

# **DATABASE MANAGEMENT SYSTEM**

**(01CT0407)**

**Department of Information Communication and  
Technology  
4<sup>rd</sup> Semester**

**Lab Manual**

**(Dec-Apr 23-24)**

## Index

Lab	Programs
1	Introduction to RDBMS and APEX Login
2	DDL Commands and Table Creation
3	DML Commands and Queries
4	DML Commands and Related Queries
5	Constraint-Based DML Commands
6	Functions and Queries
7	Implement operator, null and special operator-based queries
8	Implement Constraint-based and Group by related queries
9	Range Function and Operator based Queries
10	Join based Queries having Functions
11	Understand and implement Conditions and looping in PL/SQL
12	Implement a PL/SQL Block
13	Implement a Procedure and Function for given Statement
14	Understand and Implement Triggers

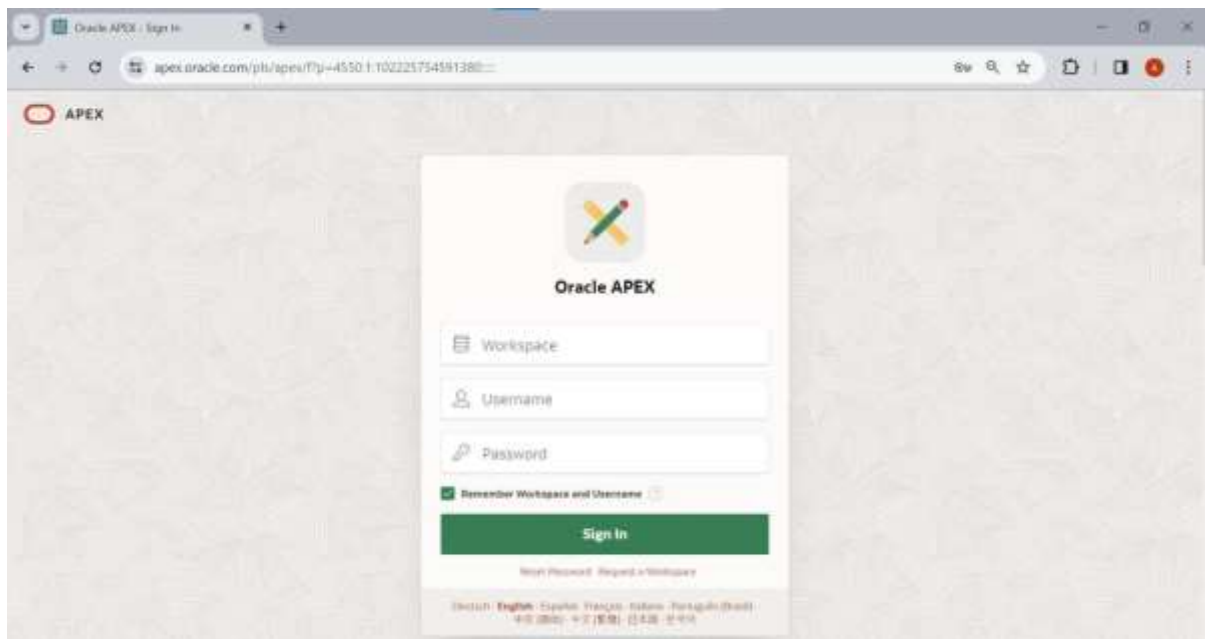
## Practical 1

**Aim: Introduction to RDBMS and APEX Login.**

### APEX login:

Step 1: <https://apex.oracle.com/pls/apex/>

Step 2: Click on Request a workspace.

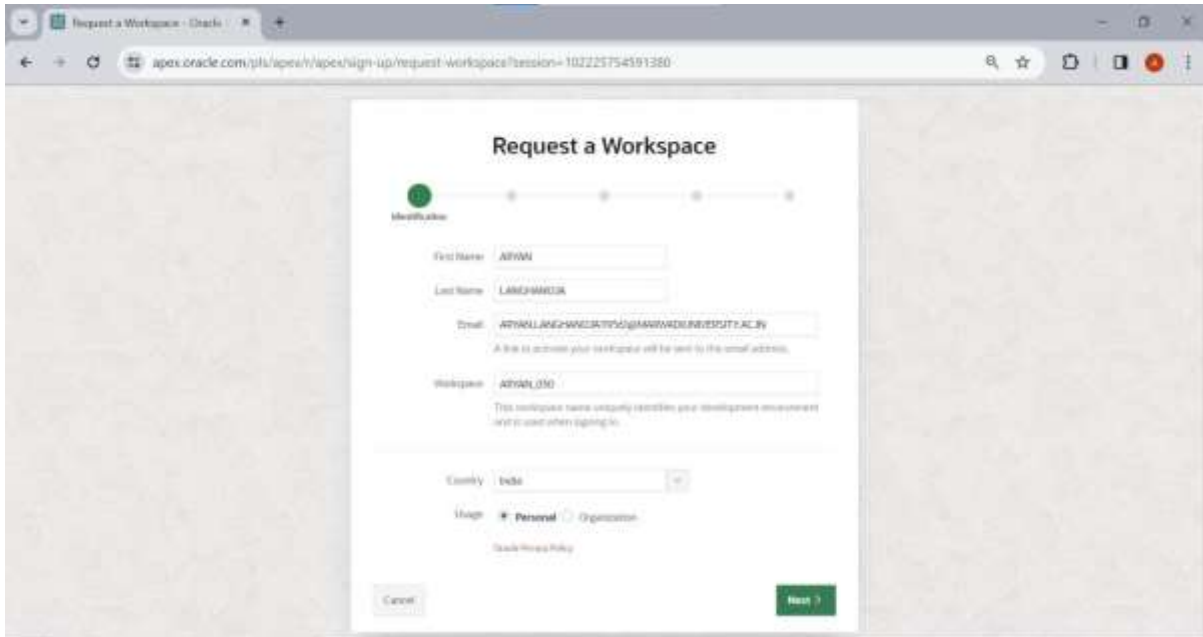


Step 3: Fill all the required details carefully like: First Name: Your name(ex. Harikesh)

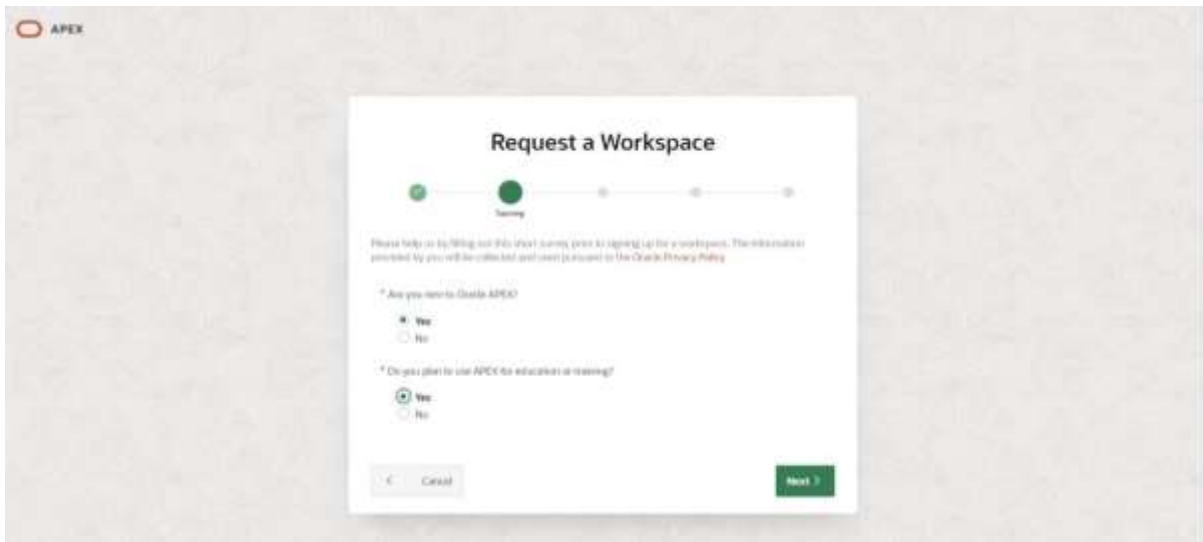
Last Name: Your last name (ex. Chauhan)

Email: Use your institute email address only.


Workspace: name\_surname And click on next.



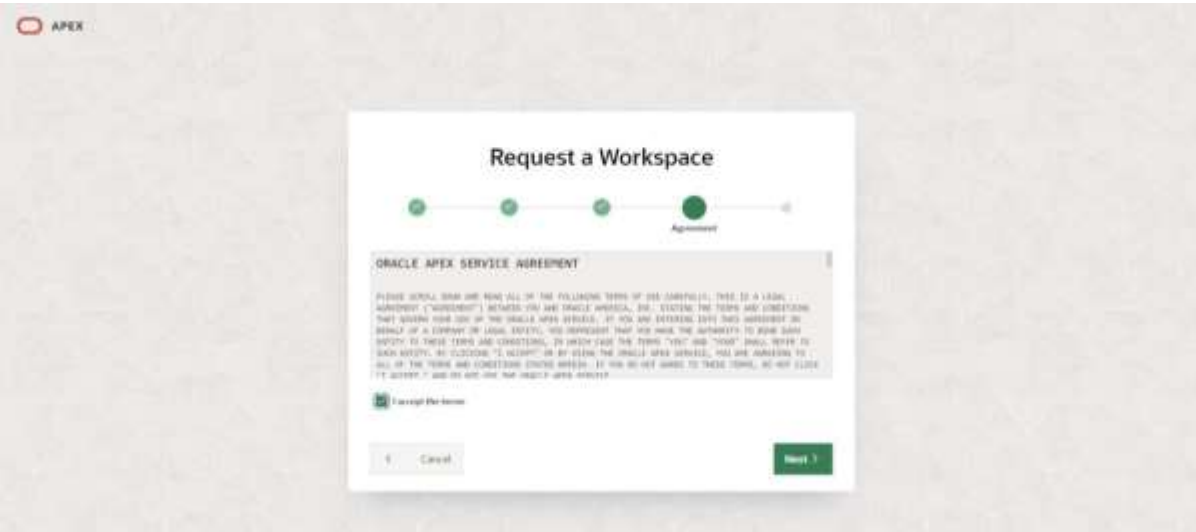
Step 4: Fill survey detail same as below and click on next.



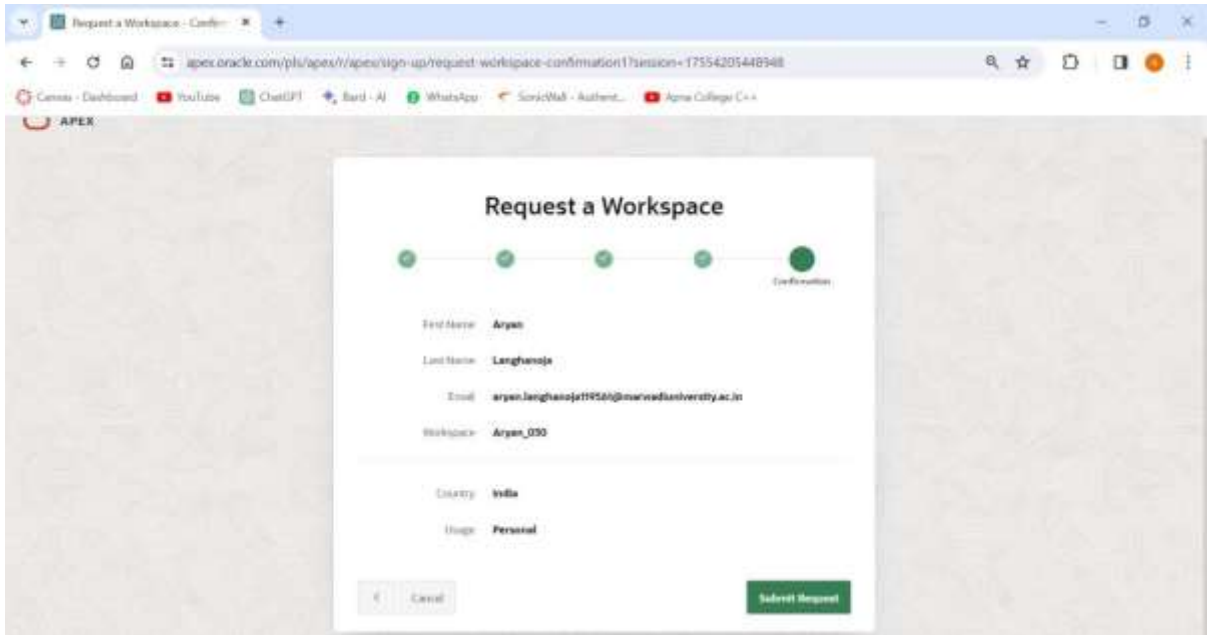
Step 5: Write appropriate text in textarea and Click on next,



Step 6: Click on check box to agree terms and conditions and Click on next.



Step 7: Click on submit request button and Click on next

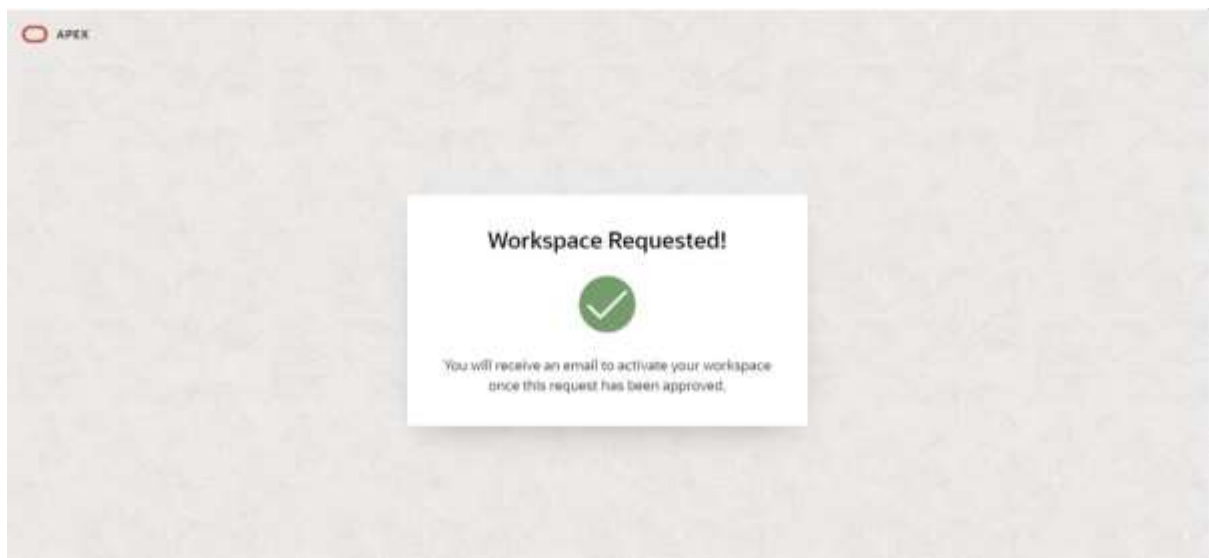


The screenshot shows a web browser window with the URL `apex.oracle.com/pls/apex/f/apex/sign-up/request-workspace-confirmation1?session=17554205448548`. The page title is "Request a Workspace - Confirm". The form is titled "Request a Workspace" and features a progress bar with five steps, the last of which is "Confirmation". The form fields are as follows:

First Name	Argan
Last Name	Langhanoja
Email	argan.langhanoja1956@marwadiversity.ac.in
Workspace	Argan_030
Country	India
Usage	Personal

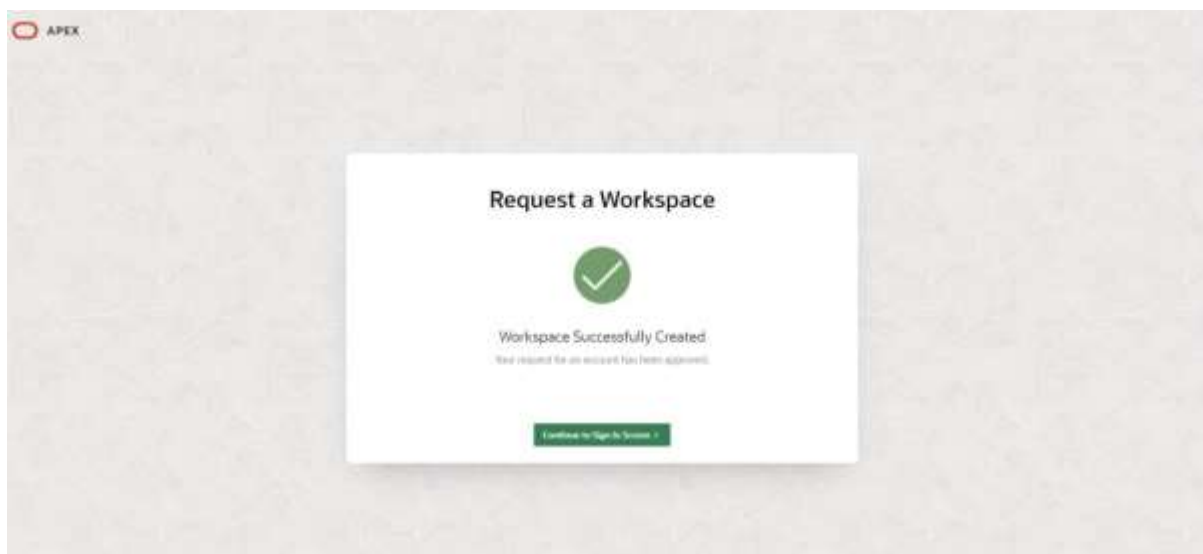
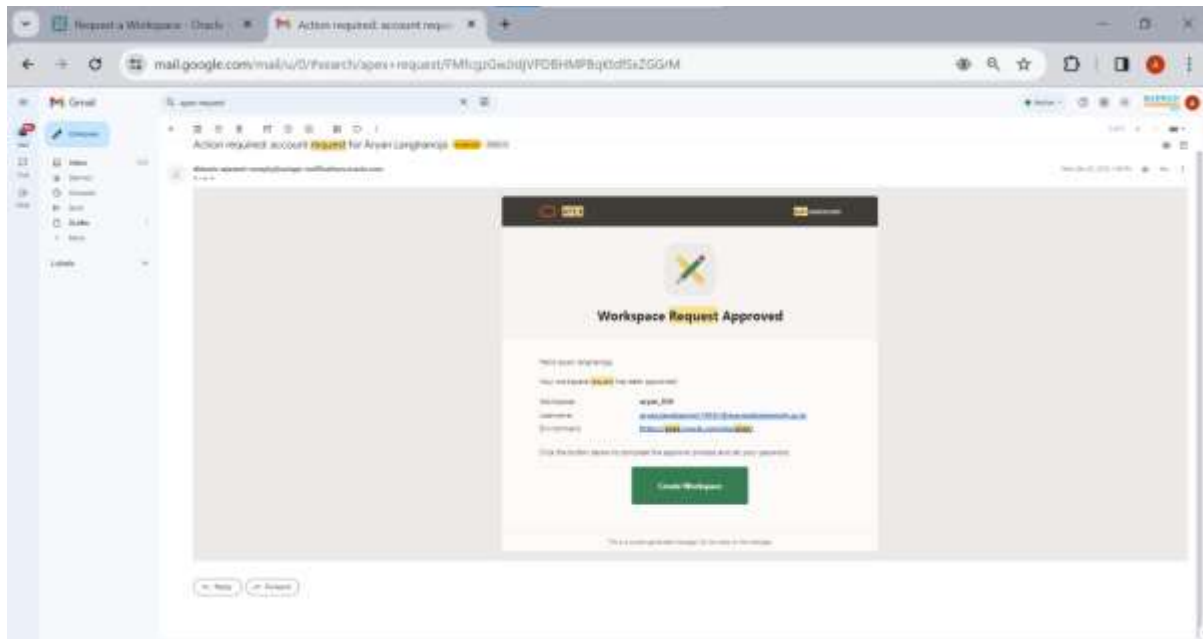
At the bottom of the form, there are two buttons: "Cancel" and "Submit Request".

Step 8: After that you will receive an email on your institute email id.

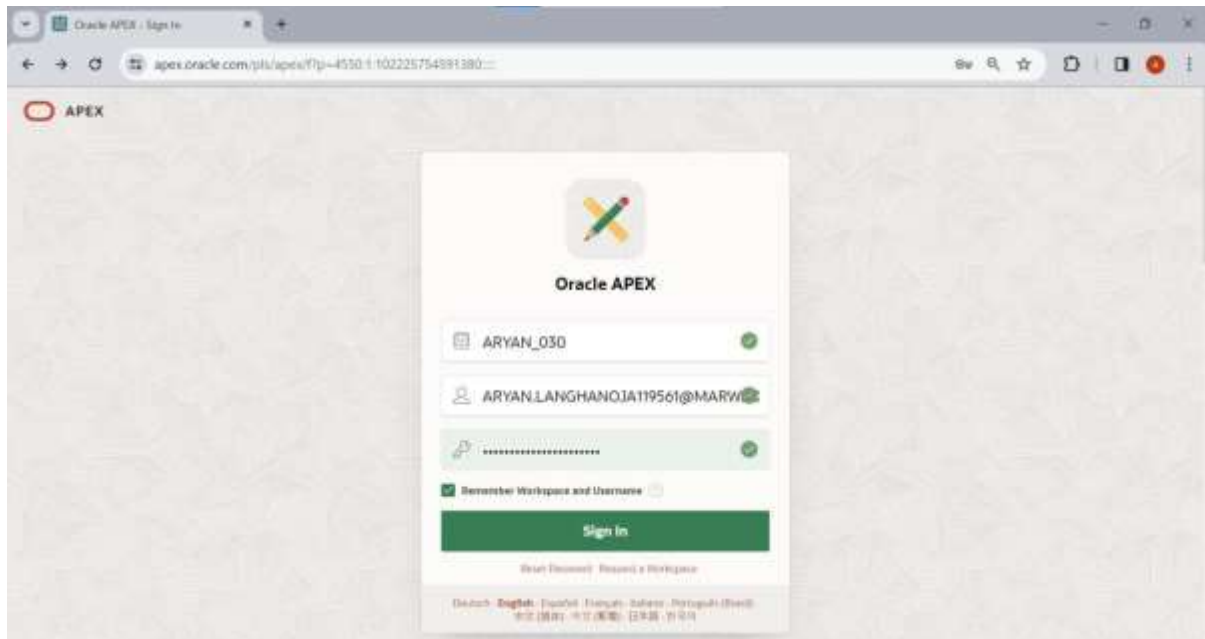


Step 9: Click on create workspace.

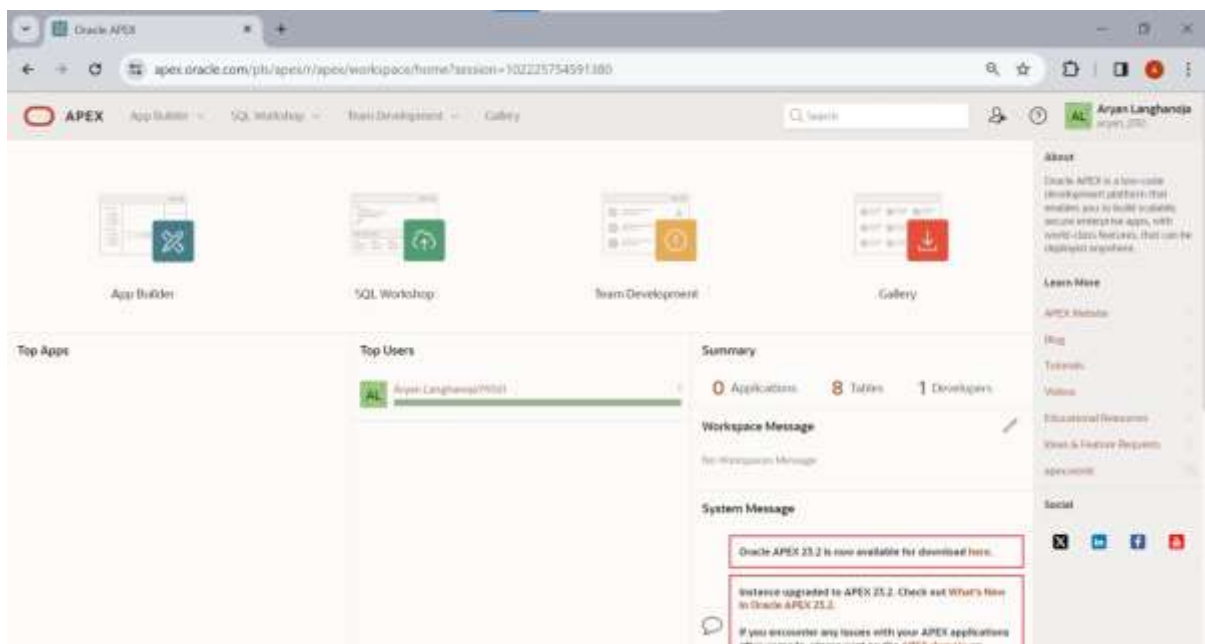
Step 10: Click on Continue to sign in screen.



Step 11: Set your new password and confirm password and click on change password.

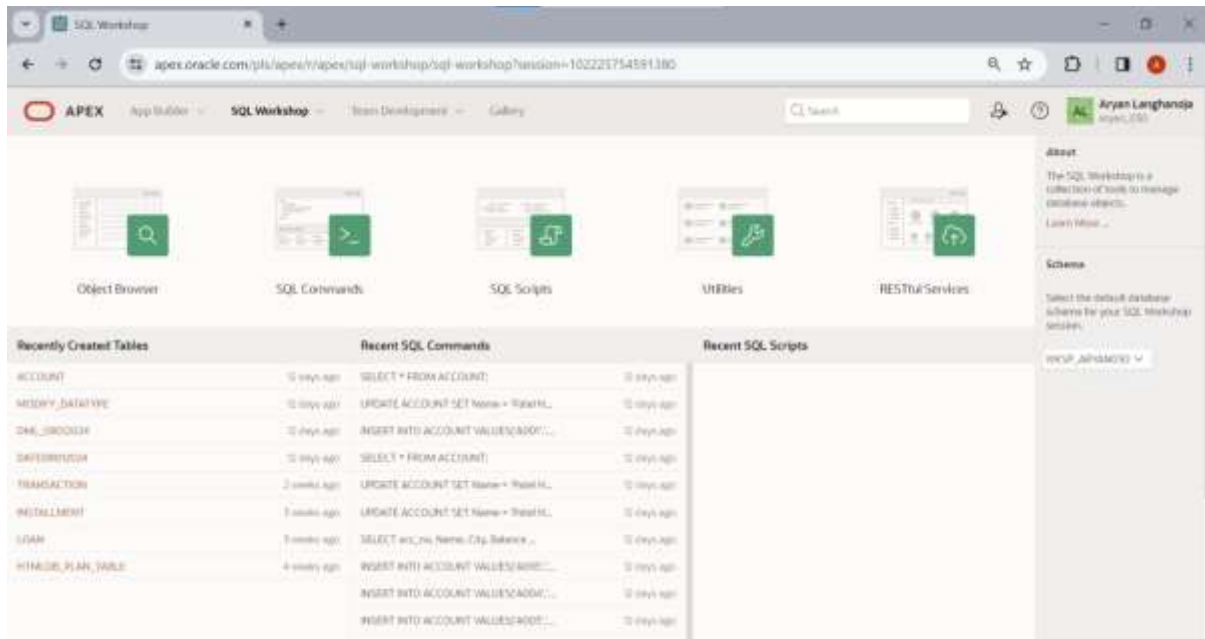


Step 12: Click on SQL workshop.

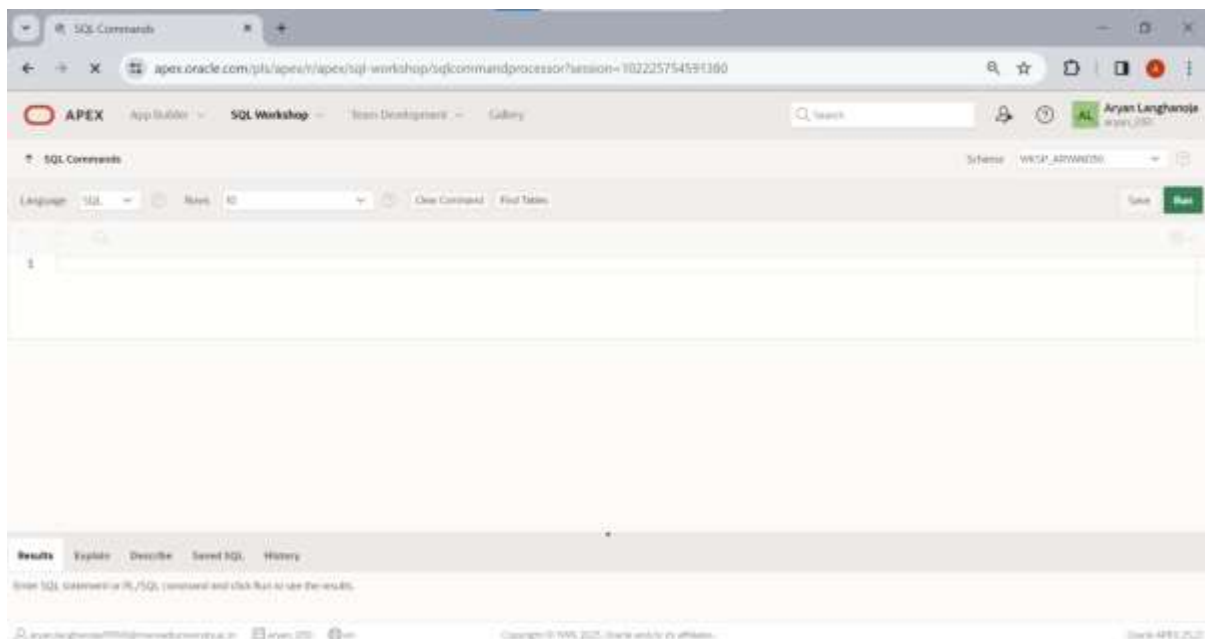


Step 13: Click on SQL Commands.





Step 14: Now you can perform your queries.

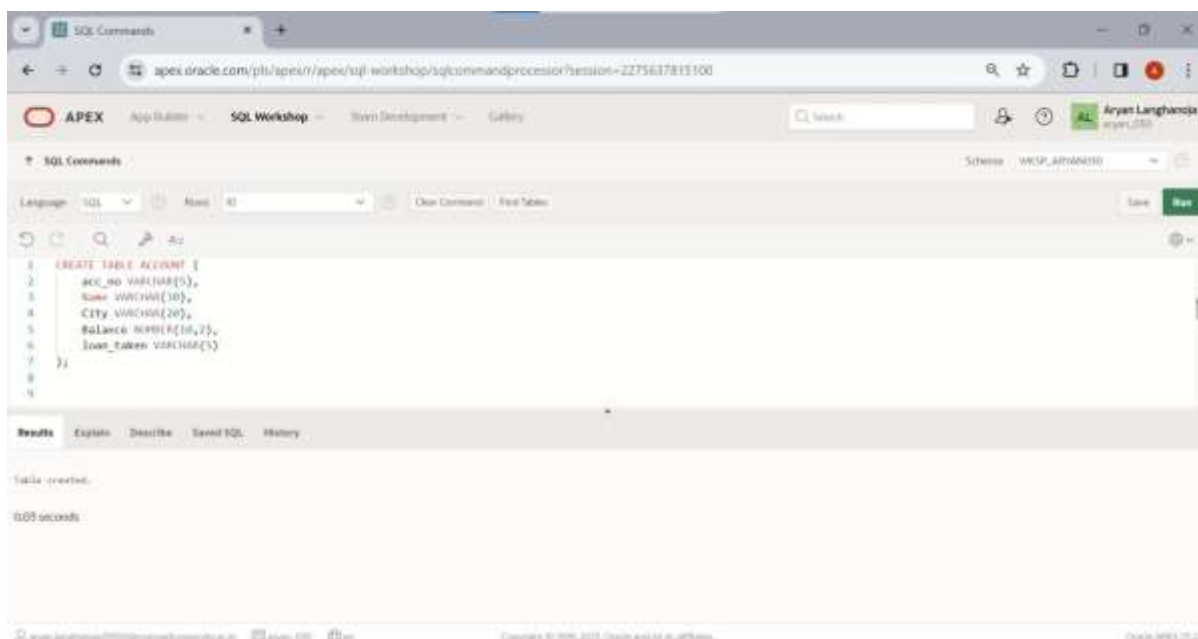


## Practical 2

### Aim: DDL Commands and Table Creation.

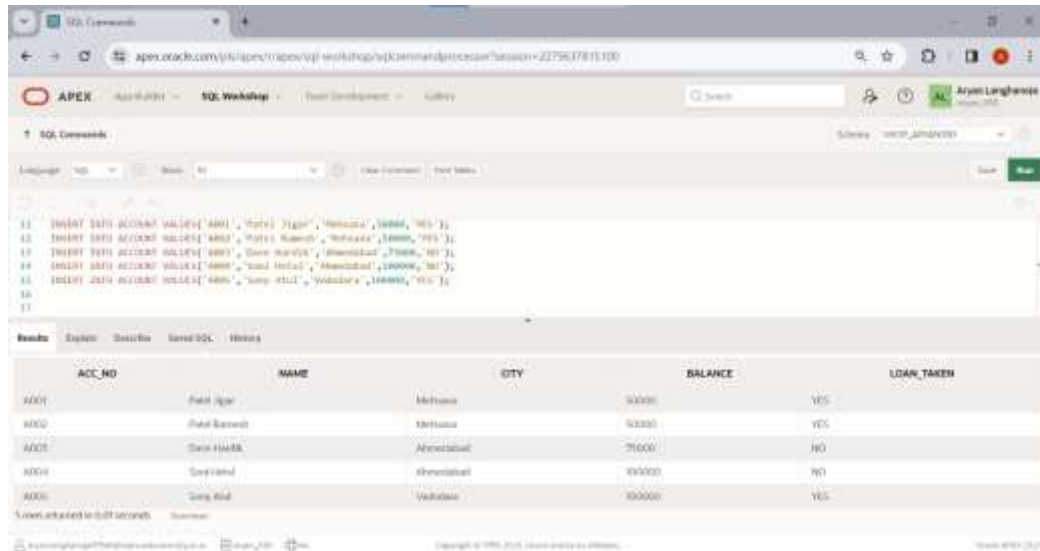
#### Create a table ACCOUNT

Column name	Data Type	Size
acc_no	varchar2	5
Name	varchar2	30
City	varchar2	20
Balance	Number	10,2
loan_taken	varchar2	5



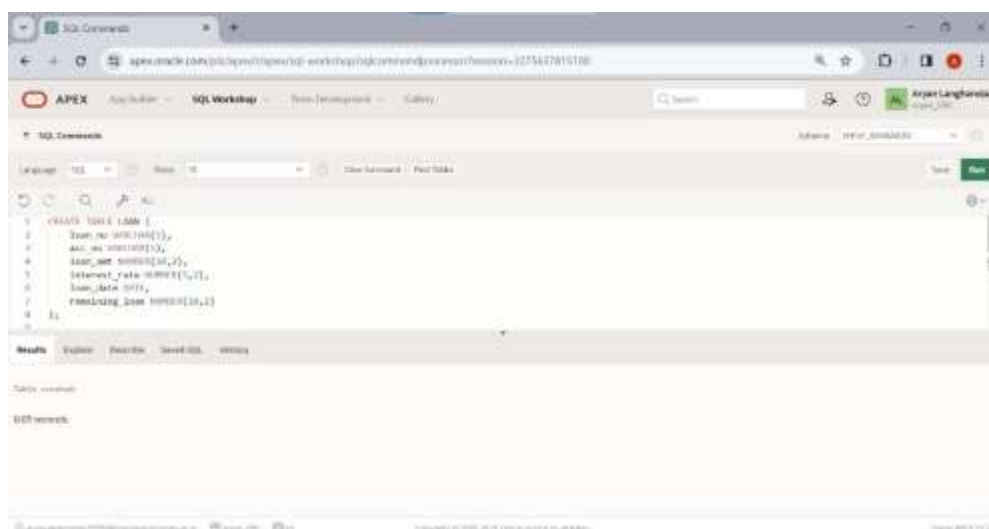
#### Insert the following records.

acc_no	Name	City	Balance	loan_taken
A001	Patel Jigar	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	YES
A003	Dave Hardik	Ahmedabad	75000	NO
A004	Soni Hetal	Ahmedabad	100000	NO
A005	Sony Atul	Vadodara	100000	YES



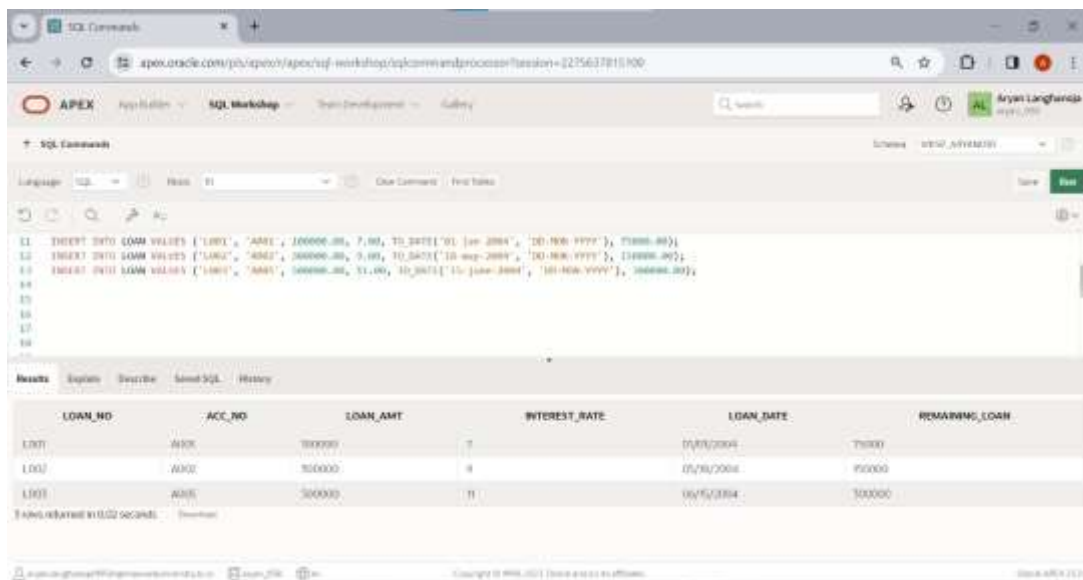
### Create a Table LOAN

Column Name	Data Type	Size
loan_no	varchar2	5
acc_no	varchar2	5
loan_amt	number	10,2
interest_rate	number	5,2
loan_date	date	
remaining_loan	number	10,2



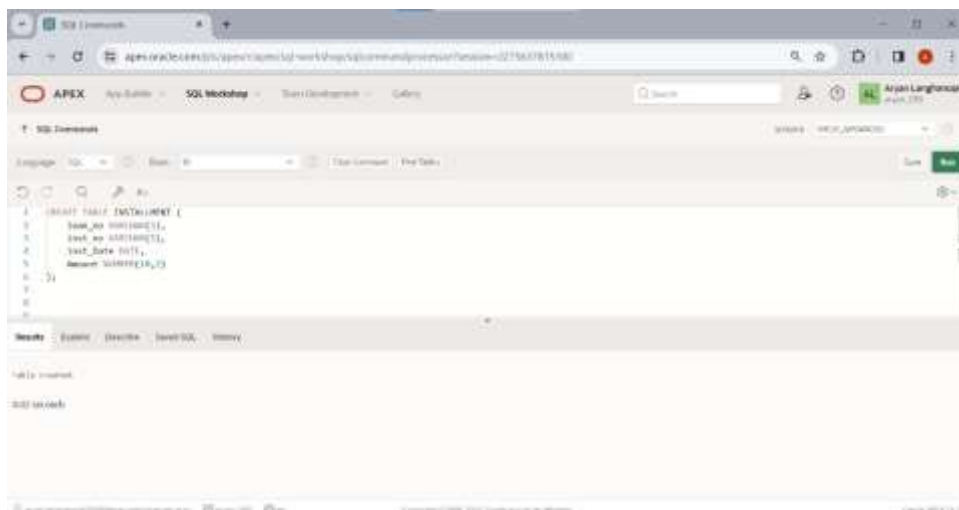
**Insert the following Records.**

Loan_no	Acc_no	Loan_amt	Interest_rate	Loan_date	Remaining_loan
L001	A001	100000	7	1-jan-04	75000
L002	A002	300000	9	18-may-04	150000
L003	A005	500000	11	15-june-04	300000



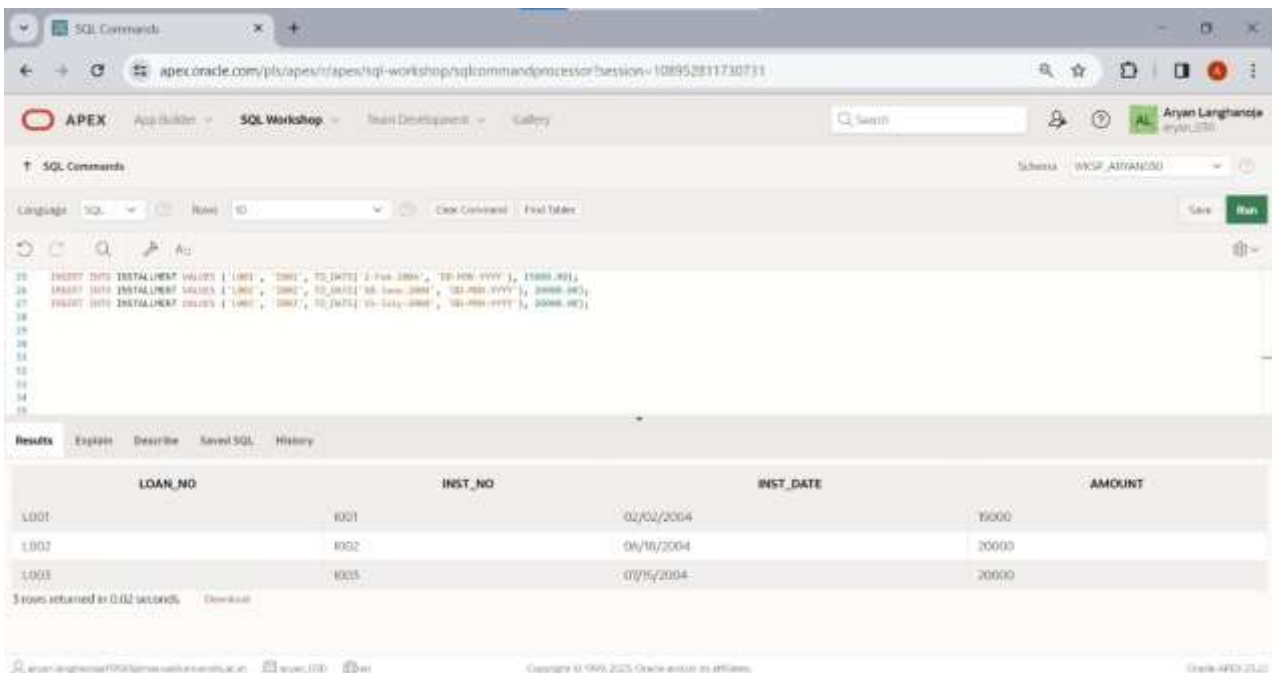
**Create a table INSTALLMENT**

Column Name	Data Type	Size
loan_no	varchar2	5
inst_no	varchar2	5
inst_Date	Date	
Amount	Number	10,2



### Insert following Records

Loan_no	Inst_no	Date	Amount
L001	I001	2-Feb-04	15000
L002	I002	18-June-04	20000
L003	I003	15-July-04	20000



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL Commands window contains three INSERT statements for the INSTALLMENT table. The Results window displays the output of these statements, showing three rows of data inserted into the table.

```

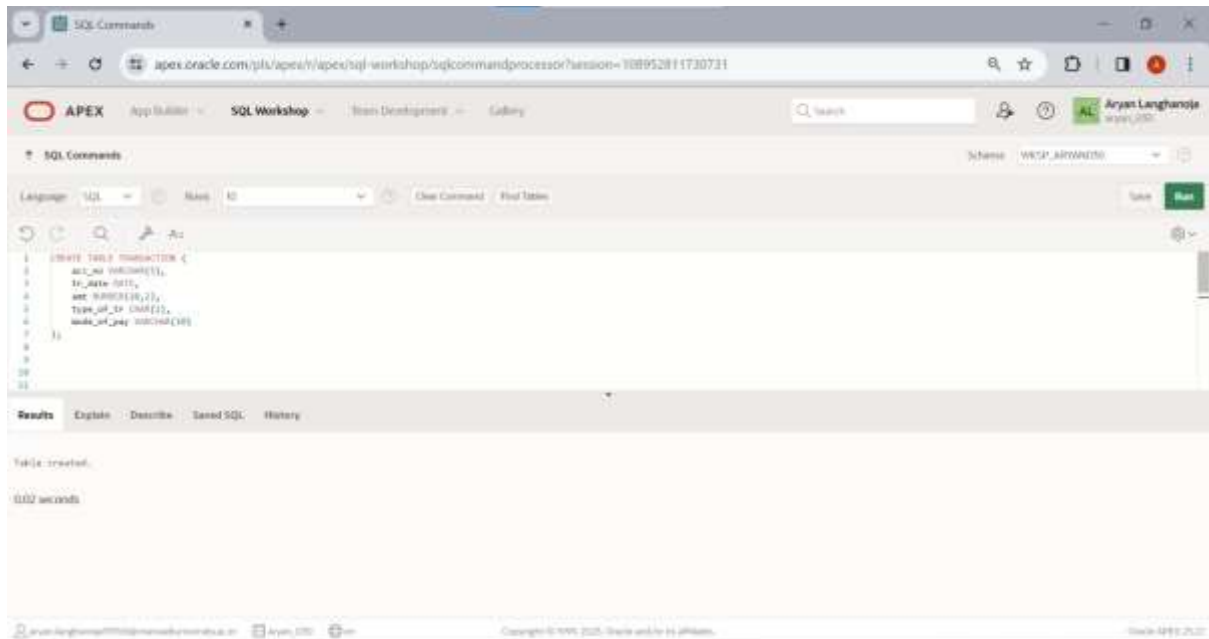
25 INSERT INTO INSTALLMENT VALUES ('L001', 'I001', TO_DATE('2-Feb-2004', 'DD-MON-YYYY'), 15000.00);
26 INSERT INTO INSTALLMENT VALUES ('L002', 'I002', TO_DATE('18-June-2004', 'DD-MON-YYYY'), 20000.00);
27 INSERT INTO INSTALLMENT VALUES ('L003', 'I003', TO_DATE('15-July-2004', 'DD-MON-YYYY'), 20000.00);
  
```

LOAN_NO	INST_NO	INST_DATE	AMOUNT
L001	I001	02/02/2004	15000
L002	I002	06/06/2004	20000
L003	I003	07/07/2004	20000

3 rows returned in 0.02 seconds

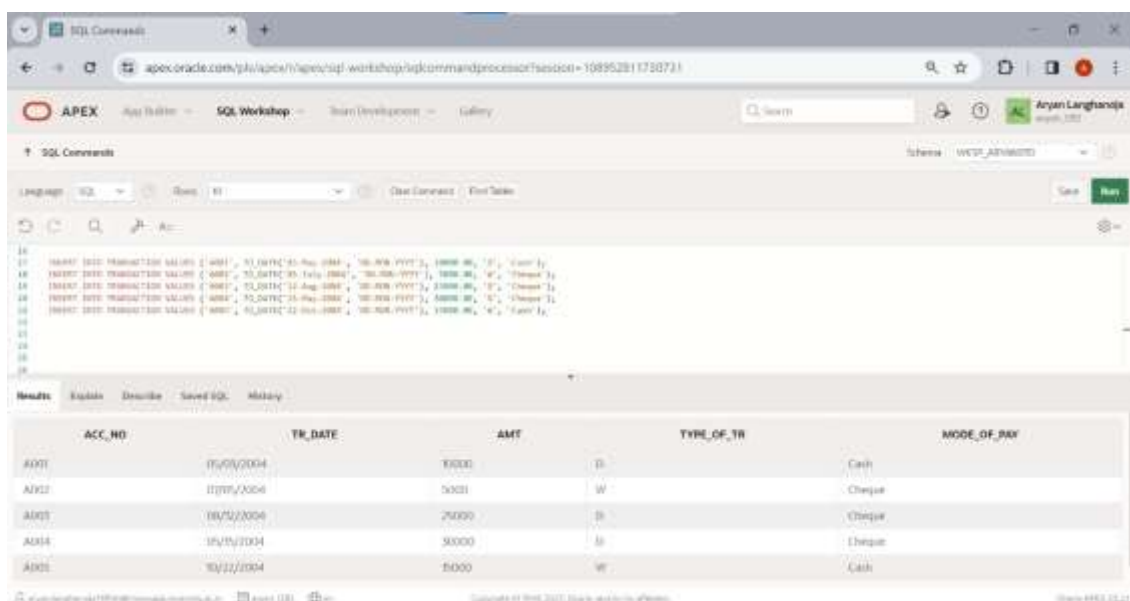
### Create a Table TRANSACTION

Column Name	Data Type	Size
acc_no	Varchar2	5
tr_Date	Date	
Amt	Number	10,2
type_of_tr	Char	1
mode_of_pay	Varchar2	10



### Insert a Following Records

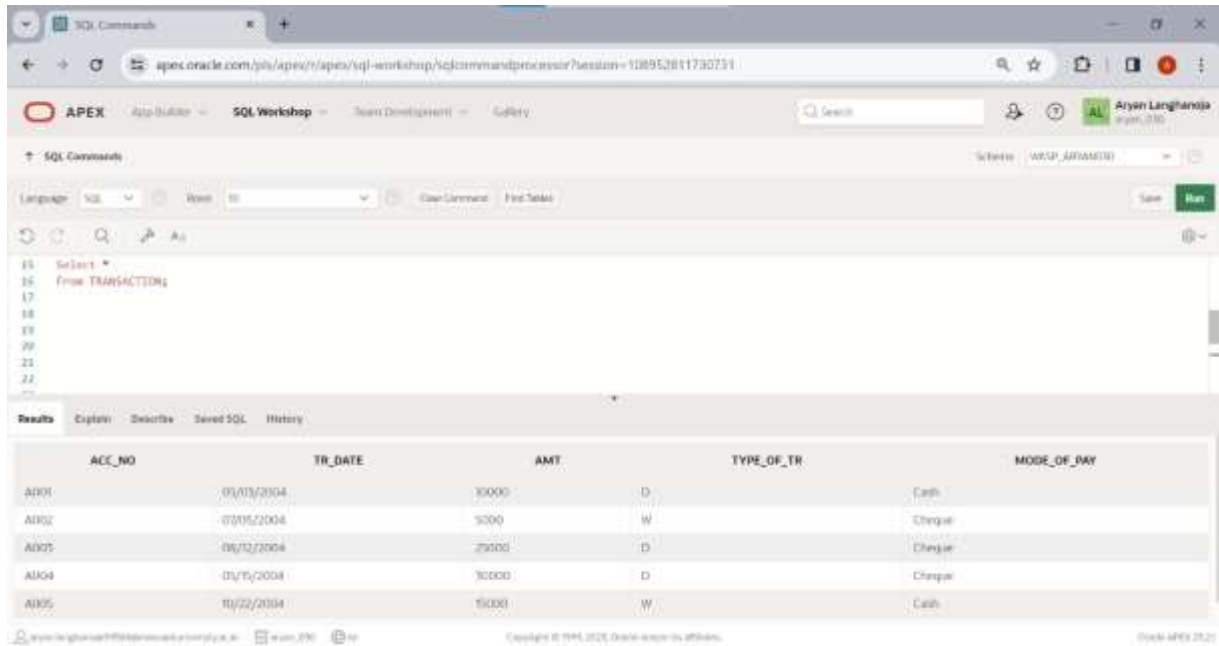
Acc_no	Date	Amt	Type_of_tr	Mode_of_pay
A001	3-may-04	10000	D	Cash
A002	5-july-04	5000	W	Cheque
A003	12-Aug-04	25000	D	Cheque
A004	15-may-04	30000	D	Cheque
A005	22-oct-04	15000	W	Cash



## Table-Account

### List of queries

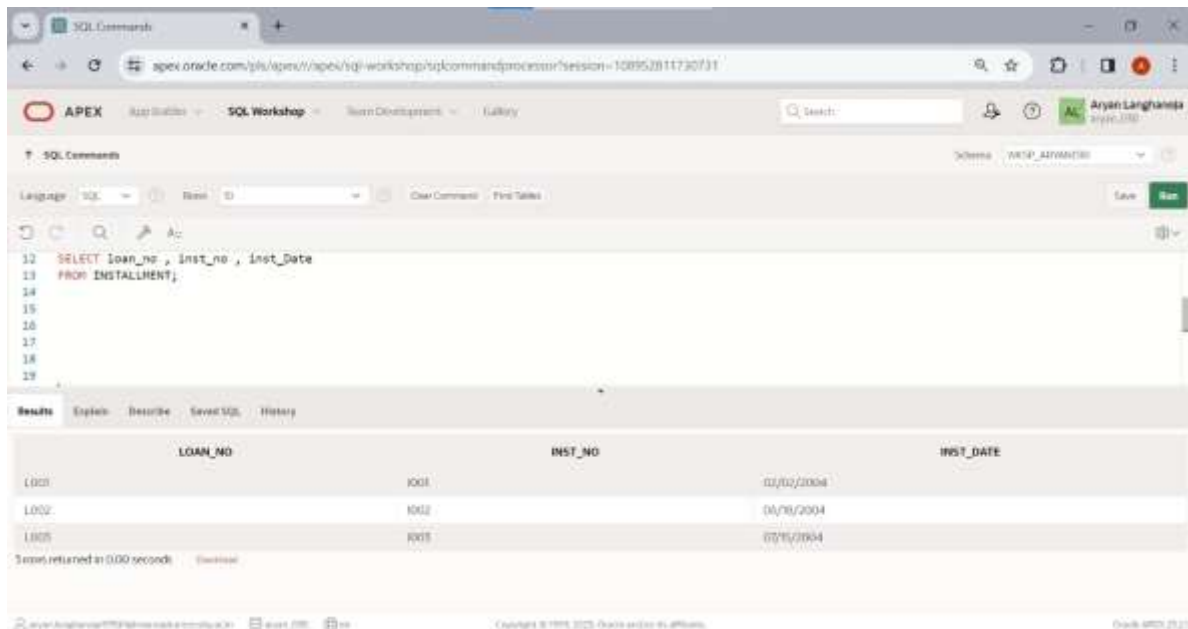
1. Display all rows and all columns of table Transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is `SELECT * FROM TRANSACTION;`. The results are displayed in a table with the following columns: **ACC\_NO**, **TR\_DATE**, **AMT**, **TYPE\_OF\_TR**, and **MODE\_OF\_PAY**.

ACC_NO	TR_DATE	AMT	TYPE_OF_TR	MODE_OF_PAY
AK01	01/01/2004	10000	D	Cash
AK02	07/05/2004	5000	W	Cheque
AK03	08/12/2004	25000	D	Cheque
AK04	01/16/2004	30000	D	Cheque
AK05	10/22/2004	15000	W	Cash

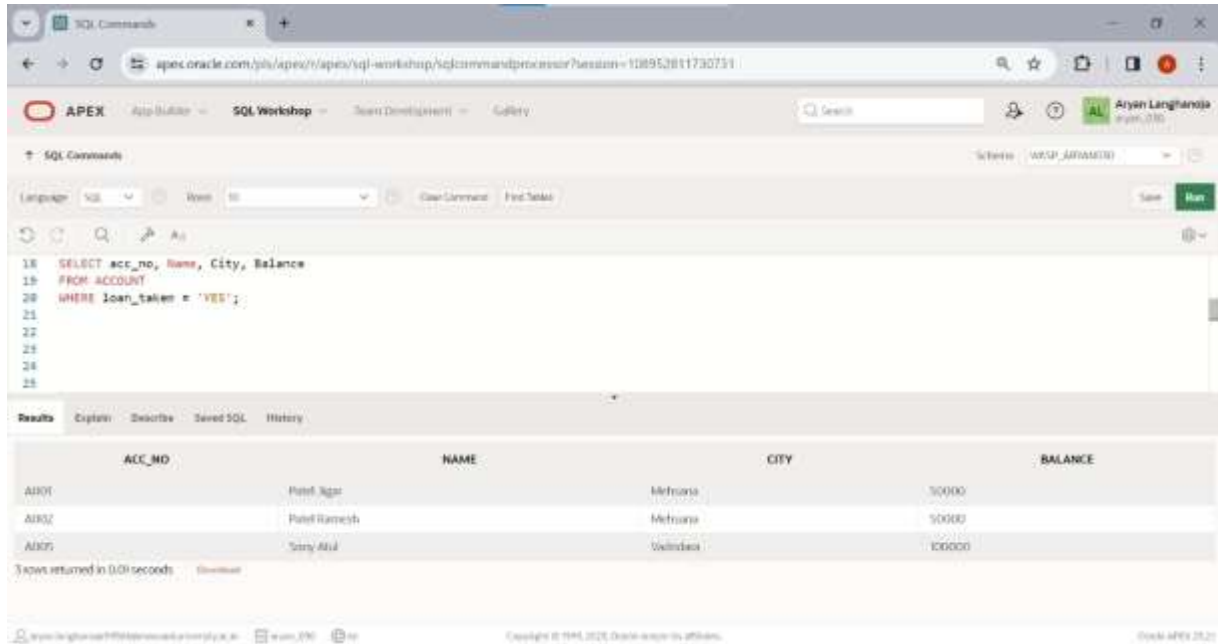
2. Display all rows and selected columns of table Installment.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is `SELECT loan_no, inst_no, inst_date FROM INSTALLMENT;`. The results are displayed in a table with the following columns: **LOAN\_NO**, **INST\_NO**, and **INST\_DATE**.

LOAN_NO	INST_NO	INST_DATE
1001	1001	01/01/2004
1002	1002	01/16/2004
1003	1003	07/16/2004

3. Display selected rows and selected columns of table Account.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

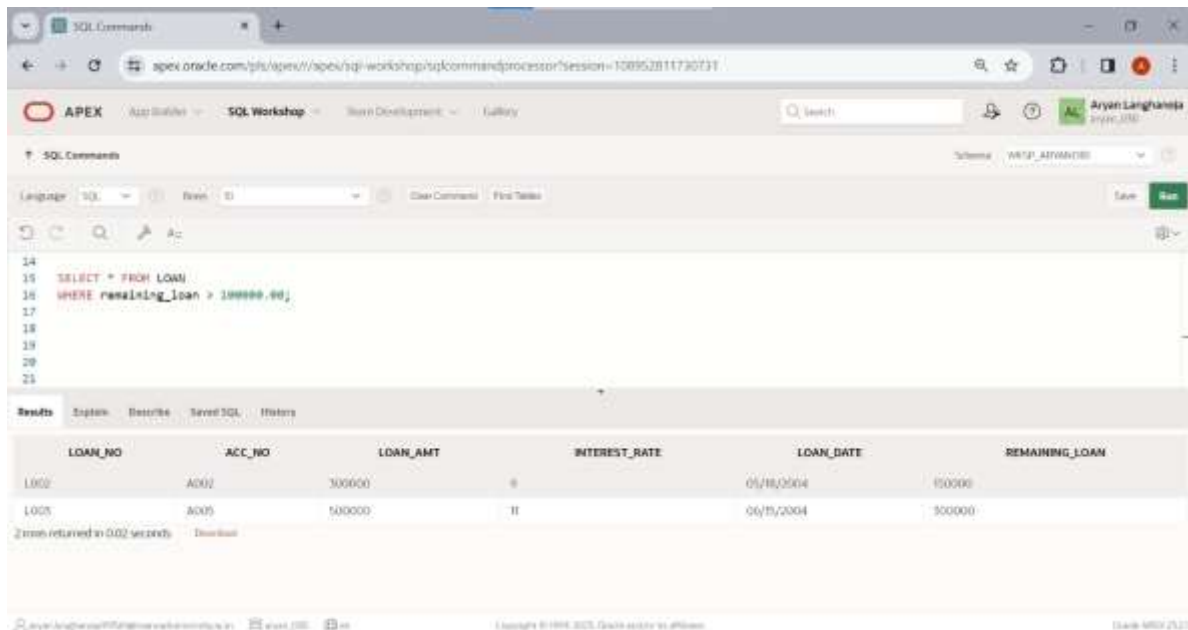
```
18 SELECT acc_no, Name, City, Balance
19 FROM ACCOUNT
20 WHERE loan_taken = 'YES';
```

The results are displayed in a table with the following columns: ACC\_NO, NAME, CITY, and BALANCE. The data returned is:

ACC_NO	NAME	CITY	BALANCE
A001	Patel Raju	Mehsana	50000
A002	Patel Ramesh	Mehsana	50000
A005	Sony Atul	Vadodra	100000

3 rows returned in 0.01 seconds.

4. Display selected rows and all columns of table loan.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
14 SELECT * FROM LOAN
15 WHERE remaining_loan > 100000.00;
```

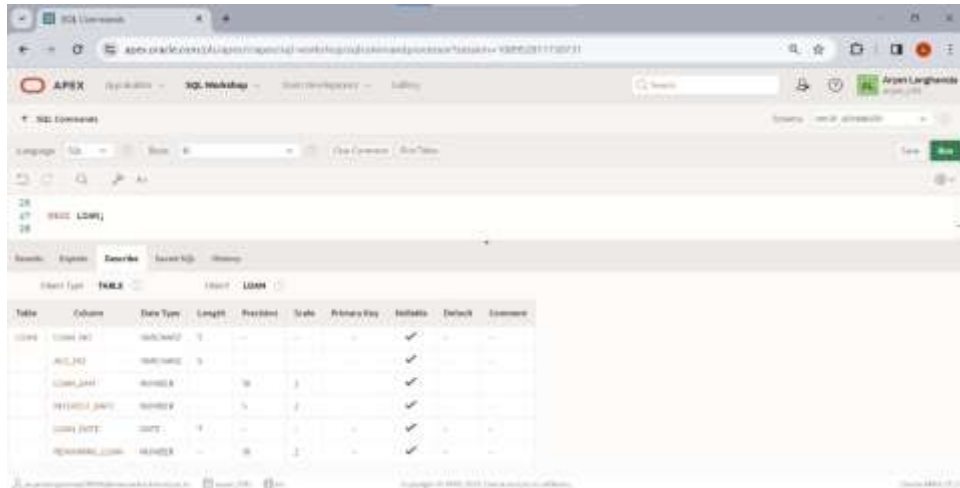
The results are displayed in a table with the following columns: LOAN\_NO, ACC\_NO, LOAN\_AMT, INTEREST\_RATE, LOAN\_DATE, and REMAINING\_LOAN. The data returned is:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN
L002	A002	500000	8	05/18/2004	150000
L005	A005	500000	11	06/19/2004	500000

2 rows returned in 0.02 seconds.

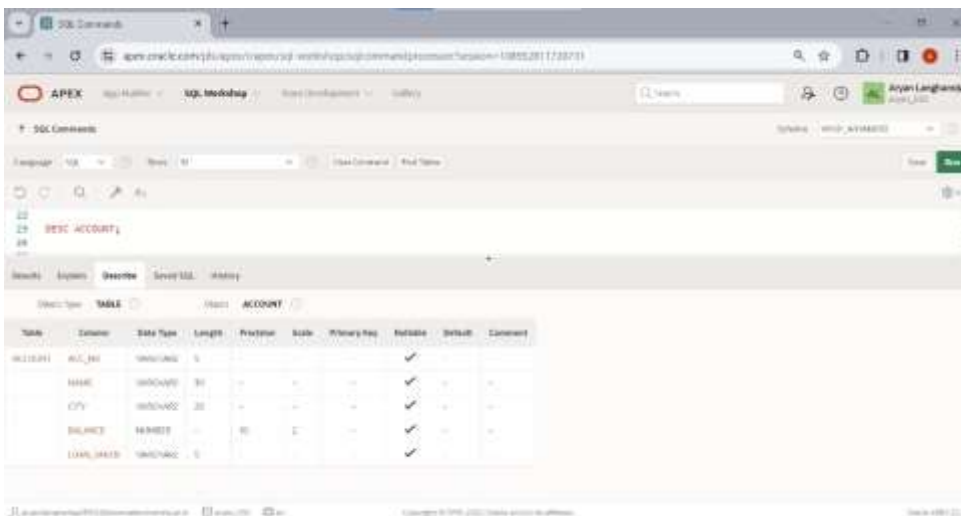


5. Show the structure of the table loan, account and transaction.
- desc loan;



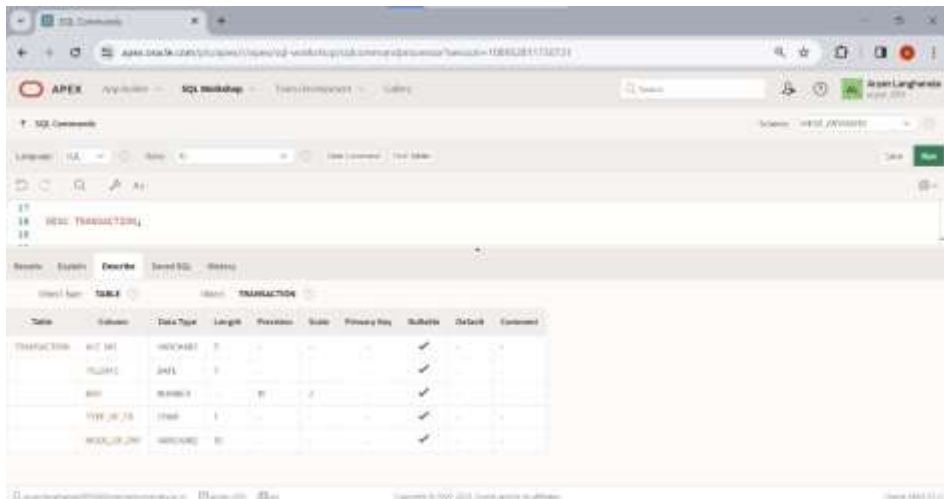
The screenshot shows the SQL Workshop interface with the 'DESCRIBE' tab selected for the 'LOAN' table. The table structure is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOAN	LOAN_ID	NUMBER	9				✓		
	ACCT_ID	NUMBER	9				✓		
	LOAN_DATE	DATE					✓		
	PRINCIPAL_AMOUNT	NUMBER	9		2		✓		
	LOAN_RATE	NUMBER	9		2		✓		
	INTEREST_RATE	NUMBER	9		2		✓		



The screenshot shows the SQL Workshop interface with the 'DESCRIBE' tab selected for the 'ACCOUNT' table. The table structure is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNT	ACCT_ID	NUMBER	9				✓		
	NAME	VARCHAR2	30				✓		
	CPF	NUMBER	20				✓		
	BALANCE	NUMBER	9		2		✓		
	LOAN_AMOUNT	NUMBER	9				✓		



The screenshot shows the SQL Workshop interface with the 'DESCRIBE' tab selected for the 'TRANSACTION' table. The table structure is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANSACTION	TXN_ID	NUMBER	9				✓		
	TXN_DATE	DATE					✓		
	AMT	NUMBER	9		2		✓		
	TXN_TYPE	CHAR	1				✓		
	ACCOUNT_ID	NUMBER	9				✓		

## Practical 3

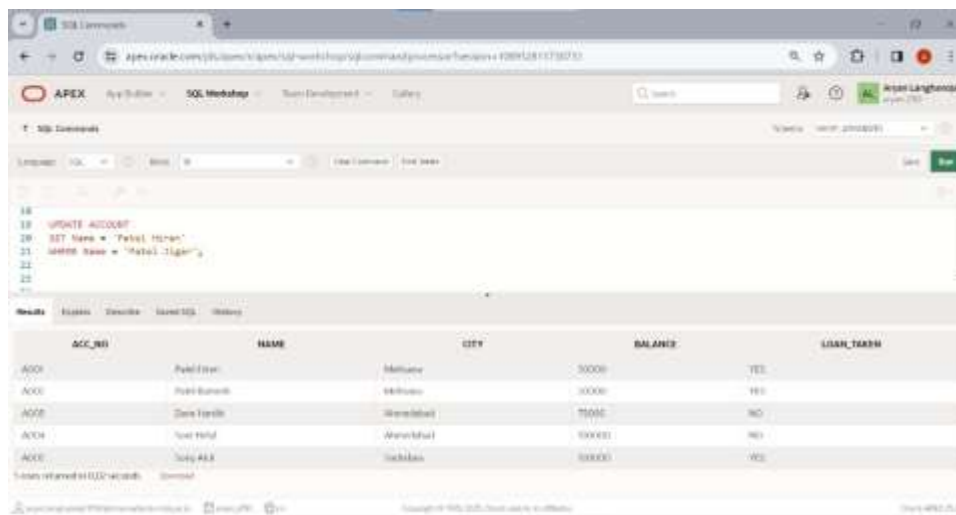
### Aim: DML Commands and Queries

Table: **ACCOUNT**.

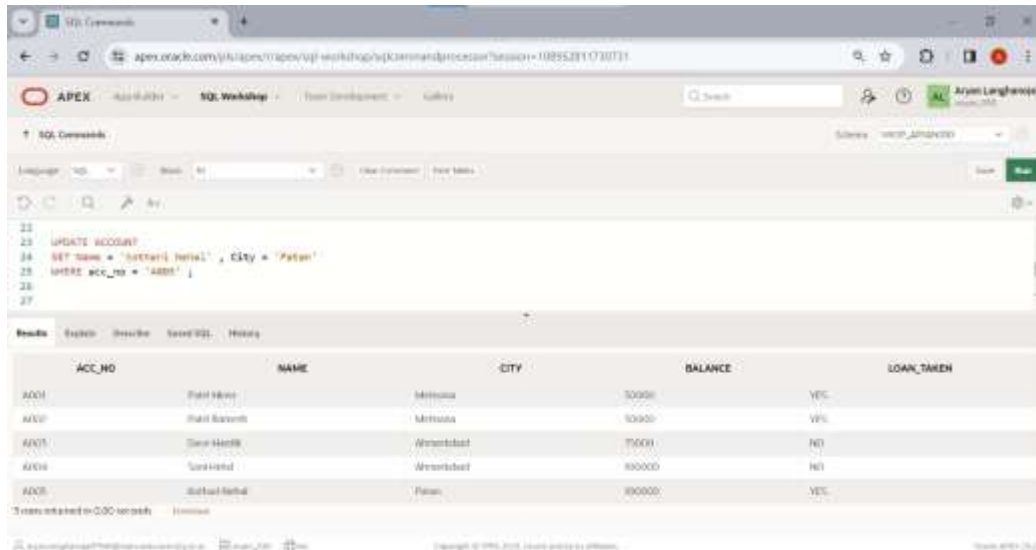
Insert the following records if you have not inserted in PRACTICAL - 1

Acc_no	Name	City	Balance	Loan_tak e
A001	Patel Jigar	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	Yes
A003	Dave Hardik	Ahmedabad	75000	NO
A004	Soni Hetal	Ahmedabad	100000	NO
A005	Soni Atul	Vadodara	100000	YES

1. Change the name 'patel jigar' to 'patel hiren'.



- Change the name and city where account number is A005. (new name = 'kothari nehal' and new city = 'patan').



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

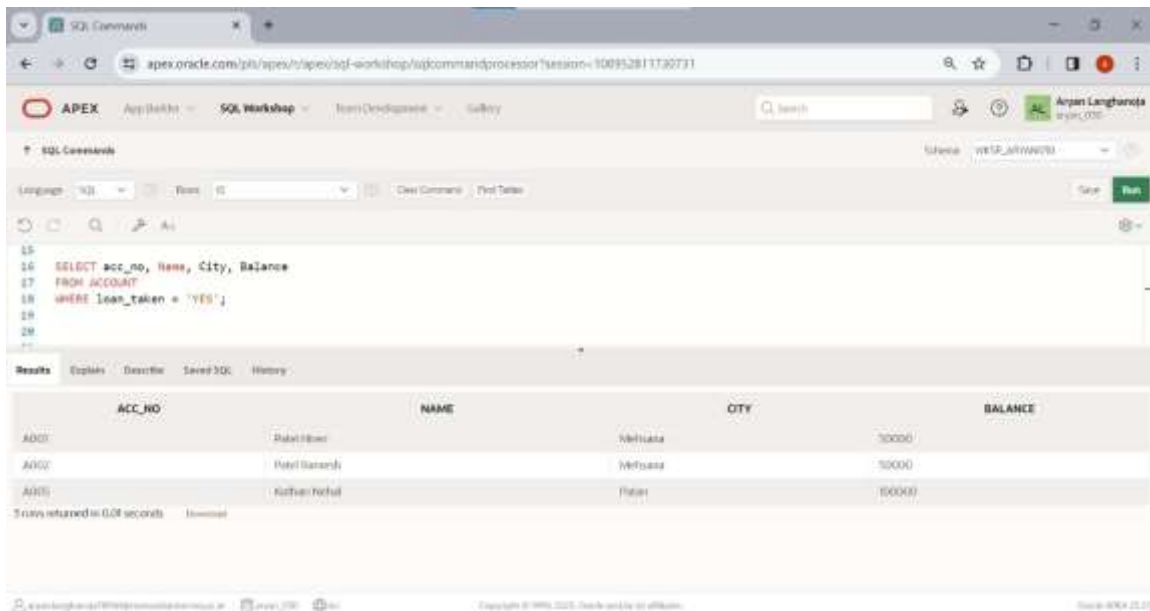
```

22 UPDATE ACCOUNT
23 SET name = 'kothari nehal', city = 'Patan'
24 WHERE acc_no = 'A005';
  
```

The results table shows the following data:

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN
A001	Rajesh Kumar	Mumbai	50000	YES
A002	Rajesh Kumar	Mumbai	50000	YES
A003	Rajesh Kumar	Mumbai	50000	NO
A004	Rajesh Kumar	Mumbai	50000	NO
A005	Kothari Nehal	Patan	50000	YES

- Display only those records where loan taken status is 'YES'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

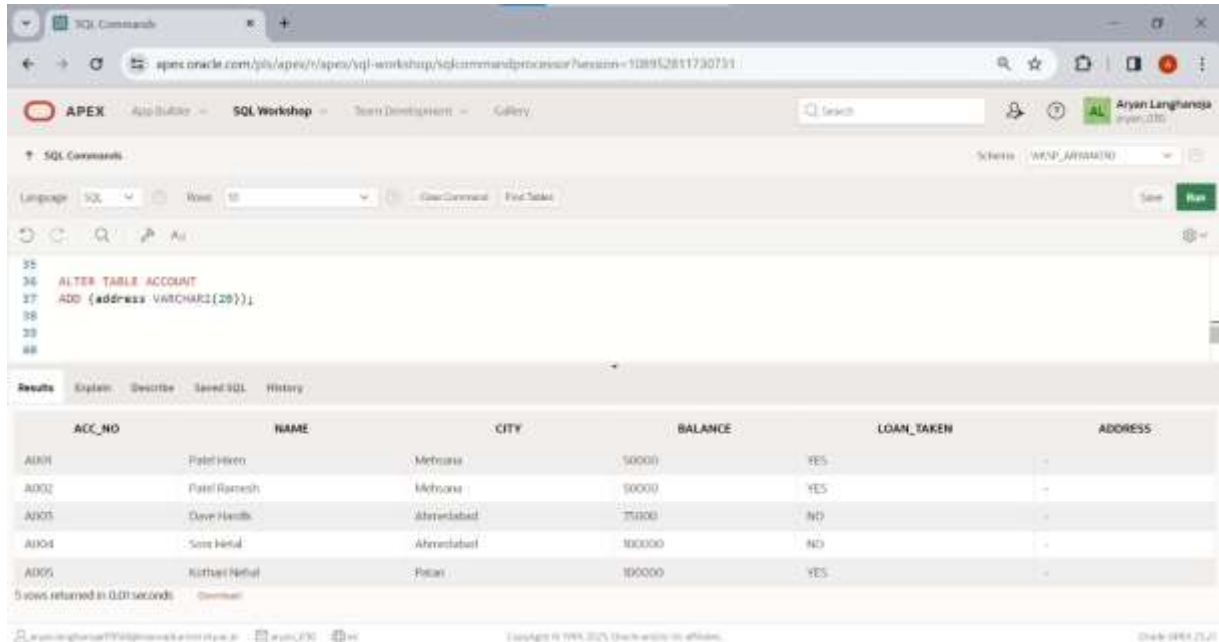
```

15 SELECT acc_no, name, city, balance
16 FROM ACCOUNT
17 WHERE loan_taken = 'YES';
  
```

The results table shows the following data:

ACC_NO	NAME	CITY	BALANCE
A001	Rajesh Kumar	Mumbai	50000
A002	Rajesh Kumar	Mumbai	50000
A005	Kothari Nehal	Patan	50000

4. Add the new column (address varchar2 (20)) into table ACCOUNT.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

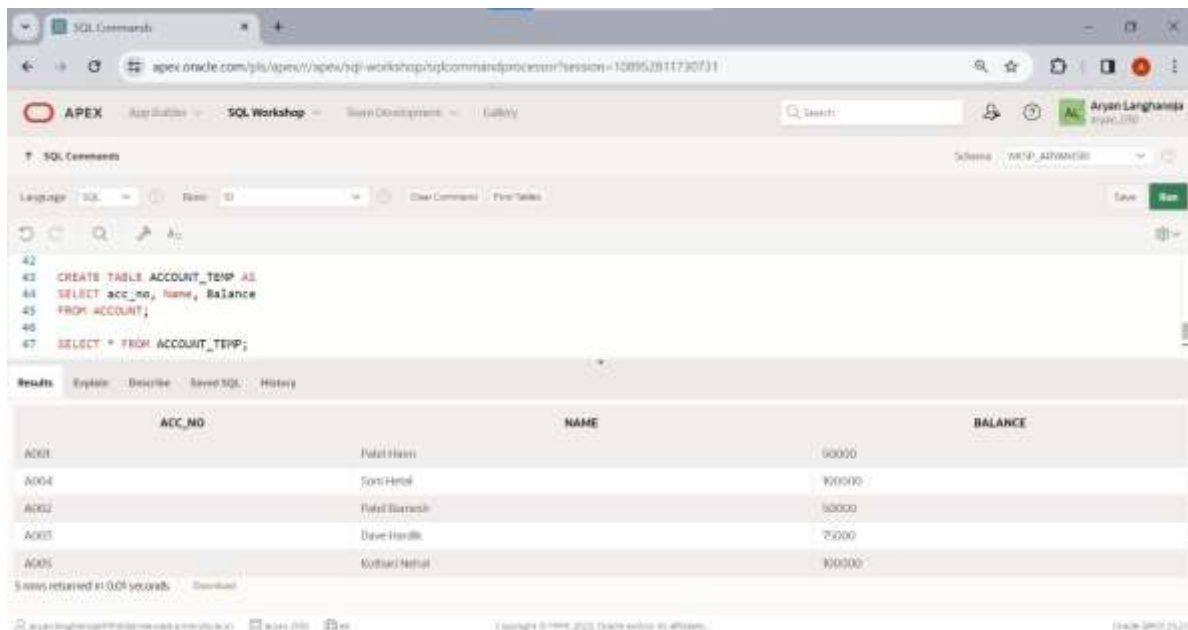
```
35
36 ALTER TABLE ACCOUNT
37 ADD (address VARCHAR2(20));
38
39
40
```

The results table shows the data after the command execution:

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN	ADDRESS
A001	Rishi Hiren	Mehsana	50000	YES	-
A002	Rishi Ramesh	Mehsana	90000	YES	-
A003	Dave Hardik	Ahmedabad	75000	NO	-
A004	Soni Hetal	Ahmedabad	80000	NO	-
A005	Kothari Nehal	Ratan	80000	YES	-

5 rows returned in 0.01 seconds

5. Create another table ACCOUNT\_TEMP (acc\_no, name, balance) from table ACCOUNT.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

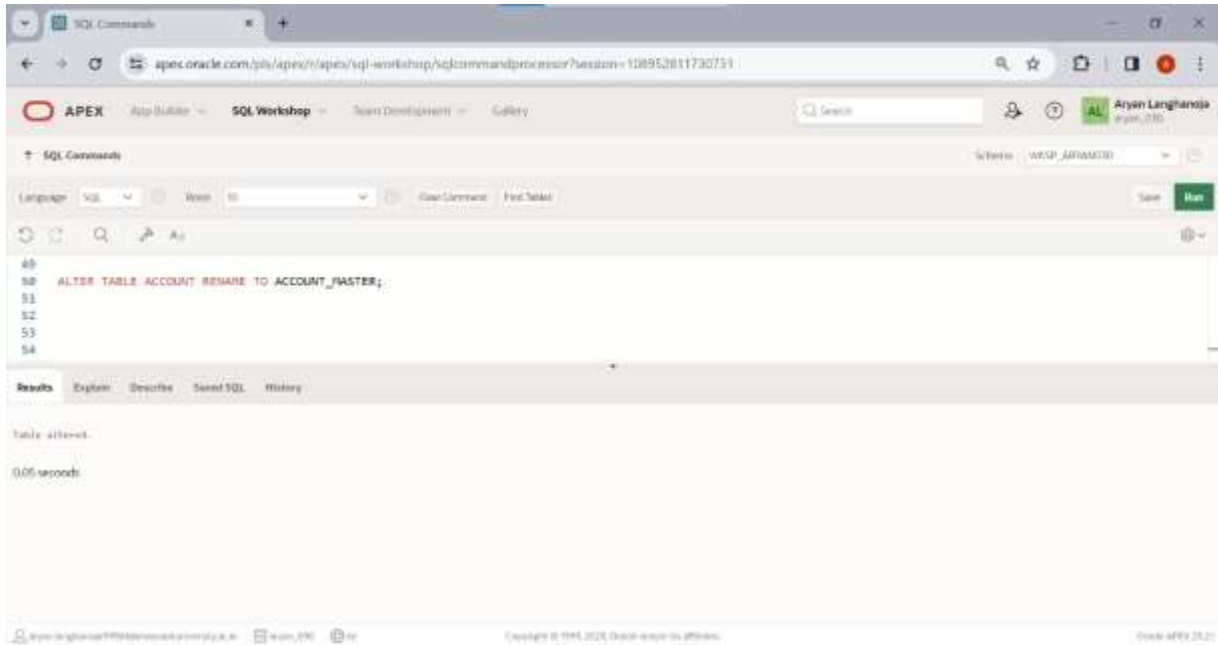
```
42
43 CREATE TABLE ACCOUNT_TEMP AS
44 SELECT acc_no, name, balance
45 FROM ACCOUNT;
46
47 SELECT * FROM ACCOUNT_TEMP;
```

The results table shows the data after the command execution:

ACC_NO	NAME	BALANCE
A001	Rishi Hiren	50000
A004	Soni Hetal	80000
A002	Rishi Ramesh	90000
A003	Dave Hardik	75000
A005	Kothari Nehal	80000

5 rows returned in 0.04 seconds

6. Rename the table ACCOUNT to ACCOUNT\_MASTER.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

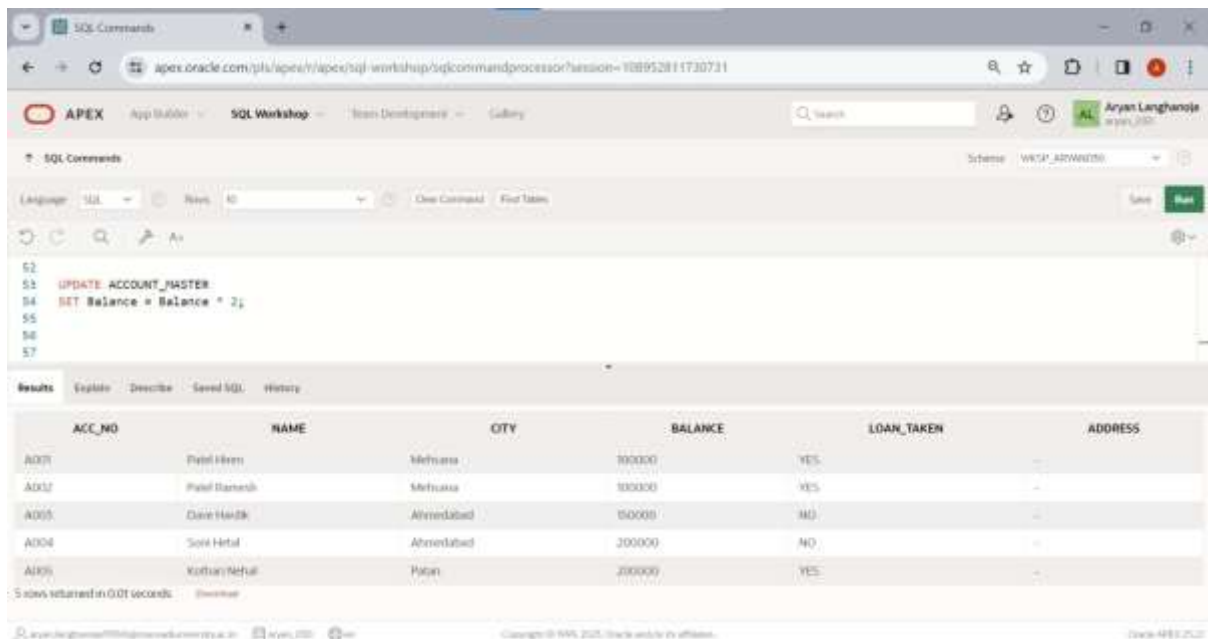
```

ALTER TABLE ACCOUNT RENAME TO ACCOUNT_MASTER;

```

The command was executed successfully, and the results pane shows "Table altered." and "0.05 seconds".

7. Update the column balance for all the account holders. (Multiply the balance by 2 for each account holders)



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

UPDATE ACCOUNT_MASTER
SET Balance = Balance * 2;

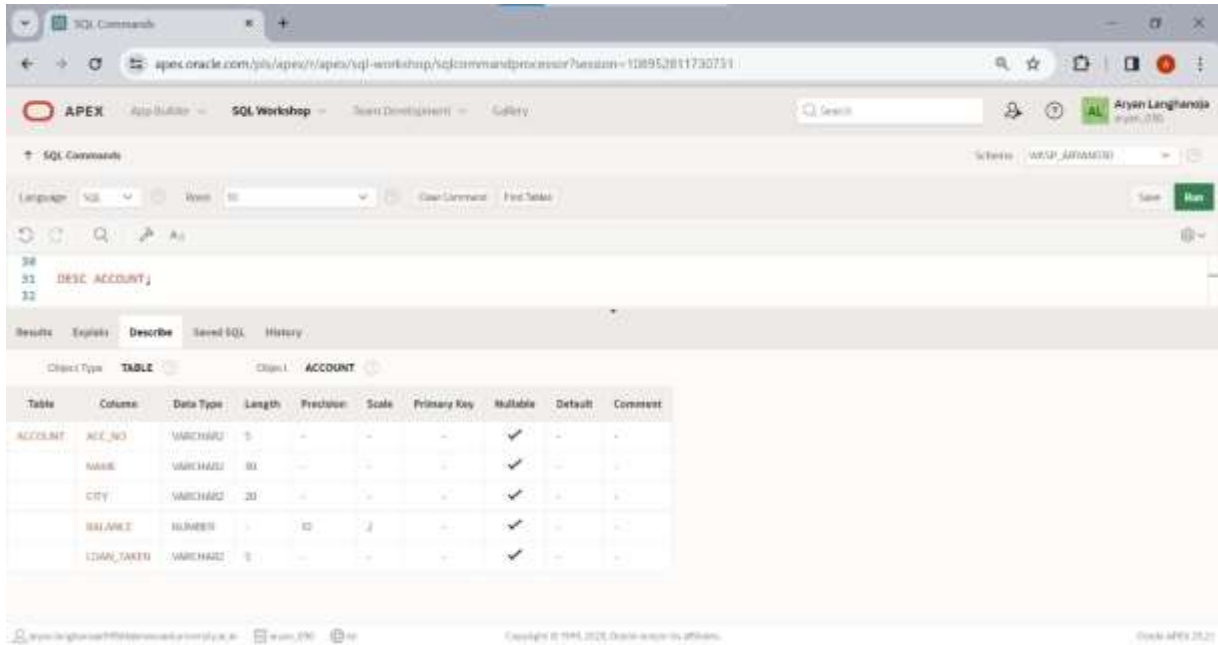
```

The command was executed successfully. Below the command, a table view shows the updated data:

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN	ADDRESS
A001	Patel Hiren	Mehruva	100000	YES	---
A002	Patel Bharad	Mehruva	100000	YES	---
A003	Dave Harsh	Ahmedabad	150000	NO	---
A004	Soni Hetal	Ahmedabad	200000	NO	---
A005	Kothari Nehal	Patan	200000	YES	---

5 rows returned in 0.01 seconds.

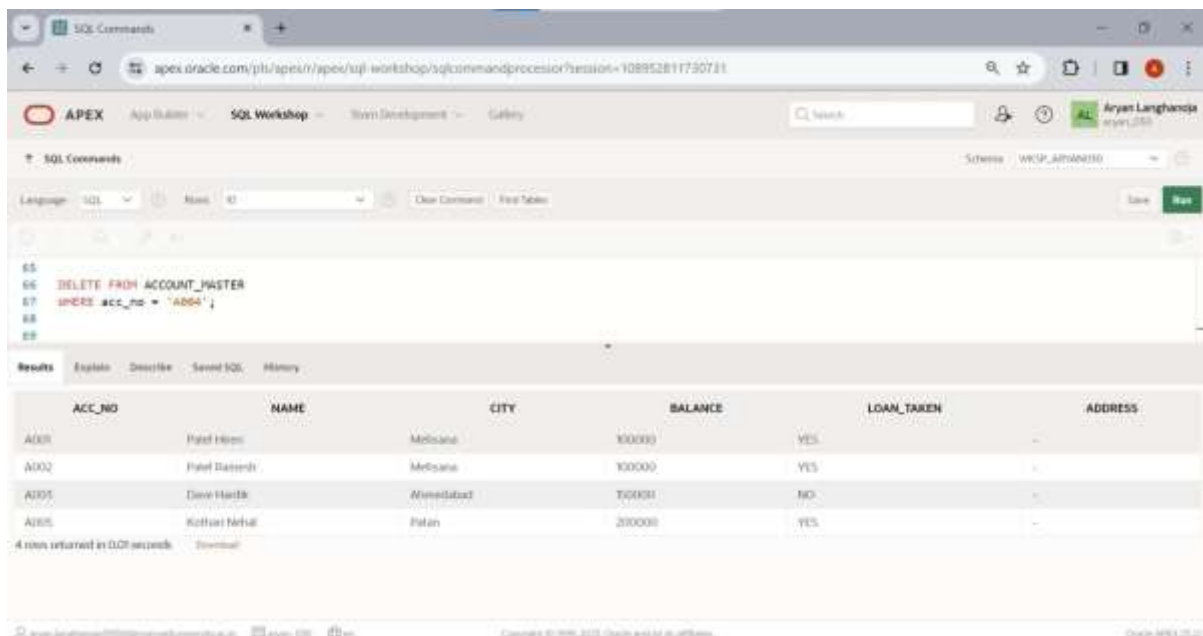
8. Describe the structure of table ACCOUNT.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is `DESC ACCOUNT;`. The results tab is selected, displaying the structure of the **ACCOUNT** table.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNT	ACC_NO	VARCHAR2	5				✓		
	NAME	VARCHAR2	80				✓		
	CITY	VARCHAR2	20				✓		
	BALANCE	NUMBER		10	2		✓		
	LOAN_TAKEN	VARCHAR2	5				✓		

9. Delete the records whose account no is A004.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is `DELETE FROM ACCOUNT_MASTER WHERE acc_no = 'A004';`. The results tab is selected, displaying the data after the deletion.

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN	ADDRESS
A001	Pooja Hiran	Meliana	100000	YES	
A002	Pooja Dattani	Meliana	100000	YES	
A003	Dani Hardik	Meliana	100000	NO	
A005	Kishan Nehal	Patna	200000	YES	

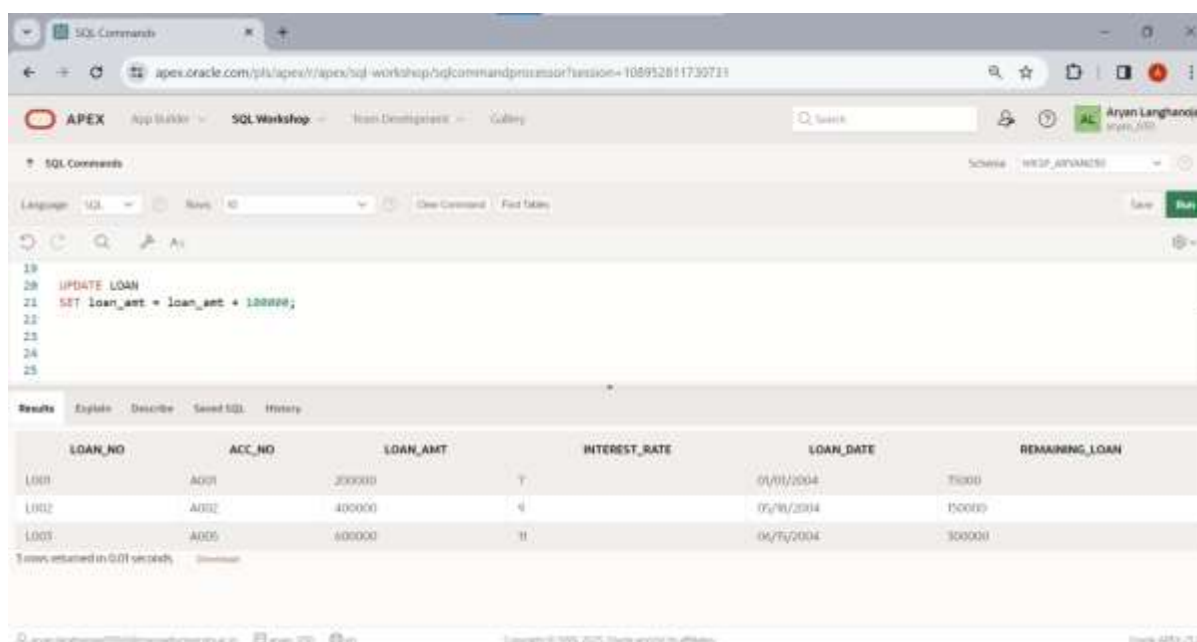
4 rows returned in 0.01 seconds

Table: **LOAN.**

Insert the following Records if you have not inserted in PRACTICAL-1

Loan_no	Acc_no	Loan_amt	Interest_rate	Loan_date	Remaining_loan
L001	A001	100000	7	1-jan-04	75000
L002	A002	300000	9	18-may-04	150000
L003	A005	500000	11	15-june-04	300000

1. For each loan holders Add 100000 Rs. Amount into the column loan\_amt.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

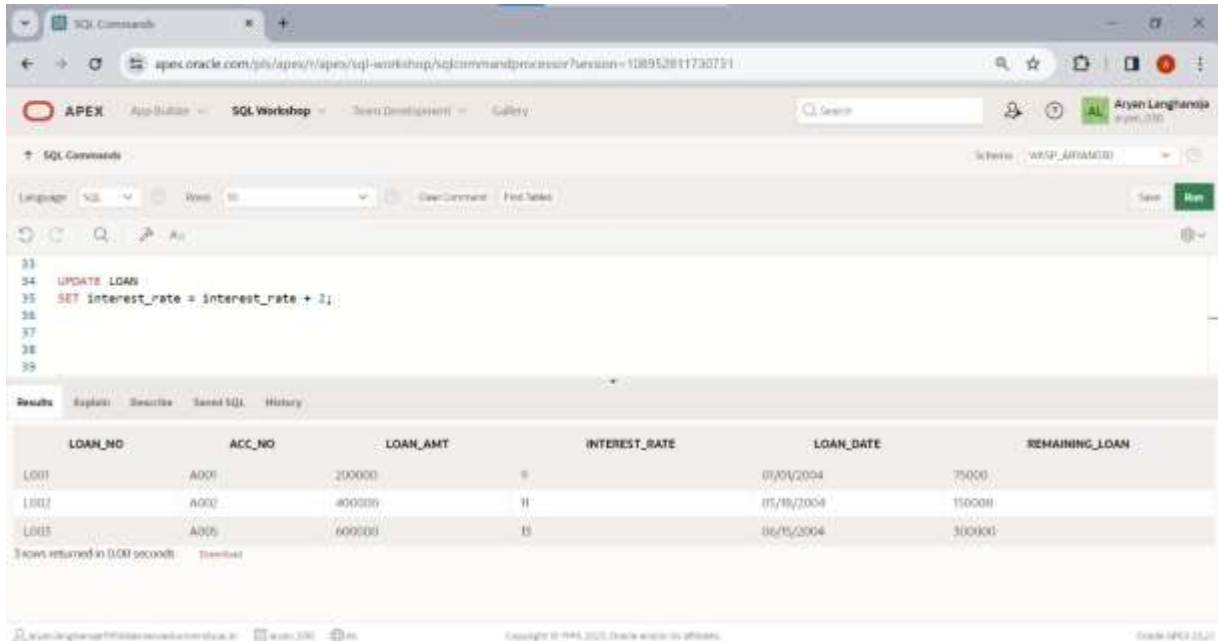
```
UPDATE LOAN
SET loan_amt = loan_amt + 100000;
```

The results table shows the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN
L001	A001	200000	7	01/01/2004	75000
L002	A002	400000	9	05/05/2004	150000
L003	A005	600000	11	06/07/2004	300000

Times elapsed in 0.01 seconds

- for each loan holders Increase the interest rate 2%.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

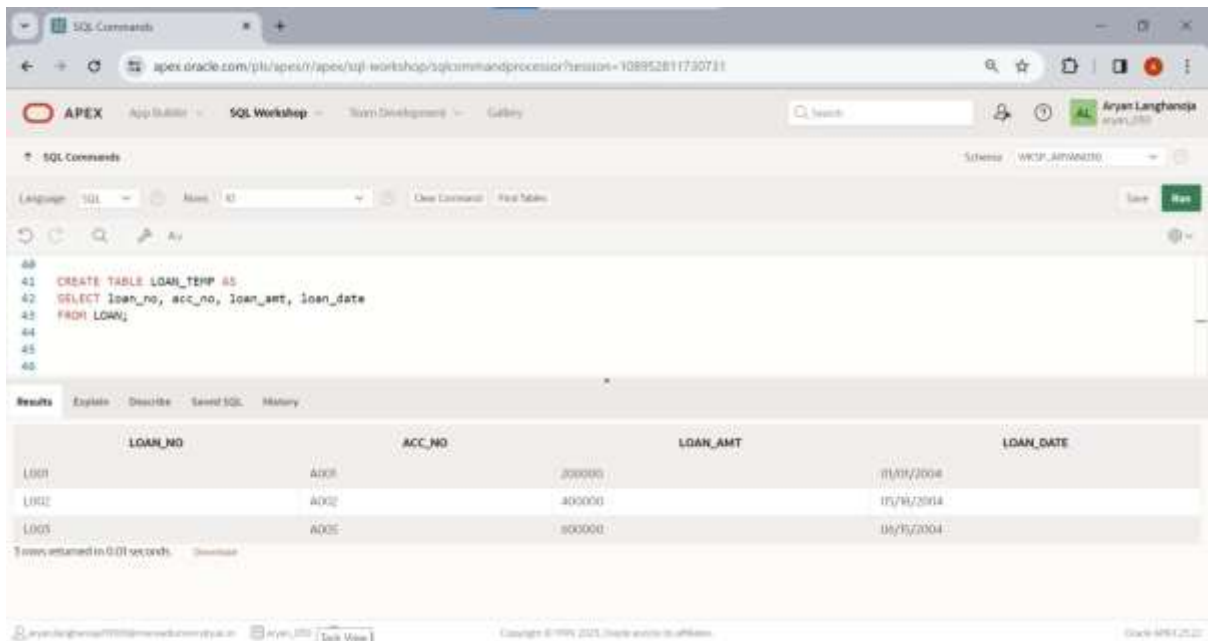
33
34 UPDATE LOAN
35 SET interest_rate = interest_rate + 2;
36
37
38
39
  
```

The results table shows the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN
LO01	A001	200000	8	01/01/2004	75000
LO02	A002	400000	11	05/10/2004	150000
LO03	A003	600000	15	06/05/2004	300000

3 rows returned in 0.00 seconds.

- Create another table LOAN\_TEMP (loan\_no, Acc\_no, loan\_amt, loan\_date) from The table LOAN.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

40
41 CREATE TABLE LOAN_TEMP AS
42 SELECT loan_no, acc_no, loan_amt, loan_date
43 FROM LOAN;
44
45
46
  
```

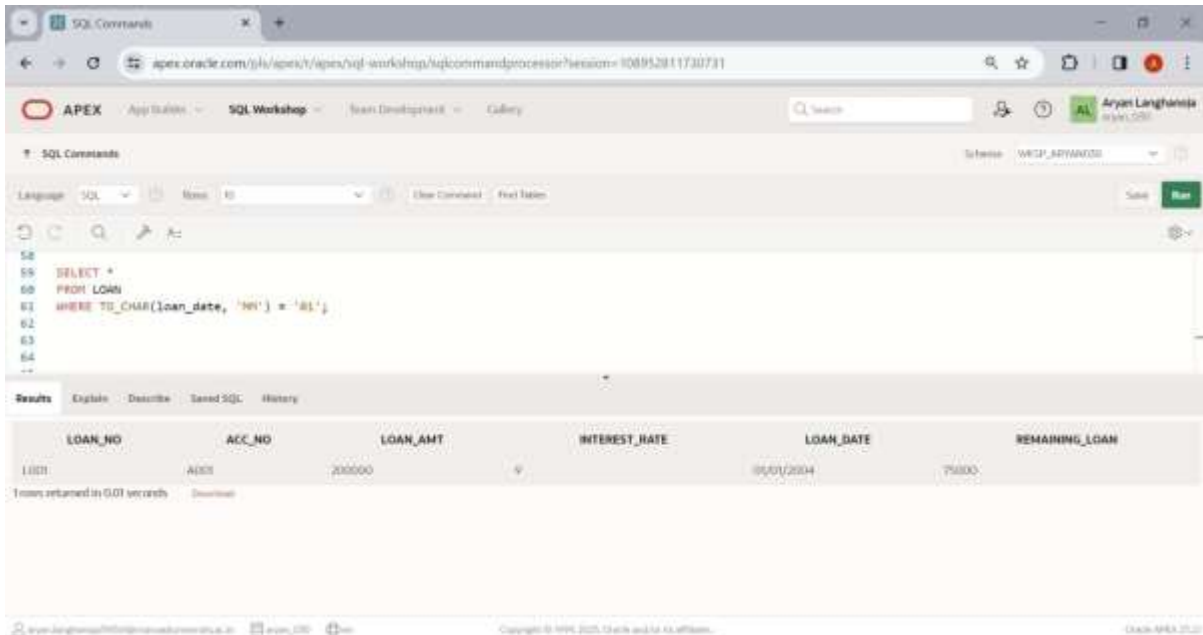
The results table shows the following data:

LOAN_NO	ACC_NO	LOAN_AMT	LOAN_DATE
LO01	A001	200000	01/01/2004
LO02	A002	400000	05/10/2004
LO03	A003	600000	06/05/2004

3 rows returned in 0.01 seconds.



4. Display only those records where loan holder taken a loan in month of January.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

```

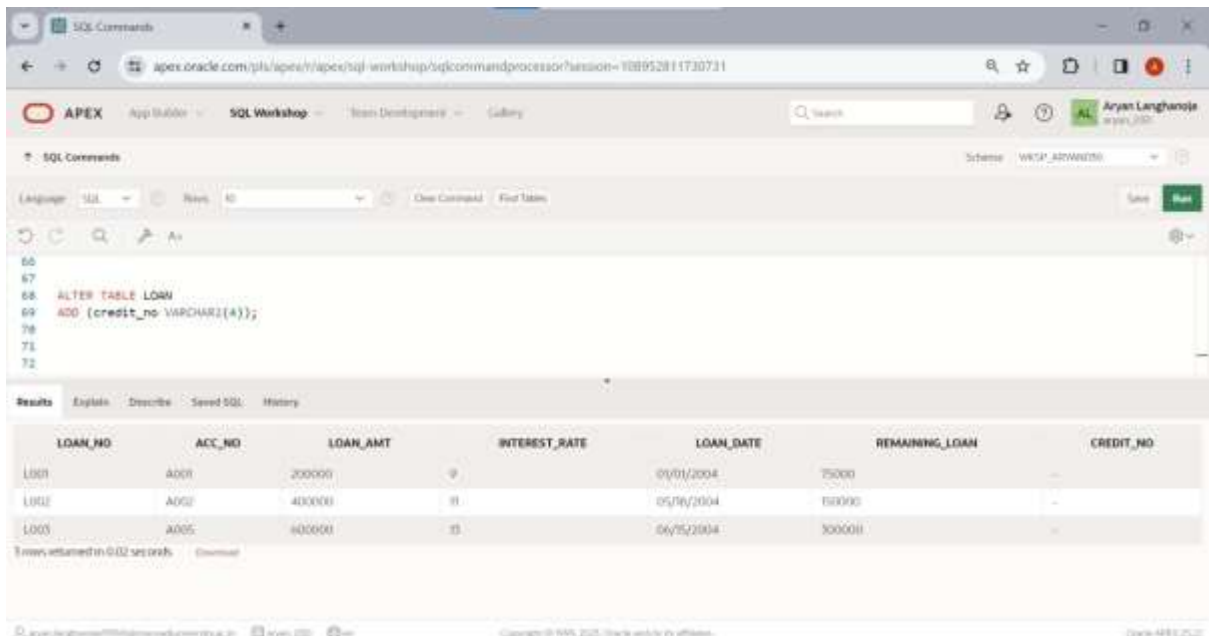
58
59 SELECT *
60 FROM LOAN
61 WHERE TO_CHAR(loan_date, 'MM') = '01';
62
63
64
  
```

The Results tab displays the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN
1001	A001	200000	9	01/01/2004	75000

1 rows returned in 0.01 seconds

5. Modify the structure of table LOAN by adding one column credit\_no varchar2 (4).



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

```

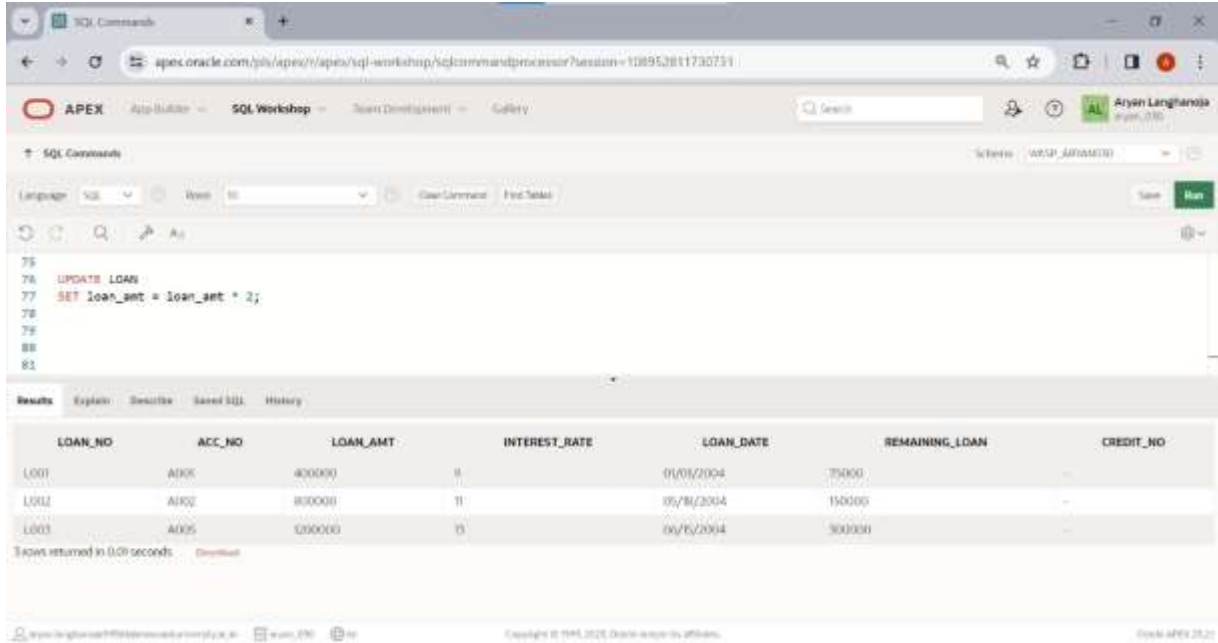
66
67
68 ALTER TABLE LOAN
69 ADD (credit_no VARCHAR2(4));
70
71
72
  
```

The Results tab displays the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN	CREDIT_NO
1001	A001	200000	9	01/01/2004	75000	--
1002	A002	400000	11	05/06/2004	100000	--
1003	A005	600000	13	06/02/2004	300000	--

3 rows returned in 0.02 seconds

6. Display the Loan amount\*2 of table LOAN.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

75
76 UPDATE LOAN
77 SET loan_amt = loan_amt * 2;
78
79
80
81

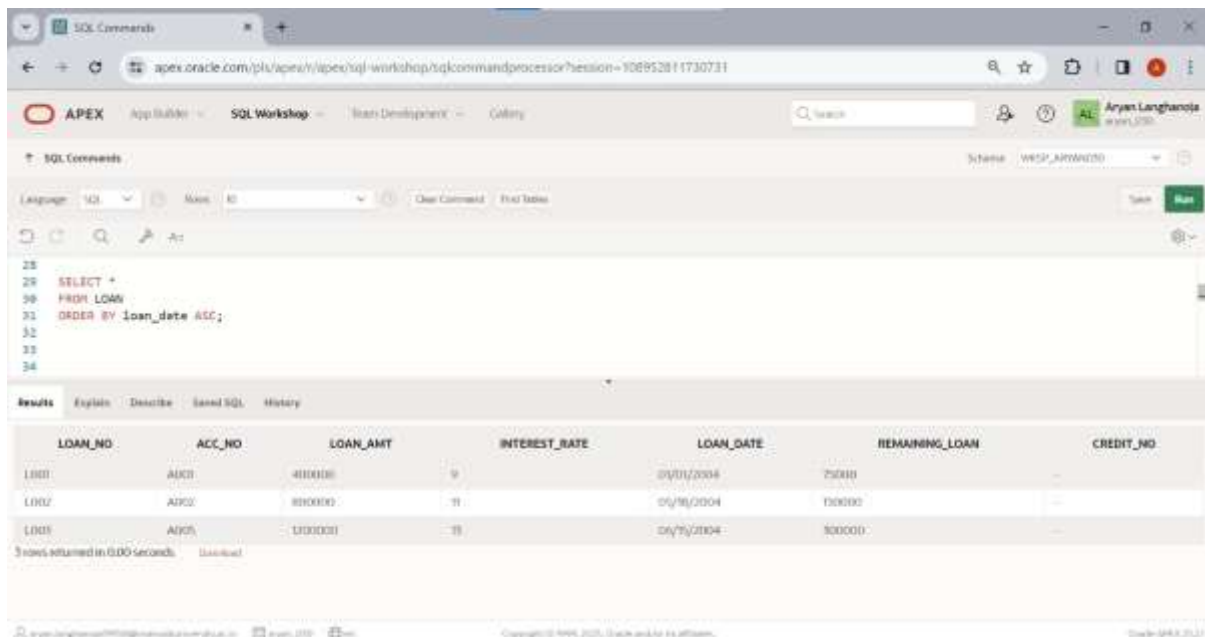
```

The results table shows the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN	CREDIT_NO
L001	A001	400000	8	01/01/2004	75000	--
L002	A002	800000	11	05/08/2004	150000	--
L003	A003	1200000	15	06/06/2004	300000	--

3 rows returned in 0.01 seconds.

7. Display the records of table LOAN by date wise in ascending order.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

28
29 SELECT *
30 FROM LOAN
31 ORDER BY loan_date ASC;
32
33
34

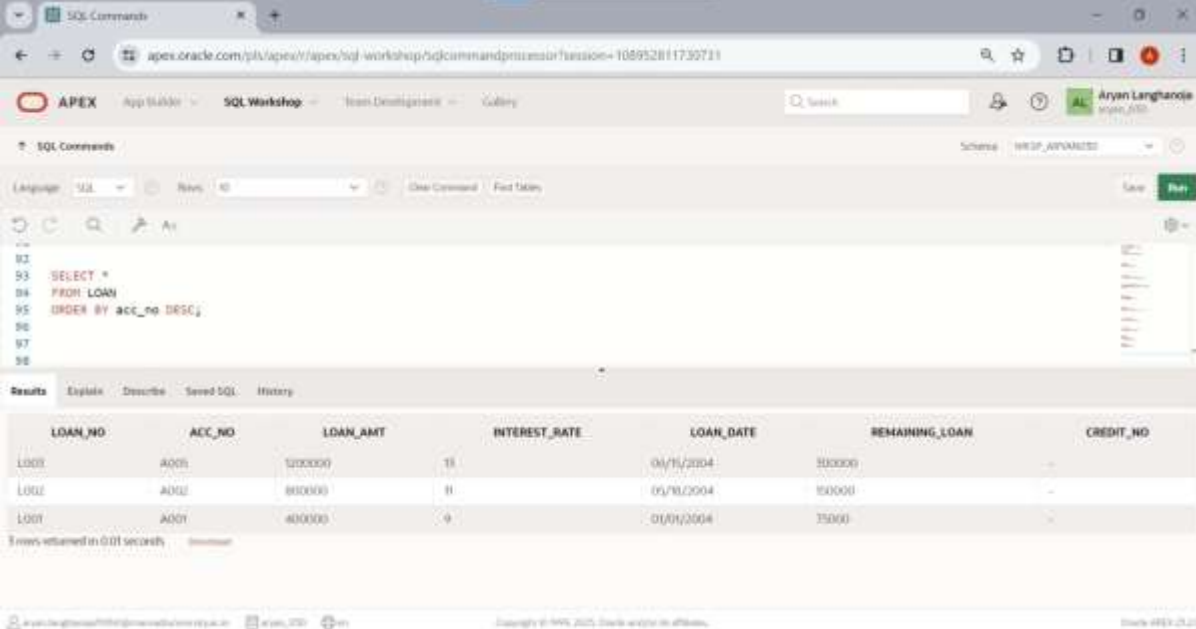
```

The results table shows the following data:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN	CREDIT_NO
L001	A001	400000	8	01/01/2004	75000	--
L002	A002	800000	11	05/08/2004	150000	--
L003	A003	1200000	15	06/06/2004	300000	--

3 rows returned in 0.00 seconds.

8. Display the records of table LOAN by account number wise in descending Order.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

12
13 SELECT *
14 FROM LOAN
15 ORDER BY acc_no DESC;
16
17
18

```

The results are displayed in a table with the following columns: LOAN\_NO, ACC\_NO, LOAN\_AMT, INTEREST\_RATE, LOAN\_DATE, REMAINING\_LOAN, and CREDIT\_NO. The results are ordered by account number in descending order.

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN	CREDIT_NO
LOAN1	ACC01	100000	15	01/11/2004	100000	-
LOAN2	ACC02	800000	11	05/10/2004	100000	-
LOAN3	ACC03	400000	9	01/01/2004	35000	-

3 rows returned in 0.01 seconds

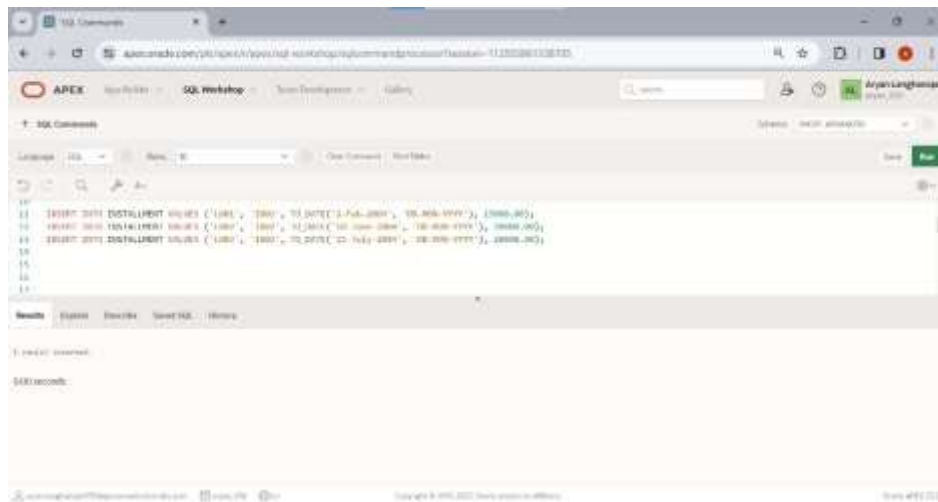
## Practical 4

### Aim: DML Commands and Related Queries

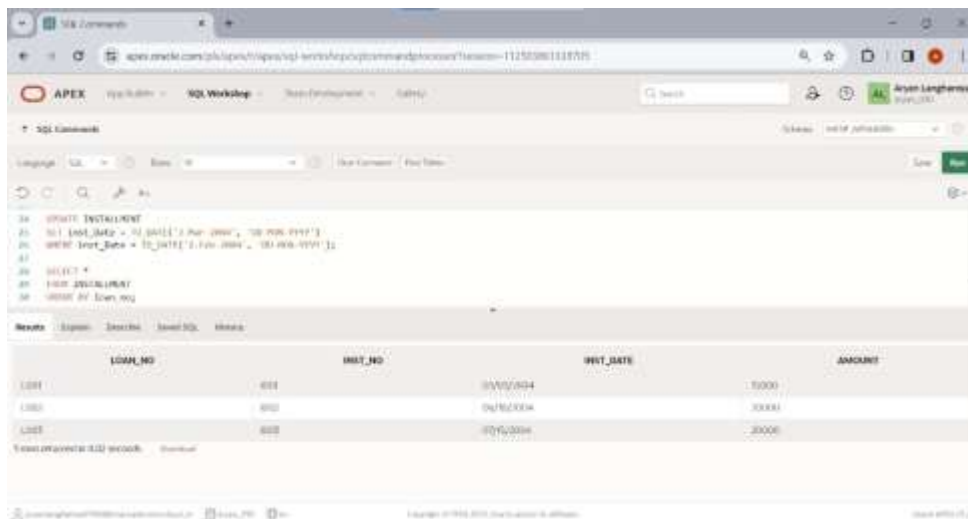
Table: **INSTALLMENT**.

1. Insert following Records if you have not inserted in PRACTICAL-1.

Loan_no	Inst_n o	Inst_Date	Amount
L001	I001	2-Feb-04	15000
L002	I002	18-June-04	20000
L003	I003	15-July-04	20000



2. Change the Inst\_Date '2-Feb-04' to '3-Mar-04'.



3. Reduce 5000 amount from all Installment holders.

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is: `UPDATE INSTALLMENT SET Amount = Amount - 5000;`. The results table shows the following data:

LOAN_NO	INST_NO	INST_DATE	AMOUNT
L001	001	01/01/2004	10000
L002	002	06/06/2004	5000
L003	003	07/07/2004	5000

3 rows returned in 0.18 seconds.

4. Add the amount 5000 where loan no is 'L003' and 'L002'.

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is: `UPDATE INSTALLMENT SET Amount = Amount + 5000 WHERE loan_no = 'L002';` followed by `UPDATE INSTALLMENT SET Amount = Amount + 5000 WHERE loan_no = 'L003';`. The results table shows the following data:

LOAN_NO	INST_NO	INST_DATE	AMOUNT
L001	001	01/01/2004	10000
L002	002	06/06/2004	20000
L003	003	07/07/2004	20000

3 rows returned in 0.08 seconds.

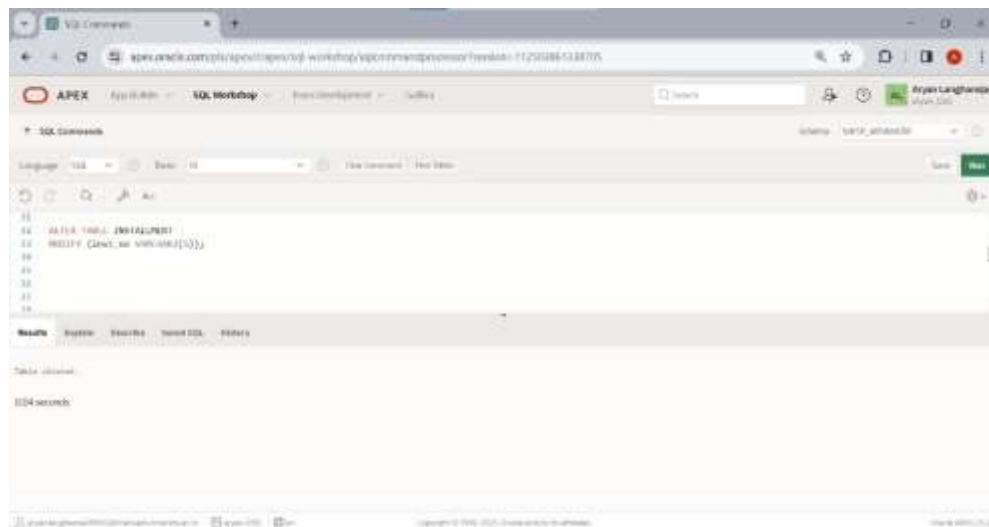
5. Change the column size of 5 to 7 where column name is Loan\_no.

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is: `ALTER TABLE INSTALLMENT MODIFY (Loan_no VARCHAR2(7));`. The results table shows the following data:

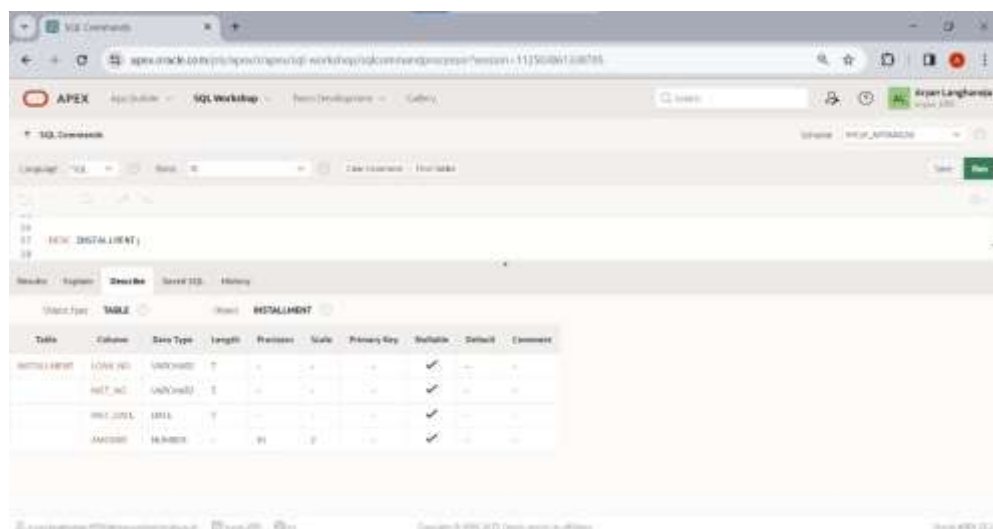
LOAN_NO	INST_NO	INST_DATE	AMOUNT
L001	001	01/01/2004	10000
L002	002	06/06/2004	20000
L003	003	07/07/2004	20000

3 rows returned in 0.08 seconds.

6. Decrease the column size 5 to 4 where column name Inst\_no.



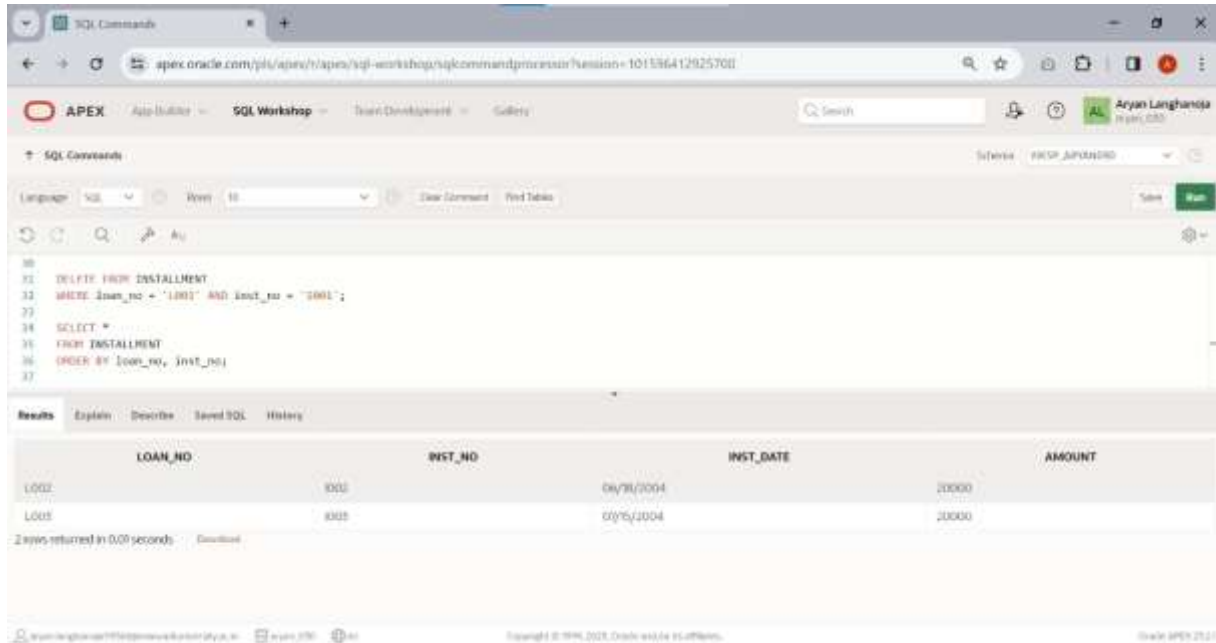
7. Show the structure of the Table.



8. Change the amount 15000 to 5000 where loan number is L001



9. Perform delete operation. (Delete only particular one record)



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

30
31 DELETE FROM INSTALLMENT
32 WHERE loan_no = '1001' AND inst_no = '1001';
33
34 SELECT *
35 FROM INSTALLMENT
36 ORDER BY loan_no, inst_no;
37

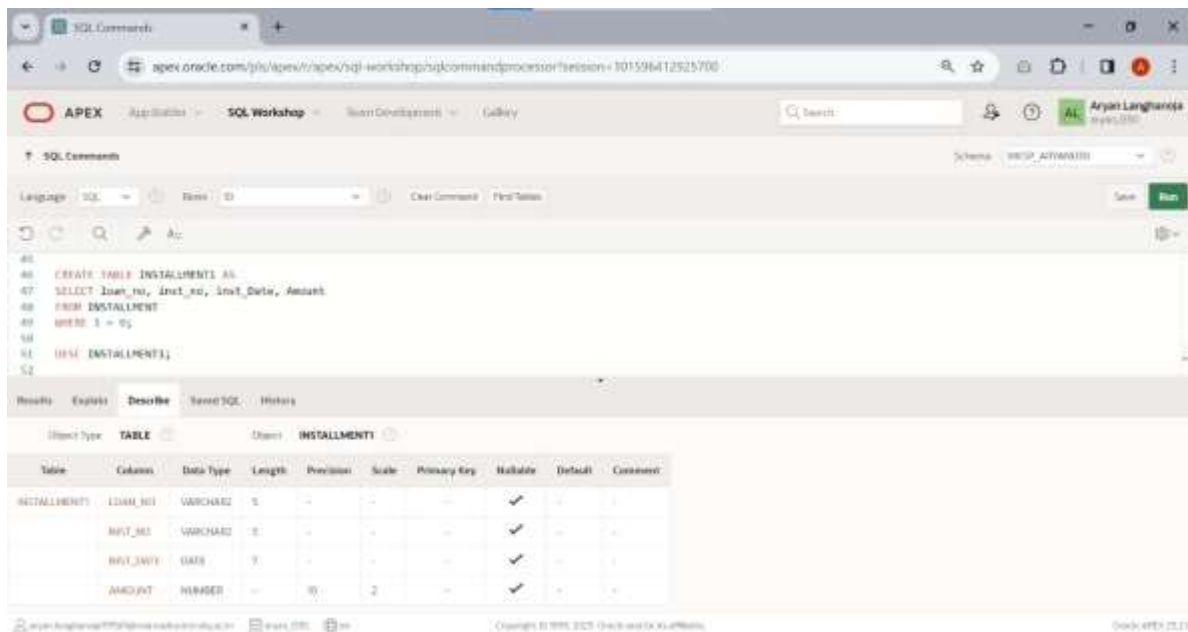
```

The results table shows two records:

LOAN_NO	INST_NO	INST_DATE	AMOUNT
1002	1002	06/01/2004	20000
1003	1003	07/05/2004	20000

2 rows returned in 0.07 seconds.

10. Only create a structure of table installment1 from table installment.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

46 CREATE TABLE INSTALLMENT1 AS
47 SELECT loan_no, inst_no, inst_date, Amount
48 FROM INSTALLMENT
49 WHERE 1 = 0;
50
51 DROP INSTALLMENT1;
52

```

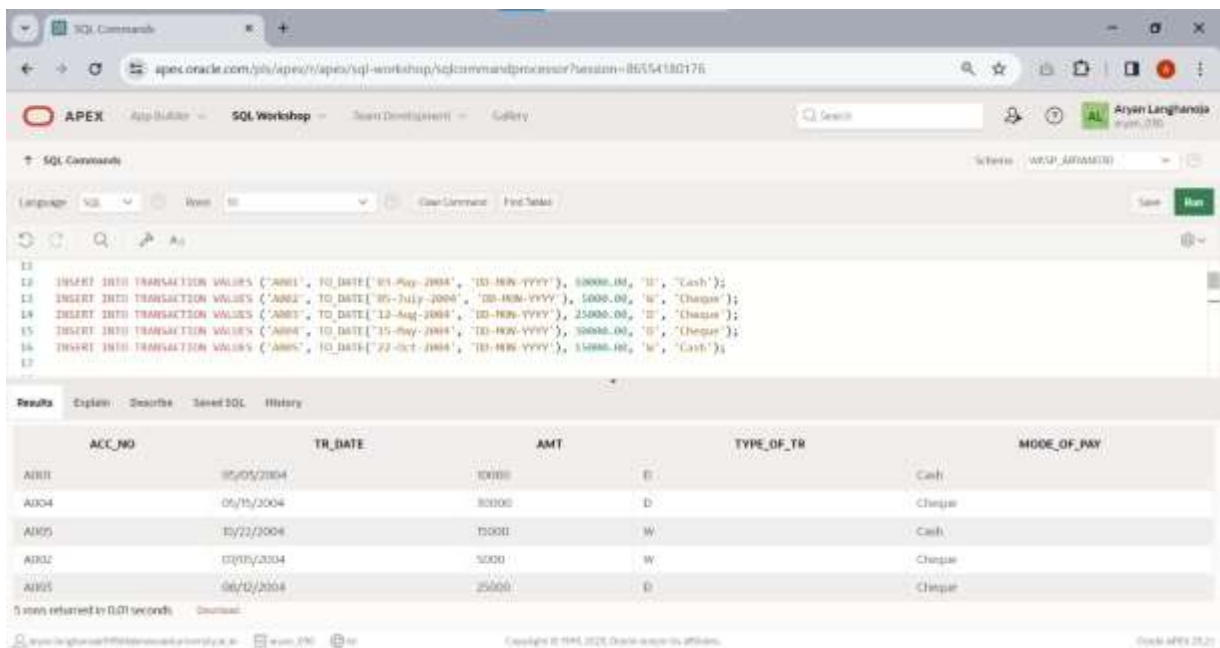
The results table shows the structure of the table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comments
INSTALLMENT1	LOAN_NO	VARCHAR2	5				✓		
	INST_NO	VARCHAR2	5				✓		
	INST_DATE	DATE	9				✓		
	AMOUNT	NUMBER		10	2		✓		

Table: **TRANSACTION**.

1. Insert a Following Records if you have not inserted in PRACTICAL-1.

Acc_no	Trans_Date	Amt	Type_of_tr	Mode_of_pay
A001	3-may-04	10000	D	Cash
A002	5-july-04	5000	W	Check
A003	12-Aug-04	25000	D	Check
A004	15-may-04	30000	D	Check
A005	22-oct-04	15000	W	Cash



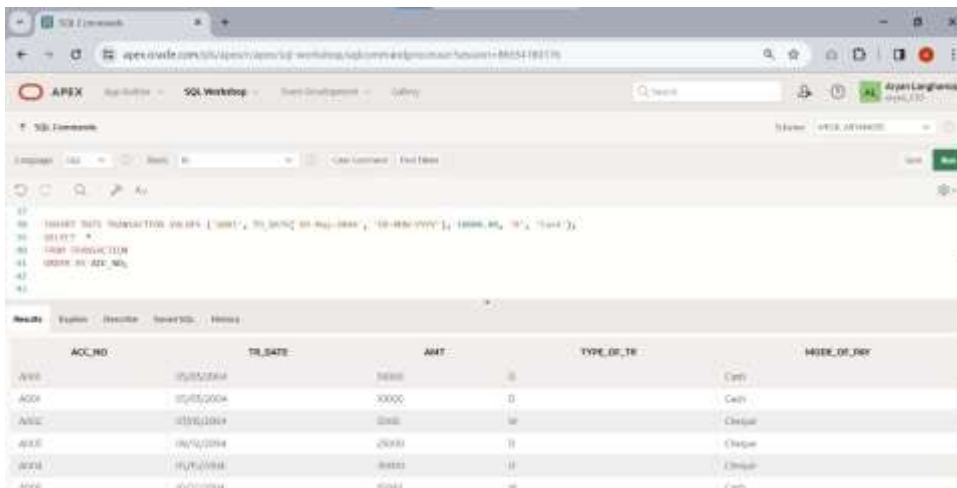
```

12 INSERT INTO TRANSACTION VALUES ('A001', TO_DATE('03-May-2004', 'DD-MON-YYYY'), 10000.00, 'D', 'Cash');
13 INSERT INTO TRANSACTION VALUES ('A002', TO_DATE('05-July-2004', 'DD-MON-YYYY'), 5000.00, 'W', 'Check');
14 INSERT INTO TRANSACTION VALUES ('A003', TO_DATE('12-Aug-2004', 'DD-MON-YYYY'), 25000.00, 'D', 'Check');
15 INSERT INTO TRANSACTION VALUES ('A004', TO_DATE('15-May-2004', 'DD-MON-YYYY'), 30000.00, 'D', 'Check');
16 INSERT INTO TRANSACTION VALUES ('A005', TO_DATE('22-Oct-2004', 'DD-MON-YYYY'), 15000.00, 'W', 'Cash');
17
  
```

ACC_NO	TR_DATE	AMT	TYPE_OF_TR	MODE_OF_PAY
A001	03/05/2004	10000	D	Cash
A004	05/15/2004	30000	D	Check
A005	05/22/2004	15000	W	Cash
A002	07/05/2004	5000	W	Check
A003	08/12/2004	25000	D	Check

5 rows returned in 0.01 seconds

2. Insert any duplicate value and display all the records without any duplicate rows.



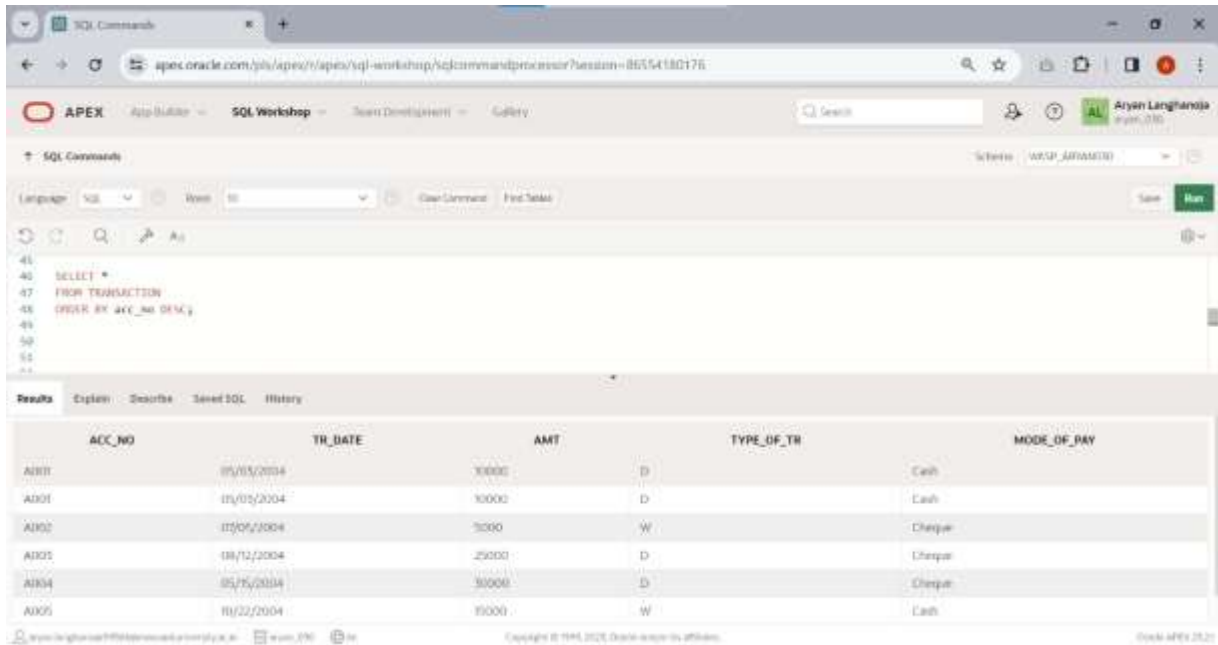
```

17
18 INSERT INTO TRANSACTION VALUES ('A001', TO_DATE('03-May-2004', 'DD-MON-YYYY'), 10000.00, 'D', 'Cash');
19 SELECT *
20 FROM TRANSACTION
21 ORDER BY ACC_NO;
22
  
```

ACC_NO	TR_DATE	AMT	TYPE_OF_TR	MODE_OF_PAY
A001	03/05/2004	10000	D	Cash
A004	05/15/2004	30000	D	Check
A002	07/05/2004	5000	W	Check
A005	05/22/2004	15000	W	Cash
A003	08/12/2004	25000	D	Check
A001	03/05/2004	10000	D	Cash



3. Select all the records in descending order(account number wise).



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

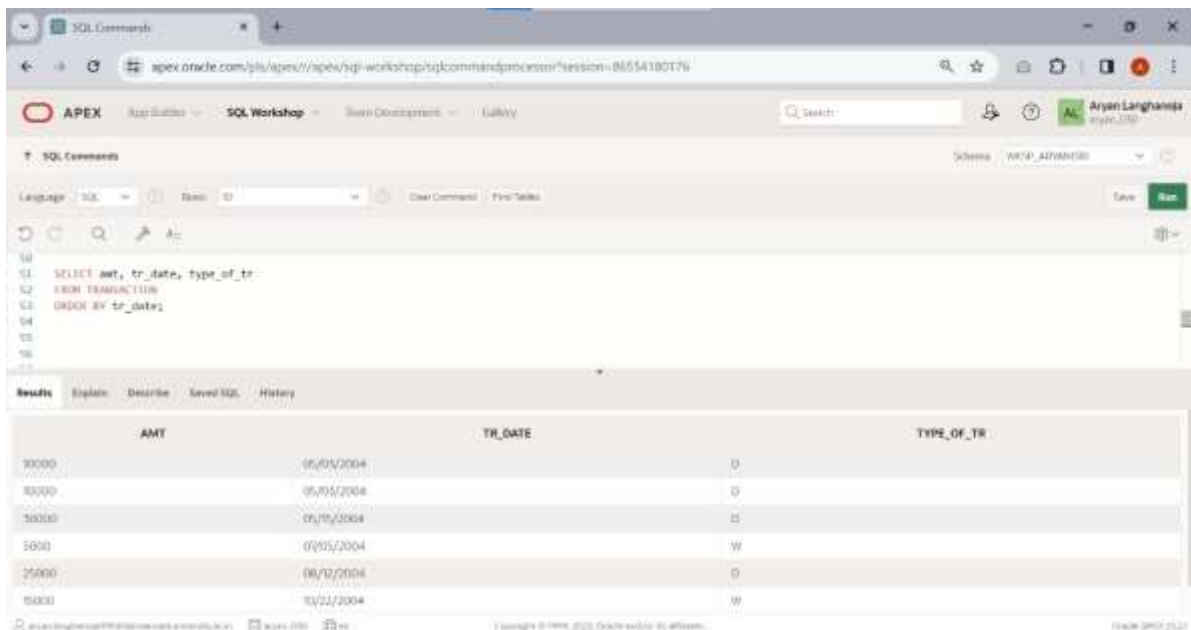
41
42 SELECT *
43 FROM TRANSACTION
44 ORDER BY acc_no DESC;
45
46
47
48
49
50
51
52

```

The results are displayed in a table with the following columns: ACC\_NO, TR\_DATE, AMT, TYPE\_OF\_TR, and MODE\_OF\_PAY.

ACC_NO	TR_DATE	AMT	TYPE_OF_TR	MODE_OF_PAY
A001	05/05/2004	30000	D	Cash
A001	05/05/2004	30000	D	Cash
A002	07/05/2004	5000	W	Cheque
A002	08/10/2004	25000	D	Cheque
A004	05/15/2004	30000	D	Cheque
A005	10/22/2004	5000	W	Cash

4. Display amt, date, and type of transaction by date wise.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

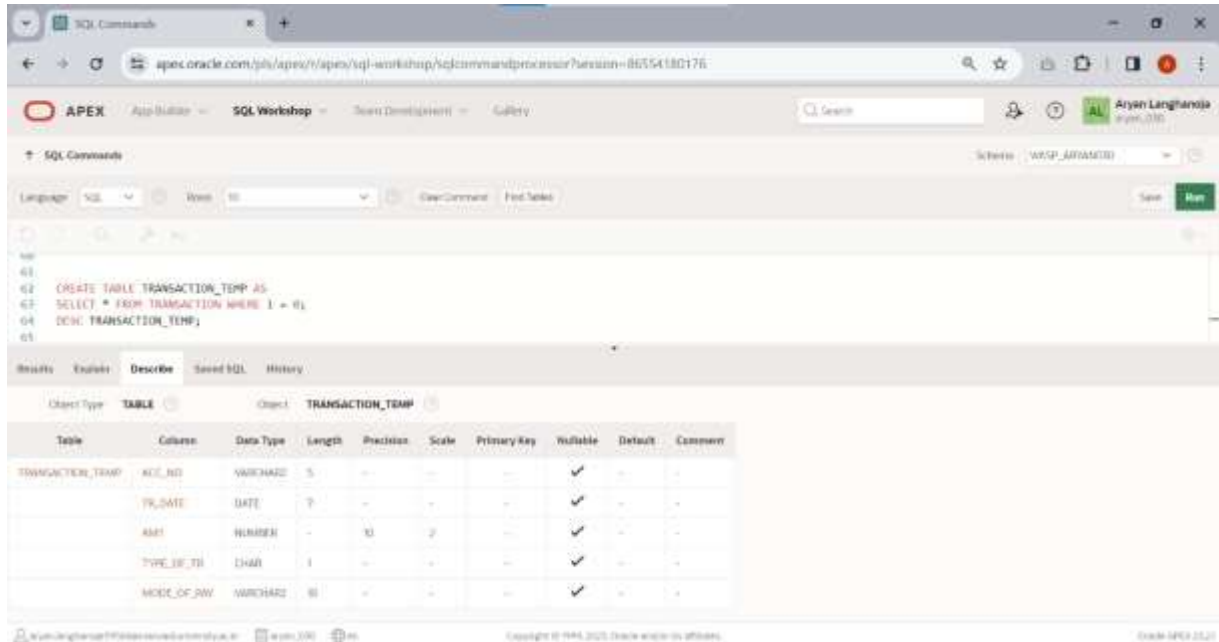
50
51 SELECT amt, tr_date, type_of_tr
52 FROM TRANSACTION
53 ORDER BY tr_date;
54
55
56
57
58
59
60
61

```

The results are displayed in a table with the following columns: AMT, TR\_DATE, and TYPE\_OF\_TR.

AMT	TR_DATE	TYPE_OF_TR
30000	05/05/2004	D
30000	05/05/2004	D
30000	05/15/2004	D
5000	07/05/2004	W
25000	08/10/2004	D
5000	10/22/2004	W

5. Create another table TRANSACTION\_TEMP from this table.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following code:

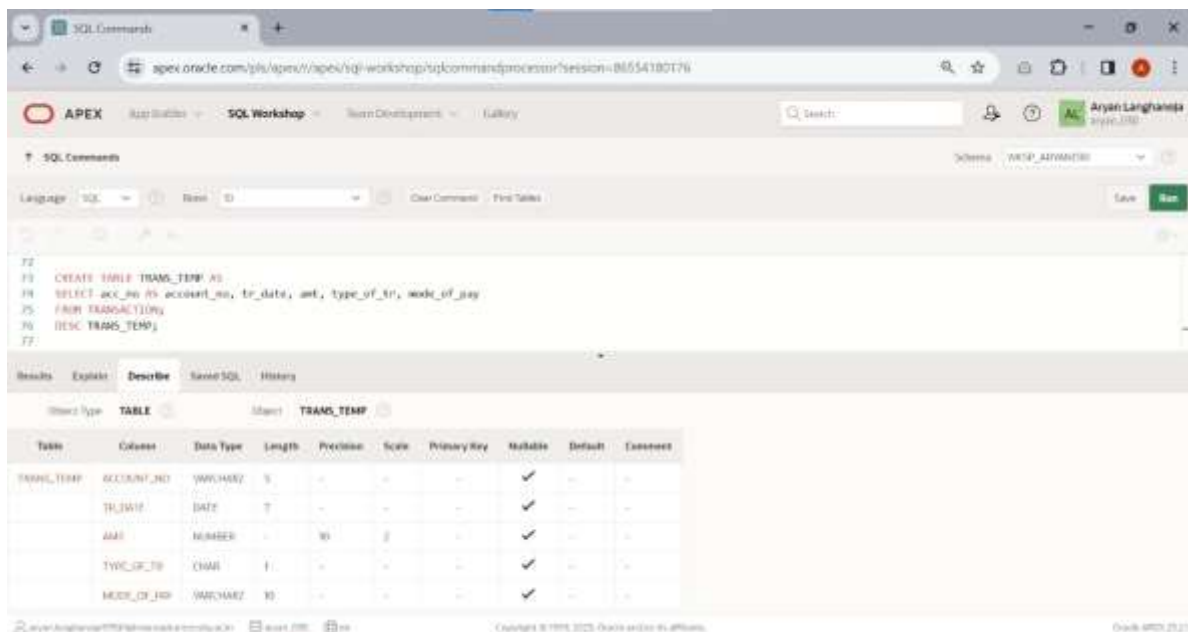
```

61
62 CREATE TABLE TRANSACTION_TEMP AS
63 SELECT * FROM TRANSACTION WHERE 1 = 0;
64 DESC TRANSACTION_TEMP;
65
  
```

The 'Describe' tab is selected, showing the structure of the TRANSACTION\_TEMP table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANSACTION_TEMP	ACC_NO	VARCHAR2	5				✓		
	TR_DATE	DATE	7				✓		
	AMT	NUMBER		10	2		✓		
	TYPE_OF_TR	CHAR	1				✓		
	MODE_OF_PAY	VARCHAR2	10				✓		

6. Create a another table TRANS\_TEMP by change the column name acc\_no to account\_no.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following code:

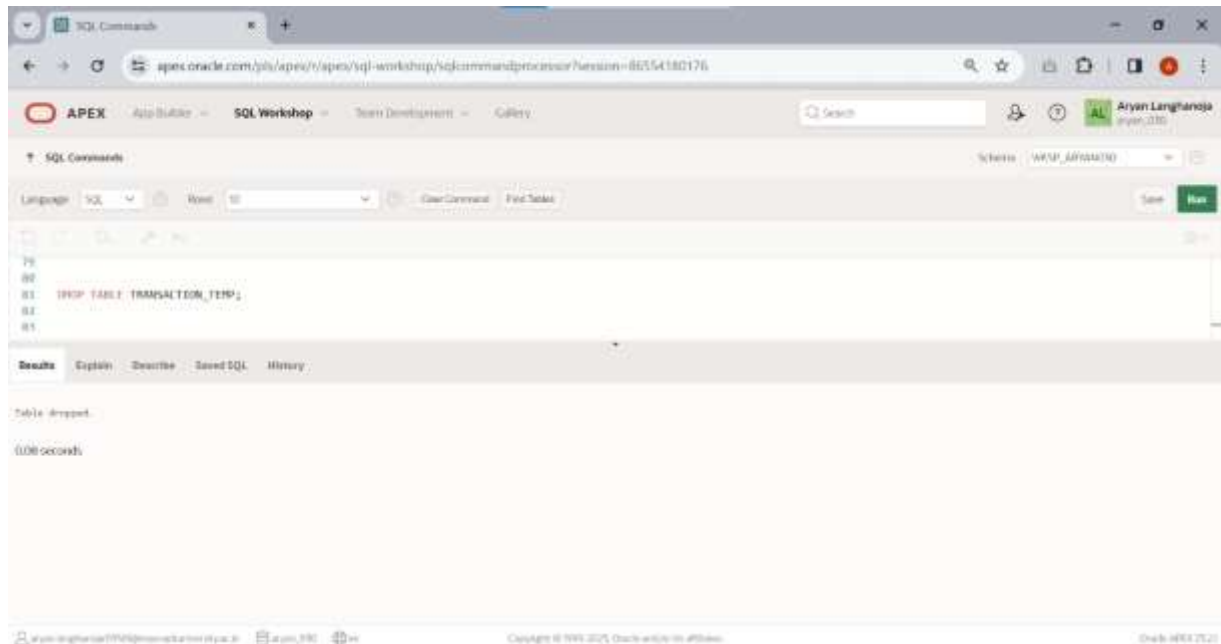
```

72
73 CREATE TABLE TRANS_TEMP AS
74 SELECT acc_no as account_no, tr_date, amt, type_of_tr, mode_of_pay
75 FROM TRANSACTION;
76 DESC TRANS_TEMP;
77
  
```

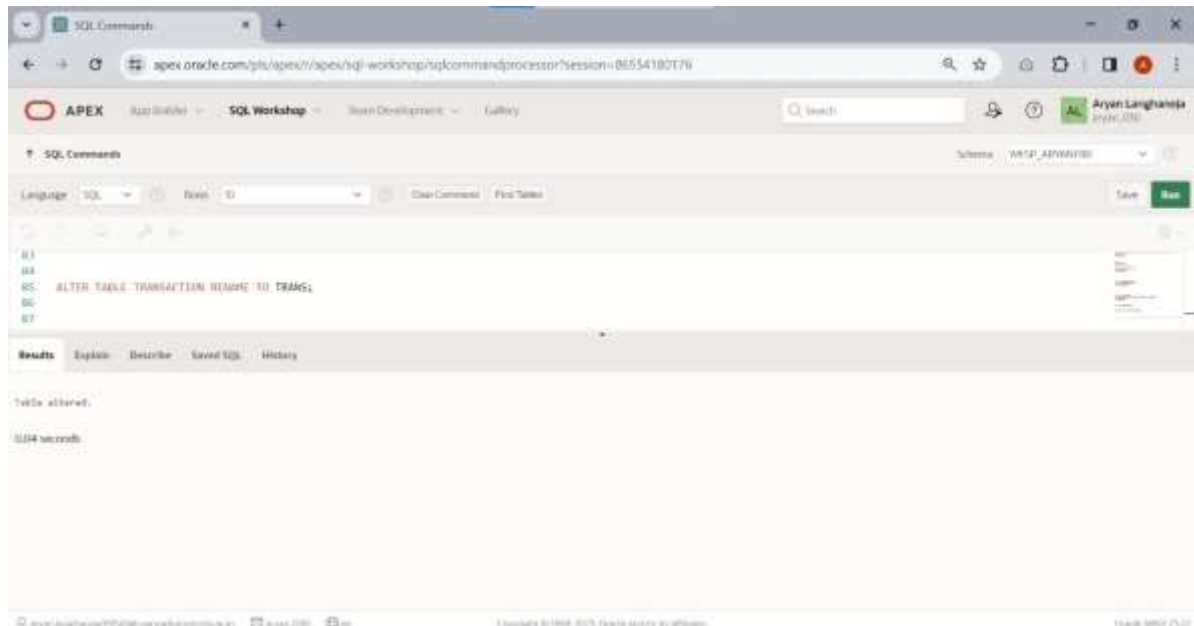
The 'Describe' tab is selected, showing the structure of the TRANS\_TEMP table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANS_TEMP	ACCOUNT_NO	VARCHAR2	5				✓		
	TR_DATE	DATE	7				✓		
	AMT	NUMBER		10	2		✓		
	TYPE_OF_TR	CHAR	1				✓		
	MODE_OF_PAY	VARCHAR2	10				✓		

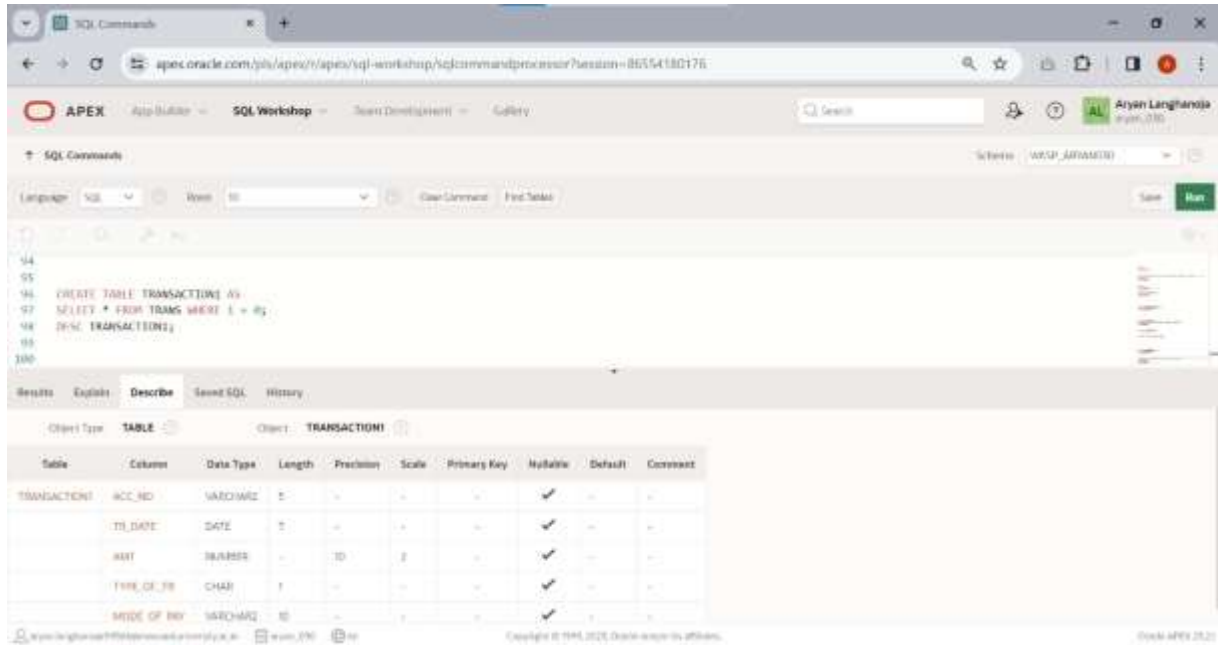
7. Delete a table TRANSACTION\_TEMP.



8. Rename the table TRANSACTION to TRANS.



9. Only create a structure of table transaction1 from table transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following code:

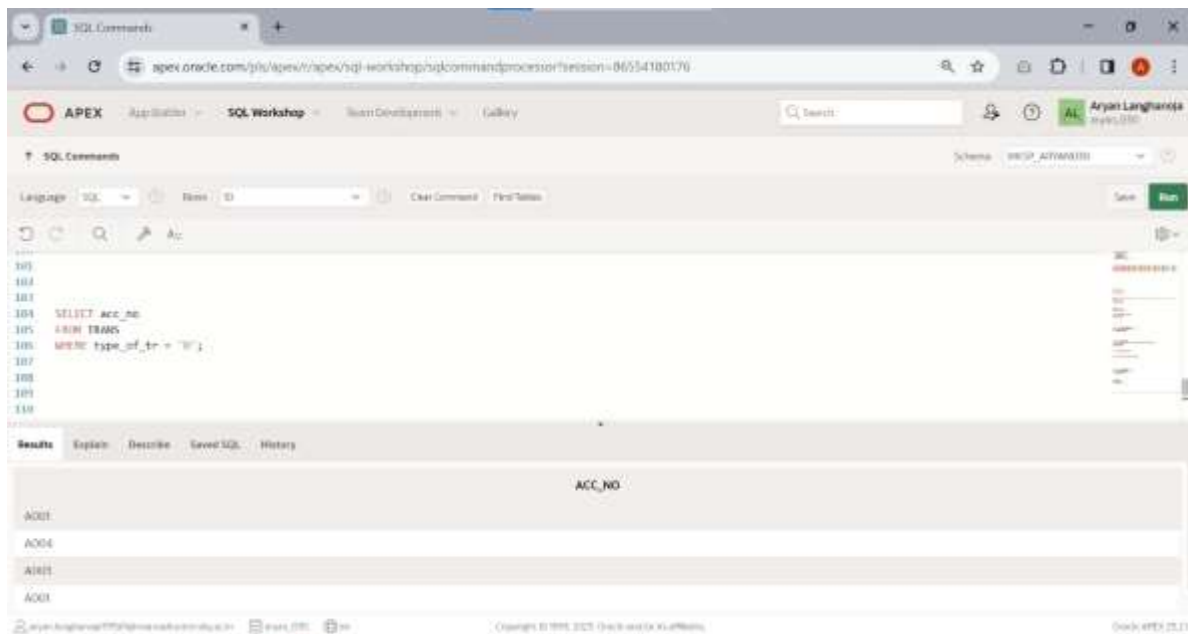
```

154
155
156 CREATE TABLE TRANSACTION1 AS
157 SELECT * FROM TRANS WHERE 1 = 0;
158 DESC TRANSACTION1;
159
160
  
```

The 'Describe' tab is selected, showing the structure of the 'TRANSACTION1' table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRANSACTION1	ACC_NO	VARCHAR2	5				✓		
	TR_DATE	DATE					✓		
	AMT	NUMBER		10	2		✓		
	TYPE_OF_TR	CHAR	1				✓		
	MODE_OF_PAY	VARCHAR2	10				✓		

10. Display account number where type of transaction is 'D'.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following code:

```

161
162
163
164
165 SELECT Acc_no
166 FROM TRANS
167 WHERE type_of_tr = 'D';
168
169
170
  
```

The 'Results' tab is selected, showing the output of the query:

ACC_NO
A001
A004
A001
A001

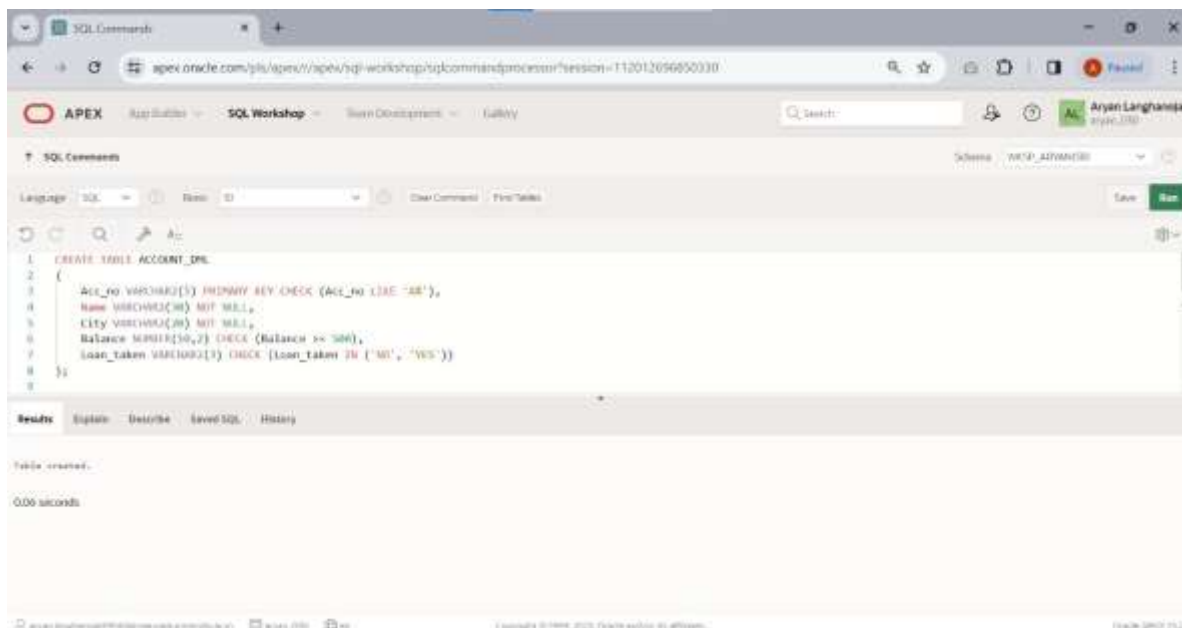
## Practical 5

### Aim: Constraint Based DML Commands

**Note:** Bold and Underline column name indicates a primary key

Create a table **ACCOUNT**.

Column name	Data Type	Size	Attributes
<u>Acc_no</u>	Varchar2	5	Primary key/first letter must start with 'A'
Name	Varchar2	30	NOT NULL
City	Varchar2	20	NOT NULL
Balance	Number	10,2	Balance >=500
Loan_taken	Varchar2	3	Values('NO', 'YES')



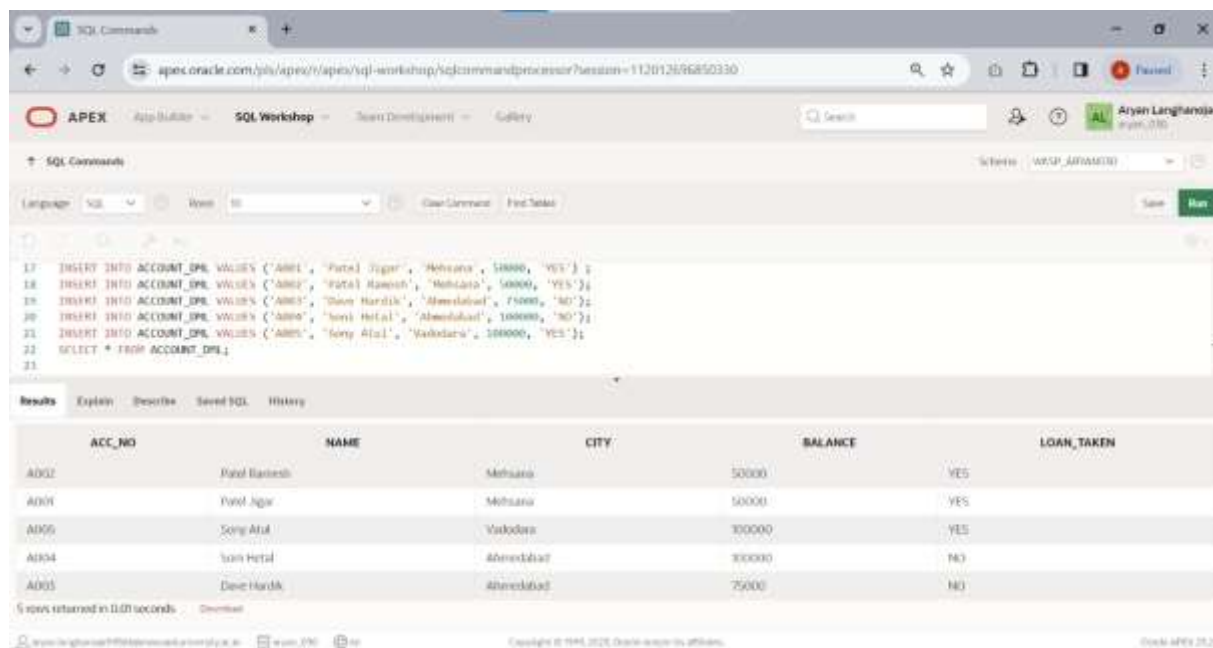
```

1 CREATE TABLE ACCOUNT_DML
2 (
3     Acc_no VARCHAR2(5) PRIMARY KEY CHECK (Acc_no LIKE 'A%'),
4     Name VARCHAR2(30) NOT NULL,
5     City VARCHAR2(20) NOT NULL,
6     Balance NUMBER(10,2) CHECK (Balance >= 500),
7     Loan_taken VARCHAR2(3) CHECK (Loan_taken IN ('NO', 'YES'))
8 );

```

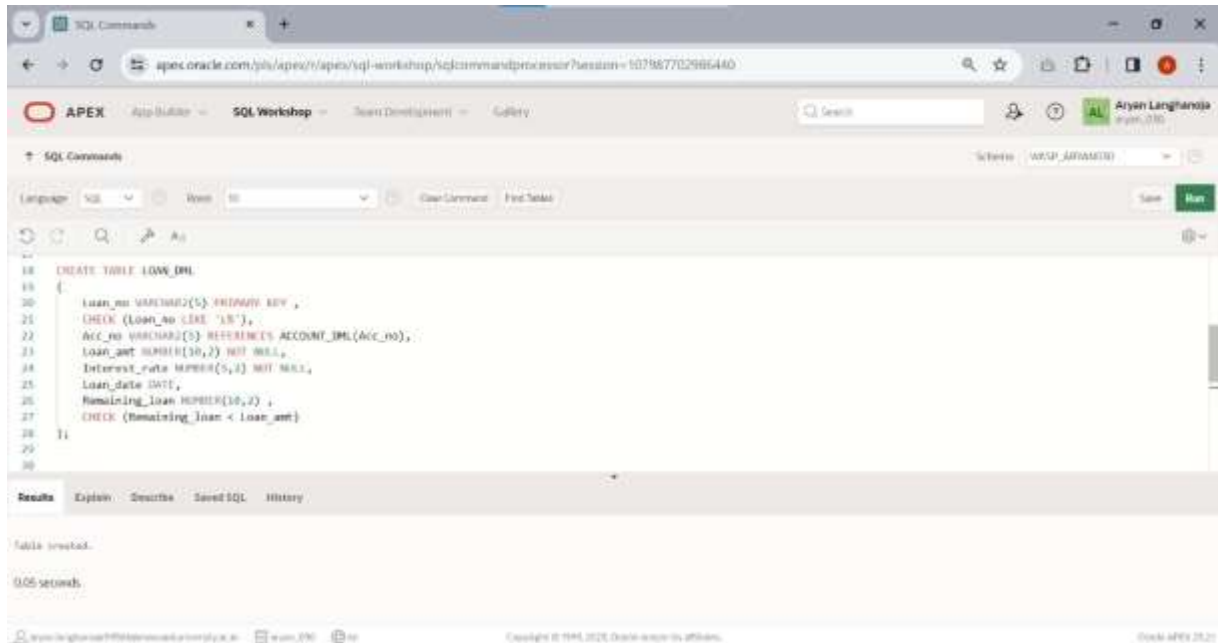
Results: Table created. 0.06 seconds

1. Insert the records using Practical list 1.



Create a Table **LOAN**.

Column Name	Data Type	Size	Attributes
<u>Loan_no</u>	Varchar 2	5	Primary Key / first letter must start with 'L'
Acc_no	Varchar 2	5	Foreign key References Acc_no of account table
Loan_amt	Number	10, 2	NOT NULL
Interest_rate	Number	5, 2	NOT NULL
Loan_date	Date		
Remaining_loan	Number	10, 2	Remaining loan < loan amount



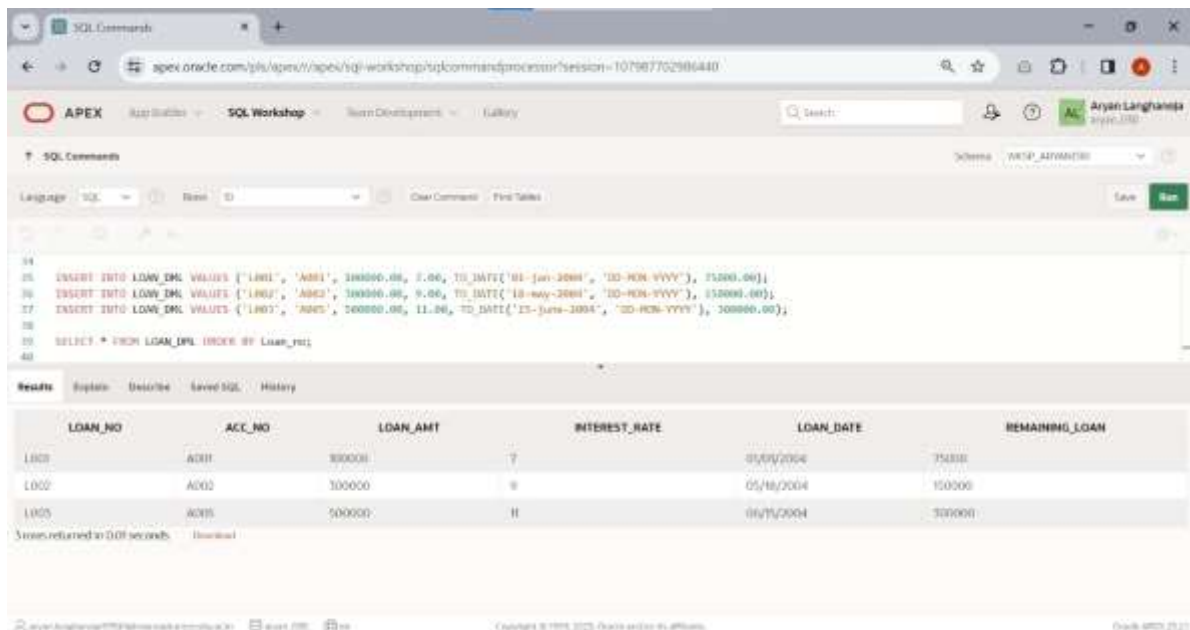
The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following code:

```

18 CREATE TABLE LOAN_DML
19 (
20   loan_no VARCHAR2(5) PRIMARY KEY,
21   CHECK (loan_no LIKE '1%'),
22   acc_no VARCHAR2(5) REFERENCES ACCOUNT_DML(acc_no),
23   loan_amt NUMBER(10,2) NOT NULL,
24   interest_rate NUMBER(5,2) NOT NULL,
25   loan_date DATE,
26   remaining_loan NUMBER(10,2),
27   CHECK (remaining_loan < loan_amt)
28 );
29
30
  
```

Below the code, the 'Results' tab shows the message: 'Table created.' and '0.05 seconds'.

1. Insert the records using practical list-1.



The screenshot shows the APEX SQL Workshop interface with the following SQL commands:

```

34
35 INSERT INTO LOAN_DML VALUES ('1001', 'A001', 300000.00, 7.00, TO_DATE('01-jan-2004', 'DD-MON-YYYY'), 75000.00);
36 INSERT INTO LOAN_DML VALUES ('1002', 'A002', 100000.00, 9.00, TO_DATE('10-may-2004', 'DD-MON-YYYY'), 150000.00);
37 INSERT INTO LOAN_DML VALUES ('1003', 'A003', 500000.00, 11.00, TO_DATE('25-june-2004', 'DD-MON-YYYY'), 300000.00);
38
39 SELECT * FROM LOAN_DML ORDER BY loan_no;
40
  
```

The 'Results' tab displays the following table:

LOAN_NO	ACC_NO	LOAN_AMT	INTEREST_RATE	LOAN_DATE	REMAINING_LOAN
1001	A001	300000	7	01/01/2004	75000
1002	A002	100000	9	05/10/2004	150000
1003	A003	500000	11	06/25/2004	300000

Below the table, it states: '3 rows returned in 0.01 seconds.' and provides a 'Download' link.

Create a table **INSTALLMENT**.

Column Name	Data Type	Size	Attributes
<u>Loan_no</u>	Varchar2	5	Foreign key References Loan_no of Loan table
Inst_no	Varchar2	5	first letter must start with 'I'
IDate	Date		NOT NULL
Amount	Number	10,2	NOT NULL

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

21
22 CREATE TABLE INSTALLMENT_DML (
23   loan_no VARCHAR(5) PRIMARY KEY REFERENCES LOAN_DML(Loan_no) ,
24   Inst_no VARCHAR(5) CHECK(Inst_no LIKE 'I%') ,
25   IDate DATE NOT NULL ,
26   Amount NUMBER(10,2) NOT NULL
27 )
28

```

The results section shows: "Table created." and "0.04 seconds".

1. Insert the records using Practical list-1

The screenshot shows the APEX SQL Workshop interface with the following SQL commands:

```

87
88 INSERT INTO INSTALLMENT_DML VALUES ('1001', '1001', TO_DATE('2-Feb-2004', 'DD-MON-YYYY'), 10000.00);
89 INSERT INTO INSTALLMENT_DML VALUES ('1002', '1002', TO_DATE('24-June-2004', 'DD-MON-YYYY'), 20000.00);
90 INSERT INTO INSTALLMENT_DML VALUES ('1003', '1003', TO_DATE('24-July-2004', 'DD-MON-YYYY'), 30000.00);
91
92 SELECT * FROM INSTALLMENT_DML ORDER BY loan_no;
93

```

The results section shows a table with the following data:

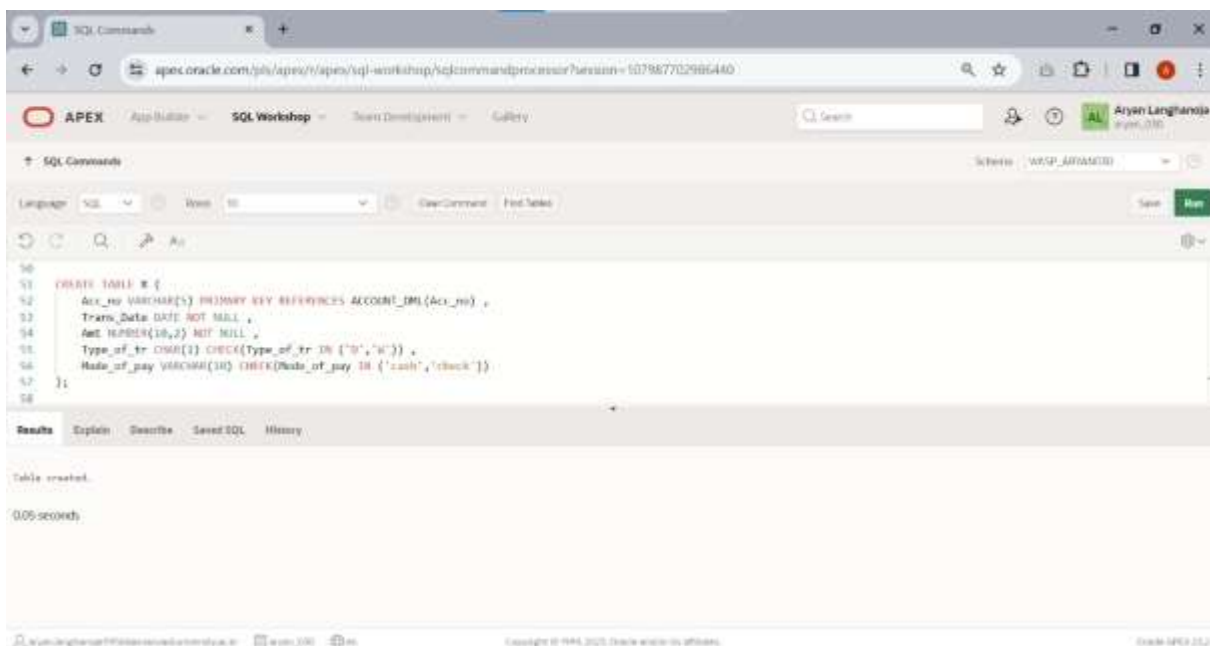
LOAN_NO	INST_NO	IDATE	AMOUNT
1001	1001	02/02/2004	10000
1002	1002	24/06/2004	20000
1003	1003	24/07/2004	30000

3 rows returned in 0.11 seconds.



Create a Table X .

Column Name	Data Type	Size	Attributes
<u>Acc_no</u>	Varchar2	5	Foreign key References Acc_no of account table
Trans_Date	Date		NOT NULL
Amt	Number	10,2	NOT NULL
Type_of_tr	Char	1	Values in ('D','W')
Mode_of_pay	Varchar2	10	Values in ('cash','check')



```

50
51 CREATE TABLE X {
52   Acc_no VARCHAR(5) PRIMARY KEY REFERENCES ACCOUNT_BMI(Acc_no),
53   Trans_Date DATE NOT NULL,
54   Amt NUMBER(10,2) NOT NULL,
55   Type_of_tr CHAR(1) CHECK(Type_of_tr IN ('D','W')),
56   Mode_of_pay VARCHAR(10) CHECK(Mode_of_pay IN ('cash','check'))
57 }
58

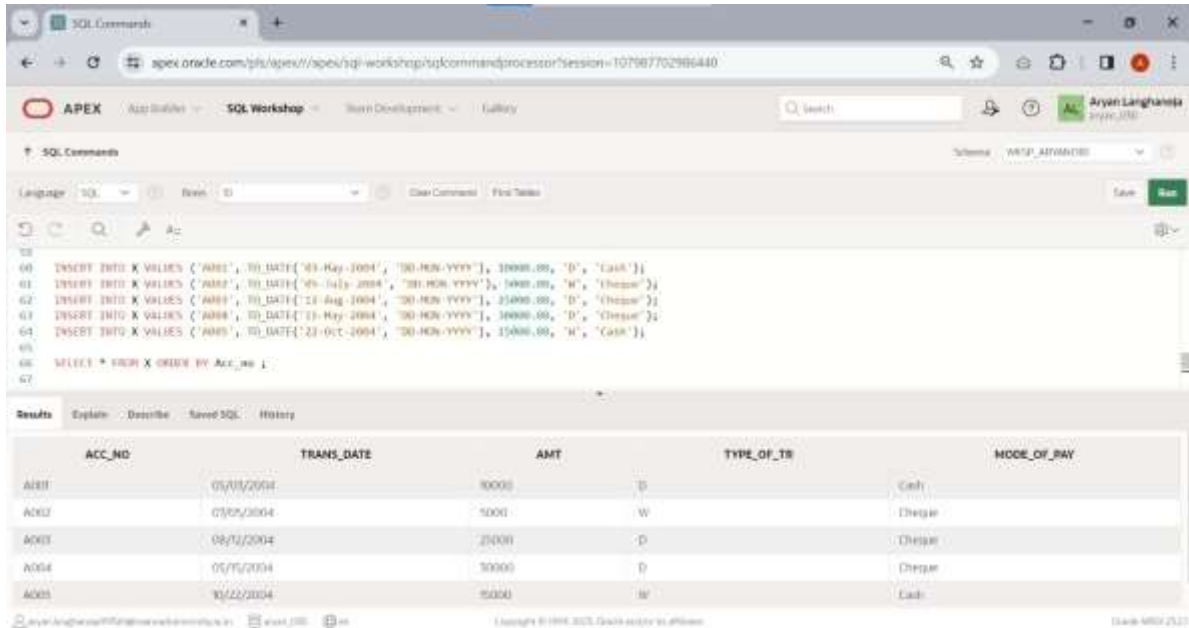
```

Results Explain Describe Saved SQL History

Table created.

0.05 seconds

1. Insert the records using Practical list-1.



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Share Development', and 'Gallery'. The user is logged in as 'Aryen Singhania'.

The 'SQL Commands' section shows the following SQL script:

```

60 INSERT INTO X VALUES ('A001', TO_DATE('01-May-2004', 'DD-MON-YYYY'), 10000.00, 'D', 'Cash');
61 INSERT INTO X VALUES ('A002', TO_DATE('05-Jul-2004', 'DD-MON-YYYY'), 5000.00, 'W', 'Cheque');
62 INSERT INTO X VALUES ('A003', TO_DATE('14-Aug-2004', 'DD-MON-YYYY'), 25000.00, 'D', 'Cheque');
63 INSERT INTO X VALUES ('A004', TO_DATE('19-May-2004', 'DD-MON-YYYY'), 30000.00, 'D', 'Cheque');
64 INSERT INTO X VALUES ('A005', TO_DATE('23-Oct-2004', 'DD-MON-YYYY'), 15000.00, 'W', 'Cash');
65
66 SELECT * FROM X ORDER BY Acc_no;
67
  
```

The 'Results' section displays the output of the SQL query, showing a table with the following columns: ACC\_NO, TRANS\_DATE, AMT, TYPE\_OF\_TB, and MODE\_OF\_PAY.

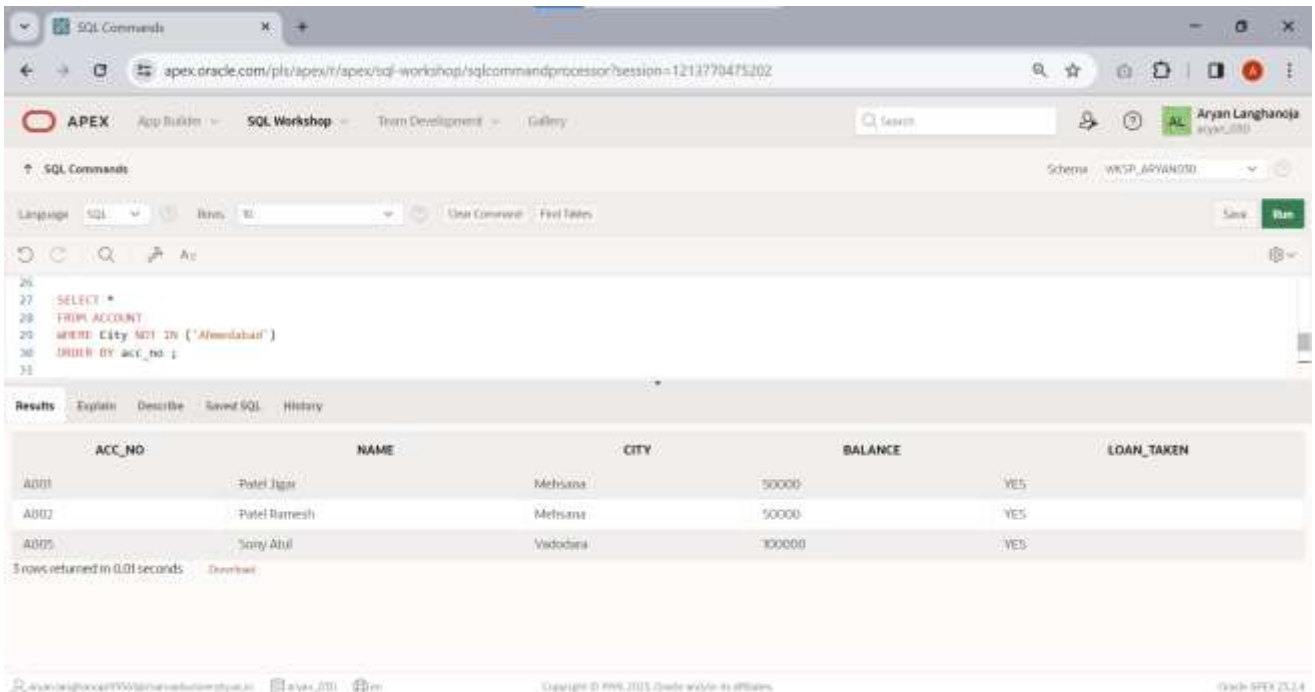
ACC_NO	TRANS_DATE	AMT	TYPE_OF_TB	MODE_OF_PAY
A001	01/05/2004	10000	D	Cash
A002	05/07/2004	5000	W	Cheque
A003	08/08/2004	25000	D	Cheque
A004	05/05/2004	30000	D	Cheque
A005	10/22/2004	15000	W	Cash

## Practical 6

### Aim: Functions and Queries

#### Using Operator: NOT,BETWEEN,NOT BETWEEN,IN,NOT IN

1. Retrieve specified information for the account holder who are not in 'Ahmedabad'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

26
27 SELECT *
28 FROM ACCOUNT
29 WHERE City NOT IN ('Ahmedabad')
30 ORDER BY acc_no ;
31

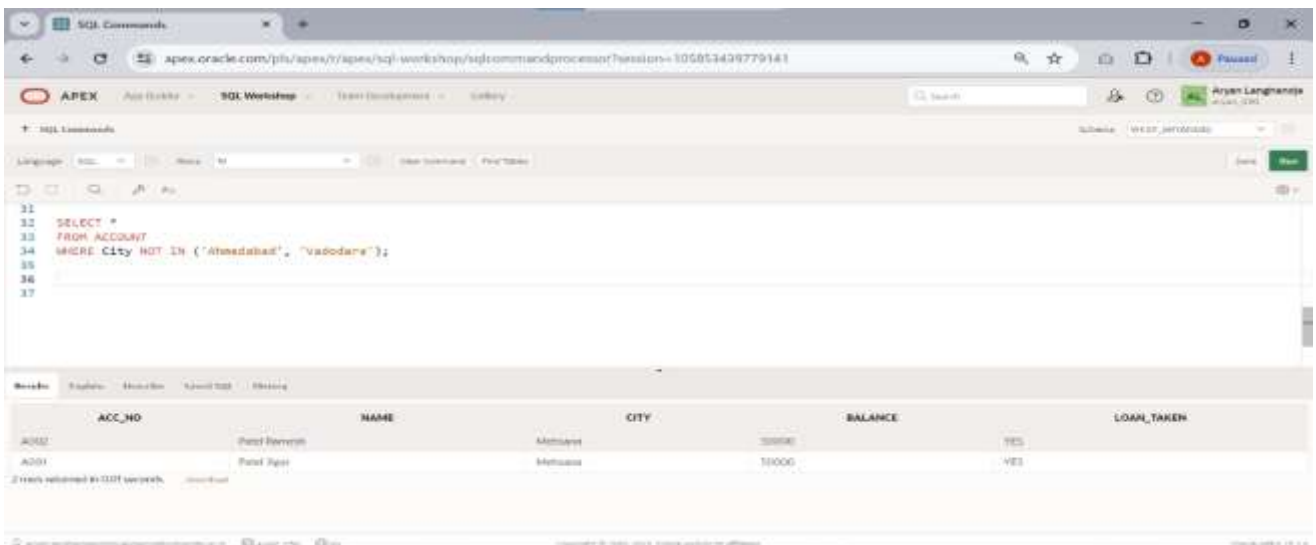
```

The results are displayed in a table with the following columns: ACC\_NO, NAME, CITY, BALANCE, and LOAN\_TAKEN.

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN
A001	Patel Jign	Mehsana	50000	YES
A002	Patel Ramesh	Mehsana	50000	YES
A005	Sony Ahil	Vadodara	100000	YES

3 rows returned in 0.01 seconds.

2. Retrieve specified information for the account holder who are not in 'Ahmedabad' or 'Vadodara'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

31
32 SELECT *
33 FROM ACCOUNT
34 WHERE City NOT IN ('Ahmedabad', 'Vadodara');
35
36
37

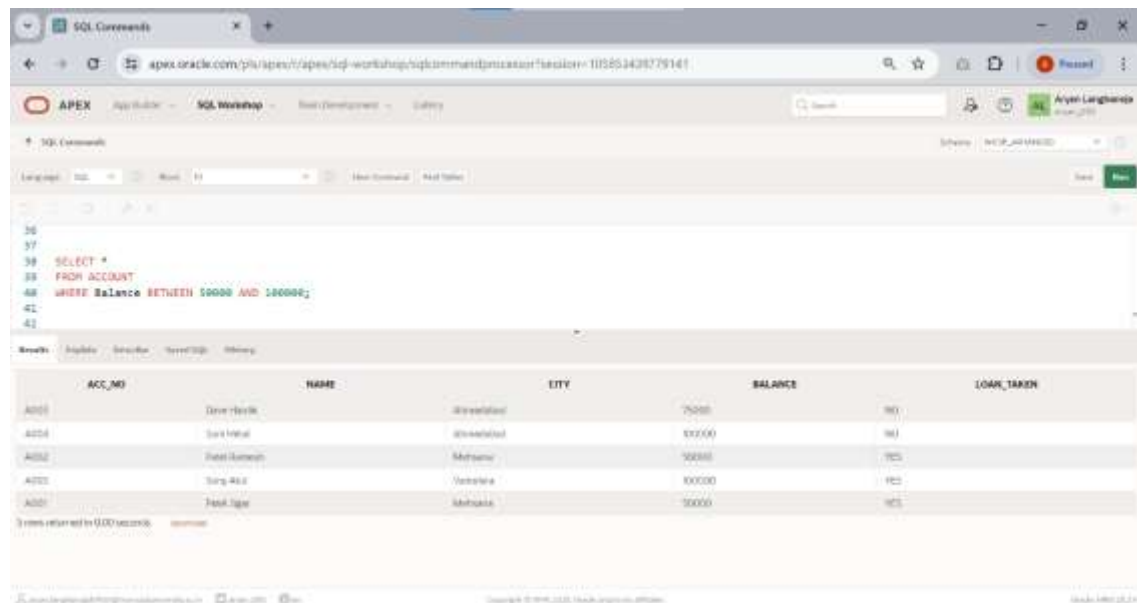
```

The results are displayed in a table with the following columns: ACC\_NO, NAME, CITY, BALANCE, and LOAN\_TAKEN.

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN
A002	Patel Ramesh	Mehsana	50000	YES
A001	Patel Jign	Mehsana	50000	YES

2 rows returned in 0.01 seconds.

- Retrieve those records of Account holder whose balance between is 50000 and 100000.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

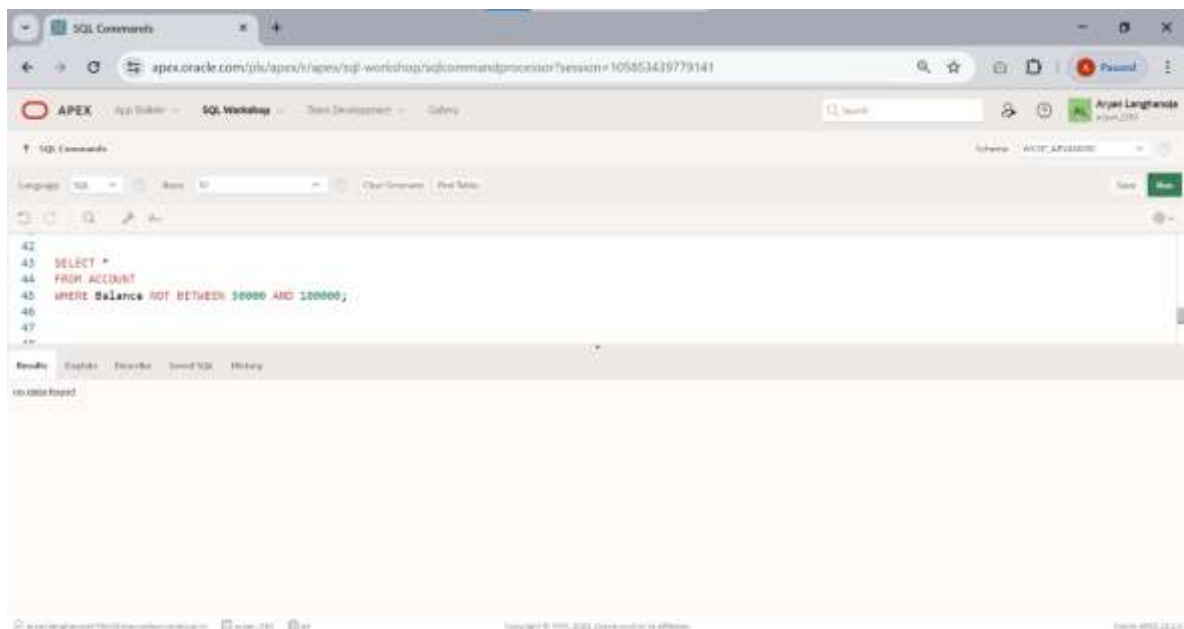
36
37
38 SELECT *
39 FROM ACCOUNT
40 WHERE Balance BETWEEN 50000 AND 100000;
41
42
  
```

The results are displayed in a table with the following columns: ACC\_NO, NAME, CITY, BALANCE, and LOAN TAKEN. The table contains 5 rows of data.

ACC_NO	NAME	CITY	BALANCE	LOAN TAKEN
A001	Deep Rishi	Chennai	75000	90
A004	Sara Patel	Chennai	100000	90
A002	Ravi Kumar	Mumbai	50000	90
A003	Singh A.K.	Chennai	100000	90
A005	Ravi Singh	Mumbai	50000	90

3 rows returned in 0.00 seconds.

- Retrieve those records of Account holder whose balance not between is 50000 and 100000.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

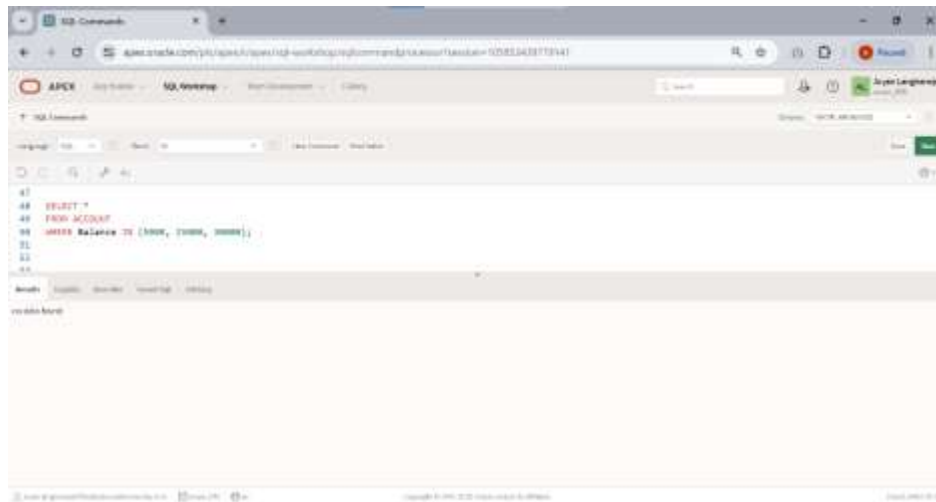
42
43 SELECT *
44 FROM ACCOUNT
45 WHERE Balance NOT BETWEEN 50000 AND 100000;
46
47
  
```

The results are displayed in a table with the following columns: ACC\_NO, NAME, CITY, BALANCE, and LOAN TAKEN. The table contains 5 rows of data.

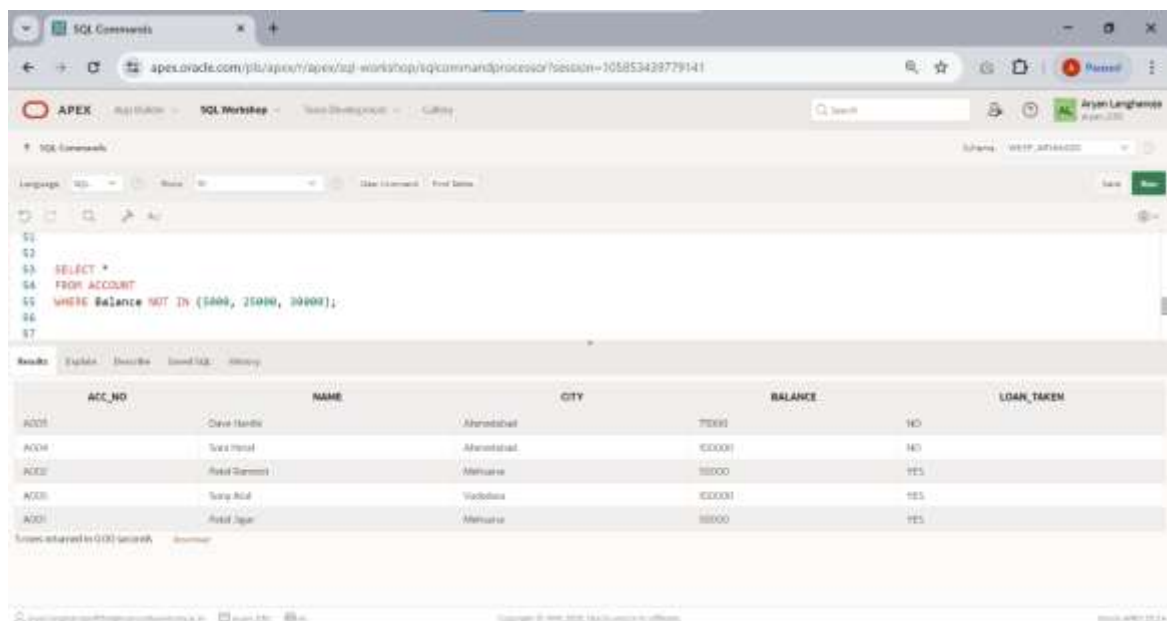
ACC_NO	NAME	CITY	BALANCE	LOAN TAKEN
A001	Deep Rishi	Chennai	75000	90
A004	Sara Patel	Chennai	100000	90
A002	Ravi Kumar	Mumbai	50000	90
A003	Singh A.K.	Chennai	100000	90
A005	Ravi Singh	Mumbai	50000	90

5 rows returned in 0.00 seconds.

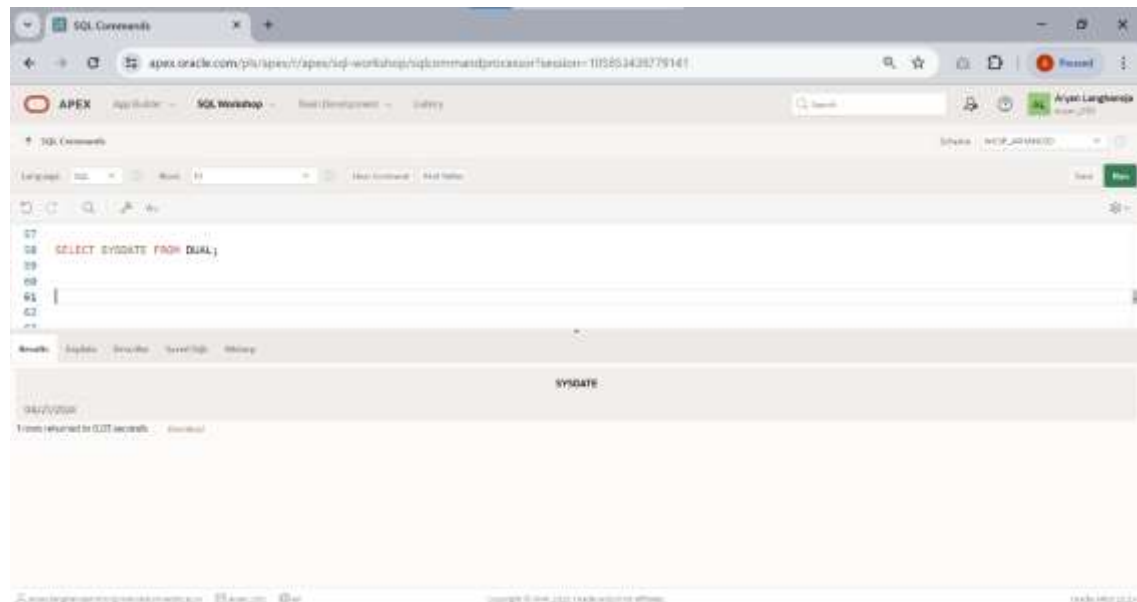
5. Display only those records whose amount is 5000, 25000, 30000.



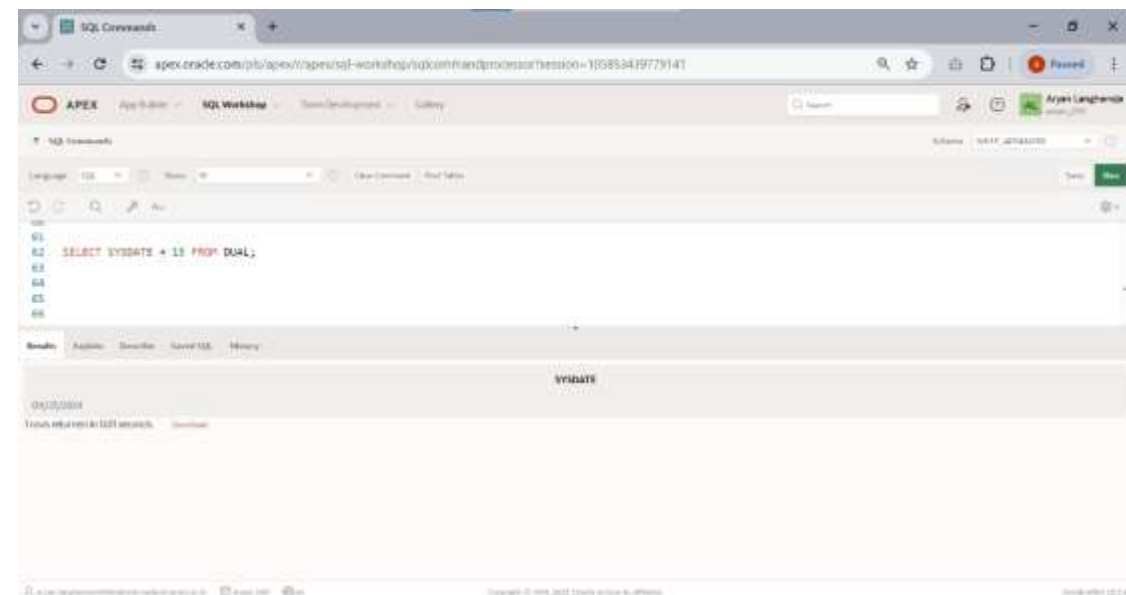
6. Display only those records whose amount not in 5000, 25000, 30000.



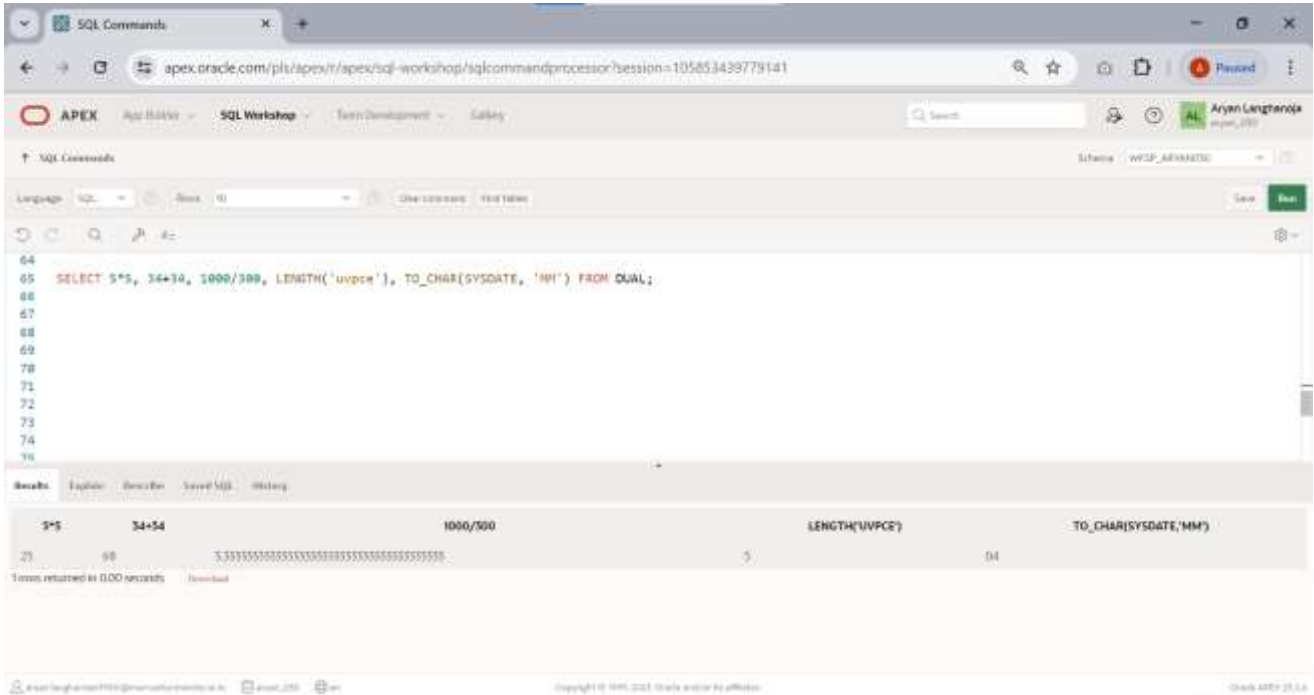
7. Display System date.



8. Find the date,15 days after today's date.



9. Perform following operation using DUAL table.  
 $5*5, 34+34, 1000/300$ , length of 'uvpce', display only month of systemdate



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

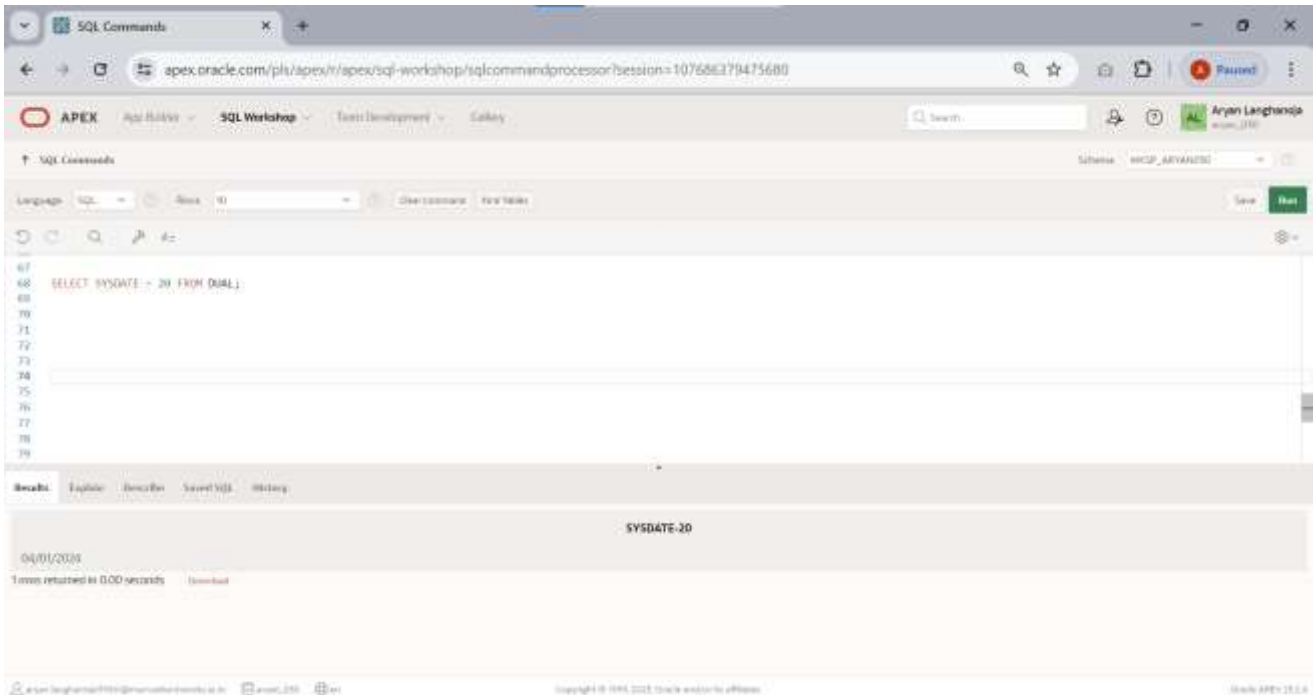
```
SELECT 5*5, 34+34, 1000/300, LENGTH('uvpcr'), TO_CHAR(SYSDATE, 'MM') FROM DUAL;
```

The results table displays the following data:

5*5	34+34	1000/300	LENGTH('UVPCR')	TO_CHAR(SYSDATE,'MM')
25	68	3.3333333333333333	5	04

Times returned in 0.00 seconds

10. Find the date, 20 days before today's date.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
SELECT SYSDATE - 20 FROM DUAL;
```

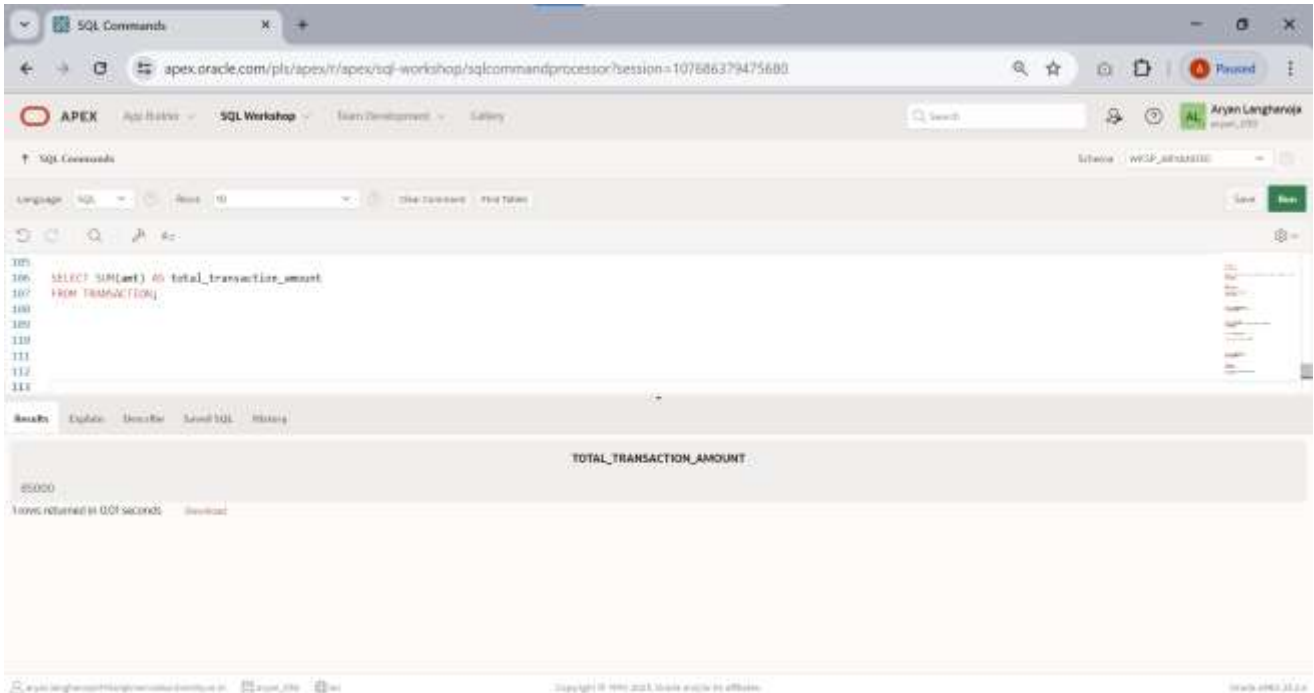
The results table displays the following data:

SYSDATE-20
04/01/2024

Times returned in 0.00 seconds

## Function Based Queries.

1. Find the total transaction amount of account holder from transaction table.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

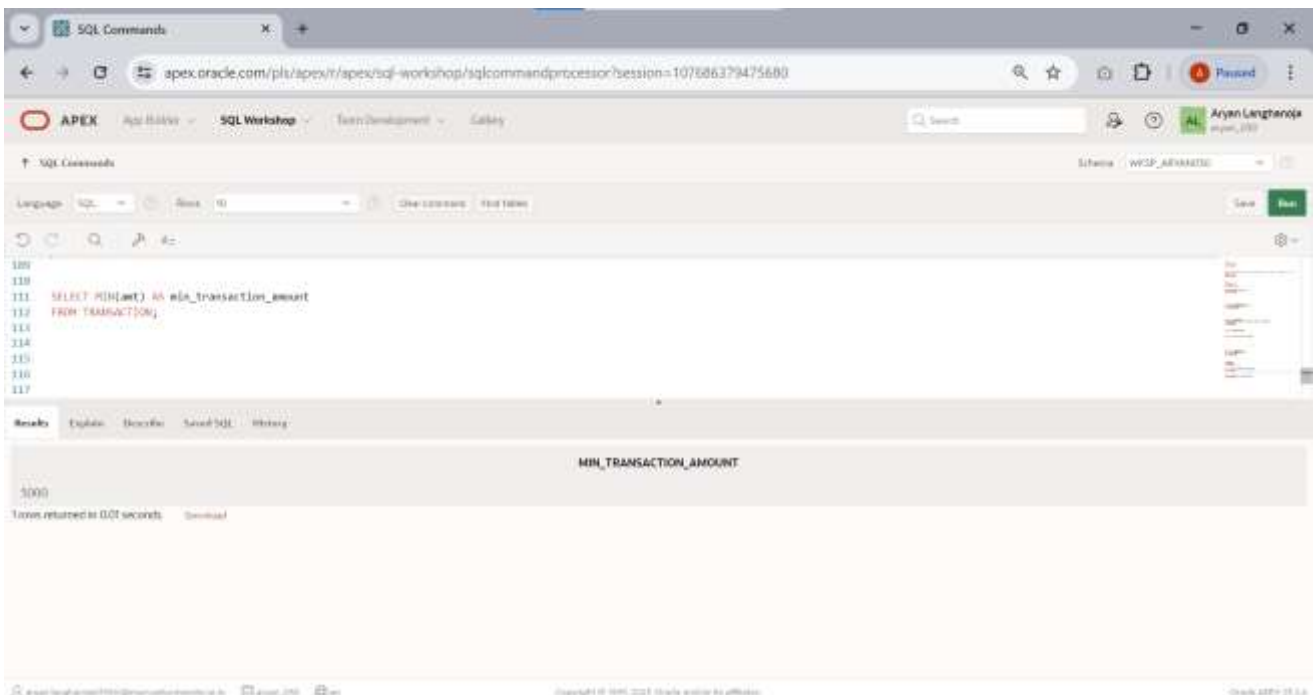
```
105  
106 SELECT SUM(amt) AS total_transaction_amount  
107 FROM TRANSACTION;  
108  
109  
110  
111  
112  
113
```

The results section displays a single row with the value 85000 under the column header TOTAL\_TRANSACTION\_AMOUNT.

TOTAL_TRANSACTION_AMOUNT
85000

1 rows returned in 0.01 seconds

2. Find minimum amount of transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
110  
111 SELECT MIN(amt) AS min_transaction_amount  
112 FROM TRANSACTION;  
113  
114  
115  
116  
117
```

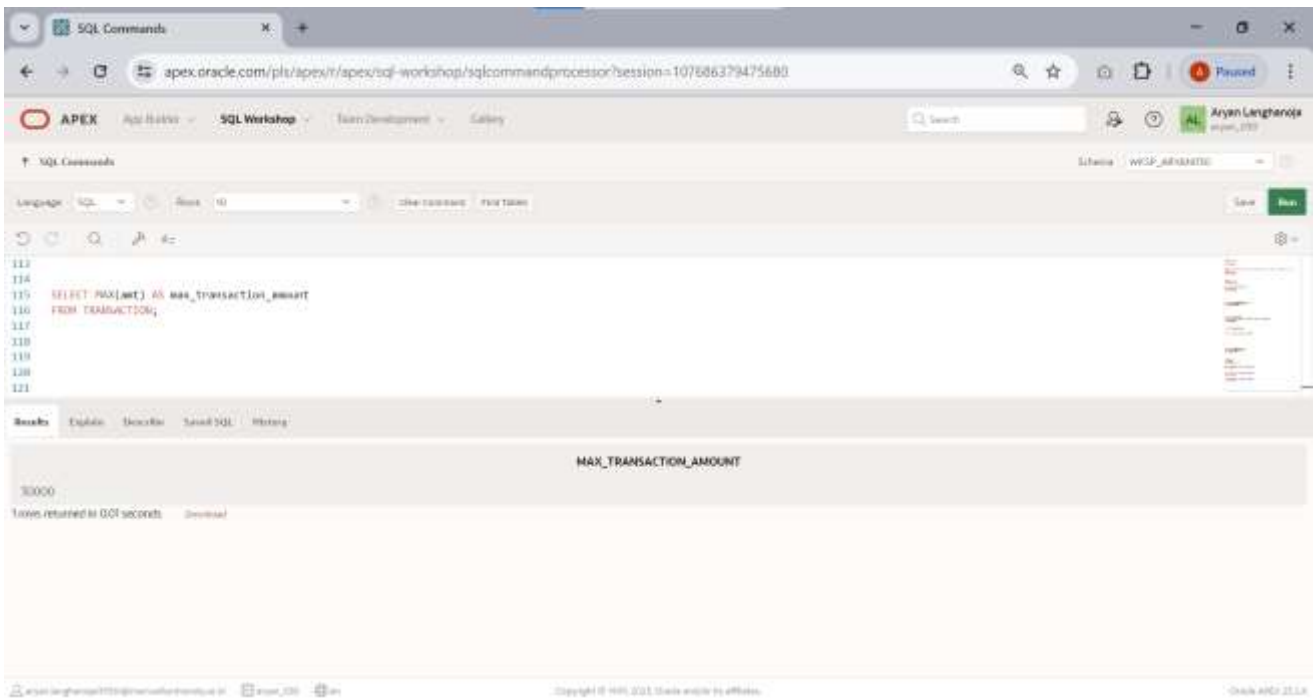
The results section displays a single row with the value 5000 under the column header MIN\_TRANSACTION\_AMOUNT.

MIN_TRANSACTION_AMOUNT
5000

1 rows returned in 0.01 seconds



3. Find maximum amount of transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

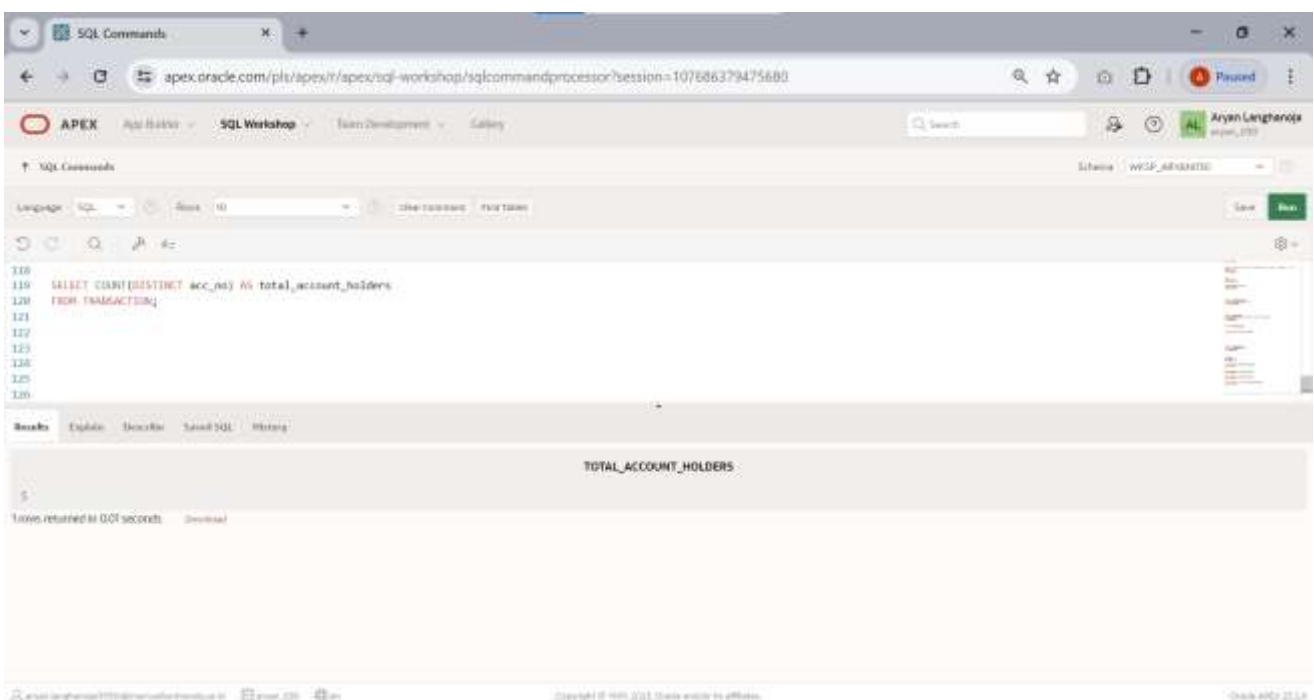
```
113  
114  
115 SELECT MAX(amt) AS max_transaction_amount  
116 FROM TRANSACTION;  
117  
118  
119  
120  
121
```

The results section displays a single row with the value 30000 under the column heading MAX\_TRANSACTION\_AMOUNT.

MAX_TRANSACTION_AMOUNT
30000

1 rows returned in 0.01 seconds. Download

4. Count the total account holders.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

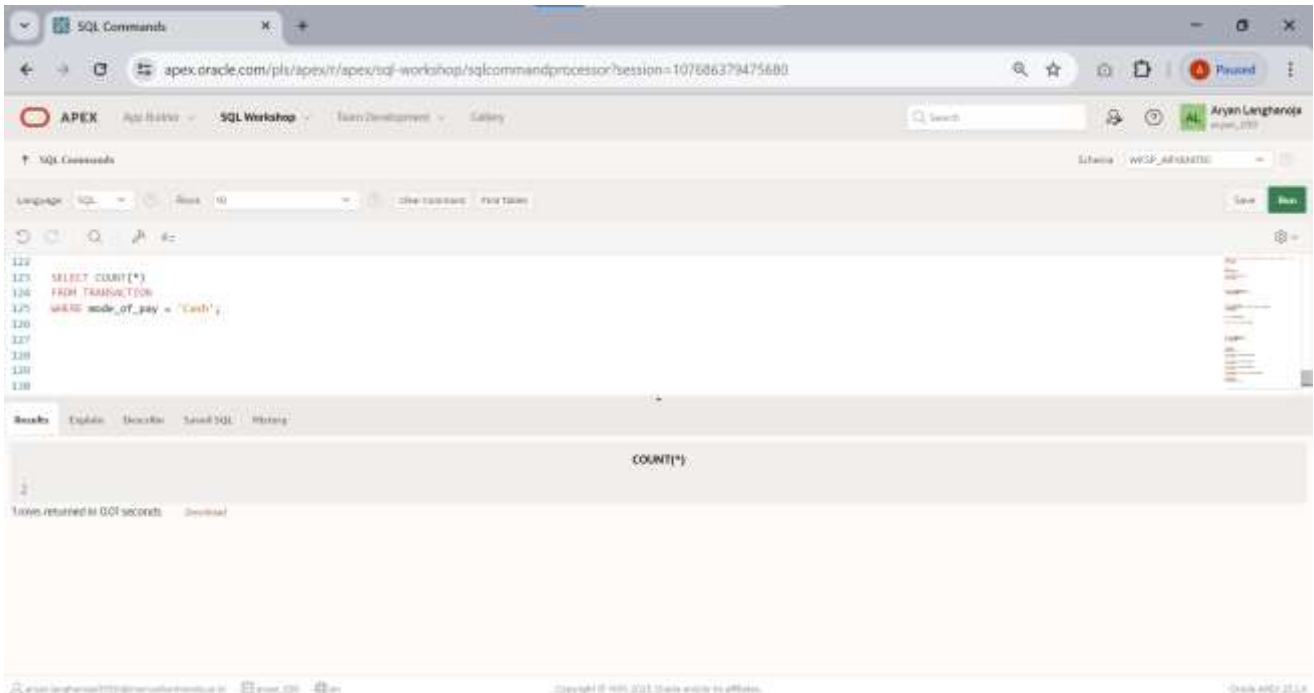
```
118  
119 SELECT COUNT(DISTINCT acc_no) AS total_account_holders  
120 FROM TRANSACTION;  
121  
122  
123  
124  
125  
126
```

The results section displays a single row with the value 5 under the column heading TOTAL\_ACCOUNT\_HOLDERS.

TOTAL_ACCOUNT_HOLDERS
5

1 rows returned in 0.01 seconds. Download

5. Count only those records whose made of payment is 'cash'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

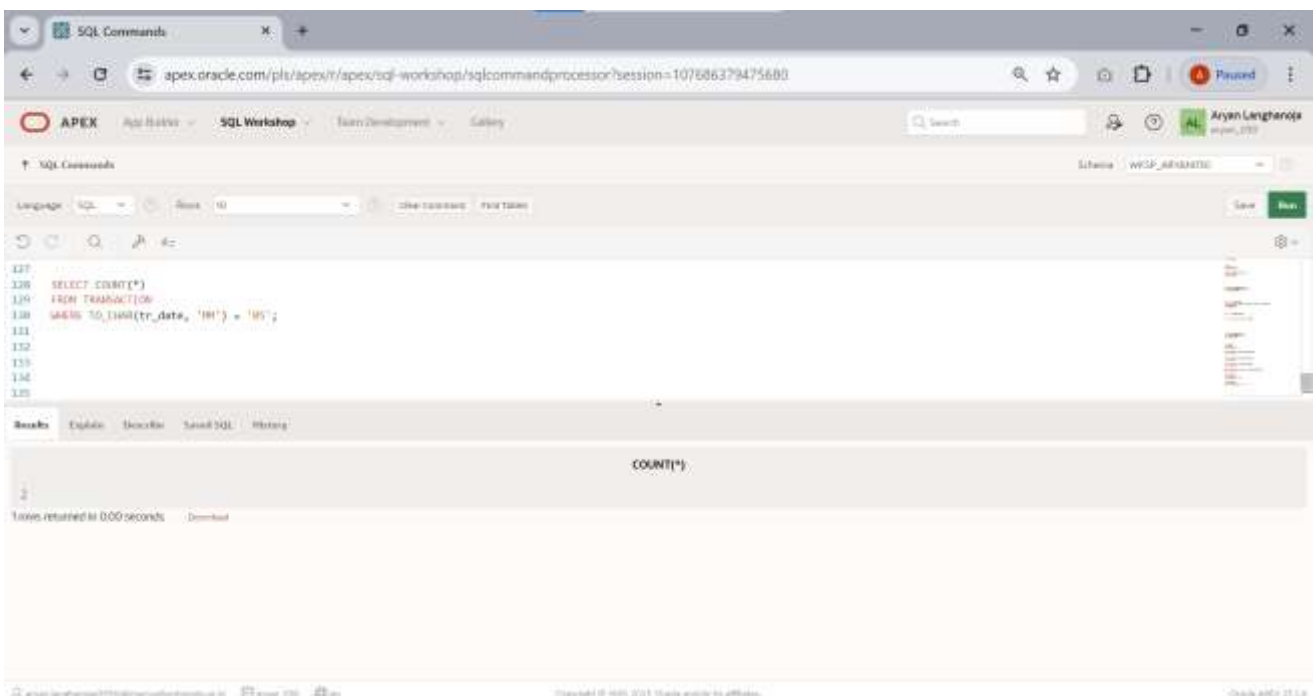
```
122  
123 SELECT COUNT(*)  
124 FROM TRANSACTION  
125 WHERE mode_of_pay = 'Cash';  
126  
127  
128  
129  
130
```

The results table shows a single row with the value 2 under the column COUNT(\*).

COUNT(*)
2

Times returned in 0.01 seconds

6. Count only those records whose transaction made in the month of 'MAY'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

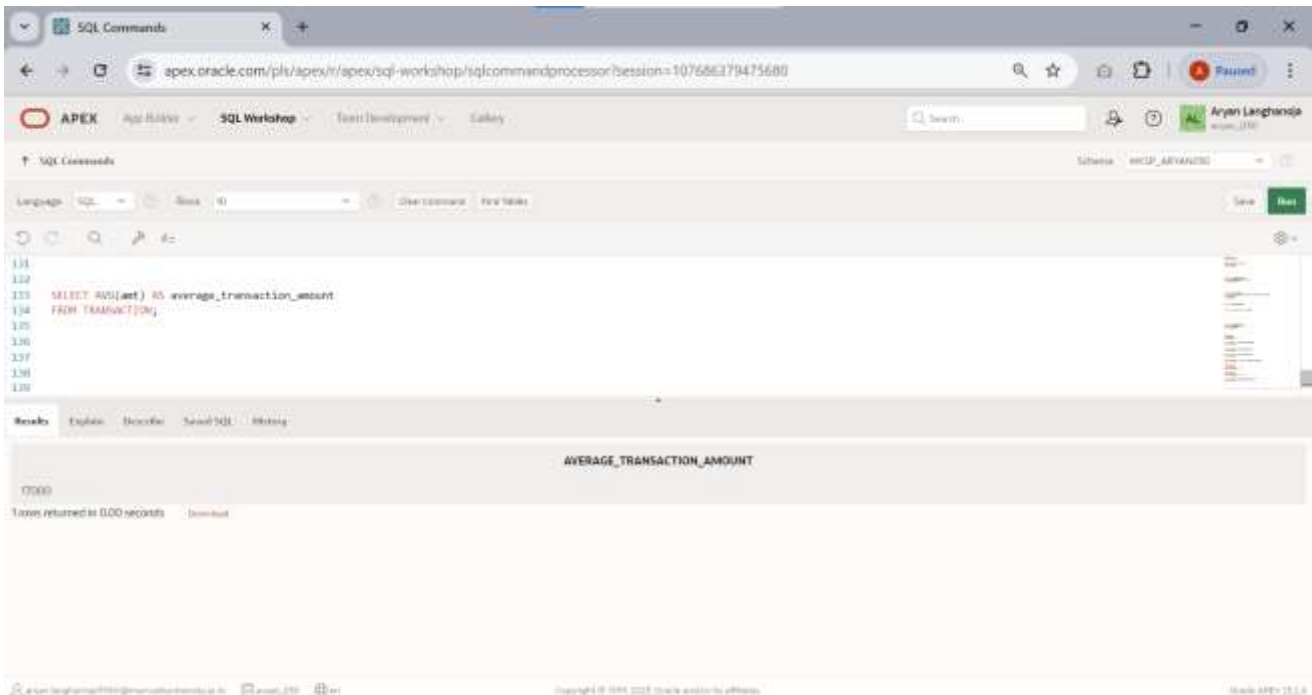
```
127  
128 SELECT COUNT(*)  
129 FROM TRANSACTION  
130 WHERE TO_CHAR(tr_date, 'MM') = '05';  
131  
132  
133  
134  
135
```

The results table shows a single row with the value 2 under the column COUNT(\*).

COUNT(*)
2

Times returned in 0.00 seconds

7. Find the average value of transaction.



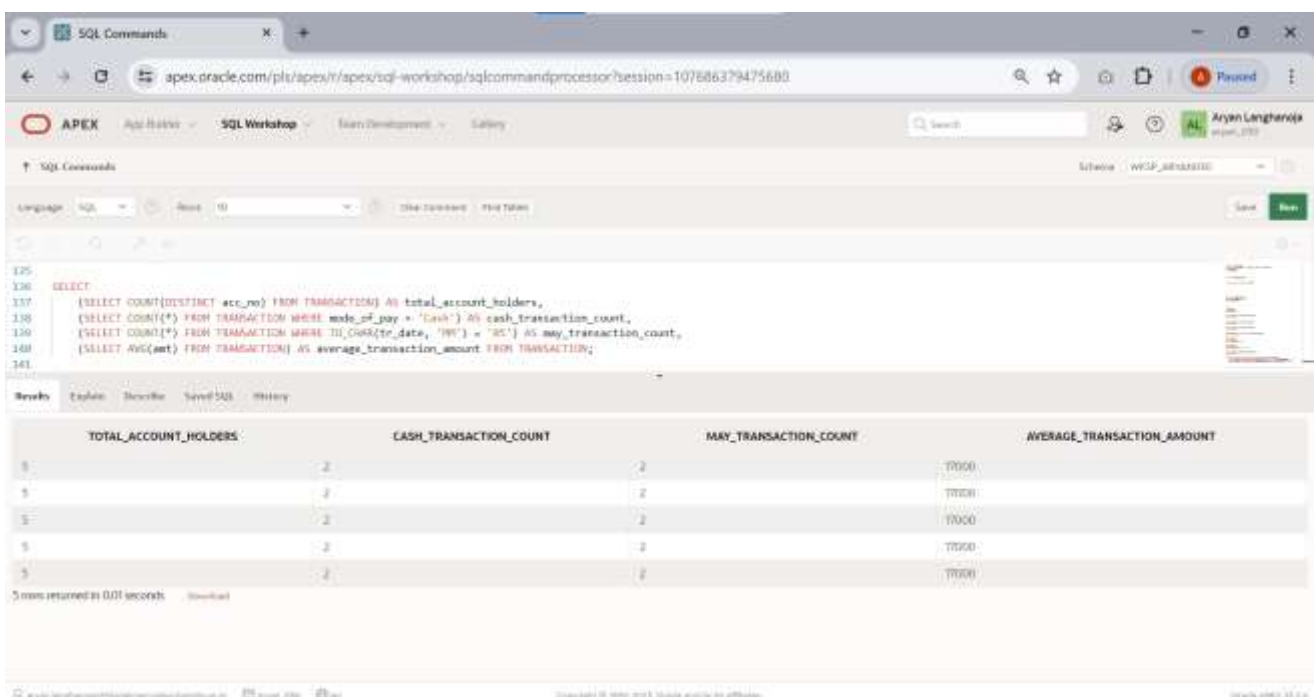
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

121
122
123 SELECT AVG(amt) AS average_transaction_amount
124 FROM TRANSACTION;
125
126
127
128
129
  
```

The results are displayed in a table with one column, **AVERAGE\_TRANSACTION\_AMOUNT**, and one row with the value **7000**.

8. Display the result of 4 rest to 4.



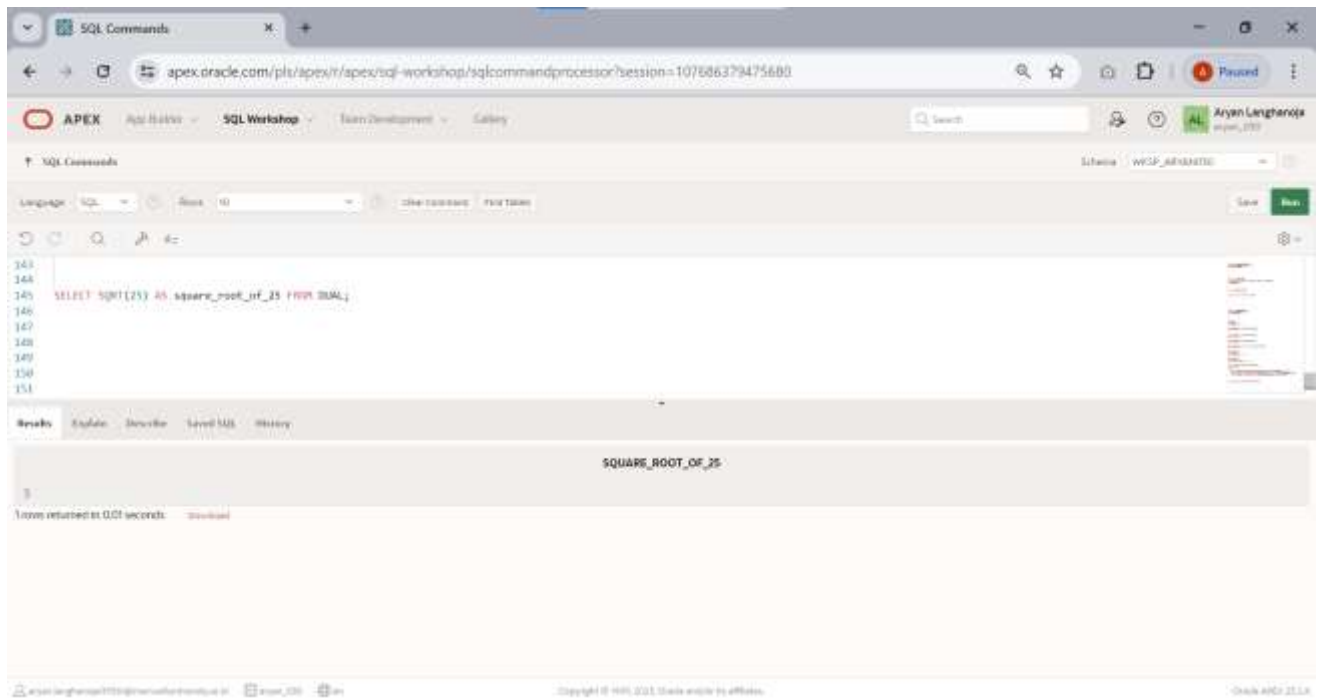
The screenshot shows the APEX SQL Workshop interface with a more complex query:

```

125
126 SELECT
127   (SELECT COUNT(DISTINCT acc_no) FROM TRANSACTION) AS total_account_holders,
128   (SELECT COUNT(*) FROM TRANSACTION WHERE mode_of_pay = 'Cash') AS cash_transaction_count,
129   (SELECT COUNT(*) FROM TRANSACTION WHERE TO_CHAR(tr_date, 'YY') = '15') AS may_transaction_count,
130   (SELECT AVG(amt) FROM TRANSACTION) AS average_transaction_amount FROM TRANSACTION;
141
  
```

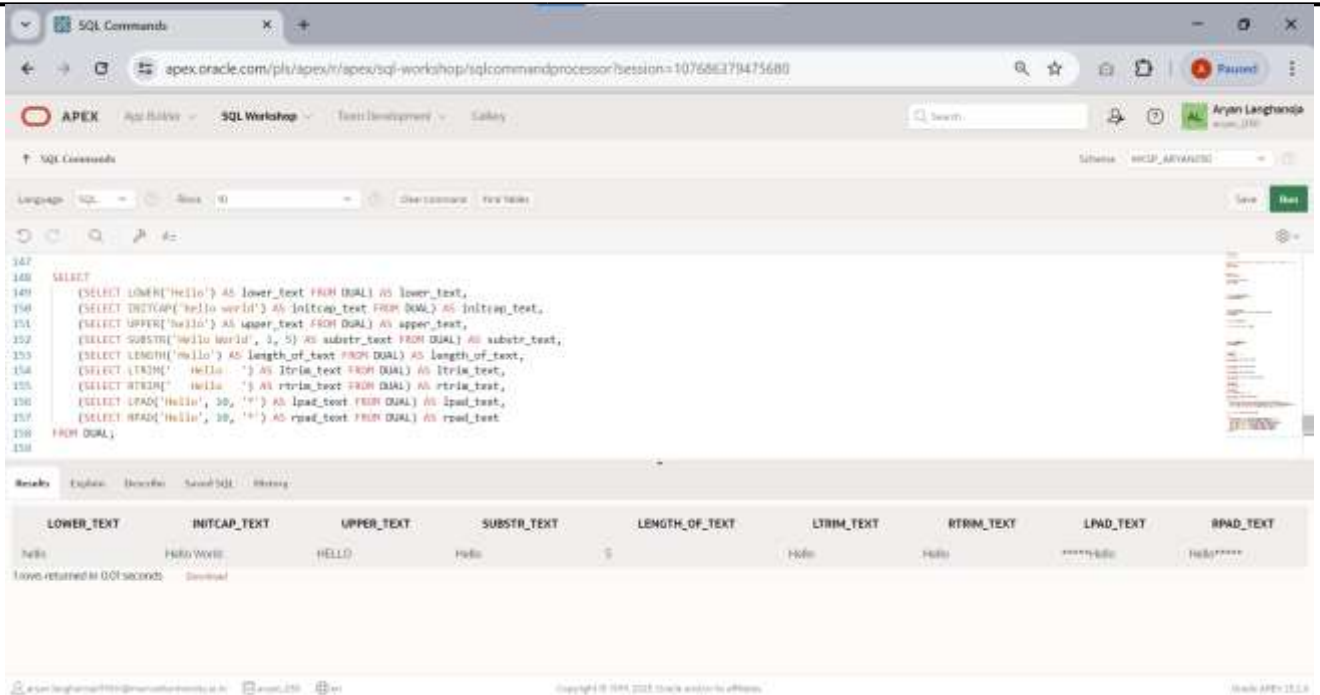
The results are displayed in a table with four columns: **TOTAL\_ACCOUNT\_HOLDERS**, **CASH\_TRANSACTION\_COUNT**, **MAY\_TRANSACTION\_COUNT**, and **AVERAGE\_TRANSACTION\_AMOUNT**. The table contains five rows of data, all with the same values: 5, 2, 2, and 7000 respectively.

9. Find the square root of 25.



10. Write the query for the following Function.

LOWER, INITCAP, UPPER, SUBSTR, LENGTH, LTRIM, RTRIM, LPAD, RPAD



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

147
148 SELECT
149   (SELECT LOWER('Hello') AS lower_text FROM DUAL) AS lower_text,
150   (SELECT INITCAP('Hello world') AS initcap_text FROM DUAL) AS initcap_text,
151   (SELECT UPPER('hello') AS upper_text FROM DUAL) AS upper_text,
152   (SELECT SUBSTR('Hello world', 1, 5) AS substr_text FROM DUAL) AS substr_text,
153   (SELECT LENGTH('Hello') AS length_of_text FROM DUAL) AS length_of_text,
154   (SELECT TRIM(' Hello ') AS ltrim_text FROM DUAL) AS ltrim_text,
155   (SELECT RTRIM(' Hello ') AS rtrim_text FROM DUAL) AS rtrim_text,
156   (SELECT LPAD('Hello', 10, '*') AS lpad_text FROM DUAL) AS lpad_text,
157   (SELECT RPAD('Hello', 10, '*') AS rpad_text FROM DUAL) AS rpad_text
158 FROM DUAL;
  
```

The Results window displays the following data:

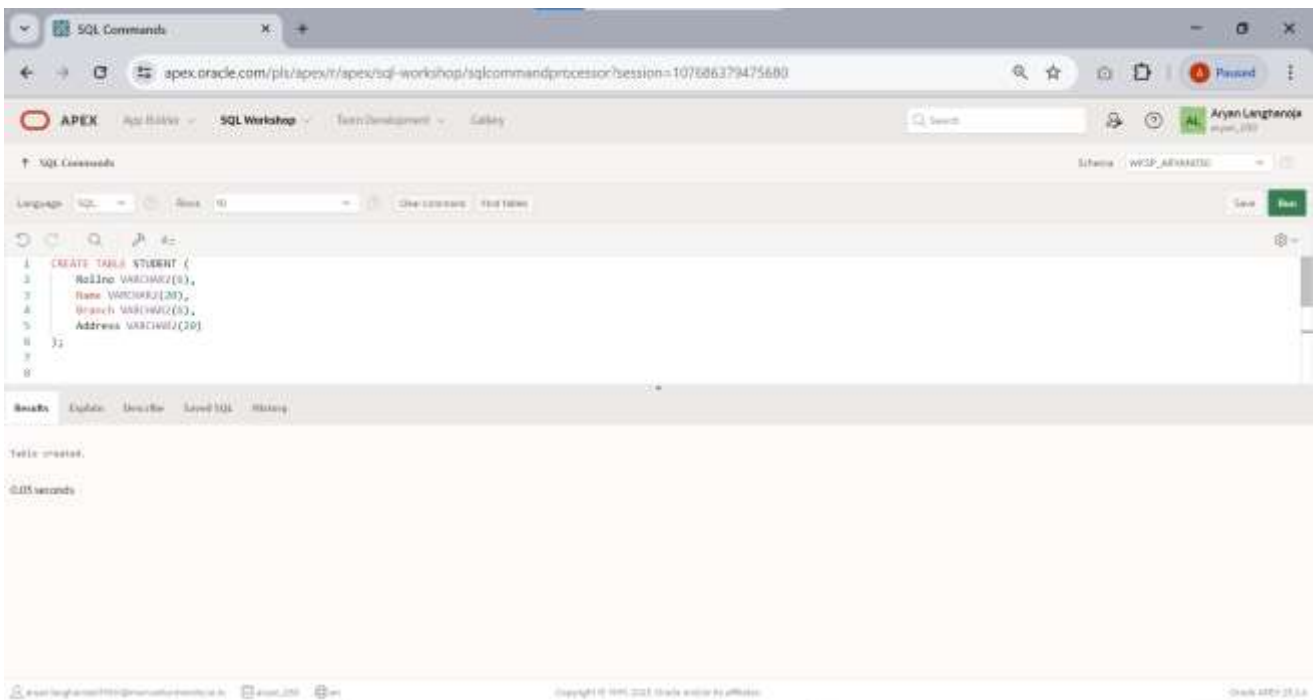
LOWER_TEXT	INITCAP_TEXT	UPPER_TEXT	SUBSTR_TEXT	LENGTH_OF_TEXT	LTRIM_TEXT	RTRIM_TEXT	LPAD_TEXT	RPAD_TEXT
hello	Hello world	HELLO	Hello	5	Hello	Hello	*****Hello	Hello*****

1 rows returned in 0.01 seconds

## CONSTRAINTS Based queries.

Create a table: **STUDENT**

Name of column	Type and Size
Rollno	Varchar2(6)
Name	Varchar2(20)
Branch	Varchar2(6)
Address	Varchar2(20)

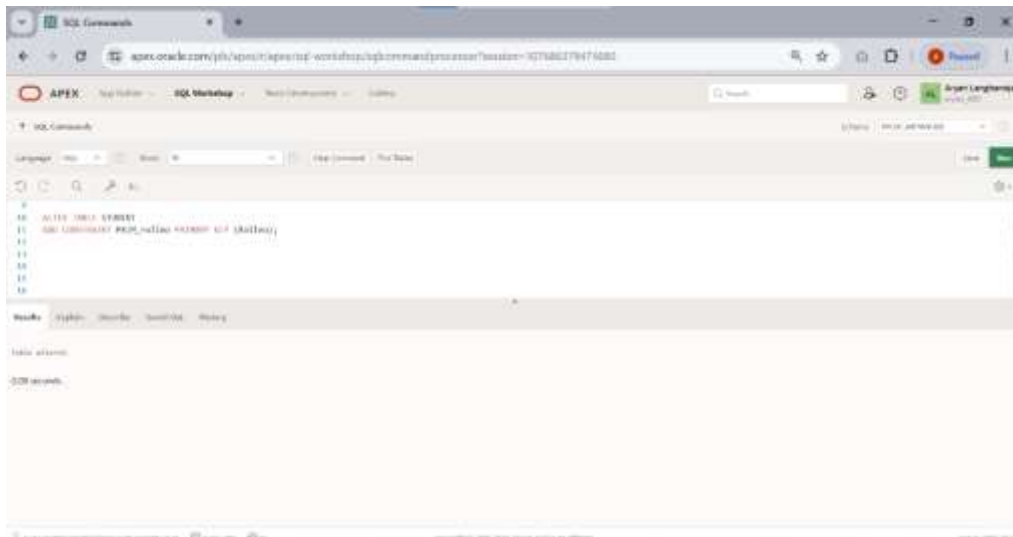


The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL statement:

```
1 CREATE TABLE STUDENT (
2   Rollno VARCHAR2(6),
3   Name VARCHAR2(20),
4   Branch VARCHAR2(6),
5   Address VARCHAR2(20)
6 );
```

The Results tab shows the output: "Table created." and "0.05 seconds".

1. Add PRIMARY KEY (roll no) and provide constraint name PRIM\_rollno.

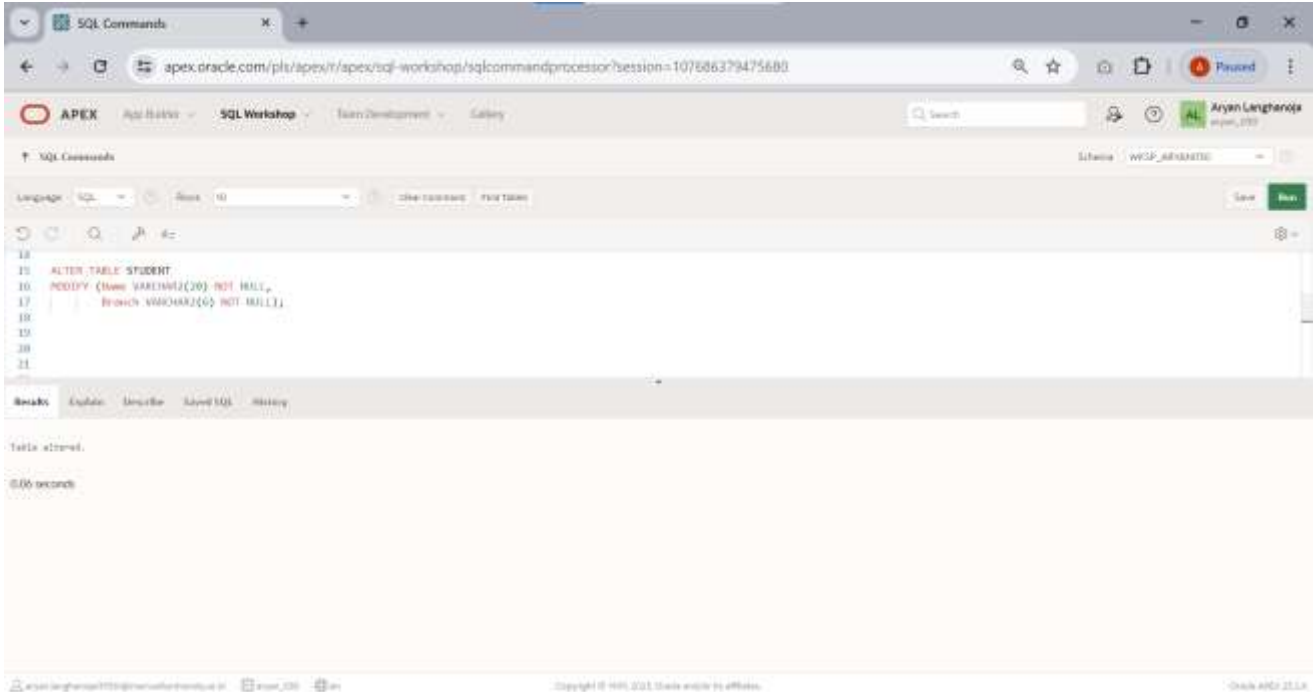


The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL statement:

```
1 ALTER TABLE STUDENT
2 ADD CONSTRAINT PRIM_rollno PRIMARY KEY (Rollno);
```

The Results tab shows the output: "Table altered." and "0.08 seconds".

2. Add NOT NULL constraint to name,branch for student table.

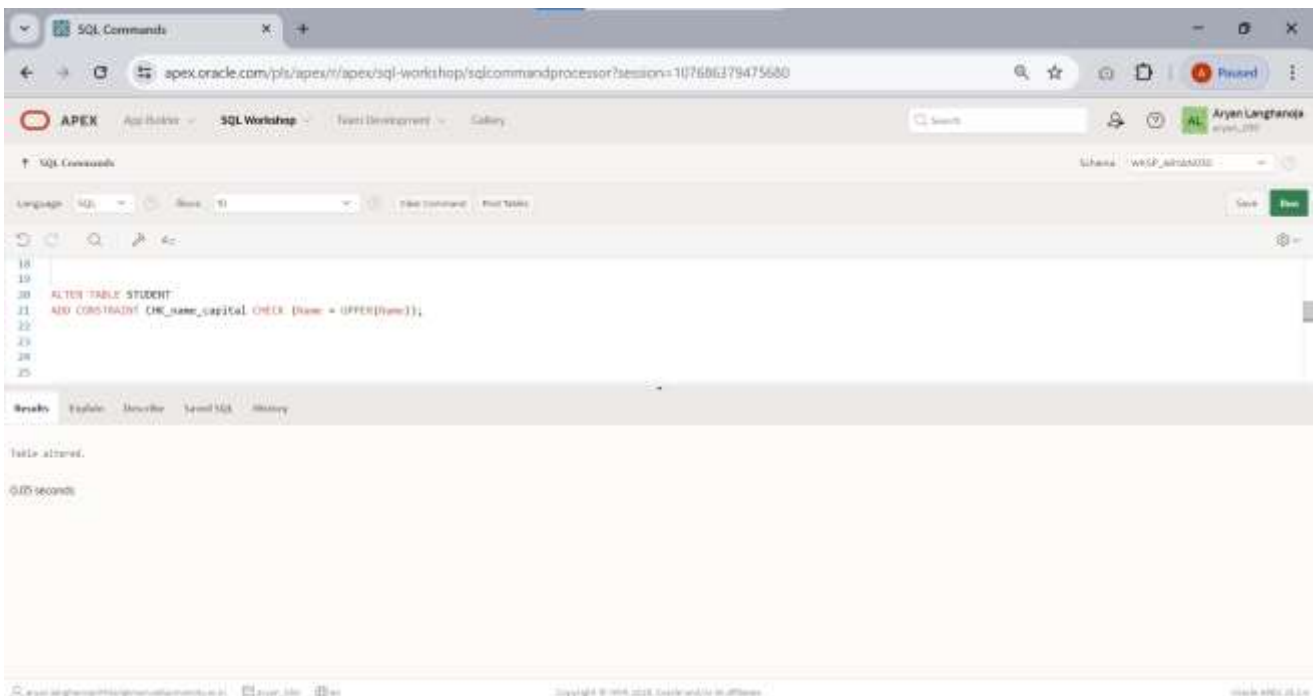


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
18  
19 ALTER TABLE STUDENT  
20 MODIFY (Name VARCHAR2(30) NOT NULL,  
21        Branch VARCHAR2(60) NOT NULL);  
22  
23  
24  
25
```

The output shows the table was altered successfully in 0.06 seconds.

3. Add check constraint and check name is in capital letter.

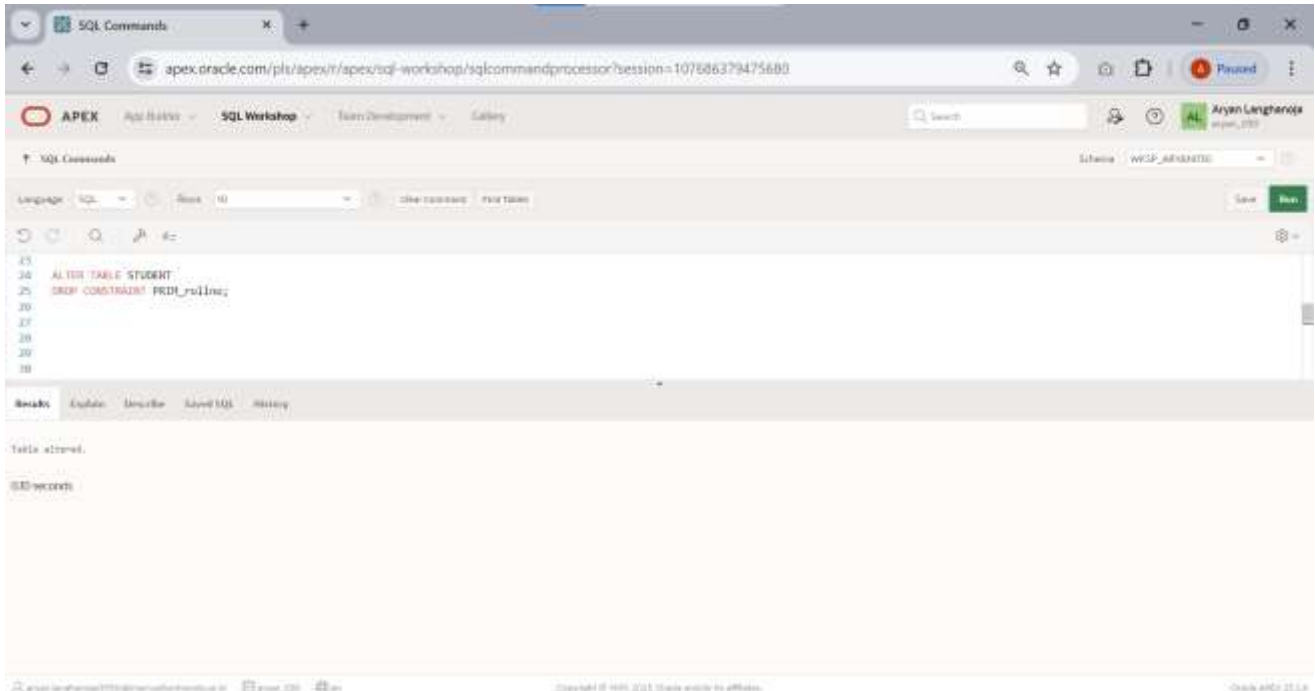


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
18  
19  
20 ALTER TABLE STUDENT  
21 ADD CONSTRAINT CHK_name_capital CHECK (Name = UPPER(Name));  
22  
23  
24  
25
```

The output shows the table was altered successfully in 0.05 seconds.

4. Drop the primary key.

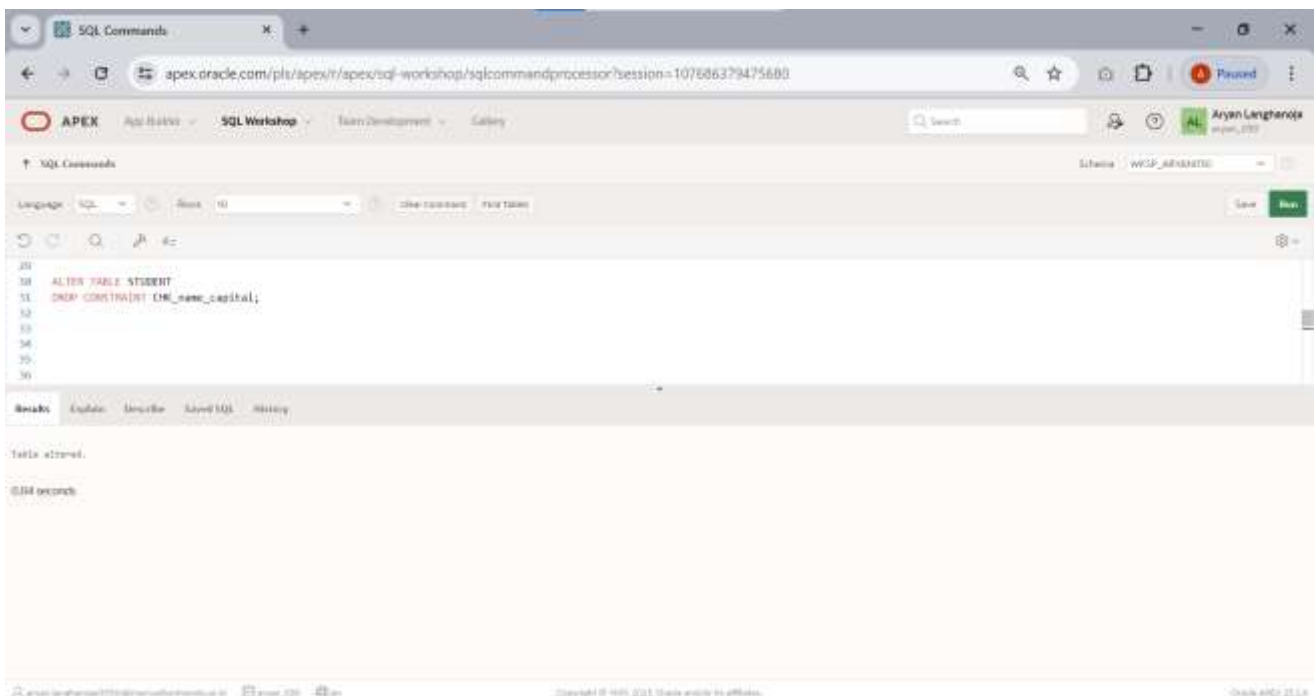


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
23  
24 ALTER TABLE STUDENT  
25 DROP CONSTRAINT PRDT_rollno;  
26  
27  
28  
29
```

The results pane shows the message: "Table altered." and "0.00 seconds".

5. Drop the constraint.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

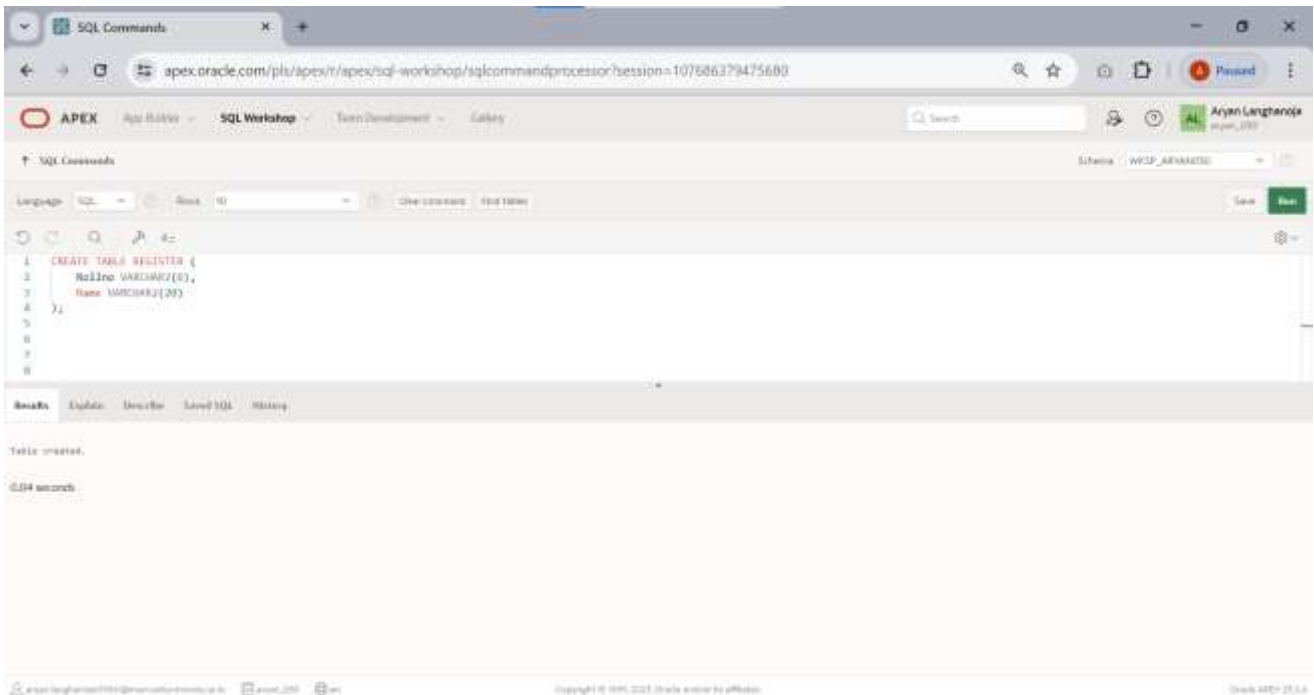
```
30  
31 ALTER TABLE STUDENT  
32 DROP CONSTRAINT CNE_name_capital;  
33  
34  
35  
36
```

The results pane shows the message: "Table altered." and "0.04 seconds".



Create a Table **REGISTER**.

Name of column	Type and Size
Rollno	Varchar2(6)
Name	Varchar2(20)



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

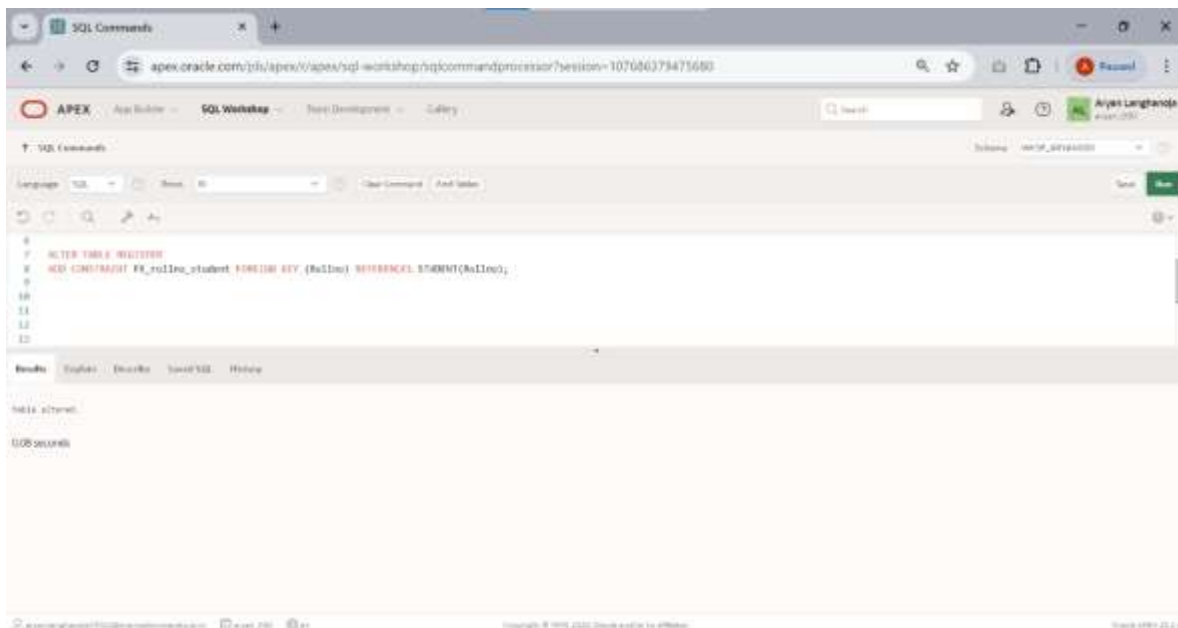
```

1 CREATE TABLE REGISTER (
2   Rollno VARCHAR2(6),
3   Name VARCHAR2(20)
4 );

```

The results pane shows the message: "Table created." and the execution time: "0.04 seconds".

1. Provide foreign key references rollno of student table.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

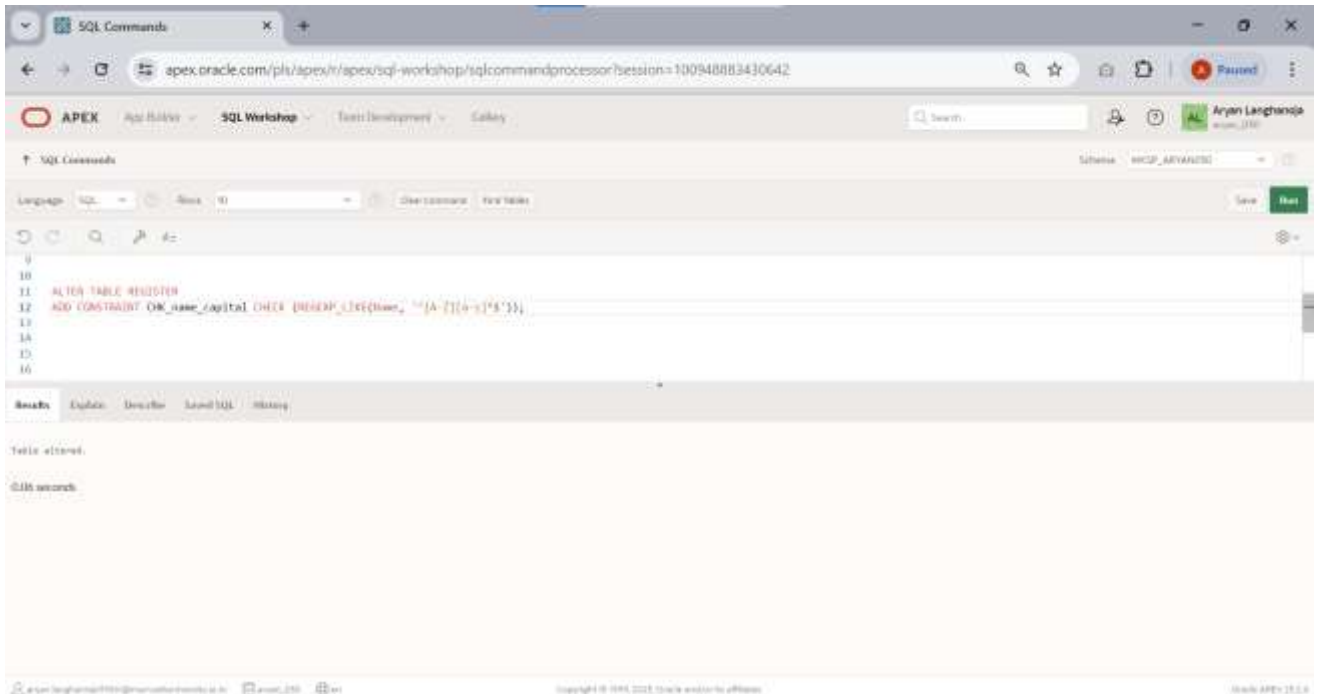
```

6 ALTER TABLE REGISTER
7 ADD CONSTRAINT FK_rollno_student FOREIGN KEY (Rollno) REFERENCES STUDENT(Rollno);

```

The results pane shows the message: "Table altered." and the execution time: "0.08 seconds".

2. Add check constraint to check name's first letter is always capital.

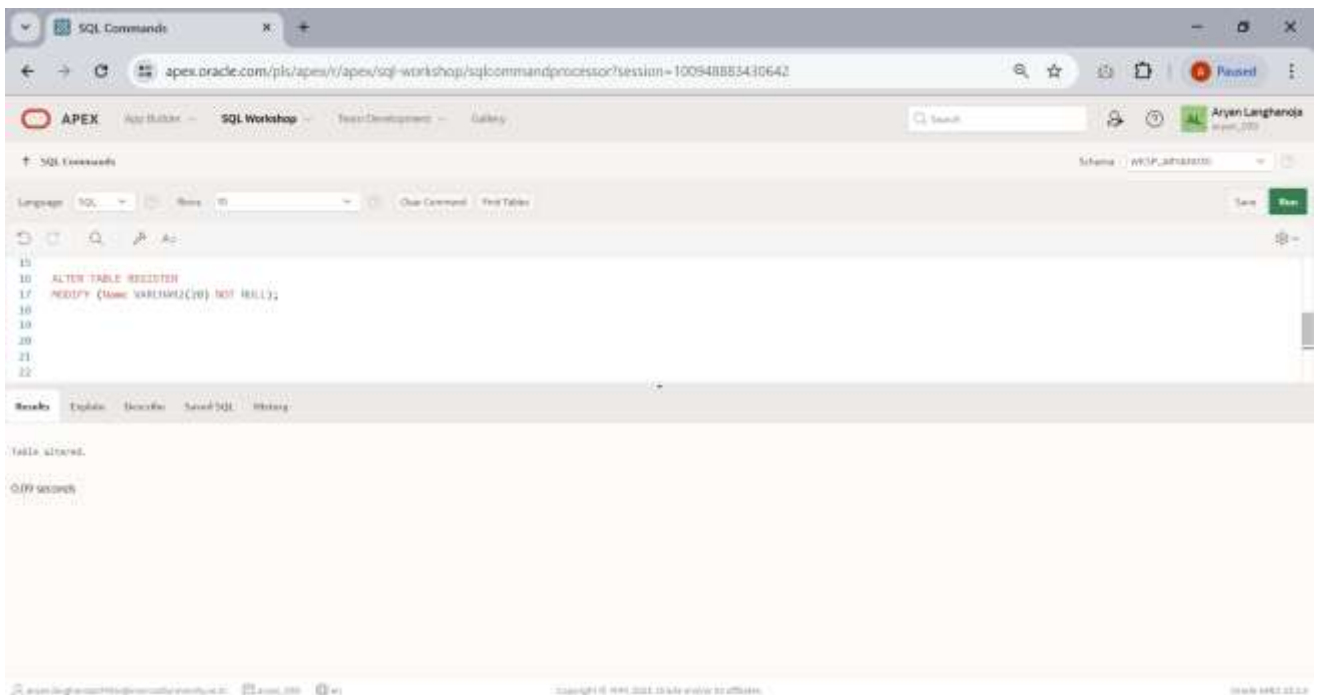


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
10 ALTER TABLE REGISTER
11 ADD CONSTRAINT OK_name_capital CHECK (REGEXP_LIKE(name, '^[A-Z][a-s]*$'));
```

The results section shows the message "Table altered." and "0.15 seconds".

3. Add NOT NULL constraint to name of register table.

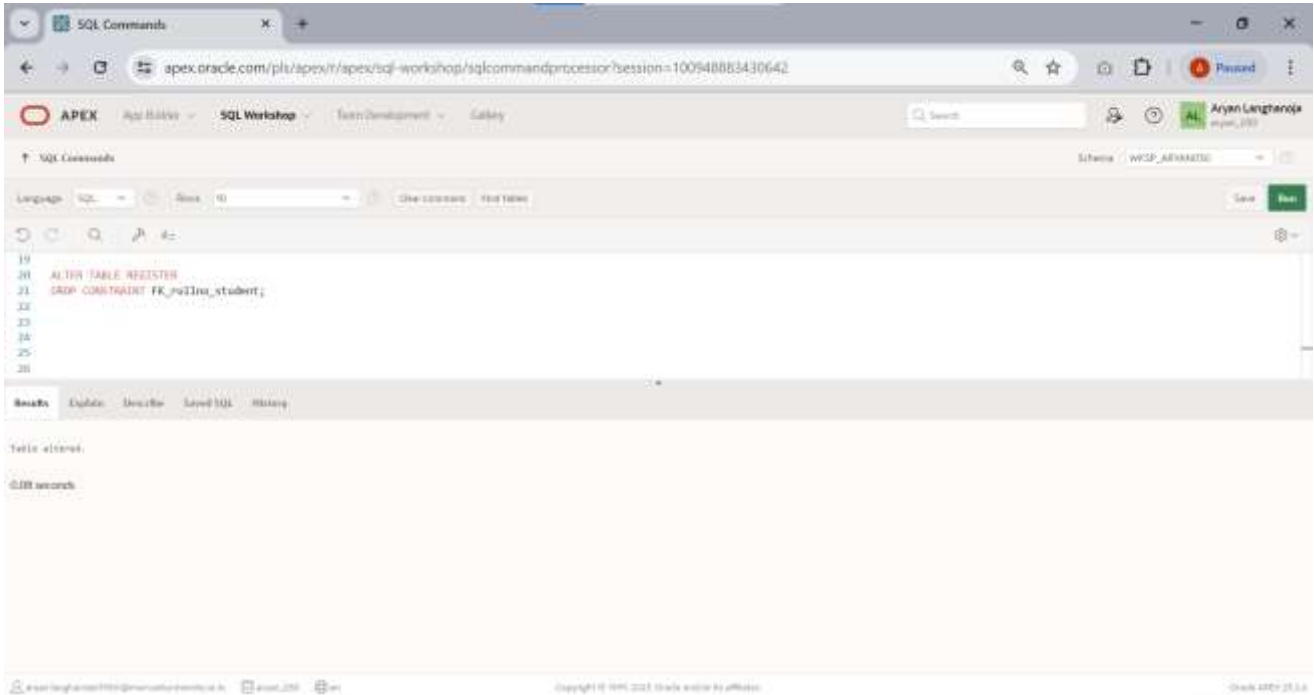


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
15
16 ALTER TABLE REGISTER
17 MODIFY (name VARCHAR2(50) NOT NULL);
```

The results section shows the message "Table altered." and "0.09 seconds".

4. Drop foreign key of REGISTER table.

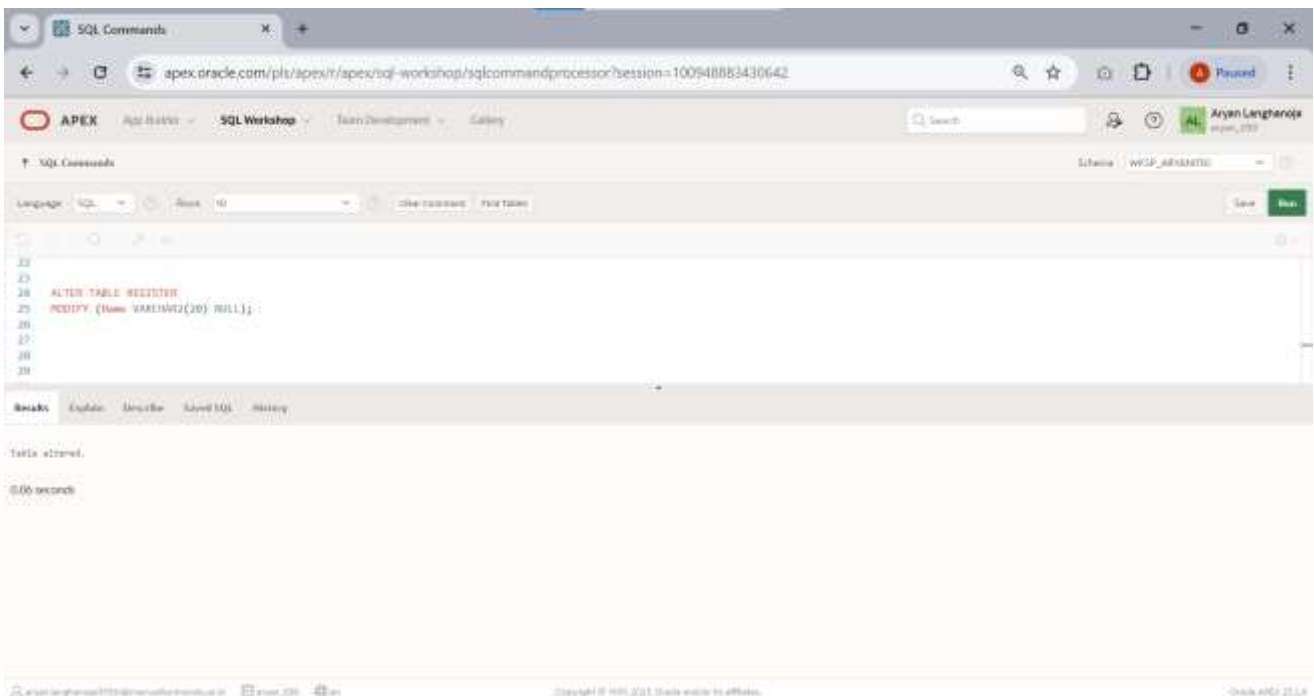


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
19  
20 ALTER TABLE REGISTER  
21 DROP CONSTRAINT FK_regstu_student;  
22  
23  
24  
25  
26
```

The results pane shows the message "Table altered." and the execution time "0.08 seconds".

5. Drop NOT NULL constraint.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
22  
23  
24 ALTER TABLE REGISTER  
25 MODIFY (name VARCHAR2(20) NULL);  
26  
27  
28  
29
```

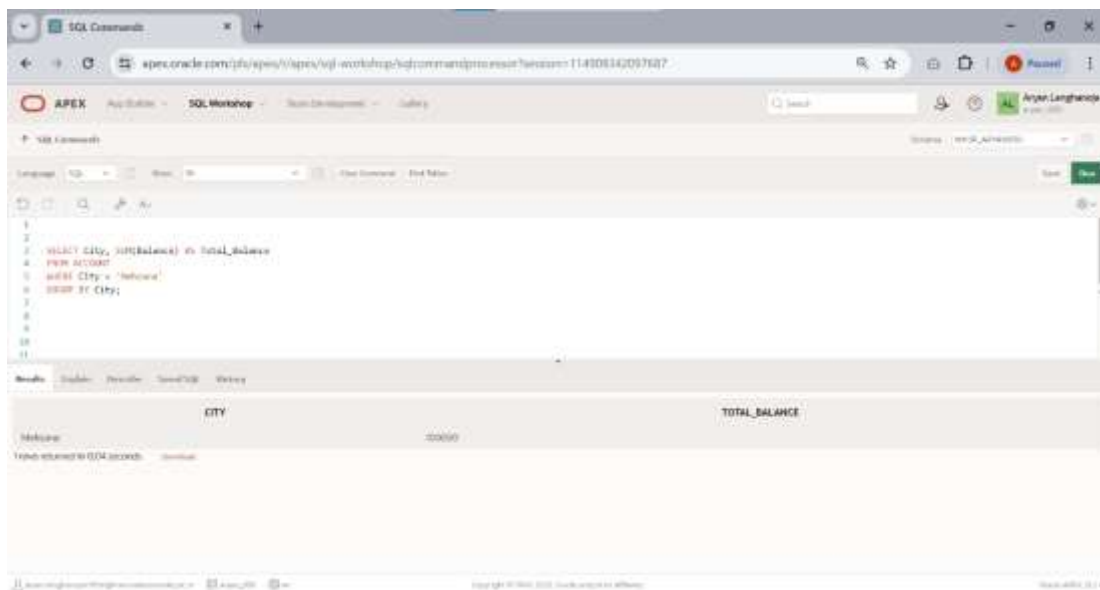
The results pane shows the message "Table altered." and the execution time "0.06 seconds".

## Practical 7

### Aim: Implement operator, null and special operator-based queries

**NOTE: for following queries use TABLES of PRACTICAL-1**

1. Display the sum of balance of account holders who's live in same city 'Mehsana' using group by clause.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

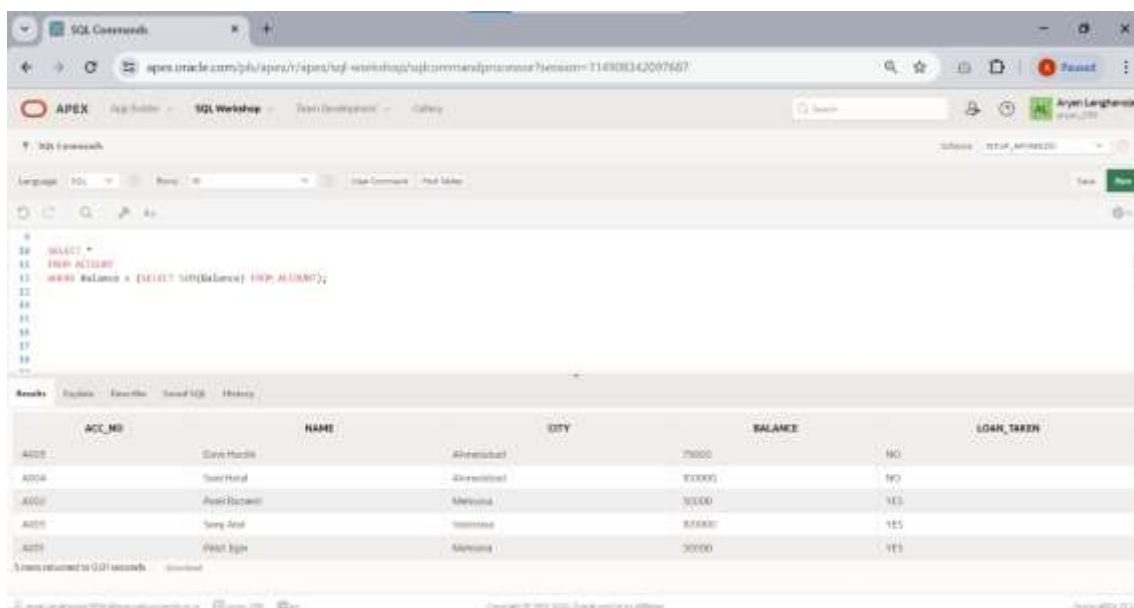
1
2
3 SELECT City, SUM(Balance) AS Total_Balance
4 FROM ACCOUNT
5 WHERE City = 'Mehsana'
6 GROUP BY City;
  
```

The Results tab displays the following table:

CITY	TOTAL_BALANCE
Mehsana	200000

Values returned in 0.04 seconds.

2. Display the information about account where balance is less than total balance of all account holders.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

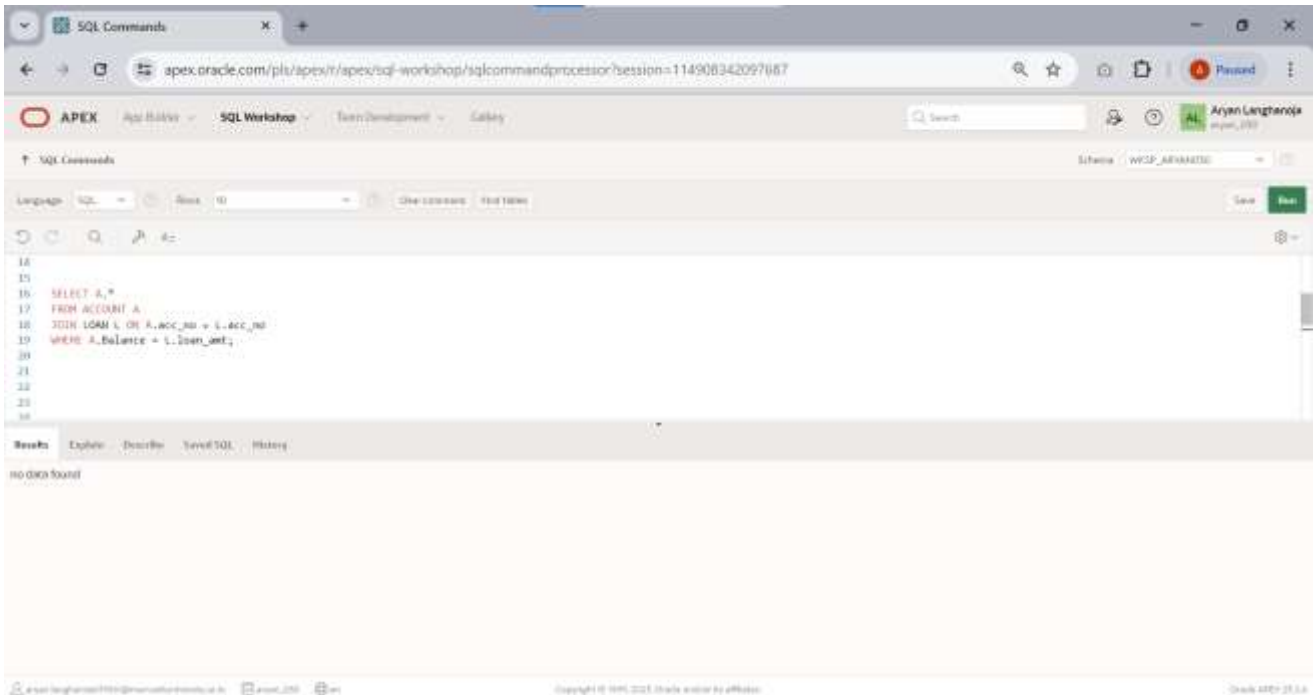
10
11
12 SELECT *
13 FROM ACCOUNT
14 WHERE Balance < (SELECT SUM(Balance) FROM ACCOUNT);
  
```

The Results tab displays the following table:

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN
ACC01	Surya Harsol	Ahmedabad	75000	NO
ACC04	Surya Harsol	Ahmedabad	100000	NO
ACC02	Amit Desai	Mehsana	30000	YES
ACC05	Surya Harsol	Mehsana	80000	YES
ACC03	Amit Desai	Mehsana	30000	YES

3 rows returned in 0.01 seconds.

- Displays the information of account holders whose loan amount and balance both are same.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

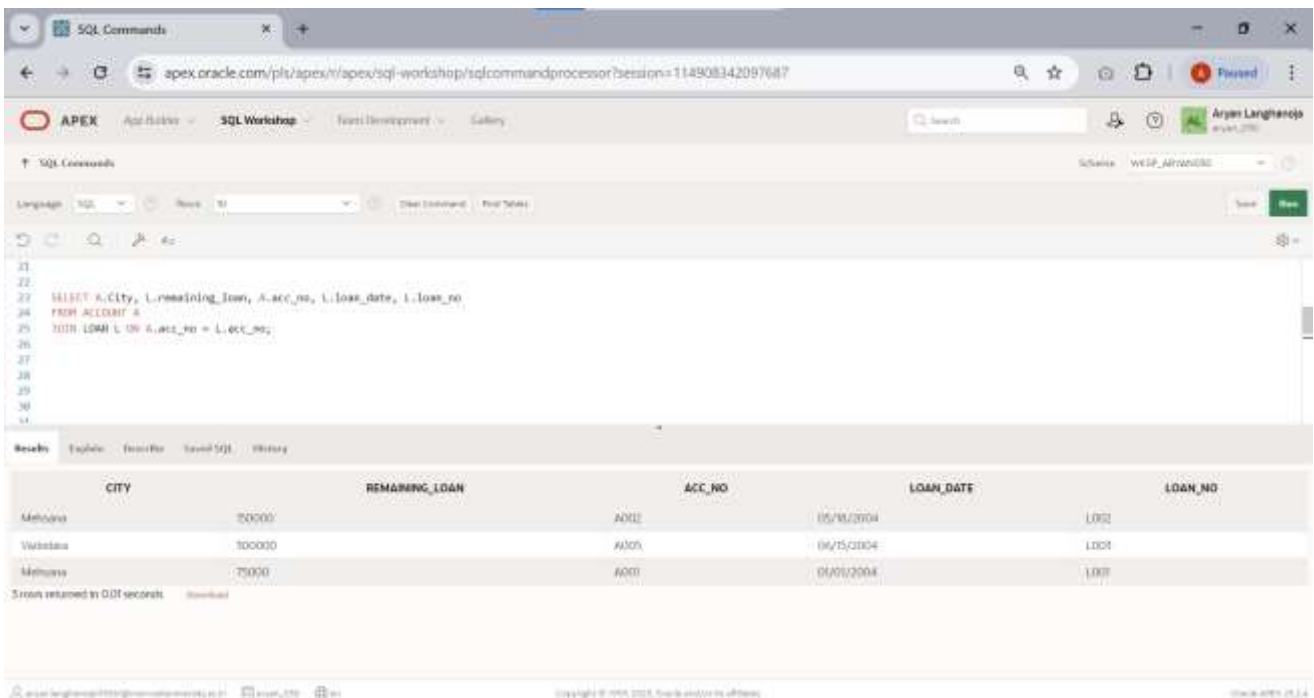
```

14
15
16 SELECT A.*
17 FROM ACCOUNT A
18 JOIN LOAN L ON A.acc_no = L.acc_no
19 WHERE A.balance = L.loan_amt;
20
21
22
23
24

```

The results section shows "No Data Found".

- Display the name of city, remaining loan amount, account, date of loan and loan number of account holders.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

21
22
23 SELECT A.City, L.remaining_loan, A.acc_no, L.loan_date, L.loan_no
24 FROM ACCOUNT A
25 JOIN LOAN L ON A.acc_no = L.loan_no;
26
27
28
29
30
31

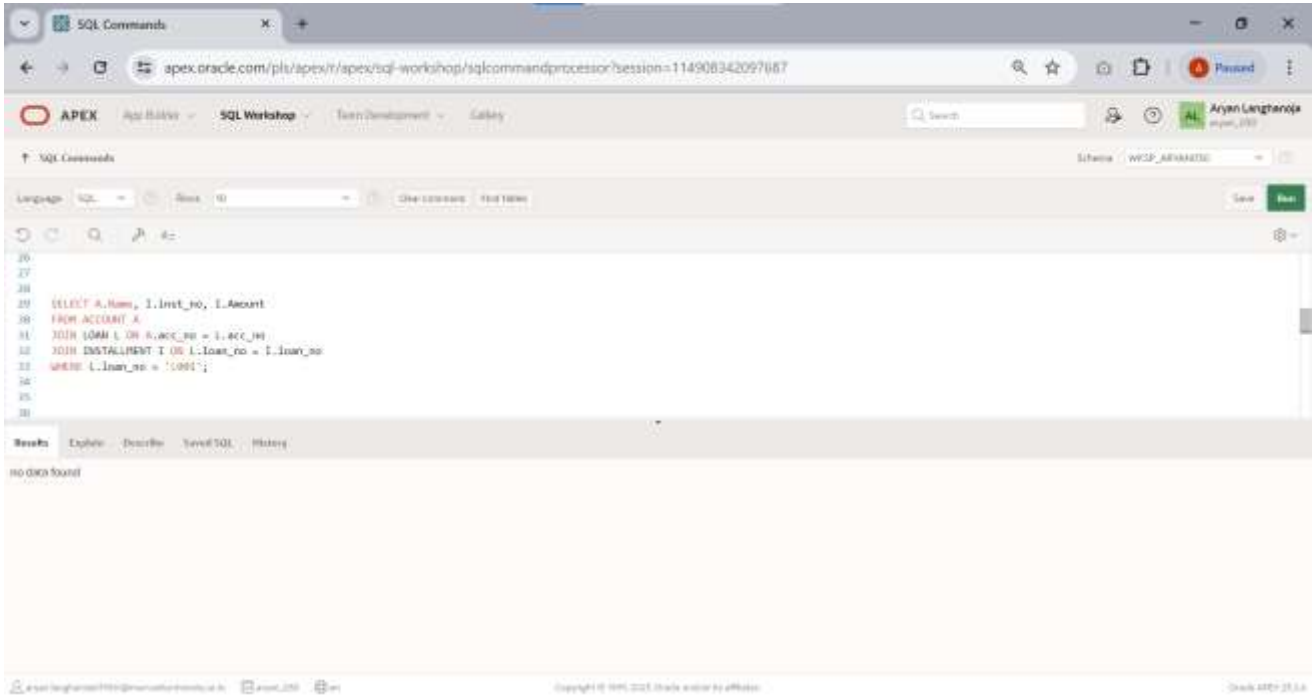
```

The results section displays a table with the following data:

CITY	REMAINING_LOAN	ACC_NO	LOAN_DATE	LOAN_NO
Meluhara	75000	A001	05/10/2004	L002
Varadaha	100000	A005	06/15/2004	L005
Meluhara	75000	A001	01/01/2004	L001

Screen returned in 0.07 seconds.

5. Display name of account holder, installment number and installment amount Whose loan number is 'L001'.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

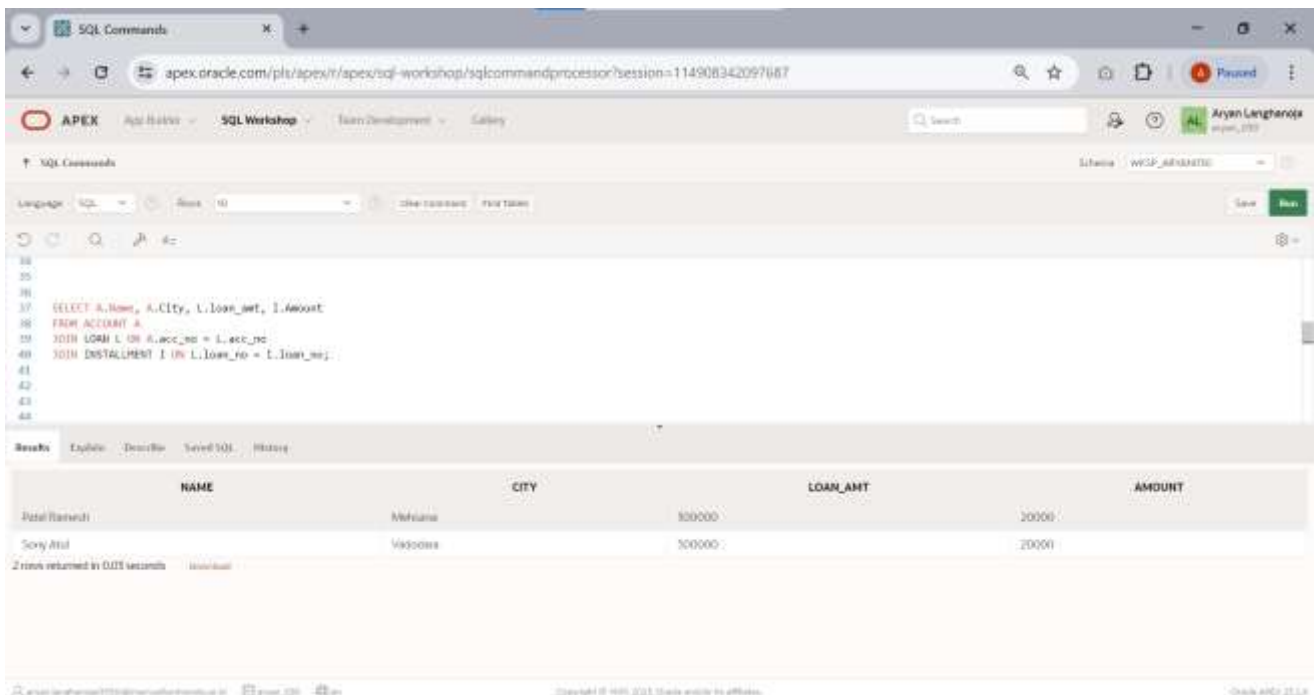
```

26
27
28
29 SELECT A.Name, I.Inst_no, I.Amount
30 FROM ACCOUNT A
31 JOIN LOAN L ON A.acc_no = L.acc_no
32 JOIN INSTALLMENT I ON L.loan_no = I.loan_no
33 WHERE I.loan_no = 'L001';
34
35
36

```

The Results tab is selected, and it displays "No data found".

6. Display name of account holder, city, loan amount and installment amount.



The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

```

38
39
40
41 SELECT A.Name, A.City, L.loan_amt, I.Amount
42 FROM ACCOUNT A
43 JOIN LOAN L ON A.acc_no = L.acc_no
44 JOIN INSTALLMENT I ON L.loan_no = I.loan_no;
45
46
47

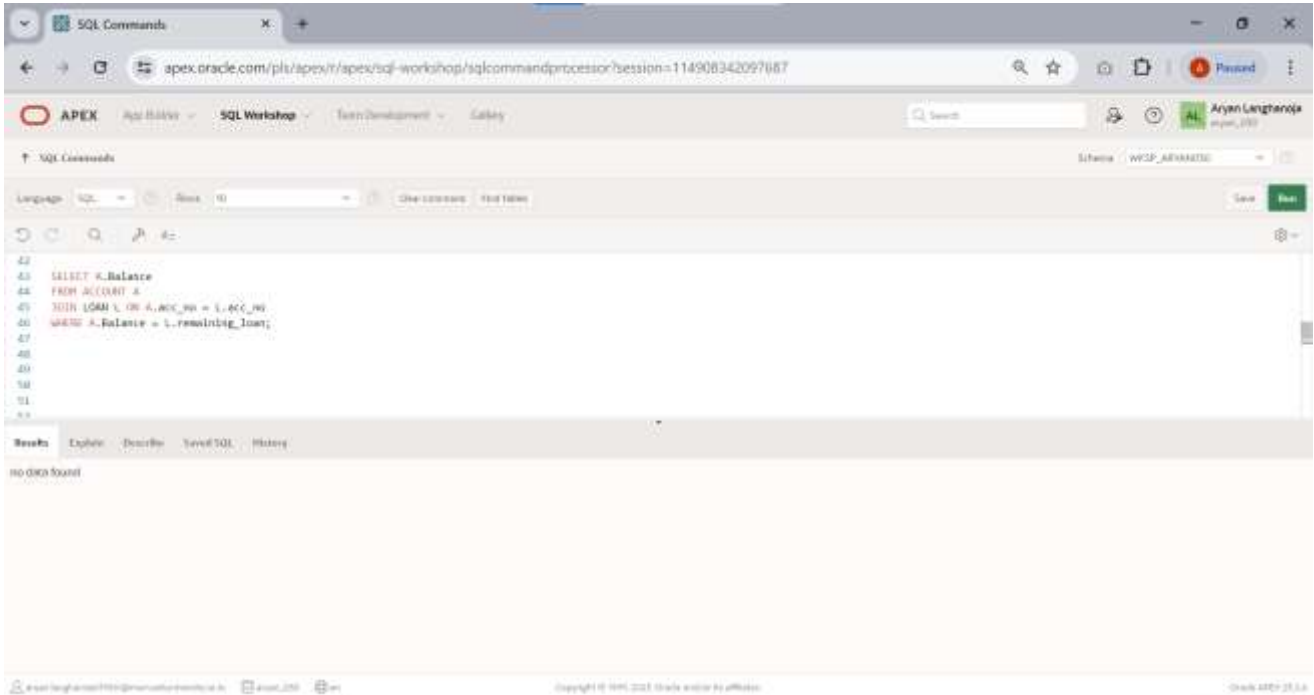
```

The Results tab is selected, and it displays the following data:

NAME	CITY	LOAN_AMT	AMOUNT
Patel Ramesh	Mehsana	500000	20000
Sony Atul	Vadodra	500000	20000

2 rows returned in 0.03 seconds

7. Display the balance of account holders whose balance and remaining loan both are same.

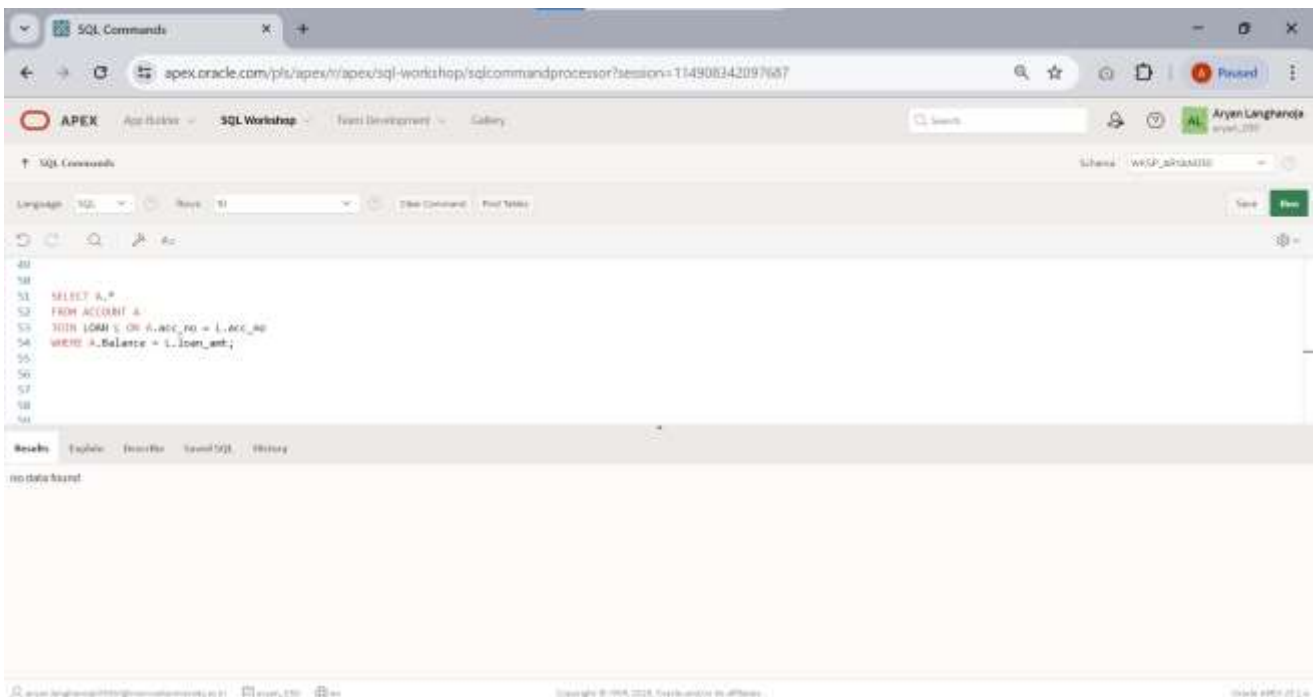


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
42  
43 SELECT A.Balance  
44 FROM ACCOUNT A  
45 JOIN LOAN L ON A.ACC_NO = L.ACC_NO  
46 WHERE A.Balance = L.remaining_loan;  
47  
48  
49  
50  
51  
52
```

The results section below the query shows "no data found".

8. List of all account holders' information whose balance is same as loan amount.

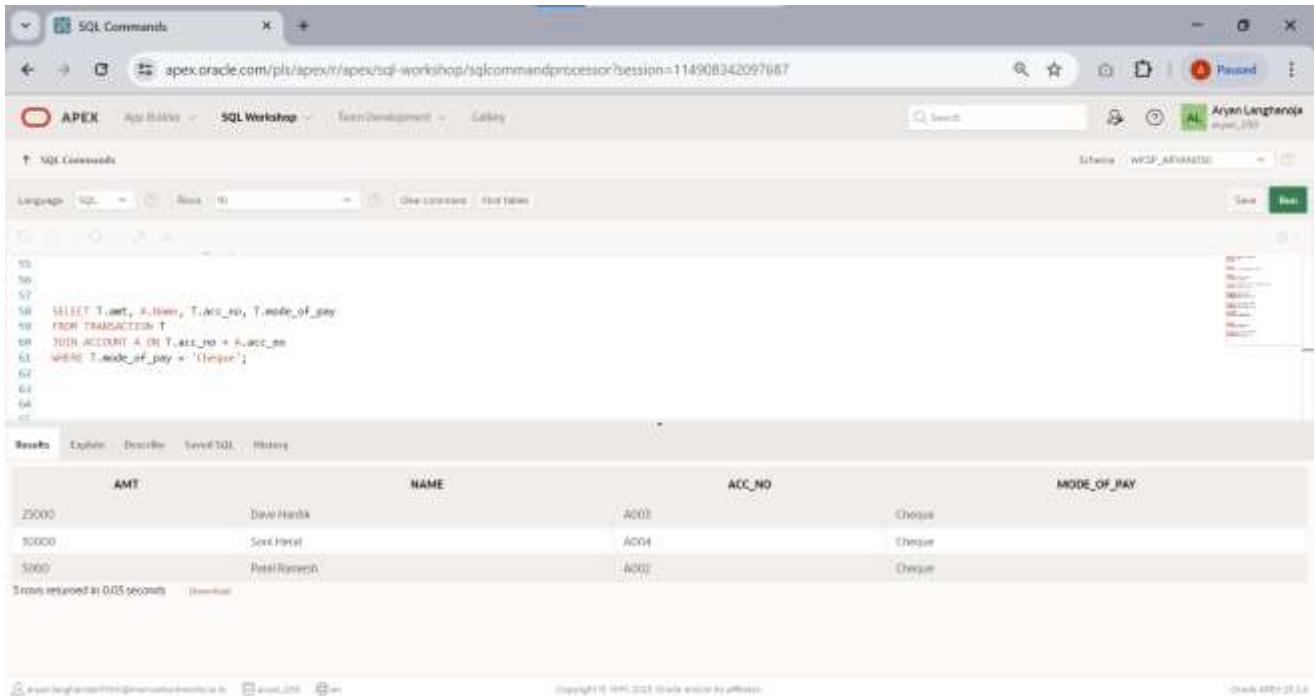


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
49  
50  
51 SELECT A.*  
52 FROM ACCOUNT A  
53 JOIN LOAN L ON A.ACC_NO = L.ACC_NO  
54 WHERE A.Balance = L.loan_amt;  
55  
56  
57  
58  
59
```

The results section below the query shows "no data found".

9. Display the amount of transaction, name of account holders, account number and mode of payment whose mode of payment is 'CHEQUE'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

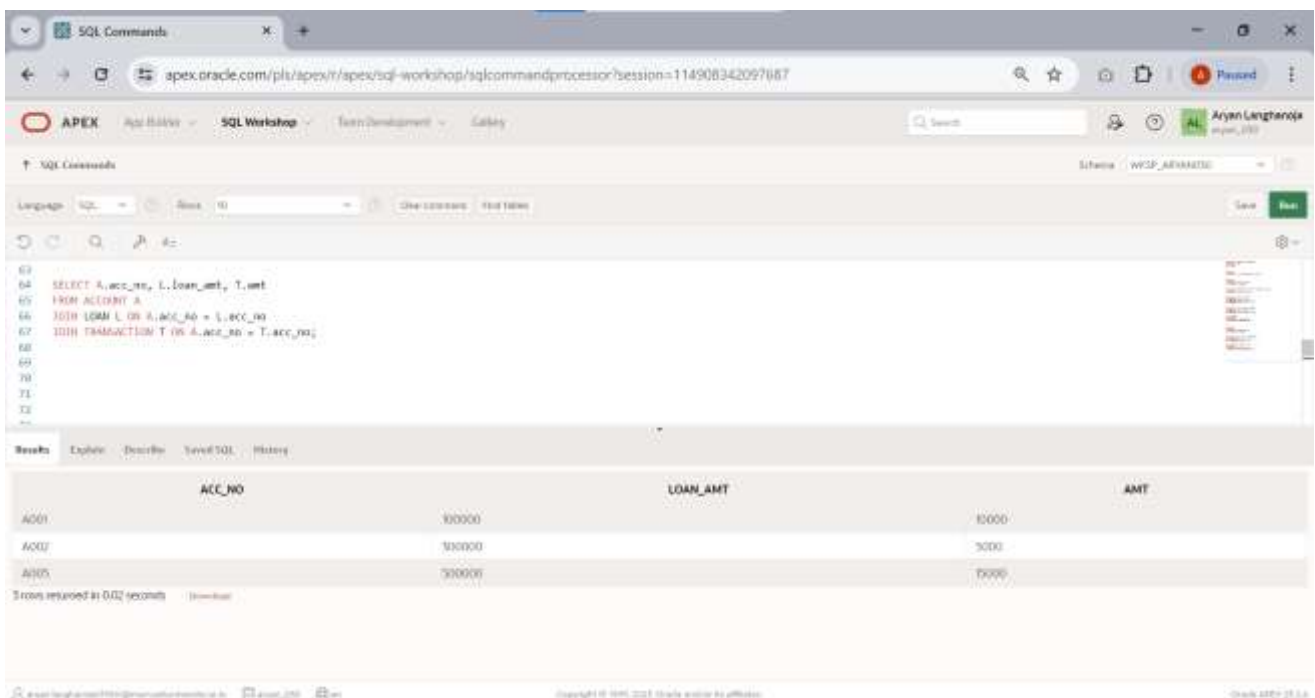
55
56
57
58 SELECT T.amt, A.name, T.acc_no, T.mode_of_pay
59 FROM TRANSACTION T
60 JOIN ACCOUNT A ON T.acc_no = A.acc_no
61 WHERE T.mode_of_pay = 'Cheque';
62
63
64
65
  
```

The results table displays the following data:

AMT	NAME	ACC_NO	MODE_OF_PAY
25000	Dave Hardik	A001	Cheque
30000	Soni Hiral	A004	Cheque
5000	Ravi Rajesh	A003	Cheque

3 rows returned in 0.05 seconds

10. Display account no, loan amount, amount of transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

63
64 SELECT A.acc_no, L.loan_amt, T.amt
65 FROM ACCOUNT A
66 JOIN LOAN L ON A.acc_no = L.acc_no
67 JOIN TRANSACTION T ON A.acc_no = T.acc_no;
68
69
70
71
72
  
```

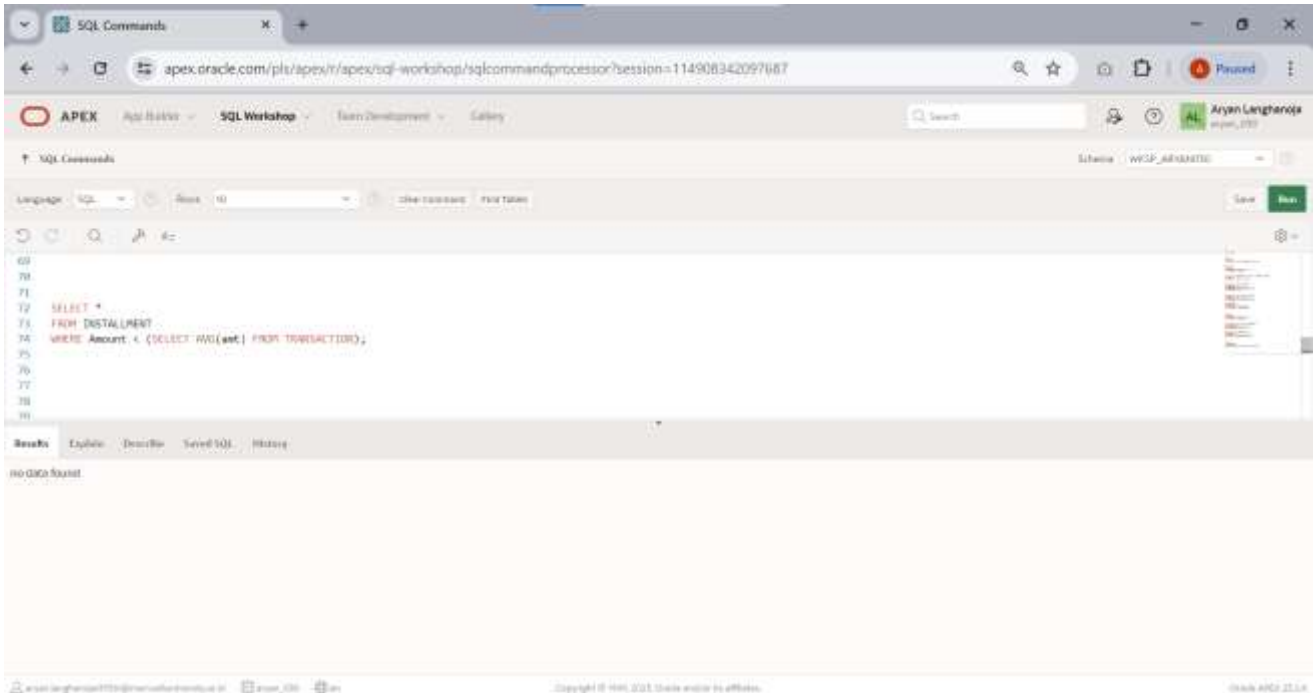
The results table displays the following data:

ACC_NO	LOAN_AMT	AMT
A001	100000	10000
A002	100000	3000
A005	100000	10000

3 rows returned in 0.02 seconds



11. List of installment information whose amount is less than average amount of transaction.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

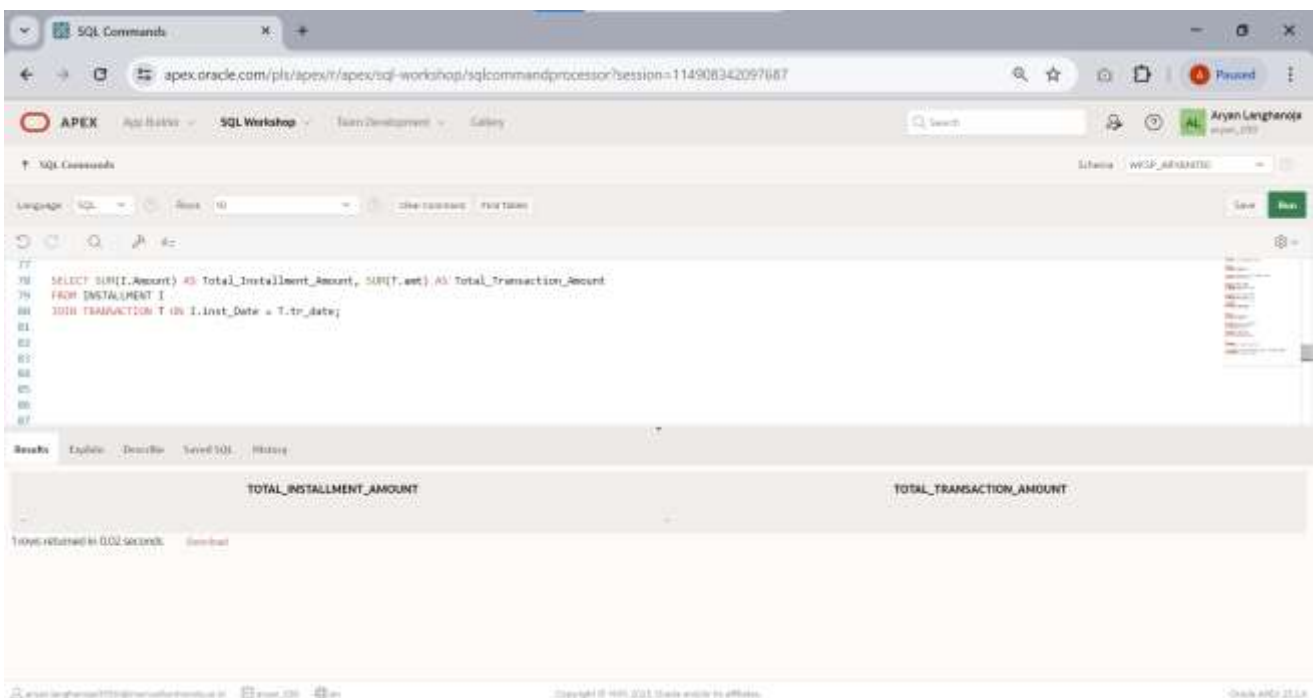
```

69
70
71
72 SELECT *
73 FROM INSTALLMENT
74 WHERE Amount < (SELECT AVG(amt) FROM TRANSACTION);
75
76
77
78
79

```

The results section shows "No Data Found".

12. Display the sum of installment amount and transaction amount.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

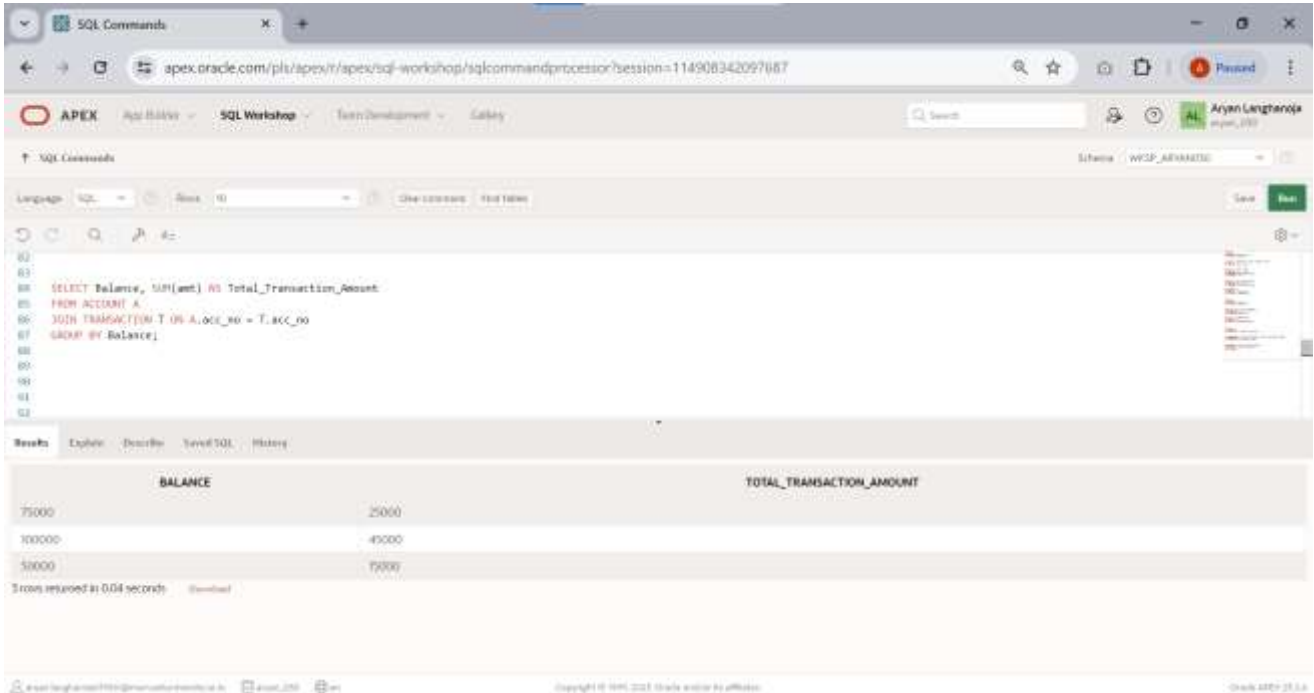
```

77
78 SELECT SUM(I.Amount) AS Total_Installment_Amount, SUM(T.amt) AS Total_Transaction_Amount
79 FROM INSTALLMENT I
80 JOIN TRANSACTION T ON I.Inst_Date = T.tr_date;
81
82
83
84
85
86
87

```

The results section shows a table with two columns: **TOTAL\_INSTALLMENT\_AMOUNT** and **TOTAL\_TRANSACTION\_AMOUNT**. The results are displayed as a single row of values.

13. Display the balance and amount of transaction group by amount and balance.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

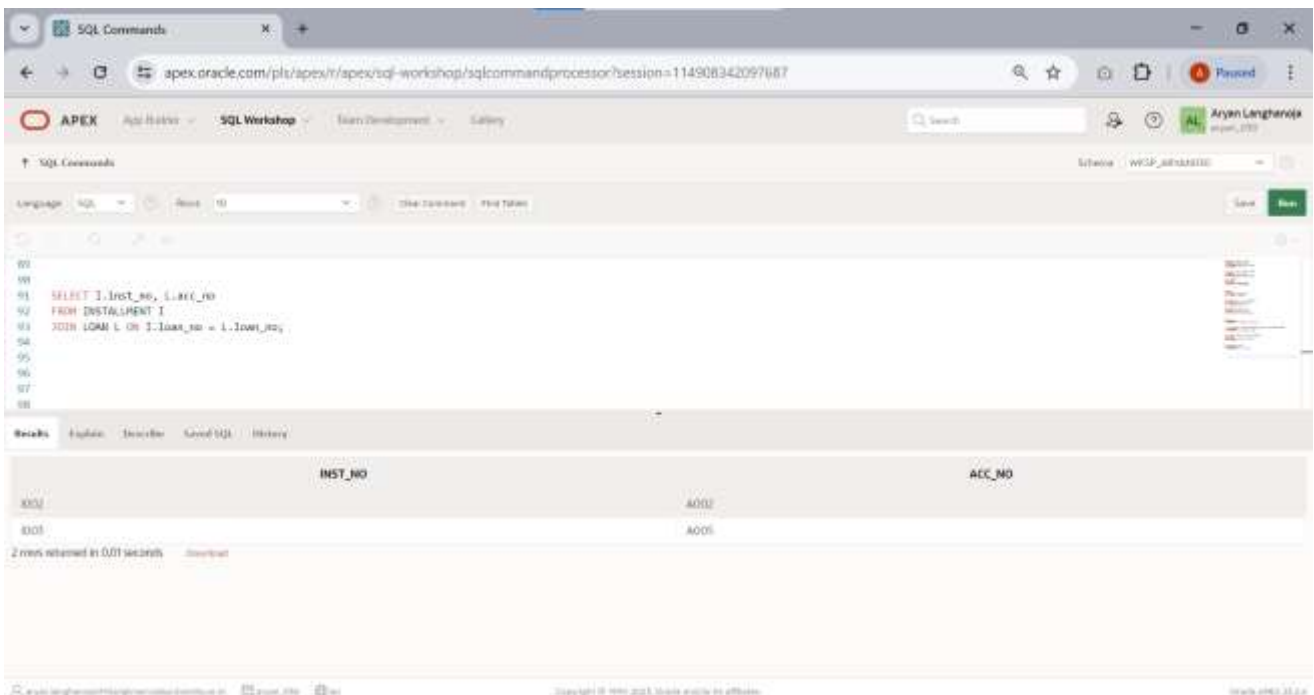
62
63
64 SELECT Balance, SUM(amt) AS Total_Transaction_Amount
65 FROM ACCOUNT A
66 JOIN TRANSACTION T ON A.ACC_NO = T.ACC_NO
67 GROUP BY Balance;
68
69
70
71
72
  
```

The results are displayed in a table with two columns: **BALANCE** and **TOTAL\_TRANSACTION\_AMOUNT**.

BALANCE	TOTAL_TRANSACTION_AMOUNT
75000	25000
100000	45000
50000	75000

3 rows returned in 0.04 seconds

14. List of installment number and account number of account holders.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

73
74
75 SELECT I.Inst_No, L.ACC_NO
76 FROM INSTALLMENT I
77 JOIN LOAN L ON I.Inst_No = L.Loan_No;
78
79
80
81
82
  
```

The results are displayed in a table with two columns: **INST\_NO** and **ACC\_NO**.

INST_NO	ACC_NO
8001	4001
8005	4005

2 rows returned in 0.01 seconds

15. Display loan amount, transaction amount and mode of payment where transaction date and loan taken date both are done in month of 'MAY'.

The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

```

104
105
106
107 SELECT L.loan_amt, T.amt, T.mode_of_pay
108 FROM LOAN L
109 JOIN TRANSACTION T ON L.acc_no = T.acc_no
110 WHERE TO_CHAR(L.loan_date, 'MM') = '05' AND TO_CHAR(T.tr_date, 'MM') = '05';
111
112
113

```

Below the query, the Results tab shows "no data found".

16. Display all the information of installment and transaction where installment date and transaction date both are done in month of 'JULY'.

The screenshot shows the APEX SQL Workshop interface. The SQL command area contains the following query:

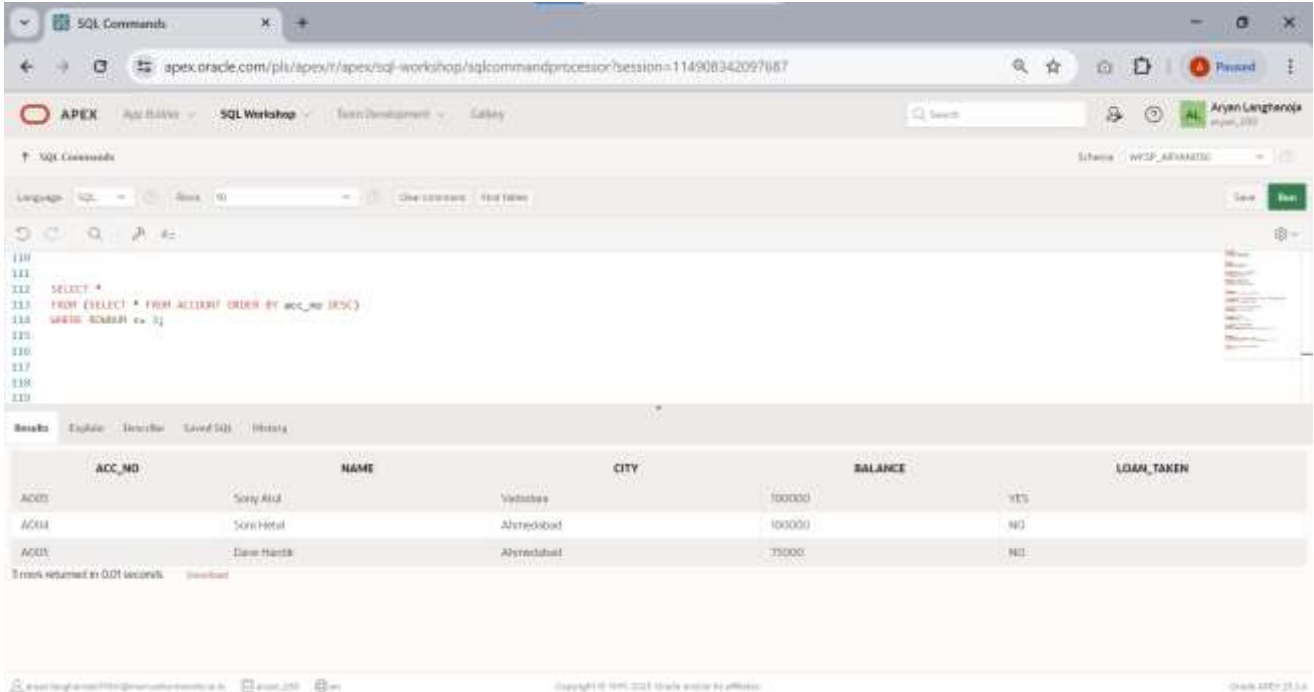
```

111
112
113
114
115 SELECT *
116 FROM INSTALLMENT I
117 JOIN TRANSACTION T ON I.inst_date = T.tr_date
118 WHERE TO_CHAR(I.inst_date, 'MM') = '07' AND TO_CHAR(T.tr_date, 'MM') = '07';
119
120
121

```

Below the query, the Results tab shows "no data found".

17. Display the last three row of account table.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

110
111
112 SELECT *
113 FROM (SELECT * FROM ACCOUNT ORDER BY acc_no DESC)
114 WHERE ROWNUM <= 3;
115
116
117
118
119

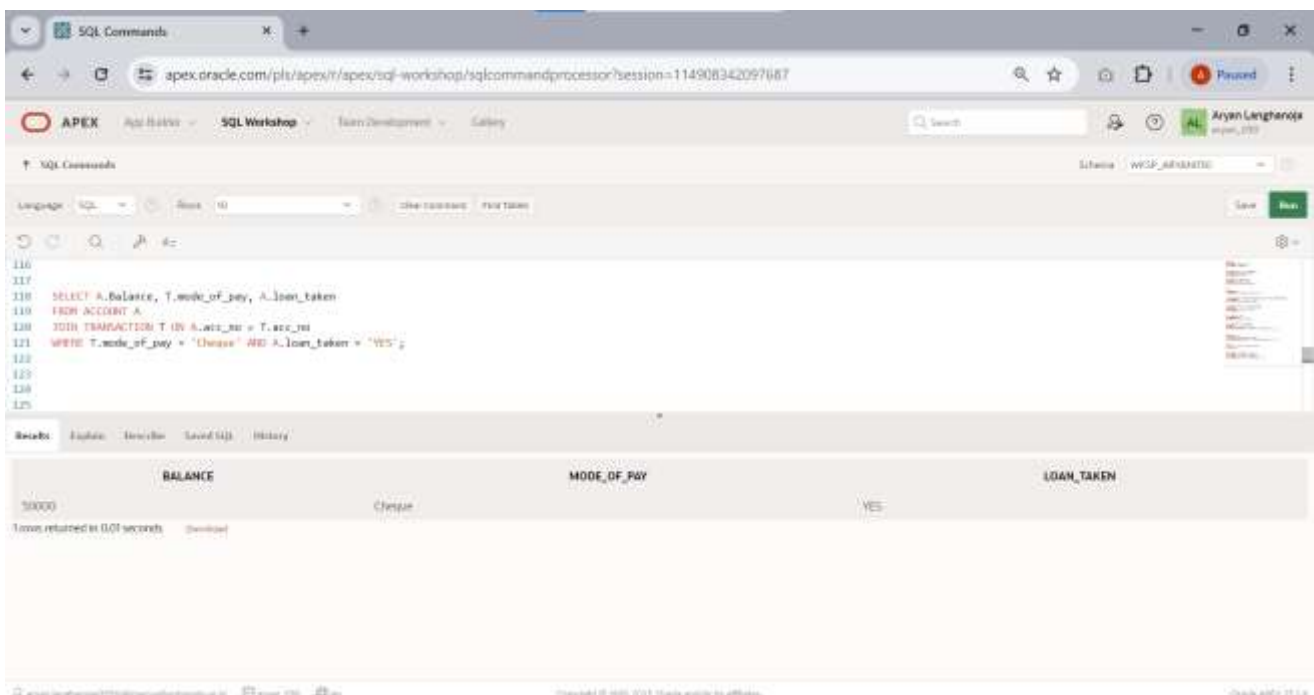
```

The results table displays the last three rows of the ACCOUNT table:

ACC_NO	NAME	CITY	BALANCE	LOAN_TAKEN
AC003	Sony Akl	Vadodra	100000	YES
AC004	Soni Hetal	Ahmedabad	100000	NO
AC005	Dave Hantik	Ahmedabad	75000	NO

Time returned in 0.01 seconds.

18. Display the balance, mode of payment, loan taken status whose mode of payment is 'CHEQUE' and loan taken is 'YES'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

110
111
112 SELECT A.balance, T.mode_of_pay, A.loan_taken
113 FROM ACCOUNT A
114 JOIN TRANSACTION T ON A.acc_no = T.acc_no
115 WHERE T.mode_of_pay = 'Cheque' AND A.loan_taken = 'YES';
116
117
118
119
120

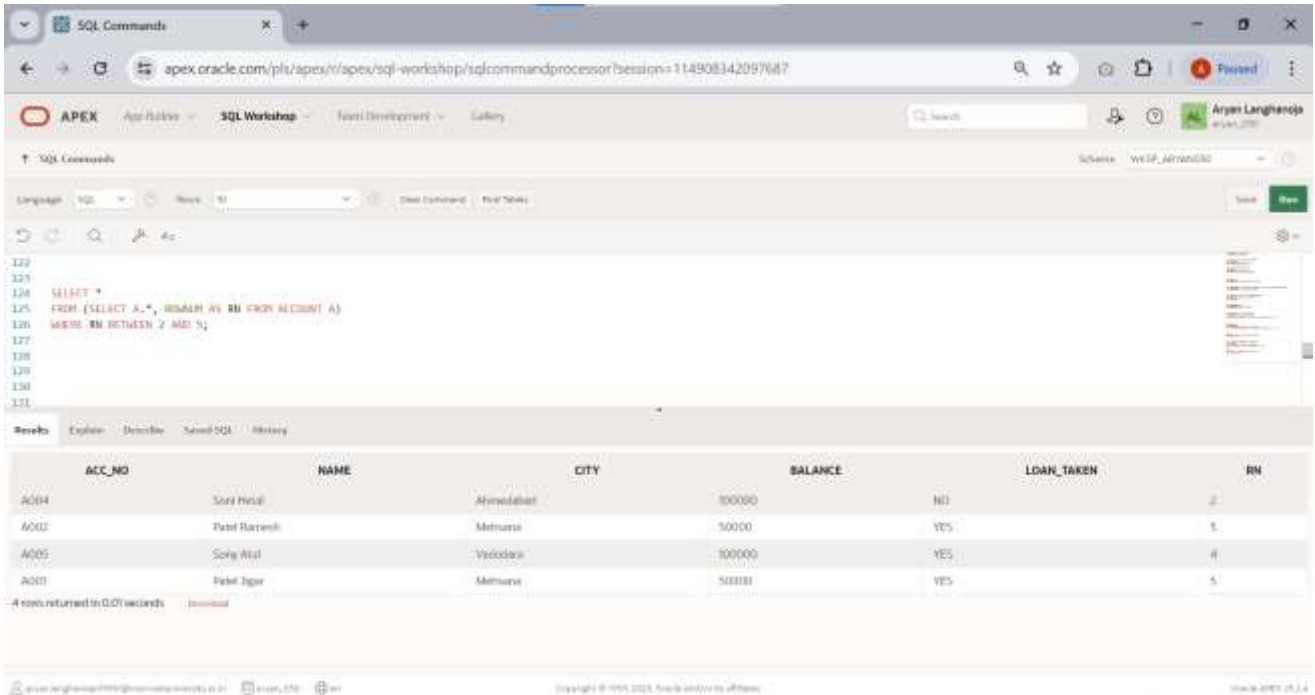
```

The results table displays the filtered data:

BALANCE	MODE_OF_PAY	LOAN_TAKEN
100000	Cheque	YES

Time returned in 0.01 seconds.

19. Retrieve only rows 2 to 5 from account table.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
SELECT *
FROM (SELECT A.*, ROWNUM AS RN FROM ACCOUNT A)
WHERE RN BETWEEN 2 AND 5;
```

The results table displays the following data:

ACC_NO	NAME	CITY	BALANCE	LOAN TAKEN	RN
A004	Soni Hetal	Amnabad	100000	NO	2
A002	Ravi Ramesh	Mumbai	50000	YES	3
A005	Soni Atul	Vadodra	100000	YES	4
A001	Ravi Jigar	Mumbai	50000	YES	5

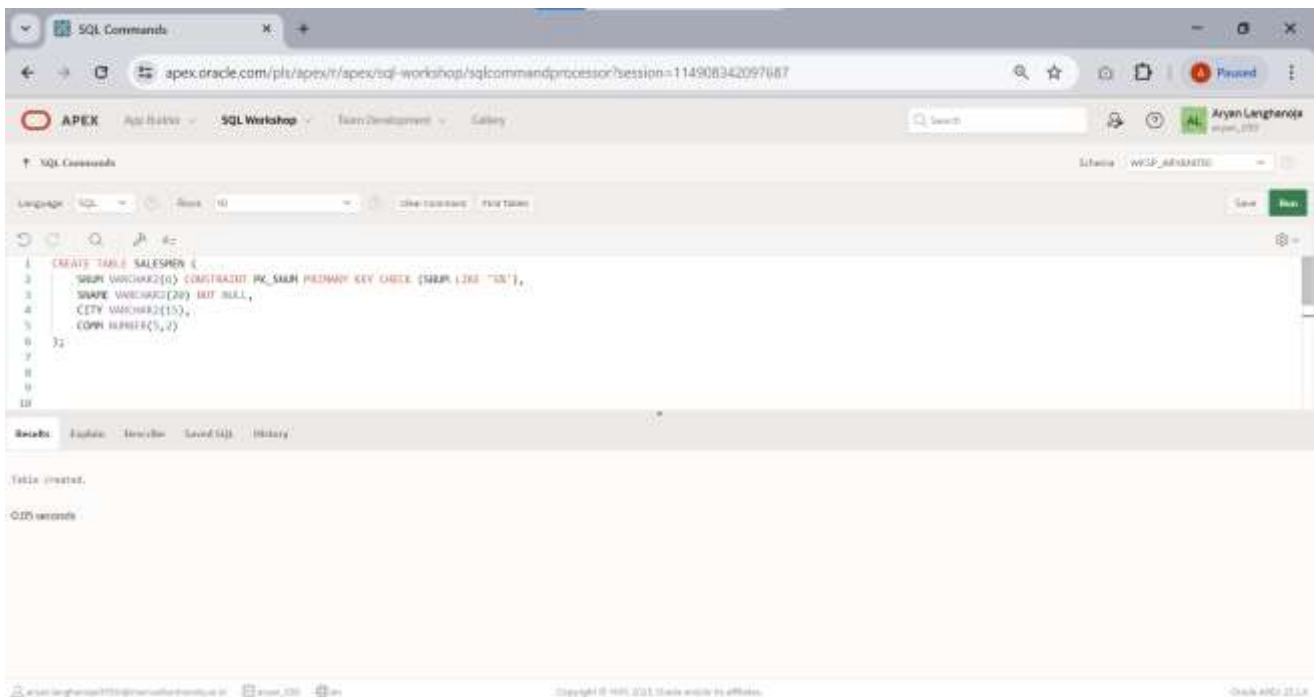
4 rows returned in 0.01 seconds

## Practical 8

**Aim: Implement Constraint based and Group by related queries.**

TABLE: SALESMEN

Column Name	Data Type	Size	Attributes
SNUM	Varchar2	6	Primary key/first letter must start with 'S'
SNAME	Varchar2	20	Not null
CITY	Varchar2	15	
COMM	Number	5,2	



The screenshot shows the SQL Workshop interface in Oracle APEX. The SQL Commands window contains the following code:

```

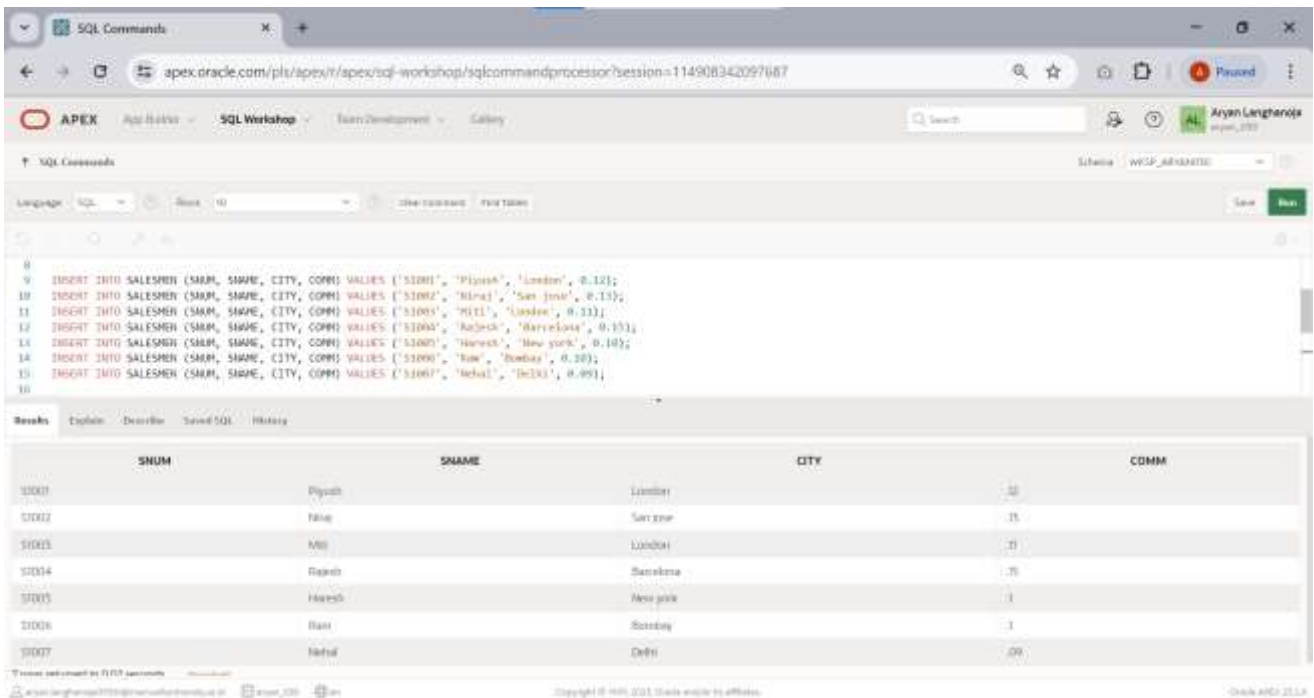
1 CREATE TABLE SALESMEN (
2   SNUM VARCHAR2(6) CONSTRAINT PK_SNUM PRIMARY KEY CHECK (SNUM LIKE 'S%'),
3   SNAME VARCHAR2(20) NOT NULL,
4   CITY VARCHAR2(15),
5   COMM NUMBER(5,2)
6 );
7
8
9
10

```

The Results window shows the message: "Table created."

Insert the following records:

SNUM	SNAME	CITY	COMM
S1001	Piyush	London	0.12
S1002	Niraj	San jose	0.13
S1003	Miti	London	0.11
S1004	Rajesh	Barcelo na	0.15
S1005	Haresh	New york	0.10
S1006	Ram	Bombay	0.10
S1007	Nehal	Delhi	0.09



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

9  INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1001', 'Piyush', 'London', 0.12);
10 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1002', 'Niraj', 'San jose', 0.13);
11 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1003', 'Miti', 'London', 0.11);
12 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1004', 'Rajesh', 'Barcelona', 0.15);
13 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1005', 'Haresh', 'New york', 0.10);
14 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1006', 'Ram', 'Bombay', 0.10);
15 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM) VALUES ('S1007', 'Nehal', 'Delhi', 0.09);
16

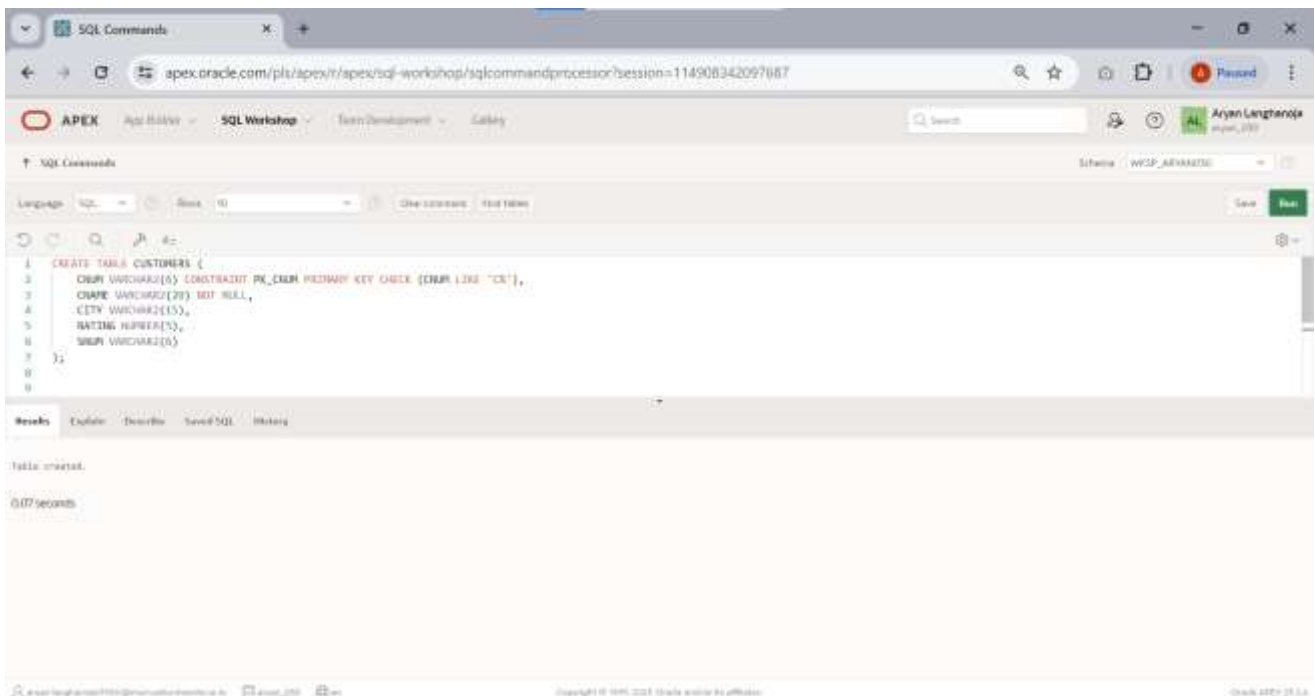
```

The results table shows the following data:

SNUM	SNAME	CITY	COMM
S1001	Piyush	London	0.12
S1002	Niraj	San jose	0.13
S1003	Miti	London	0.11
S1004	Rajesh	Barcelona	0.15
S1005	Haresh	New york	0.10
S1006	Ram	Bombay	0.10
S1007	Nehal	Delhi	0.09

**TABLE: CUSTOMER**

Column Name	Data Type	Size	Attributes
CNUM	Varchar2	6	Primary key/first letter must start with 'C'
CNAME	Varchar2	20	Not null
CITY	Varchar2	15	
RATING	Number	5	
SNUM	Varchar2	6	



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command editor contains the following code:

```

1 CREATE TABLE CUSTOMER (
2   CNUM VARCHAR2(6) CONSTRAINT PK_CNUM PRIMARY KEY CHECK (CNUM LIKE 'C%'),
3   CNAME VARCHAR2(20) NOT NULL,
4   CITY VARCHAR2(15),
5   RATING NUMBER(5),
6   SNUM VARCHAR2(6)
7 );
8
9
10

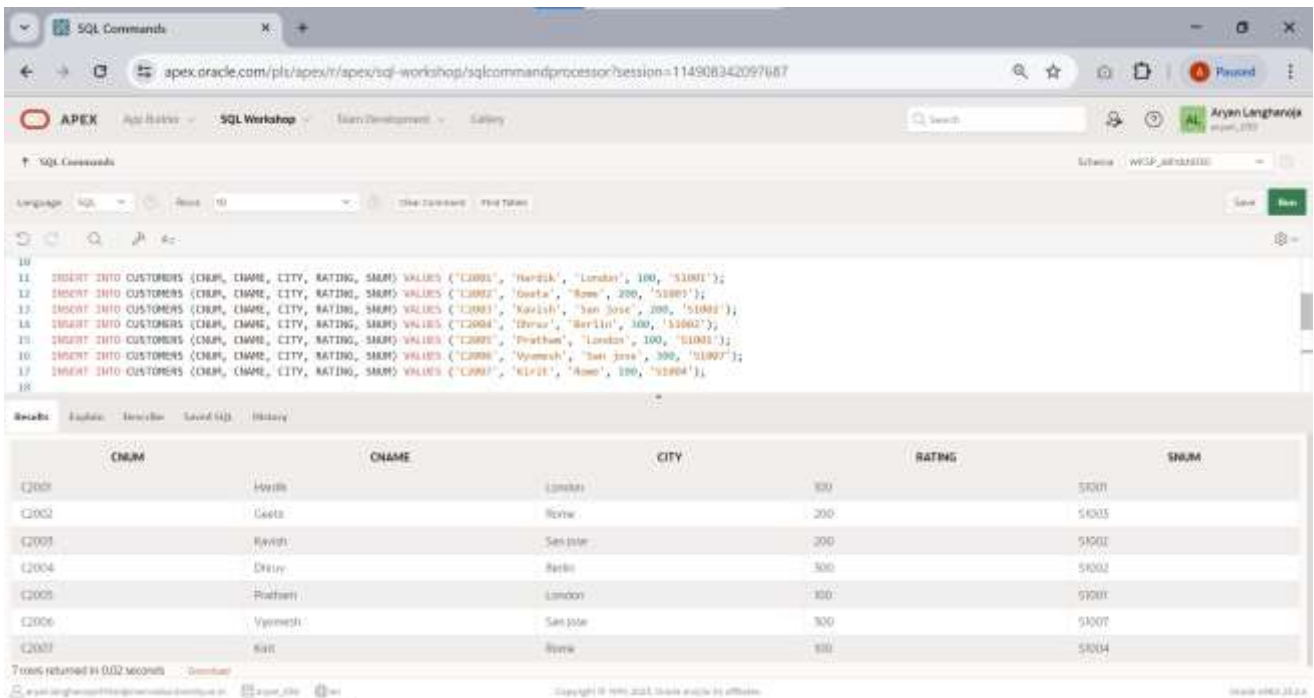
```

The Results pane shows the message: "Table created." and the execution time: "0.07 seconds".



Insert the following records

CNUM	CNAME	CITY	RATING	SNUM
C2001	Hardik	London	100	S1001
C2002	Geeta	Rome	200	S1003
C2003	Kavish	San jose	200	S1002
C2004	Dhruv	Berlin	300	S1002
C2005	Pratham	London	100	S1001
C2006	Vyomesh	San jose	300	S1007
C2007	Kirit	Rome	100	S1004



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command executed is:

```

11 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2001', 'Hardik', 'London', 100, 'S1001');
12 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2002', 'Geeta', 'Rome', 200, 'S1003');
13 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2003', 'Kavish', 'San Jose', 200, 'S1002');
14 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2004', 'Dhruv', 'Berlin', 300, 'S1002');
15 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2005', 'Pratham', 'London', 100, 'S1001');
16 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2006', 'Vyomesh', 'San Jose', 300, 'S1007');
17 INSERT INTO CUSTOMERS (CNUM, CNAME, CITY, RATING, SNUM) VALUES ('C2007', 'Kirit', 'Rome', 100, 'S1004');
18

```

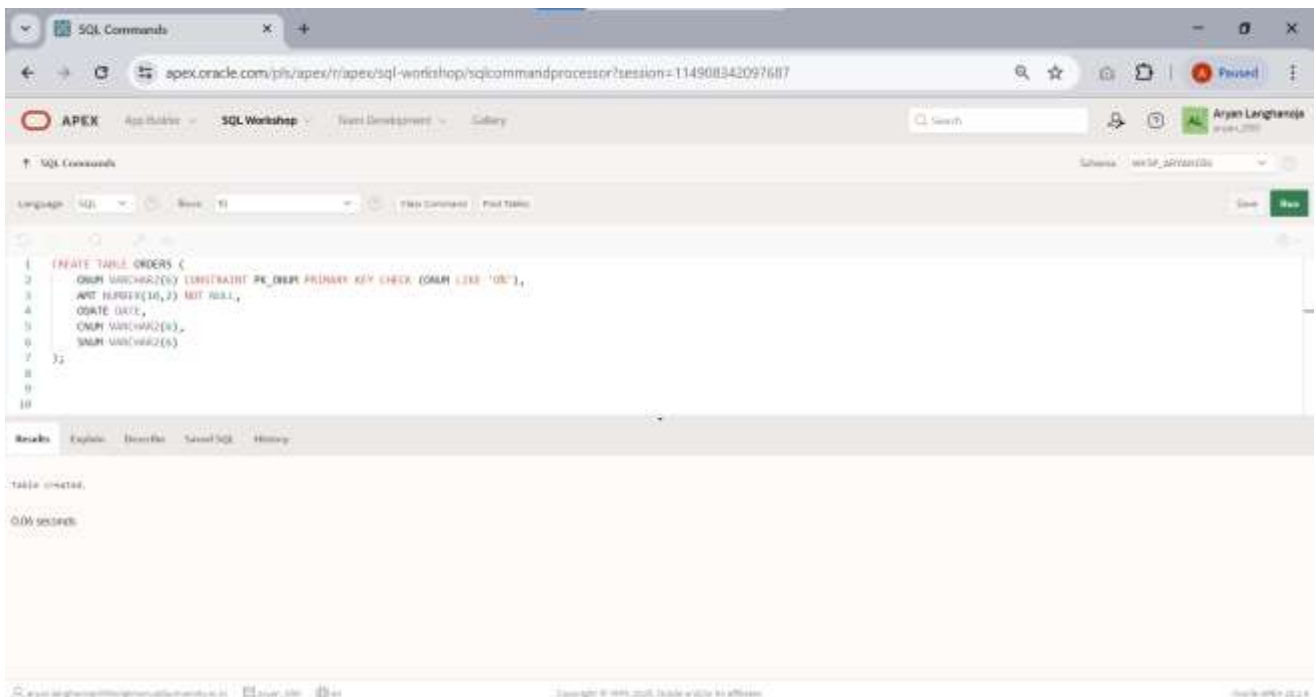
The results table shows the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2001	Hardik	London	100	S1001
C2002	Geeta	Rome	200	S1003
C2003	Kavish	San Jose	200	S1002
C2004	Dhruv	Berlin	300	S1002
C2005	Pratham	London	100	S1001
C2006	Vyomesh	San Jose	300	S1007
C2007	Kirit	Rome	100	S1004

7 rows returned in 0.02 seconds.

**TABLE: ORDER**

Column Name	Data Type	Size	Attributes
ONUM	Varchar2	6	Primary key/first letter must start with 'O'
AMT	Number	10, 2	Not null
ODATE	Date		
CNUM	Varchar2	6	
SNUM	Varchar2	6	



The screenshot shows the SQL Workshop interface in Oracle APEX. The SQL command window contains the following code:

```

1 CREATE TABLE ORDERS (
2   ONUM VARCHAR2(6) CONSTRAINT PK_ORDERS PRIMARY KEY CHECK (ONUM LIKE 'O%'),
3   AMT NUMBER(10,2) NOT NULL,
4   ODATE DATE,
5   CNUM VARCHAR2(6),
6   SNUM VARCHAR2(6)
7 );
8
9
10

```

The Results window shows the execution output:

```

Table created.
0.06953468

```

Insert the following records

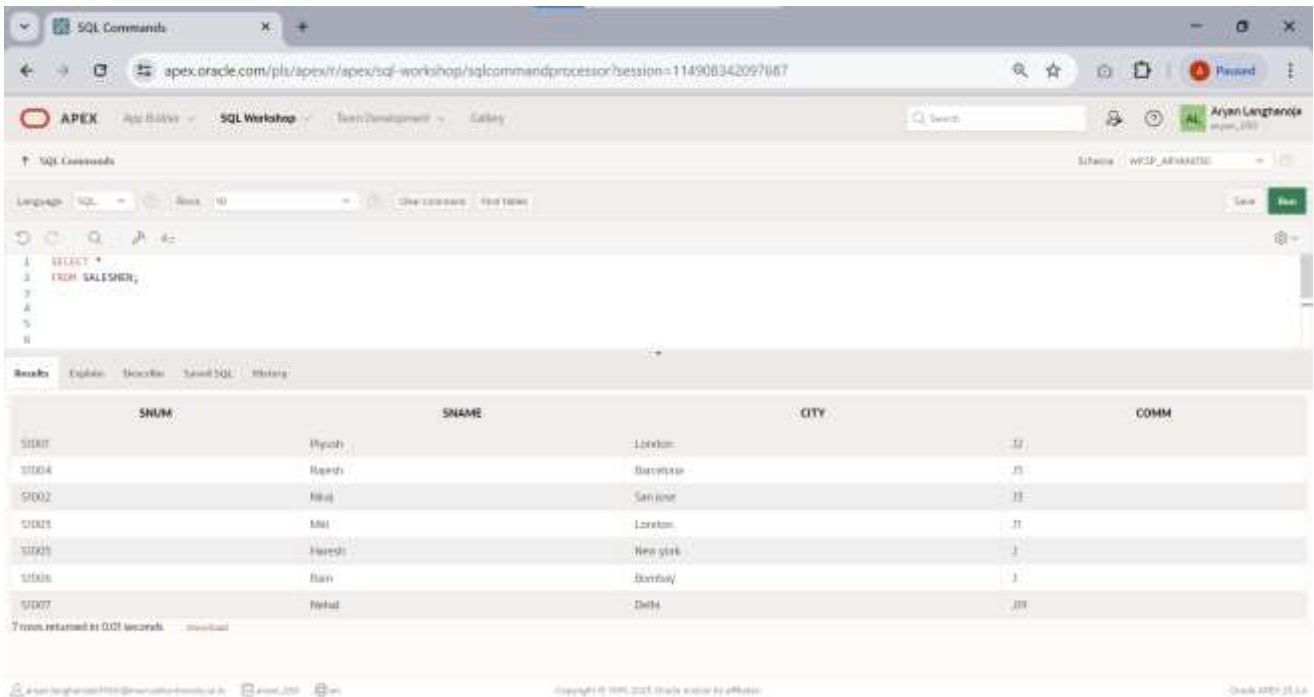
ONUM	AMT	ODATE	CNUM	SNUM
O3001	18.69	10-Mar-90	C2008	S1007
O3003	767.19	10-Mar-90	C2001	S1001
O3002	1900.10	03-Oct-90	C2007	S1004
O3005	5160.45	04-Oct-90	C2003	S1002
O3006	1098.16	10-Mar_90	C2008	S1007
O3009	1713.23	10-April-90	C2002	S1003
O3007	75.75	10-April-90	C2004	S1002
O3008	4723.00	10-May-90	C2006	S1001
O3010	1309.95	10-May-90	C2004	S1002
O3011	9891.88	10-June-90	C2006	S1001

The screenshot shows the APEX SQL Workshop interface. The top section displays the 'SQL Commands' window with a session ID. Below this, the 'Table Editor' window shows a table with the following columns: ONUM, AMT, ODATE, CNUM, and SNUM. The table contains 11 rows of data, which are the records to be inserted. The bottom section shows the 'Results' window, which displays the data in a grid format, confirming the insertion of the records.

**Perform following queries.**

## SELECT

1. Display all the information of salesmen.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

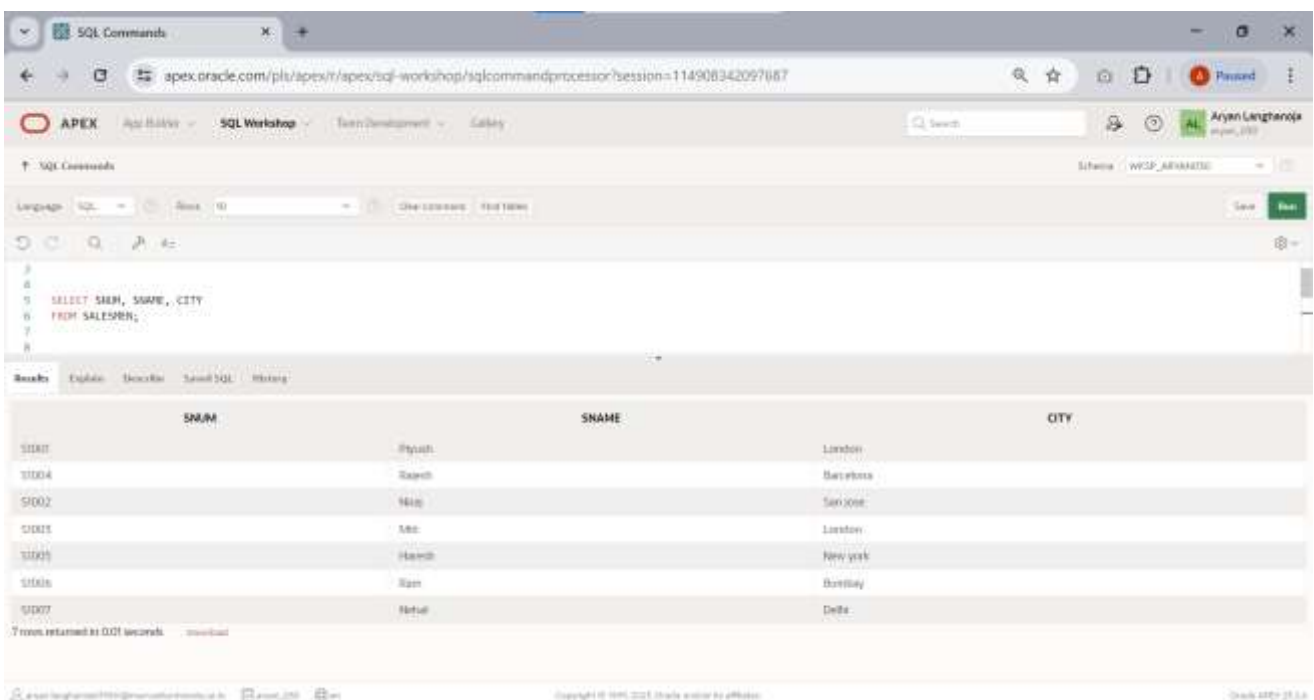
```
SELECT *
FROM SALESMEN;
```

The results are displayed in a table with the following columns: SNUM, SNAME, CITY, and COMM. There are 7 rows of data.

SNUM	SNAME	CITY	COMM
S0001	Riyadh	London	12
S0004	Riyadh	Barcelona	25
S0002	Riyadh	San Jose	15
S0003	Mex	London	21
S0005	Riyadh	New York	3
S0006	Riyadh	Bombay	1
S0007	Riyadh	Delhi	20

7 rows returned in 0.02 seconds

2. Display snum,sname,city from salesmen table.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

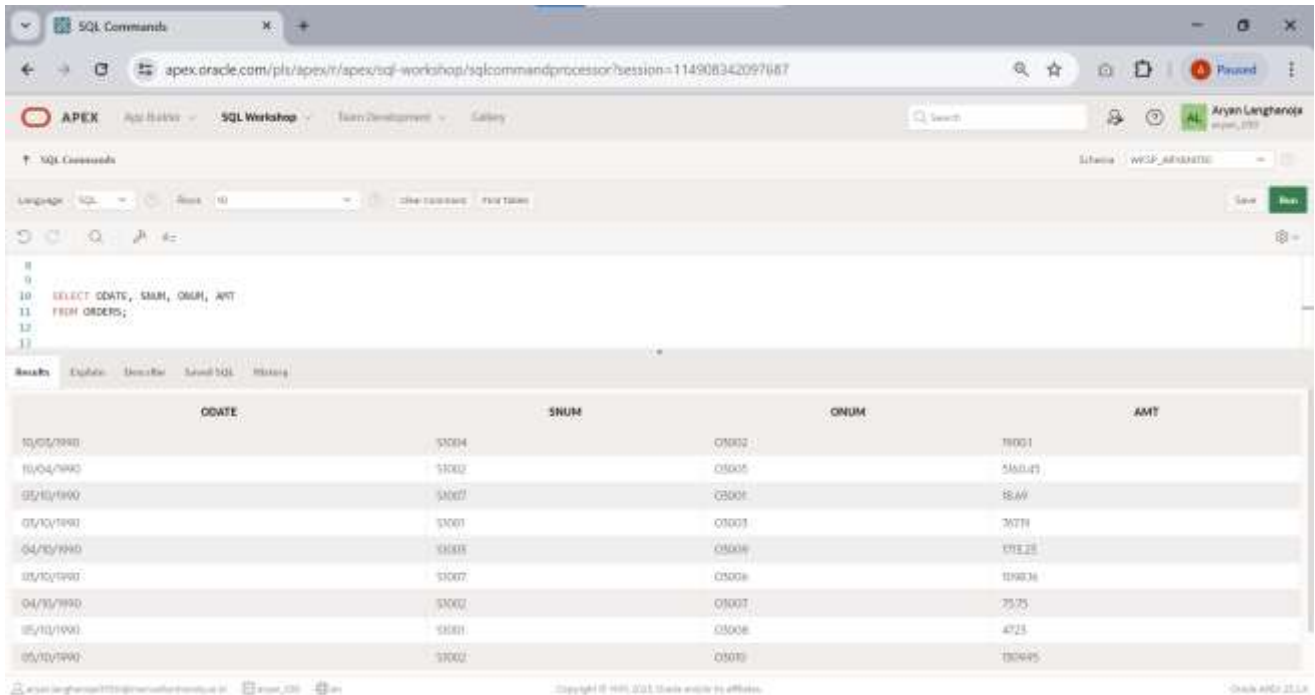
```
SELECT SNUM, SNAME, CITY
FROM SALESMEN;
```

The results are displayed in a table with the following columns: SNUM, SNAME, and CITY. There are 7 rows of data.

SNUM	SNAME	CITY
S0001	Riyadh	London
S0004	Riyadh	Barcelona
S0002	Riyadh	San Jose
S0003	Mex	London
S0005	Riyadh	New York
S0006	Riyadh	Bombay
S0007	Riyadh	Delhi

7 rows returned in 0.02 seconds

3. Display odate,snum,onum and amt from orders.



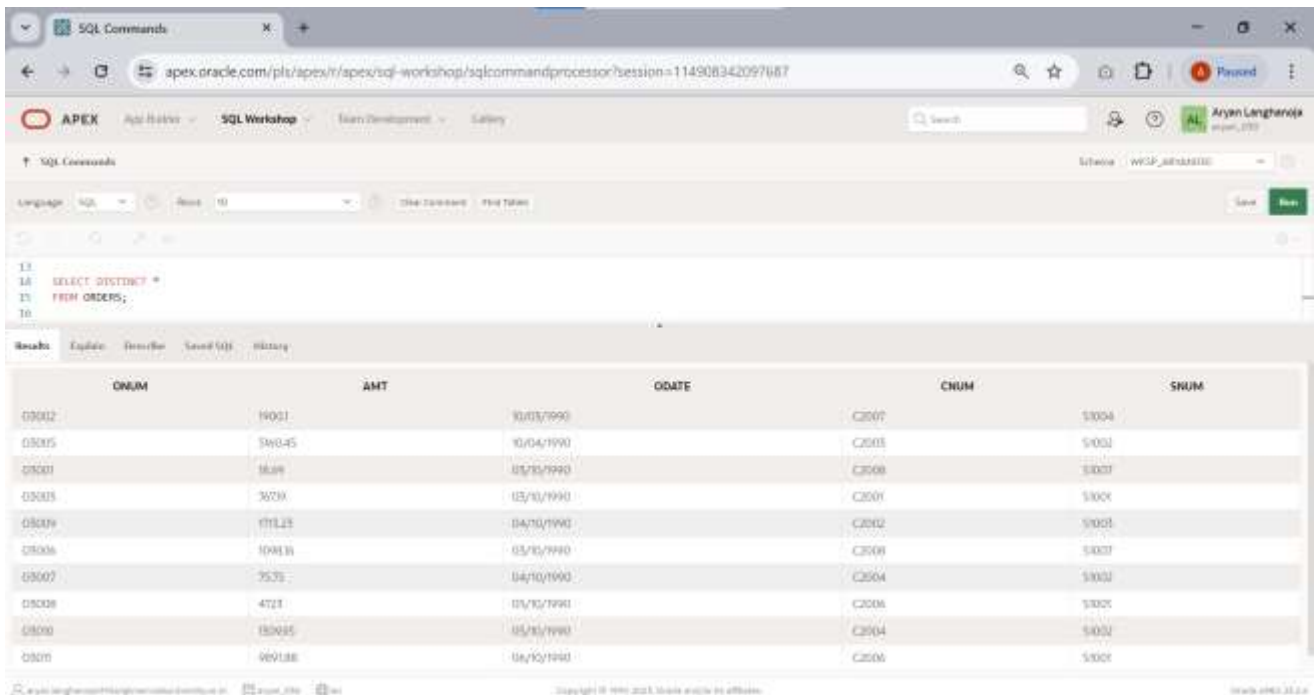
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT ODATE, SNUM, ONUM, AMT
FROM ORDERS;
```

The results are displayed in a table with the following columns: ODATE, SNUM, ONUM, and AMT. The data is as follows:

ODATE	SNUM	ONUM	AMT
10/03/1990	S004	O002	19001
10/04/1990	S002	O005	560.45
05/10/1990	S007	O001	18.69
01/10/1990	S001	O003	30711
04/10/1990	S003	O009	1711.23
05/10/1990	S007	O008	1090.31
04/10/1990	S002	O007	75.75
05/10/1990	S001	O006	4721
05/10/1990	S002	O010	130945

4. Display the information of orders without duplication.



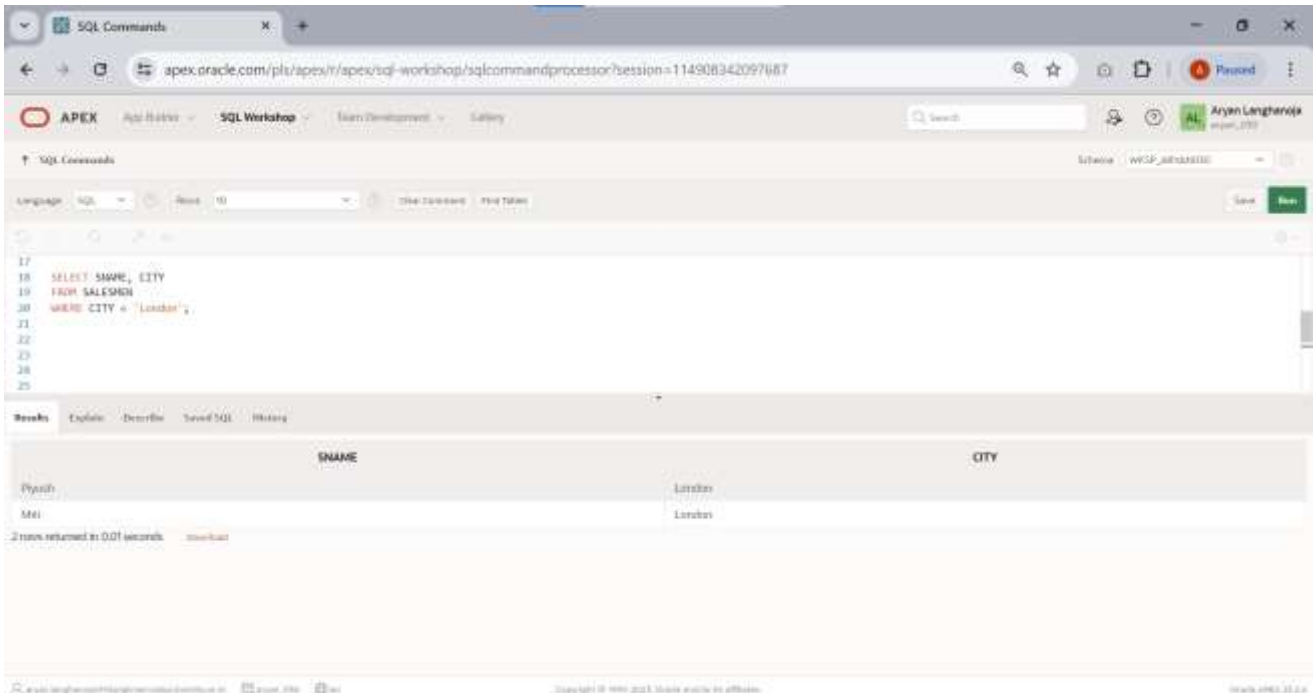
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT DISTINCT *
FROM ORDERS;
```

The results are displayed in a table with the following columns: ONUM, AMT, ODATE, CNUM, and SNUM. The data is as follows:

ONUM	AMT	ODATE	CNUM	SNUM
O002	19001	10/03/1990	C007	S004
O005	560.45	10/04/1990	C005	S002
O001	18.69	05/10/1990	C008	S007
O003	30711	01/10/1990	C001	S001
O009	1711.23	04/10/1990	C002	S003
O006	1090.31	05/10/1990	C006	S007
O007	75.75	04/10/1990	C004	S002
O008	4721	05/10/1990	C006	S001
O010	130945	05/10/1990	C004	S002
O004	969188	08/10/1990	C006	S001

5. List of sname, city from salesmen where city is 'LONDON'.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

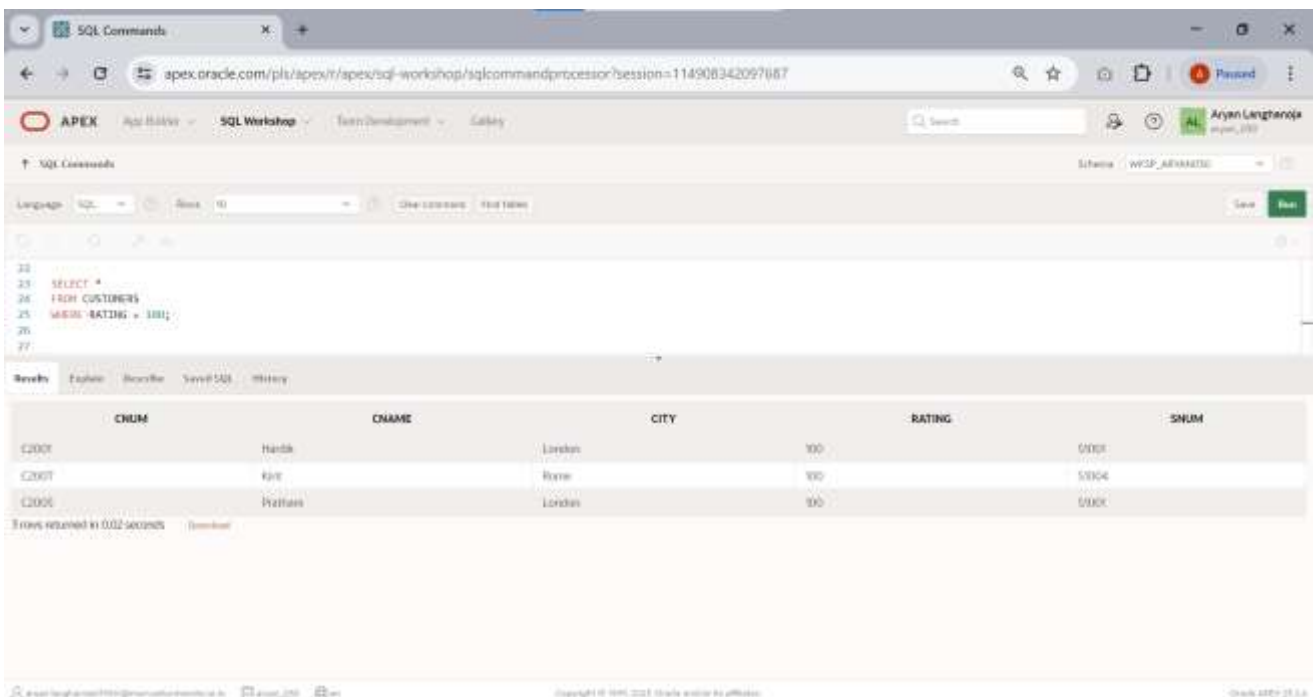
17
18 SELECT SNAME, CITY
19 FROM SALESMEN
20 WHERE CITY = 'LONDON';
21
22
23
24
25
  
```

The results are displayed in a table with columns SNAME and CITY:

SNAME	CITY
Pyramid	LONDON
Mel	LONDON

2 rows returned in 0.01 seconds.

6. List all records of customers where rating is equal to 100.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

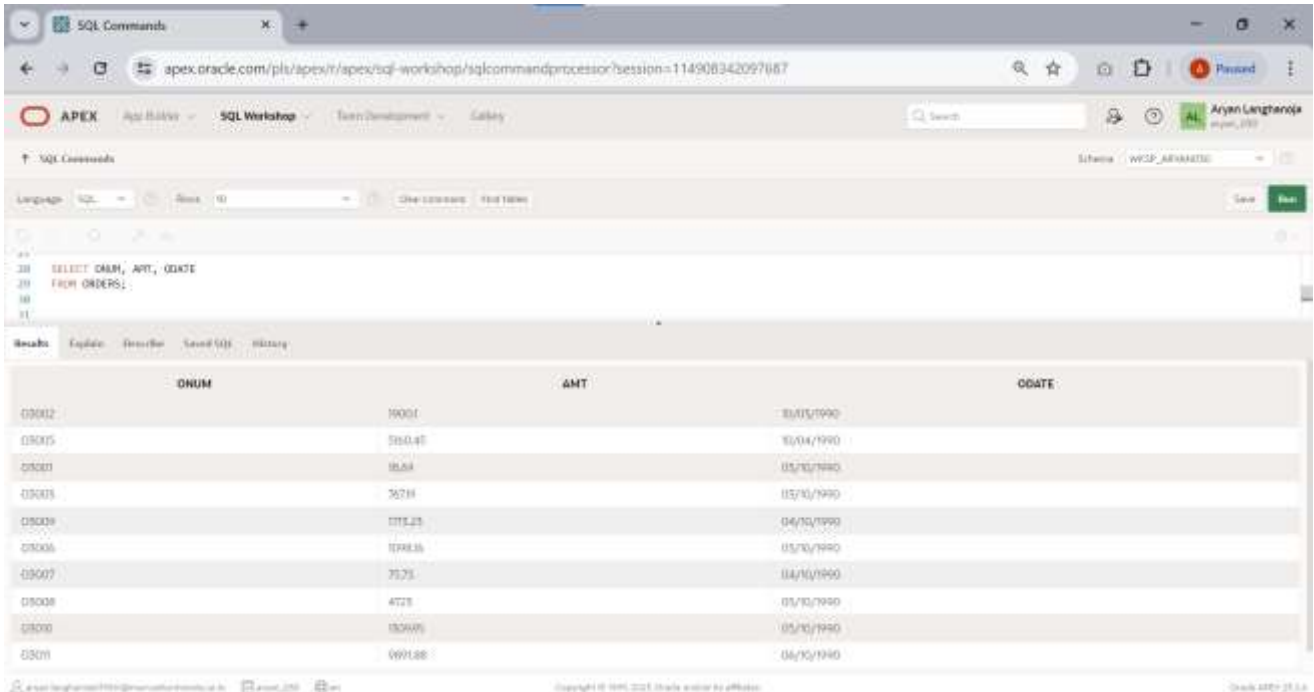
22
23 SELECT *
24 FROM CUSTOMERS
25 WHERE RATING = 100;
26
27
  
```

The results are displayed in a table with columns CNUM, CNAME, CITY, RATING, and SNUM:

CNUM	CNAME	CITY	RATING	SNUM
C2001	Hardik	LONDON	100	00001
C2007	Ravi	BANG	100	00004
C2006	Pratham	LONDON	100	00003

3 rows returned in 0.02 seconds.

7. Write a select command that produces the order number, amount and date for all rows in the order table.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

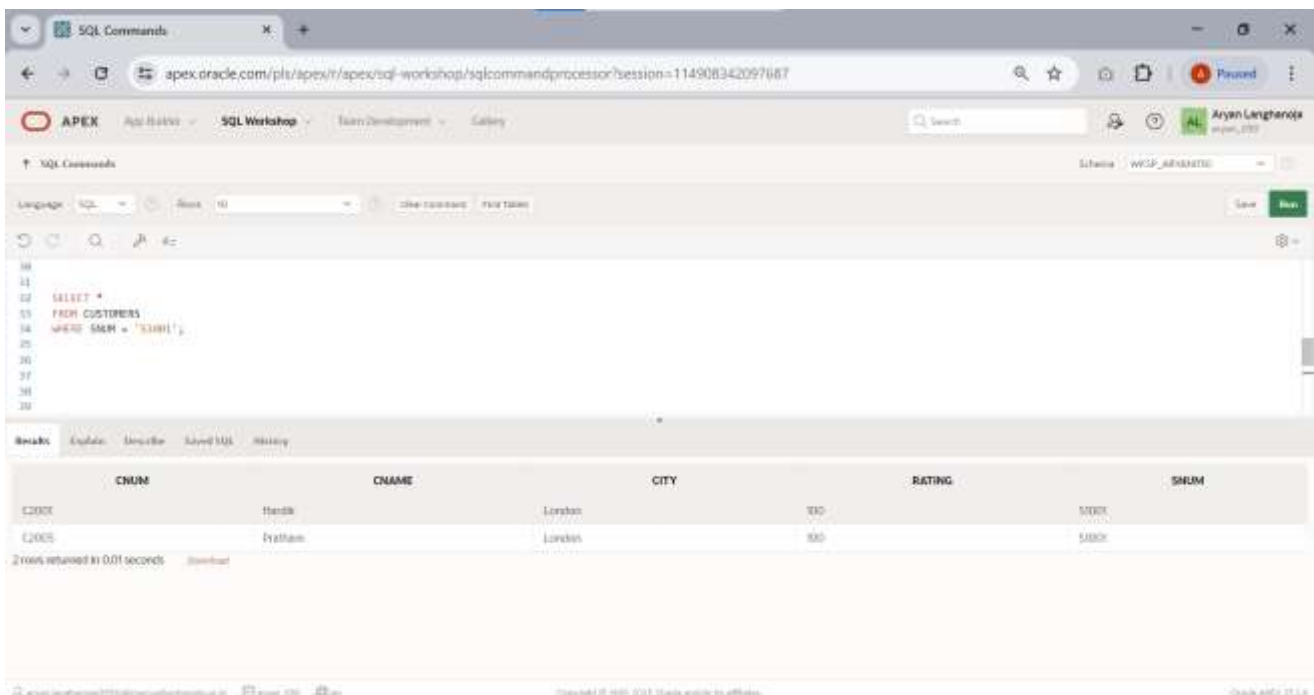
SELECT ONUM, AMT, ODATE
FROM ORDERS;

```

The results are displayed in a table with the following columns: ONUM, AMT, and ODATE. The data is as follows:

ONUM	AMT	ODATE
03002	99001	03/03/1990
03005	590.47	03/04/1990
03007	88.64	05/03/1990
03008	567.6	05/03/1990
03009	078.25	04/03/1990
03006	0798.36	05/03/1990
03007	75.75	04/03/1990
03008	4728	05/03/1990
03009	030605	05/03/1990
03001	6691.88	04/03/1990

8. Produces all rows from the customer table for which the salesperson's number is S1001.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

SELECT *
FROM CUSTOMERS
WHERE SALESPERSON = 'S1001';

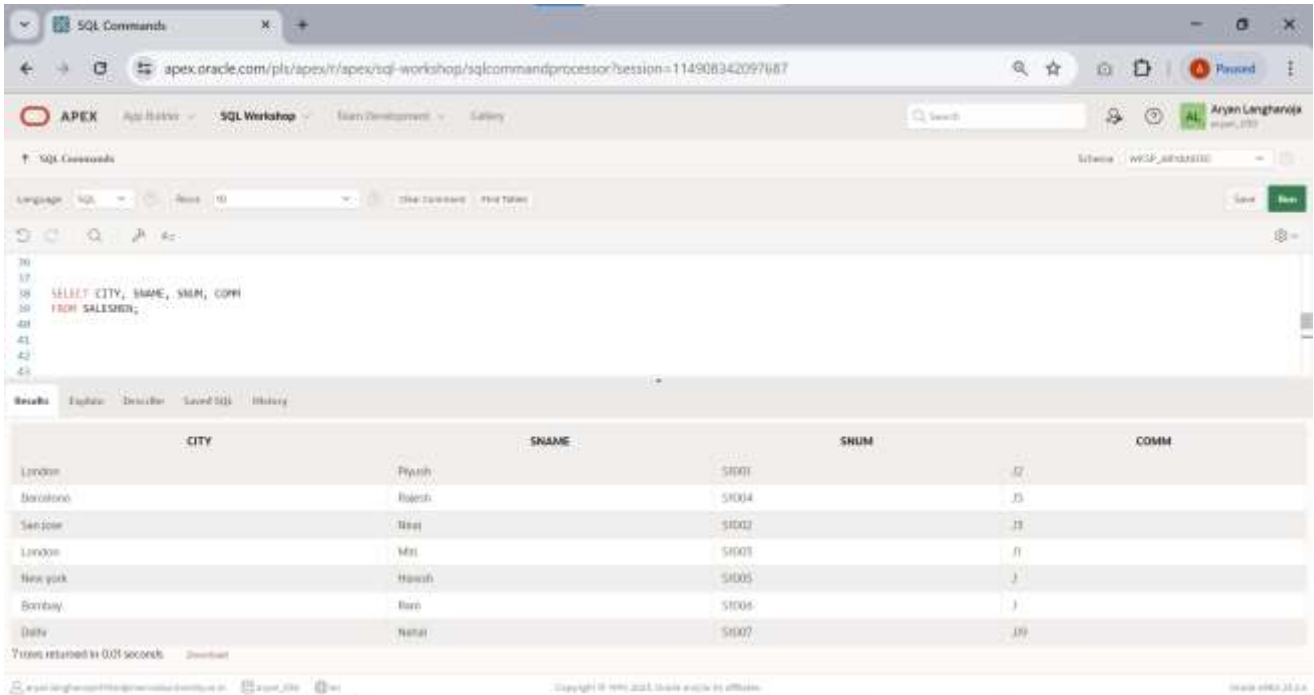
```

The results are displayed in a table with the following columns: CNUM, CNAME, CITY, RATING, and SNUM. The data is as follows:

CNUM	CNAME	CITY	RATING	SNUM
C2001	Harold	London	100	M001
C2005	Pythian	London	800	S000

2 rows returned in 0.01 seconds

9. Display the salesperson table with the column in the following order: city,sname,snum,comm.



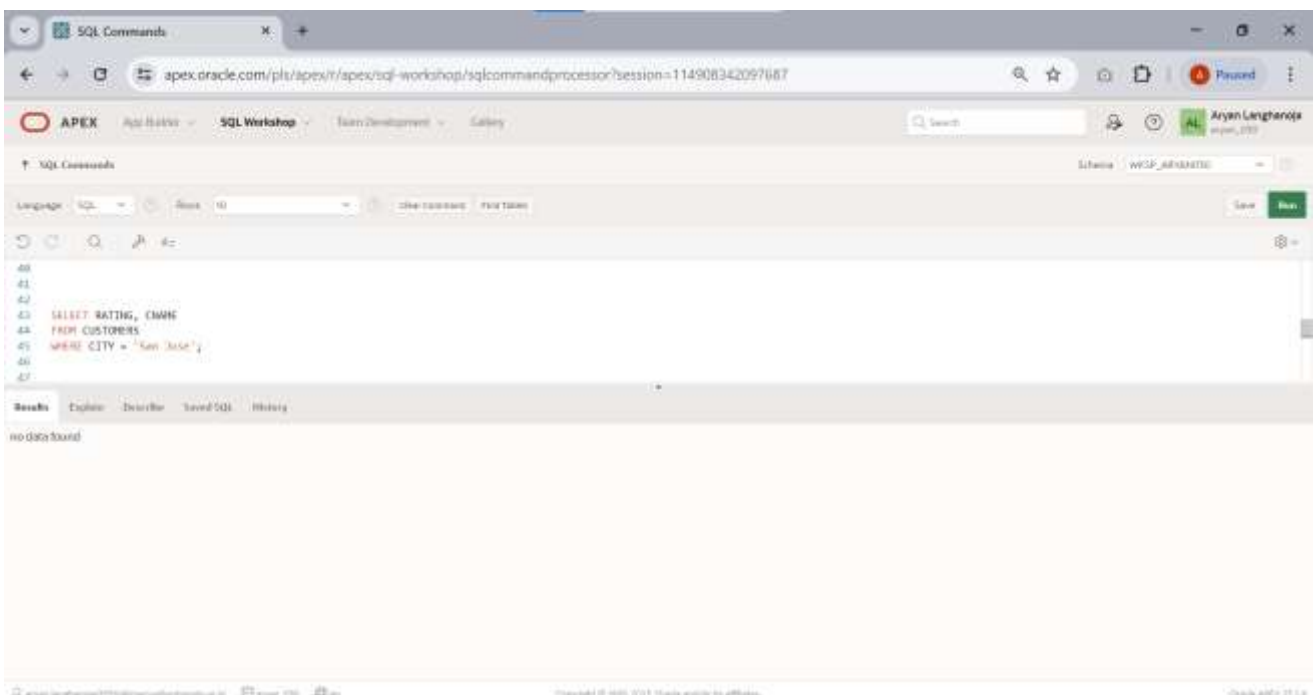
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
SELECT CITY, SNAME, SNUM, COMM
FROM SALESPER;
```

The results are displayed in a table with the following columns: CITY, SNAME, SNUM, and COMM. The data is as follows:

CITY	SNAME	SNUM	COMM
London	Peter	5001	12
Boston	Robert	5004	25
San Jose	Neel	5002	18
London	Mel	5003	11
New York	Harold	5005	3
Bombay	Rao	5006	1
Dalh	Natal	5007	19

10. Write a select command that produces the rating followed by the name of each customer in SAN JOSE.



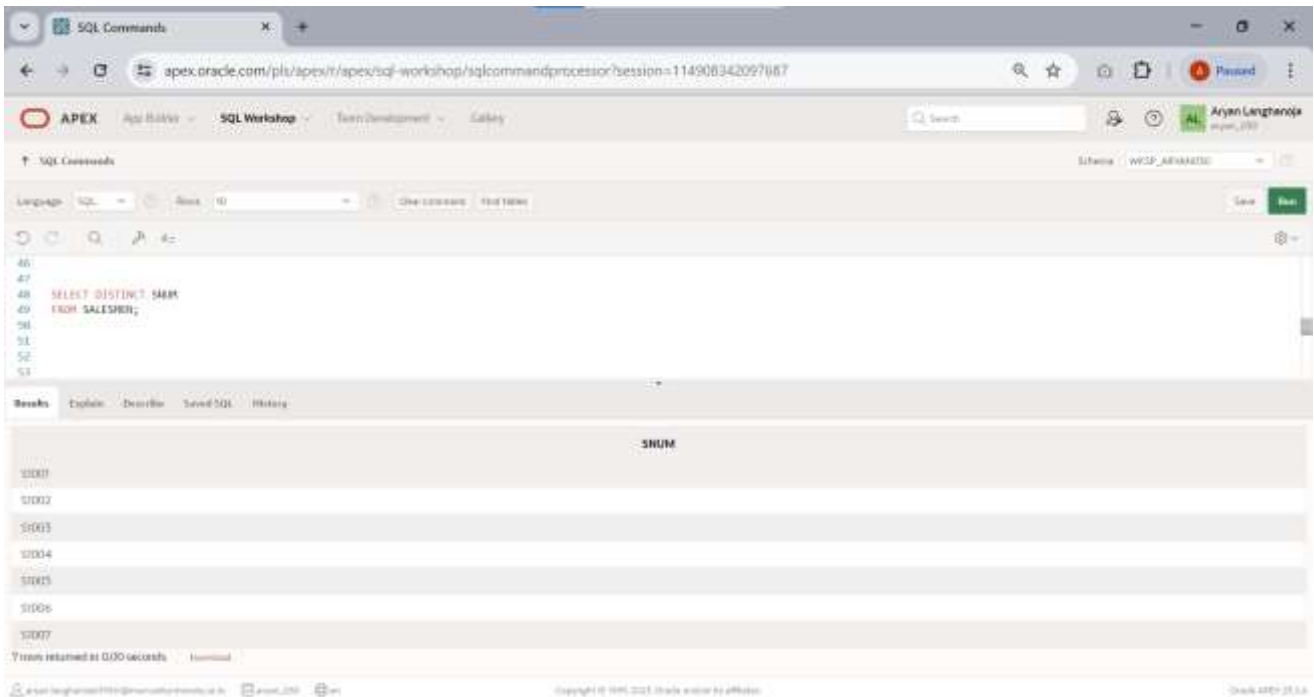
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
SELECT RATING, CNAME
FROM CUSTOMER
WHERE CITY = 'San Jose';
```

The results section shows "no data found".



11. Display SNUM values of all salesmen without any repeat.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
46  
47  
48 SELECT DISTINCT SNUM  
49 FROM SALESMEN;  
50  
51  
52  
53
```

The results are displayed in a table with the column header **SNUM**. The values listed are:

SNUM
S0001
S0002
S0003
S0004
S0005
S0006
S0007

7 rows returned in 0.00 seconds. The interface also shows the user 'Aryan Langhanda' and the schema 'WSP\_AIRMANTE'.

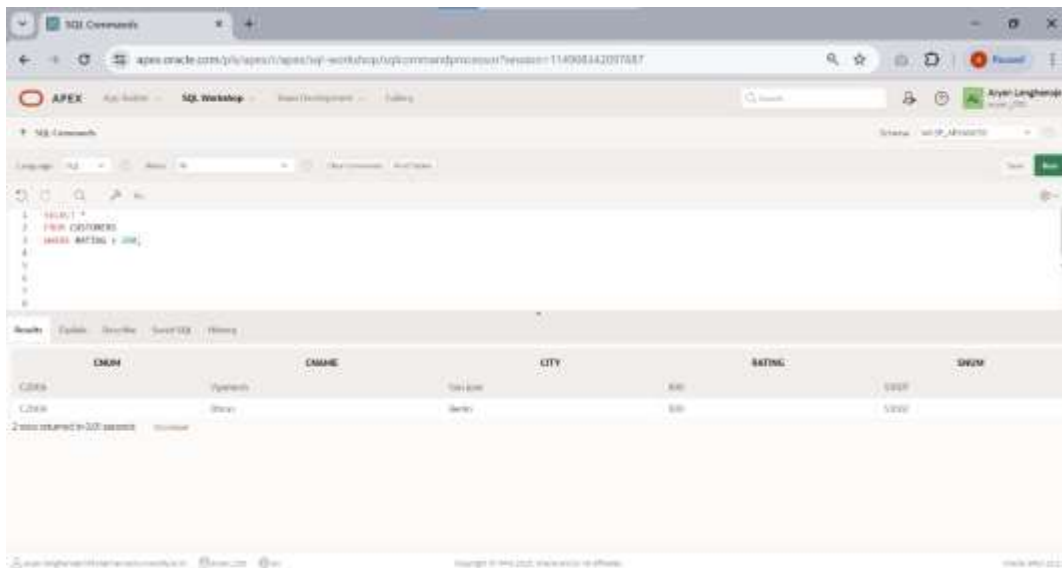
## Practical 9

### Aim: Range Function and Operator based Queries

Perform following queries on the above generated tables.

### OPERATORS

1. List all customers with a rating above 200.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

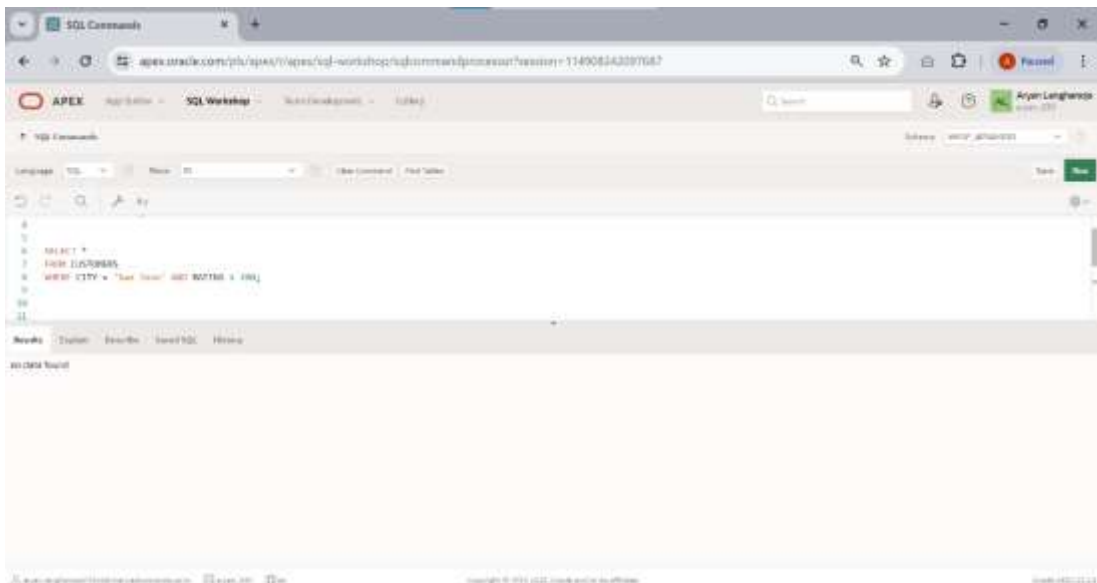
1 SELECT *
2 FROM CUSTOMERS
3 WHERE RATING > 200;

```

The results are displayed in a table with the following columns: CUSTID, CNAME, CITY, RNO, RATING, and SNO. Two rows are returned:

CUSTID	CNAME	CITY	RNO	RATING	SNO
C0006	Sydney	Sydney	500	5000	
C0008	Shanghai	Shanghai	500	5000	

2. List all customers in SAN JOSE who have a rating above 200.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

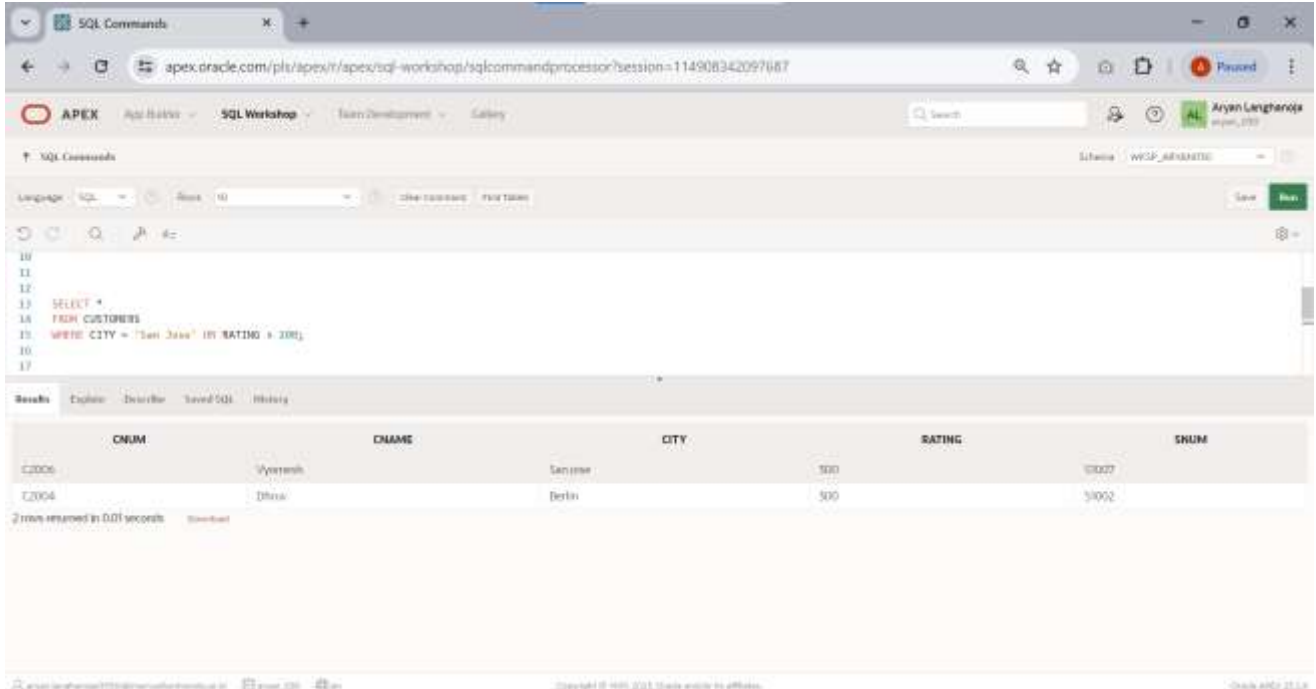
```

1 SELECT *
2 FROM CUSTOMERS
3 WHERE CITY = 'San Jose' AND RATING > 200;

```

The results are displayed in a table with the following columns: CUSTID, CNAME, CITY, RNO, RATING, and SNO. No data is returned.

3. List all customers who were either located in SAN JOSE or had a rating above 200.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

10
11
12
13 SELECT *
14 FROM CUSTOMERS
15 WHERE CITY = 'San Jose' OR RATING > 200;
16
17

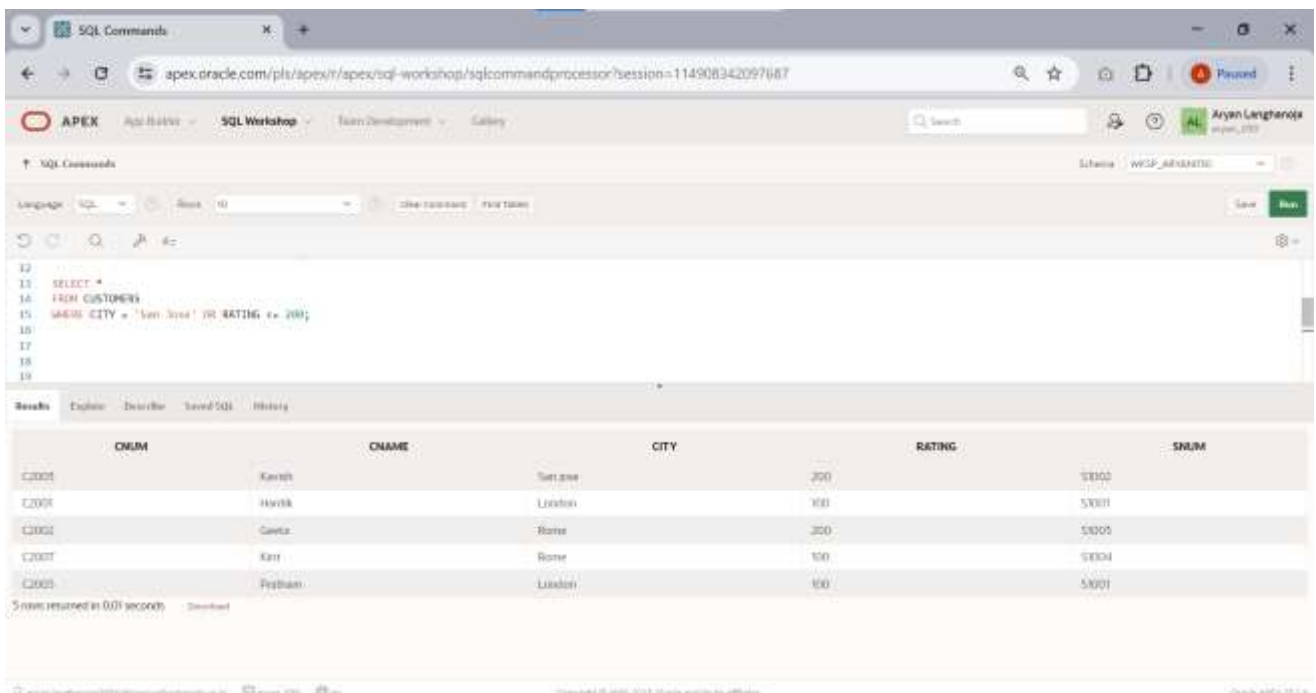
```

The results table displays the following data:

CUSTOMER_ID	CUSTOMER_NAME	CITY	RATING	STATUS
C2006	Vyomesh	San Jose	300	50027
C2004	Dhruv	Berlin	500	50002

2 rows returned in 0.01 seconds

4. List of all customers who were either located in SAN JOSE or not rating above 200.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

12
13 SELECT *
14 FROM CUSTOMERS
15 WHERE CITY = 'San Jose' OR RATING <= 200;
16
17
18
19

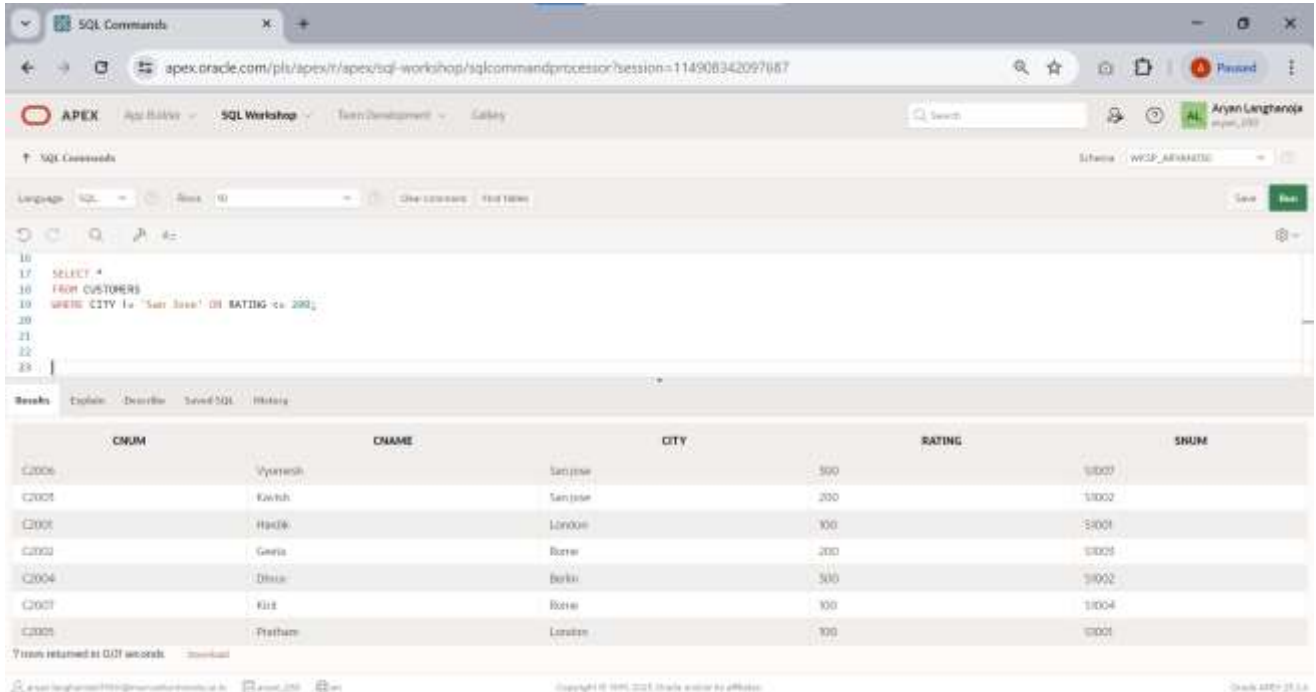
```

The results table displays the following data:

CUSTOMER_ID	CUSTOMER_NAME	CITY	RATING	STATUS
C2004	Karishk	San Jose	200	50002
C2001	Havish	London	100	50001
C2002	Gavita	Rome	200	50005
C2007	Kart	Rome	100	50004
C2005	Pratham	London	100	50001

5 rows returned in 0.01 seconds

5. List of all customers who were not located in SAN JOSE or rating is not above 200.



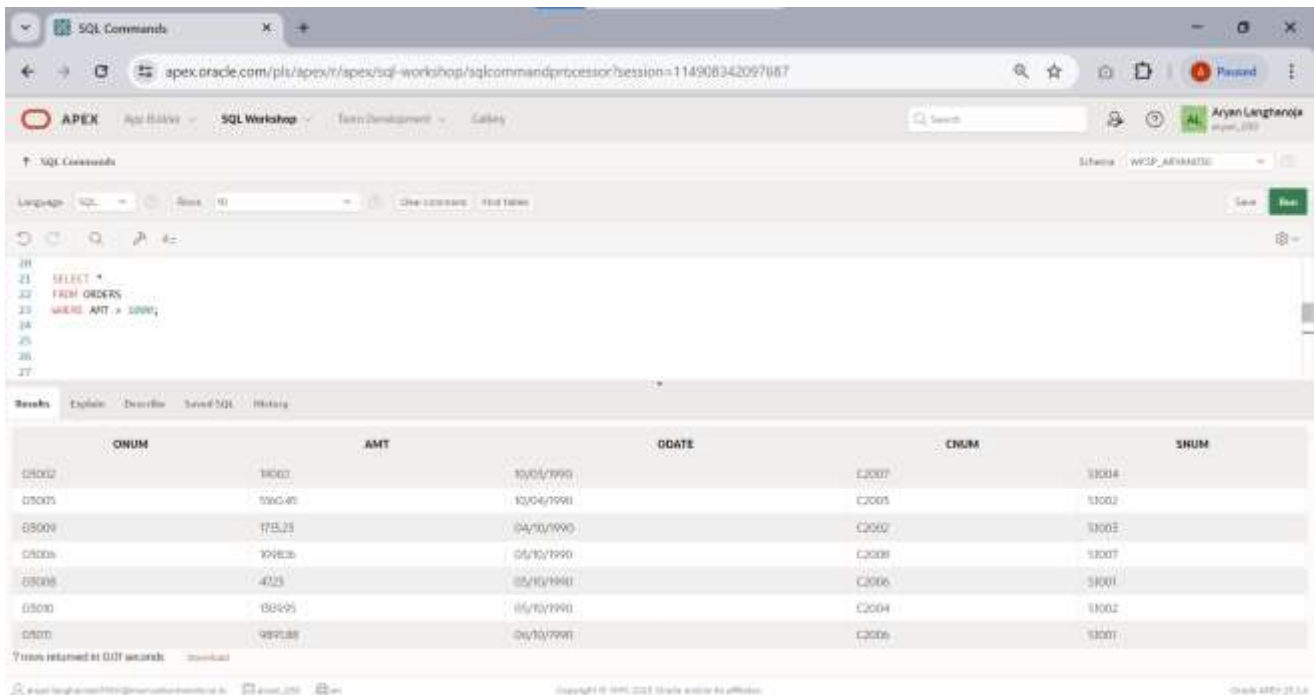
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT *
FROM CUSTOMERS
WHERE CITY != 'San Jose' OR RATING <= 200;
```

The results table displays the following data:

CNUM	CNAME	CITY	RATING	SHUM
C2006	Vyomesh	San Jose	300	11003
C2001	Karthi	San Jose	200	11002
C2003	HadiB	London	100	11001
C2002	Genia	Rome	200	11005
C2004	Dhara	Berlin	300	11002
C2007	Kish	Rome	100	11004
C2005	Pratham	London	100	11001

6. Write a query that will give you all orders for more than \$1000.



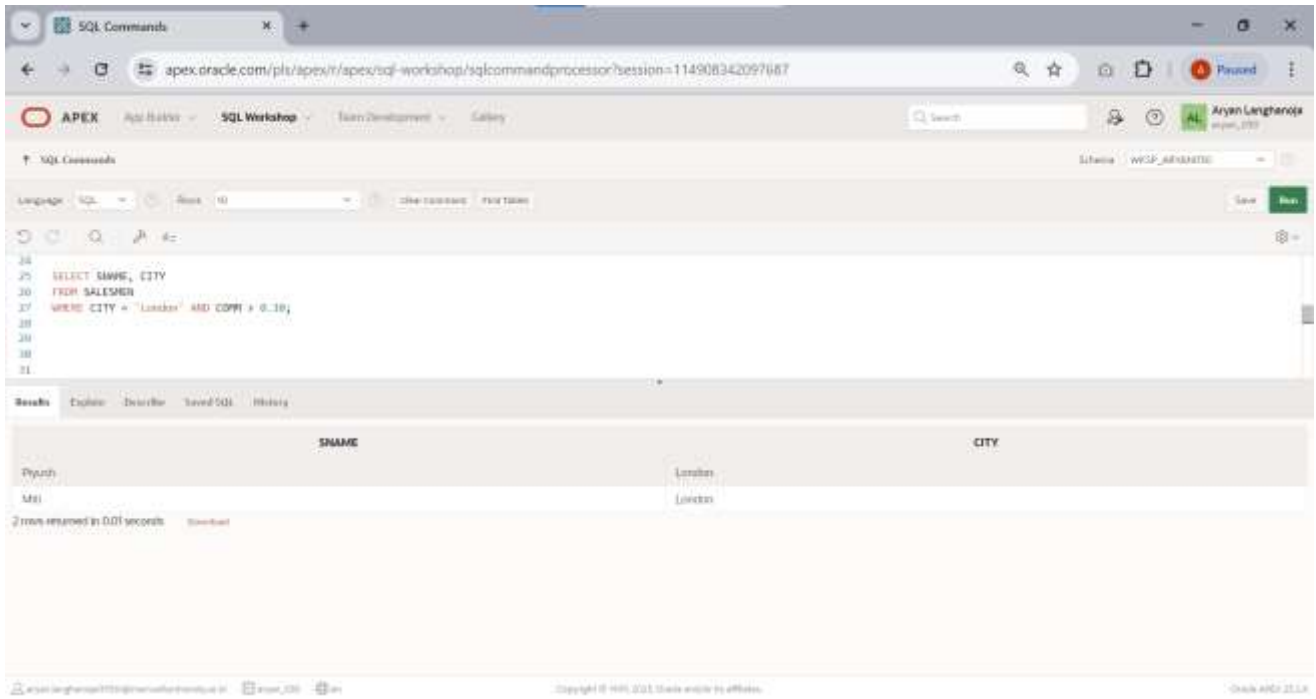
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT *
FROM ORDERS
WHERE AMT > 1000;
```

The results table displays the following data:

ONUM	AMT	ODATE	CNUM	SHUM
O1002	1100.00	10/01/1990	C2007	11004
O1005	1000.00	10/04/1990	C2003	11002
O1009	1750.00	04/10/1990	C2002	11002
O1006	1000.00	05/10/1990	C2008	11007
O1008	400.00	05/10/1990	C2006	11001
O1010	1000.00	06/10/1990	C2004	11002
O1011	980.00	06/10/1990	C2005	11001

7. Write a query that will give you the names and cities of all salesmen in LONDON with a commission above 0.10.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

24 SELECT SNAME, CITY
25 FROM SALESMEN
26 WHERE CITY = 'London' AND COMM > 0.10;
27
28
29
30
31

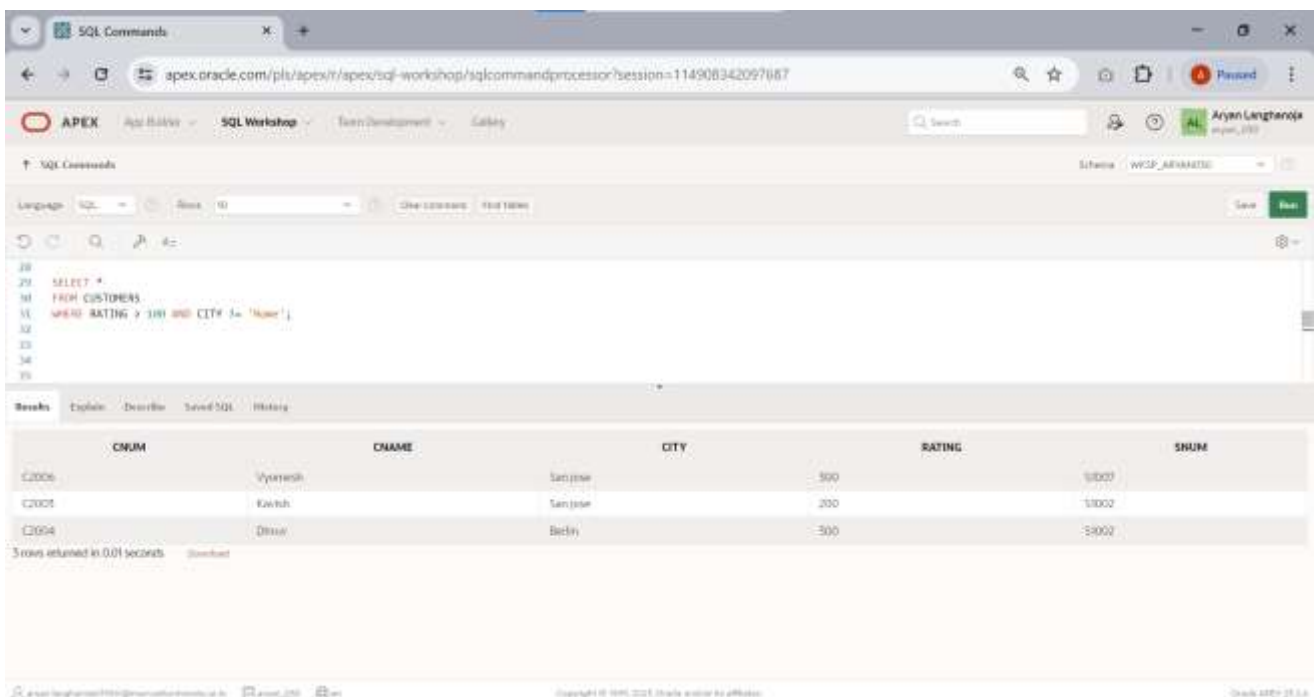
```

The results table shows two rows:

SNAME	CITY
Piyush	London
Mel	London

2 rows returned in 0.01 seconds

8. Write a query on the customers table whose output will exclude all customers with a rating <= 100 and they are located in ROME.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

28 SELECT *
29 FROM CUSTOMERS
30 WHERE RATING > 100 AND CITY != 'Rome';
31
32
33
34
35

```

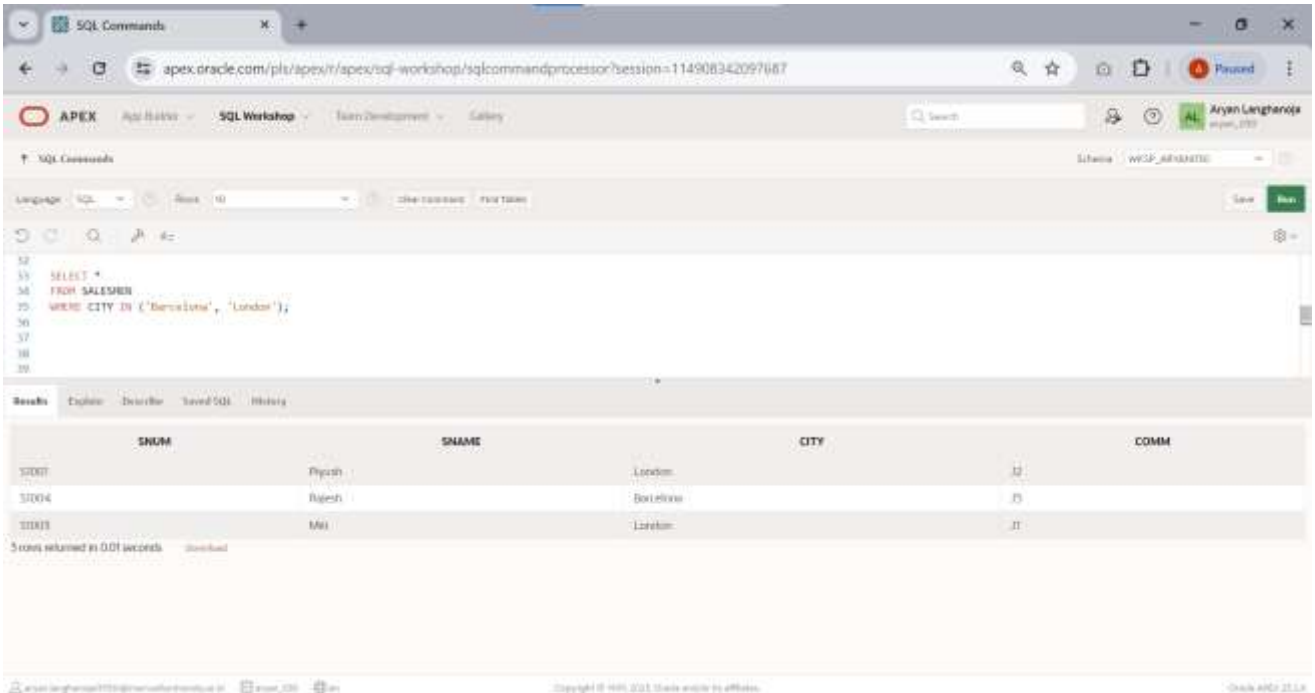
The results table shows three rows:

CNUM	CNAME	CITY	RATING	SNAME
C2006	Vyomesh	Sanjose	500	S0003
C2001	Kavish	Sanjose	200	S1002
C2004	Dhrum	Berlin	300	S1002

3 rows returned in 0.01 seconds

## SPECIAL OPERATORS

1. Display all salesmen that were located in either BARCELONA or LONDON(use IN keyword).



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

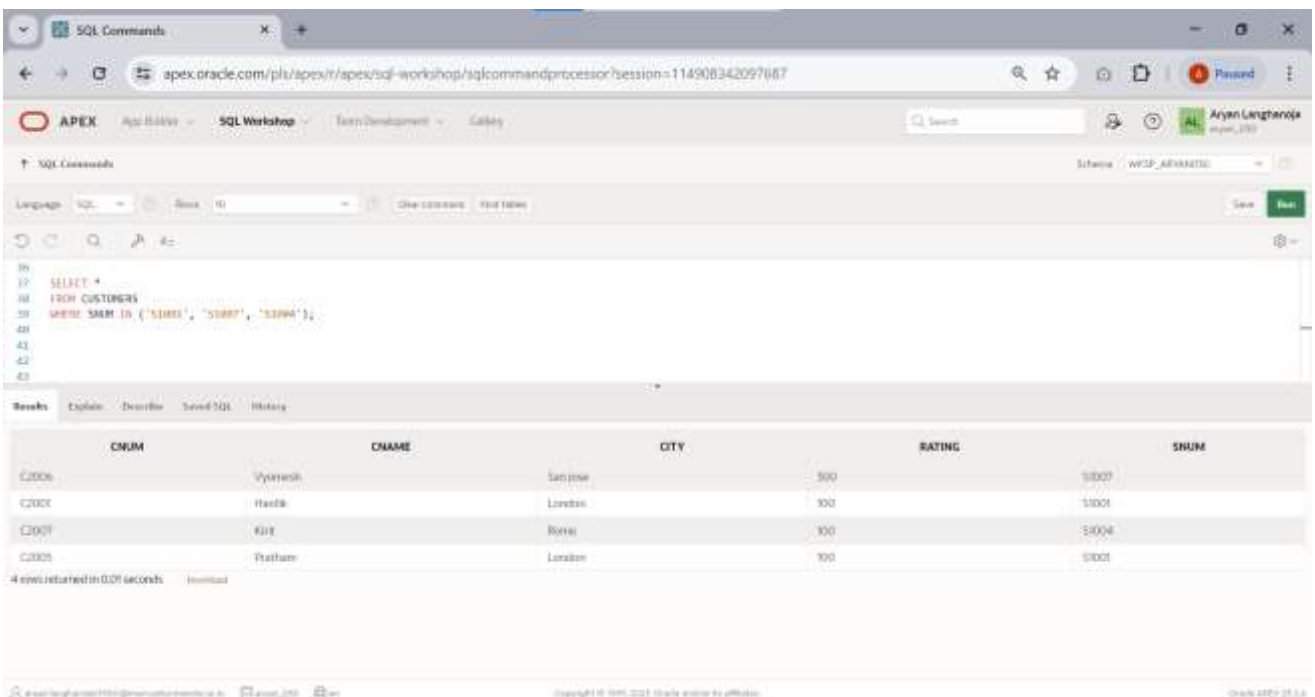
```

32
33 SELECT *
34 FROM SALESMEN
35 WHERE CITY IN ('Barcelona', 'London');
36
37
38
39
  
```

The results table shows 3 rows returned in 0.01 seconds:

SRMID	SNAME	CITY	COMM
S1007	Piyush	London	12
S1004	Rajesh	Barcelona	75
S1003	Mu	London	21

2. Find all customers matched with salesmen S1001,S1007 and S1004.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

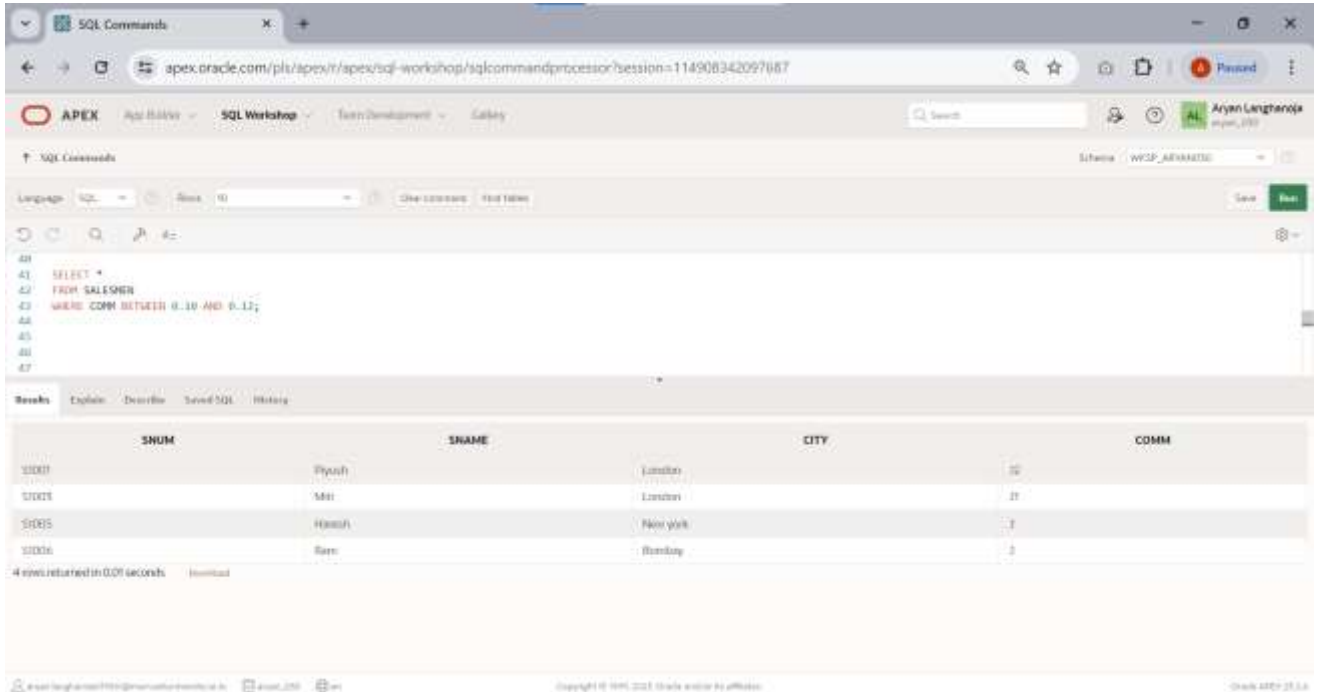
```

36
37 SELECT *
38 FROM CUSTOMERS
39 WHERE SRM IN ('S1001', 'S1007', 'S1004');
40
41
42
43
  
```

The results table shows 4 rows returned in 0.01 seconds:

CID	CNAME	CITY	RATING	SRM
C2006	Vyomesh	San Jose	500	S1007
C2007	Harish	London	100	S1001
C2009	Rishi	Rome	100	S1004
C2008	Ratham	London	100	S1001

- Display all salesmen with commission between 0.10 and 0.12.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

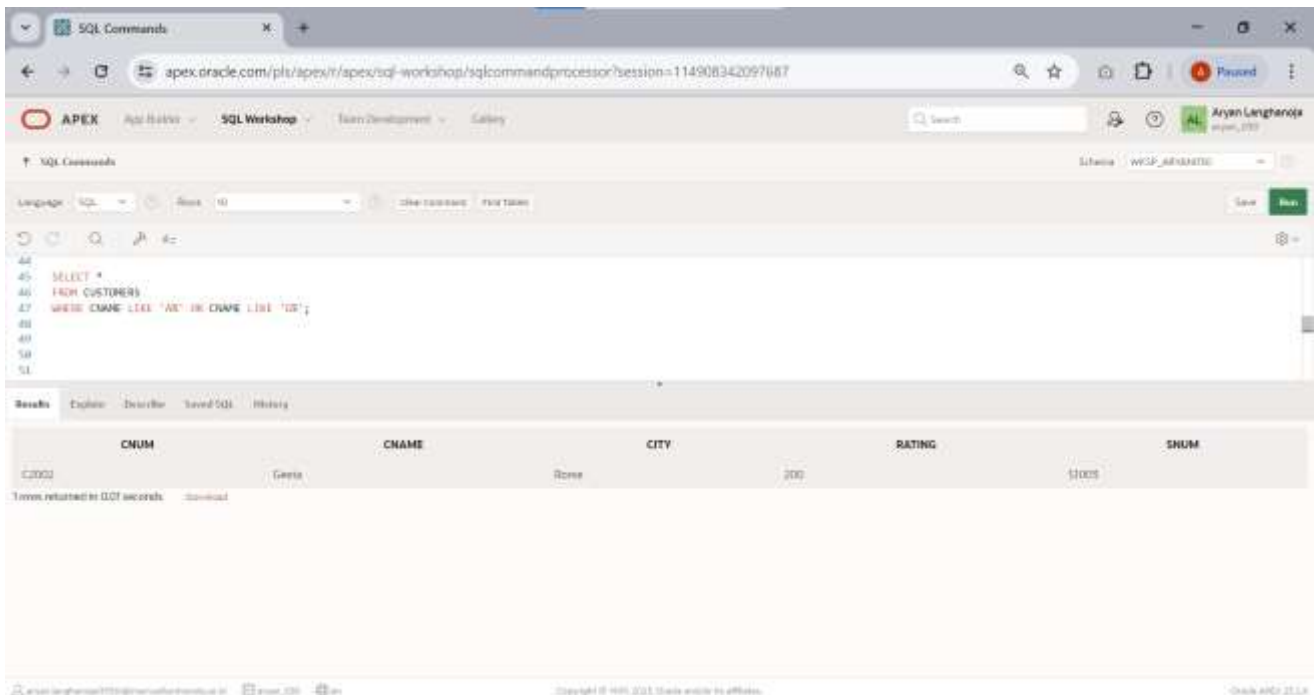
40
41 SELECT *
42 FROM SALESMEN
43 WHERE COMM BETWEEN 0.10 AND 0.12;
44
45
46
47
  
```

The results table displays the following data:

SNUM	SNAME	CITY	COMM
10001	Piyush	London	0.10
10002	Mai	London	0.11
10005	Hanshi	New York	0.12
10006	Ram	Romney	0.12

4 rows returned in 0.01 seconds

- Select all customers whose names fall in a 'A' and 'G' alphabetical.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

44
45 SELECT *
46 FROM CUSTOMERS
47 WHERE CNAME LIKE 'A%' OR CNAME LIKE 'G%';
48
49
50
51
  
```

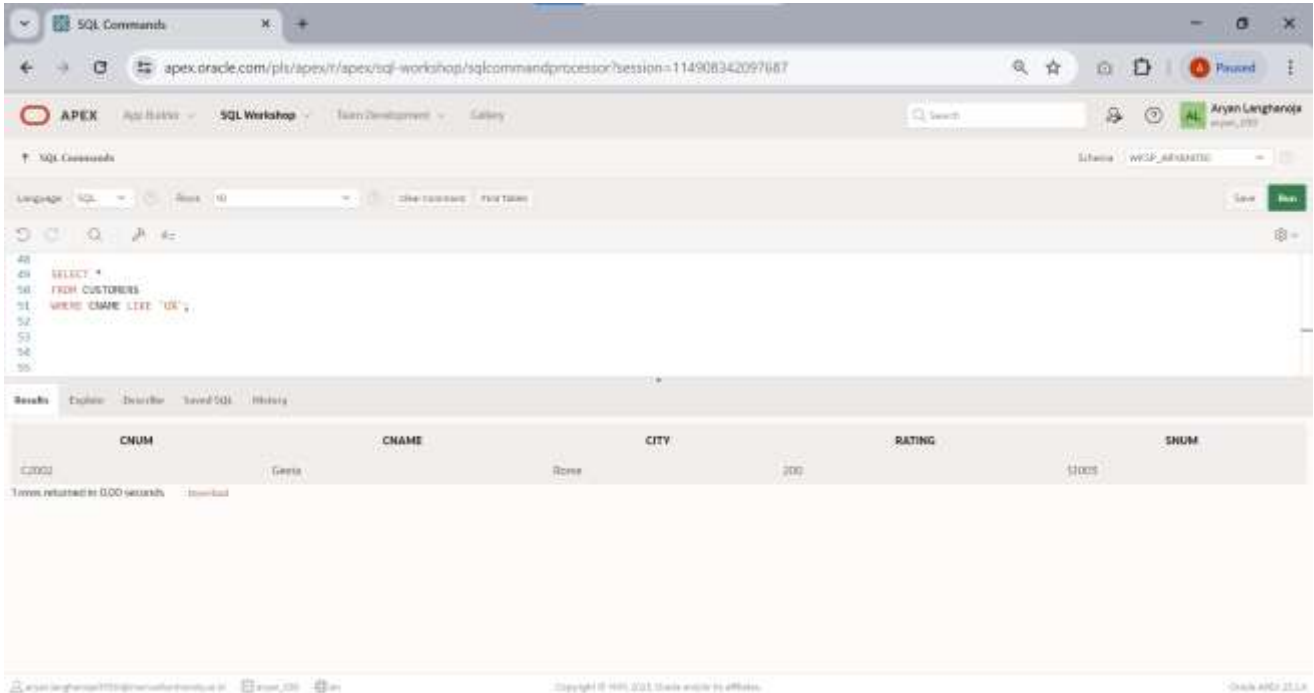
The results table displays the following data:

CNUM	CNAME	CITY	RATING	SHUM
10001	Genia	Rome	100	10005

1 rows returned in 0.01 seconds

## LIKE OPERATORS.

1. List all the customers whose names begin with 'G'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

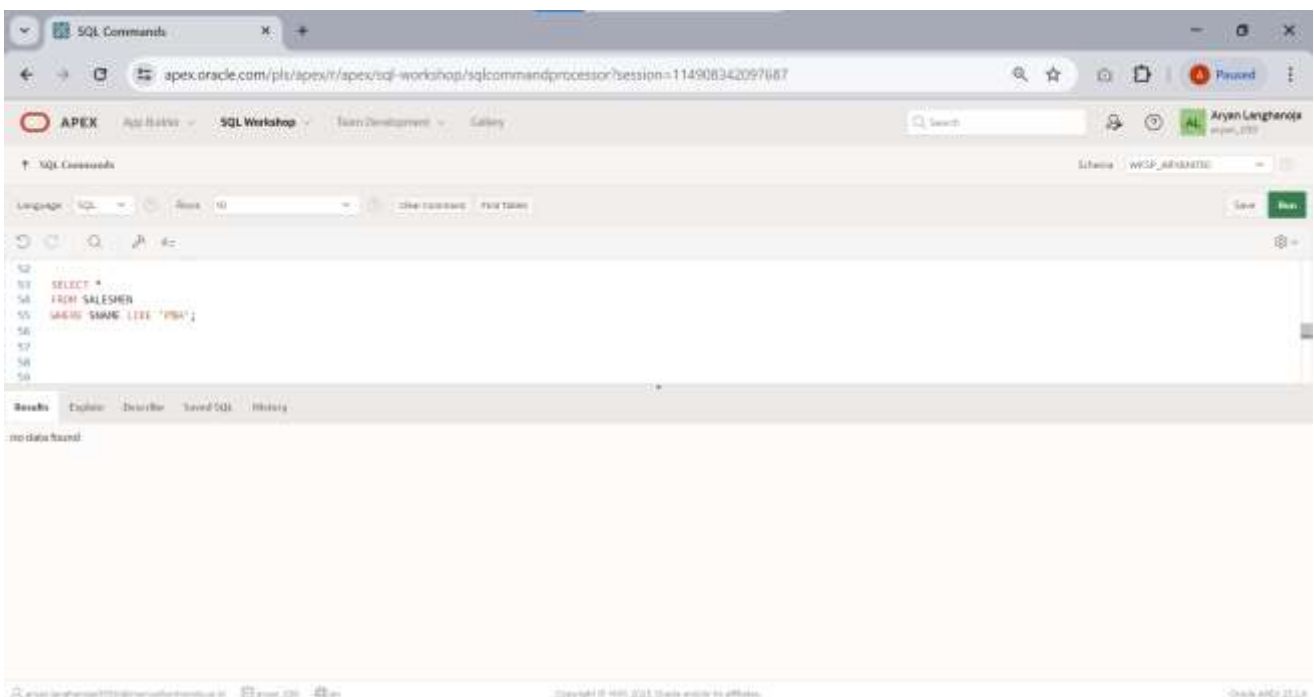
```
SELECT *  
FROM CUSTOMERS  
WHERE CNAME LIKE 'G%';
```

The results are displayed in a table with the following columns: CNUM, CNAME, CITY, RATING, and SNUM. The first row of data is:

CNUM	CNAME	CITY	RATING	SNUM
1000	Gena	Rome	100	1000

Times returned in 0.00 seconds. Download

2. List all salesmen whose sname start with letter 'P' and end letter is 'H'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

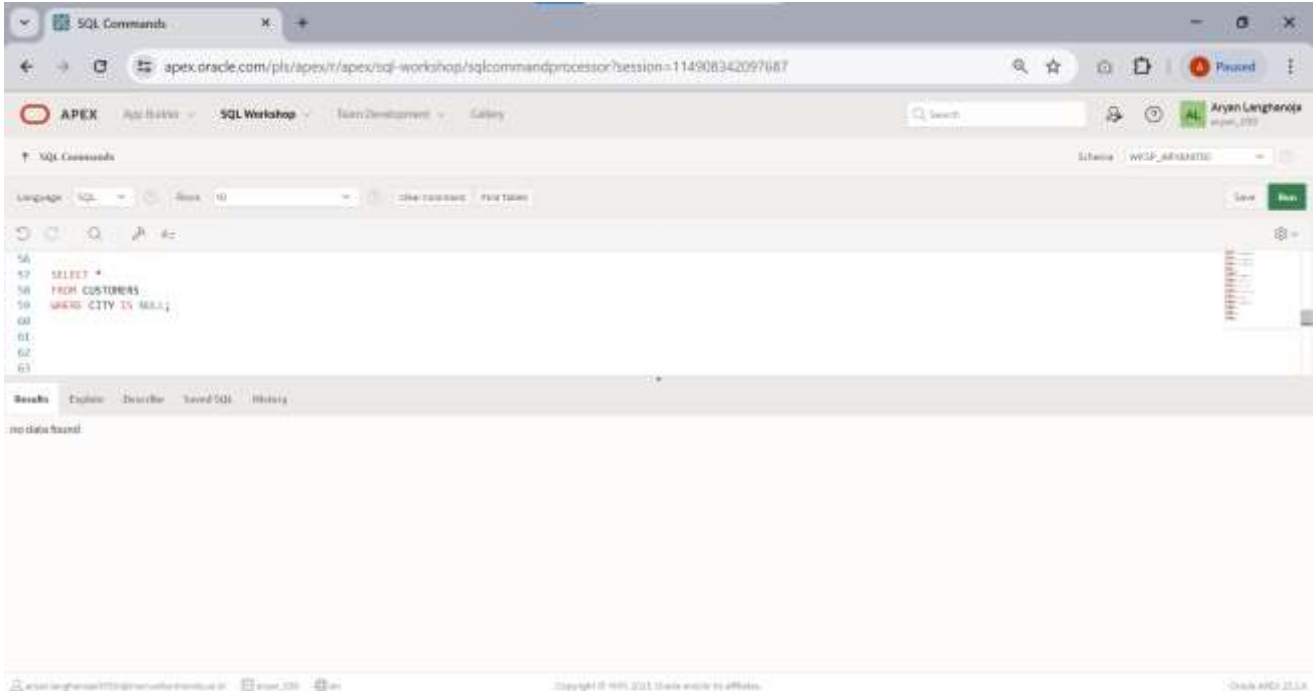
```
SELECT *  
FROM SALESMEN  
WHERE SNAME LIKE 'P%H';
```

The results section displays "no data found".



## NULL OPERATORS

1. Find all records in customer table with NULL values in the city column.



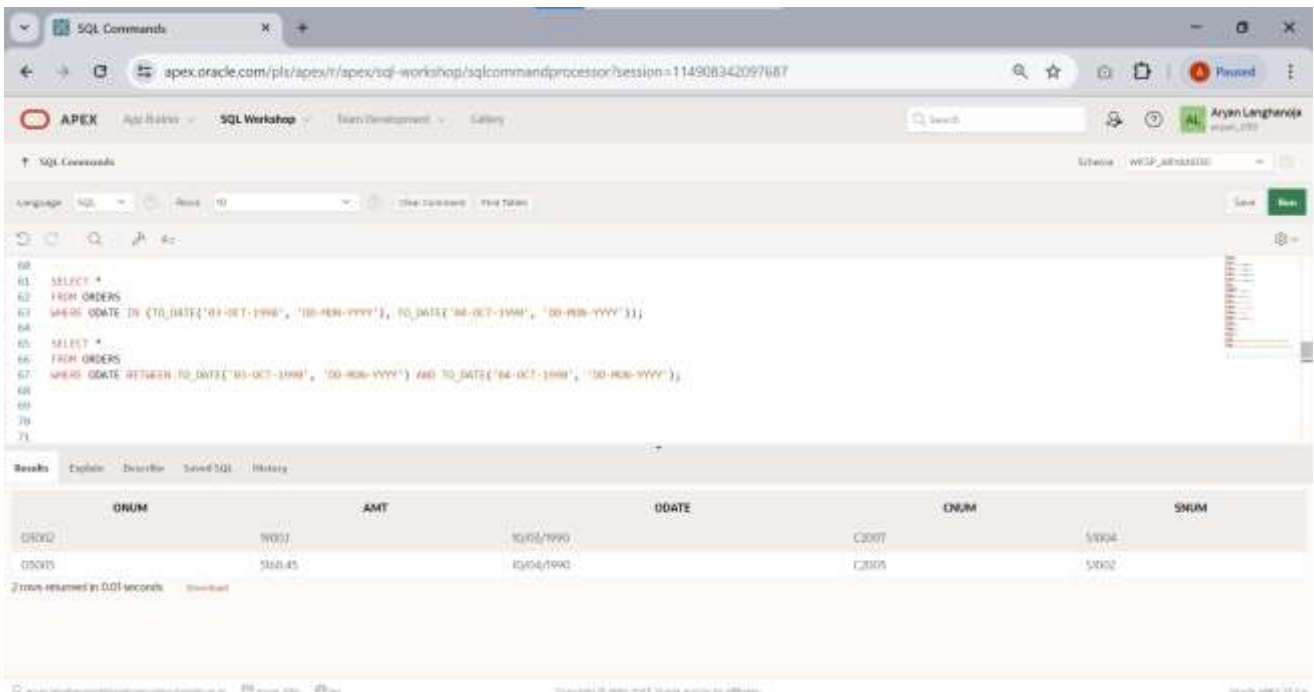
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

54
55
56 SELECT *
57 FROM CUSTOMERS
58 WHERE CITY IS NULL;
59
60
61
62
63
  
```

The results section shows "no data found".

2. Write a two queries that will produce all orders taken on October 3rd or 4th ,1990 ( use IN operator and Use BETWEEN operator )



The screenshot shows the APEX SQL Workshop interface with two queries entered:

```

68
69
70 SELECT *
71 FROM ORDERS
72 WHERE ODATE IN ('03-OCT-1990', '04-OCT-1990');
73
74
75
76 SELECT *
77 FROM ORDERS
78 WHERE ODATE BETWEEN TO_DATE('03-OCT-1990', 'DD-MON-YYYY') AND TO_DATE('04-OCT-1990', 'DD-MON-YYYY');
79
80
81
  
```

The results section shows two rows of data:

ORDID	AMT	ODATE	CNUM	SNUM
00002	9000	10/03/1990	C0007	S0004
00005	5150.45	10/04/1990	C0005	S0002

2 rows returned in 0.01 seconds

3. Write a query that selects all of the customers matched with S1001 and S1002.

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
SELECT *
FROM CUSTOMERS
WHERE SNUM IN ('S1001', 'S1002');
```

The results table displays the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2001	Karish	Sanjose	200	S1002
C2001	Hank	London	100	S1001
C2004	Dinesh	Berlin	300	S1002
C2005	Porham	London	100	S1001

4 rows returned in 0.01 seconds

4. Write a query that will produce all of the customers whose names begin with a letter from A to H.

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

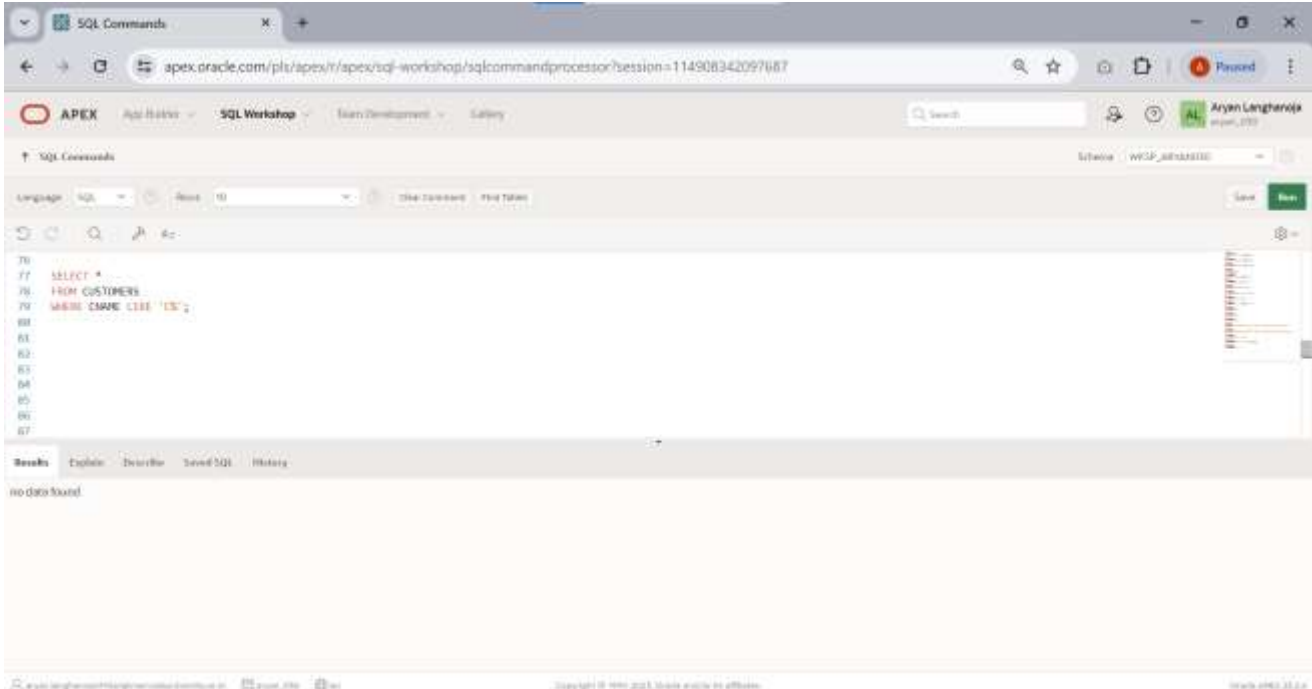
```
SELECT *
FROM CUSTOMERS
WHERE CNAME BETWEEN 'A' AND 'H' ORDER BY CNAME;
```

The results table displays the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2004	Dinesh	Berlin	300	S1002
C2001	Hank	London	100	S1001

2 rows returned in 0.01 seconds

5. Write a query that selects all customers whose names begin with 'C'.



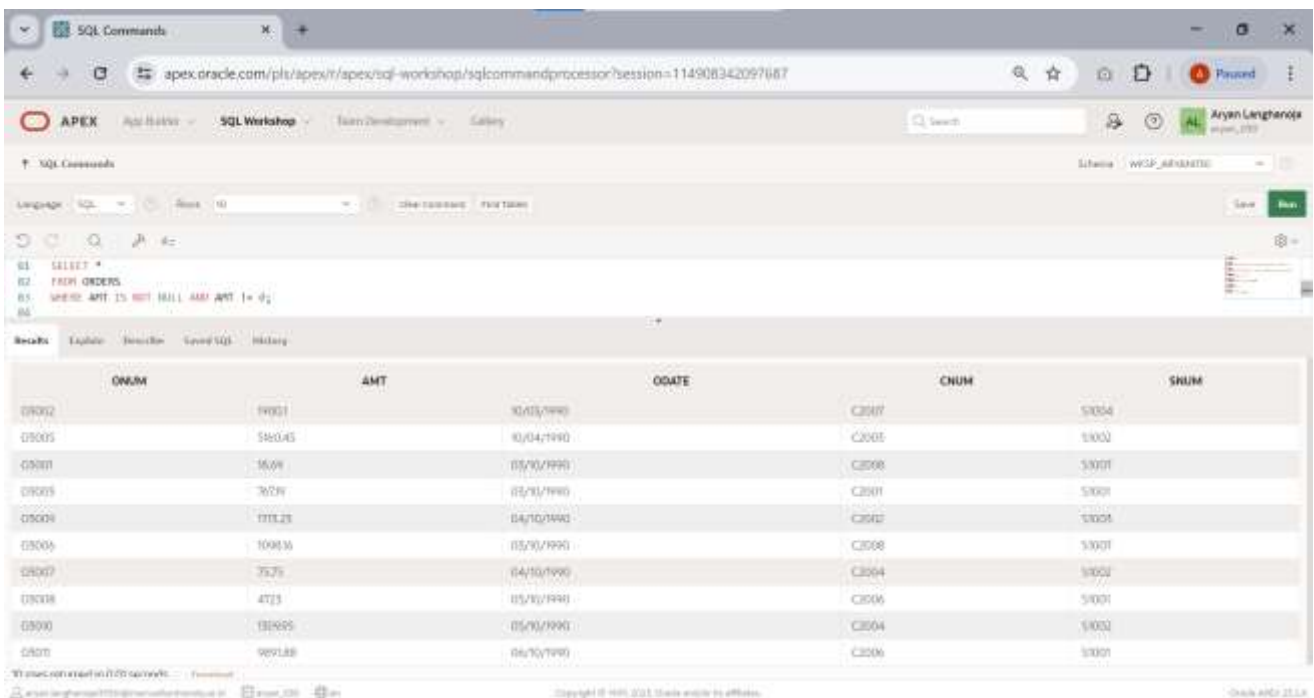
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

76 SELECT *
77 FROM CUSTOMERS
78 WHERE NAME LIKE 'C%';
  
```

The results section shows "no data found".

6. Write a query that selects all orders without ZEROS or NULLS in amt field..



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

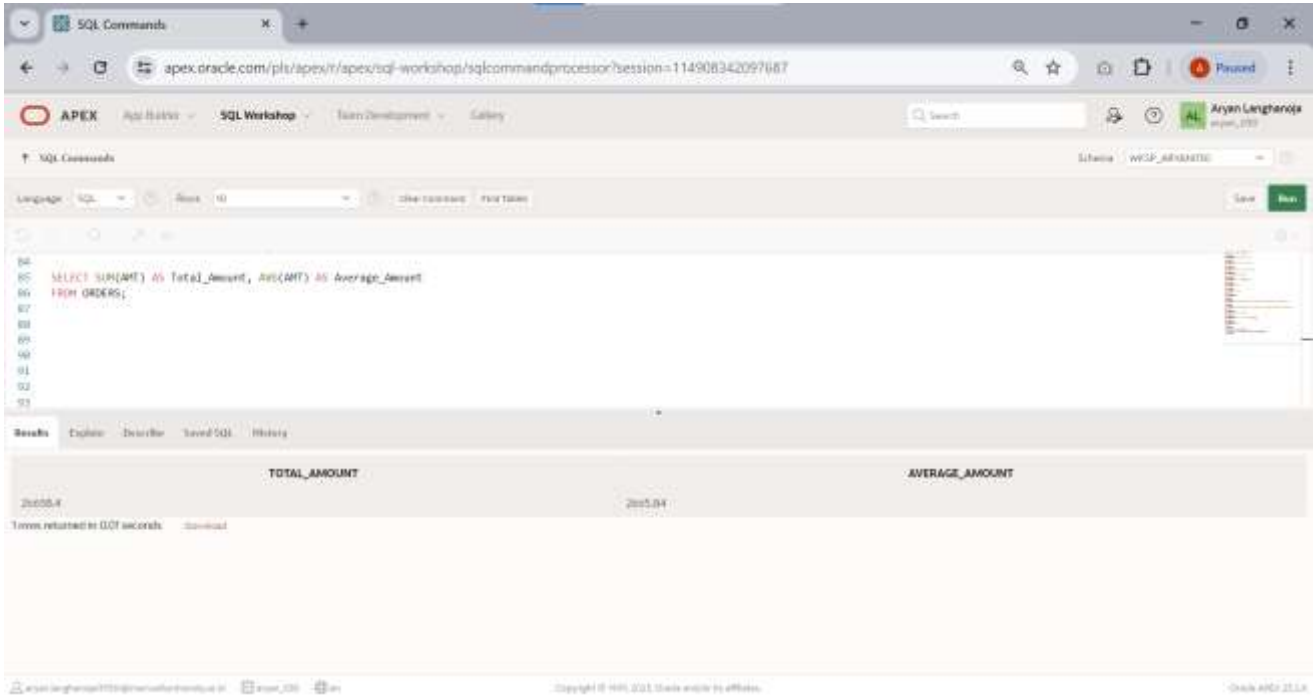
81 SELECT *
82 FROM ORDERS
83 WHERE AMT IS NOT NULL AND AMT != 0;
  
```

The results section displays a table with the following data:

ORDR	AMT	ODATE	CNUM	SNUM
09002	19001	10/03/1990	C2007	S1004
09005	580.45	10/04/1990	C2005	S1002
09008	16.69	05/05/1990	C2008	S1001
09009	70.29	05/05/1990	C2001	S1001
09009	1118.28	04/10/1990	C2042	S1005
09009	1091.36	05/05/1990	C2008	S1001
09007	75.75	04/09/1990	C2004	S1002
09008	47.73	05/05/1990	C2006	S1001
09000	131695	05/05/1990	C2004	S1002
09005	9891.88	04/05/1990	C2006	S1001

## FUNCTIONS

1. Display sum of amt, average of orders.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

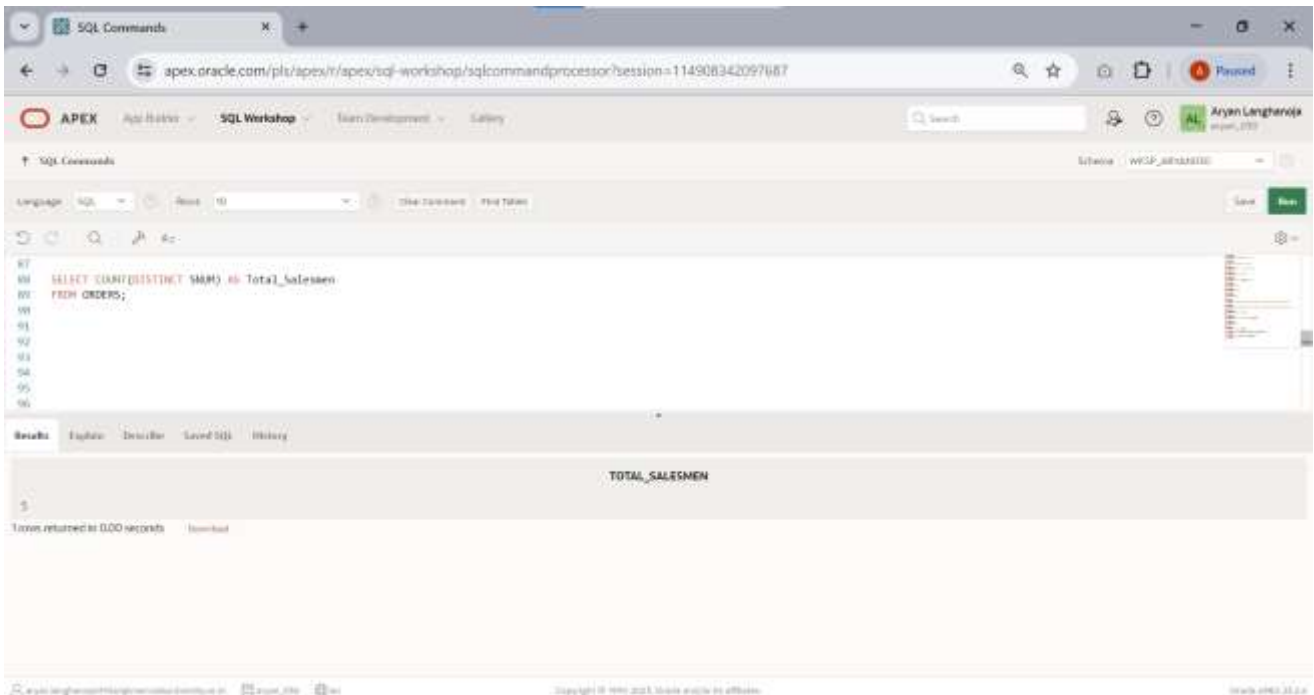
```
84  
85 SELECT SUM(AMT) AS Total_Amount, AVG(AMT) AS Average_Amount  
86 FROM ORDERS;  
87  
88  
89  
90  
91  
92  
93
```

The results are displayed in a table with two columns: **TOTAL\_AMOUNT** and **AVERAGE\_AMOUNT**. The values are 20055.4 and 2005.04 respectively.

TOTAL_AMOUNT	AVERAGE_AMOUNT
20055.4	2005.04

Times returned in 0.07 seconds. Download

2. To count the numbers of salesmen without duplication in the orders tables.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

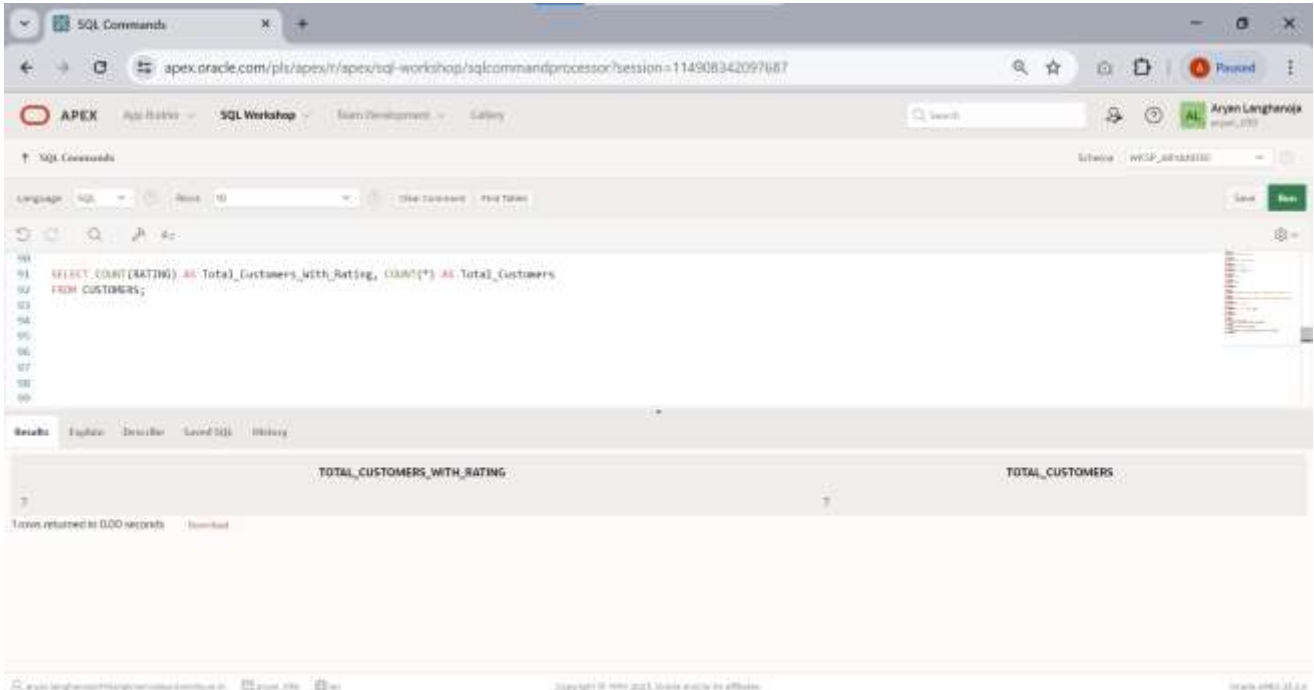
```
87  
88 SELECT COUNT(DISTINCT SSM) AS Total_Salesmen  
89 FROM ORDERS;  
90  
91  
92  
93  
94  
95  
96
```

The results are displayed in a table with one column: **TOTAL\_SALESMEN**. The value is 5.

TOTAL_SALESMEN
5

Times returned in 0.00 seconds. Download

3. Count the rating of customers (with NULL and without NULL).



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

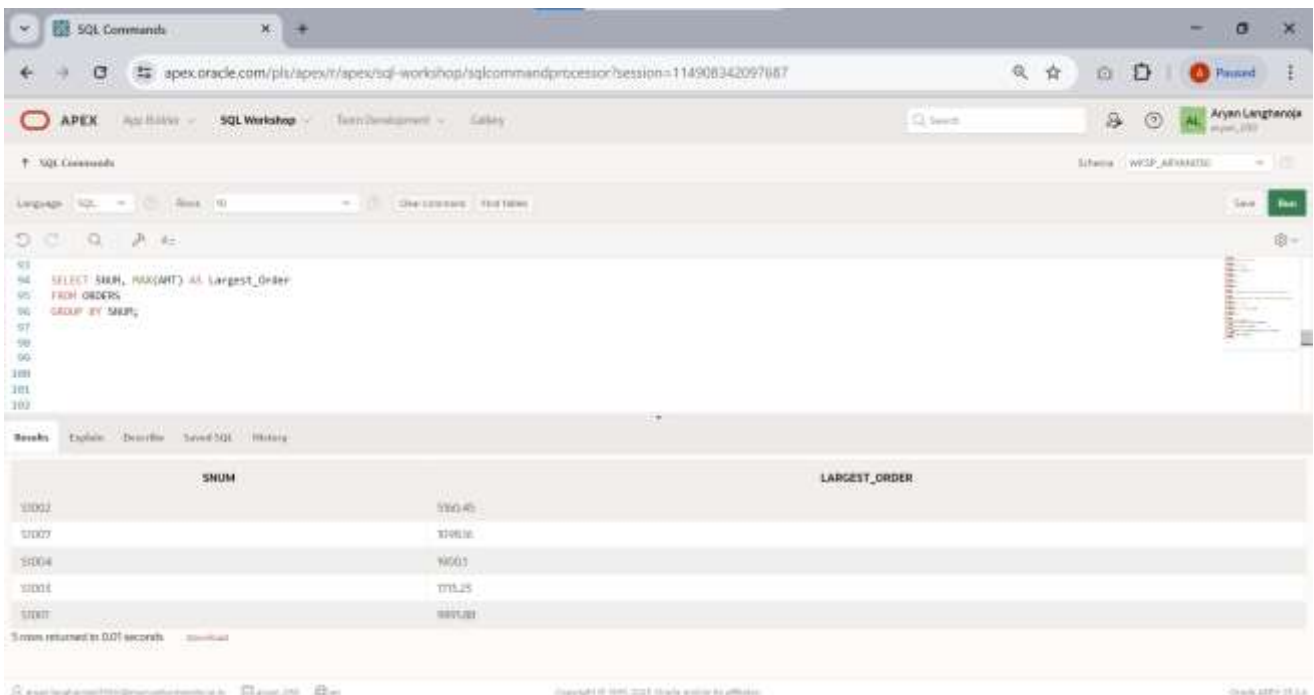
91 SELECT COUNT(RATING) AS Total_Customers_with_Rating, COUNT(*) AS Total_Customers
92 FROM CUSTOMERS;
93
94
95
96
97
98
99
  
```

The results table shows two columns: **TOTAL\_CUSTOMERS\_WITH\_RATING** and **TOTAL\_CUSTOMERS**. The first row of data shows values 7 and 7 respectively.

TOTAL_CUSTOMERS_WITH_RATING	TOTAL_CUSTOMERS
7	7

Rows returned in 0.00 seconds

4. Find the largest order taken by each salesperson.(hint: use group by)



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

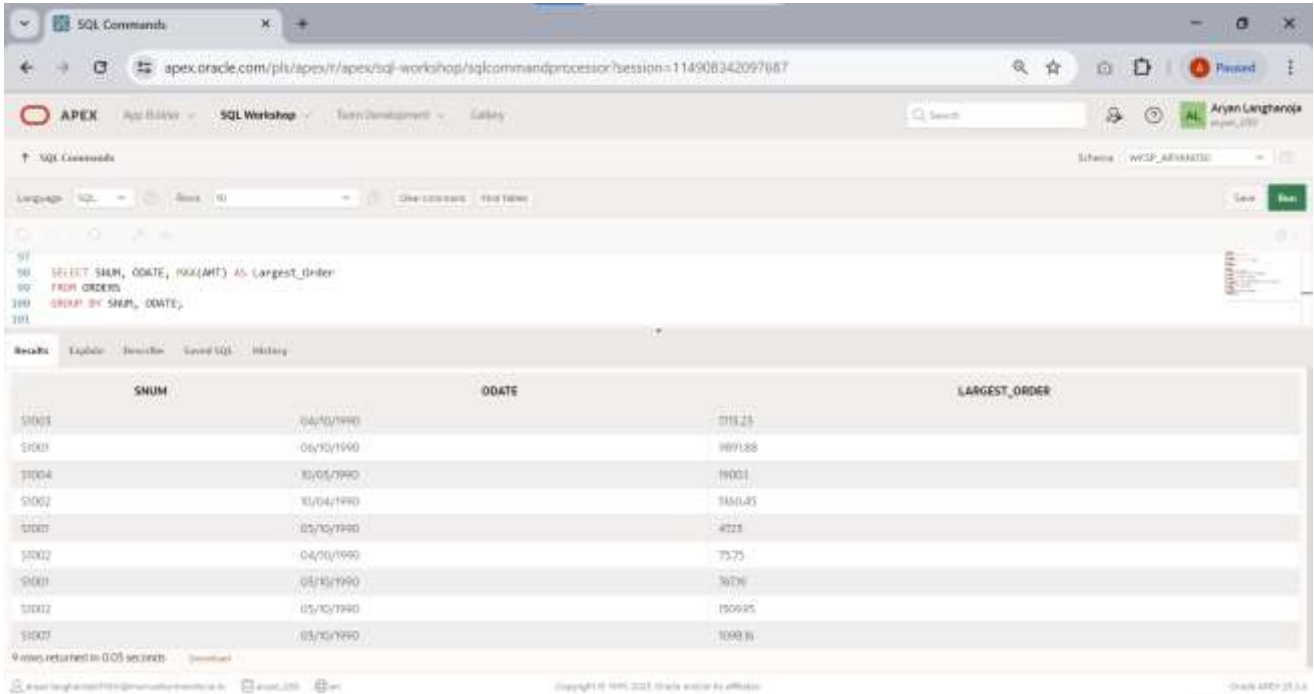
103
104 SELECT SNUM, MAX(AMT) AS Largest_Order
105 FROM ORDERS
106 GROUP BY SNUM;
107
108
109
110
111
112
  
```

The results table shows two columns: **SNUM** and **LARGEST\_ORDER**. The data rows are:

SNUM	LARGEST_ORDER
10002	5500.40
10007	1046.16
10004	900.05
10016	1715.25
10001	9891.89

Rows returned in 0.01 seconds

5. Find the largest order taken by each salesperson on each date.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

97
98 SELECT SNUM, ODATE, MAX(AMT) AS Largest_Order
99 FROM ORDERS
100 GROUP BY SNUM, ODATE;
101

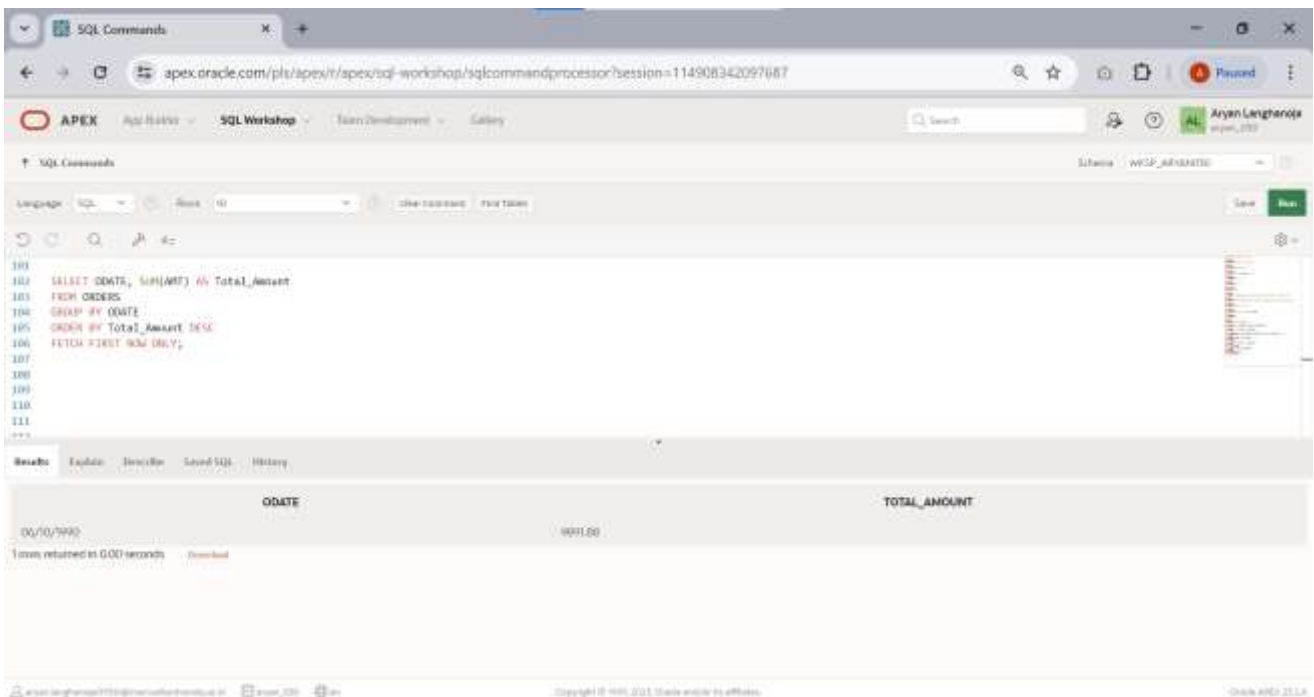
```

The results table displays the following data:

SNUM	ODATE	LARGEST_ORDER
S1003	04/10/1990	3719.25
S1001	06/10/1990	8891.88
S1004	10/05/1990	19003
S1002	11/04/1990	1550.43
S1007	05/10/1990	4725
S1002	04/10/1990	75.75
S1003	05/10/1990	3079
S1002	05/10/1990	1509.95
S1007	05/10/1990	1098.9

9 rows returned in 0.05 seconds

6. Find out which day had the higher total amount ordered.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

101
102 SELECT ODATE, SUM(AMT) AS Total_Amount
103 FROM ORDERS
104 GROUP BY ODATE
105 ORDER BY Total_Amount DESC;
106
107
108
109
110
111

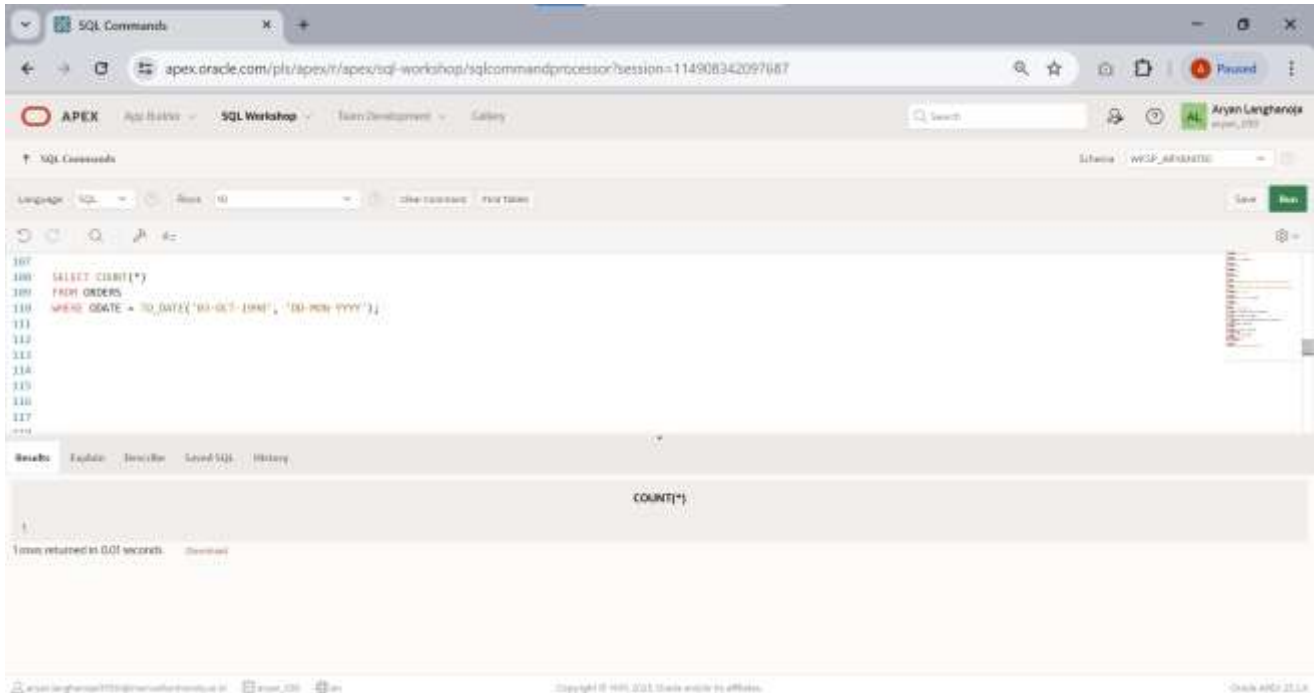
```

The results table displays the following data:

ODATE	TOTAL_AMOUNT
05/10/1990	8891.88

1 rows returned in 0.00 seconds

7. Write a query that counts all orders for October 3rd.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

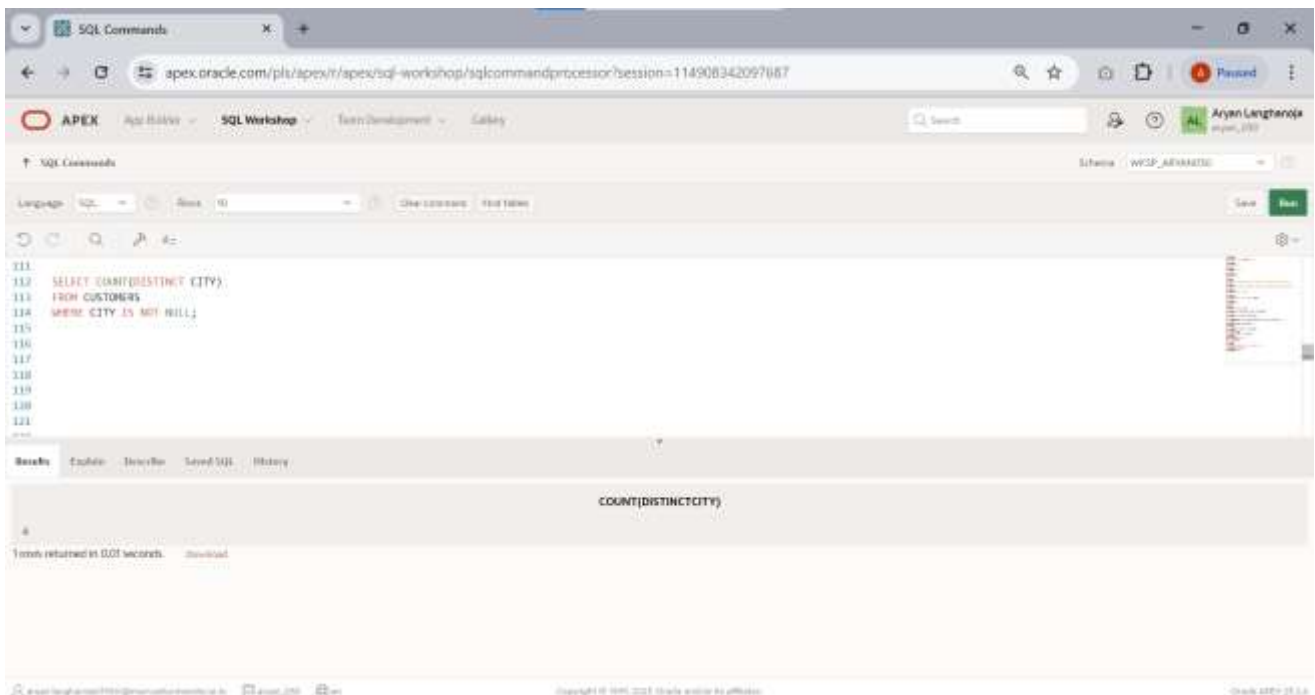
107
108 SELECT COUNT(*)
109 FROM ORDERS
110 WHERE ODATE = TO_DATE('03-OCT-1991', 'DD-MON-YYYY')
111
112
113
114
115
116
117
118
  
```

The Results tab shows the query execution results:

COUNT(*)
1

1 row returned in 0.01 seconds.

8. Write a query that counts the number of different non-NULL city in the customer table.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

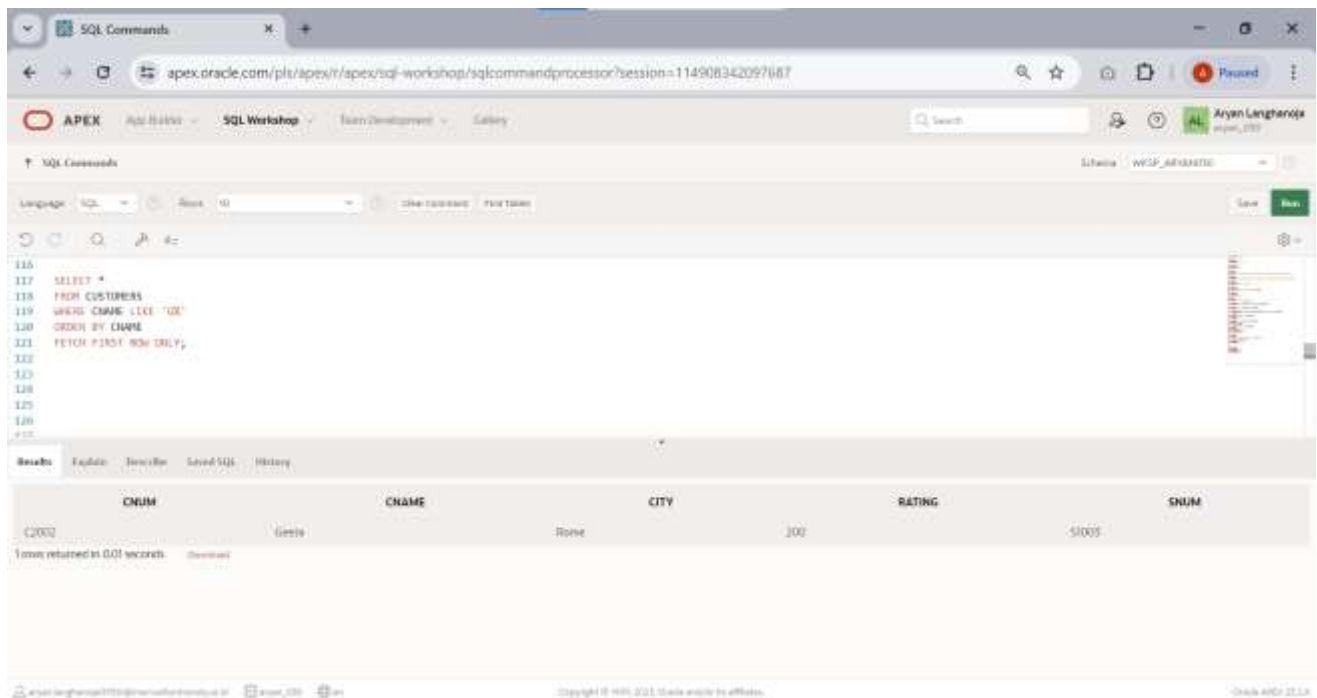
111
112 SELECT COUNT(DISTINCT CITY)
113 FROM CUSTOMERS
114 WHERE CITY IS NOT NULL
115
116
117
118
119
120
121
122
  
```

The Results tab shows the query execution results:

COUNT(DISTINCT CITY)
4

1 row returned in 0.01 seconds.

9. Write a query that selects the first customer in alphabetical order whose name begin with 'G'.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```
115  
117 SELECT *  
118 FROM CUSTOMERS  
119 WHERE CNAME LIKE 'G%'  
120 ORDER BY CNAME  
121 FETCH FIRST ROW ONLY;  
122  
123  
124  
125  
126
```

The Results tab shows the following data:

CNUM	CNAME	CITY	RATING	SNUM
12002	Geeta	Rome	300	50005

1 row returned in 0.01 seconds.

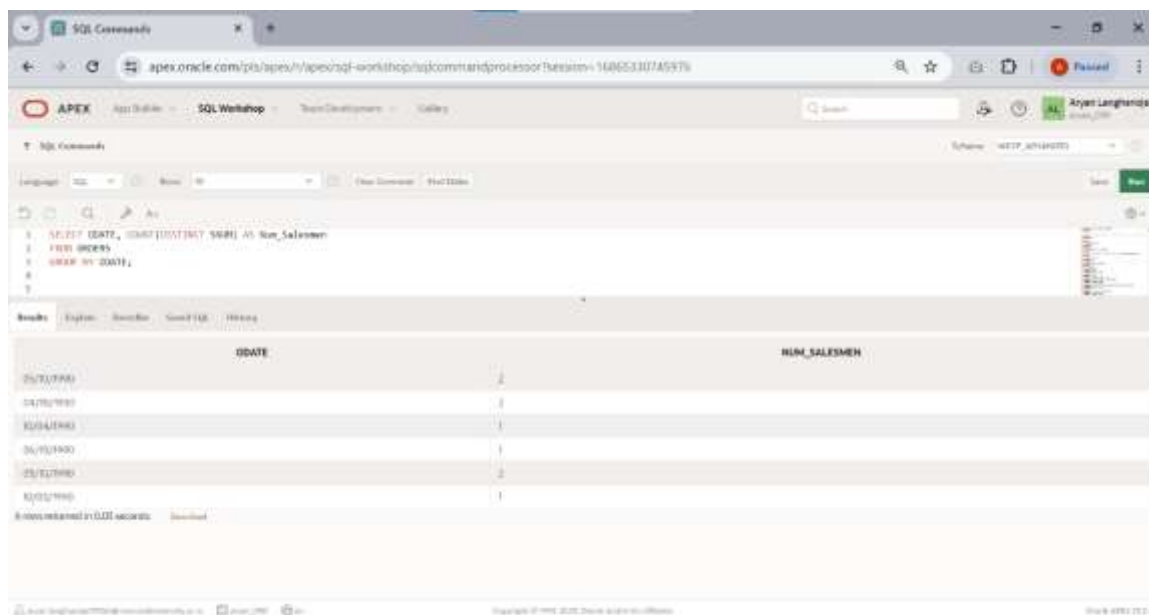


## Practical 10

### Aim: Join based Queries having Functions

Perform queries on above tables

1. Write a query that counts the number of salesmen registering orders for each day(if a salesperson has more than one order on a given day , he or she should be counted only once)



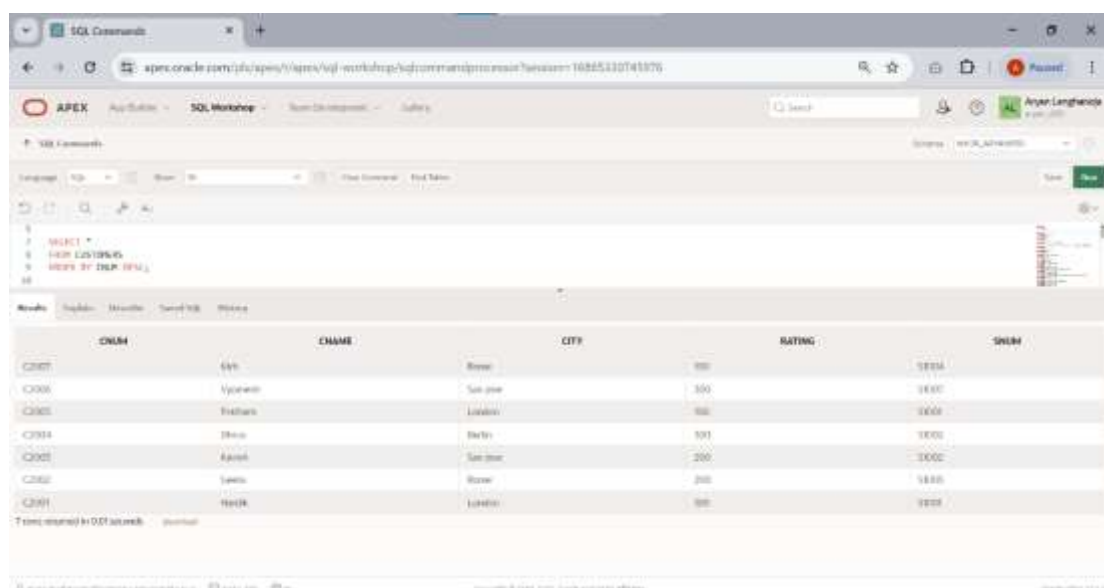
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT DATE, COUNT(DISTINCT SNUM) AS Num_Salesmen
FROM ORDERS
GROUP BY DATE;
```

The results table shows the following data:

DATE	NUM_SALESMEN
25/10/1990	2
24/10/1990	1
22/10/1990	1
26/10/1990	1
25/10/1990	2
23/10/1990	1

2. Display all the information in descending orders(use column CNUM).



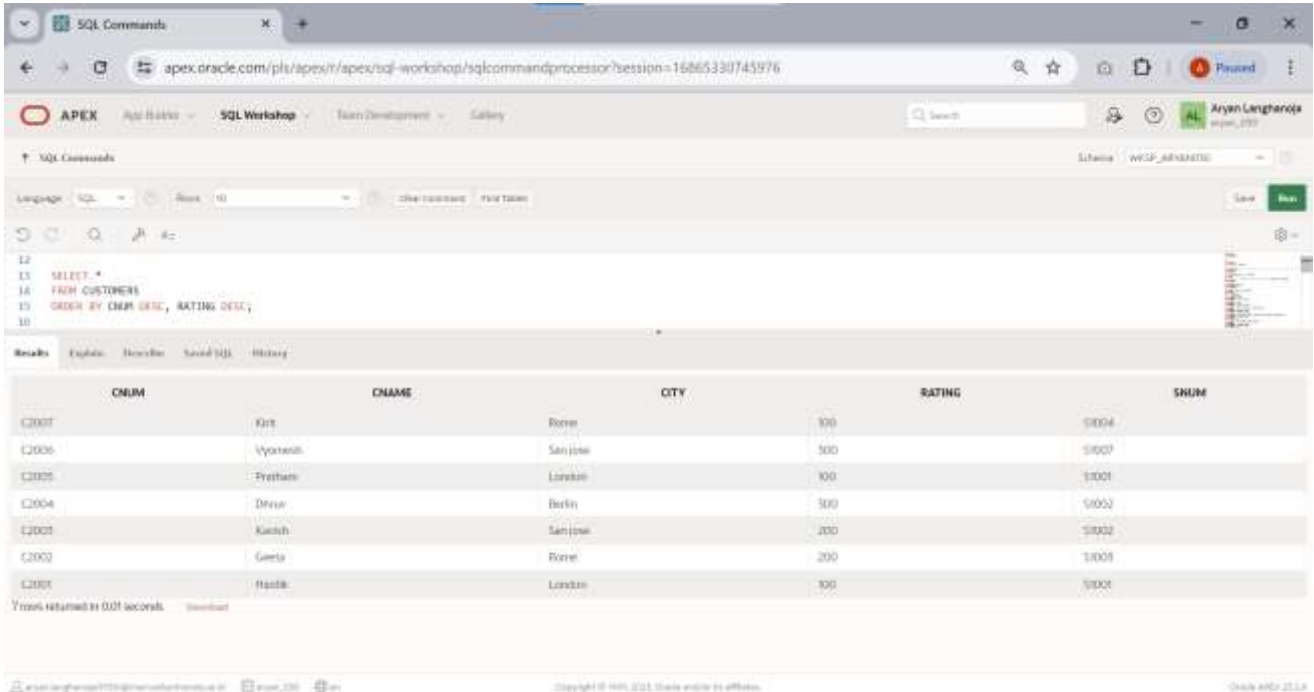
The screenshot shows the APEX SQL Workshop interface. The query entered is:

```
SELECT *
FROM ORDERS
ORDER BY CNUM DESC;
```

The results table shows the following data in descending order of CNUM:

CNUM	CNAME	CITY	RATING	SNUM
C007	SAH	Rome	800	1000
C006	Myerwell	San Jose	500	1000
C005	Frederick	London	800	1000
C004	Shiro	Berlin	500	1000
C003	Ajmer	San Jose	200	1000
C002	Leeds	Rome	200	1000
C001	HARRY	London	800	1000

3. Display all the information in descending orders(use column CNUM,AMT).



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

12 SELECT *
13 FROM CUSTOMERS
14 ORDER BY CNUM DESC, RATING DESC;
15

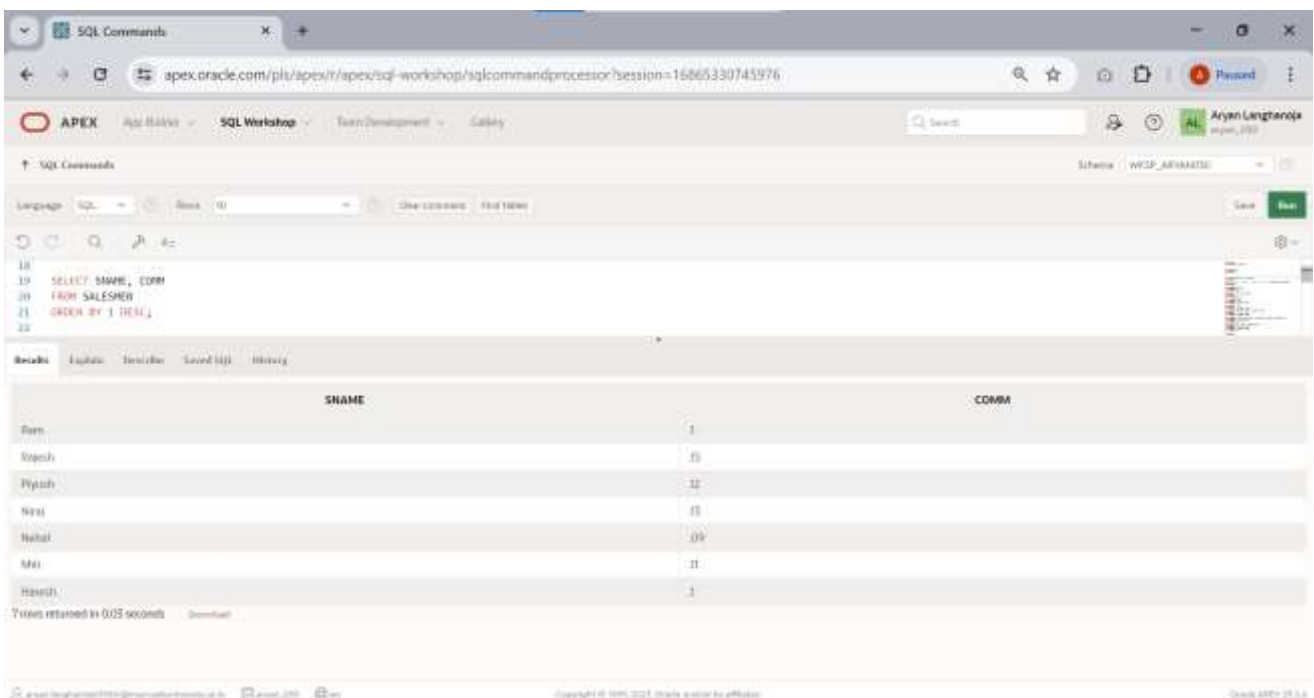
```

The results are displayed in a table with the following columns: CNUM, CNAME, CITY, RATING, and SNUM. The data is sorted in descending order by CNUM.

CNUM	CNAME	CITY	RATING	SNUM
C2007	Rita	Rome	100	S0004
C2006	Vyonens	San Jose	500	S0009
C2005	Prithvi	London	100	S0001
C2004	Dhrup	Berlin	300	S0002
C2003	Kishu	San Jose	200	S0002
C2002	Geeta	Rome	200	S0001
C2001	Hadi	London	100	S0001

7 rows returned in 0.01 seconds.

4. Display sname and comm. From salesmen in descending order(in place of column name use column number).



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

18 SELECT SNAME, COMM
19 FROM SALESMEN
20 ORDER BY 1 DESC;
21

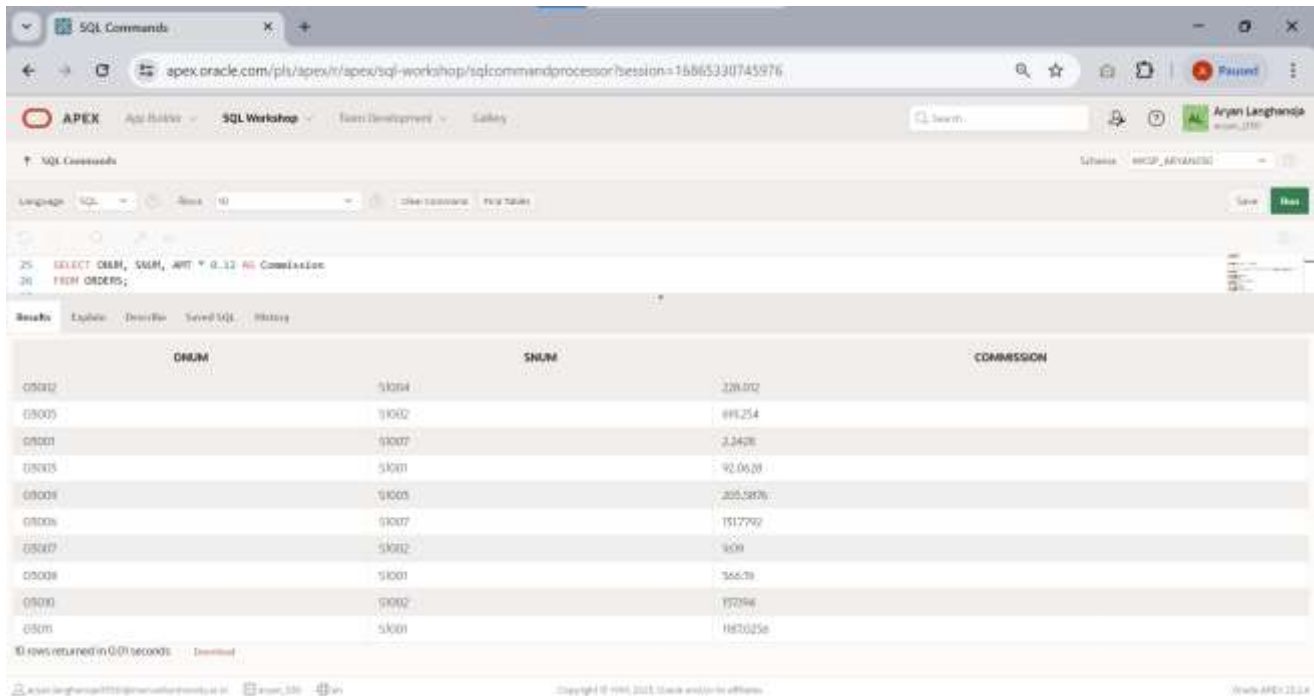
```

The results are displayed in a table with the following columns: SNAME and COMM. The data is sorted in descending order by the first column (SNAME).

SNAME	COMM
Rita	1
Kishu	10
Prithvi	12
Rita	10
Hadi	109
Mai	11
Hadi	1

7 rows returned in 0.03 seconds.

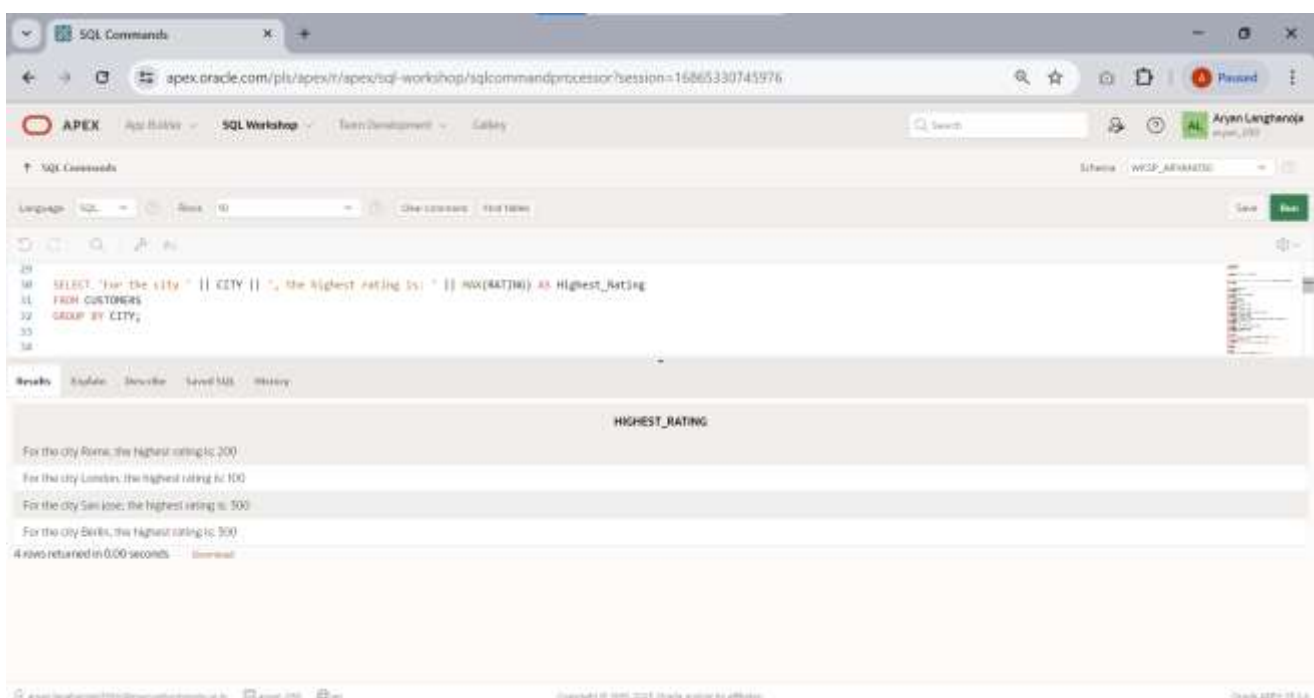
5. Assume each salesperson has a 0.12 commission. Write a query on the orders table that will produce the order number, the salesperson number and the amount of the salesperson's commission for that order.



The screenshot shows the APEX SQL Workshop interface. The query entered is: `SELECT ORDER#, SALESP#, AMT * 0.12 AS Commission FROM ORDERS;` The results are displayed in a table with columns: ORDER#, SALESP#, and COMMISSION.

ORDER#	SALESP#	COMMISSION
05012	51004	226.032
05005	51002	898.254
05001	51007	2,3428
05005	51001	92,0628
05004	51005	205,5828
05006	51007	151,7792
05007	51002	909
05008	51001	566.78
05003	51002	152946
05001	51001	1820256

6. Write a query on the customers table that will find the highest rating in each city. Put the output in this form. For the city (city) , the highest rating is: (rating).



The screenshot shows the APEX SQL Workshop interface. The query entered is: `SELECT 'For the city ' || CITY || ', the highest rating is: ' || MAX(RATING) AS Highest_Rating FROM CUSTOMERS GROUP BY CITY;` The results are displayed in a table with column: HIGHEST\_RATING.

HIGHEST_RATING
For the city Rome, the highest rating is: 200
For the city London, the highest rating is: 100
For the city San Jose, the highest rating is: 500
For the city Berlin, the highest rating is: 300

7. Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

25
26 SELECT RATING, CNAME, CNUM
27 FROM CUSTOMERS
28 ORDER BY RATING DESC;
29
30

```

The results are displayed in a table with the following data:

RATING	CNAME	CNUM
300	Dheer	C2004
300	Vijayesh	C2006
200	Raviish	C2005
200	Geeta	C2002
100	Hemish	C2003
100	Kali	C2007
100	Prashant	C2005

7 rows returned in 0.07 seconds

8. Write a query that totals the orders for each day and places the results in descending order.

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```

43
44 SELECT ODATE, SUM(AMT) AS Total_Amount
45 FROM ORDERS
46 GROUP BY ODATE
47 ORDER BY Total_Amount DESC;
48
49

```

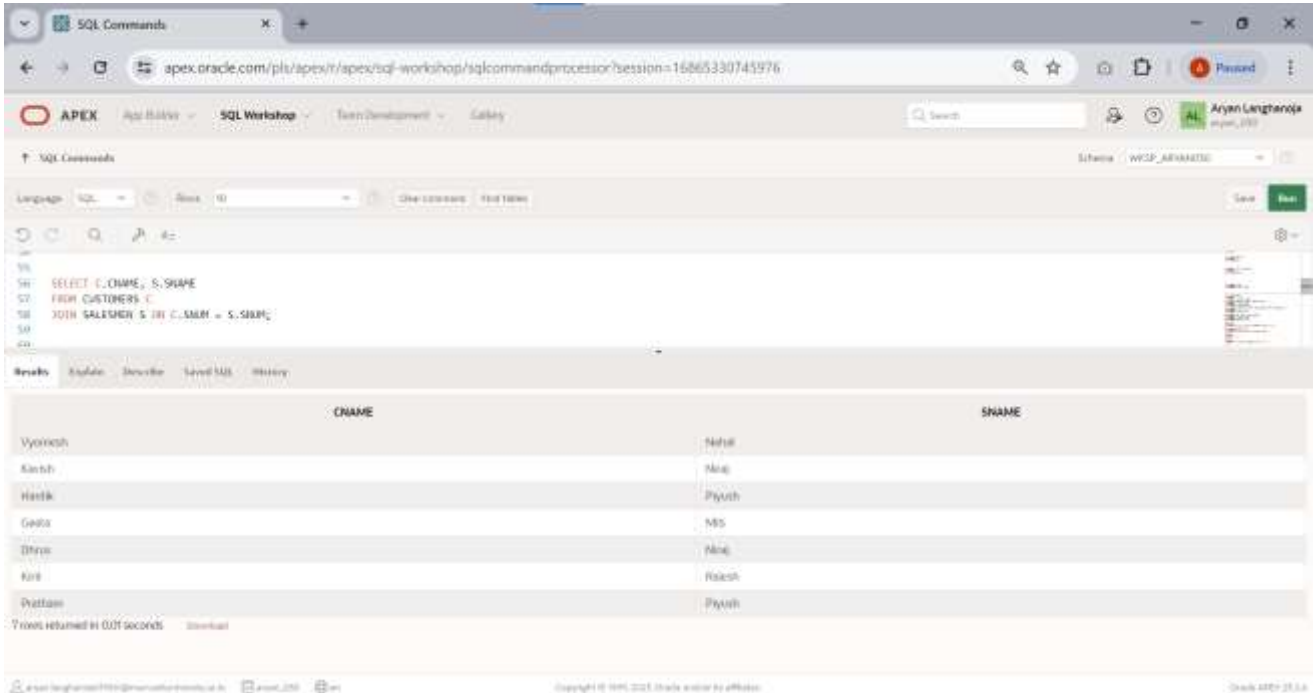
The results are displayed in a table with the following data:

ODATE	TOTAL_AMOUNT
06/10/2000	6091.88
07/10/2000	6052.97
10/04/2000	5860.45
10/05/2000	1000.01
07/06/2000	8884.04
04/10/2000	5788.08

6 rows returned in 0.07 seconds

## JOIN

9. Show the names of all customers matched with the salesmen serving them.



The screenshot shows the APEX SQL Workshop interface. The query entered is:

```

SELECT C.CNAME, S.SNAME
FROM CUSTOMERS C
JOIN SALESMEN S ON C.SALEM = S.SALEM;

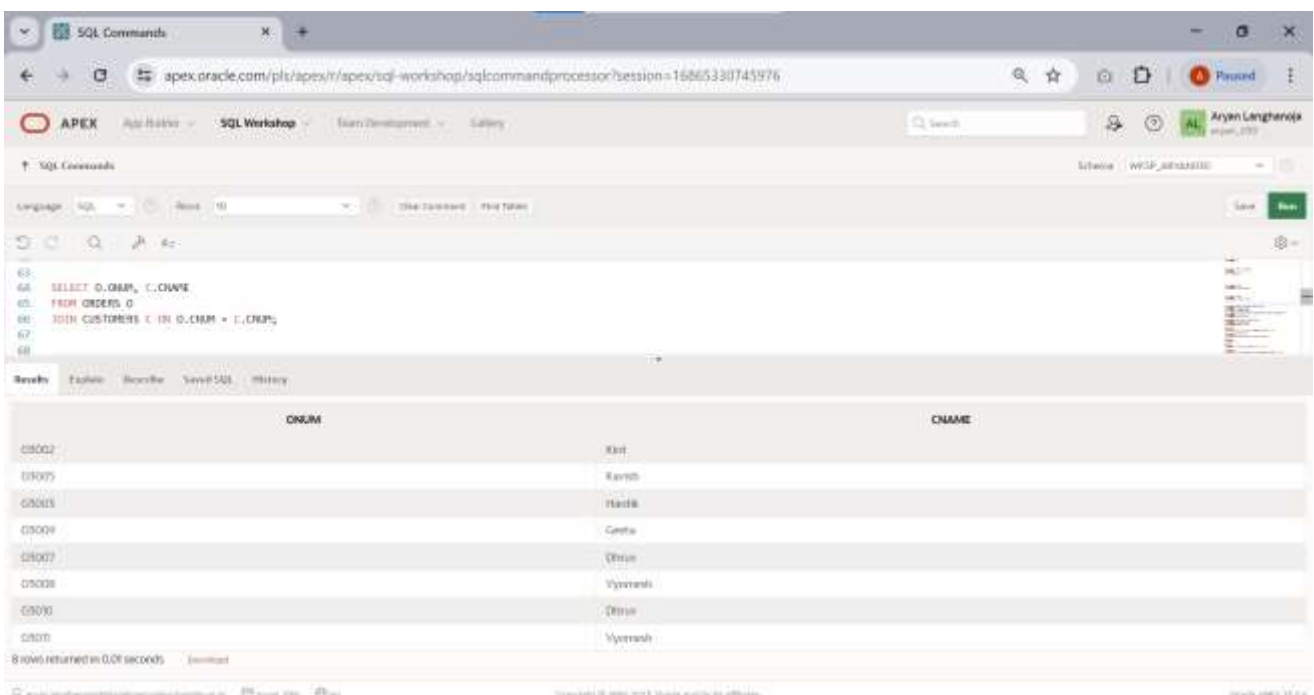
```

The results table displays the following data:

CNAME	SNAME
Vyomesh	Neha
Kirti	Neha
Hardik	Pyush
Geta	Neha
Divya	Neha
Kirti	Rakesh
Vyomesh	Pyush

7 rows returned in 0.01 seconds

10. Write a query that lists each order number followed by the name of the customer who made the order.



The screenshot shows the APEX SQL Workshop interface. The query entered is:

```

SELECT O.ORDN, C.CNAME
FROM ORDERS O
JOIN CUSTOMERS C ON O.CNUM = C.CNUM;

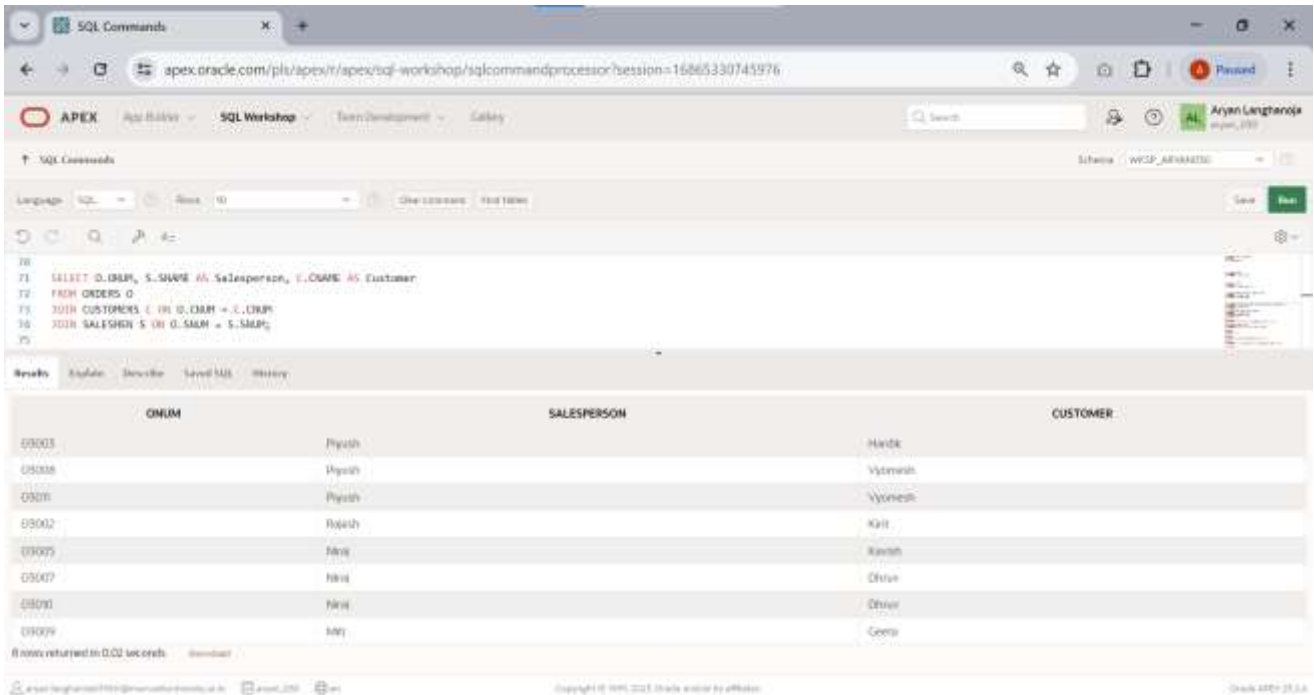
```

The results table displays the following data:

ORDN	CNAME
01002	Kirti
01005	Kirti
01003	Hardik
01009	Geta
01007	Divya
01008	Vyomesh
01010	Divya
01011	Vyomesh

8 rows returned in 0.01 seconds

11. Write a query that gives the names of both the salesperson and the customer for each order after the order number.



The screenshot shows the SQL Workshop interface with the following query and results:

```

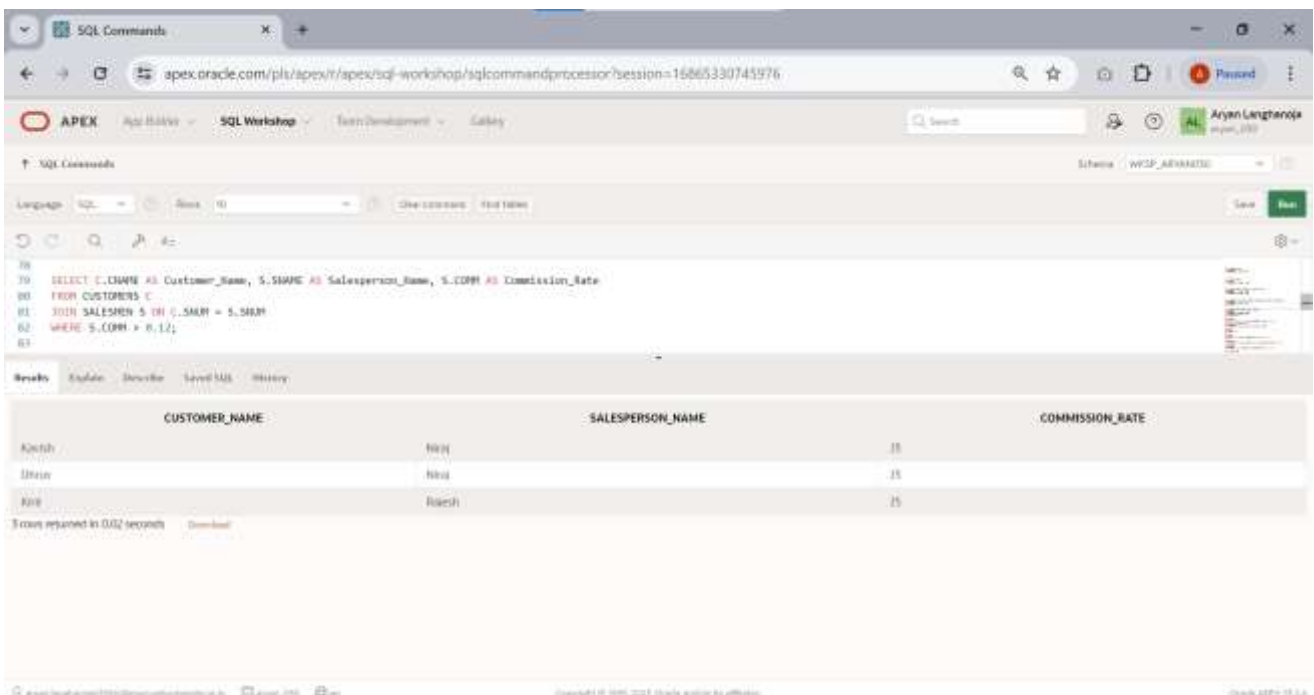
70
71 SELECT O.ORD#, S.SNAME AS Salesperson, C.CNAME AS Customer
72 FROM ORDERS O
73 JOIN CUSTOMERS C ON O.CUST# = C.CUST#
74 JOIN SALESMEN S ON O.SALE# = S.SALE#
75

```

ORD#	SALESPERSON	CUSTOMER
00003	Piyush	Hardik
00008	Piyush	Vishwesh
00016	Piyush	Vishwesh
00002	Rishabh	Kriti
00005	Amit	Rishabh
00007	Himil	Divya
00010	Himil	Divya
00009	Amit	Geeta

8 rows returned in 0.02 seconds

12. Write a query that produces all customers serviced by salesmen with a commission above 0.12. Output the customer's name, the salesperson's name and the salesperson's rate of commission.



The screenshot shows the SQL Workshop interface with the following query and results:

```

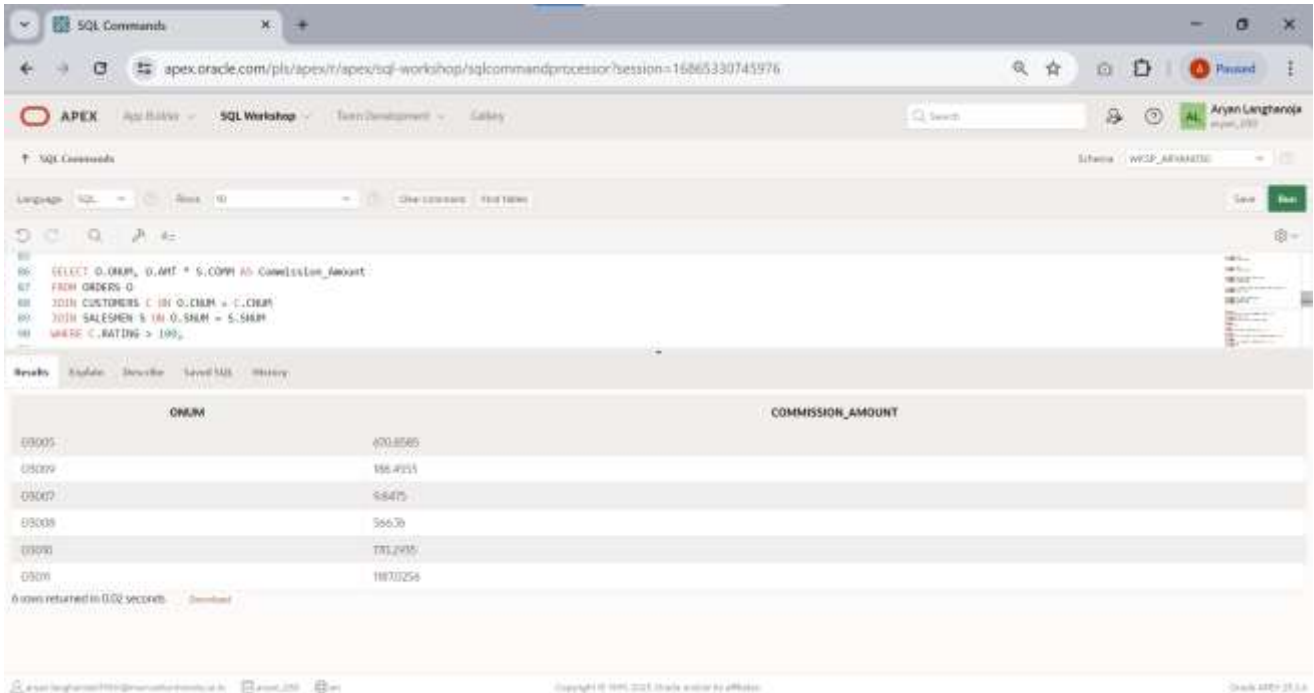
70 SELECT C.CNAME AS Customer_Name, S.SNAME AS Salesperson_Name, S.COM# AS Commission_Rate
71 FROM CUSTOMERS C
72 JOIN SALESMEN S ON C.SALE# = S.SALE#
73 WHERE S.COM# > 0.12;
74

```

CUSTOMER_NAME	SALESPERSON_NAME	COMMISSION_RATE
Kriti	Himil	.15
Divya	Himil	.15
Rishabh	Rishabh	.15

3 rows returned in 0.02 seconds

13. Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

65 SELECT O.ORD#, O.AMT * S.COMM AS Commission_Amount,
66 FROM ORDERS O
67 JOIN CUSTOMERS C ON O.CUS# = C.CUS#
68 JOIN SALESPERSON S ON O.SAL# = S.SAL#
69 WHERE C.RATING > 100;

```

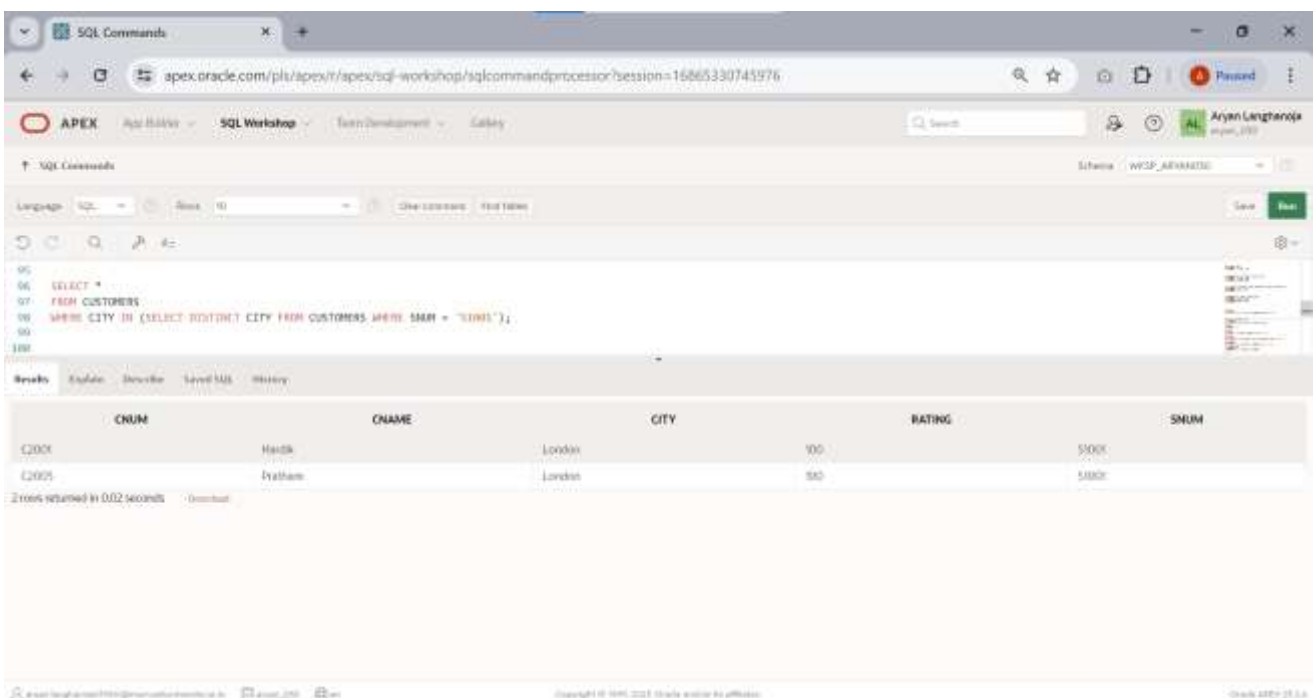
The results table shows the following data:

ORD#	COMMISSION_AMOUNT
03005	400.8585
03009	155.4255
03007	5.8475
03008	566.70
03010	731.2435
03011	1187.0254

6 rows returned in 0.02 seconds.

## OTHERS

14. List all customer located in cities where salesperson 'PIYUSH' has customer.



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

95 SELECT *
96 FROM CUSTOMERS
97 WHERE CITY IN (SELECT DISTINCT CITY FROM CUSTOMERS WHERE SAL# = '51001');

```

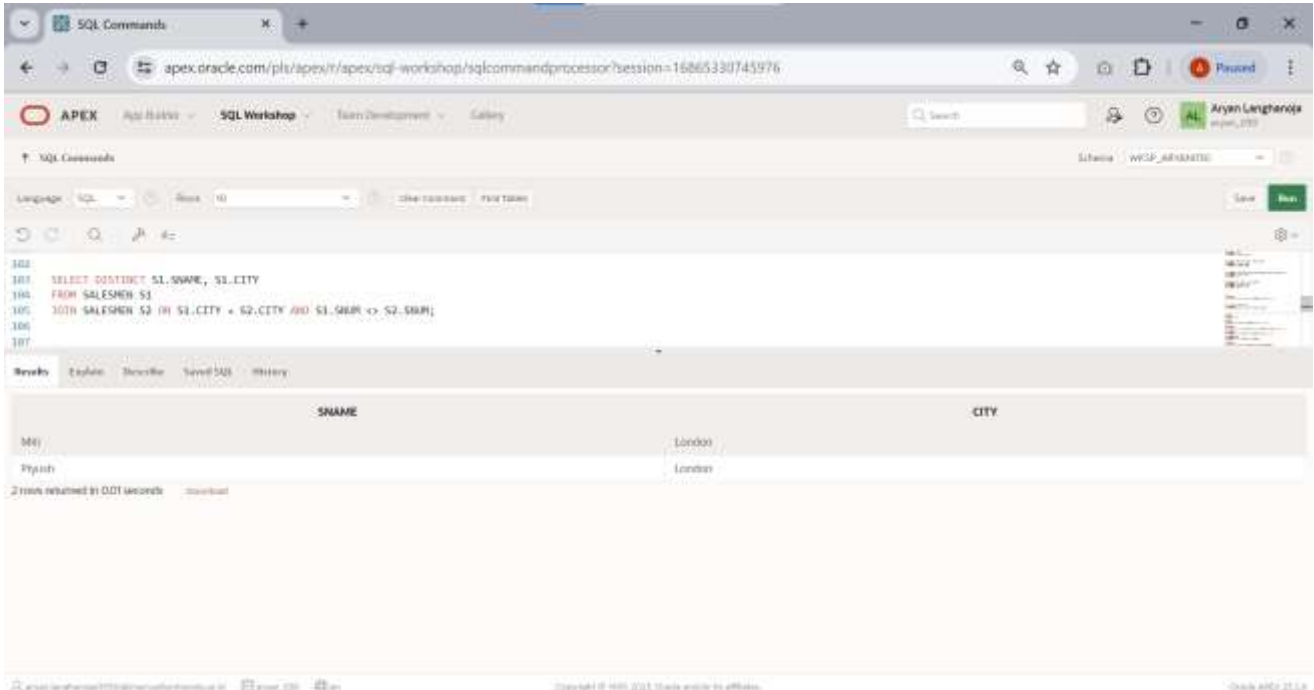
The results table shows the following data:

CUS#	CNAME	CITY	RATING	SAL#
C2001	Hardik	London	100	51001
C2005	Pratham	London	500	51001

2 rows returned in 0.02 seconds.



15. List all salesmen who are living in same city without duplicate rows.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

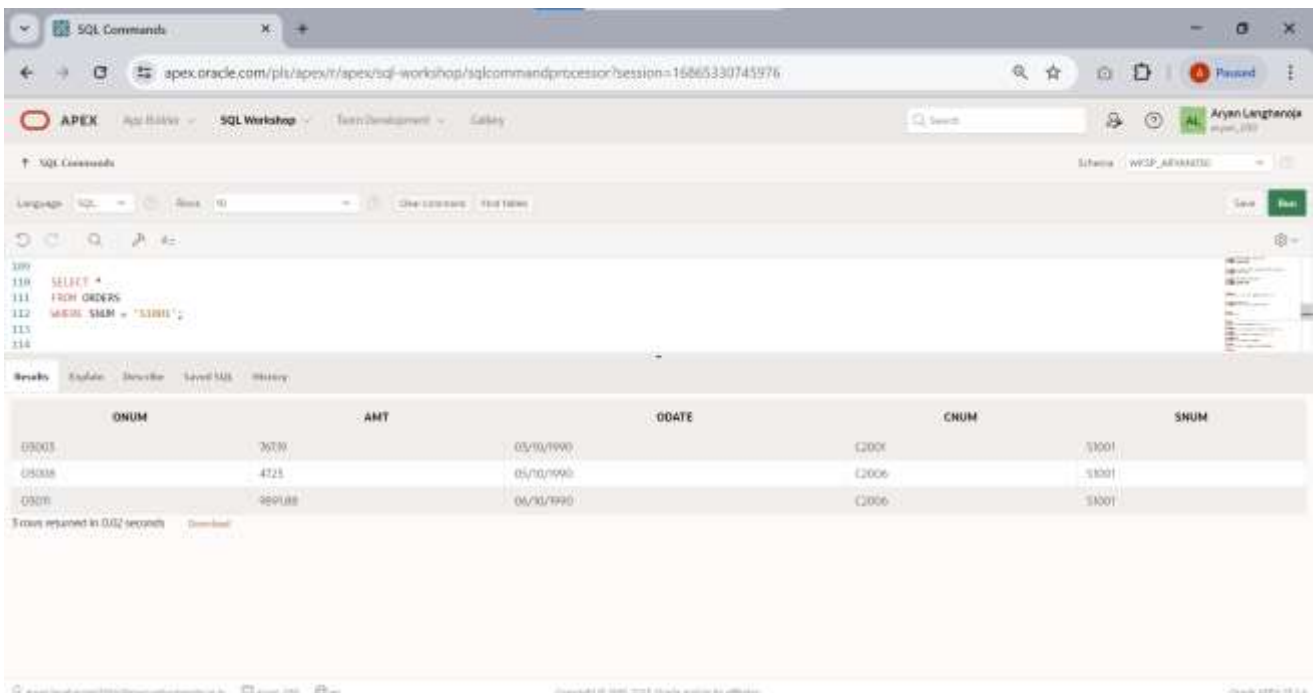
102
103 SELECT DISTINCT S1.SNAME, S1.CITY
104 FROM SALESMEN S1
105 JOIN SALESMEN S2 ON S1.CITY = S2.CITY AND S1.SNAM <> S2.SNAM;
106
107
  
```

The results table shows two rows of data:

SNAME	CITY
Mel	London
Piyush	London

2 rows returned in 0.01 seconds

16. Extract all orders of 'PIYUSH'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

109
110 SELECT *
111 FROM ORDERS
112 WHERE SNAME = 'PIYUSH';
113
114
  
```

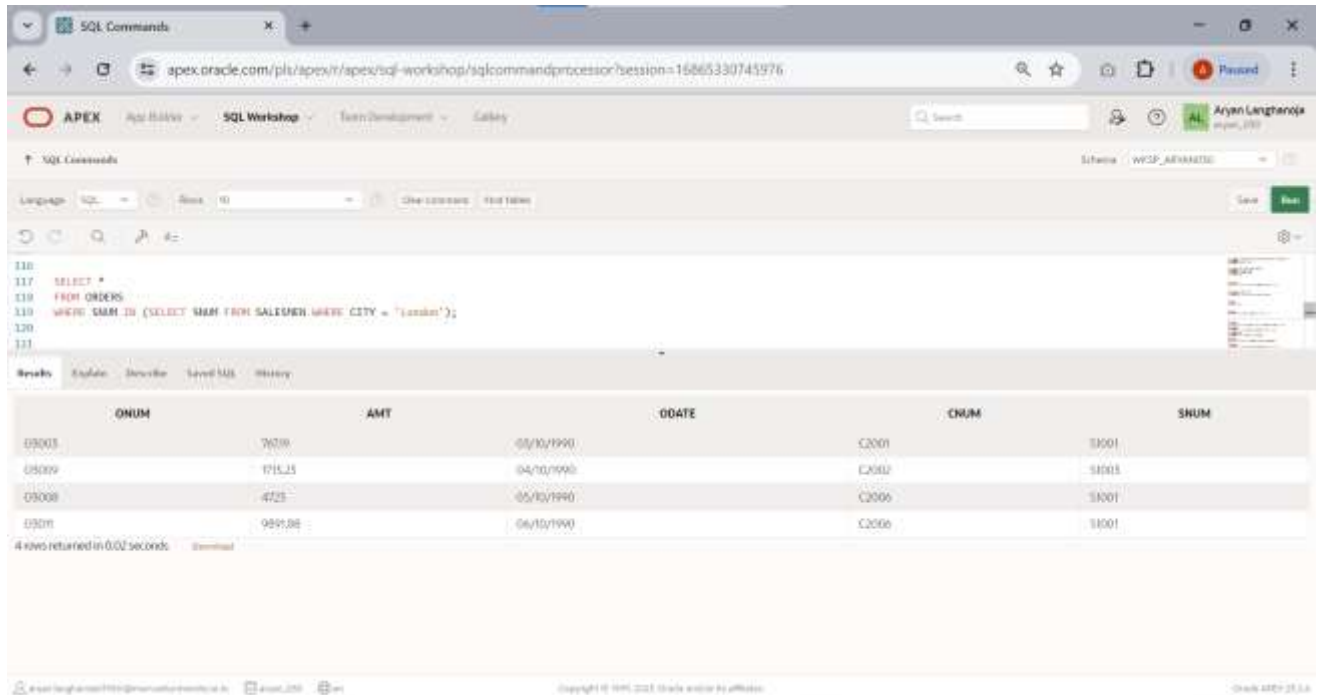
The results table shows three rows of data:

ONUM	AMT	ODATE	CNUM	SNAME
09003	30739	05/10/1999	C2004	PIYUSH
09008	4725	05/10/1999	C2006	PIYUSH
09016	9849.88	06/10/1999	C2006	PIYUSH

3 rows returned in 0.02 seconds



17. Extract all orders of LONDON'S salesmen.



```

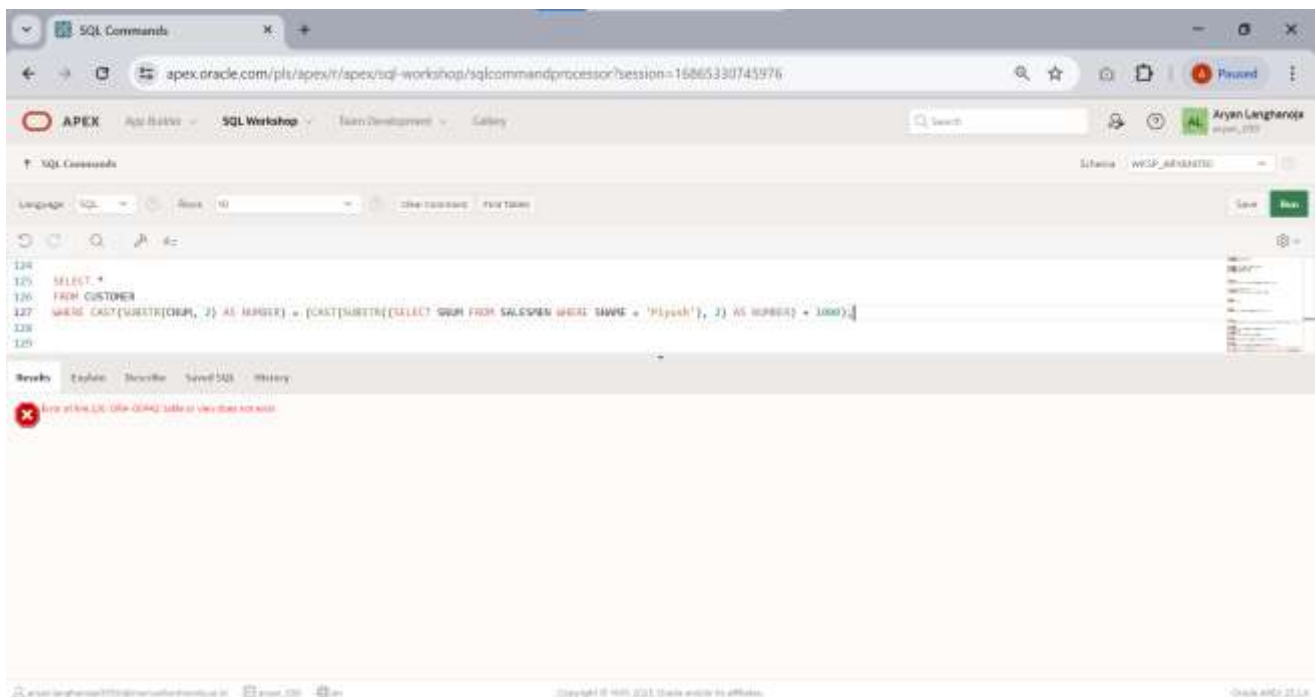
116 SELECT *
117 FROM ORDERS
118 WHERE SNUM IN (SELECT SNUM FROM SALESMEN WHERE CITY = 'London');
119
120
121

```

ONUM	AMT	ODATE	CNUM	SNUM
09003	76709	03/10/1990	C2001	S0001
09009	1715.25	04/10/1990	C2002	S0003
09008	4725	05/10/1990	C2006	S0001
09011	9891.88	06/10/1990	C2006	S0001

4 rows returned in 0.02 seconds

18. Find all customers whose cnum is 1000 above than the snum of 'PIYUSH'.



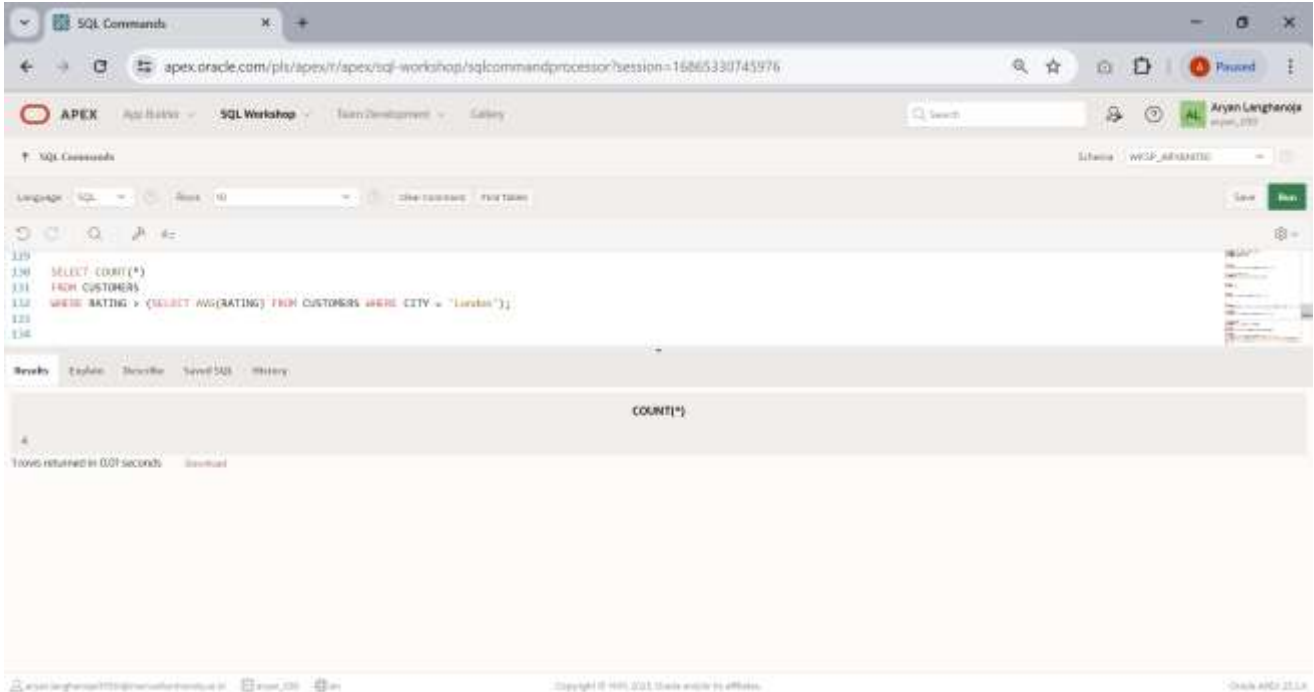
```

124 SELECT *
125 FROM CUSTOMER
126 WHERE CAST(SUBSTR(CNUM, 2) AS NUMBER) = (CAST(SUBSTR((SELECT SNUM FROM SALESMEN WHERE SNAME = 'Piyush'), 2) AS NUMBER) + 1000);
127
128
129

```

Error: ORA-00942: table or view does not exist

19. Count the no. of customers with the rating above than average Rating of 'LONDON'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

129
130 SELECT COUNT(*)
131 FROM CUSTOMERS
132 WHERE RATING > (SELECT AVG(RATING) FROM CUSTOMERS WHERE CITY = 'London');
133
134

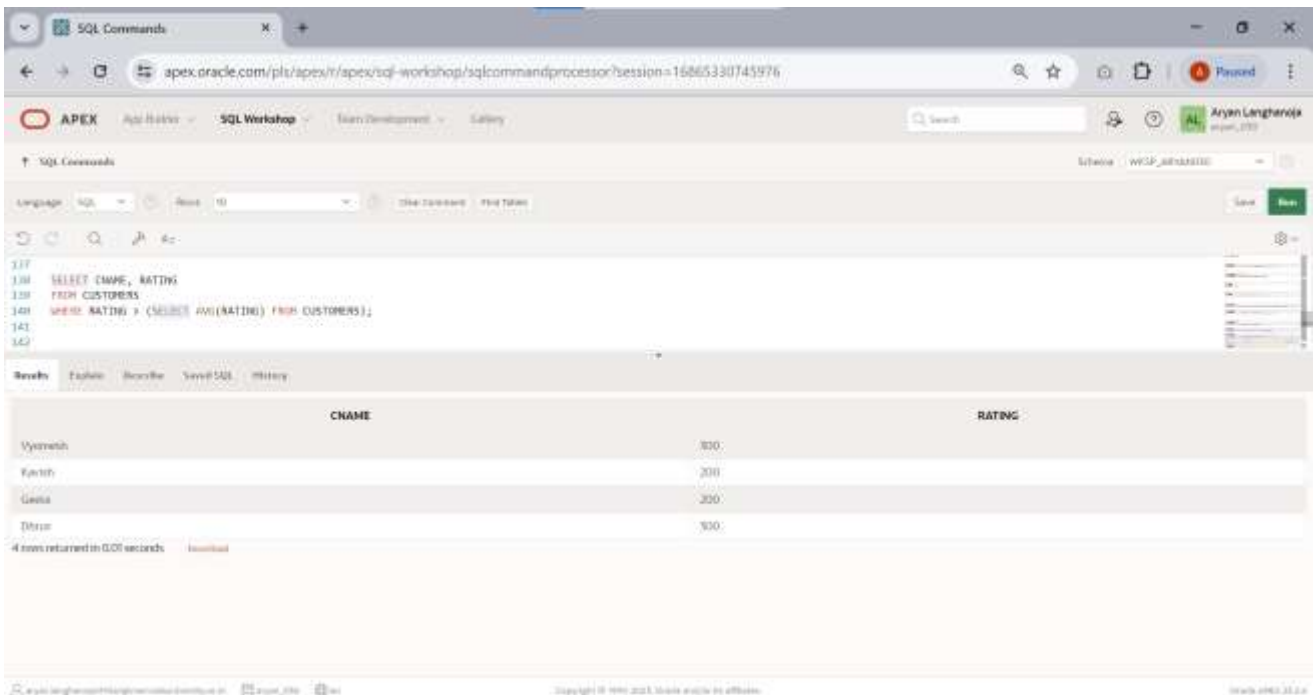
```

The results pane shows a single row with the value 4 under the column header COUNT(\*).

COUNT(*)
4

4 rows returned in 0.07 seconds

20. Produce the name and rating of all customers who have above average Rating.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

137
138 SELECT CNAME, RATING
139 FROM CUSTOMERS
140 WHERE RATING > (SELECT AVG(RATING) FROM CUSTOMERS);
141
142

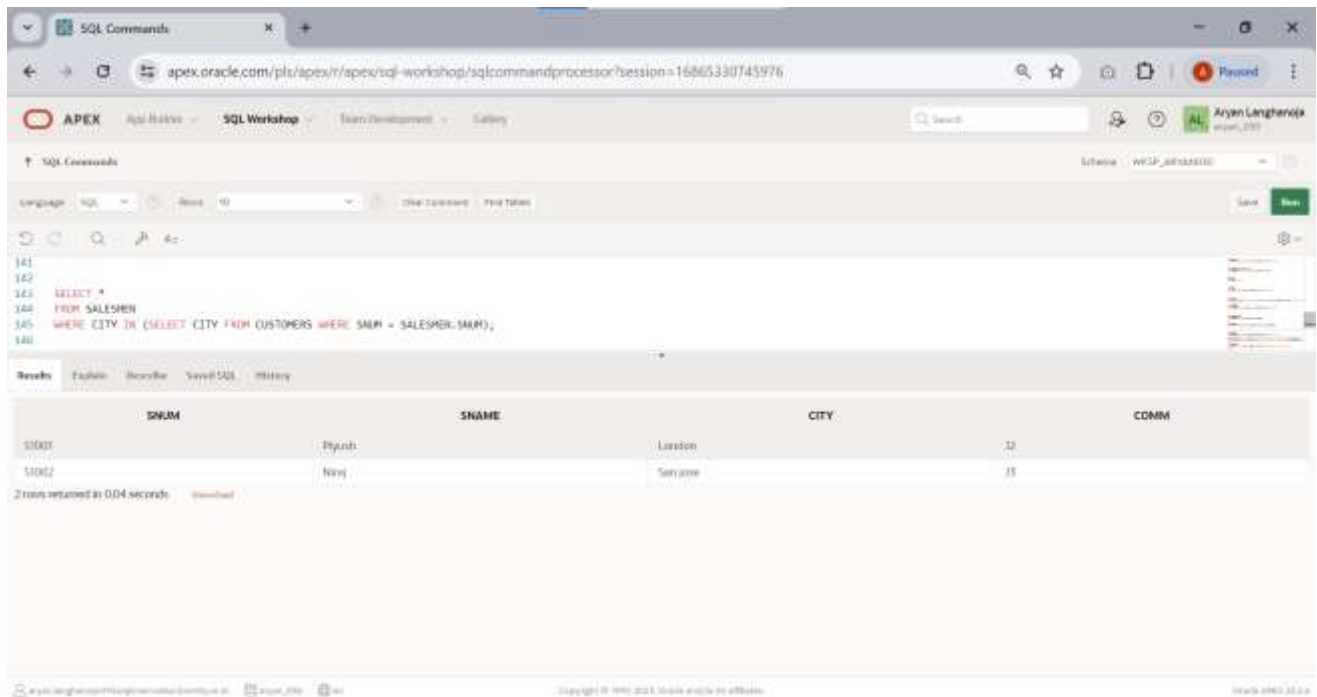
```

The results pane shows a table with two columns: CNAME and RATING. The data rows are:

CNAME	RATING
Vyomesh	800
Kashif	200
Ganesh	200
Dhruv	300

4 rows returned in 0.05 seconds

21. List all salesmen with customers located in their cities.



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

141
142
143 SELECT *
144 FROM SALESMEN
145 WHERE CITY IN (SELECT CITY FROM CUSTOMERS WHERE SALES = SALESPEX.SALES);
146

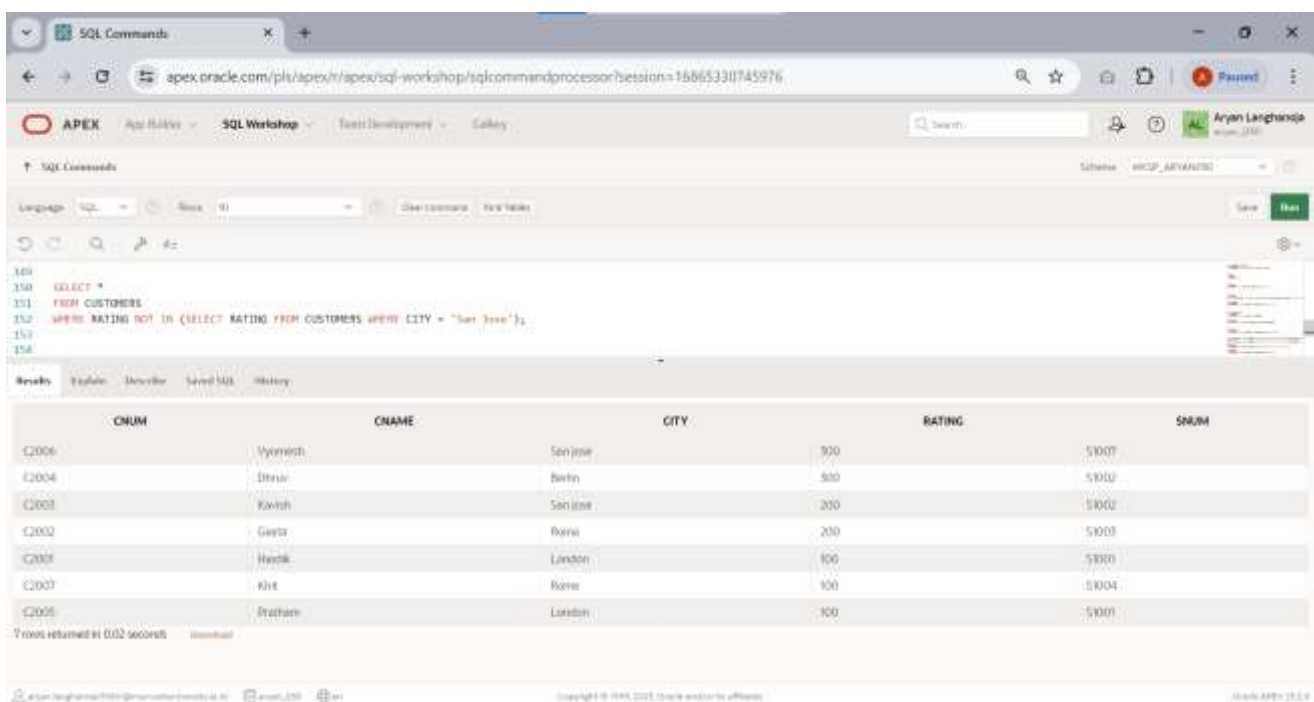
```

The results table shows the following data:

SNUM	SNAME	CITY	COMM
S1001	Ryan	London	12
S1002	Nora	San Jose	15

2 rows returned in 0.04 seconds.

22. Select all customers whose rating doesn't match with any rating customer of 'SAN JOSE'.



The screenshot shows the Oracle APEX SQL Workshop interface. The query entered is:

```

149
150 SELECT *
151 FROM CUSTOMERS
152 WHERE RATING NOT IN (SELECT RATING FROM CUSTOMERS WHERE CITY = 'San Jose');
153
154

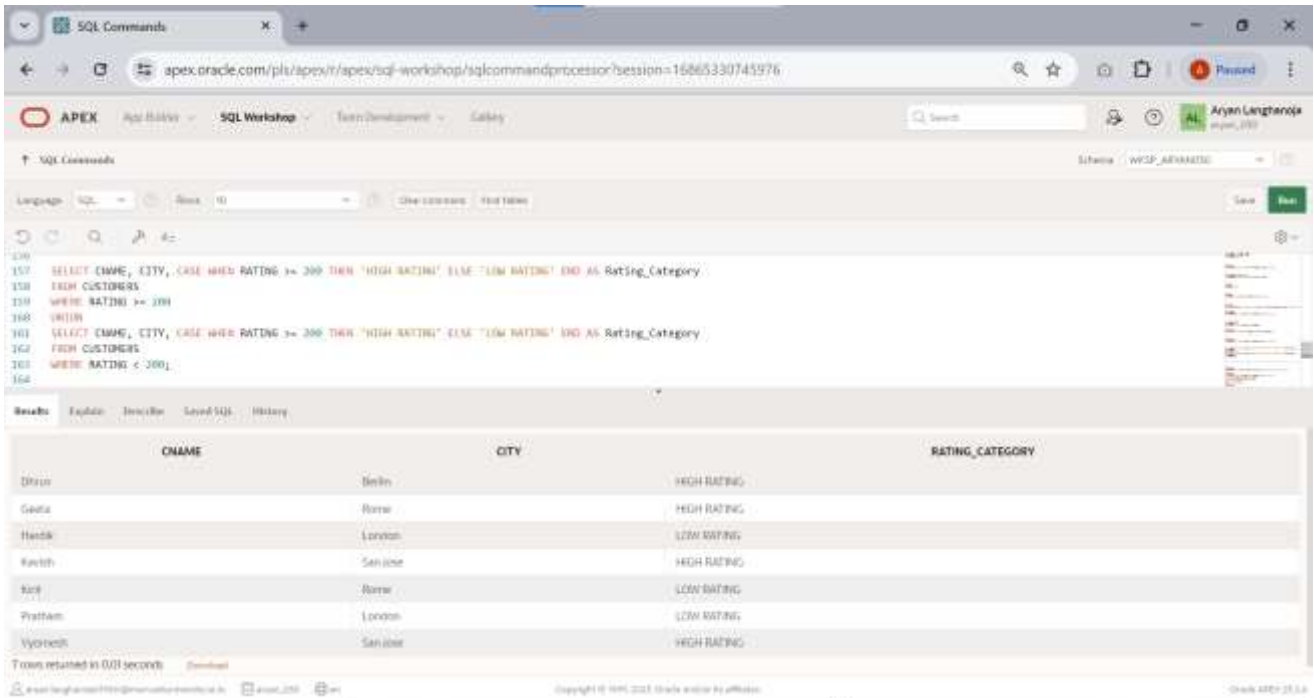
```

The results table shows the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2006	Myeresh	San Jose	900	S1007
C2004	Dharu	Berlin	800	S1002
C2003	Kurish	San Jose	200	S1002
C2002	Gera	Rome	200	S1003
C2008	Hack	London	100	S1001
C2007	Kut	Rome	100	S1004
C2005	Pratham	London	100	S1001

7 rows returned in 0.02 seconds.

23. Create a union of two queries that shows the names, cities and ratings of all customers. Those with rating of  $\geq 200$  should display 'HIGH RATING' and those with  $< 200$  should display 'LOW RATING'.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

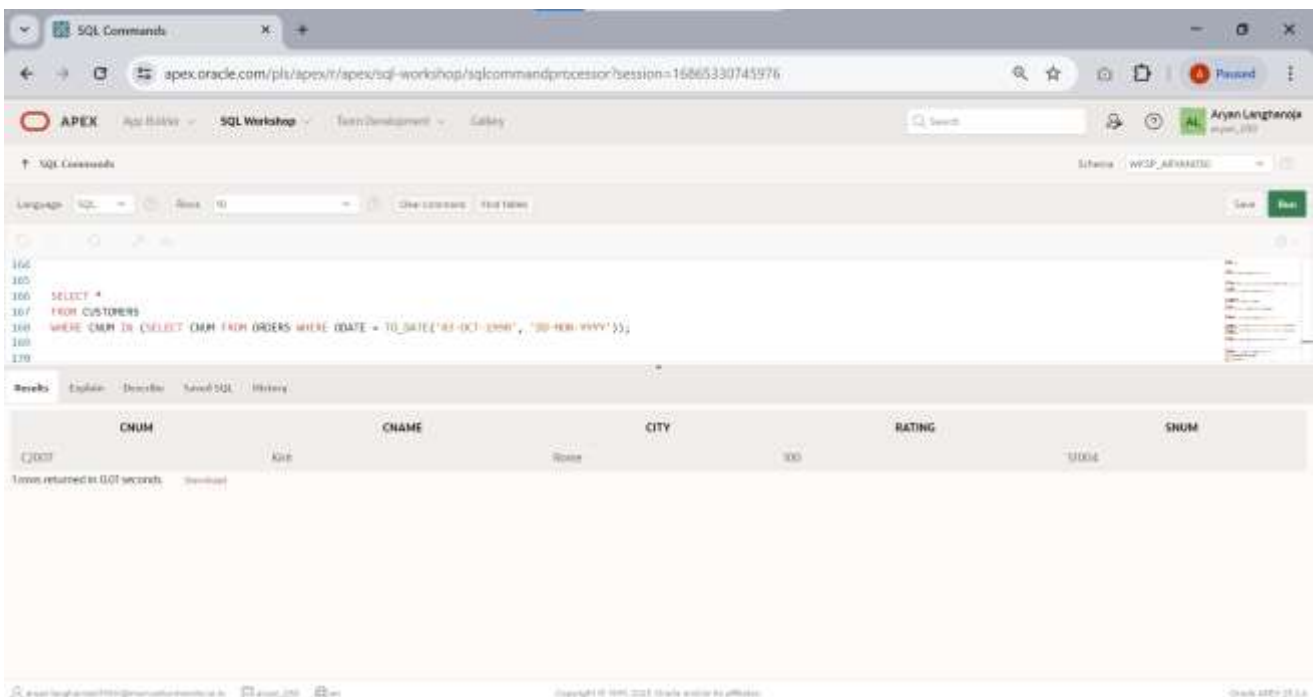
156 SELECT CNAME, CITY, CASE WHEN RATING >= 200 THEN 'HIGH RATING' ELSE 'LOW RATING' END AS Rating_Category
157 FROM CUSTOMERS
158 WHERE RATING >= 200
159 UNION
160 SELECT CNAME, CITY, CASE WHEN RATING < 200 THEN 'HIGH RATING' ELSE 'LOW RATING' END AS Rating_Category
161 FROM CUSTOMERS
162 WHERE RATING < 200;
163

```

The Results tab shows the following data:

CNAME	CITY	RATING_CATEGORY
Diana	Berlin	HIGH RATING
Geeta	Rome	HIGH RATING
Hendrik	London	LOW RATING
Kersti	San Jose	HIGH RATING
Kofi	Rome	LOW RATING
Prattima	London	LOW RATING
Vyornesh	San Jose	HIGH RATING

24. Find all customers with orders on 3<sup>rd</sup> october 1990 using correlate sub query.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following query:

```

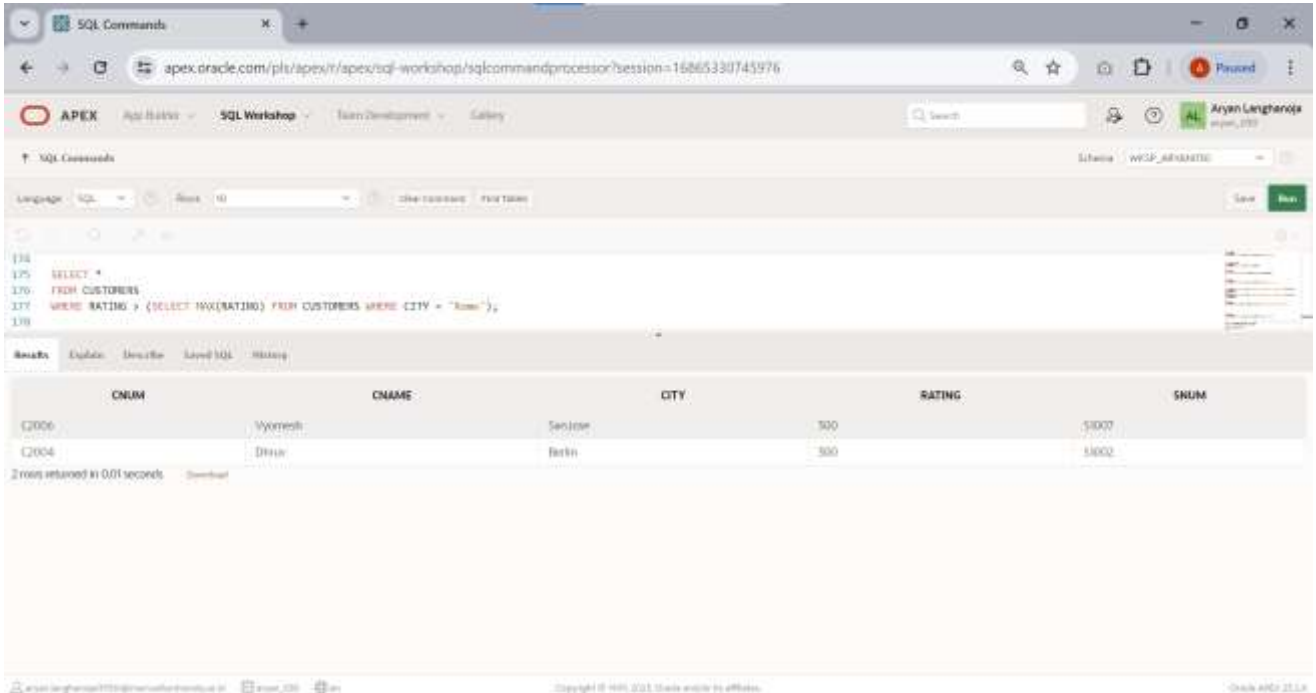
166 SELECT *
167 FROM CUSTOMERS
168 WHERE CNUM IN (SELECT CNUM FROM ORDERS WHERE ODATE = TO_DATE('03-OCT-1990', 'DD-MON-YYYY'));
169

```

The Results tab shows the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2003	Kofi	Rome	100	US004

25. Find all customers having rating greater than any customer in 'ROME'.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

136 SELECT *
137 FROM CUSTOMERS
138 WHERE RATING > (SELECT MAX(RATING) FROM CUSTOMERS WHERE CITY = 'ROME');
139

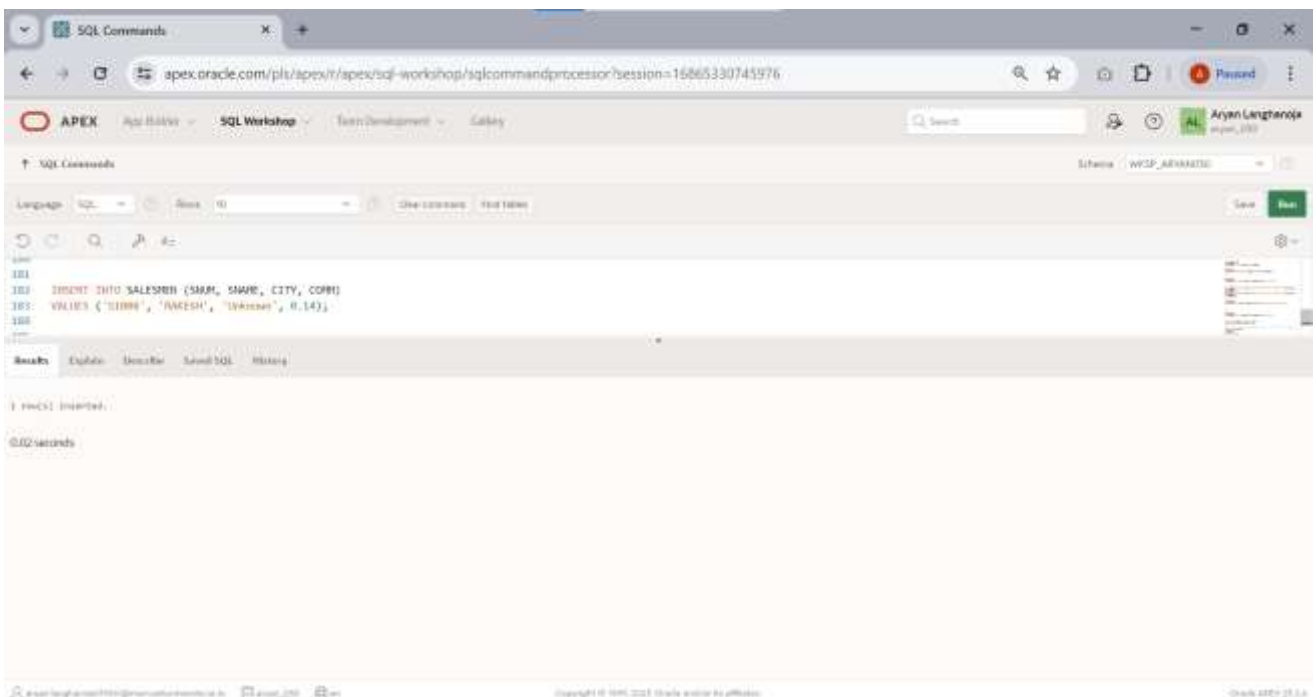
```

The results table shows the following data:

CNUM	CNAME	CITY	RATING	SNUM
C2006	Vyomesh	San Jose	500	S1007
C2004	Dhrui	Berlin	500	S1002

2 rows returned in 0.01 seconds

26. Insert a row into salesmen table with the values snum is s1008,salesmen name is 'RAKESH', city is unknown and commission is 14%.



The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```

131
132 INSERT INTO SALESMEN (SNUM, SNAME, CITY, COMM)
133 VALUES ('S1008', 'RAKESH', 'UNKNOWN', 0.14);
134

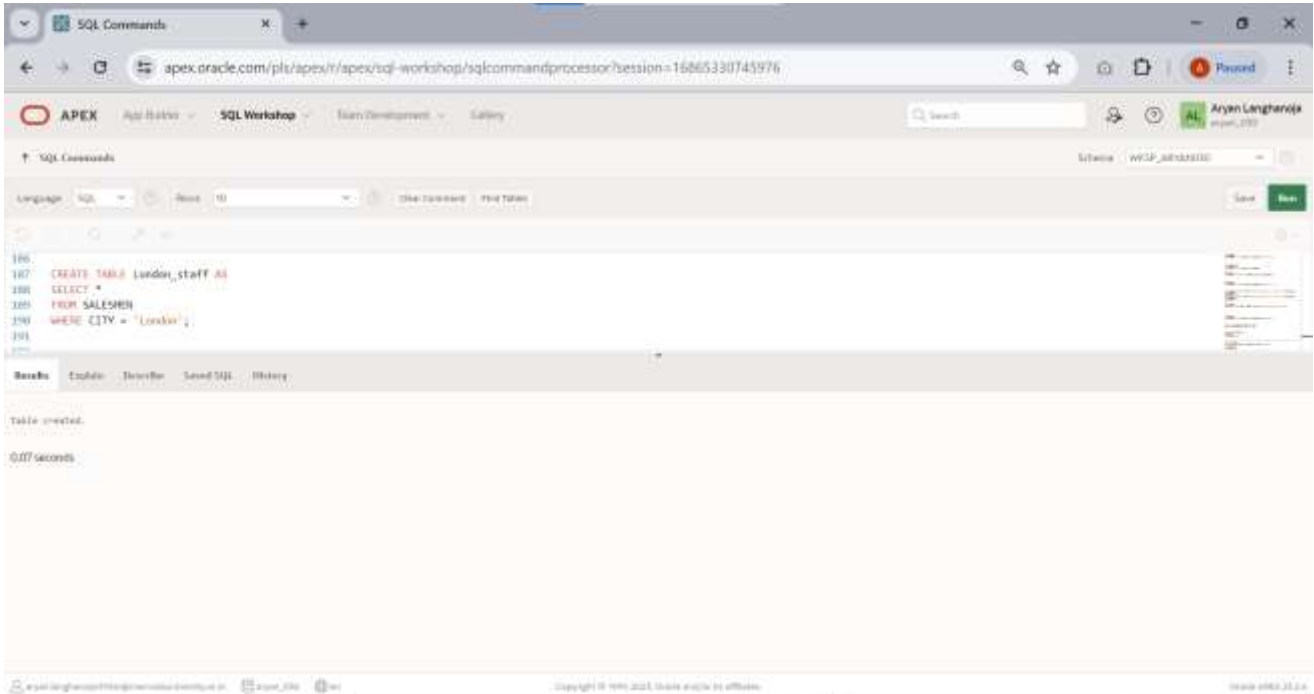
```

The results section shows:

1 row(s) inserted.

0.02 seconds

27. Create another table London\_staff having same structure as salesmen table.

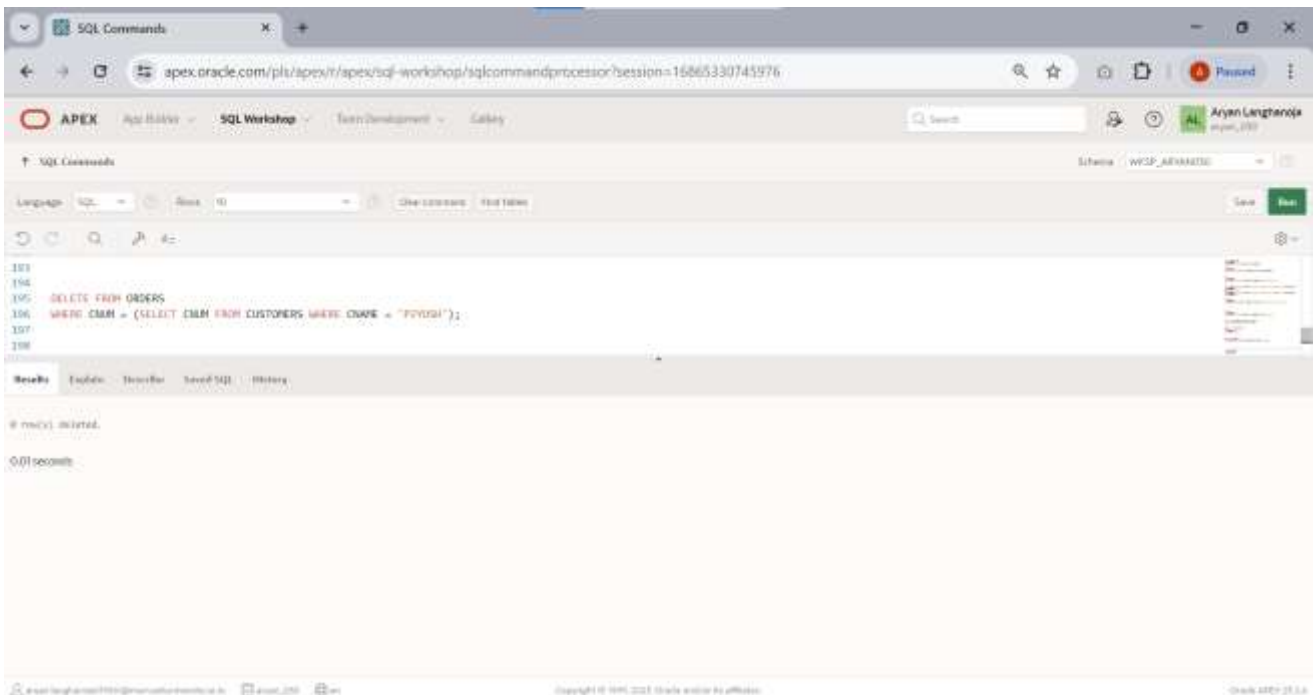


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
186 CREATE TABLE London_staff AS
187 SELECT *
188 FROM SALESMEN
189 WHERE CITY = 'London';
190
```

The results section shows: "Table created." and "0.07 seconds".

28. Delete all orders from customer 'PIYUSH' from the order table.

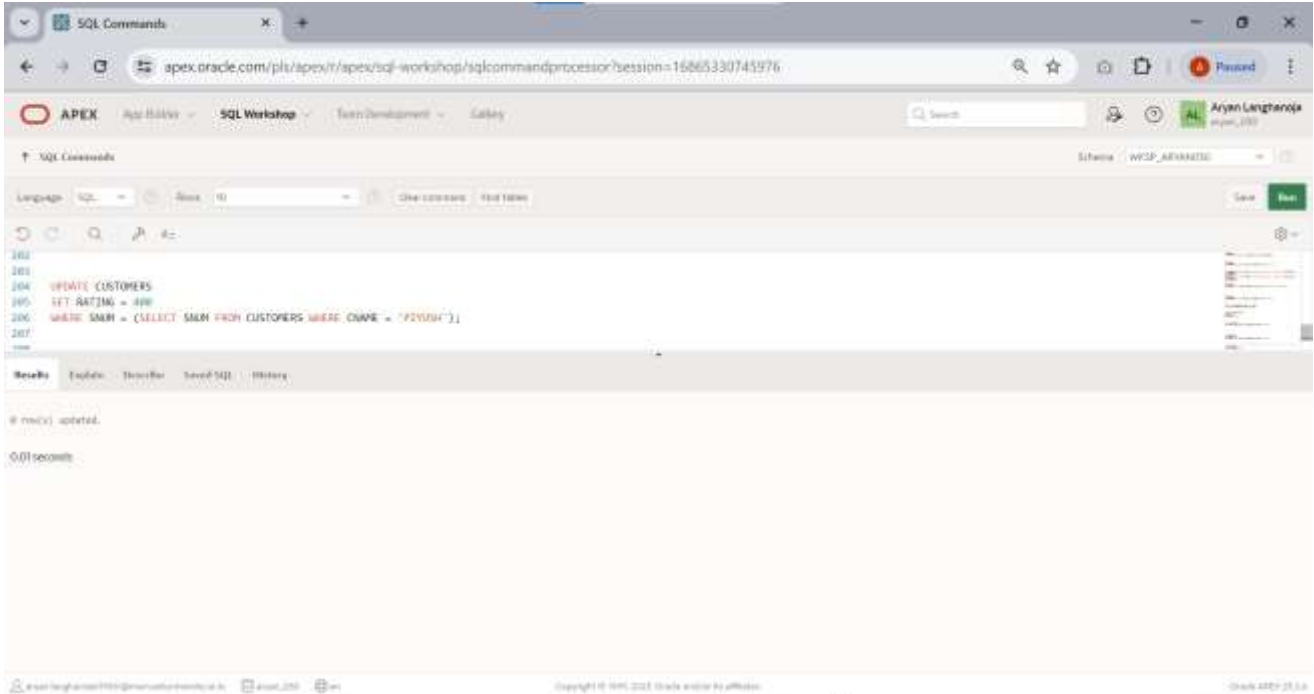


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
193
194
195 DELETE FROM ORDERS
196 WHERE CNAME = (SELECT CNAME FROM CUSTOMERS WHERE CNAME = 'PIYUSH');
197
198
```

The results section shows: "0 row(s) deleted." and "0.01 seconds".

29. Set the ratings of all the customers of PIYUSH to 400.



The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```

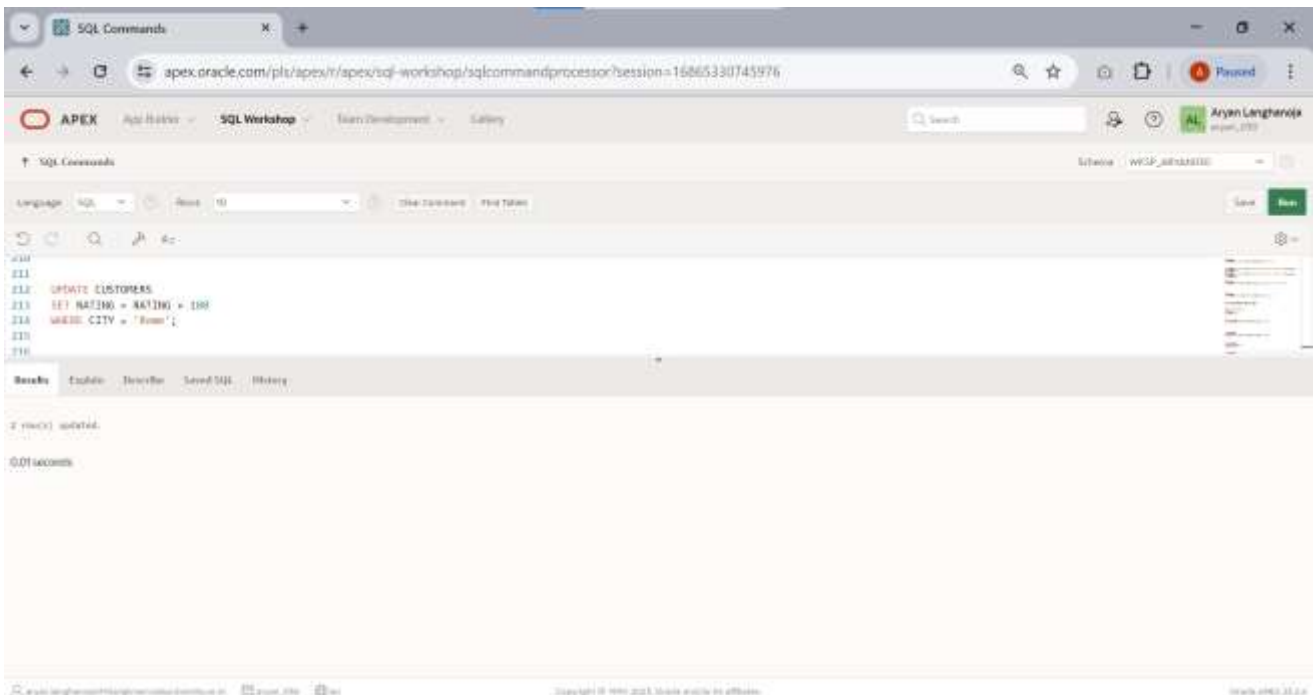
202
203
204 UPDATE CUSTOMERS
205 SET RATING = 400
206 WHERE NAME = (SELECT NAME FROM CUSTOMERS WHERE NAME = 'PIYUSH');
207
  
```

The Results pane shows the execution output:

```

8 row(s) updated.
0.01 seconds
  
```

30. Increase the rating of all the customers in ROME by 100.



The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```

210
211
212 UPDATE CUSTOMERS
213 SET RATING = RATING + 100
214 WHERE CITY = 'Rome';
215
  
```

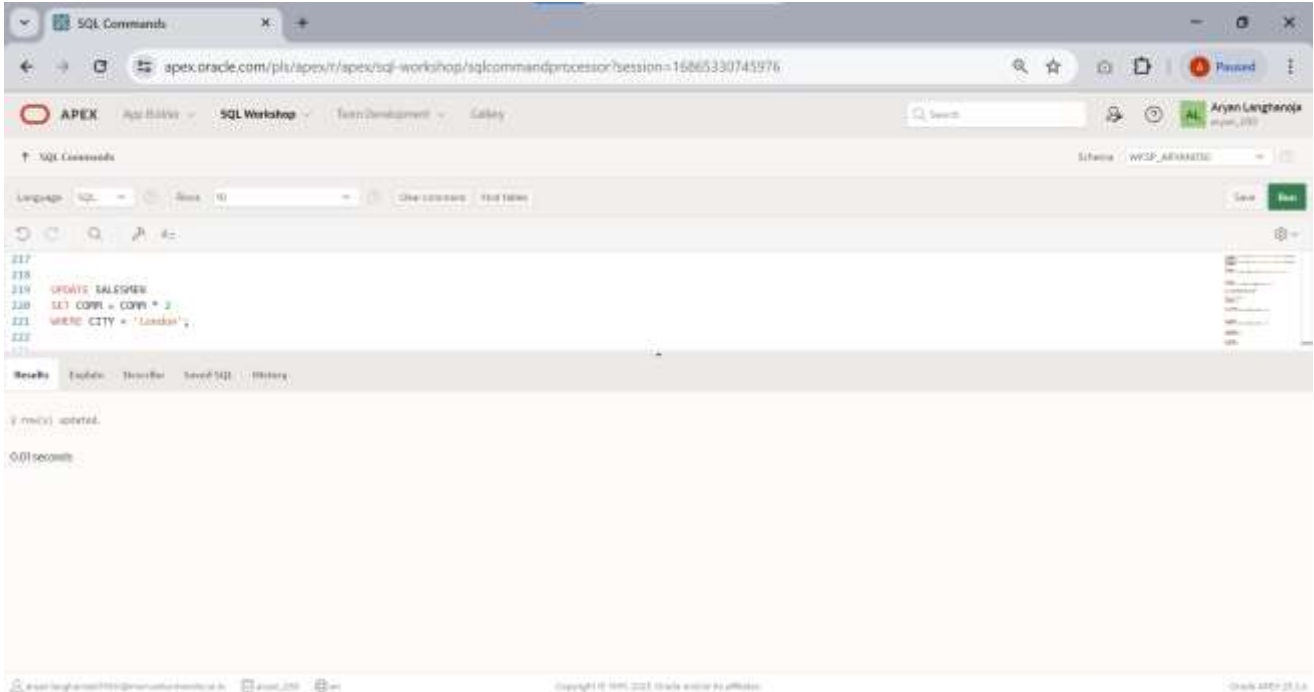
The Results pane shows the execution output:

```

2 row(s) updated.
0.01 seconds
  
```



31. Double the commission of all salesmen of LONDON.

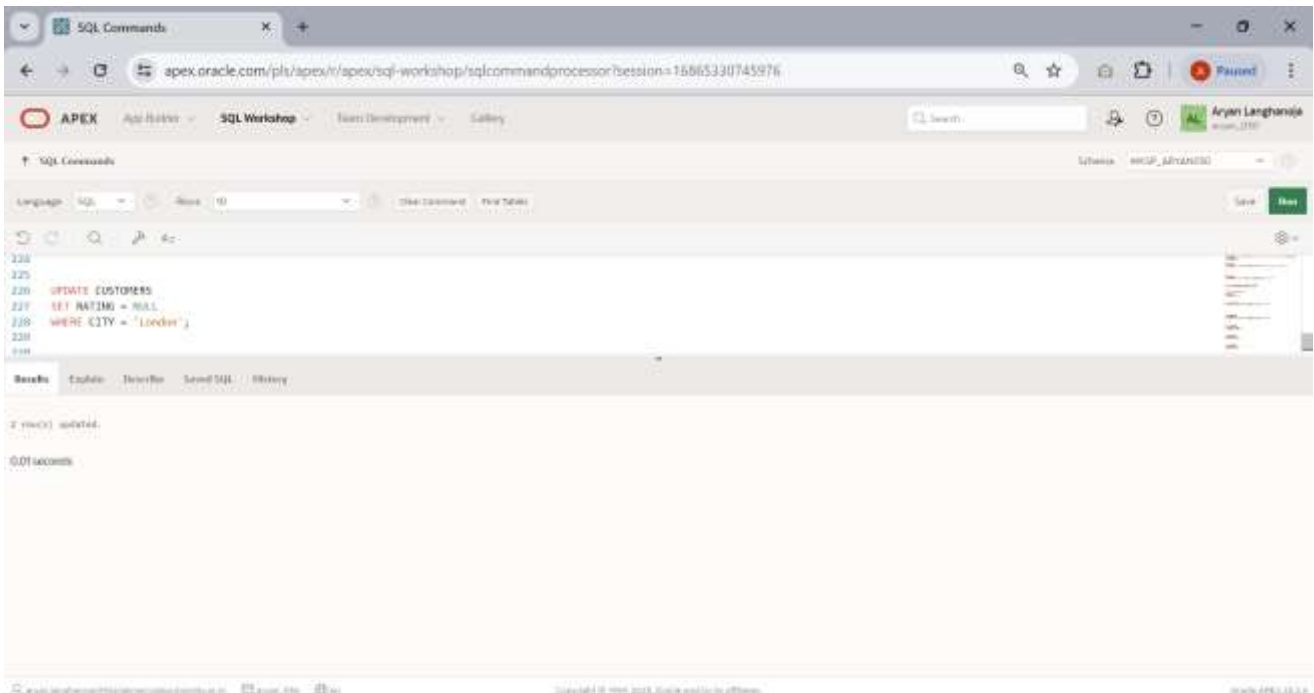


The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
217  
218  
219 UPDATE SALESMEN  
220 SET COMM = COMM * 2  
221 WHERE CITY = 'London';  
222
```

The results pane shows: 2 row(s) updated. 0.01 seconds.

32. Set ratings for all customers in LONDON to NULL.



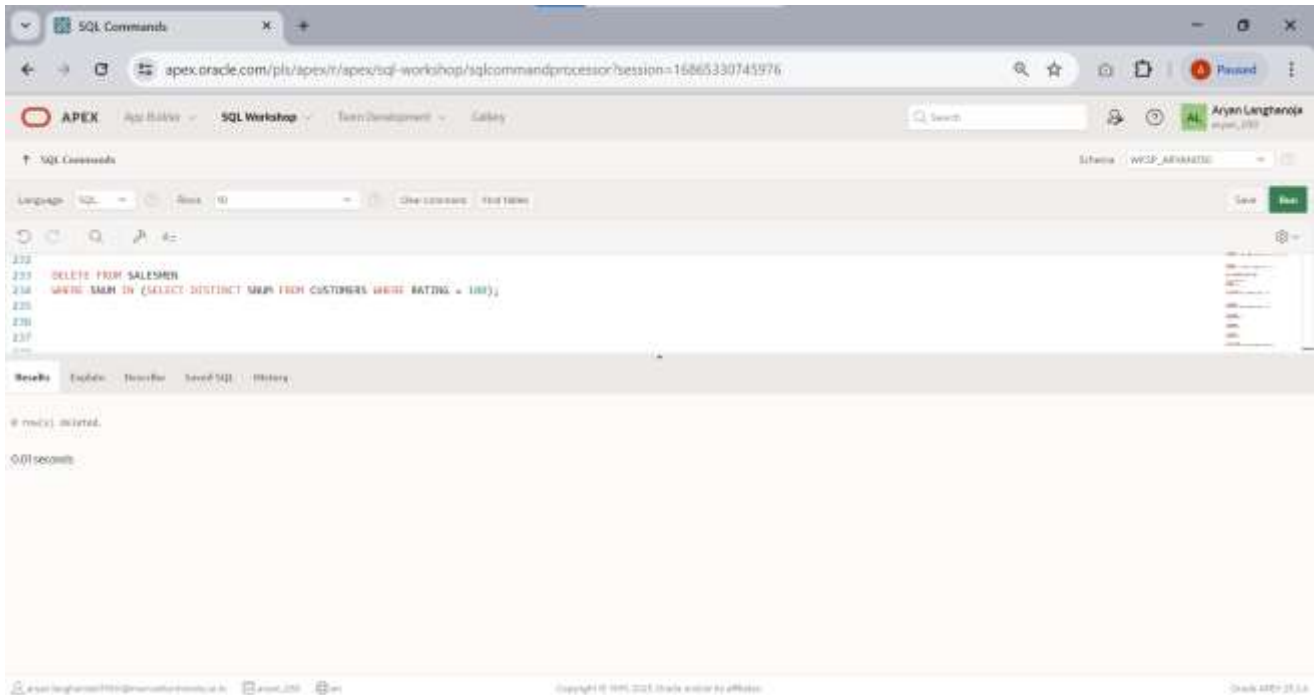
The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
223  
224  
225 UPDATE CUSTOMERS  
226 SET RATING = NULL  
227 WHERE CITY = 'London';  
228
```

The results pane shows: 2 row(s) updated. 0.01 seconds.



33. Delete all salesmen who have at least one customer with a rating of 100 from salesmen table.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

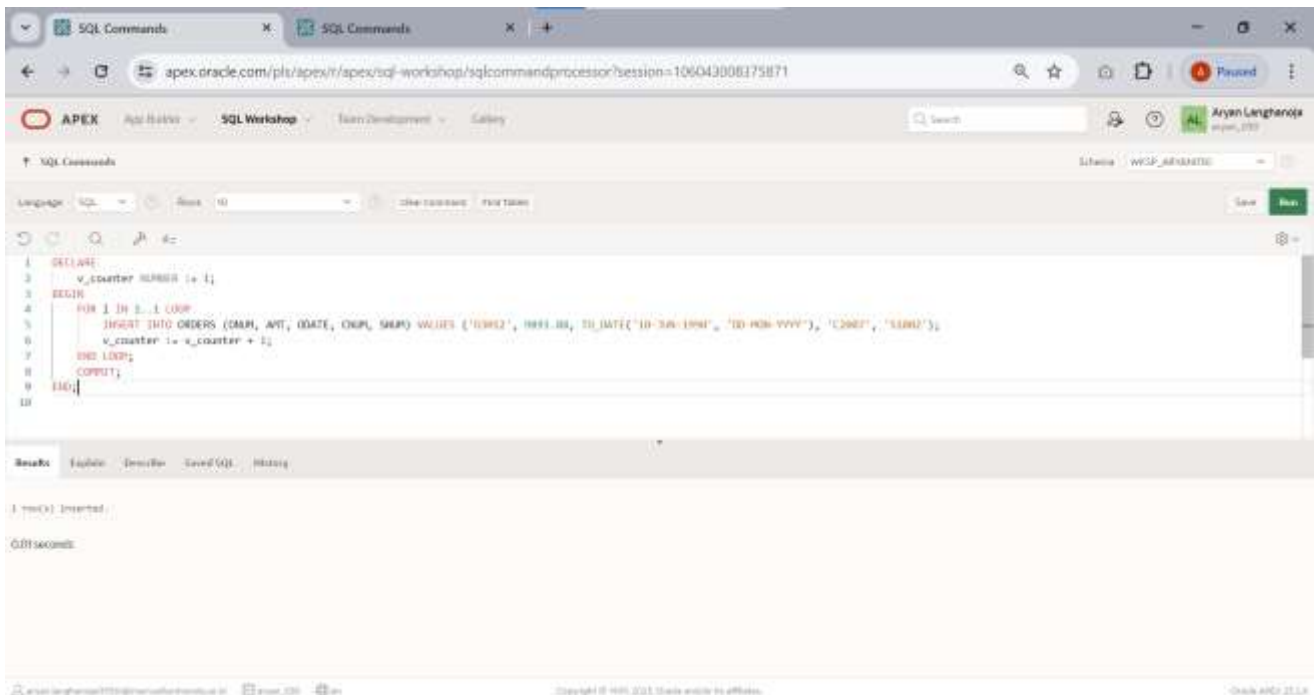
```
DELETE FROM SALESMEN  
WHERE SAHM IN (SELECT DISTINCT SAHM FROM CUSTOMERS WHERE RATING = 100);
```

The results pane shows the message: 0 row(s) deleted. The execution time is 0.01 seconds.

## Practical 11

### Aim: Understand and implement Conditions and looping in PL/SQL

1. Write a PL/SQL program using FOR loop to insert ten rows into a database table.



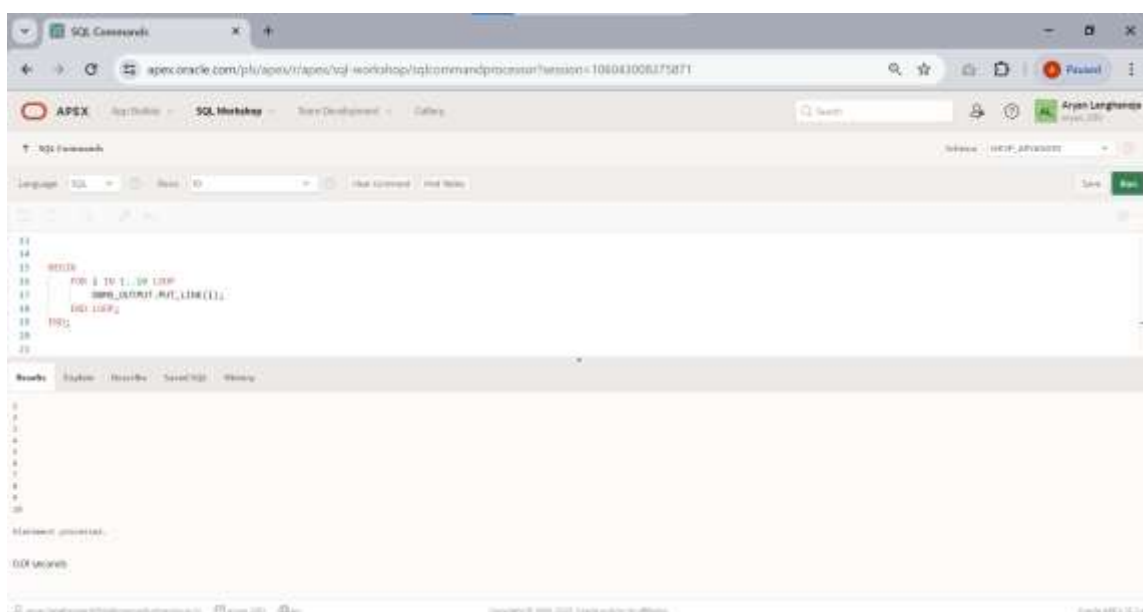
```

1 DECLARE
2   v_counter NUMBER := 1;
3 BEGIN
4   FOR i IN 1..10 LOOP
5     INSERT INTO ORDERS (CUST_ID, AMT, ODATE, CNPL, SHRP, VALUET) VALUES ('00002', 1000.00, TO_DATE('10-JAN-1991', 'DD-MON-YYYY'), 'C20002', 'S20002');
6     v_counter := v_counter + 1;
7   END LOOP;
8   COMMIT;
9 END;

```

Results: 1 row(s) inserted.  
0.01 seconds

2. Write a PL/SQL program to print integers from 1 to 10 by using PL/SQL FOR loop



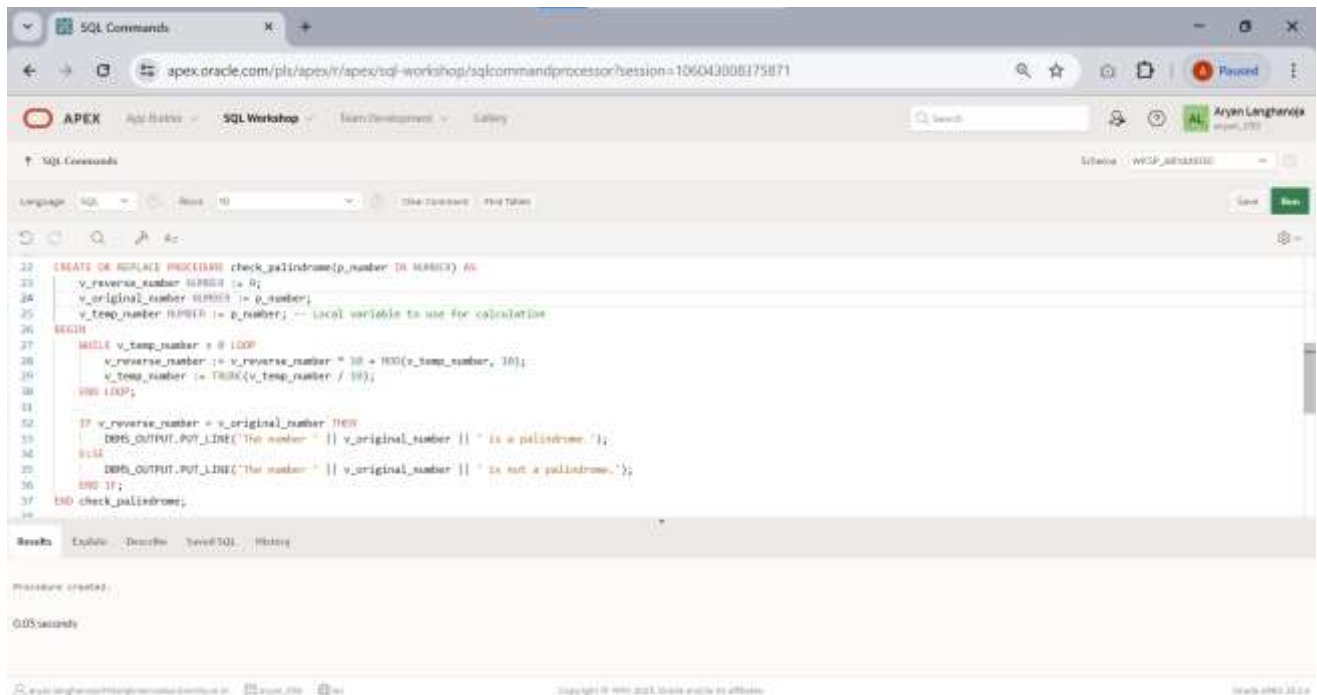
```

13 BEGIN
14   FOR i TO 10 LOOP
15     DBMS_OUTPUT.PUT_LINE(i);
16   END LOOP;
17 END;

```

Results: 10  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Statement processed.  
0.04 seconds

3. Create the procedure for palindrome of given number



```
22 CREATE OR REPLACE PROCEDURE check_palindrome(p_number IN NUMBER) AS
23   v_reverse_number NUMBER := 0;
24   v_original_number NUMBER := p_number;
25   v_temp_number NUMBER := p_number; -- local variable to use for calculation
26 BEGIN
27   WHILE v_temp_number > 0 LOOP
28     v_reverse_number := v_reverse_number * 10 + MOD(v_temp_number, 10);
29     v_temp_number := TRUNC(v_temp_number / 10);
30   END LOOP;
31
32   IF v_reverse_number = v_original_number THEN
33     DBMS_OUTPUT.PUT_LINE('The number ' || v_original_number || ' is a palindrome.');

Results   Execute   Describe   Saved SQL   History



Procedure created:



0.05 seconds



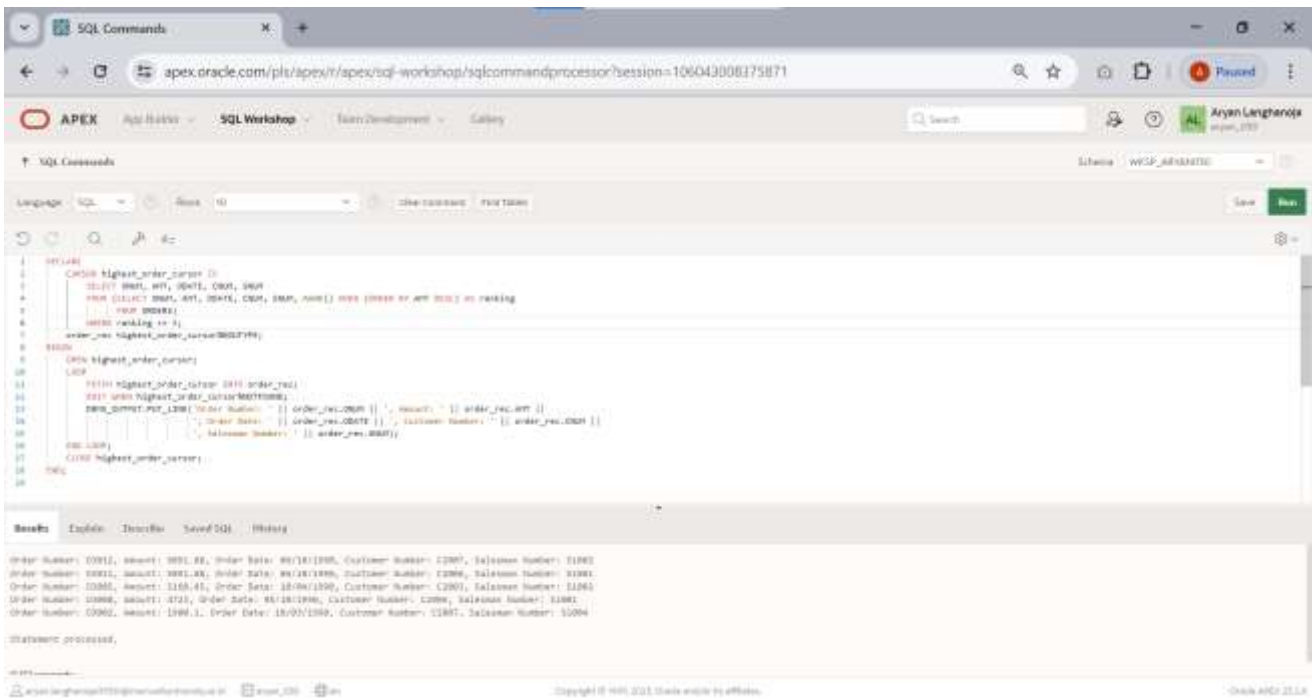
Copyright © 1996 and 1998 Oracle Corporation and/or its affiliates. Oracle and/or its affiliates may have one or more registered trademarks or other intellectual property rights in the United States and/or other countries.


```

## Practical 12

### Aim: Implement a PL/SQL Block.

- Given the table ORDER (ONUM, AMT, ODATE, CNUM, SNUM) write a cursor to select the five highest amount (AMT) order details from the table.



```

1 DECLARE
2   CURSOR highest_order_cursor IS
3     SELECT ONUM, AMT, ODATE, CNUM, SNUM
4     FROM (SELECT ONUM, AMT, ODATE, CNUM, SNUM, ROW() OVER (ORDER BY AMT DESC) AS ranking
5           FROM ORDER)
6     WHERE ranking <= 5;
7   order_rec highest_order_cursor%ROWTYPE;
8 BEGIN
9   OPEN highest_order_cursor;
10  LOOP
11    FETCH highest_order_cursor INTO order_rec;
12    EXIT WHEN highest_order_cursor%NOTFOUND;
13    DBMS_OUTPUT.PUT_LINE('Order Number: ' || order_rec.ONUM || ', Amount: ' || order_rec.AMT || ',
14                          'Order Date: ' || order_rec.ODATE || ', Customer Number: ' || order_rec.CNUM || ',
15                          'Salesman Number: ' || order_rec.SNUM);
16  END LOOP;
17  CLOSE highest_order_cursor;
18 END;
  
```

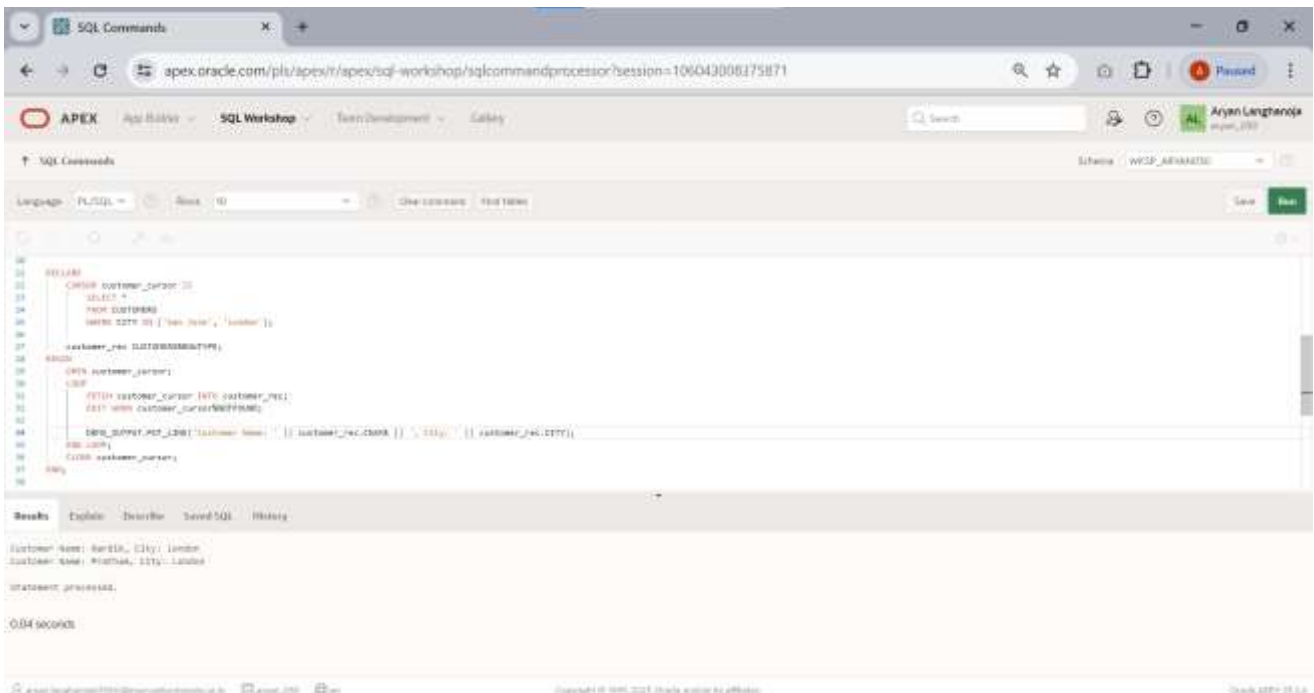
Results: Explain: Describe: Saved SQL: History

```

Order Number: 10012, Amount: 9891.85, Order Date: 08/18/1998, Customer Number: C2000, Salesman Number: S1000
Order Number: 10011, Amount: 9891.85, Order Date: 08/18/1998, Customer Number: C2006, Salesman Number: S1004
Order Number: 10005, Amount: 5169.45, Order Date: 10/06/1999, Customer Number: C2001, Salesman Number: S1003
Order Number: 10006, Amount: 4723, Order Date: 05/18/1998, Customer Number: C2006, Salesman Number: S1001
Order Number: 10002, Amount: 1999.3, Order Date: 10/01/1999, Customer Number: S1001, Salesman Number: S1004
  
```

Statement processed.

- To write a Cursor to display the list of customers who are living in San jose or London.



```

23 DECLARE
24   CURSOR customer_cursor IS
25     SELECT *
26     FROM CUSTOMER
27     WHERE CITY IN ('San Jose', 'London');
28   customer_rec customer_cursor%ROWTYPE;
29 BEGIN
30   OPEN customer_cursor;
31   LOOP
32     FETCH customer_cursor INTO customer_rec;
33     EXIT WHEN customer_cursor%NOTFOUND;
34     DBMS_OUTPUT.PUT_LINE('Customer Name: ' || customer_rec.CNAME || ', City: ' || customer_rec.CITY);
35   END LOOP;
36   CLOSE customer_cursor;
37 END;
  
```

Results: Explain: Describe: Saved SQL: History

```

Customer Name: San Jose, City: London
Customer Name: Phoenix, City: London
  
```

Statement processed.

0.04 seconds

## Practical 13

### Aim Implement a Procedure and Function for given Statement

1. Calculate hra,da, gross and net by using PL/SQL program

Basic	HRA	DA
15000	12%	8%
12000	10%	6%
9000	7%	4%
OTHERS	5%	200/-

The screenshot shows the APEX SQL Workshop interface. The main editor contains the following PL/SQL code:

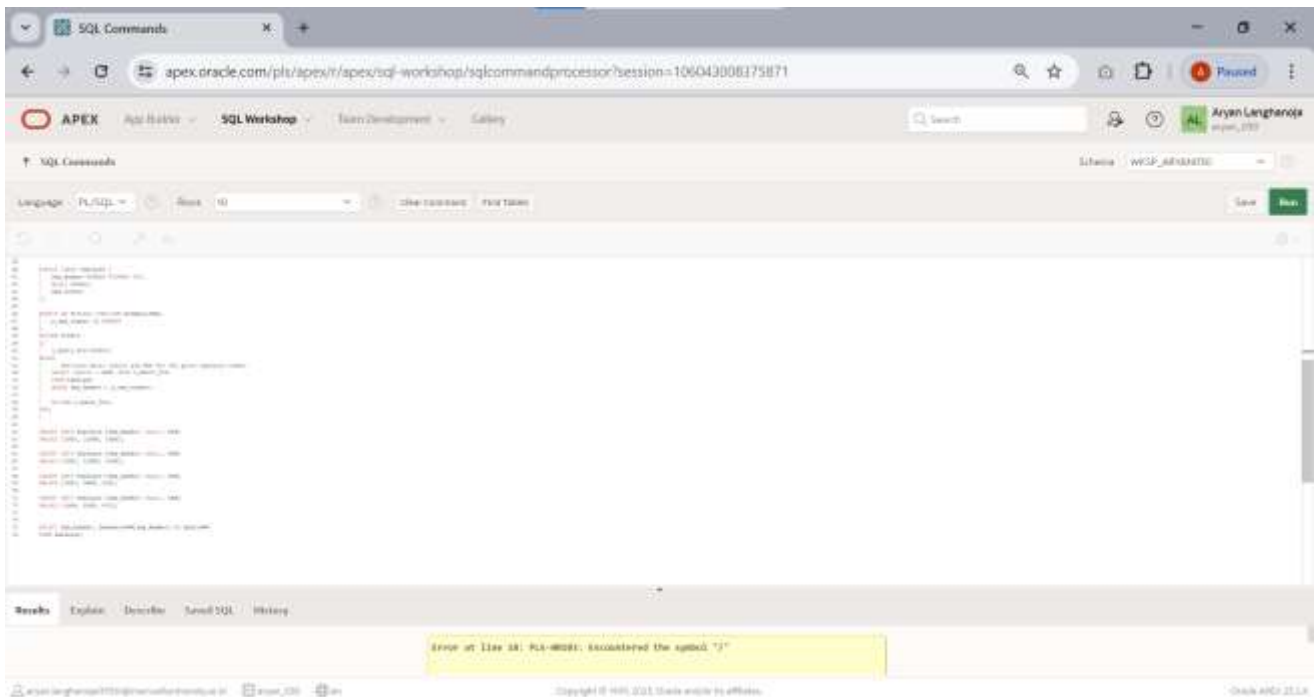
```

1  CREATE OR REPLACE PROCEDURE calculate_hra_da_gross_net (
2    p_basic IN NUMBER,
3    p_hra IN NUMBER,
4    p_da IN NUMBER,
5    p_gross IN NUMBER,
6    p_net IN NUMBER,
7    p_hra_rate IN NUMBER,
8    p_da_rate IN NUMBER,
9    p_gross_rate IN NUMBER,
10   p_net_rate IN NUMBER)
11  IS
12    v_hra NUMBER;
13    v_da NUMBER;
14    v_gross NUMBER;
15    v_net NUMBER;
16  BEGIN
17    v_hra := p_basic * p_hra_rate / 100;
18    v_da := p_basic * p_da_rate / 100;
19    v_gross := p_gross_rate / 100;
20    v_net := p_net_rate / 100;
21  END;
22
23  CREATE OR REPLACE PROCEDURE calculate_hra_da_gross_net (
24    p_basic IN NUMBER,
25    p_hra IN NUMBER,
26    p_da IN NUMBER,
27    p_gross IN NUMBER,
28    p_net IN NUMBER,
29    p_hra_rate IN NUMBER,
30    p_da_rate IN NUMBER,
31    p_gross_rate IN NUMBER,
32    p_net_rate IN NUMBER)
33  IS
34    v_hra NUMBER;
35    v_da NUMBER;
36    v_gross NUMBER;
37    v_net NUMBER;
38  BEGIN
39    v_hra := p_basic * p_hra_rate / 100;
40    v_da := p_basic * p_da_rate / 100;
41    v_gross := p_gross_rate / 100;
42    v_net := p_net_rate / 100;
43  END;
44
45  BEGIN
46    calculate_hra_da_gross_net (15000, 12, 8, 15000, 15000, 12, 8, 15000, 15000);
47    calculate_hra_da_gross_net (12000, 10, 6, 12000, 12000, 10, 6, 12000, 12000);
48    calculate_hra_da_gross_net (9000, 7, 4, 9000, 9000, 7, 4, 9000, 9000);
49    calculate_hra_da_gross_net (200, 5, 200, 200, 200, 5, 200, 200, 200);
50  END;

```

The error message at the bottom of the screen reads: "Error at line 38: PL/SQL: Encountered the symbol '}'".

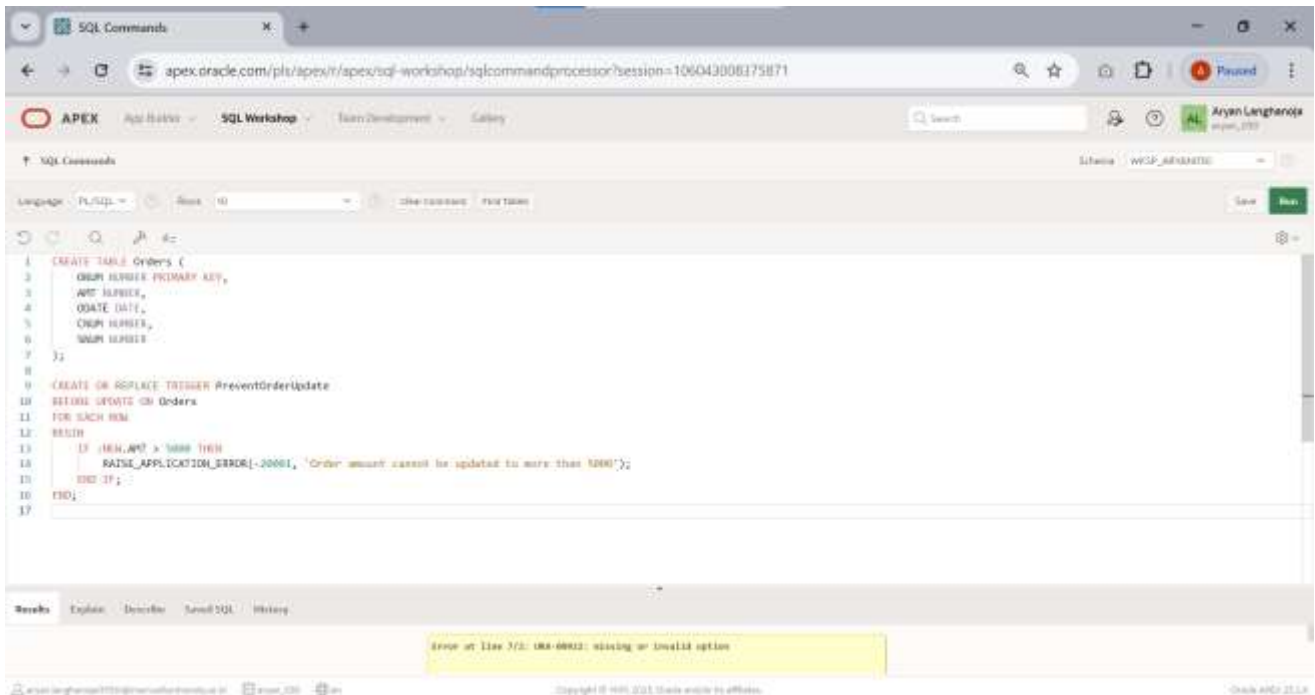
2. Write a function to accept employee number as parameter and return Basic +HRA together as single column



## Practical 14

### Aim: Understand and Implement Triggers.

1. Whenever order amount is updated and its value becomes more than 5000 a trigger has to be raised preventing the operation.



The screenshot shows the APEX SQL Workshop interface. The SQL command window contains the following code:

```
1 CREATE TABLE Orders (
2   ORDER NUMBER PRIMARY KEY,
3   AMT NUMBER,
4   ODATE DATE,
5   CNAME NUMBER,
6   SNAME NUMBER
7 );
8
9 CREATE OR REPLACE TRIGGER PreventOrderUpdate
10 BEFORE UPDATE ON Orders
11 FOR EACH ROW
12 BEGIN
13   IF (NEW.AMT > 5000 THEN
14     RAISE_APPLICATION_ERROR(-20001, 'Order amount cannot be updated to more than 5000');
15   END IF;
16 END;
```

The Results tab at the bottom displays an error message: "Error at Line 3/2: ORA-00002: missing or invalid option".