
Practical Activity#21#22#23#24

The list of all the created tables after executing the provided SQL script are listed below:

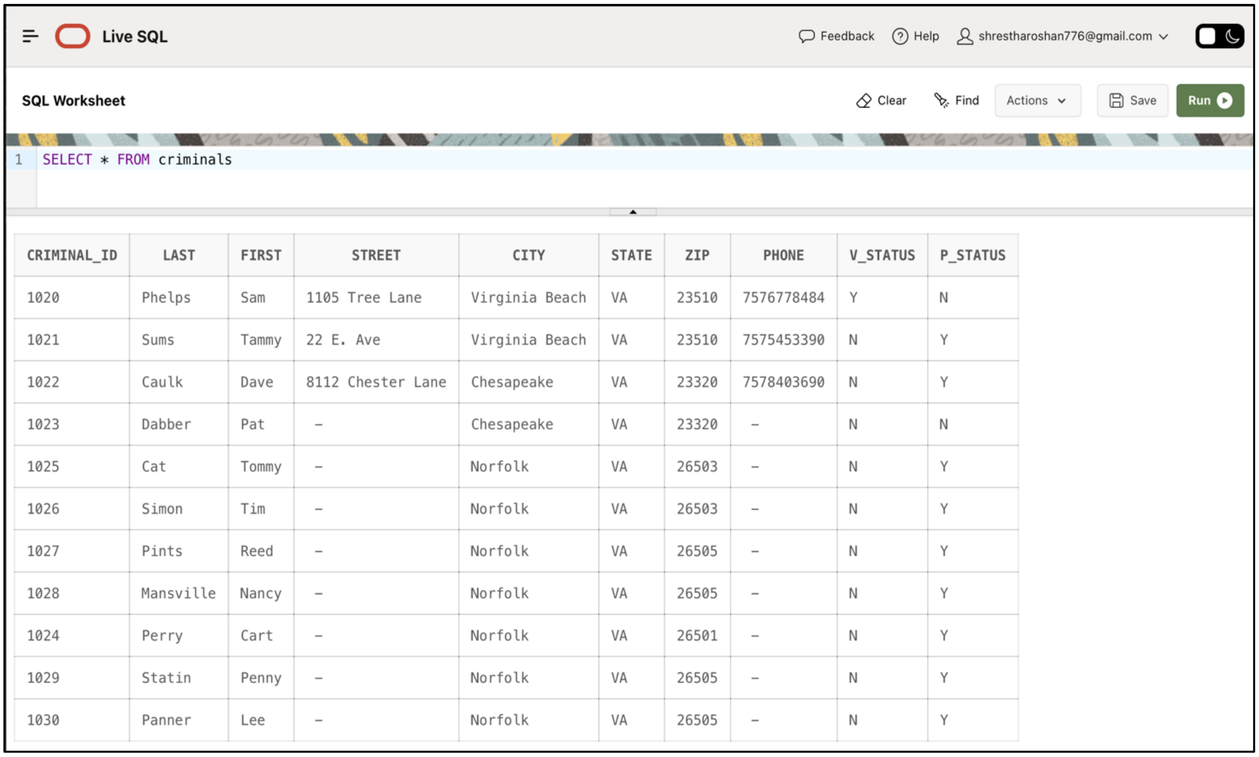
Table: aliases



```
SELECT * FROM aliases
```

ALIAS_ID	CRIMINAL_ID	ALIAS
100	1020	Bat
101	1022	Cabby

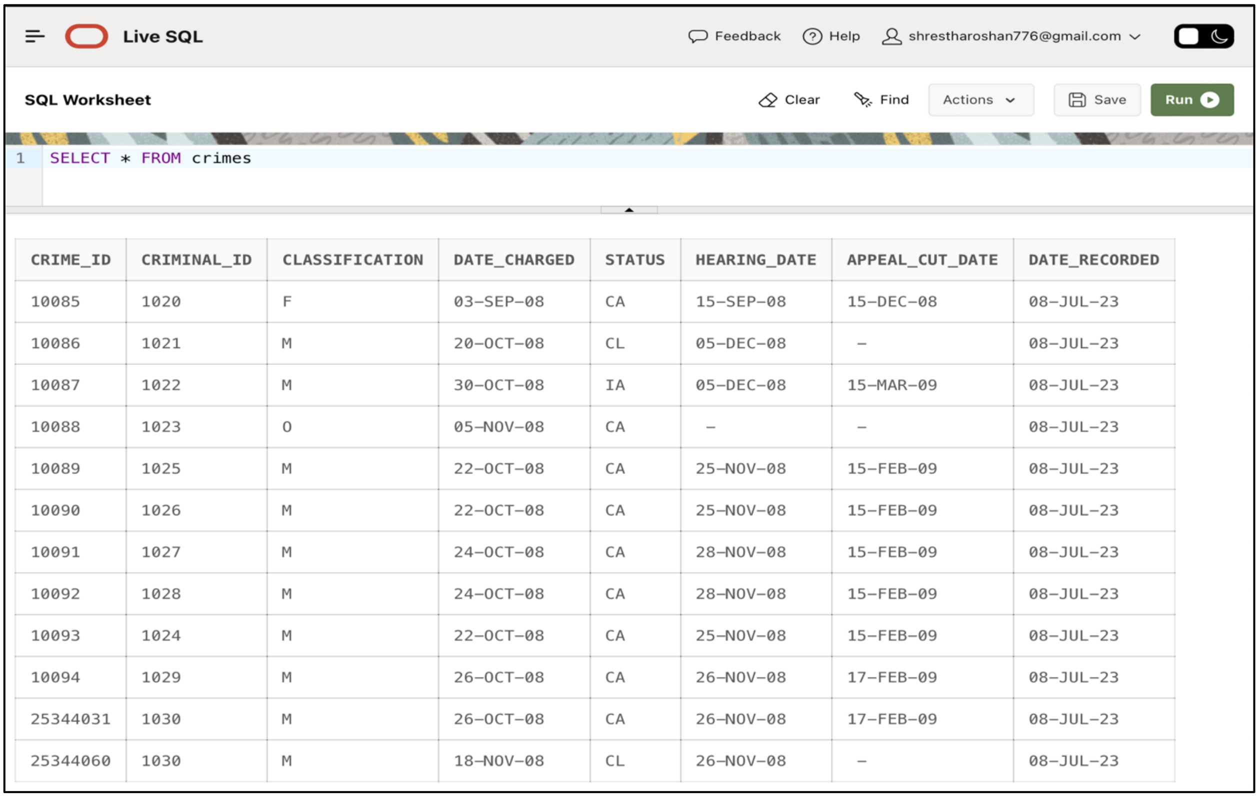
Table : criminals



```
SELECT * FROM criminals
```

CRIMINAL_ID	LAST	FIRST	STREET	CITY	STATE	ZIP	PHONE	V_STATUS	P_STATUS
1020	Phelps	Sam	1105 Tree Lane	Virginia Beach	VA	23510	7576778484	Y	N
1021	Sums	Tammy	22 E. Ave	Virginia Beach	VA	23510	7575453390	N	Y
1022	Caulk	Dave	8112 Chester Lane	Chesapeake	VA	23320	7578403690	N	Y
1023	Dabber	Pat	-	Chesapeake	VA	23320	-	N	N
1025	Cat	Tommy	-	Norfolk	VA	26503	-	N	Y
1026	Simon	Tim	-	Norfolk	VA	26503	-	N	Y
1027	Pints	Reed	-	Norfolk	VA	26505	-	N	Y
1028	Mansville	Nancy	-	Norfolk	VA	26505	-	N	Y
1024	Perry	Cart	-	Norfolk	VA	26501	-	N	Y
1029	Statin	Penny	-	Norfolk	VA	26505	-	N	Y
1030	Panner	Lee	-	Norfolk	VA	26505	-	N	Y

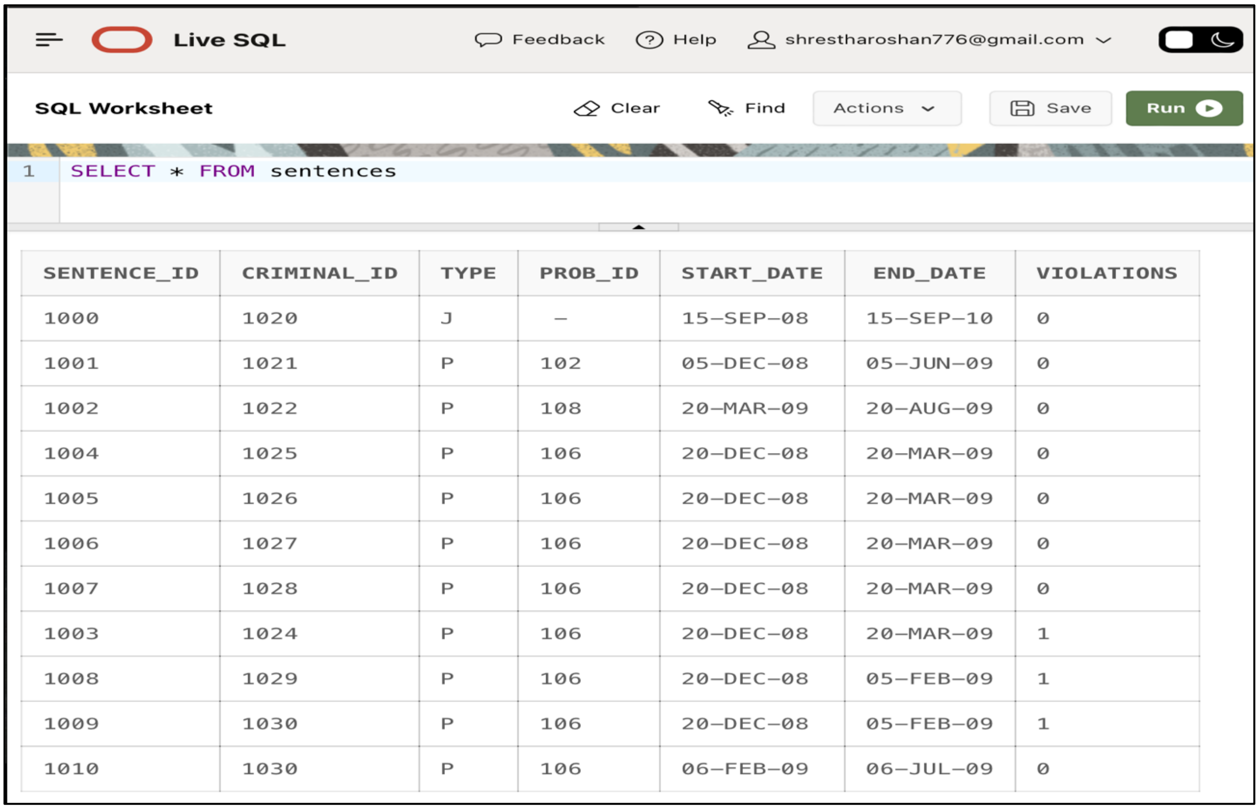
Table : crimes



```
SELECT * FROM crimes
```

CRIME_ID	CRIMINAL_ID	CLASSIFICATION	DATE_CHARGED	STATUS	HEARING_DATE	APPEAL_CUT_DATE	DATE_RECORDED
10085	1020	F	03-SEP-08	CA	15-SEP-08	15-DEC-08	08-JUL-23
10086	1021	M	20-OCT-08	CL	05-DEC-08	-	08-JUL-23
10087	1022	M	30-OCT-08	IA	05-DEC-08	15-MAR-09	08-JUL-23
10088	1023	O	05-NOV-08	CA	-	-	08-JUL-23
10089	1025	M	22-OCT-08	CA	25-NOV-08	15-FEB-09	08-JUL-23
10090	1026	M	22-OCT-08	CA	25-NOV-08	15-FEB-09	08-JUL-23
10091	1027	M	24-OCT-08	CA	28-NOV-08	15-FEB-09	08-JUL-23
10092	1028	M	24-OCT-08	CA	28-NOV-08	15-FEB-09	08-JUL-23
10093	1024	M	22-OCT-08	CA	25-NOV-08	15-FEB-09	08-JUL-23
10094	1029	M	26-OCT-08	CA	26-NOV-08	17-FEB-09	08-JUL-23
25344031	1030	M	26-OCT-08	CA	26-NOV-08	17-FEB-09	08-JUL-23
25344060	1030	M	18-NOV-08	CL	26-NOV-08	-	08-JUL-23

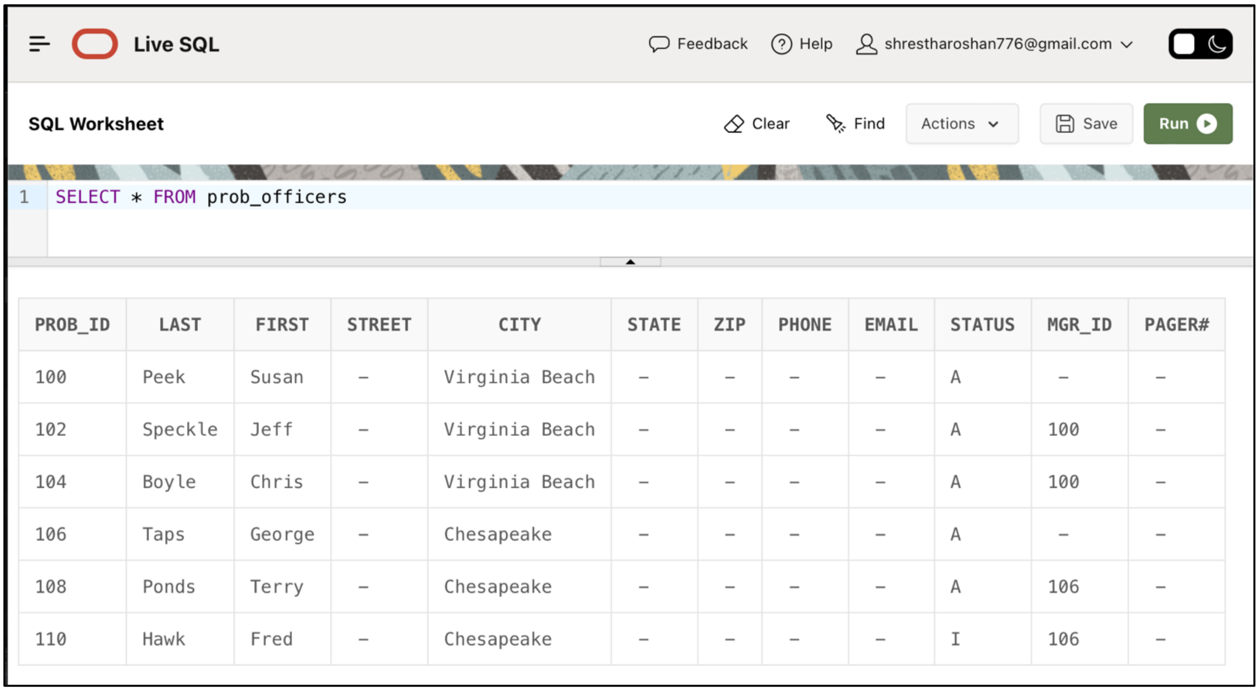
Table : sentences



```
SELECT * FROM sentences
```

SENTENCE_ID	CRIMINAL_ID	TYPE	PROB_ID	START_DATE	END_DATE	VIOLATIONS
1000	1020	J	-	15-SEP-08	15-SEP-10	0
1001	1021	P	102	05-DEC-08	05-JUN-09	0
1002	1022	P	108	20-MAR-09	20-AUG-09	0
1004	1025	P	106	20-DEC-08	20-MAR-09	0
1005	1026	P	106	20-DEC-08	20-MAR-09	0
1006	1027	P	106	20-DEC-08	20-MAR-09	0
1007	1028	P	106	20-DEC-08	20-MAR-09	0
1003	1024	P	106	20-DEC-08	20-MAR-09	1
1008	1029	P	106	20-DEC-08	05-FEB-09	1
1009	1030	P	106	20-DEC-08	05-FEB-09	1
1010	1030	P	106	06-FEB-09	06-JUL-09	0

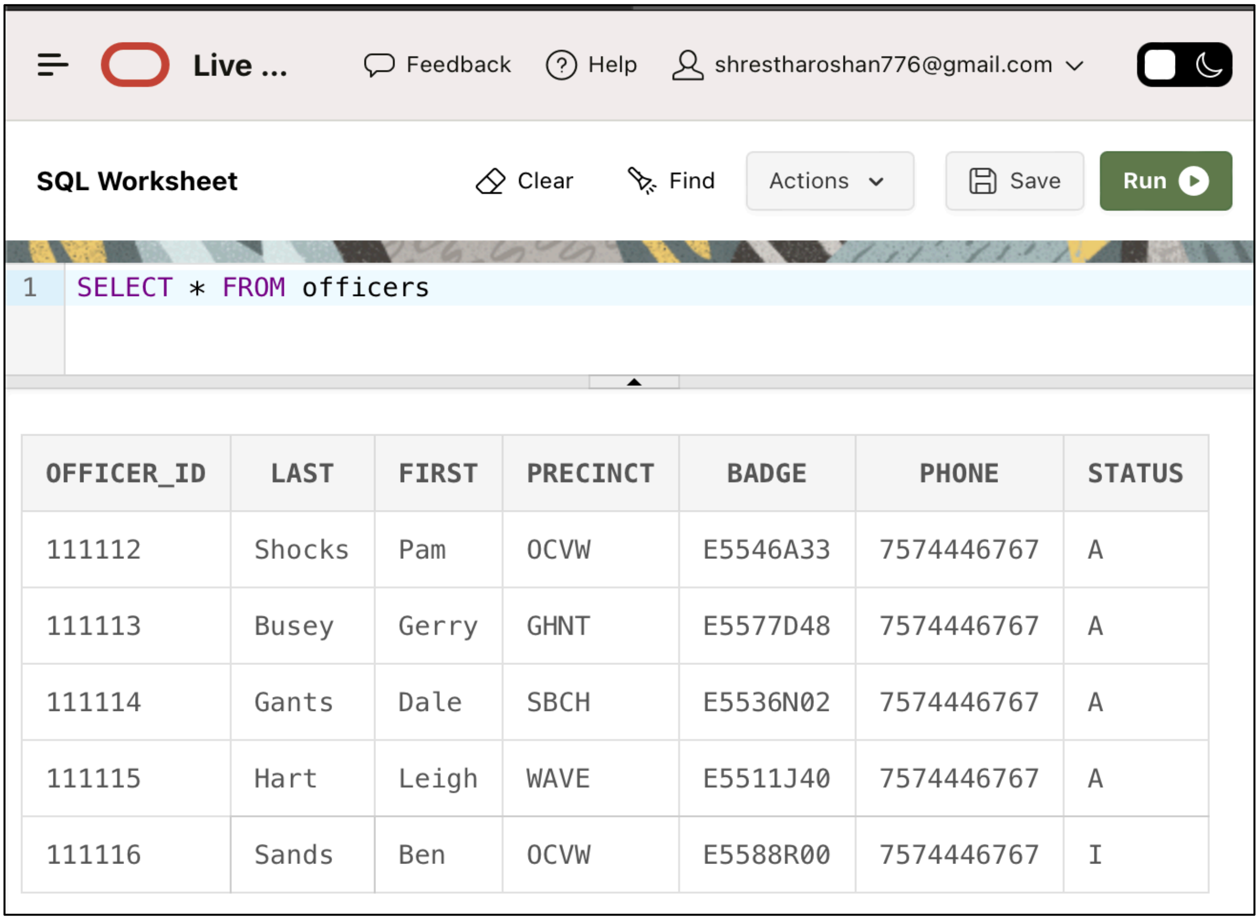
Table : prob_officers



```
SELECT * FROM prob_officers
```

PROB_ID	LAST	FIRST	STREET	CITY	STATE	ZIP	PHONE	EMAIL	STATUS	MGR_ID	PAGER#
100	Peek	Susan	-	Virginia Beach	-	-	-	-	A	-	-
102	Speckle	Jeff	-	Virginia Beach	-	-	-	-	A	100	-
104	Boyle	Chris	-	Virginia Beach	-	-	-	-	A	100	-
106	Taps	George	-	Chesapeake	-	-	-	-	A	-	-
108	Ponds	Terry	-	Chesapeake	-	-	-	-	A	106	-
110	Hawk	Fred	-	Chesapeake	-	-	-	-	I	106	-

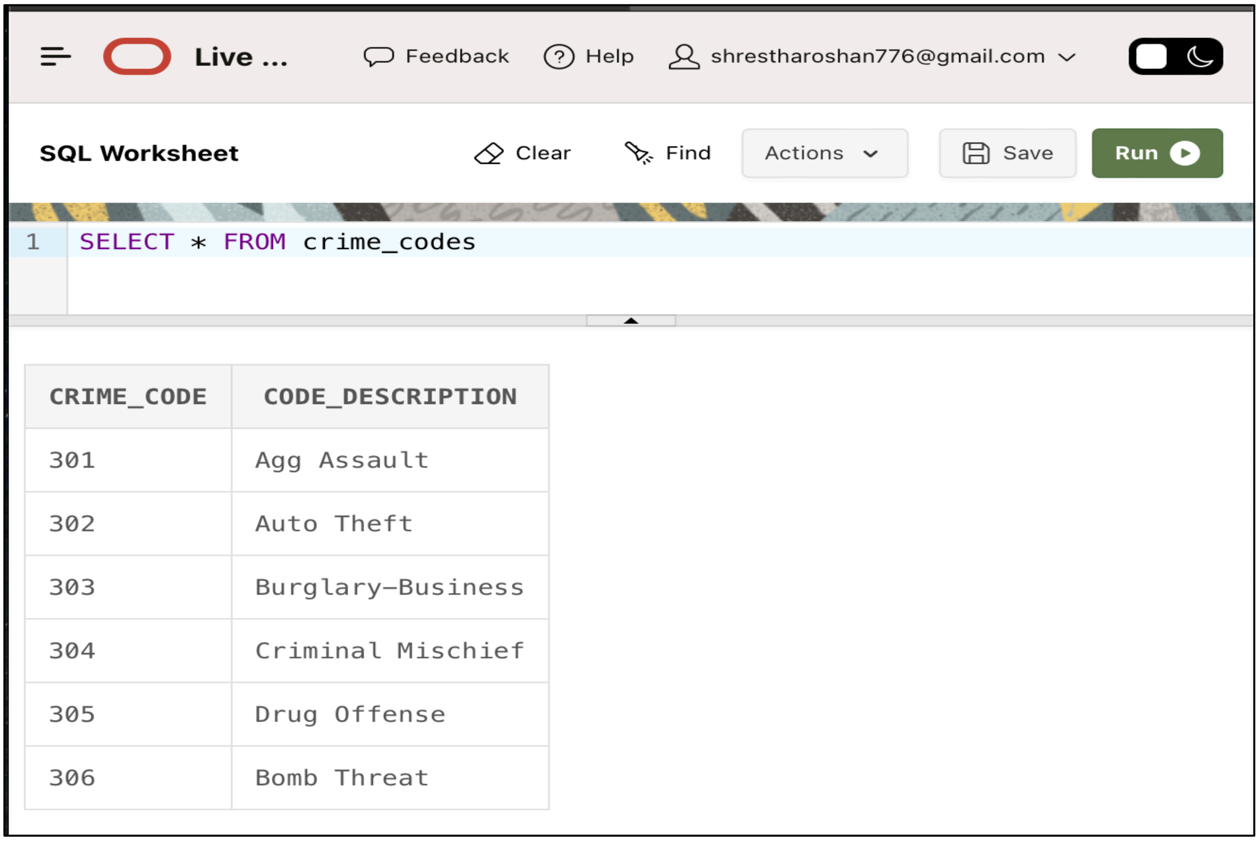
Table: officers



```
SELECT * FROM officers
```

OFFICER_ID	LAST	FIRST	PRECINCT	BADGE	PHONE	STATUS
111112	Shocks	Pam	OCVW	E5546A33	7574446767	A
111113	Busey	Gerry	GHNT	E5577D48	7574446767	A
111114	Gants	Dale	SBCH	E5536N02	7574446767	A
111115	Hart	Leigh	WAVE	E5511J40	7574446767	A
111116	Sands	Ben	OCVW	E5588R00	7574446767	I

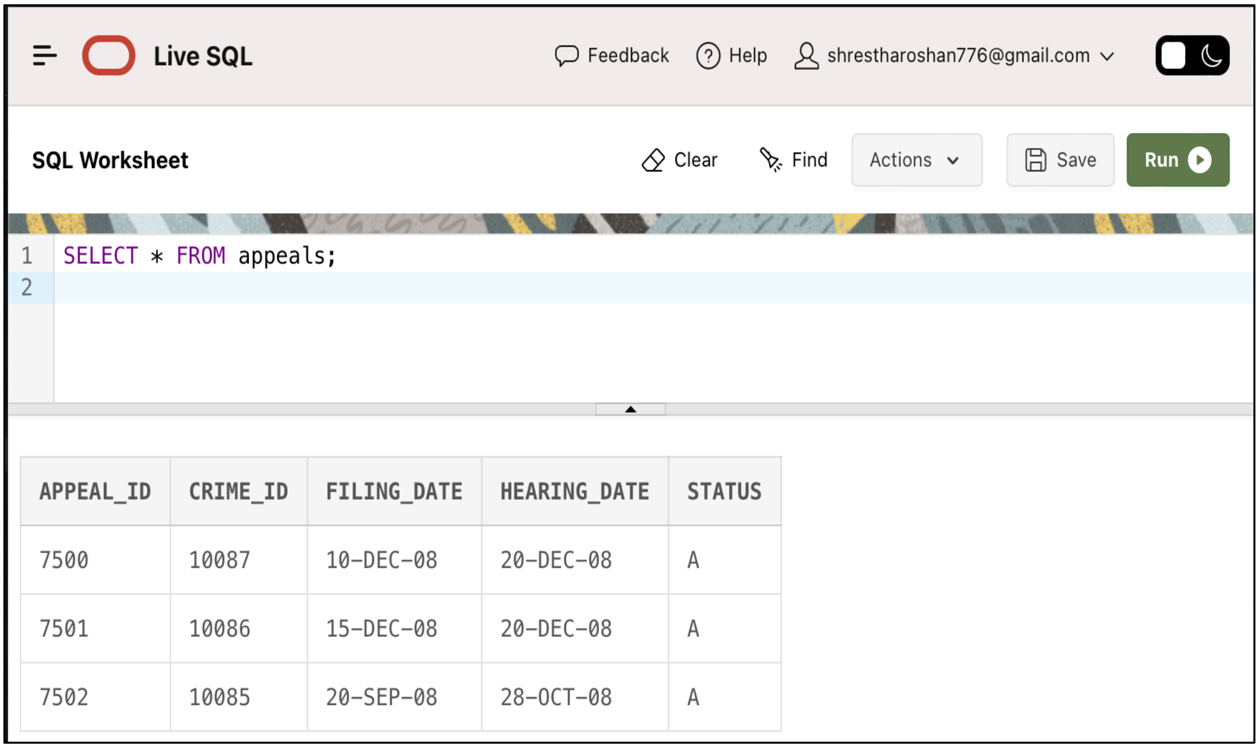
Table: crime_codes



```sql
SELECT * FROM crime_codes
```

CRIME_CODE	CODE_DESCRIPTION
301	Agg Assault
302	Auto Theft
303	Burglary-Business
304	Criminal Mischief
305	Drug Offense
306	Bomb Threat

Table: appeals

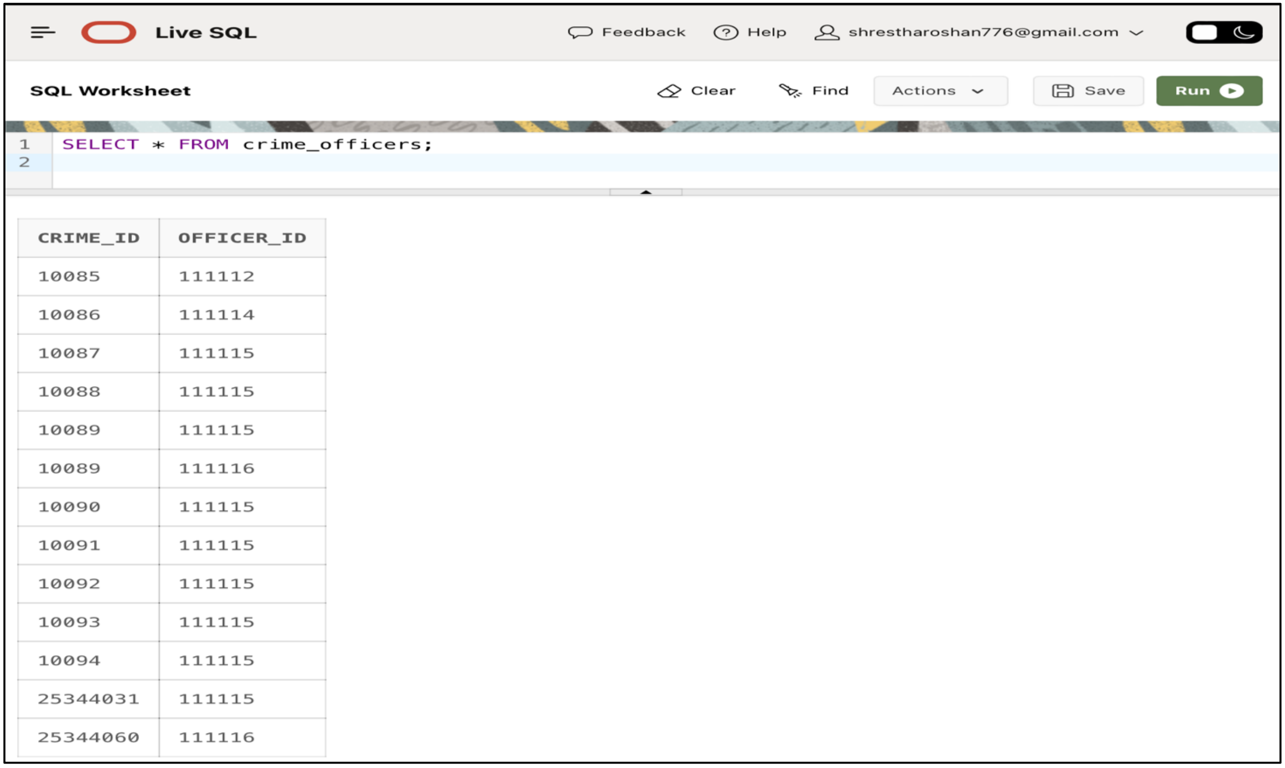


```sql
SELECT * FROM appeals;

```

APPEAL_ID	CRIME_ID	FILING_DATE	HEARING_DATE	STATUS
7500	10087	10-DEC-08	20-DEC-08	A
7501	10086	15-DEC-08	20-DEC-08	A
7502	10085	20-SEP-08	28-OCT-08	A

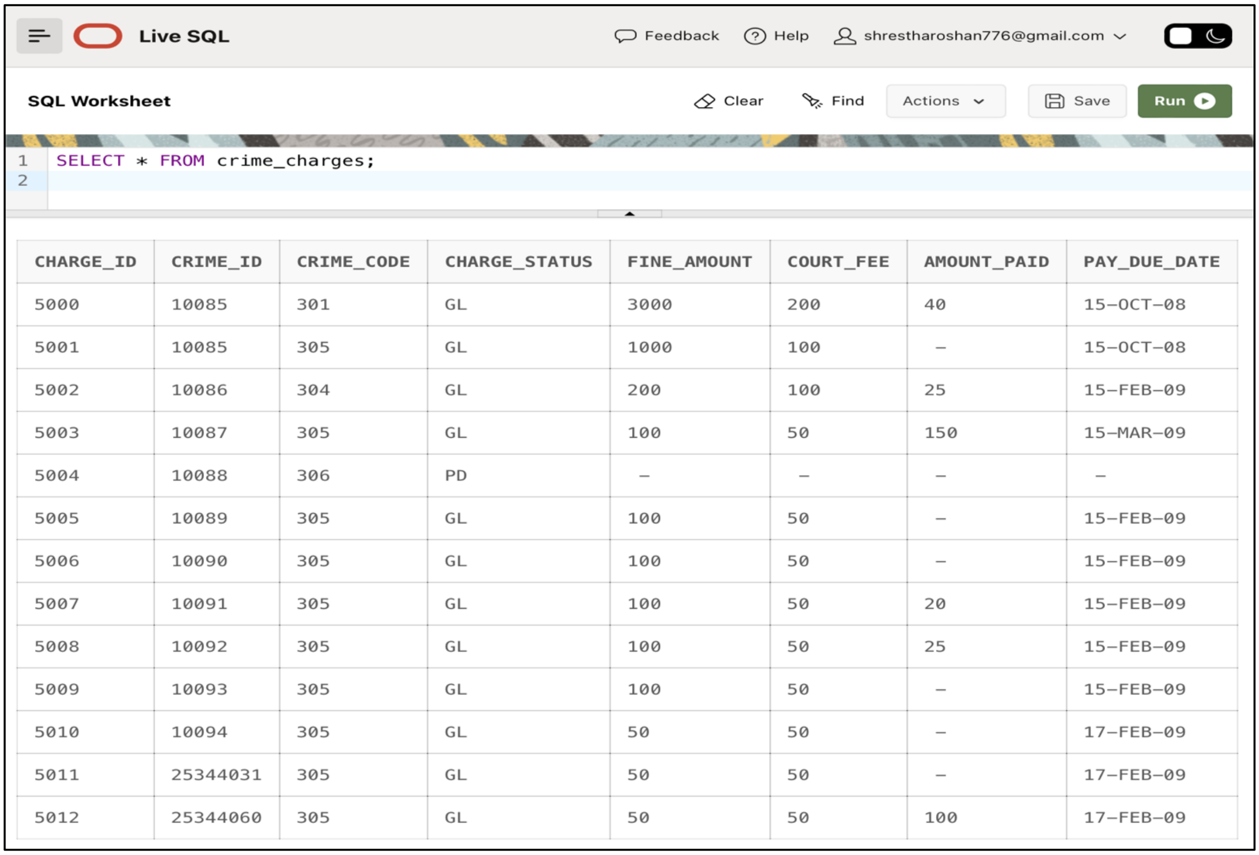
Table: crime_officers



```
SELECT * FROM crime_officers;
```

CRIME_ID	OFFICER_ID
10085	111112
10086	111114
10087	111115
10088	111115
10089	111115
10089	111116
10090	111115
10091	111115
10092	111115
10093	111115
10094	111115
25344031	111115
25344060	111116

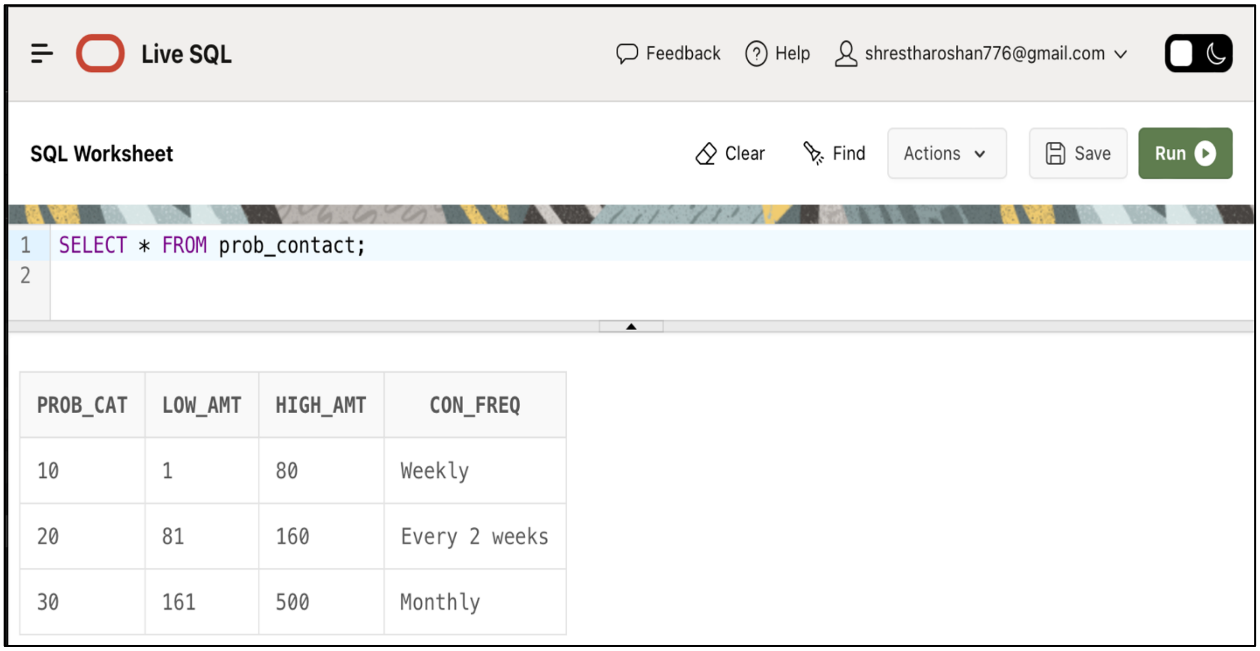
Table: crime_charges



```
SELECT * FROM crime_charges;
```

CHARGE_ID	CRIME_ID	CRIME_CODE	CHARGE_STATUS	FINE_AMOUNT	COURT_FEE	AMOUNT_PAID	PAY_DUE_DATE
5000	10085	301	GL	3000	200	40	15–OCT–08
5001	10085	305	GL	1000	100	–	15–OCT–08
5002	10086	304	GL	200	100	25	15–FEB–09
5003	10087	305	GL	100	50	150	15–MAR–09
5004	10088	306	PD	–	–	–	–
5005	10089	305	GL	100	50	–	15–FEB–09
5006	10090	305	GL	100	50	–	15–FEB–09
5007	10091	305	GL	100	50	20	15–FEB–09
5008	10092	305	GL	100	50	25	15–FEB–09
5009	10093	305	GL	100	50	–	15–FEB–09
5010	10094	305	GL	50	50	–	17–FEB–09
5011	25344031	305	GL	50	50	–	17–FEB–09
5012	25344060	305	GL	50	50	100	17–FEB–09

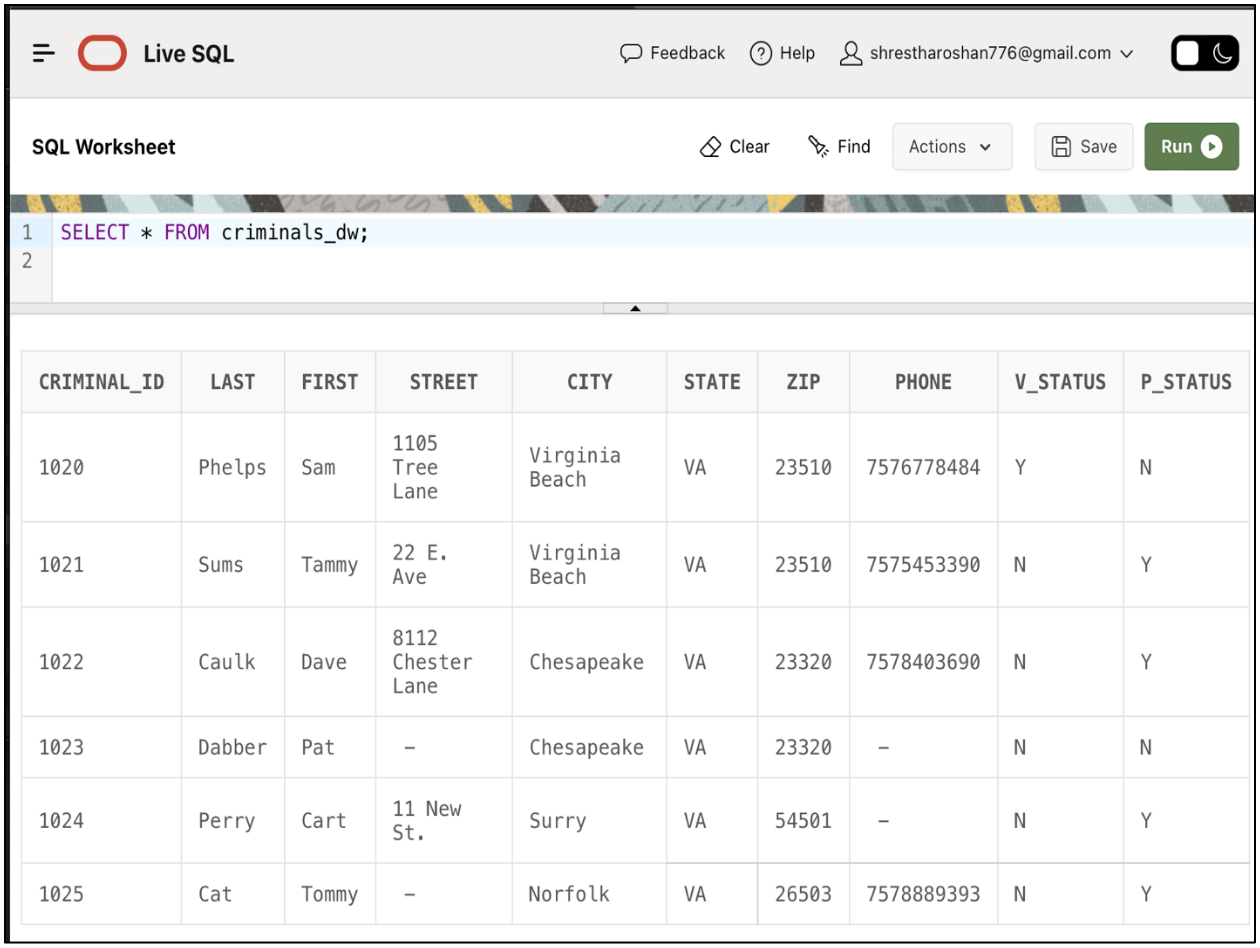
Table: prob_contact



```
SELECT * FROM prob_contact;
```

PROB_CAT	LOW_AMT	HIGH_AMT	CON_FREQ
10	1	80	Weekly
20	81	160	Every 2 weeks
30	161	500	Monthly

Table: criminals_dw



```
SELECT * FROM criminals_dw;
```

CRIMINAL_ID	LAST	FIRST	STREET	CITY	STATE	ZIP	PHONE	V_STATUS	P_STATUS
1020	Phelps	Sam	1105 Tree Lane	Virginia Beach	VA	23510	7576778484	Y	N
1021	Sums	Tammy	22 E. Ave	Virginia Beach	VA	23510	7575453390	N	Y
1022	Caulk	Dave	8112 Chester Lane	Chesapeake	VA	23320	7578403690	N	Y
1023	Dabber	Pat	-	Chesapeake	VA	23320	-	N	N
1024	Perry	Cart	11 New St.	Surry	VA	54501	-	N	Y
1025	Cat	Tommy	-	Norfolk	VA	26503	7578889393	N	Y

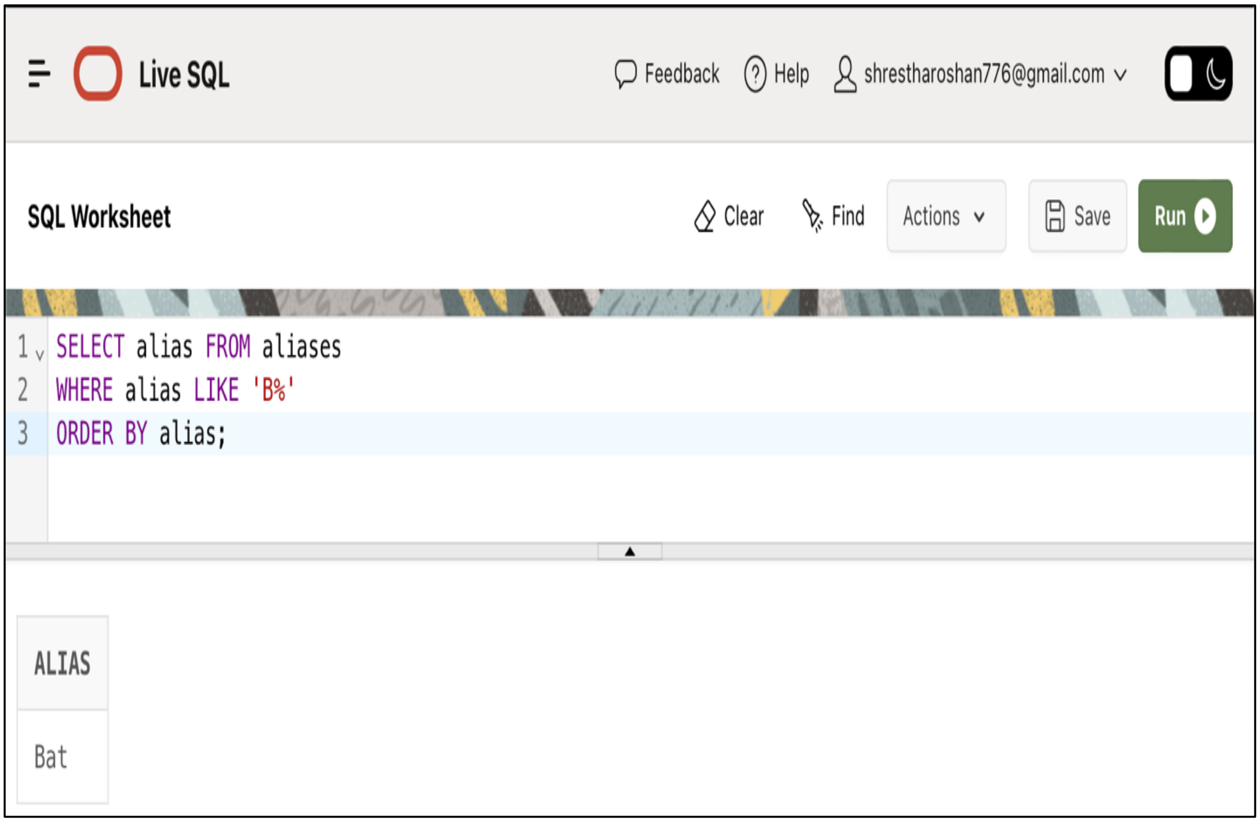
Practical Activity #1

1. List all criminal aliases beginning with the letter B.

In order to list all the criminal aliases that begins with the letter **B**, we need to use **LIKE** operator which uses pattern matching method to search for the value in respective column's value. The query I used is as below:

```
SELECT alias_id, criminal_id, alias
FROM aliases
WHERE alias LIKE 'B%'
```

The **LIKE** keyword does a pattern matching search where the pattern 'B%' specifies that we need to look for aliases starting with the letter B, followed by any number of characters. The '%' symbol is a wildcard character that matches any number of characters. The output of the above script is below:



```
SELECT alias FROM aliases
WHERE alias LIKE 'B%'
ORDER BY alias;
```

```
ALIAS
Bat
```

2. List all crimes that occurred (were charged) during the month November 2008. List the crime ID, criminal ID, date charged, and classification.

To list all the crimes that occurred during the given month and year, we need to convert the dates into the **DD-MON-YY** format using the below query:

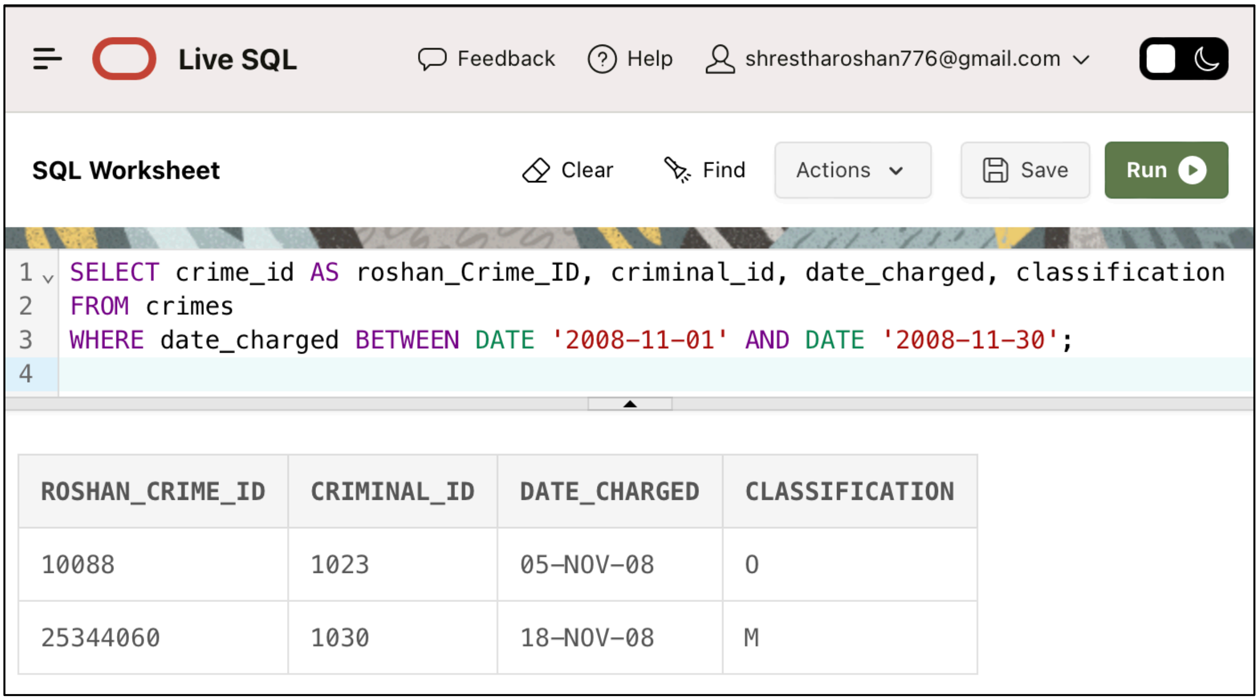
```
Alter session set nls_date_format='DD-MON-YY';
```

This is used to modify the default date format in Oracle for the current session. Dates are displayed in the format 'day-month-year' when the **NLS_DATE_FORMAT** parameter is set to **'DD-MON-YY'**.

After doing the necessary changes we can use the below query to reterive the required data:

SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, classification
FROM crimes
WHERE date_charged BETWEEN DATE '2008-11-01' AND DATE '2008-11-30';

The SQL query above retrieves the **crime_id** (renamed as **roshan_Crime_ID** in the output), **criminal_id**, **date_charged,** and **classification** from the **crimes** table. It narrows the results to only those rows where the date charged occurs between November 1, 2008, and November 30, 2008.The output after executing the above query can be visualized in the snippet below:



```
SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, classification
FROM crimes
WHERE date_charged BETWEEN DATE '2008-11-01' AND DATE '2008-11-30';
```

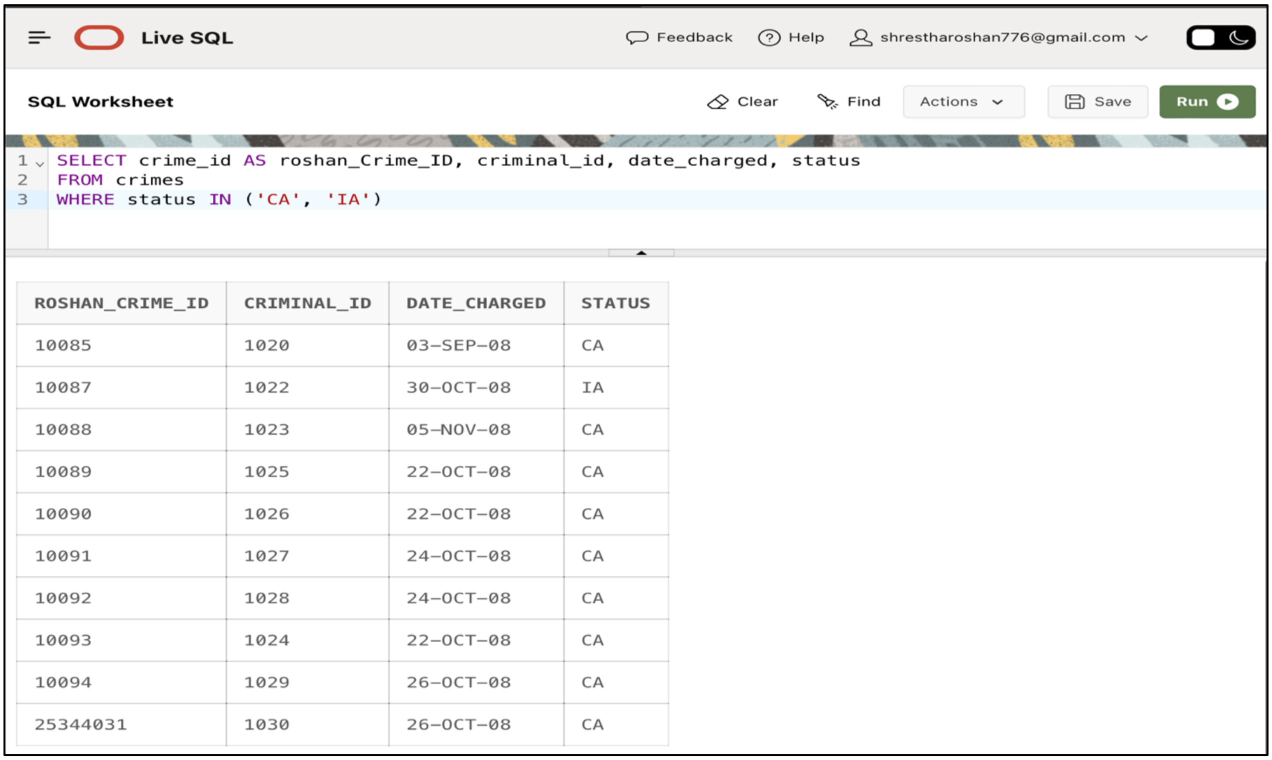
ROSHAN_CRIME_ID	CRIMINAL_ID	DATE_CHARGED	CLASSIFICATION
10088	1023	05-NOV-08	O
25344060	1030	18-NOV-08	M

3. List all crimes with a status of CA (can appeal) or IA (in appeal). List the crime ID, criminal ID, date charged, and status.

In order to view the list of all crimes that has status of **CA(can appeal)** or **IA(In appeal)**. We can execute the query below:

SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, status
FROM crimes
WHERE status IN ('CA', 'IA')

Here using the **AS** keyword, we are selecting the **crime_id** and storing it as **roshan_Crime_ID** which will be new column name while displaying the result, along with**, criminal_id, date_charged, status**. The **FROM** keyword specifies that from which table we are going to select data, here we are taking data from **crimes** table, and finally using the **WHERE** keyword we are applying the condition where the data will be filtered if it has value as **CA** or **IA**, in the status column. The output from the query execution can be observed below:



```
SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, status
FROM crimes
WHERE status IN ('CA', 'IA')
```

ROSHAN_CRIME_ID	CRIMINAL_ID	DATE_CHARGED	STATUS
10085	1020	03-SEP-08	CA
10087	1022	30-OCT-08	IA
10088	1023	05-NOV-08	CA
10089	1025	22-OCT-08	CA
10090	1026	22-OCT-08	CA
10091	1027	24-OCT-08	CA
10092	1028	24-OCT-08	CA
10093	1024	22-OCT-08	CA
10094	1029	26-OCT-08	CA
25344031	1030	26-OCT-08	CA

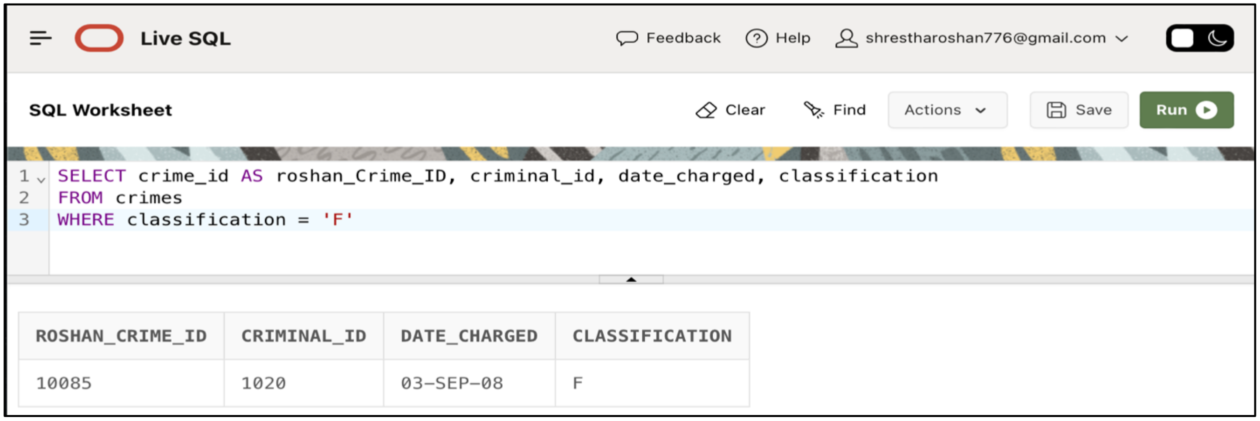
Practical Activity #2

4. List all crimes classified as a felony. List the crime ID, criminal ID, date charged, and classification. (Classification is 'F').

To retrieve the list of all crimes with classification as **felony** we can execute the below query:

SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, classification
FROM crimes
WHERE classification = 'F'

Initially the **SELECT** statement will specifies the columns that we want to include in the result, here we are trying to retrieve the data from **criminal_id, date_charged, classification** columns. And using the **FROM** statement we are representing the table in which we will apply the search query i.e. **crimes** table. Then the **WHERE** statement will specity the condition that make sure that the result will have **classification** as **'F'**. The output from the query is attached below:



```
SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, classification
FROM crimes
WHERE classification = 'F'
```

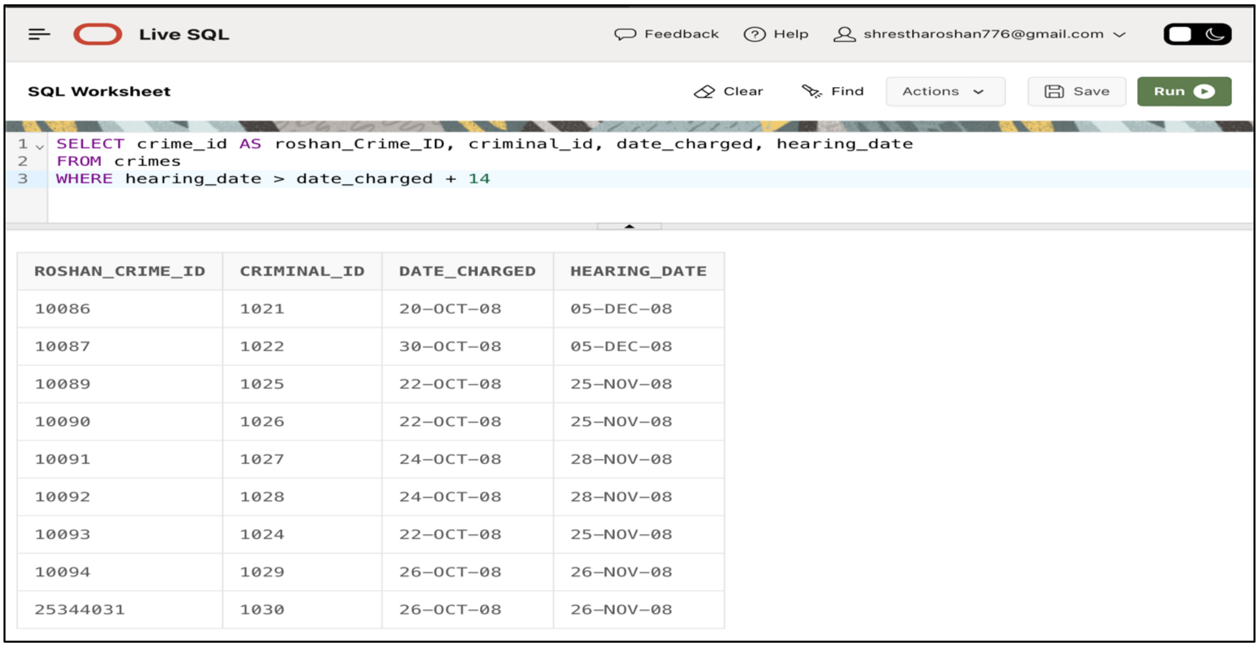
ROSHAN_CRIME_ID	CRIMINAL_ID	DATE_CHARGED	CLASSIFICATION
10085	1020	03-SEP-08	F

5. List all crimes with a hearing date more than 14 days after the date charged. List the crime ID, criminal ID, date charged, and hearing date.

To list all the crimes with a hearing date more than 14 days after the charged date, we can use the query below:

SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, hearing_date
FROM crimes
WHERE hearing_date > date_charged + 14

In this query we are trying to retrieve the list of crimes from **crimes** table where the value of **hearing_date**, is more than 14 days after the **date_charged**. And we want our result output to include the **crime_id** with name **roshan_Crime_ID**, alomg with **criminal_id, date_charged, hearing_date**. The output from the query can be visualized below:



```
SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, hearing_date
FROM crimes
WHERE hearing_date > date_charged + 14
```

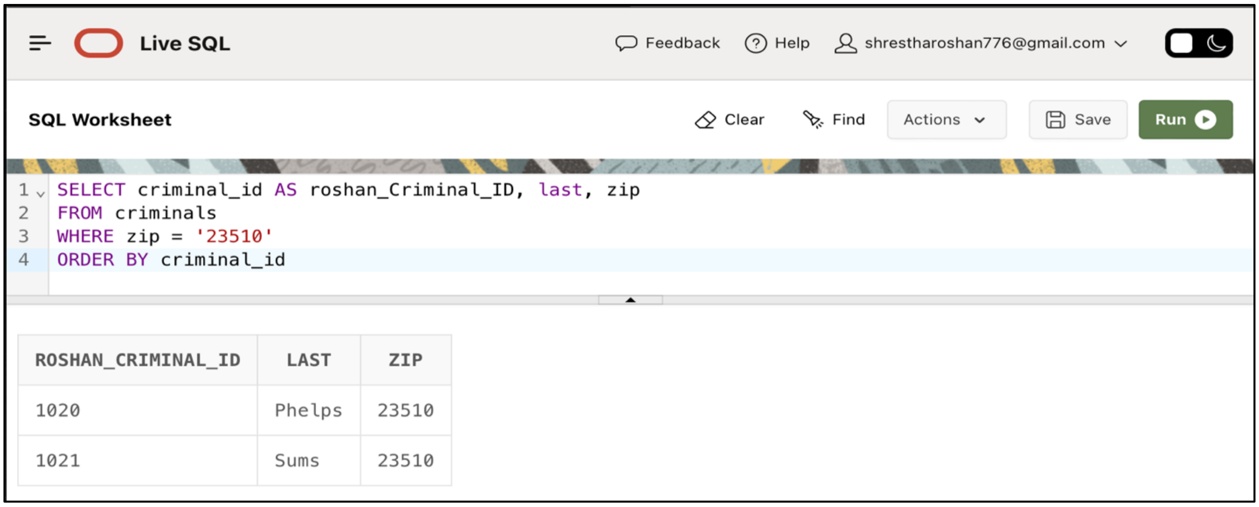
ROSHAN_CRIME_ID	CRIMINAL_ID	DATE_CHARGED	HEARING_DATE
10086	1021	20-OCT-08	05-DEC-08
10087	1022	30-OCT-08	05-DEC-08
10089	1025	22-OCT-08	25-NOV-08
10090	1026	22-OCT-08	25-NOV-08
10091	1027	24-OCT-08	28-NOV-08
10092	1028	24-OCT-08	28-NOV-08
10093	1024	22-OCT-08	25-NOV-08
10094	1029	26-OCT-08	26-NOV-08
25344031	1030	26-OCT-08	26-NOV-08

6. List all criminals with the zip code 23510. List the criminal ID, last name, and zip code. Sort the list by criminal ID.

To list all the criminals where the zip code is 23510, we can run the below query:

SELECT criminal_id AS roshan_Criminal_ID, last, zip
FROM criminals
WHERE zip = '23510'
ORDER BY criminal_id

The query above retrieves the **criminal_id** from the criminals table and renames it **roshan_Criminal_ID**. It also chooses the columns for **last** and **zip**. The query narrows the results to only those with the **zip code '23510'**. Finally, the results are ordered ascendingly by the **criminal_id** column. The output of the executed query is represented as below:



```
SELECT criminal_id AS roshan_Criminal_ID, last, zip
FROM criminals
WHERE zip = '23510'
ORDER BY criminal_id
```

ROSHAN_CRIMINAL_ID	LAST	ZIP
1020	Phelps	23510
1021	Sums	23510

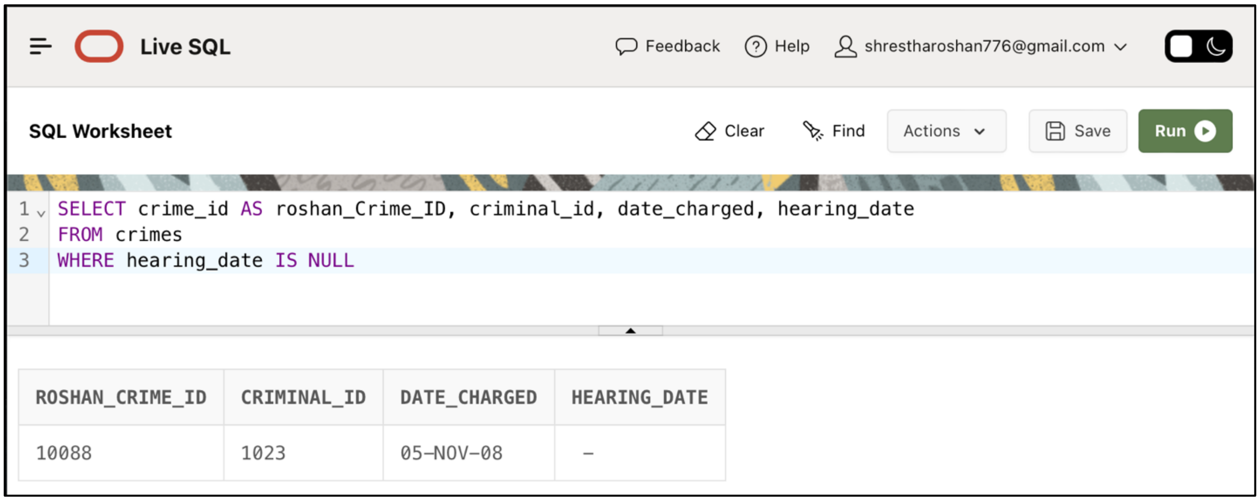
Practical Activity #3

7. List all crimes that don't have a hearing date scheduled. List the crime ID, criminal ID, date charged, and hearing date.

To retrieve the data of all crimes that don't have a hearing date scheduled we can execute the query below:

SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, hearing_date
FROM crimes
WHERE hearing_date IS NULL

The SQL query above retrieves the **crime_id** (renamed **roshan_Crime_ID** in result), criminal ID, date charged, and hearing date from the crimes table. It restricts the results to entries where the hearing date is NULL (not scheduled). The output of the query is attached below:



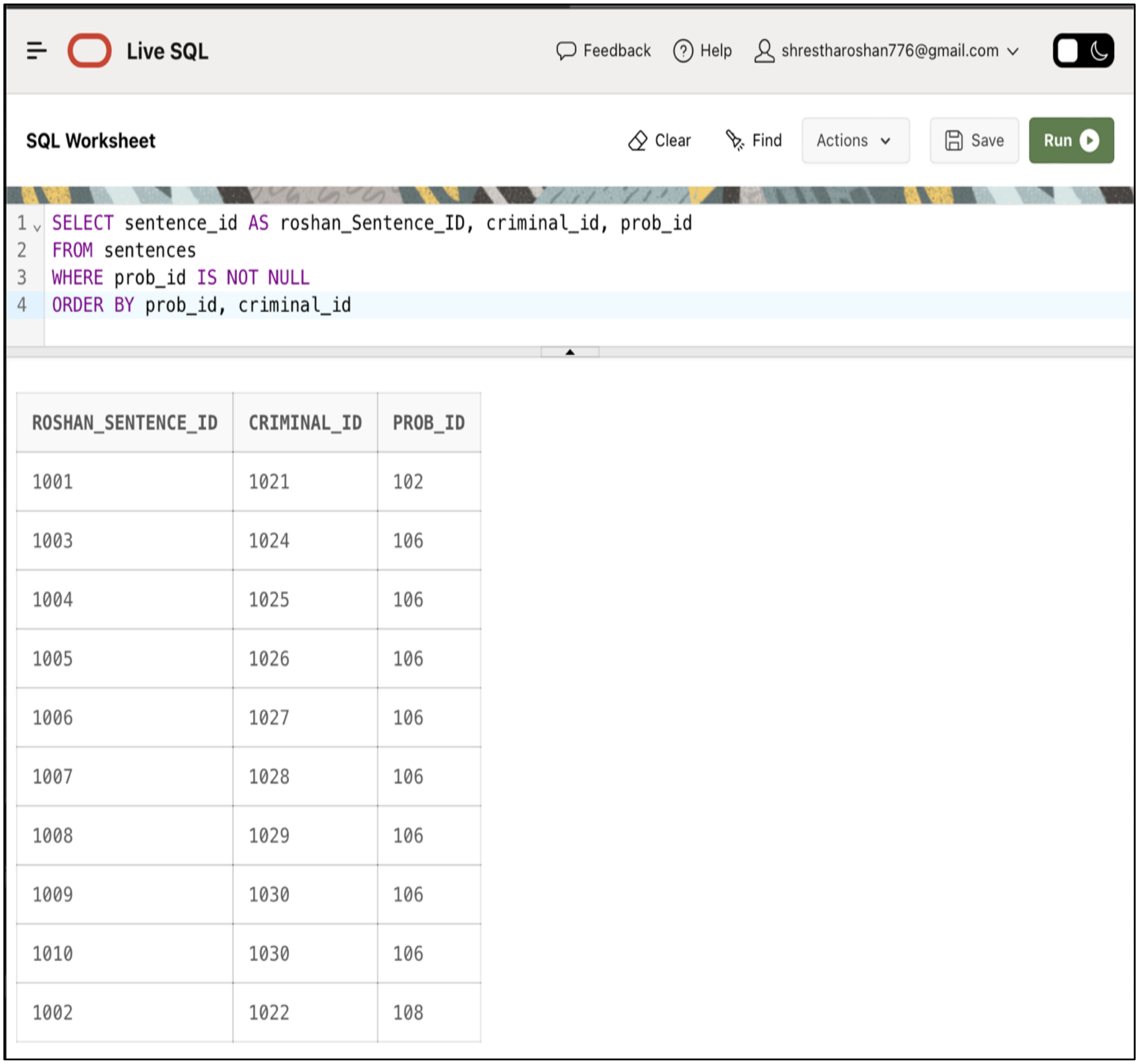
```
SELECT crime_id AS roshan_Crime_ID, criminal_id, date_charged, hearing_date
FROM crimes
WHERE hearing_date IS NULL
```

ROSHAN_CRIME_ID	CRIMINAL_ID	DATE_CHARGED	HEARING_DATE
10088	1023	05-NOV-08	-

8. List all sentences with a probation officer assigned. List the sentence ID, criminal ID, and probation officer ID. Sort the list by probation officer ID and then criminal ID.
In order to list down all the sentences which has a probation officer assigned, we can use the query below:

SELECT sentence_id AS roshan_Sentence_ID, criminal_id, prob_id
FROM sentences
WHERE prob_id IS NOT NULL
ORDER BY prob_id, criminal_id

From the sentences table, the SQL query pulls the **sentence_id** (renamed **roshan_Sentence_ID** in the output), **criminal_id**, and **prob_id**. It restricts the results to rows where the **prob_id** is not **NULL**. The results are then ordered by **prob_id,** first, followed by **criminal_id**. The output is attached bwlow:



```
SELECT sentence_id AS roshan_Sentence_ID, criminal_id, prob_id
FROM sentences
WHERE prob_id IS NOT NULL
ORDER BY prob_id, criminal_id
```

ROSHAN_SENTENCE_ID	CRIMINAL_ID	PROB_ID
1001	1021	102
1003	1024	106
1004	1025	106
1005	1026	106
1006	1027	106
1007	1028	106
1008	1029	106
1009	1030	106
1010	1030	106
1002	1022	108

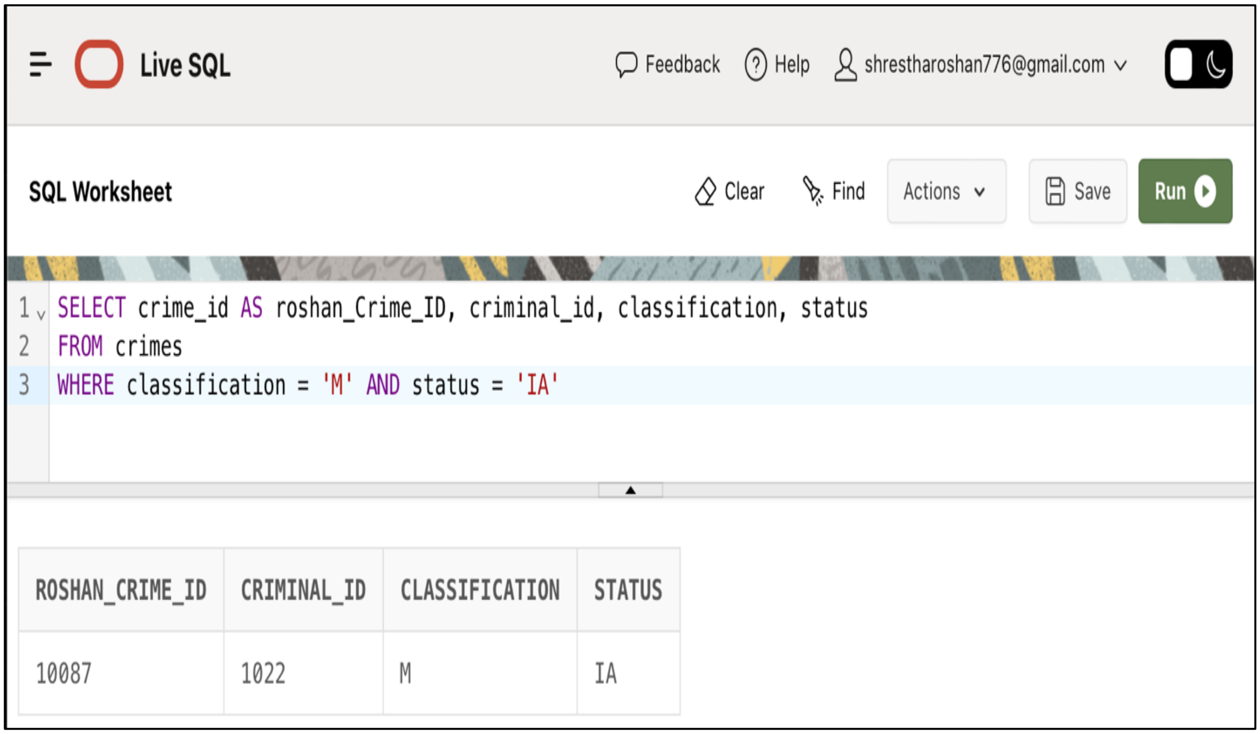
Practical Activity #4

9. List all crimes that are classified as misdemeanors (classification ='M') and are currently in appeal (stat is 'IA'). List the crime ID, criminal ID, classification, and status.

To list down all the crimes that are under the classification as misdemeanors and are currently under the status as in appeal(IA), we can execute the query below:

SELECT crime_id AS roshan_Crime_ID, criminal_id, classification, status
FROM crimes
WHERE classification = 'M' AND status = 'IA'

The SQL query above retrieves the **crime_id** (renamed as **roshan_Crime_ID** in the output), **criminal_id, classification and status** from the **crimes** table. It narrows the results to only those with the categorization 'M' (misdemeanour) and the status 'IA' (in appeal). The output from the query is attached below:



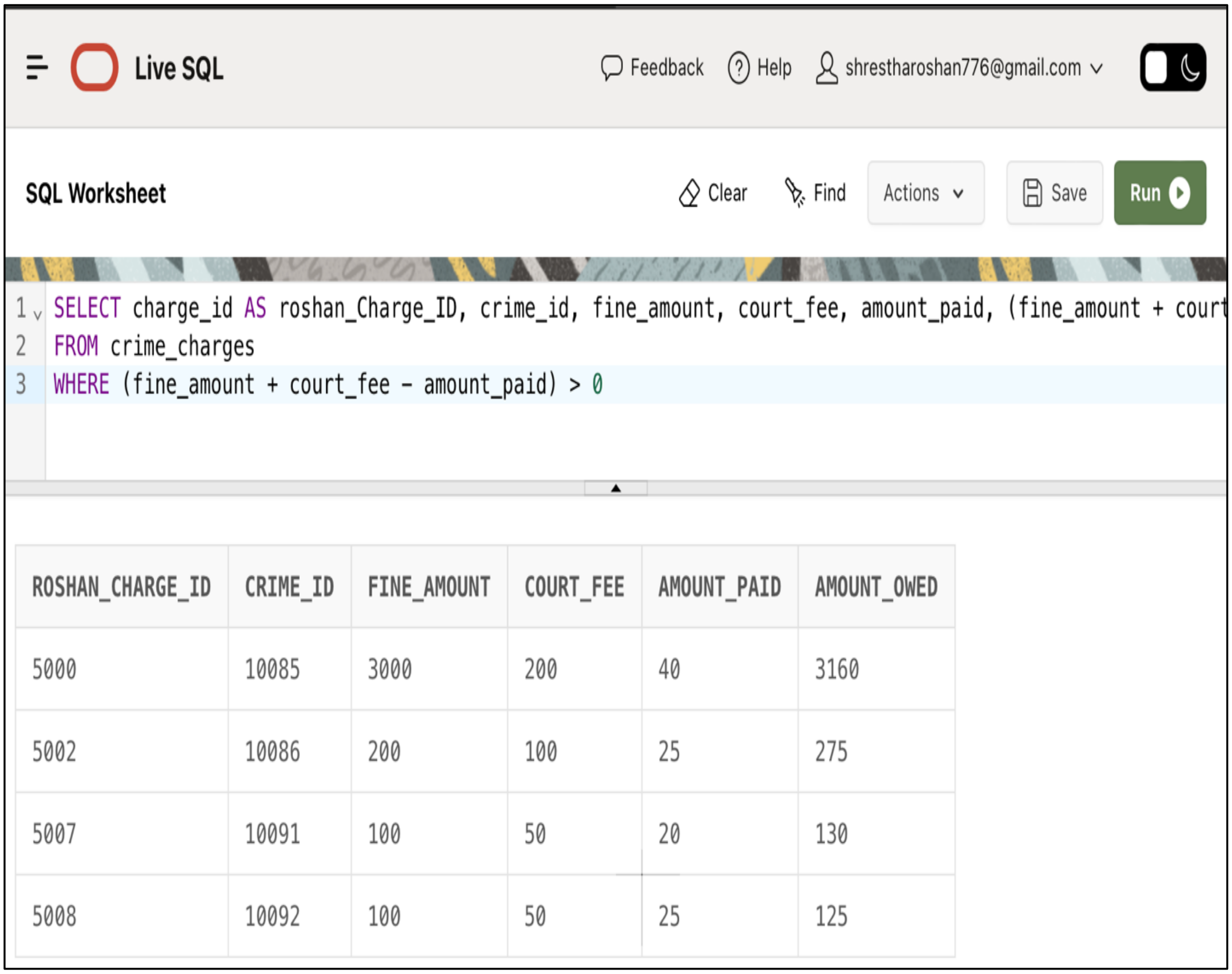
```
SELECT crime_id AS roshan_Crime_ID, criminal_id, classification, status
FROM crimes
WHERE classification = 'M' AND status = 'IA'
```

ROSHAN_CRIME_ID	CRIMINAL_ID	CLASSIFICATION	STATUS
10087	1022	M	IA

10. List all crime charges with a balance owed. List the charge ID, crime ID, fine amount, court fee, amount paid, and amount owed.
To list down the all the charges with a balance owed we can use the query below:

SELECT charge_id AS roshan_Charge_ID, crime_id, fine_amount, court_fee, amount_paid, (fine_amount + court_fee - amount_paid) AS amount_owed
FROM crime_charges
WHERE (fine_amount + court_fee - amount_paid) > 0

The SQL query above picks the **charge_id** (renamed as **roshan_Charge_ID**), **crime_id**, **fine_amount**, **court_fee**, and **amount_paid**, and calculates the amount owed as the difference between the total of the fine amount and the amount paid. It returns rows when the calculated amount owed exceeds zero. The output is attached below:



```
SELECT charge_id AS roshan_Charge_ID, crime_id, fine_amount, court_fee, amount_paid, (fine_amount + court
FROM crime_charges
WHERE (fine_amount + court_fee - amount_paid) > 0
```

ROSHAN_CHARGE_ID	CRIME_ID	FINE_AMOUNT	COURT_FEE	AMOUNT_PAID	AMOUNT_OWED
5000	10085	3000	200	40	3160
5002	10086	200	100	25	275
5007	10091	100	50	20	130
5008	10092	100	50	25	125