**Database Design – 2023S**
**Student ID:** 901142
**Student Name:** Roshan Shrestha
**Practical Activity#25#26#27#28**
---
The list of all the created tables after executing the provided SQL scripts are listed below as following:

**Table: Customers**

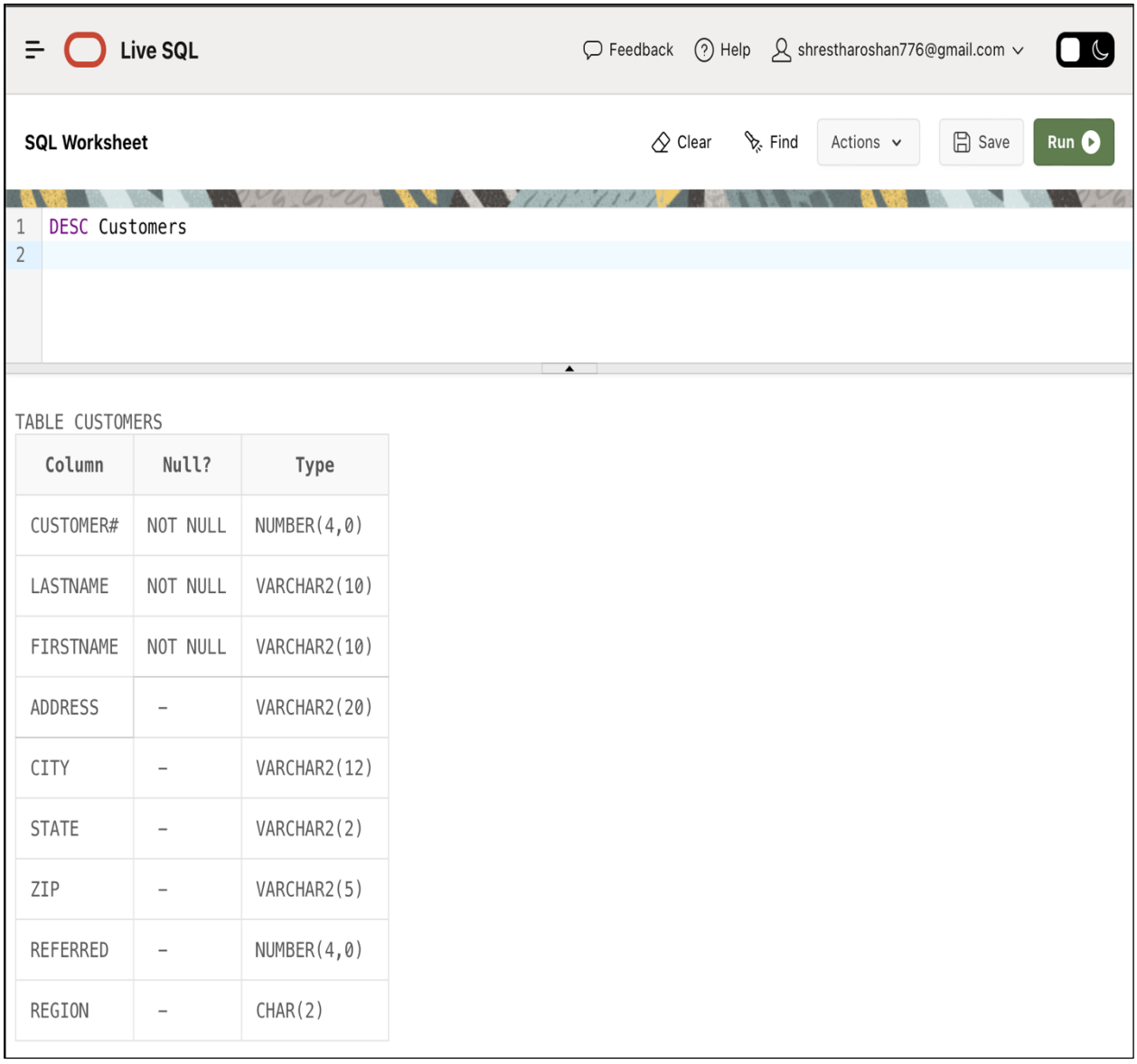



TABLE CUSTOMERS

| Column | Null? | Type |
|---|---|---|
| CUSTOMER# | NOT NULL | NUMBER(4,0) |
| LASTNAME | NOT NULL | VARCHAR2(10) |
| FIRSTNAME | NOT NULL | VARCHAR2(10) |
| ADDRESS | - | VARCHAR2(20) |
| CITY | - | VARCHAR2(12) |
| STATE | - | VARCHAR2(2) |
| ZIP | - | VARCHAR2(5) |
| REFERRED | - | NUMBER(4,0) |
| REGION | - | CHAR(2) |

## Table: Orders

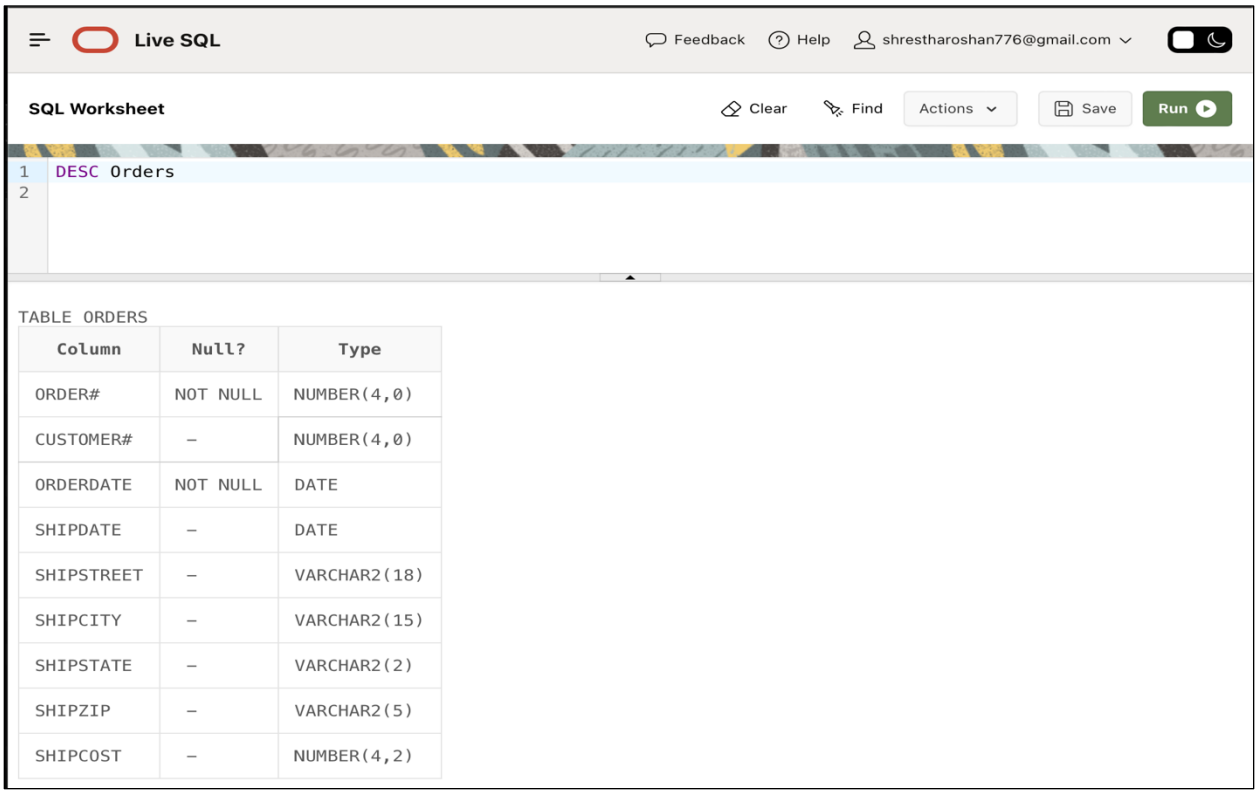



TABLE ORDERS

| Column | Null? | Type |
|---|---|---|
| ORDER# | NOT NULL | NUMBER(4,0) |
| CUSTOMER# | – | NUMBER(4,0) |
| ORDERDATE | NOT NULL | DATE |
| SHIPDATE | – | DATE |
| SHIPSTREET | – | VARCHAR2(18) |
| SHIPCITY | – | VARCHAR2(15) |
| SHIPSTATE | – | VARCHAR2(2) |
| SHIPZIP | – | VARCHAR2(5) |
| SHIPCOST | – | NUMBER(4,2) |

## Table: Publisher

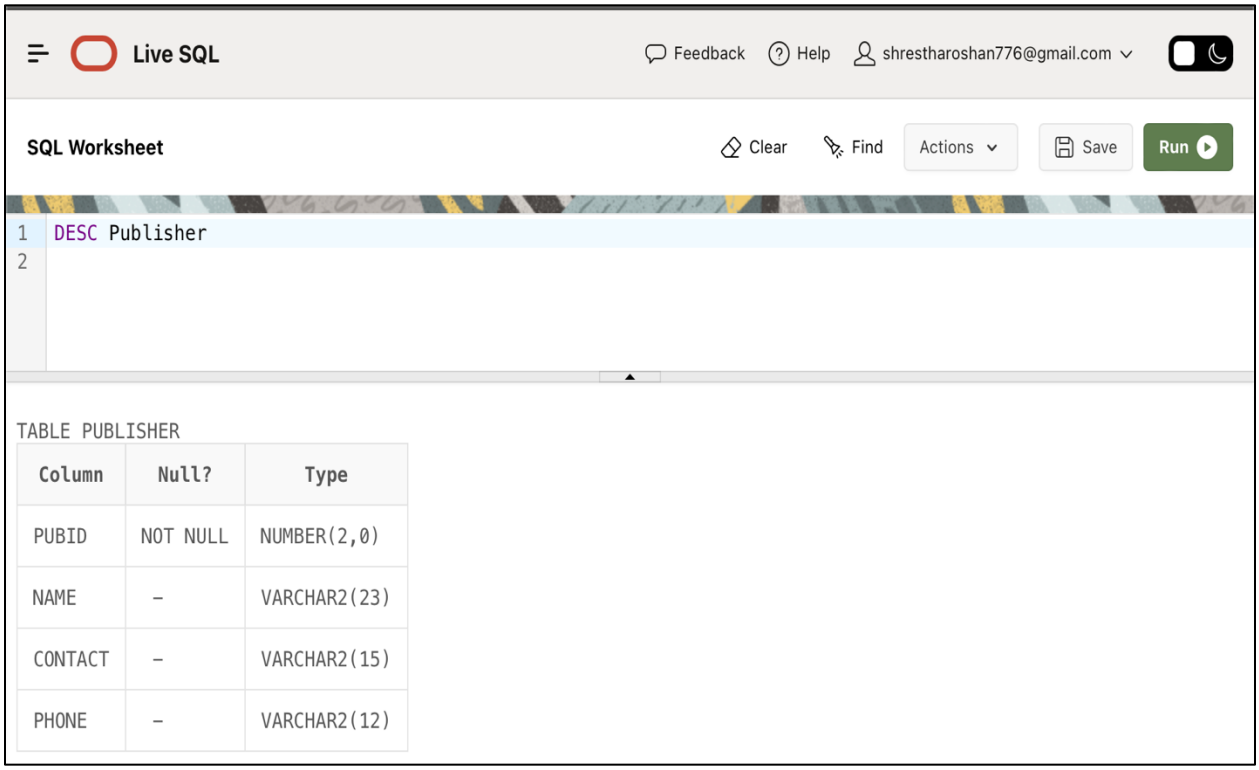



TABLE PUBLISHER

| Column | Null? | Type |
|---|---|---|
| PUBID | NOT NULL | NUMBER(2,0) |
| NAME | – | VARCHAR2(23) |
| CONTACT | – | VARCHAR2(15) |
| PHONE | – | VARCHAR2(12) |

## Table: Author

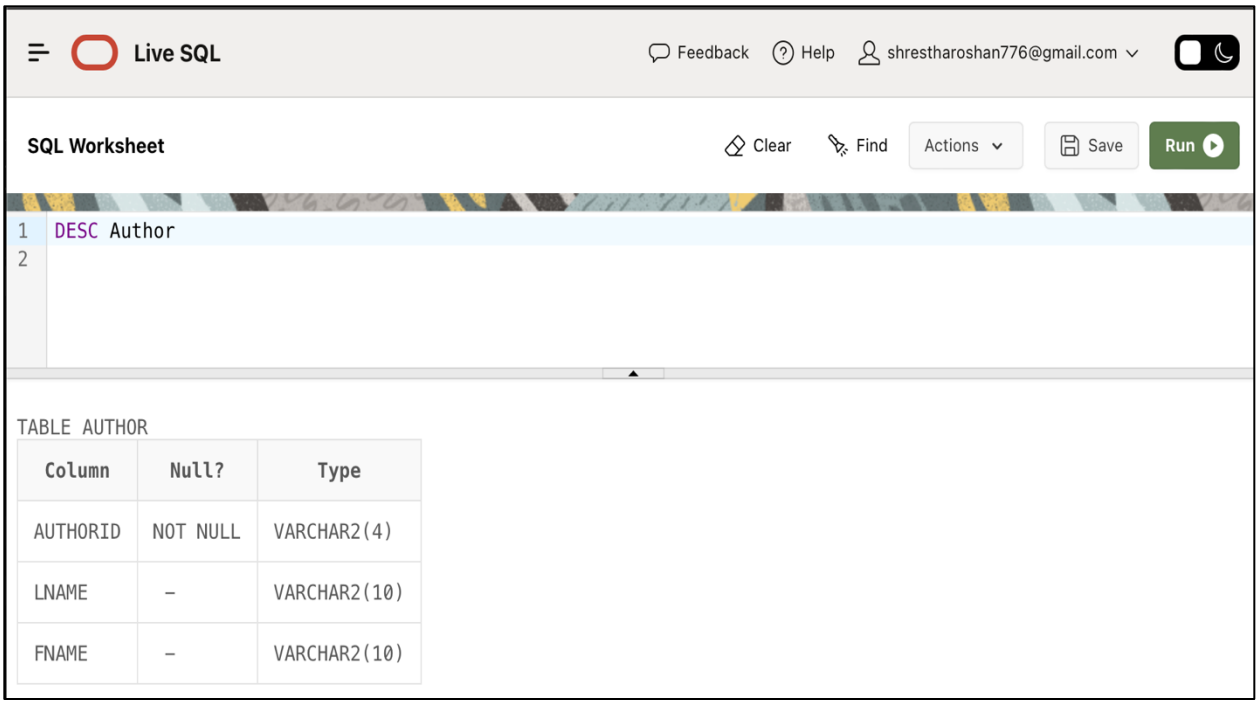



TABLE AUTHOR

| Column | Null? | Type |
|---|---|---|
| AUTHORID | NOT NULL | VARCHAR2(4) |
| LNAME | - | VARCHAR2(10) |
| FNAME | - | VARCHAR2(10) |

## Table: Books

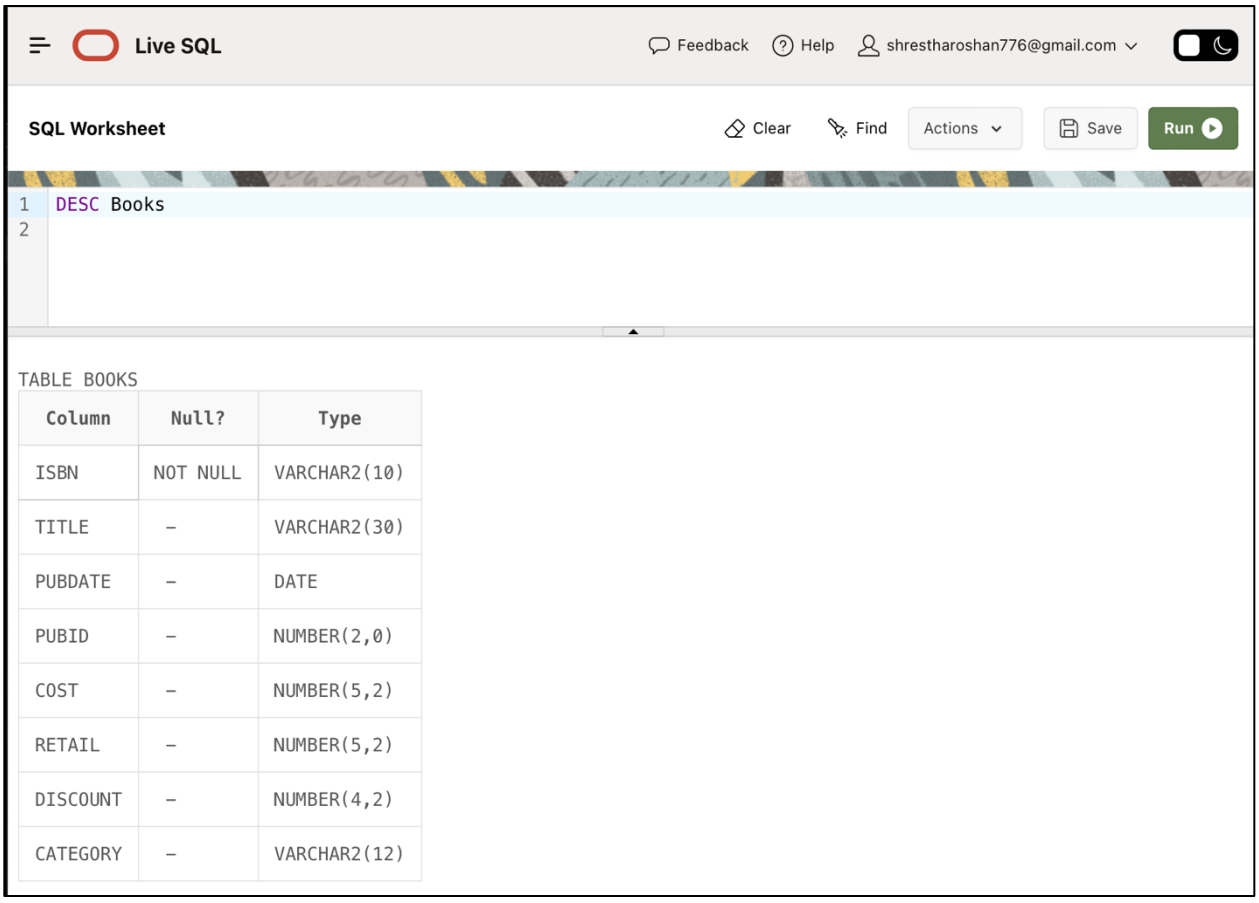



TABLE BOOKS

| Column | Null? | Type |
|---|---|---|
| ISBN | NOT NULL | VARCHAR2(10) |
| TITLE | - | VARCHAR2(30) |
| PUBDATE | - | DATE |
| PUBID | - | NUMBER(2,0) |
| COST | - | NUMBER(5,2) |
| RETAIL | - | NUMBER(5,2) |
| DISCOUNT | - | NUMBER(4,2) |
| CATEGORY | - | VARCHAR2(12) |

**Table: ORDERITEMS**

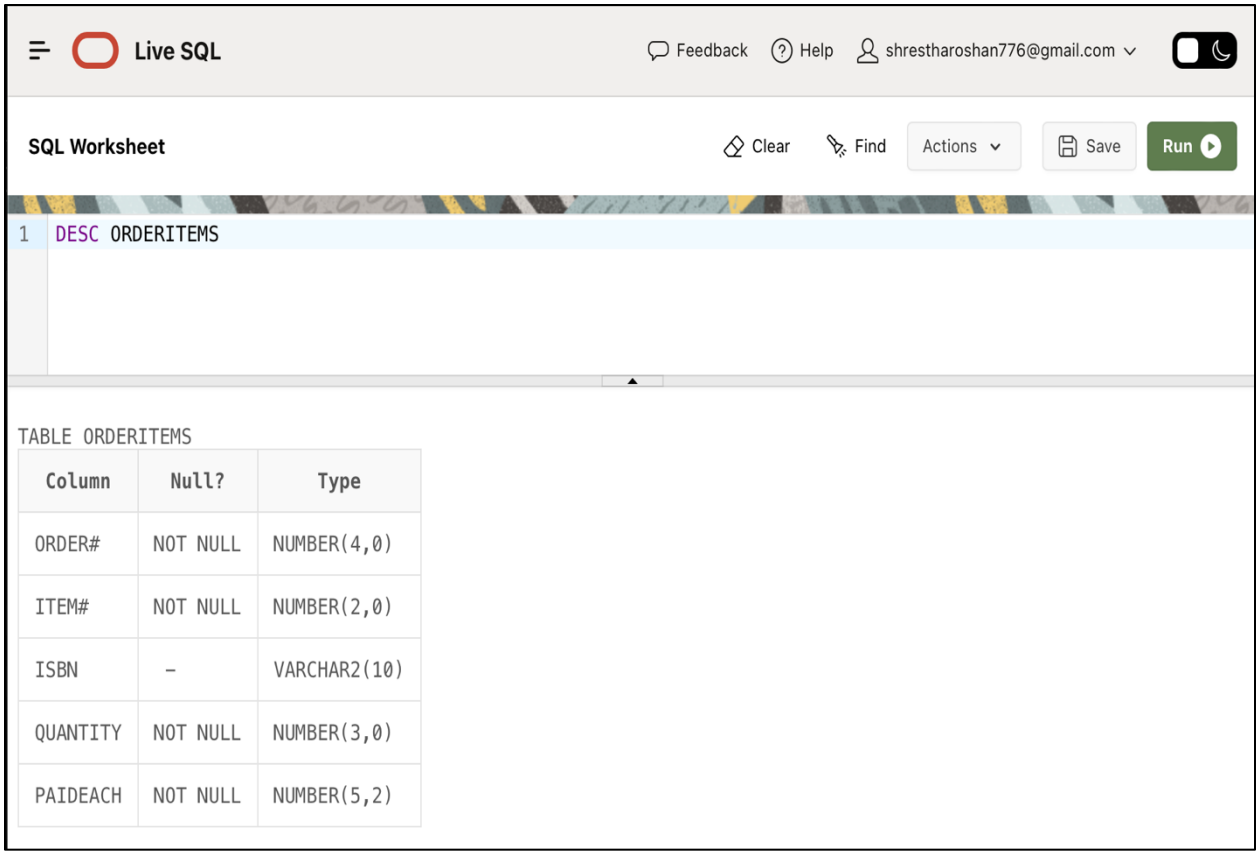



TABLE ORDERITEMS

| Column | Null? | Type |
|---|---|---|
| ORDER# | NOT NULL | NUMBER(4,0) |
| ITEM# | NOT NULL | NUMBER(2,0) |
| ISBN | - | VARCHAR2(10) |
| QUANTITY | NOT NULL | NUMBER(3,0) |
| PAIDEACH | NOT NULL | NUMBER(5,2) |

**Table: BOOKAUTHOR**

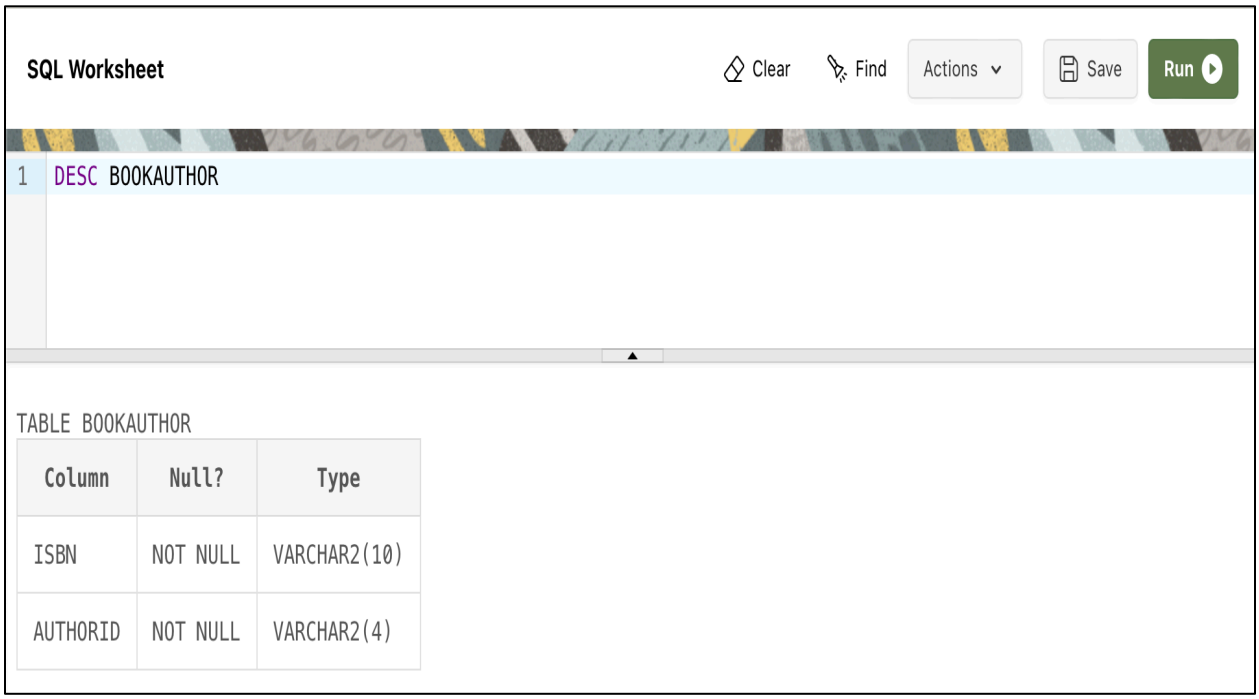



TABLE BOOKAUTHOR

| Column | Null? | Type |
|---|---|---|
| ISBN | NOT NULL | VARCHAR2(10) |
| AUTHORID | NOT NULL | VARCHAR2(4) |

## Table: acctmanager

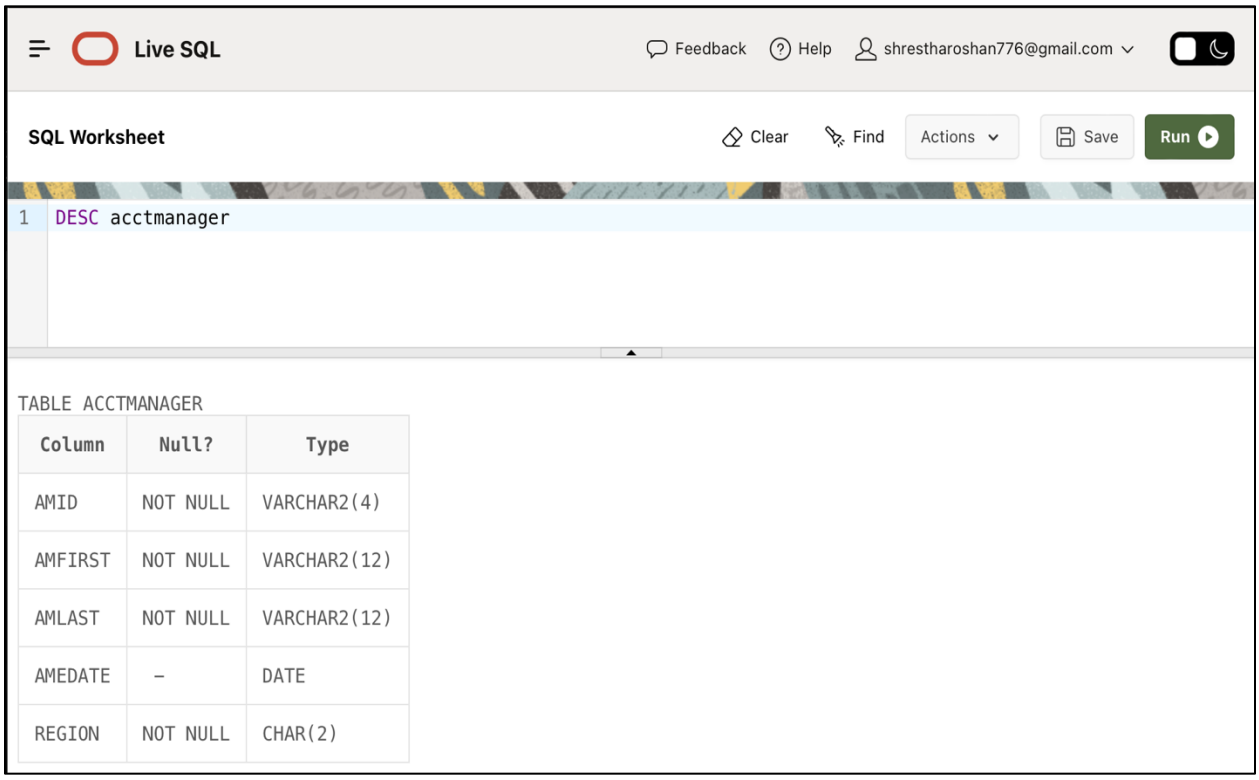



TABLE ACCTMANAGER

| Column | Null? | Type |
|---|---|---|
| AMID | NOT NULL | VARCHAR2(4) |
| AMFIRST | NOT NULL | VARCHAR2(12) |
| AMLAST | NOT NULL | VARCHAR2(12) |
| AMEDATE | - | DATE |
| REGION | NOT NULL | CHAR(2) |

## Table: acctmanager2

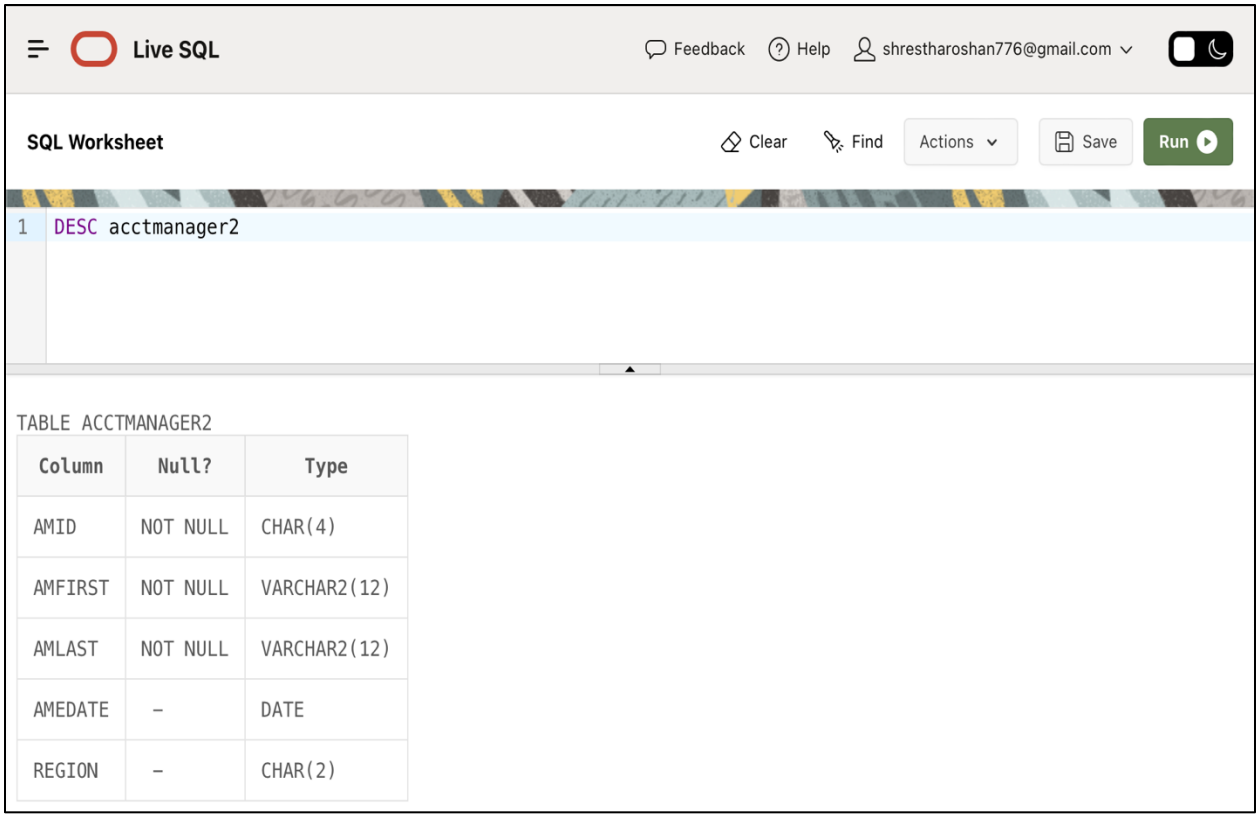



TABLE ACCTMANAGER2

| Column | Null? | Type |
|---|---|---|
| AMID | NOT NULL | CHAR(4) |
| AMFIRST | NOT NULL | VARCHAR2(12) |
| AMLAST | NOT NULL | VARCHAR2(12) |
| AMEDATE | - | DATE |
| REGION | - | CHAR(2) |

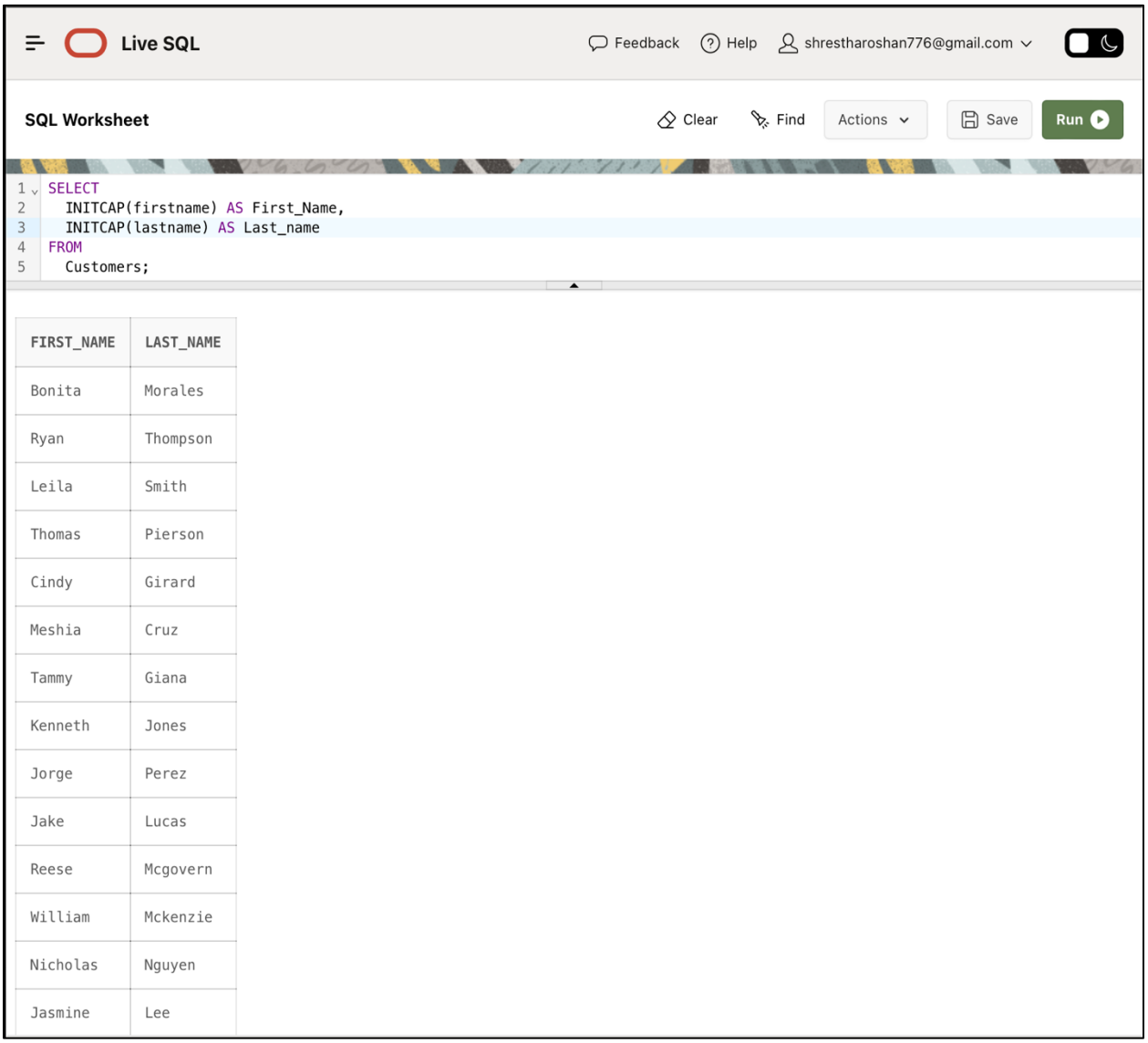
## Activity #1

**1. Produce a list of all customer names in which the first letter of the first and last names are in uppercase, and the rest are in lowercase.**

In order to list down all the customer names in which the first letter if the first and last names are in uppercase, and the rest in lowercase, we can use the query below:

```
SELECT
  INITCAP(firstname),
  INITCAP(lastname)
FROM
  Customers;
```

The **firstname** and **lastname** fields from the **Customers** table are returned by this query. In both the first and last name values, the **INITCAP** function is used to capitalise the first letter of each word. The query will return a list of customer names with suitably capitalised first and last names, ensuring that they appear in a consistent pattern independent of their original casing in the database. The output from the query can be visualized in the screenshot below:



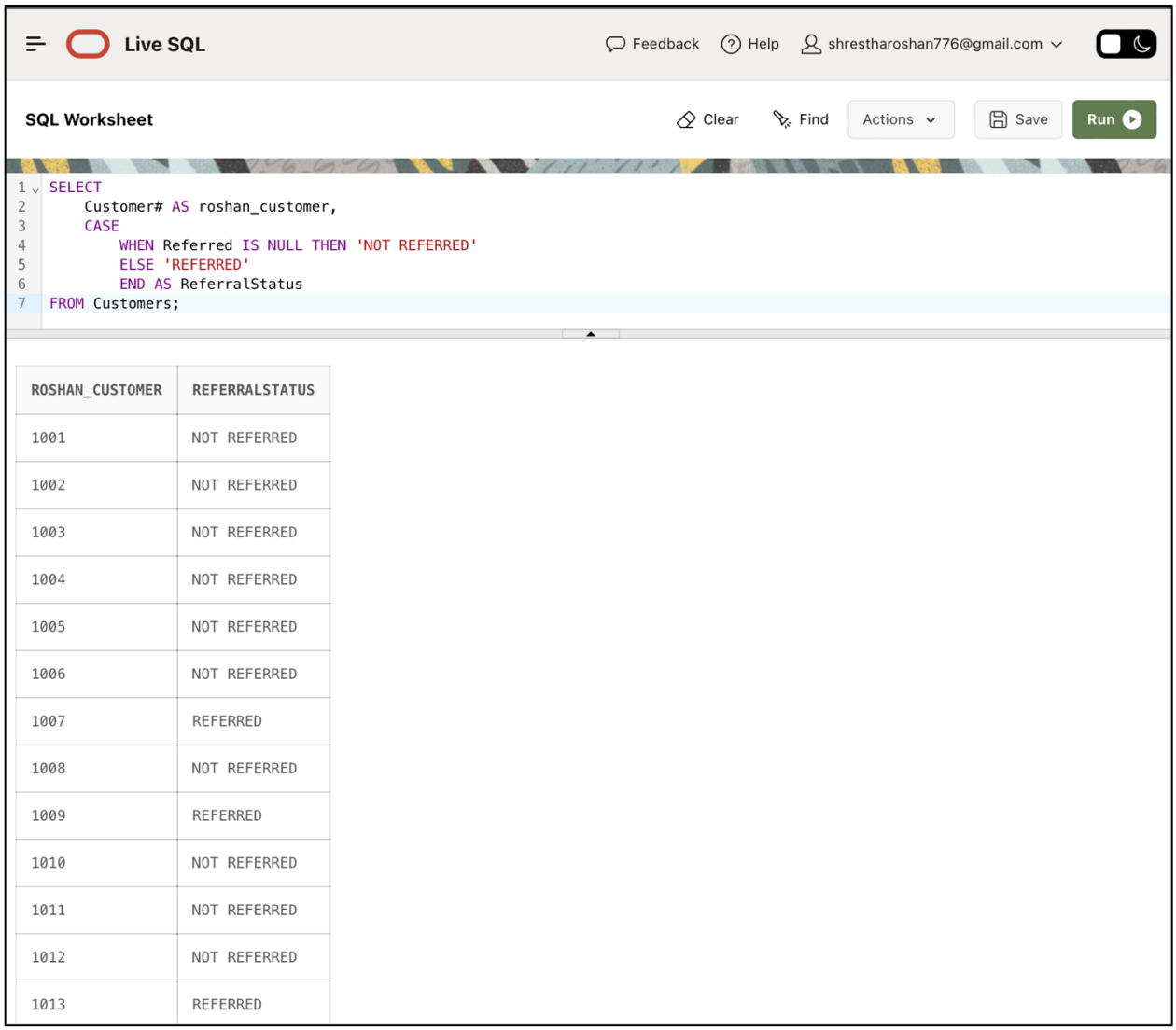

| FIRST_NAME | LAST_NAME |
|---|---|
| Bonita | Morales |
| Ryan | Thompson |
| Leila | Smith |
| Thomas | Pierson |
| Cindy | Girard |
| Meshia | Cruz |
| Tammy | Giana |
| Kenneth | Jones |
| Jorge | Perez |
| Jake | Lucas |
| Reese | Mcgovern |
| William | Mckenzie |
| Nicholas | Nguyen |
| Jasmine | Lee |

**2. Create a list of all customer numbers along with text indicating whether the customer has been referred by another customer. Display the text "NOT REFERRED" if the customer wasn't referred to JustLee Books by another customer or "REFERRED" if the customer was referred.**

To get the desired data we can run the following SQL query:

```
SELECT
Customer# AS roshan_customer,
        CASE
                WHEN Referred IS NULL THEN 'NOT REFERRED'
        ELSE 'REFERRED'
        END AS ReferralStatus
FROM Customers;
```

This query retrieves the Customer column from the Customers table and renames it as **roshan_customer**. It also includes a calculated column called **ReferralStatus** using a **CASE** statement. The **CASE** statement checks if the **Referred** column is **NULL**, and if so, it returns **NOT REFERRED**; otherwise, it returns **REFERRED**. The query will display each customer's number along with their referral status, indicating whether they were referred or not. The output is shown below:




| ROSHAN_CUSTOMER | REFERRALSTATUS |
|---|---|
| 1001 | NOT REFERRED |
| 1002 | NOT REFERRED |
| 1003 | NOT REFERRED |
| 1004 | NOT REFERRED |
| 1005 | NOT REFERRED |
| 1006 | NOT REFERRED |
| 1007 | REFERRED |
| 1008 | NOT REFERRED |
| 1009 | REFERRED |
| 1010 | NOT REFERRED |
| 1011 | NOT REFERRED |
| 1012 | NOT REFERRED |
| 1013 | REFERRED |

## Activity #2

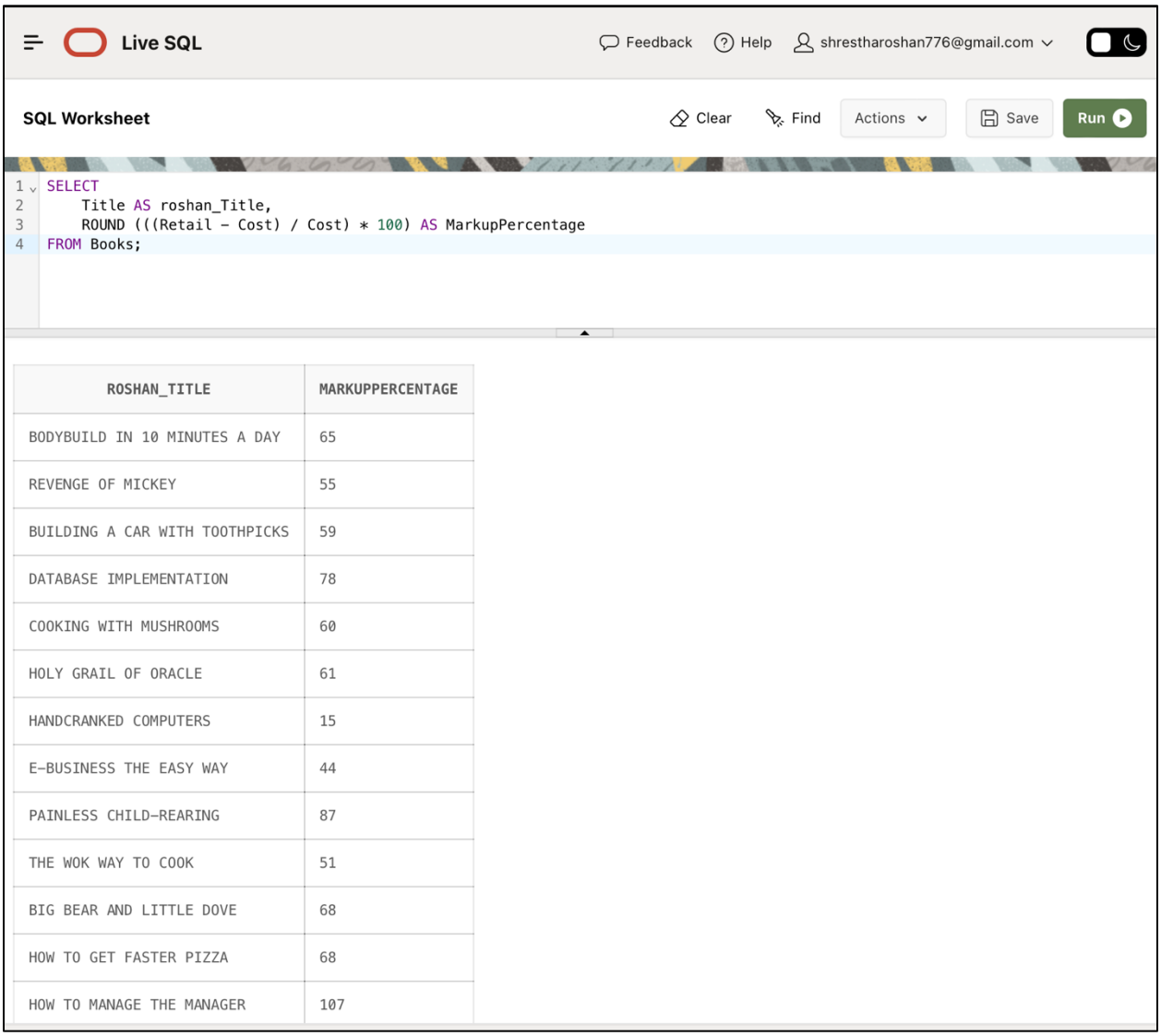
**1. Display a list of all book titles and the percentage of markup for each book. The percentage of markup should be displayed as a whole number (that is, multiplied by 100) with no decimal position, followed by a percent sign (for example, .2793 = 28%). (The percentage of markup should reflect the difference between the retail and cost amounts as a percent of the cost.) of markup should reflect the difference between the retail and cost amounts as a percent of the cost.**

We can use the query as below to retrieve the required data as mentioned in the question:

```
SELECT
        Title AS roshan_Title,
        ROUND (((Retail - Cost) / Cost) * 100) AS MarkupPercentage
FROM Books;
```

This query retrieves the **Title** column from the **Books** table and renames it as "roshan_Title." It also calculates the "MarkupPercentage" column using the ROUND function. The formula subtracts the "Cost" from the "Retail" and then divides the result by the "Cost" to get the markup percentage. The query will display each book title along with its calculated markup percentage.

The output from the above query is shown in screenshot attached below:



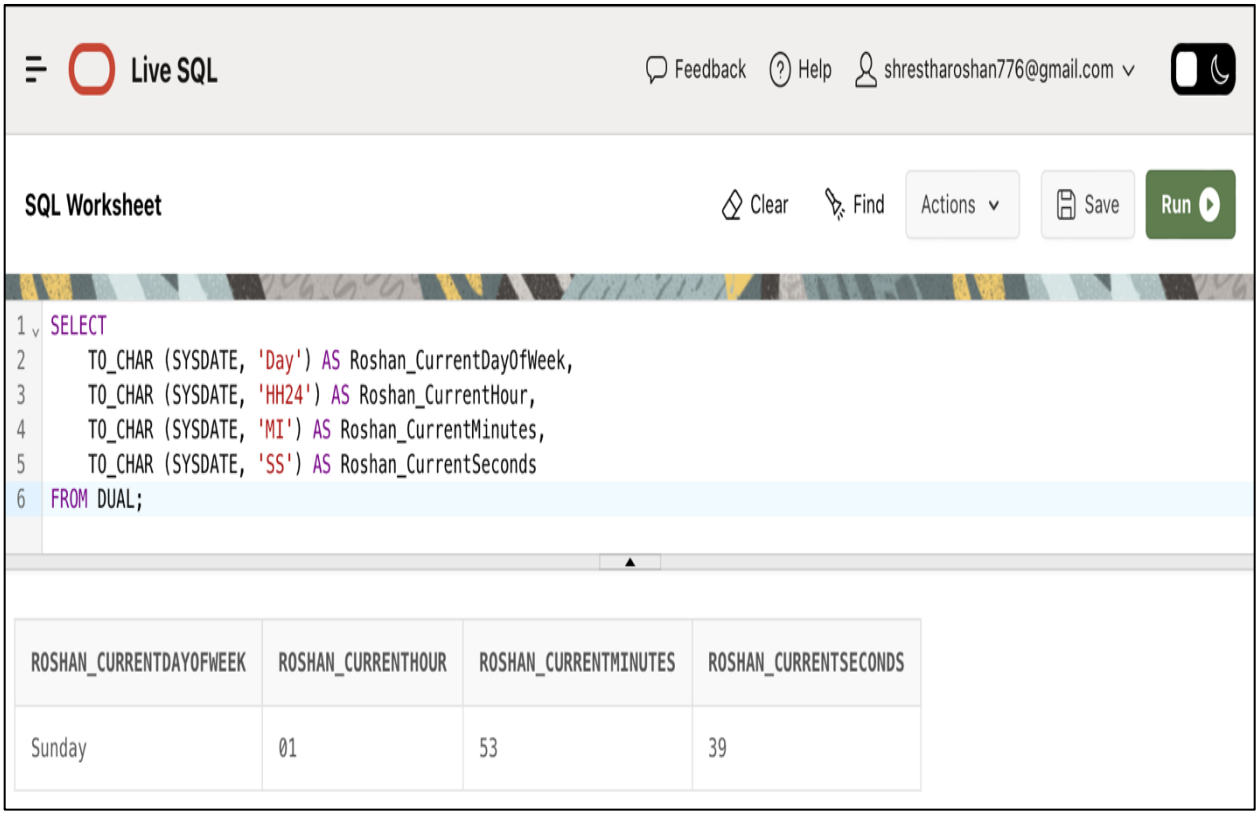

| ROSHAN_TITLE | MARKUPPERCENTAGE |
|---|---|
| BODYBUILD IN 10 MINUTES A DAY | 65 |
| REVENGE OF MICKEY | 55 |
| BUILDING A CAR WITH TOOTHPICKS | 59 |
| DATABASE IMPLEMENTATION | 78 |
| COOKING WITH MUSHROOMS | 60 |
| HOLY GRAIL OF ORACLE | 61 |
| HANDCRANKED COMPUTERS | 15 |
| E-BUSINESS THE EASY WAY | 44 |
| PAINLESS CHILD-REARING | 87 |
| THE WOK WAY TO COOK | 51 |
| BIG BEAR AND LITTLE DOVE | 68 |
| HOW TO GET FASTER PIZZA | 68 |
| HOW TO MANAGE THE MANAGER | 107 |

**2. Display the current day of the week, hour, minutes, and seconds of the current date setting on the computer you're using.**

To filter the data based on current day of the week, hour, minutes, and seconds of the current date setting on our system, we can use the query below:

```
SELECT
        TO_CHAR (SYSDATE, 'Day') AS Roshan_CurrentDayOfWeek,
        TO_CHAR (SYSDATE, 'HH24') AS Roshan_CurrentHour,
        TO_CHAR (SYSDATE, 'MI') AS Roshan_CurrentMinutes,
        TO_CHAR (SYSDATE, 'SS') AS Roshan_CurrentSeconds
FROM DUAL;
```

This query retrieves the current day of the week, hour, minutes, and seconds from the system date and time using the **TO_CHAR** function. It uses specific date format codes (**Day** for day of the week, **HH24** for 24-hour format hour, **MI** for minutes, and **SS** for seconds) to format the output. The query will display the current day of the week, hour, minutes, and seconds as separate columns from the **DUAL** table. The output from the above query is attached below:



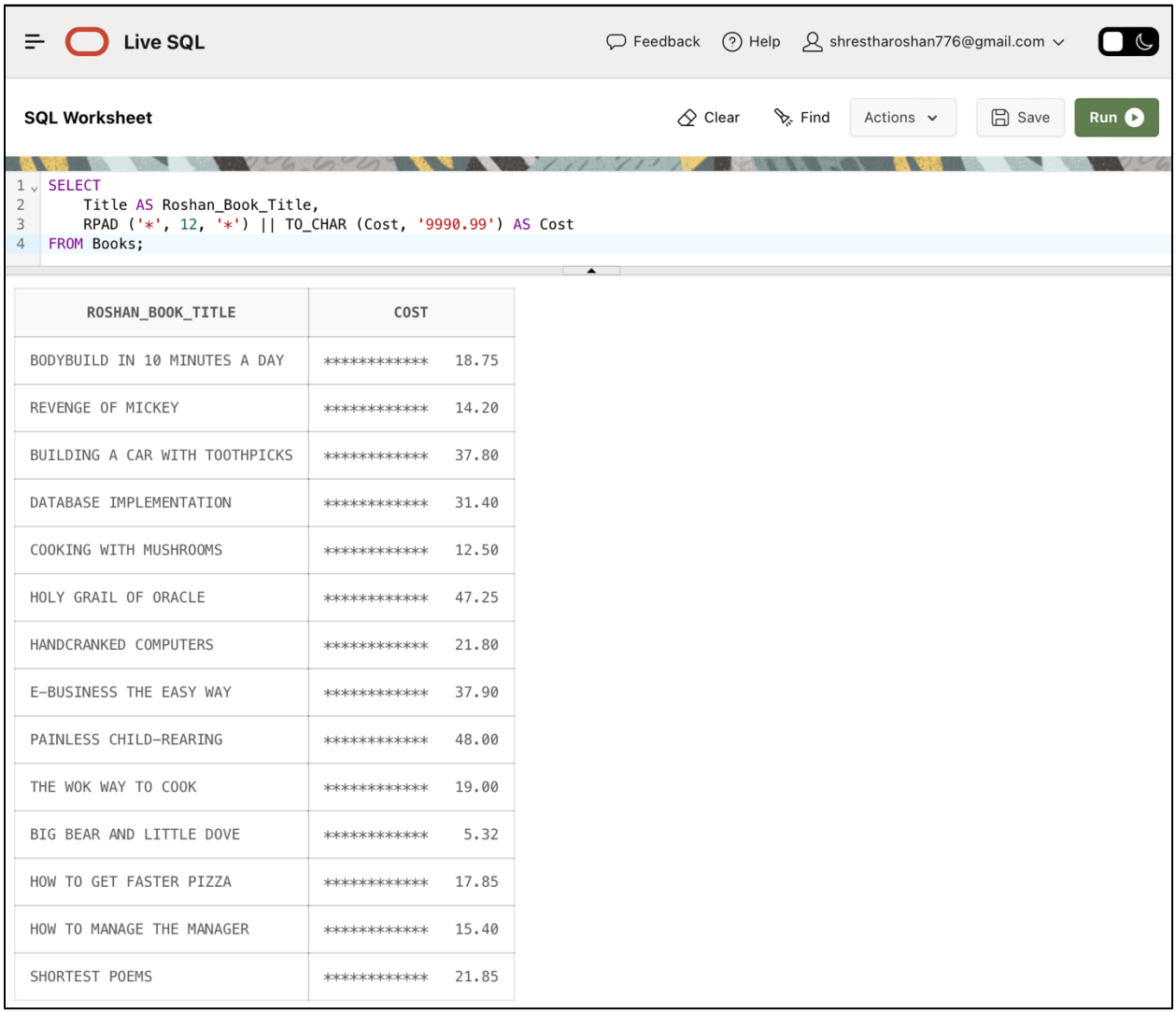
## Activity #3

**1. Create a list of all book titles and costs. Precede each book's cost with asterisks so that the width of the displayed Cost field is 12.**

To retrieve the above-mentioned data from the database we can use the query below:

```
SELECT
        Title AS Roshan_Book_Title,
        RPAD ('*', 12, '*') || TO_CHAR (Cost, '9990.99') AS Cost
FROM Books;
```

This query retrieves the **Title** and **Cost** columns from the **Books** table. It renames the **Title** column as **Roshan_Book_Title**. For the **Cost** column, it first pads the value with asterisks on the left side to make the total length 12 characters, and then converts the **Cost** value to a formatted string with two decimal places. The query will display each book title along with the **Cost** value, where the **Cost** is preceded by asterisks to create a total length of 12 characters. The result is shown below:



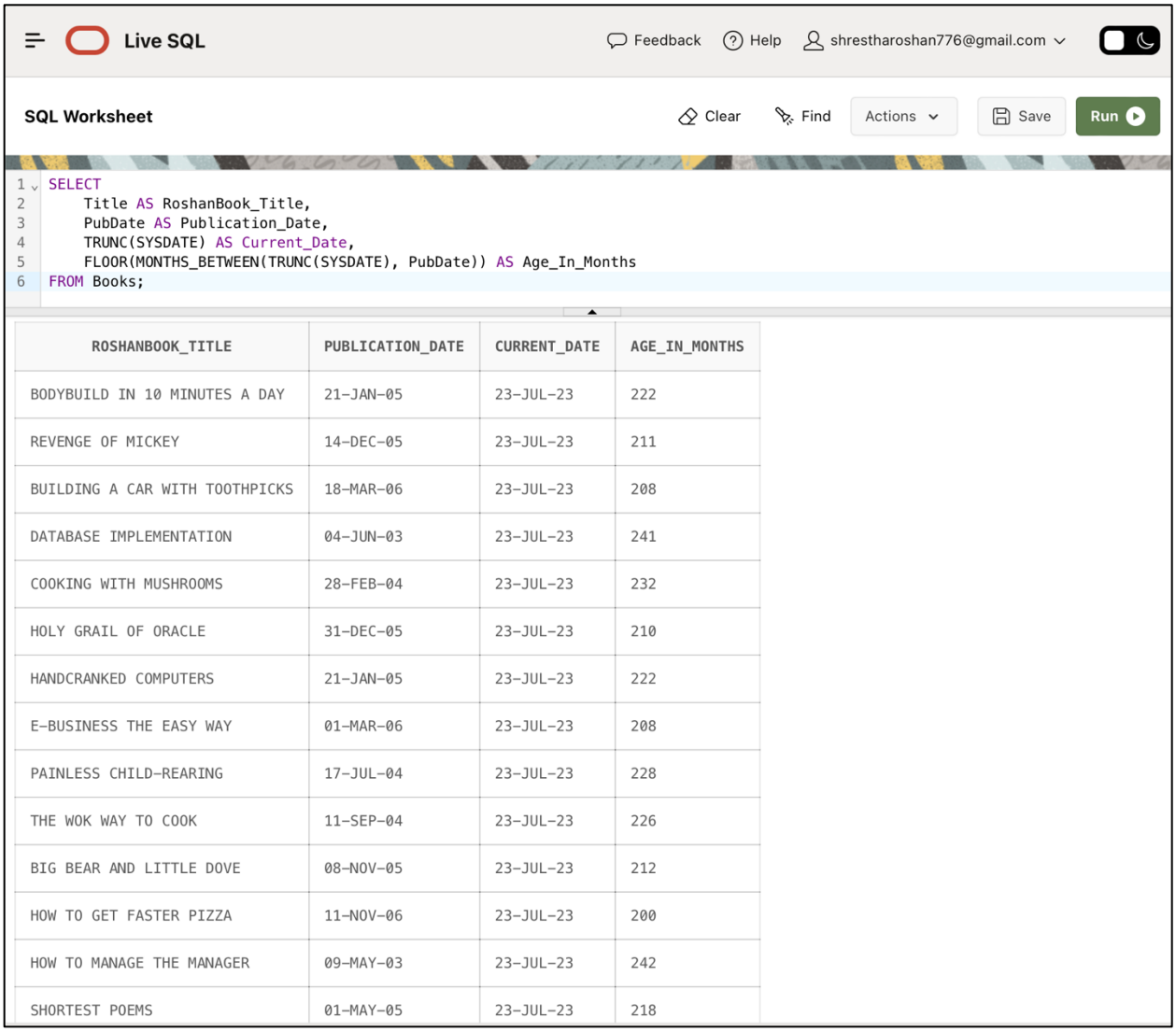

| ROSHAN_BOOK_TITLE | COST |
|---|---|
| BODYBUILD IN 10 MINUTES A DAY | ********** 18.75 |
| REVENGE OF MICKEY | ********** 14.20 |
| BUILDING A CAR WITH TOOTHPICKS | ********** 37.80 |
| DATABASE IMPLEMENTATION | ********** 31.40 |
| COOKING WITH MUSHROOMS | ********** 12.50 |
| HOLY GRAIL OF ORACLE | ********** 47.25 |
| HANDCRANKED COMPUTERS | ********** 21.80 |
| E-BUSINESS THE EASY WAY | ********** 37.90 |
| PAINLESS CHILD-REARING | ********** 48.00 |
| THE WOK WAY TO COOK | ********** 19.00 |
| BIG BEAR AND LITTLE DOVE | ********** 5.32 |
| HOW TO GET FASTER PIZZA | ********** 17.85 |
| HOW TO MANAGE THE MANAGER | ********** 15.40 |
| SHORTEST POEMS | ********** 21.85 |

**2. Using today's date, determine the age (in months) of each book that JustLee sells. Make sure only whole months are displayed; ignore any portions of months. Display the book title, publication date, current date, and age.**

To perform the above mentioned action, we can use the query below:

```
SELECT
        Title AS RoshanBook_Title, PubDate AS Publication_Date,
        TRUNC(SYSDATE) AS Current_Date,
        FLOOR(MONTHS_BETWEEN(TRUNC(SYSDATE), PubDate)) AS
        Age_In_Months
FROM Books;
```

This query retrieves the **Title** and **PubDate** columns from the **Books** table and renames them as **RoshanBook_Title** and **Publication_Date**, respectively. It also includes two calculated columns: **Current_Date**, which is the current date truncated to remove the time part, and **Age_In_Months**, which calculates the number of whole months between the current date and the publication date of each book using the **MONTHS_BETWEEN** function and then rounds down to the nearest integer using the **FLOOR** function. The query will display each book's title, publication date, current date, and age in months since publication.




| ROSHANBOOK_TITLE | PUBLICATION_DATE | CURRENT_DATE | AGE_IN_MONTHS |
|---|---|---|---|
| BODYBUILD IN 10 MINUTES A DAY | 21-JAN-05 | 23-JUL-23 | 222 |
| REVENGE OF MICKEY | 14-DEC-05 | 23-JUL-23 | 211 |
| BUILDING A CAR WITH TOOTHPICKS | 18-MAR-06 | 23-JUL-23 | 208 |
| DATABASE IMPLEMENTATION | 04-JUN-03 | 23-JUL-23 | 241 |
| COOKING WITH MUSHROOMS | 28-FEB-04 | 23-JUL-23 | 232 |
| HOLY GRAIL OF ORACLE | 31-DEC-05 | 23-JUL-23 | 210 |
| HANDCRANKED COMPUTERS | 21-JAN-05 | 23-JUL-23 | 222 |
| E-BUSINESS THE EASY WAY | 01-MAR-06 | 23-JUL-23 | 208 |
| PAINLESS CHILD-REARING | 17-JUL-04 | 23-JUL-23 | 228 |
| THE WOK WAY TO COOK | 11-SEP-04 | 23-JUL-23 | 226 |
| BIG BEAR AND LITTLE DOVE | 08-NOV-05 | 23-JUL-23 | 212 |
| HOW TO GET FASTER PIZZA | 11-NOV-06 | 23-JUL-23 | 200 |
| HOW TO MANAGE THE MANAGER | 09-MAY-03 | 23-JUL-23 | 242 |
| SHORTEST POEMS | 01-MAY-05 | 23-JUL-23 | 218 |

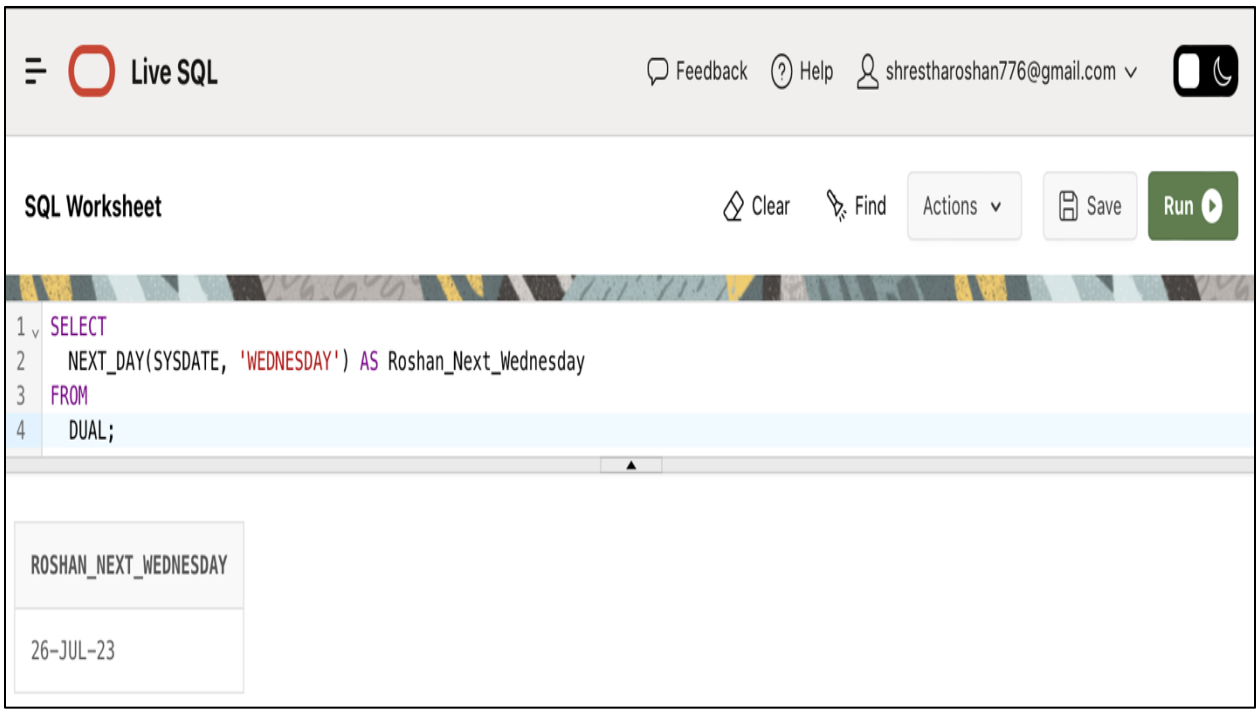
## Activity #4

**1. Determine the calendar date of the next occurrence of Wednesday, based on today's date.**

We can use the following query to determine the calendar date of the next occurrence of Wednesday, based on today's date:

```
SELECT
        NEXT_DAY(SYSDATE, 'WEDNESDAY') AS Roshan_Next_Wednesday
FROM DUAL;
```

This query uses the **NEXT_DAY** function to calculate the date of the next Wednesday from the current date (**SYSDATE**). The result will be displayed as **Roshan_Next_Wednesday**. The query will return the date of the upcoming Wednesday from the current date, or if the current date is a Wednesday, it will return the same date.



**2. Produce a list of each customer number and the third and fourth digits of his or her zip code. The query should also display the position of the first occurrence of a 3 in the customer number, if it exists.**

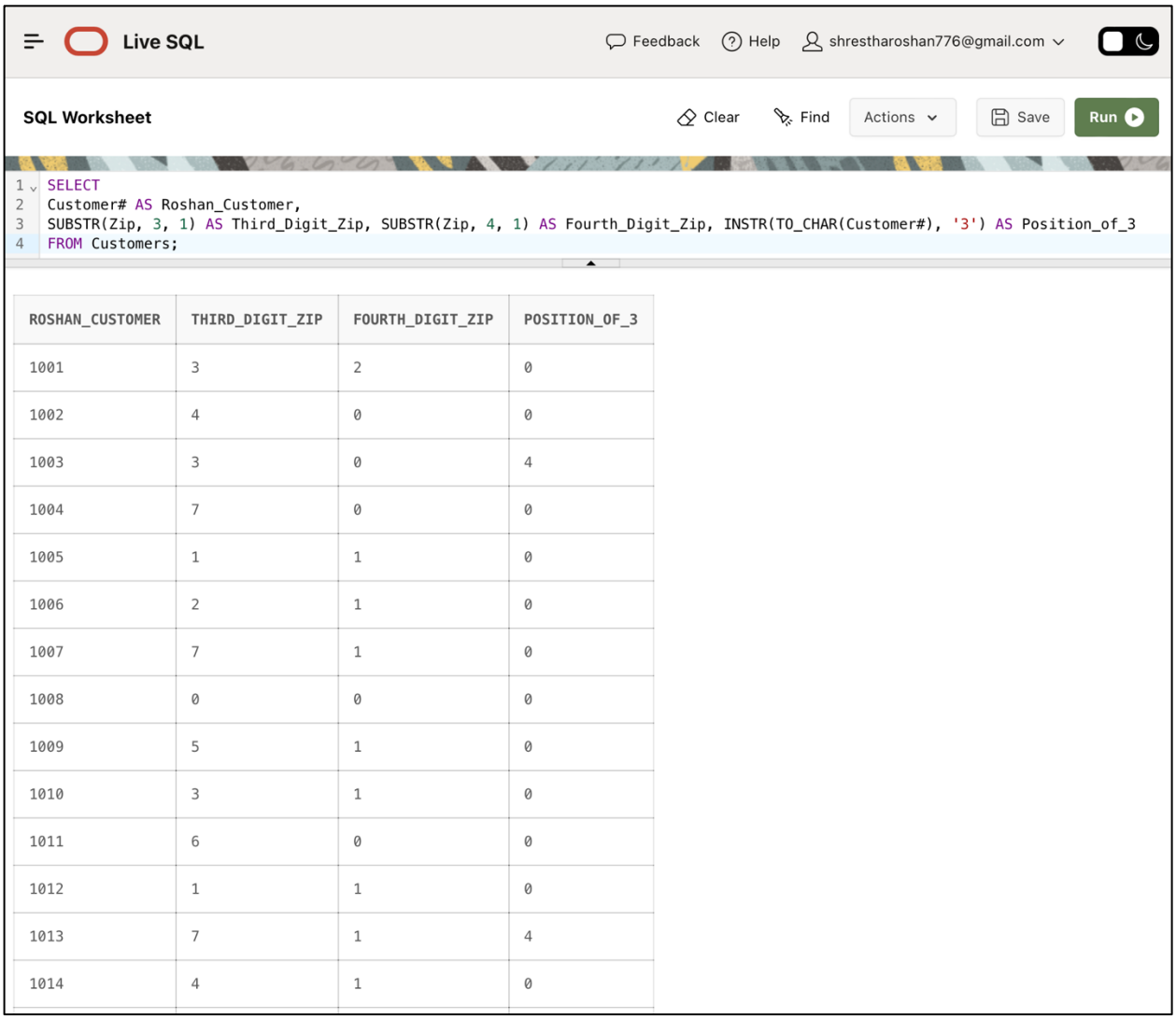
We can use the query below to fetch the required data as mentioned in question:

```
SELECT
        Customer# AS Roshan_Customer,
        SUBSTR(Zip, 3, 1) AS Third_Digit_Zip, SUBSTR(Zip, 4, 1) AS Fourth_Digit_Zip,
INSTR(TO_CHAR(Customer#), '3') AS Position_of_3
FROM Customers;
```

This query retrieves the **Customer** and **Zip** columns from the **Customers** table, renaming **Customer** as **Roshan_Customer**. It also includes three calculated columns:

1. **Third_Digit_Zip** extracts the third digit from the **Zip** value.
2. **Fourth_Digit_Zip** extracts the fourth digit from the **Zip** value.
3. **Position_of_3** determines the position of the first occurrence of the digit '3' in the **Customer** value, if it exists.

The query will display each customer's number, the third and fourth digits of their zip code, and the position of the first '3' digit in their customer number (if applicable).




| ROSHAN_CUSTOMER | THIRD_DIGIT_ZIP | FOURTH_DIGIT_ZIP | POSITION_OF_3 |
|---|---|---|---|
| 1001 | 3 | 2 | 0 |
| 1002 | 4 | 0 | 0 |
| 1003 | 3 | 0 | 4 |
| 1004 | 7 | 0 | 0 |
| 1005 | 1 | 1 | 0 |
| 1006 | 2 | 1 | 0 |
| 1007 | 7 | 1 | 0 |
| 1008 | 0 | 0 | 0 |
| 1009 | 5 | 1 | 0 |
| 1010 | 3 | 1 | 0 |
| 1011 | 6 | 0 | 0 |
| 1012 | 1 | 1 | 0 |
| 1013 | 7 | 1 | 4 |
| 1014 | 4 | 1 | 0 |