

Ian Morgan-Graham  
COSC 641 Database Management Systems II  
May 5, 2021

**Blockchain Finance  
Database Final Project  
COSC 641  
CHAPTER 1**

You are employed to work on an existing project. This new financial organization (bank) is based on Bitcoin. The base tables and some data are already provided for you. The ER diagram and structure of tables are in a supplemental document.

Please number your chapters correctly. I grade one chapter at a time.

The data and based tables are under PROJECT2020 account. You can access each table by using PROJECT2020.table\_name. You can copy each table into your Oracle account like:

```
CREATE TABLE branch AS
SELECT      *
FROM        PROJECT2020.branch;
```

Check your data:

```
SELECT *
FROM branch;
```

Check the attributes:

```
DESC branch;
```

Note 1: Frequently it is required to make changes to the base tables (add or delete fields). You can use ALTER TABLE. To add/delete fields to your tables.

The added fields will have value of NULL for now.

Note 2: All of the data must be in your based tables. We will not create FORCED view.

Note 3: For complex queries you can create intermediate views and create view from view.

Note 4: We will use the views and sequences later on in the project. For now, just create them.

Note 4: The data in the tables are test data. By no means they are complete. You welcome to add additional records to the tables.

Please try to keep the names tables as close to the original names as you can. If there is any missing data, feel free to add it to your tables.

List of tables in PROJECT2020 are:

1. BRANCH
2. DEPARTMENT
3. CAR
4. DRIVER (emp drivers' car)
5. CUSTOMER
6. BANK\_EMPLOYEE
7. JOB
8. EMP\_ANNUAL\_DATA
9. BRANCH\_EMPLOYEE (emp work at branch)
10. ATM
11. BRANCH\_ACCESS\_POINTS (branch has access points)
12. BRANCH\_MANAGER (emp manager branch)
13. LOAN
14. LOAN\_PROJECT
15. DEPOSIT\_ACC
16. DEPOSIT\_ACC\_PRODUCT
17. CD\_ACCOUNT
18. CD\_PRODUCT
19. CREDIT\_ACCOUNT
20. CREDIT\_PRODUCT
21. LOAN\_PAYMENT
22. DEPOSIT\_ACCT\_TRANSACTION
23. CREDIT\_ACCT\_TRANSACTION

## **CHAPTER 1**

Create the following views: Please show you **code** (query) and **result** of running your code.

^^^^^^^^^^^^^^^^^^^^^^^^

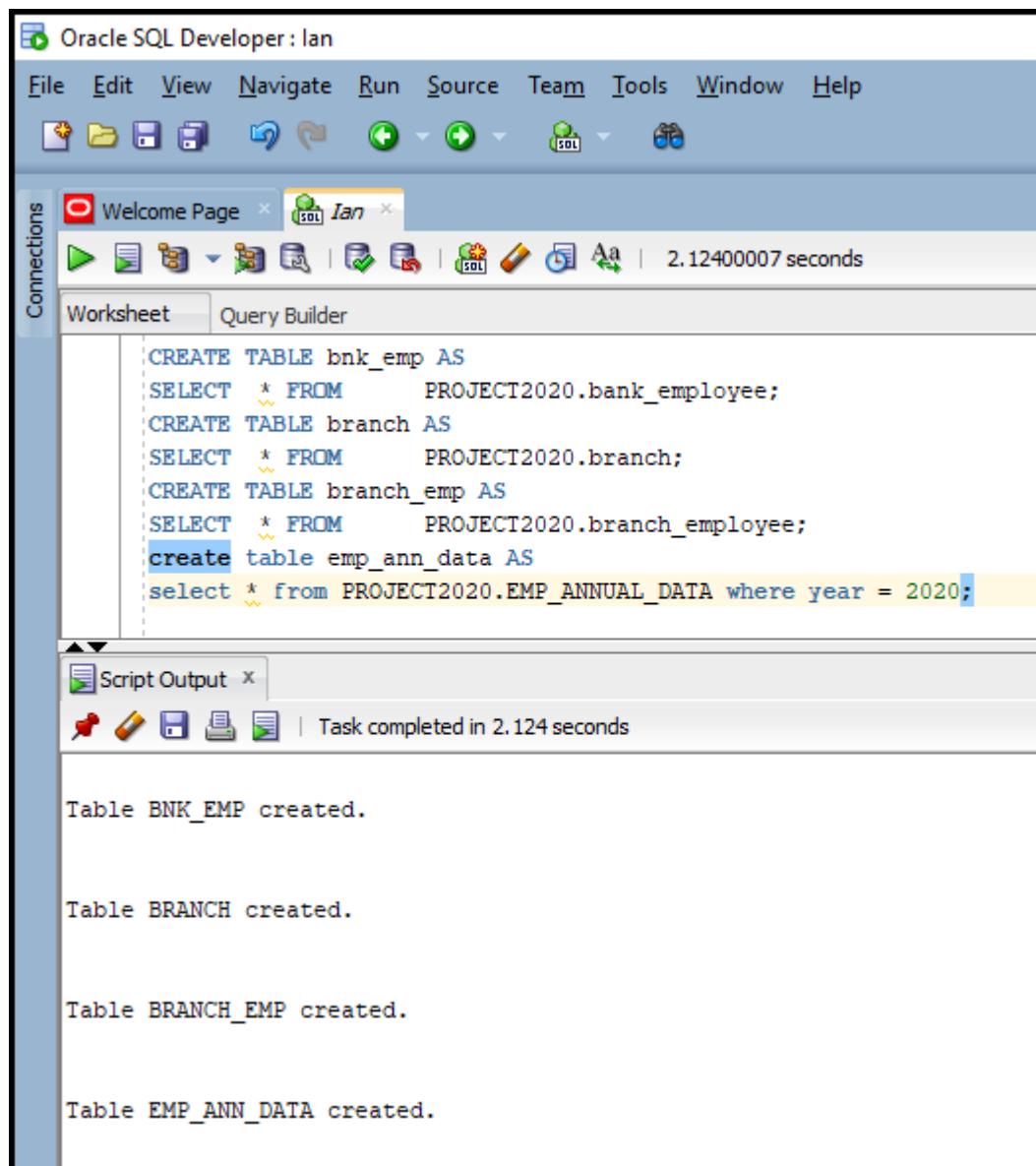
## **CHAPTER 1A Starts here**

^^^^^^^^^^^^^^^^^^^^^^^^

A. Create view employee\_data with the following attributes:

Name of Employee (first, middle, last)
Address
Zip code of Employee Address
SSN
Title
Current Year
Current Yearly Salary
Current Tax Deduction Rate
The date s/he was employed at the Current Branch
Birth Date
Age of Employee
Employee Branch Phone Extension s/he Works at
Branch Phone Number
Branch Name (Employee Works at)
Highest Degree
Highest Degree date

## Copy and rename tables from PROJECT2020 account to my Oracle account



The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left has a connection named 'Ian' selected. The main workspace contains a 'Worksheet' tab with the following SQL script:

```
CREATE TABLE bnk_emp AS
SELECT * FROM PROJECT2020.bank_employee;
CREATE TABLE branch AS
SELECT * FROM PROJECT2020.branch;
CREATE TABLE branch_emp AS
SELECT * FROM PROJECT2020.branch_employee;
create table emp_ann_data AS
select * from PROJECT2020.EMP_ANNUAL_DATA where year = 2020;
```

The 'Script Output' tab below shows the results of the execution:

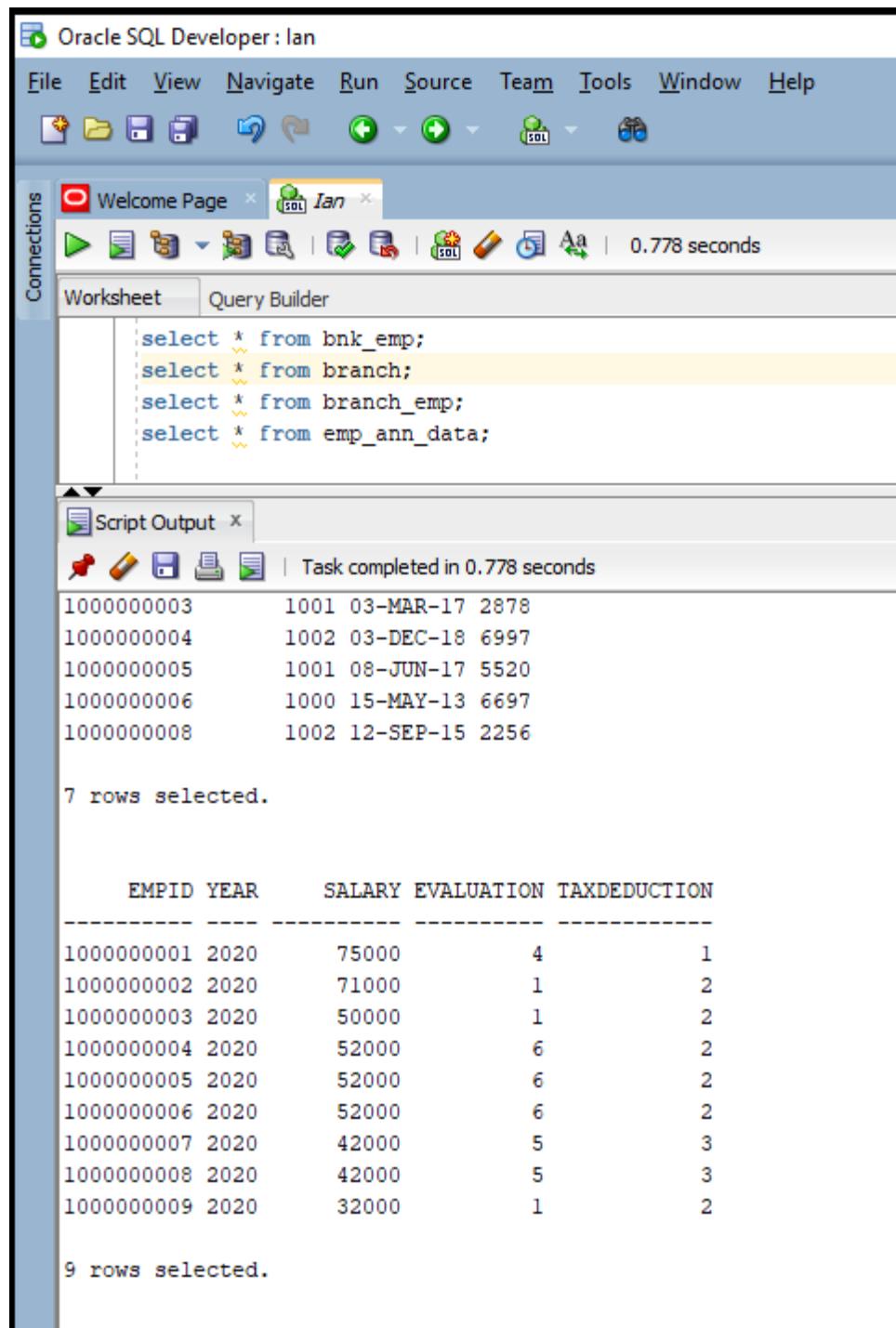
```
Table BNK_EMP created.

Table BRANCH created.

Table BRANCH_EMP created.

Table EMP_ANN_DATA created.
```

**Verify contents of copied and renamed tables from PROJECT2020 account to my Oracle account**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
select * from bnk_emp;
select * from branch;
select * from branch_emp;
select * from emp_ann_data;
```

Script Output

Task completed in 0.778 seconds

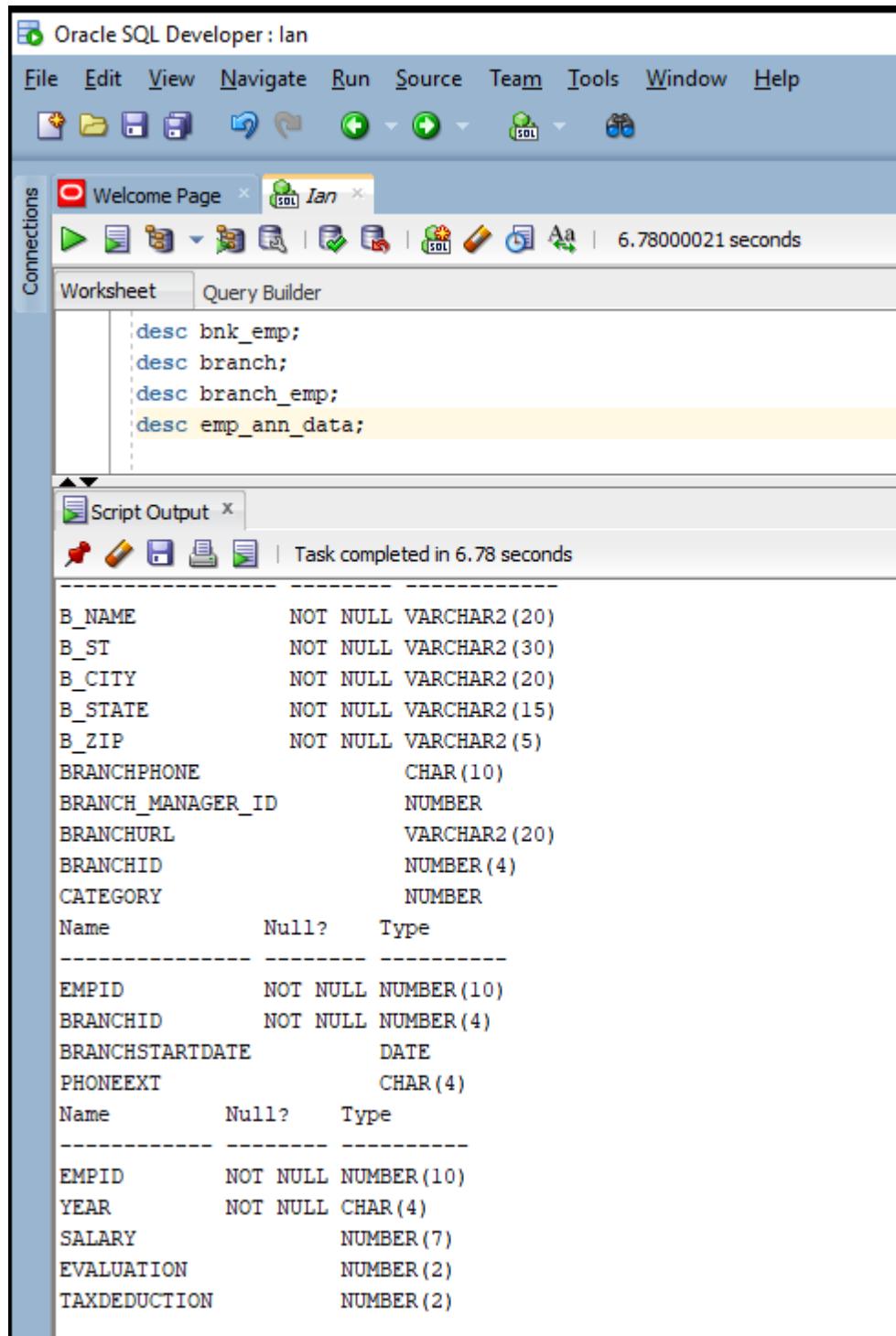
EMPID	YEAR	SALARY	EVALUATION	TAXDEDUCTION
1000000003	1001	03-MAR-17	2878	
1000000004	1002	03-DEC-18	6997	
1000000005	1001	08-JUN-17	5520	
1000000006	1000	15-MAY-13	6697	
1000000008	1002	12-SEP-15	2256	

7 rows selected.

EMPID	YEAR	SALARY	EVALUATION	TAXDEDUCTION
1000000001	2020	75000	4	1
1000000002	2020	71000	1	2
1000000003	2020	50000	1	2
1000000004	2020	52000	6	2
1000000005	2020	52000	6	2
1000000006	2020	52000	6	2
1000000007	2020	42000	5	3
1000000008	2020	42000	5	3
1000000009	2020	32000	1	2

9 rows selected.

## Describe variables of copied and renamed tables from PROJECT2020 account to my Oracle account



The screenshot shows the Oracle SQL Developer interface with the following details:

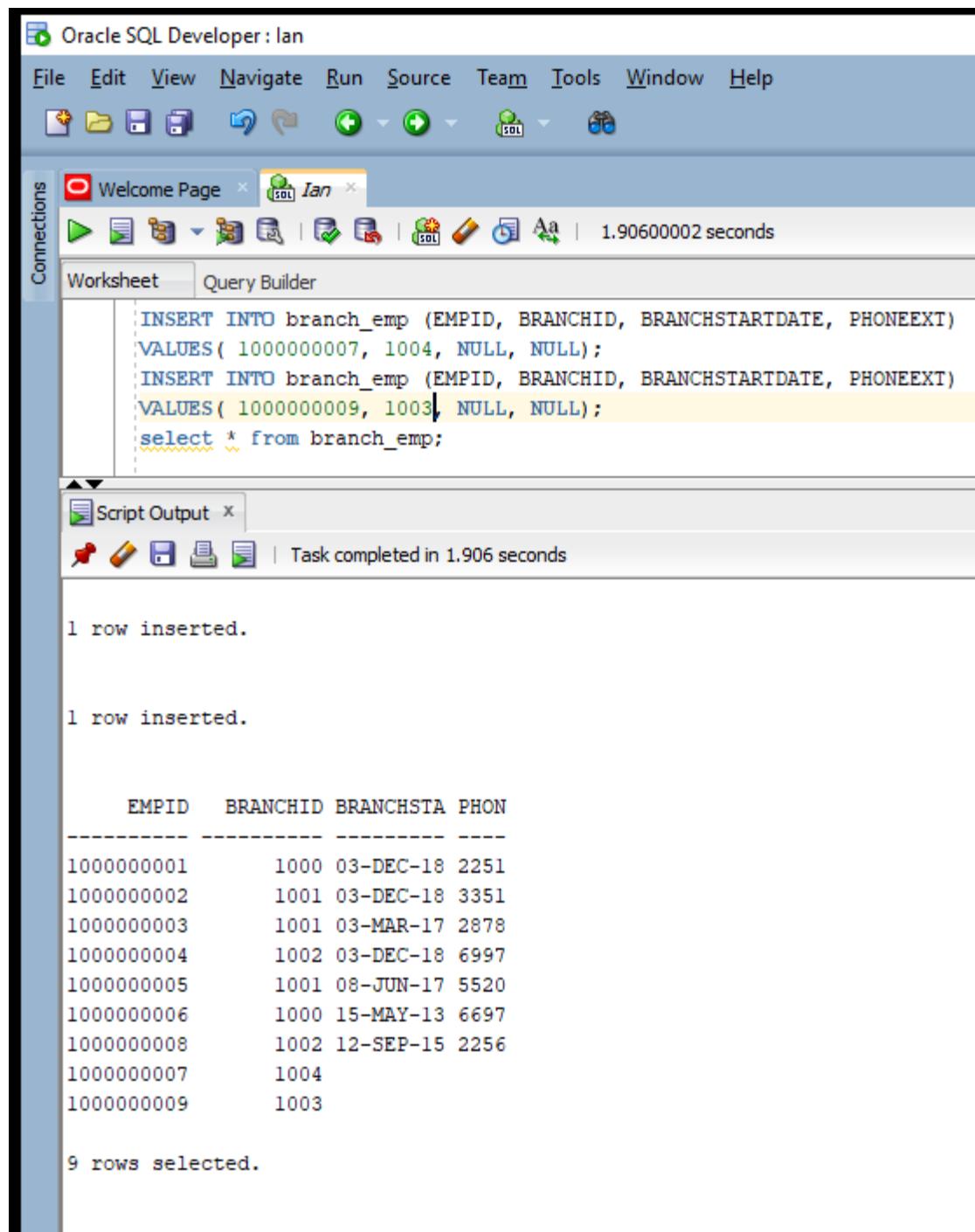
- Worksheet:** The main pane where the following SQL code is entered:

```
desc bnk_emp;
desc branch;
desc branch_emp;
desc emp_ann_data;
```
- Script Output:** The bottom pane showing the results of the executed SQL code:

```
B_NAME          NOT NULL VARCHAR2 (20)
B_ST            NOT NULL VARCHAR2 (30)
B_CITY          NOT NULL VARCHAR2 (20)
B_STATE         NOT NULL VARCHAR2 (15)
B_ZIP           NOT NULL VARCHAR2 (5)
BRANCHPHONE     CHAR (10)
BRANCH_MANAGER_ID NUMBER
BRANCHURL       VARCHAR2 (20)
BRANCHID        NUMBER (4)
CATEGORY        NUMBER
Name            Null?    Type
-----          -----
EMPID           NOT NULL NUMBER(10)
BRANCHID        NOT NULL NUMBER(4)
BRANCHSTARTDATE DATE
PHONEEXT        CHAR (4)
Name            Null?    Type
-----          -----
EMPID           NOT NULL NUMBER(10)
YEAR            NOT NULL CHAR(4)
SALARY          NUMBER(7)
EVALUATION      NUMBER(2)
TAXDEDUCTION    NUMBER(2)
```

The output indicates the task completed in 6.78 seconds.

## Insert two rows into table branch\_emp and verify insertions



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
INSERT INTO branch_emp (EMPID, BRANCHID, BRANCHSTARTDATE, PHONEEXT)
VALUES( 1000000007, 1004, NULL, NULL);
INSERT INTO branch_emp (EMPID, BRANCHID, BRANCHSTARTDATE, PHONEEXT)
VALUES( 1000000009, 1003, NULL, NULL);
select * from branch_emp;
```

Script Output

Task completed in 1.906 seconds

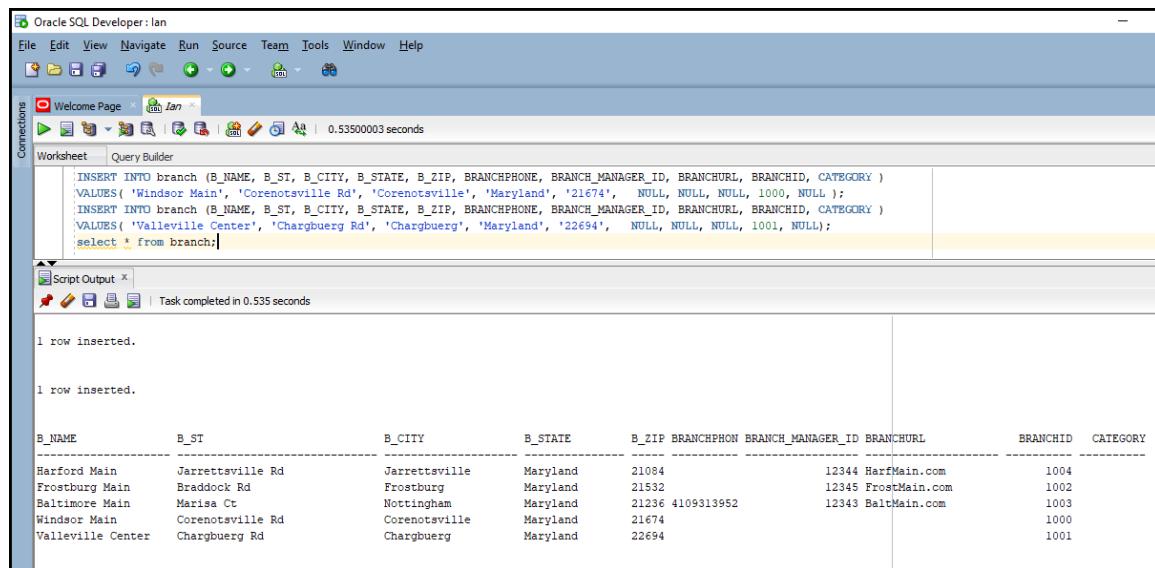
1 row inserted.

1 row inserted.

EMPID	BRANCHID	BRANCHSTA	PHON
1000000001	1000	03-DEC-18	2251
1000000002	1001	03-DEC-18	3351
1000000003	1001	03-MAR-17	2878
1000000004	1002	03-DEC-18	6997
1000000005	1001	08-JUN-17	5520
1000000006	1000	15-MAY-13	6697
1000000008	1002	12-SEP-15	2256
1000000007	1004		
1000000009	1003		

9 rows selected.

## Insert two rows into table branch\_emp and verify insertions



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL script:

```
INSERT INTO branch (B_NAME, B_ST, B_CITY, B_STATE, B_ZIP, BRANCHPHONE, BRANCH_MANAGER_ID, BRANCHURL, BRANCHID, CATEGORY )
VALUES ( 'Windsor Main', 'Corenotsville Rd', 'Corenotsville', 'Maryland', '21674', NULL, NULL, NULL, 1000, NULL );
INSERT INTO branch (B_NAME, B_ST, B_CITY, B_STATE, B_ZIP, BRANCHPHONE, BRANCH_MANAGER_ID, BRANCHURL, BRANCHID, CATEGORY )
VALUES ( 'Valleville Center', 'Chargbuerg Rd', 'Chargbuerg', 'Maryland', '22694', NULL, NULL, NULL, 1001, NULL );
select * from branch;
```

The 'Script Output' tab shows the results of the execution:

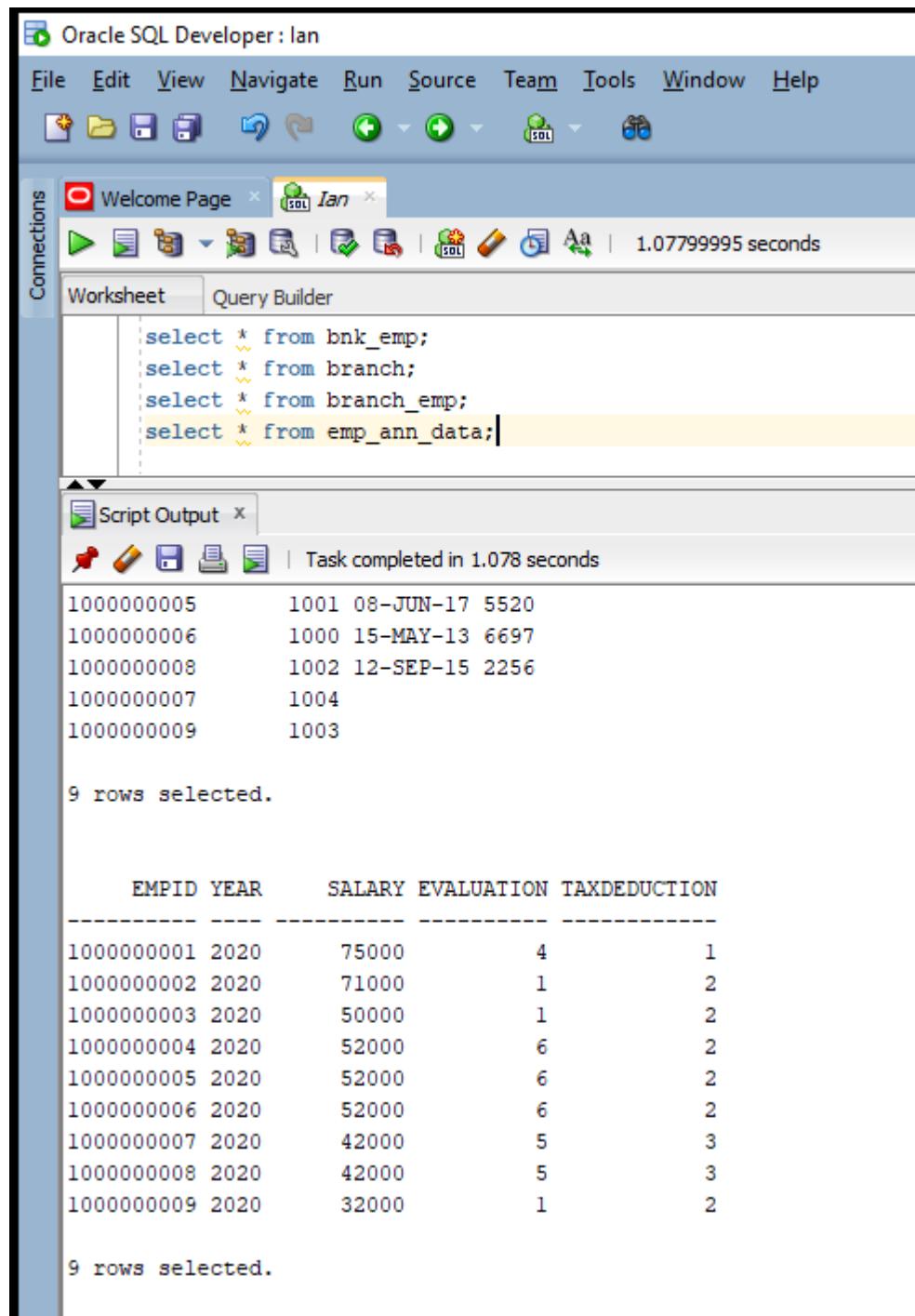
```
1 row inserted.

1 row inserted.
```

A 'Results' tab is visible below, showing the data from the 'branch' table:

B_NAME	B_ST	B_CITY	B_STATE	B_ZIP	BRANCHPHONE	BRANCH_MANAGER_ID	BRANCHURL	BRANCHID	CATEGORY
Harford Main	Jarrettsville Rd	Jarrettsville	Maryland	21084	12344	HarfMain.com		1004	
Frostburg Main	Braddock Rd	Frostburg	Maryland	21532	12345	FrostMain.com		1002	
Baltimore Main	Marissa Ct	Nottingham	Maryland	21236	4109313952	12343	BaltMain.com	1003	
Windsor Main	Corenotsville Rd	Corenotsville	Maryland	21674				1000	
Valleville Center	Chargbuerg Rd	Chargbuerg	Maryland	22694				1001	

## Verify contents of copied and renamed tables from PROJECT2020 account to my Oracle account



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian Worksheet Query Builder

```
select * from bnk_emp;
select * from branch;
select * from branch_emp;
select * from emp_ann_data;
```

Script Output

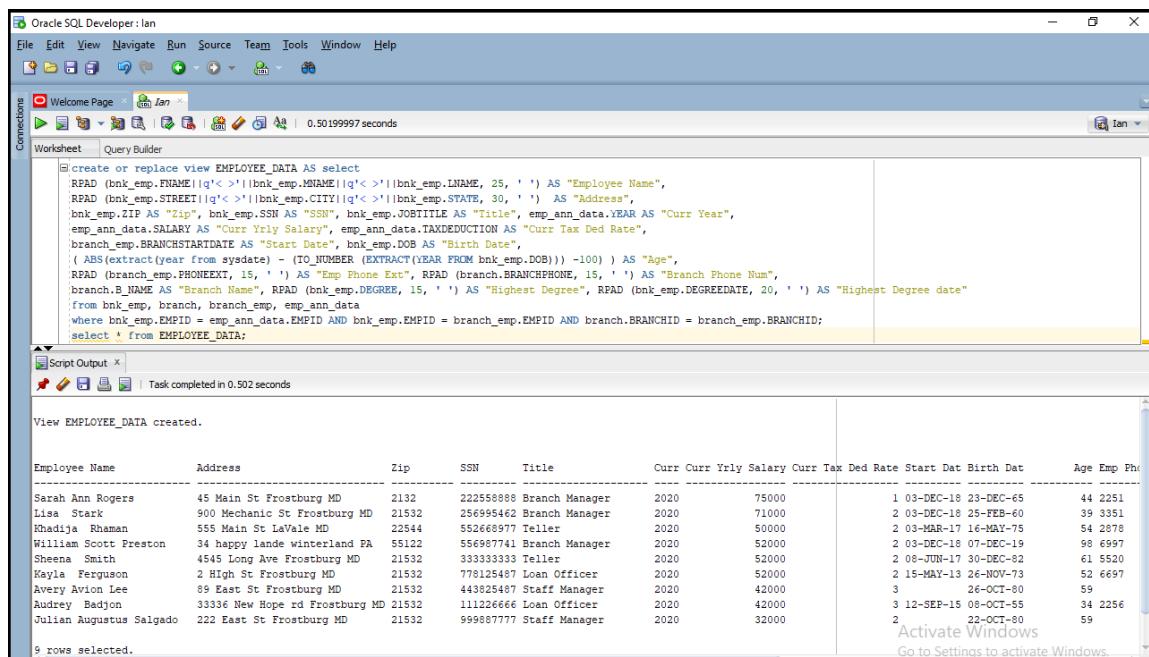
Task completed in 1.078 seconds

EMPID	YEAR	salary	EVALUATION	TAXDEDUCTION
1000000001	2020	75000	4	1
1000000002	2020	71000	1	2
1000000003	2020	50000	1	2
1000000004	2020	52000	6	2
1000000005	2020	52000	6	2
1000000006	2020	52000	6	2
1000000007	2020	42000	5	3
1000000008	2020	42000	5	3
1000000009	2020	32000	1	2

9 rows selected.

**Finally, create view employee\_data with the following attributes using copied tables bnk\_emp, branch, branch\_emp, and emp\_ann\_data:**

Name of Employee (first, middle, last)
Address
Zip code of Employee Address
SSN
Title
Current Year
Current Yearly Salary
Current Tax Deduction Rate
The date s/he was employed at the Current Branch
Birth Date
Age of Employee
Employee Branch Phone Extension s/he Works at
Branch Phone Number
Branch Name (Employee Works at)
Highest Degree
Highest Degree date



```

create or replace view EMPLOYEE_DATA AS select
  RPAD(bnk_emp.FNAME||' '||bnk_emp.MNAME||' '||bnk_emp.LNAME, 25, ' ') AS "Employee Name",
  RPAD(bnk_emp.STREET||' '||bnk_emp.CITY||' '||bnk_emp.STATE, 30, ' ') AS "Address",
  bnk_emp.ZIP AS "Zip", bnk_emp.SSN AS "SSN", bnk_emp.JOBTITLE AS "Title", emp_ann_data.YEAR AS "Curr Year",
  emp_ann_data.SALARY AS "Curr Yrly Salary", emp_ann_data.TAXREDUCTION AS "Curr Tax Ded Rate",
  branch_emp.BRANCHSTARTDATE AS "Start Date", bnk_emp.DOB AS "Birth Date",
  ( ABS(extract(year from sysdate) - (TO_NUMBER (EXTRACT(YEAR FROM bnk_emp.DOB)) -100) ) AS "Age",
  RPAD(branch_emp.PHONEEXT, 15, ' ') AS "Emp Phone Ext", RPAD(branch.BRANCHPHONE, 15, ' ') AS "Branch Phone Num",
  branch.B_NAME AS "Branch Name", RPAD(bnk_emp.DEGREE, 15, ' ') AS "Highest Degree", RPAD(bnk_emp.DEGREEDATE, 20, ' ') AS "Highest Degree date"
  from bnk_emp, branch, branch_emp, emp_ann_data
  where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch_emp.EMPID AND branch.BRANCHID = branch_emp.BRANCHID;
  select * from EMPLOYEE_DATA;

```

View EMPLOYEE\_DATA created.

Employee Name	Address	Zip	SSN	Title	Curr Yrly Salary	Curr Tax Ded Rate	Start Dat	Birth Dat	Age	Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222550888	Branch Manager	2020	75000	1 03-DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256995462	Branch Manager	2020	71000	2 03-DEC-18	25-FEB-60	39	3351
Rhadja Rhaman	555 Main St LaVale MD	22544	552669977	Teller	2020	50000	2 03-MAR-17	16-MAY-75	54	2878
William Scott Preston	34 Happy lands winterland PA	55122	556997741	Branch Manager	2020	52000	2 03-DEC-18	07-DEC-19	98	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000	2 08-JUN-17	30-DEC-82	61	5520
Kayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000	2 15-MAY-13	26-NOV-73	52	6697
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020	42000	3	26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020	42000	3 12-SEP-15	08-OCT-55	34	2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020	32000	2	22-OCT-80	59	

9 rows selected.

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 0.5019997 seconds Ian

Worksheet Query Builder

```
create or replace view EMPLOYEE_DATA AS select
  READ (bnk_emp.FNAME||' '||bnk_emp.MNAME||' '||bnk_emp.LNAME, 25, ' ') AS "Employee Name",
  READ (bnk_emp.STREET||' '||bnk_emp.CITY||' '||bnk_emp.STATE, 30, ' ') AS "Address",
  bnk_emp.ZIP AS "Zip", bnk_emp.SSN AS "SSN", bnk_emp.JOBTITLE AS "Title", emp_ann_data.YEAR AS "Curr Year",
  emp_ann_data.SALARY AS "Curr Ifly Salary", emp_ann_data.TAXREDUCTION AS "Curr Tax Ded Rate",
  branch.emp.BRANCHSTARDATE AS "Start Date", bnk_emp.DOB AS "Birth Date",
  (ABS(extract(year from systdate) - (TO_NUMBER (EXTRACT(YEAR FROM bnk_emp.DOB)) -100)) ) AS "Age",
  READ (branch_emp.PHONEXT, 15, ' ') AS "Emp Phone Ext", READ (branch.BRANCHPHONE, 15, ' ') AS "Branch Phone Num",
  branch.emp AS "Branch Name", READ (bnk_emp.DEGREE, 15, ' ') AS "Highest Degree", READ (bnk_emp.DEGREEDATE, 20, ' ') AS "Highest Degree date"
from bnk_emp, branch, branch.emp, emp_ann_data
where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch.emp.EMPID AND branch.BRANCHID = branch.emp.BRANCHID;
select * from EMPLOYEE_DATA;
```

Script Output x Task completed in 0.502 seconds

	Curr	Curr	Yrly	Salary	Curr	Tax	Ded	Rate	Start	Dat	Birth	Dat	Age	Emp	Phone	Ext	Branch	Phone	Nu	Branch	Name	Highest	Degree	Highest	Degree	date
ch Manager	2020		75000		1	03-DEC-18	23-DEC-65		44	2251							Windsor Main		Doc		12-MAY-93					
ch Manager	2020		71000		2	03-DEC-18	25-FEB-60		39	3351							Valleeville Center		Mast		05-JUN-87					
er	2020		50000		2	03-MAR-17	16-MAY-75		54	2878							Valleeville Center		AA		12-DEC-95					
ch Manager	2020		52000		2	03-DEC-18	07-DEC-19		98	6997							Frostburg Main		Mast		12-JUN-82					
kr	2020		52000		2	08-JUN-17	30-DEC-82		61	5520							Valleeville Center		Bach		04-JUN-06					
Officer	2020		52000		2	15-MAY-13	26-NOV-73		52	6697							Windsor Main		Mast		04-JUN-06					
Officer	2020		42000		3		26-OCT-80		59							Harford Main		Mast		04-JUN-06						
Officer	2020		42000		3	12-SEP-15	08-OCT-55		34	2256						Frostburg Main		Doc		04-JUN-06						
Officer	2020		32000		2		22-OCT-80		59							Baltimore Main		Mast		06-JUN-10						

^^^^^^^^^^^^^^^^^^^^^^^^

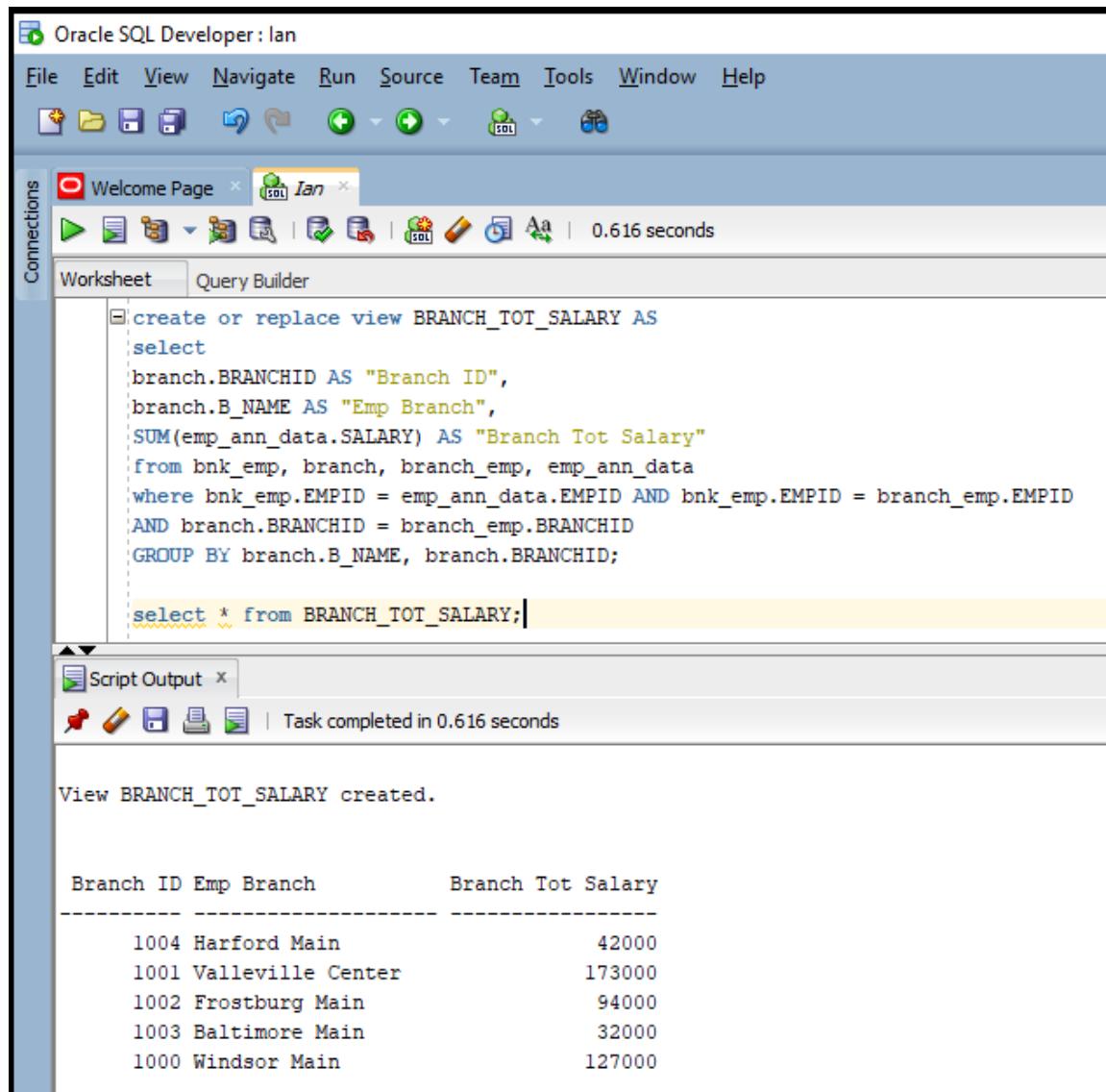
## CHAPTER 1B Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

B. Create view employee\_salary with the following attributes:

Name
Current Year
SSN
Current Salary
Branch Employee Works at
Total Cost of Employee Salaries at the branch s/he works
Highest salary at his/her branch
Average salary at his/her branch

**Create view branch\_tot\_salary**



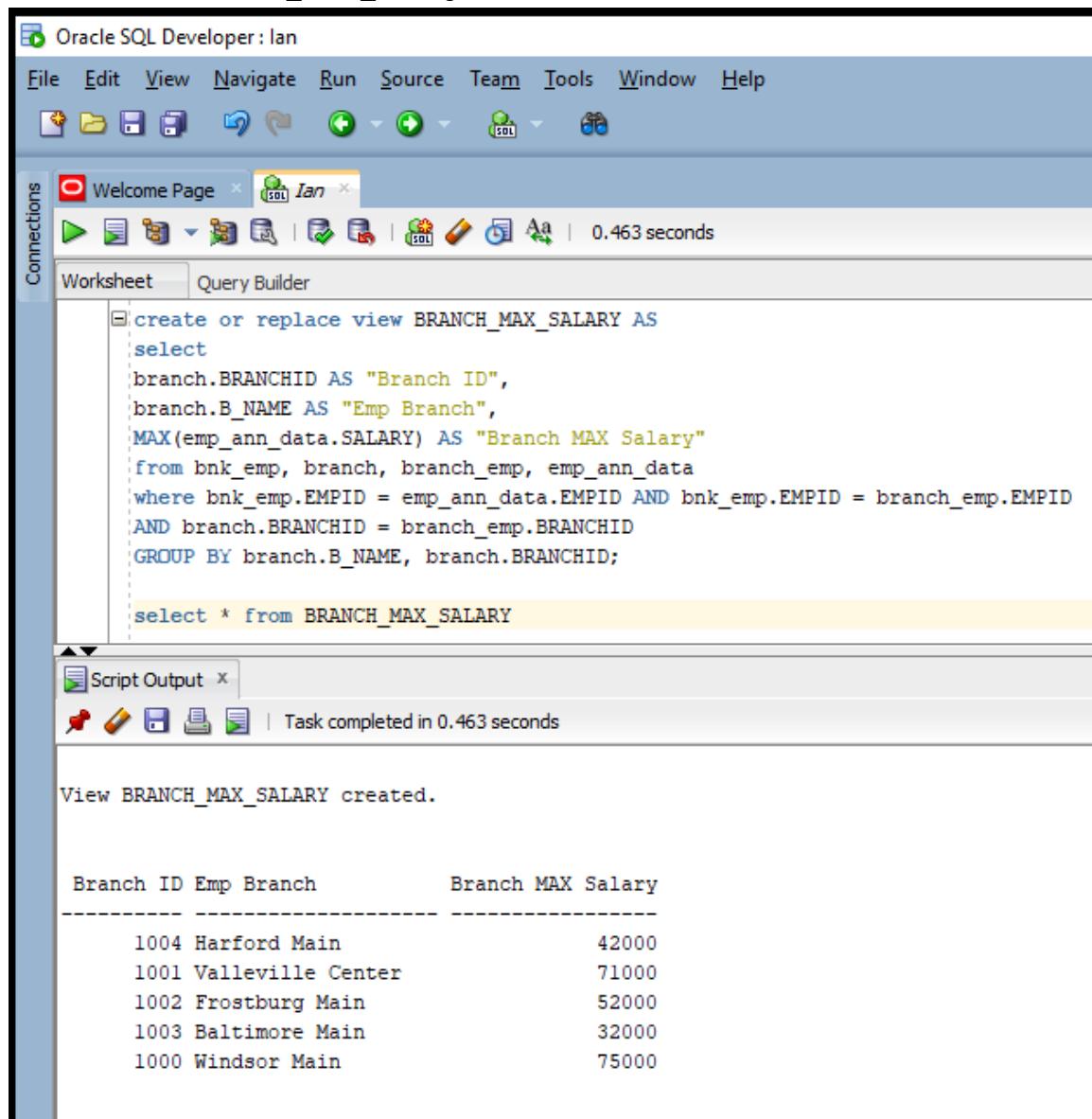
```
create or replace view BRANCH_TOT_SALARY AS
select
branch.BRANCHID AS "Branch ID",
branch.B_NAME AS "Emp Branch",
SUM(emp_ann_data.SALARY) AS "Branch Tot Salary"
from bnk_emp, branch, branch_emp, emp_ann_data
where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch_emp.EMPID
AND branch.BRANCHID = branch_emp.BRANCHID
GROUP BY branch.B_NAME, branch.BRANCHID;

select * from BRANCH_TOT_SALARY;
```

View BRANCH\_TOT\_SALARY created.

Branch ID	Emp Branch	Branch Tot Salary
1004	Harford Main	42000
1001	Valleville Center	173000
1002	Frostburg Main	94000
1003	Baltimore Main	32000
1000	Windsor Main	127000

## Create view branch\_max\_salary



The screenshot shows the Oracle SQL Developer interface with the following details:

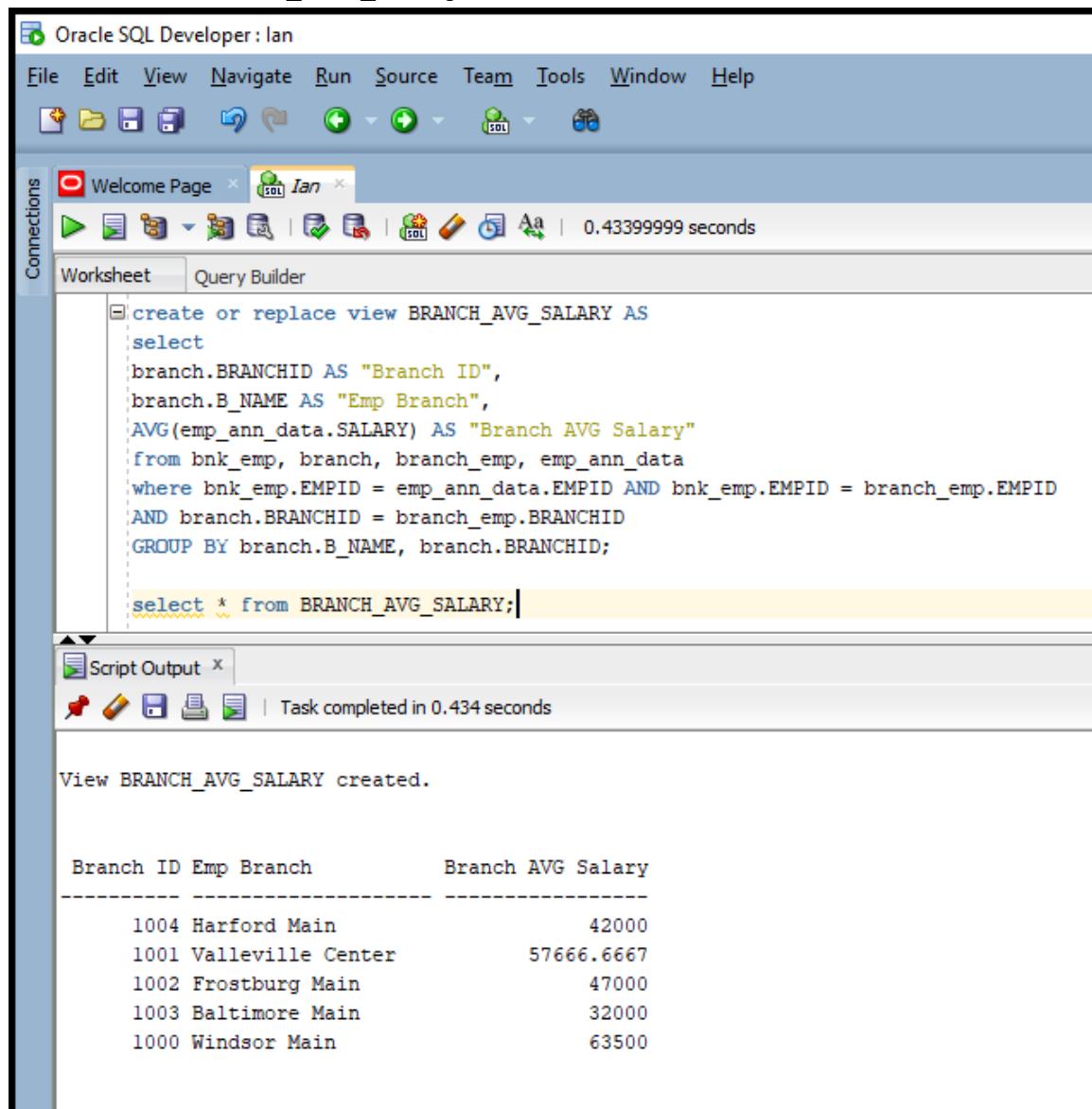
- File Menu:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Toolbar:** Includes icons for New, Open, Save, Undo, Redo, Run, and SQL.
- Connections:** Shows a connection to "Welcome Page" and "Ian".
- Worksheet Tab:** Active tab, showing the SQL code for creating the view.
- Script Output Tab:** Shows the result of the query execution.
- Code in Worksheet:**

```
create or replace view BRANCH_MAX_SALARY AS
select
branch.BRANCHID AS "Branch ID",
branch.B_NAME AS "Emp Branch",
MAX(emp_ann_data.SALARY) AS "Branch MAX Salary"
from bnk_emp, branch, branch_emp, emp_ann_data
where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch_emp.EMPID
AND branch.BRANCHID = branch_emp.BRANCHID
GROUP BY branch.B_NAME, branch.BRANCHID;

select * from BRANCH_MAX_SALARY
```
- Script Output:** Task completed in 0.463 seconds.
- Output:** View BRANCH\_MAX\_SALARY created.
- Table Output:**

Branch ID	Emp Branch	Branch MAX Salary
1004	Harford Main	42000
1001	Valleville Center	71000
1002	Frostburg Main	52000
1003	Baltimore Main	32000
1000	Windsor Main	75000

## Create view branch\_max\_salary



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the SQL code for creating a view:

```
create or replace view BRANCH_AVG_SALARY AS
select
branch.BRANCHID AS "Branch ID",
branch.B_NAME AS "Emp Branch",
AVG(emp_ann_data.SALARY) AS "Branch AVG Salary"
from bnk_emp, branch, branch_emp, emp_ann_data
where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch_emp.EMPID
AND branch.BRANCHID = branch_emp.BRANCHID
GROUP BY branch.B_NAME, branch.BRANCHID;

select * from BRANCH_AVG_SALARY;
```

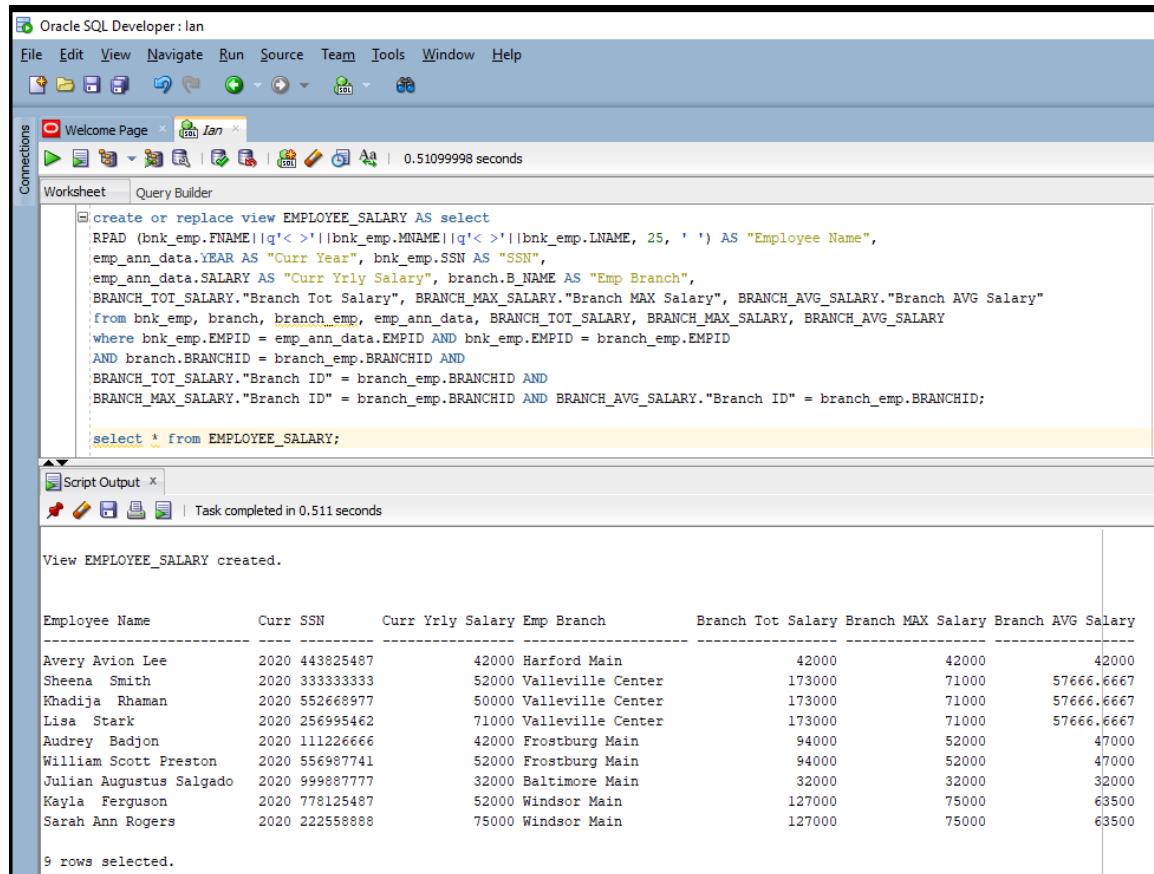
The 'Script Output' tab shows the result of the execution:

```
View BRANCH_AVG_SALARY created.
```

Branch ID	Emp Branch	Branch AVG Salary
1004	Harford Main	42000
1001	Valleville Center	57666.6667
1002	Frostburg Main	47000
1003	Baltimore Main	32000
1000	Windsor Main	63500

**Finally, create view employee\_salary with the following attributes above created views and tables:**

Name
Current Year
SSN
Current Salary
Branch Employee Works at
Total Cost of Employee Salaries at the branch s/he works
Highest salary at his/her branch
Average salary at his/her branch



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL script for creating the EMPLOYEE\_SALARY view. The script joins four tables: bnk\_emp, branch, branch\_emp, and emp\_ann\_data, using various aliases and conditions to calculate branch totals, maximum salaries, and average salaries. Below the script, the 'Script Output' tab shows the message 'Task completed in 0.511 seconds' and the confirmation 'View EMPLOYEE\_SALARY created.' A data grid then displays the results of the query, showing nine rows of employee information with columns for Employee Name, Curr SSN, Curr Yrly Salary, Emp Branch, Branch Tot Salary, Branch MAX Salary, and Branch AVG Salary.

```

create or replace view EMPLOYEE_SALARY AS select
  RPAD (bnk_emp.FNAME||q'>||bnk_emp.MNAME||q'>||bnk_emp.LNAME, 25, ' ') AS "Employee Name",
  emp_ann_data.IYEAR AS "Curr Year", bnk_emp.SSN AS "SSN",
  emp_ann_data.SALARY AS "Curr Yrly Salary", branch.B_NAME AS "Emp Branch",
  BRANCH_TOT_SALARY."Branch Tot Salary", BRANCH_MAX_SALARY."Branch MAX Salary", BRANCH_AVG_SALARY."Branch AVG Salary"
  from bnk_emp, branch, branch_emp, emp_ann_data, BRANCH_TOT_SALARY, BRANCH_MAX_SALARY, BRANCH_AVG_SALARY
  where bnk_emp.EMPID = emp_ann_data.EMPID AND bnk_emp.EMPID = branch_emp.EMPID
  AND branch.BRANCHID = branch_emp.BRANCHID AND
  BRANCH_TOT_SALARY."Branch ID" = branch_emp.BRANCHID AND
  BRANCH_MAX_SALARY."Branch ID" = branch_emp.BRANCHID AND BRANCH_AVG_SALARY."Branch ID" = branch_emp.BRANCHID;

  select * from EMPLOYEE_SALARY;

```

Employee Name	Curr SSN	Curr Yrly Salary	Emp Branch	Branch Tot Salary	Branch MAX Salary	Branch AVG Salary
Avery Avion Lee	2020 443825487	42000	Harford Main	42000	42000	42000
Sheena Smith	2020 333333333	52000	Valleville Center	173000	71000	57666.6667
Khadija Rhaman	2020 552668977	50000	Valleville Center	173000	71000	57666.6667
Lisa Stark	2020 256995462	71000	Valleville Center	173000	71000	57666.6667
Audrey Badjon	2020 111226666	42000	Frostburg Main	94000	52000	47000
William Scott Preston	2020 556987741	52000	Frostburg Main	94000	52000	47000
Julian Augustus Salgado	2020 999887777	32000	Baltimore Main	32000	32000	32000
Kayla Ferguson	2020 778125487	52000	Windsor Main	127000	75000	63500
Sarah Ann Rogers	2020 222558888	75000	Windsor Main	127000	75000	63500

9 rows selected.

^^^^^^^^^^^^^^^^^^^^^^^^

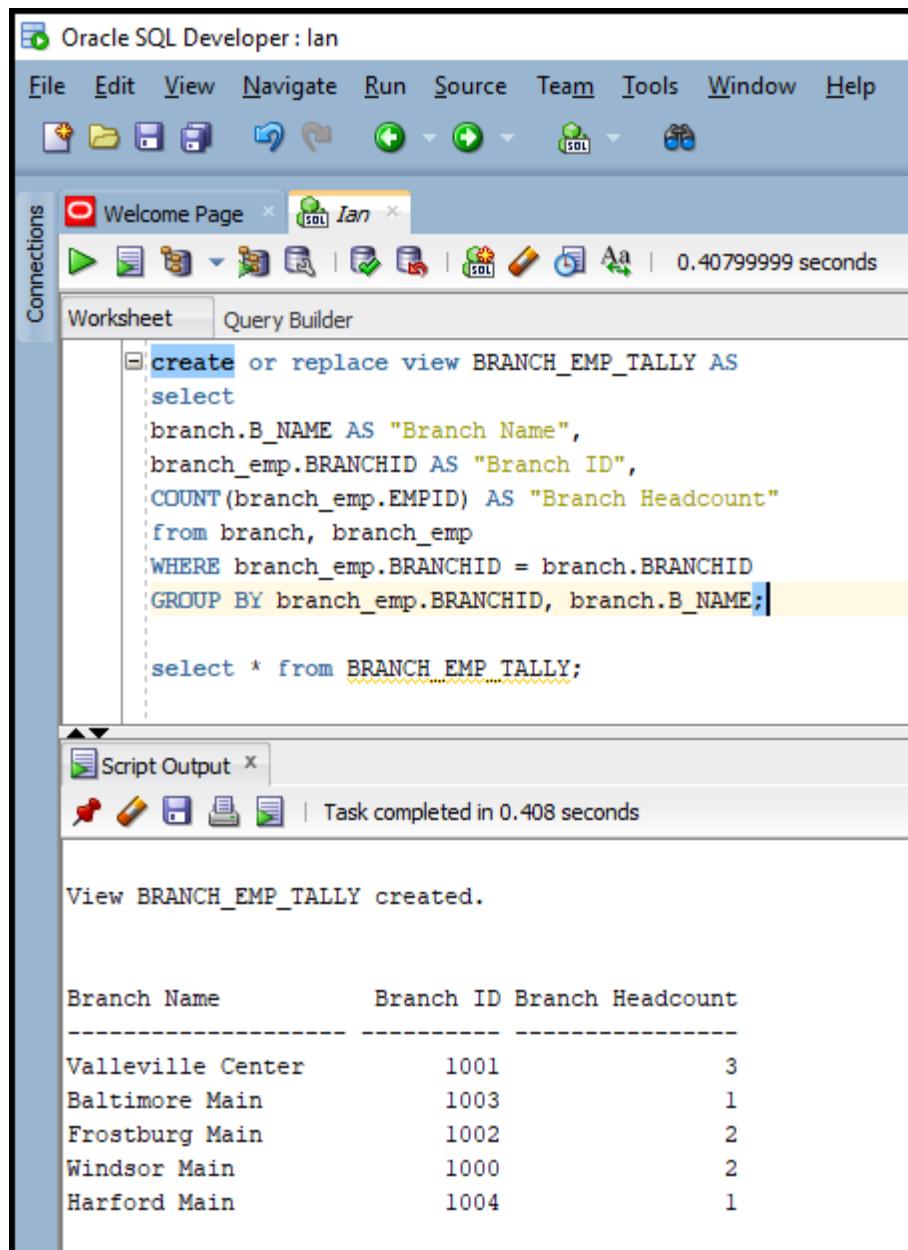
## **CHAPTER 1C Starts here**

^^^^^^^^^^^^^^^^^^^^^^^^

C. Create view branch\_data with the following attributes:

Branch ID
Branch Name
Address
Phone Number
Fax Number <input type="text"/> to be inputted later
Number of Employee at this Branch
Category
Manager Name
Total Transactions Done at this Branch for year 2020

## Create needed views and tables:



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following SQL code:

```
create or replace view BRANCH_EMP_TALLY AS
select
branch.B_NAME AS "Branch Name",
branch_emp.BRANCHID AS "Branch ID",
COUNT(branch_emp.EMPID) AS "Branch Headcount"
from branch, branch_emp
WHERE branch_emp.BRANCHID = branch.BRANCHID
GROUP BY branch_emp.BRANCHID, branch.B_NAME;

select * from BRANCH_EMP_TALLY;
```

The code is highlighted with syntax coloring. The 'Script Output' tab below shows the results of the execution:

```
View BRANCH_EMP_TALLY created.

Branch Name      Branch ID Branch Headcount
-----
Valleeville Center      1001      3
Baltimore Main          1003      1
Frostburg Main          1002      2
Windsor Main             1000      2
Harford Main             1004      1
```

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
create or replace view BRANCH_LEADERS AS
select
EMPLOYEE_DATA."Branch Name" AS "Branch Supervised",
EMPLOYEE_DATA."Employee Name" AS "Leader Name",
EMPLOYEE_DATA."Title" AS "Leader Title"
from EMPLOYEE_DATA
WHERE EMPLOYEE_DATA."Title" = 'Branch Manager' OR EMPLOYEE_DATA."Title" = 'Staff Manager';
```

Script Output

Task completed in 0.317 seconds

View BRANCH\_LEADERS created.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
create table depo_trans AS
SELECT * FROM PROJECT2020.DEPOSIT_ACCT_TRANSACTION;

create table cred_trans AS
SELECT * FROM PROJECT2020.CREDIT_ACCT_TRANSACTION;
```

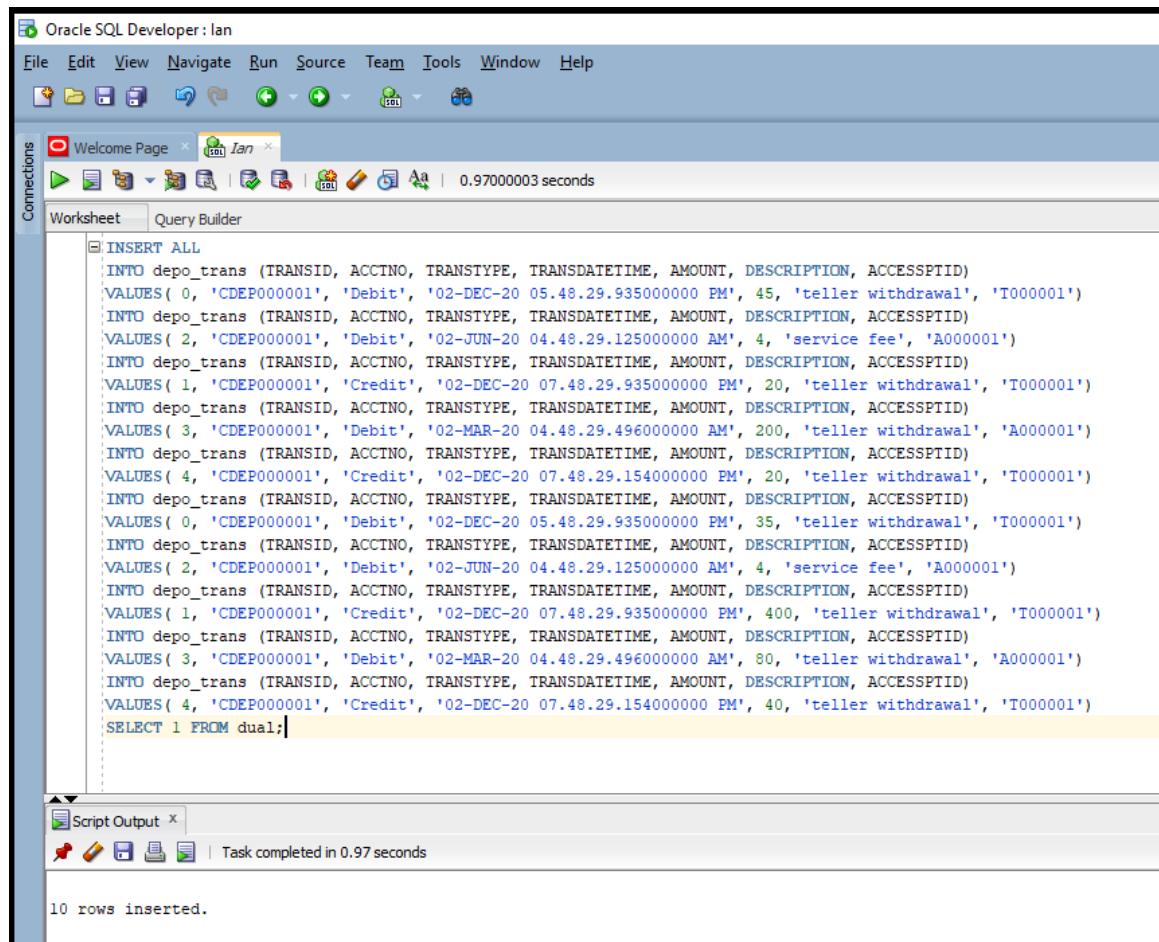
Script Output

Task completed in 0.716 seconds

Table DEPO\_TRANS created.

Table CRED\_TRANS created.

## Insert entries into table depo\_trans:



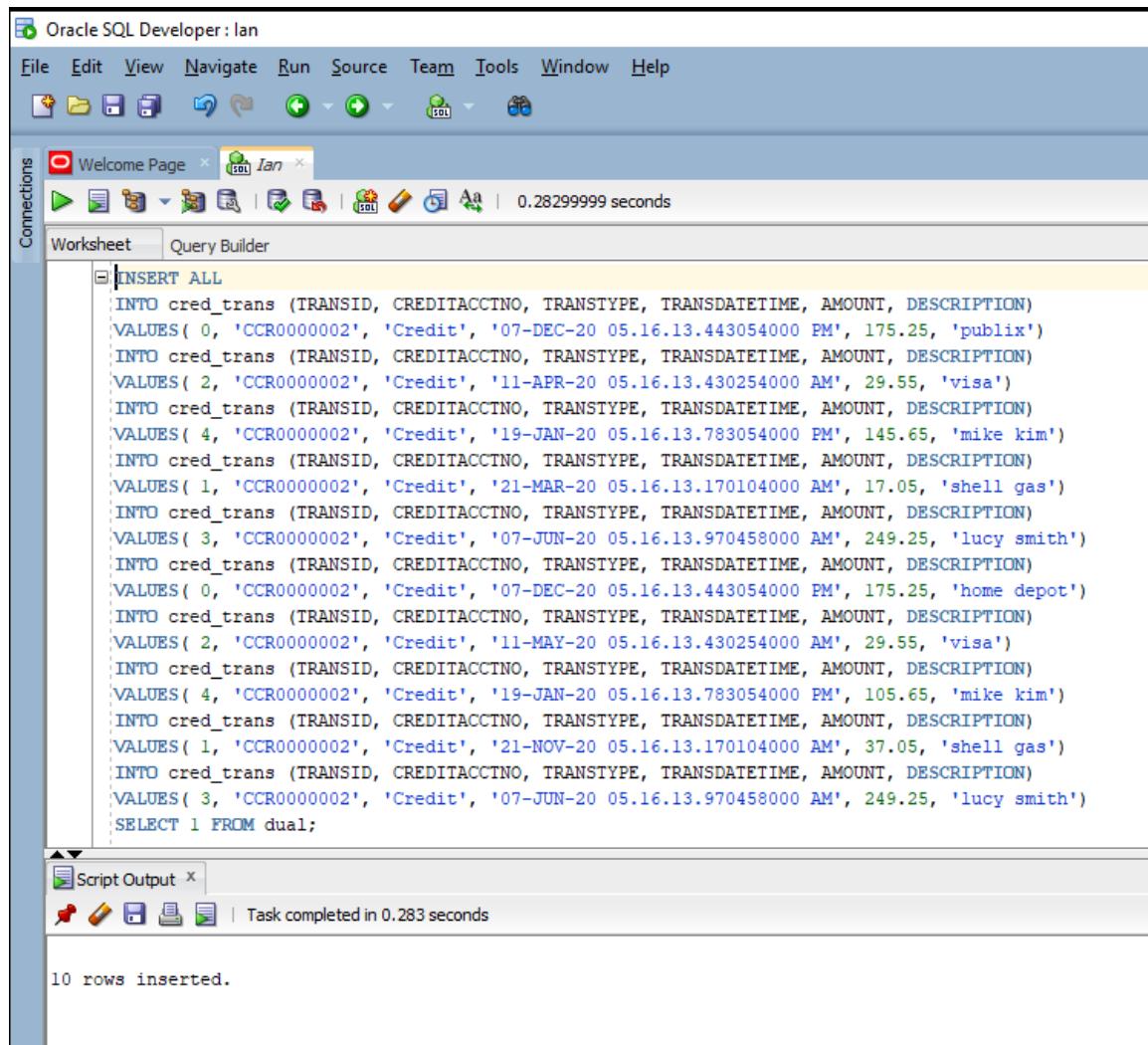
The screenshot shows the Oracle SQL Developer interface with a query script in the Worksheet tab. The script inserts 10 rows into the depo\_trans table. The rows represent various transactions (Debit and Credit) with specific dates, amounts, and descriptions. The script uses the 'ALL' keyword and includes a final 'SELECT 1 FROM dual;' statement. The 'Script Output' tab at the bottom shows a message indicating the task completed in 0.97 seconds.

```
ALL
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 0, 'CDEP000001', 'Debit', '02-DEC-20 05.48.29.935000000 PM', 45, 'teller withdrawal', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 2, 'CDEP000001', 'Debit', '02-JUN-20 04.48.29.125000000 AM', 4, 'service fee', 'A000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 20, 'teller withdrawal', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 3, 'CDEP000001', 'Debit', '02-MAR-20 04.48.29.496000000 AM', 200, 'teller withdrawal', 'A000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 4, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 20, 'teller withdrawal', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 0, 'CDEP000001', 'Debit', '02-DEC-20 05.48.29.935000000 PM', 35, 'teller withdrawal', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 2, 'CDEP000001', 'Debit', '02-JUN-20 04.48.29.125000000 AM', 4, 'service fee', 'A000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 400, 'teller withdrawal', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 3, 'CDEP000001', 'Debit', '02-MAR-20 04.48.29.496000000 AM', 80, 'teller withdrawal', 'A000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 4, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 40, 'teller withdrawal', 'T000001')
  SELECT 1 FROM dual;
```

Script Output | Task completed in 0.97 seconds

10 rows inserted.

## Insert entries into table cred\_trans:



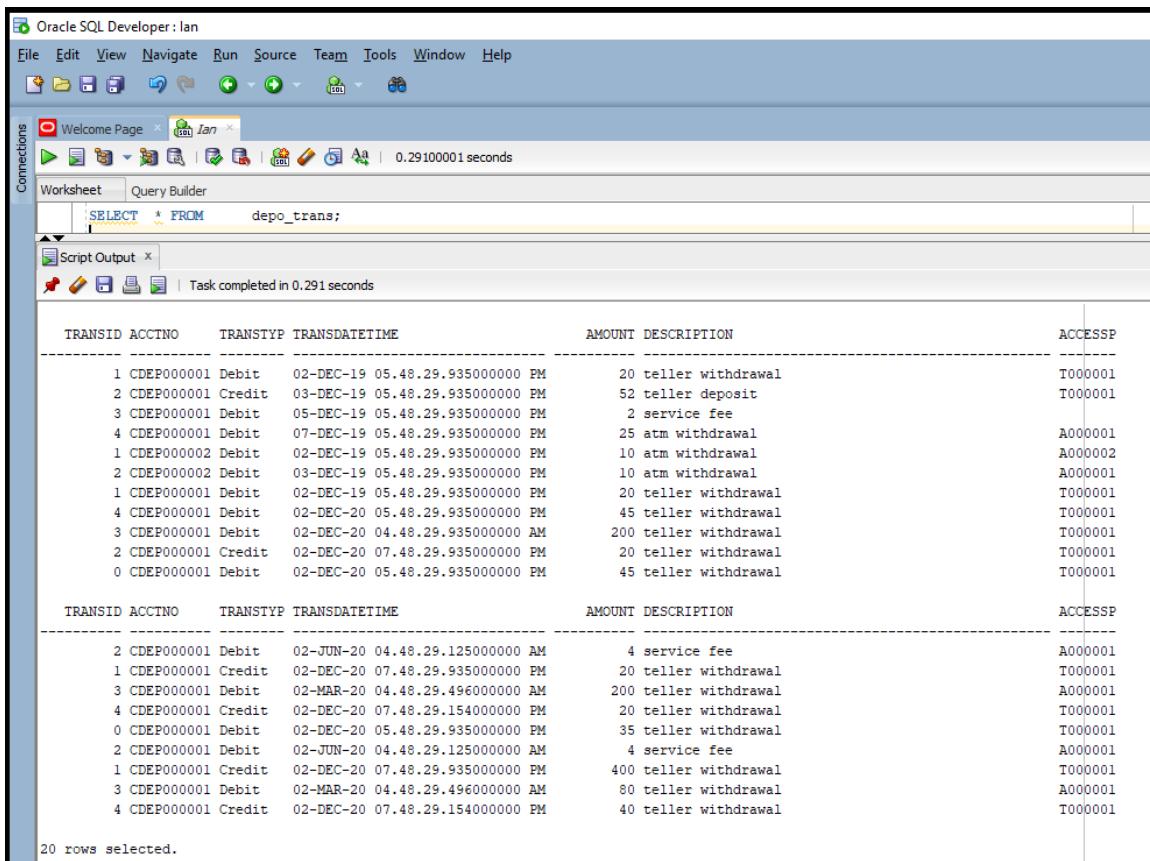
The screenshot shows the Oracle SQL Developer interface with a connection named 'ian'. The 'Worksheet' tab is active, displaying an SQL script for inserting data into the 'cred\_trans' table. The script uses the 'INSERT ALL' syntax to insert 10 rows. The 'Script Output' tab at the bottom shows the message '10 rows inserted.' indicating the successful execution of the script. The top status bar shows the execution time as 0.28299999 seconds.

```
INSERT ALL
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 0, 'CCR0000002', 'Credit', '07-DEC-20 05.16.13.443054000 PM', 175.25, 'publix')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 2, 'CCR0000002', 'Credit', '11-APR-20 05.16.13.430254000 AM', 29.55, 'visa')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 4, 'CCR0000002', 'Credit', '19-JAN-20 05.16.13.783054000 PM', 145.65, 'mike kim')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 1, 'CCR0000002', 'Credit', '21-MAR-20 05.16.13.170104000 AM', 17.05, 'shell gas')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 3, 'CCR0000002', 'Credit', '07-JUN-20 05.16.13.970458000 AM', 249.25, 'lucy smith')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 0, 'CCR0000002', 'Credit', '07-DEC-20 05.16.13.443054000 PM', 175.25, 'home depot')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 2, 'CCR0000002', 'Credit', '11-MAY-20 05.16.13.430254000 AM', 29.55, 'visa')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 4, 'CCR0000002', 'Credit', '19-JAN-20 05.16.13.783054000 PM', 105.65, 'mike kim')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 1, 'CCR0000002', 'Credit', '21-NOV-20 05.16.13.170104000 AM', 37.05, 'shell gas')
  INTO cred_trans (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES( 3, 'CCR0000002', 'Credit', '07-JUN-20 05.16.13.970458000 AM', 249.25, 'lucy smith')
  SELECT 1 FROM dual;
```

Script Output | Task completed in 0.283 seconds

10 rows inserted.

## Verify entries into table depo\_trans:



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 0.29100001 seconds

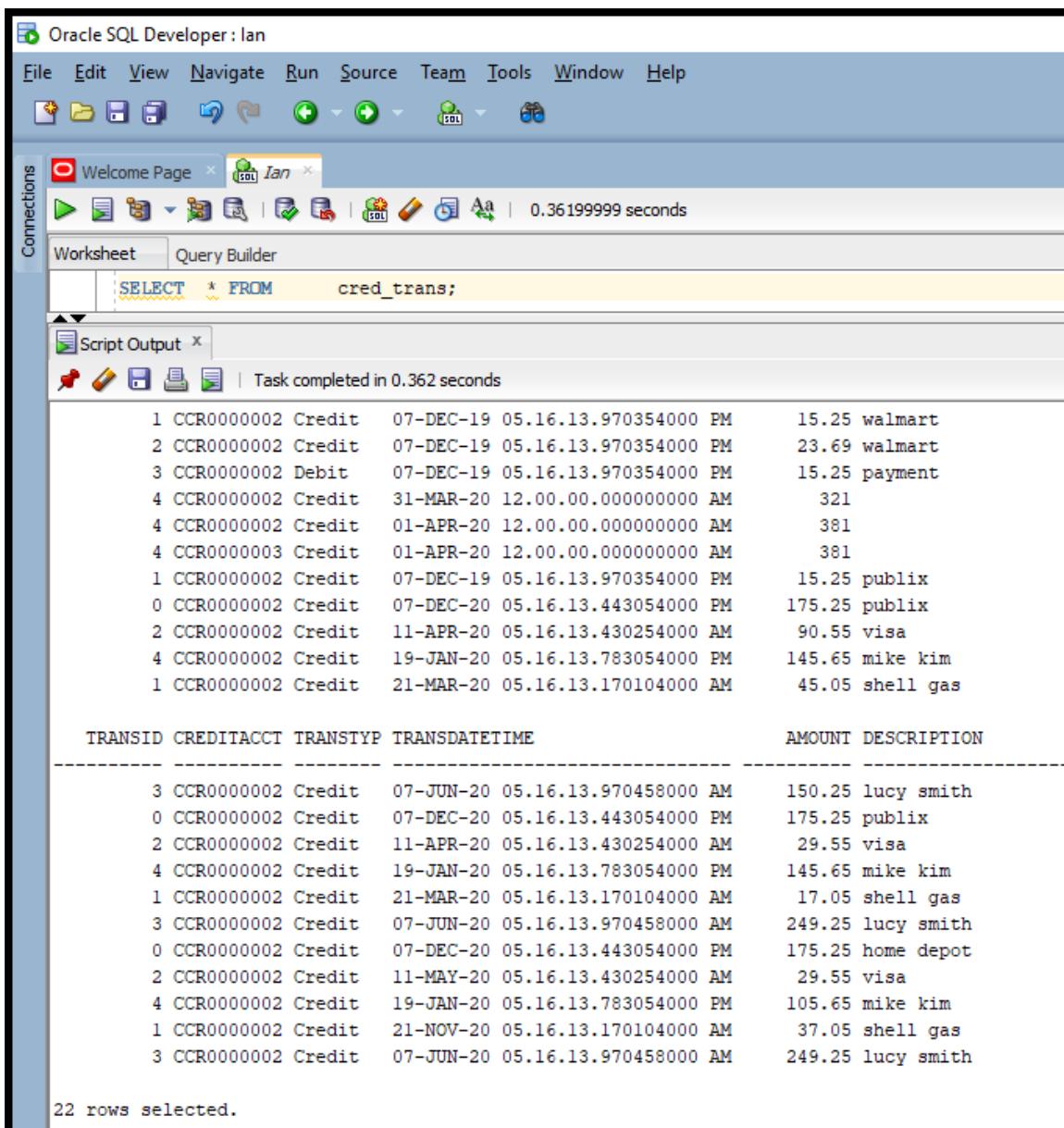
Worksheet Query Builder

Script Output Task completed in 0.291 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP00001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
2	CDEP00001	Credit	03-DEC-19 05.48.29.93500000 PM	52	teller deposit	T000001
3	CDEP00001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	
4	CDEP00001	Debit	07-DEC-19 05.48.29.93500000 PM	25	atm withdrawal	A000001
1	CDEP00002	Debit	02-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000002
2	CDEP00002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001
1	CDEP00001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
4	CDEP00001	Debit	02-DEC-20 05.48.29.93500000 PM	45	teller withdrawal	T000001
3	CDEP00001	Debit	02-DEC-20 04.48.29.93500000 AM	200	teller withdrawal	T000001
2	CDEP00001	Credit	02-DEC-20 07.48.29.93500000 PM	20	teller withdrawal	T000001
0	CDEP00001	Debit	02-DEC-20 05.48.29.93500000 PM	45	teller withdrawal	T000001
TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
2	CDEP00001	Debit	02-JUN-20 04.48.29.12500000 AM	4	service fee	A000001
1	CDEP00001	Credit	02-DEC-20 07.48.29.93500000 PM	20	teller withdrawal	T000001
3	CDEP00001	Debit	02-MAR-20 04.48.29.49600000 AM	200	teller withdrawal	A000001
4	CDEP00001	Credit	02-DEC-20 07.48.29.15400000 PM	20	teller withdrawal	T000001
0	CDEP00001	Debit	02-DEC-20 05.48.29.93500000 PM	35	teller withdrawal	T000001
2	CDEP00001	Debit	02-JUN-20 04.48.29.12500000 AM	4	service fee	A000001
1	CDEP00001	Credit	02-DEC-20 07.48.29.93500000 PM	400	teller withdrawal	T000001
3	CDEP00001	Debit	02-MAR-20 04.48.29.49600000 AM	80	teller withdrawal	A000001
4	CDEP00001	Credit	02-DEC-20 07.48.29.15400000 PM	40	teller withdrawal	T000001

20 rows selected.

## Verify entries into table cred\_trans:



The screenshot shows the Oracle SQL Developer interface with a query being run against a database connection named 'Ian'. The 'Worksheet' tab is active, displaying the SQL query:

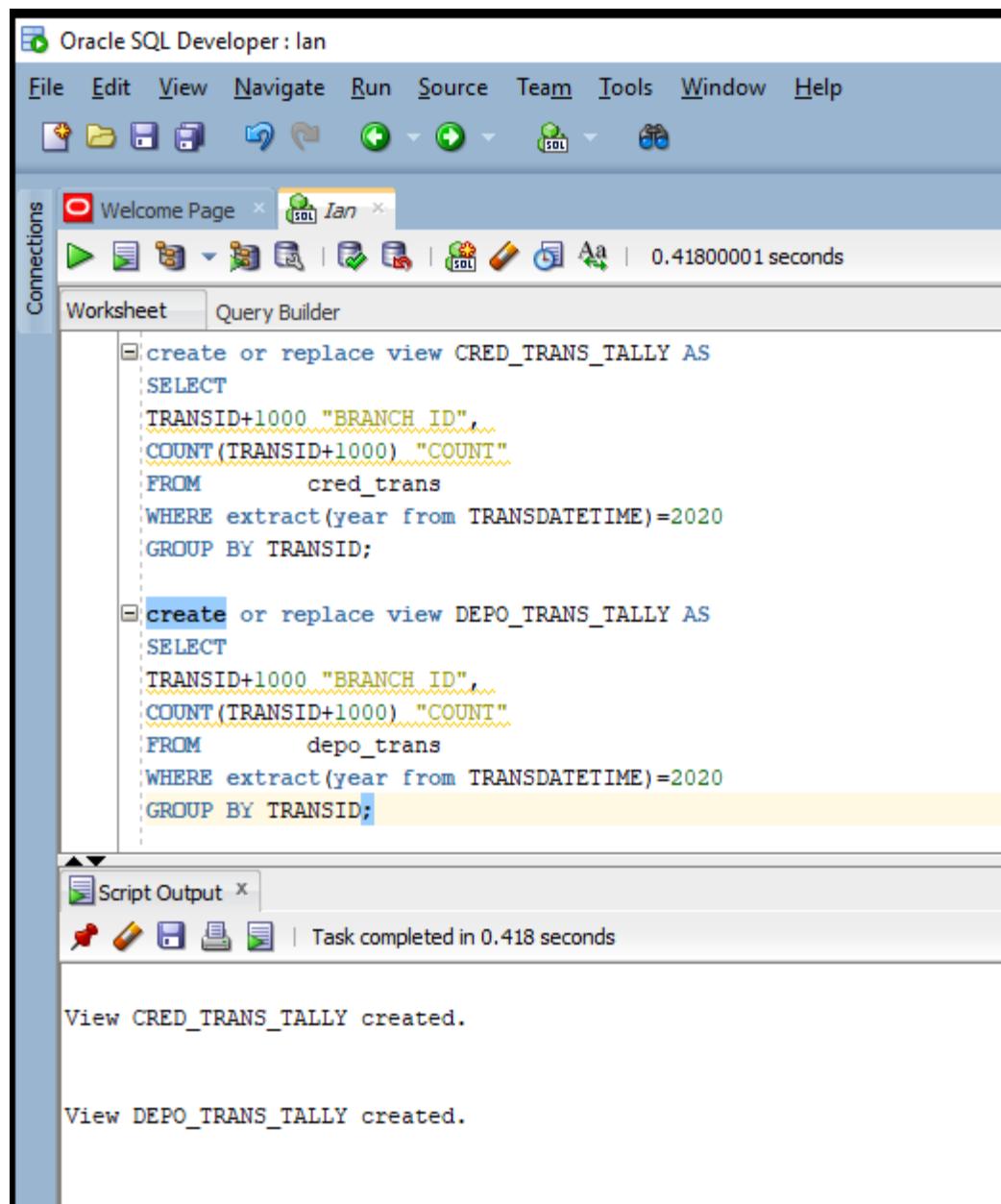
```
SELECT * FROM cred_trans;
```

The 'Script Output' tab shows the results of the query, which are two sets of data. The first set is a table with columns: TRANSID, CREDITACCT, TRANSTYP, TRANSDATETIME, AMOUNT, and DESCRIPTION. The second set is a table with the same columns. The results show various transactions, including Credit entries for different accounts and dates, with amounts ranging from 15.25 to 249.25, and descriptions like 'walmart', 'payment', 'publix', 'visa', 'mike kim', 'shell gas', and 'home depot'. The total number of rows selected is 22.

TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
1	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	15.25	walmart
2	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	23.69	walmart
3	CCR0000002	Debit	07-DEC-19 05.16.13.970354000 PM	15.25	payment
4	CCR0000002	Credit	31-MAR-20 12.00.00.000000000 AM	321	
4	CCR0000002	Credit	01-APR-20 12.00.00.000000000 AM	381	
4	CCR0000003	Credit	01-APR-20 12.00.00.000000000 AM	381	
1	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	15.25	publix
0	CCR0000002	Credit	07-DEC-20 05.16.13.443054000 PM	175.25	publix
2	CCR0000002	Credit	11-APR-20 05.16.13.430254000 AM	90.55	visa
4	CCR0000002	Credit	19-JAN-20 05.16.13.783054000 PM	145.65	mike kim
1	CCR0000002	Credit	21-MAR-20 05.16.13.170104000 AM	45.05	shell gas
TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
3	CCR0000002	Credit	07-JUN-20 05.16.13.970458000 AM	150.25	lucy smith
0	CCR0000002	Credit	07-DEC-20 05.16.13.443054000 PM	175.25	publix
2	CCR0000002	Credit	11-APR-20 05.16.13.430254000 AM	29.55	visa
4	CCR0000002	Credit	19-JAN-20 05.16.13.783054000 PM	145.65	mike kim
1	CCR0000002	Credit	21-MAR-20 05.16.13.170104000 AM	17.05	shell gas
3	CCR0000002	Credit	07-JUN-20 05.16.13.970458000 AM	249.25	lucy smith
0	CCR0000002	Credit	07-DEC-20 05.16.13.443054000 PM	175.25	home depot
2	CCR0000002	Credit	11-MAY-20 05.16.13.430254000 AM	29.55	visa
4	CCR0000002	Credit	19-JAN-20 05.16.13.783054000 PM	105.65	mike kim
1	CCR0000002	Credit	21-NOV-20 05.16.13.170104000 AM	37.05	shell gas
3	CCR0000002	Credit	07-JUN-20 05.16.13.970458000 AM	249.25	lucy smith

22 rows selected.

Create views cred\_trans\_trans and depo\_trans\_tally using tables depo\_trans and cred\_trans



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying two SQL scripts for creating views. The first script creates 'CRED\_TRANS\_TALLY' and the second creates 'DEPO\_TRANS\_TALLY'. Both scripts use a SELECT statement with COUNT and GROUP BY clauses. The 'Script Output' tab at the bottom shows the successful creation of both views.

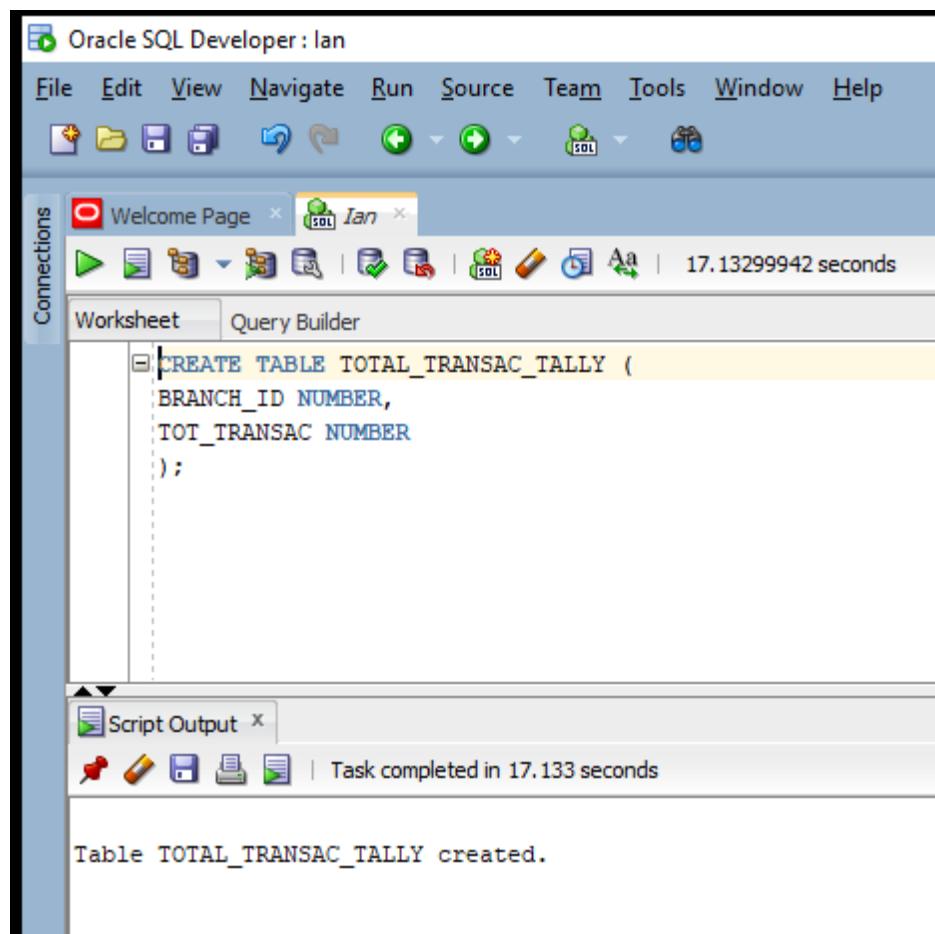
```
create or replace view CRED_TRANS_TALLY AS
SELECT
    TRANSID+1000 "BRANCH_ID",
    COUNT(TRANSID+1000) "COUNT"
FROM
    cred_trans
WHERE extract(year from TRANSDATETIME)=2020
GROUP BY TRANSID;

create or replace view DEPO_TRANS_TALLY AS
SELECT
    TRANSID+1000 "BRANCH_ID",
    COUNT(TRANSID+1000) "COUNT"
FROM
    depo_trans
WHERE extract(year from TRANSDATETIME)=2020
GROUP BY TRANSID;
```

View CRED\_TRANS\_TALLY created.

View DEPO\_TRANS\_TALLY created.

## Create empty table total\_transac\_tally



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations and SQL tasks. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet area contains the following SQL code:

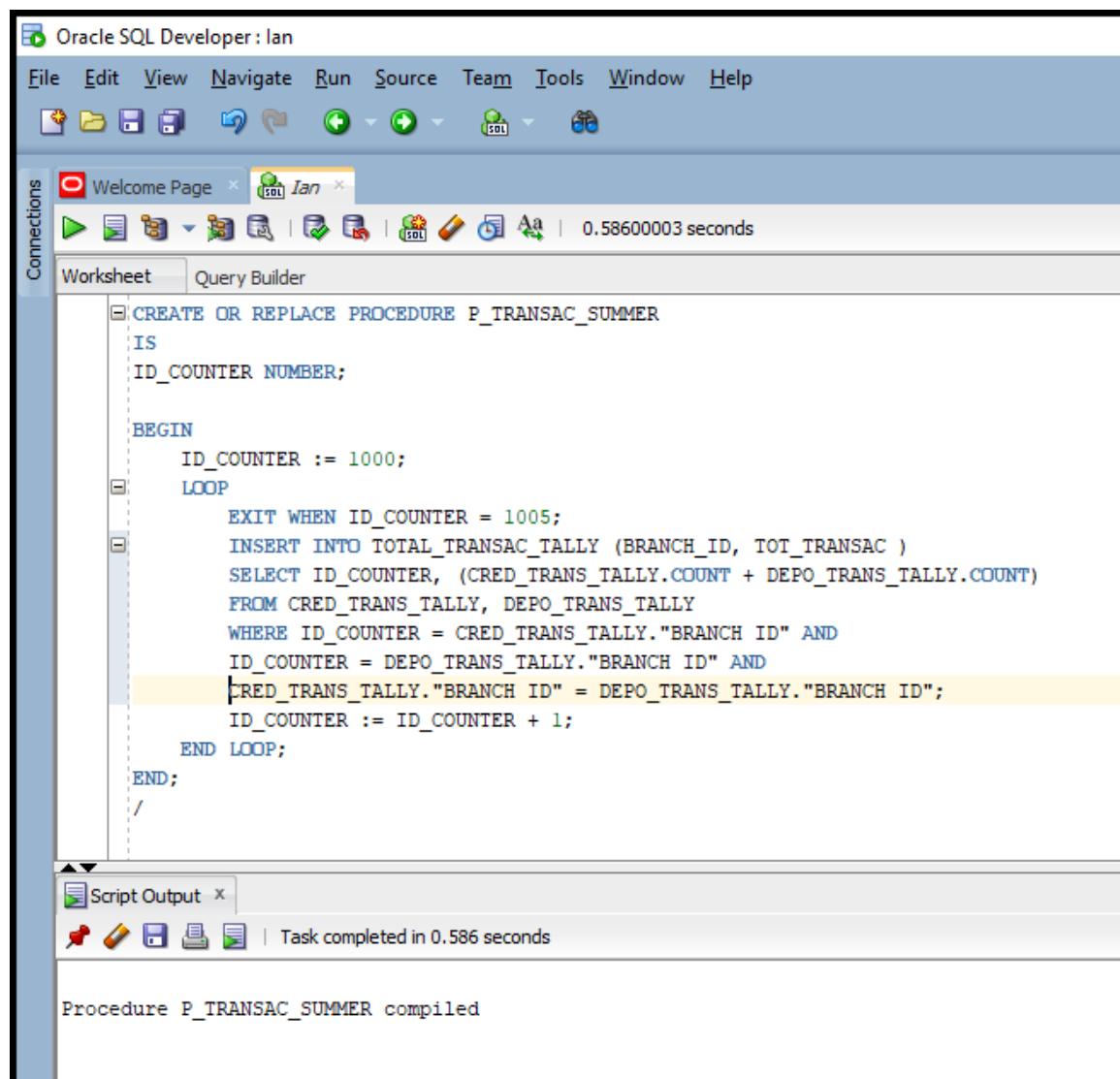
```
CREATE TABLE TOTAL_TRANSAC_TALLY (
  BRANCH_ID NUMBER,
  TOT_TRANSAC NUMBER
);
```

The "Script Output" tab at the bottom shows the result of the execution:

```
Table TOTAL_TRANSAC_TALLY created.
```

The status bar at the bottom right indicates "Task completed in 17.133 seconds".

**Create looping procedure to fill empty table total\_transac\_tally.  
Procedure sums counts from cred\_trans\_tally and depo\_trans\_tally into  
total\_transac\_tally.**



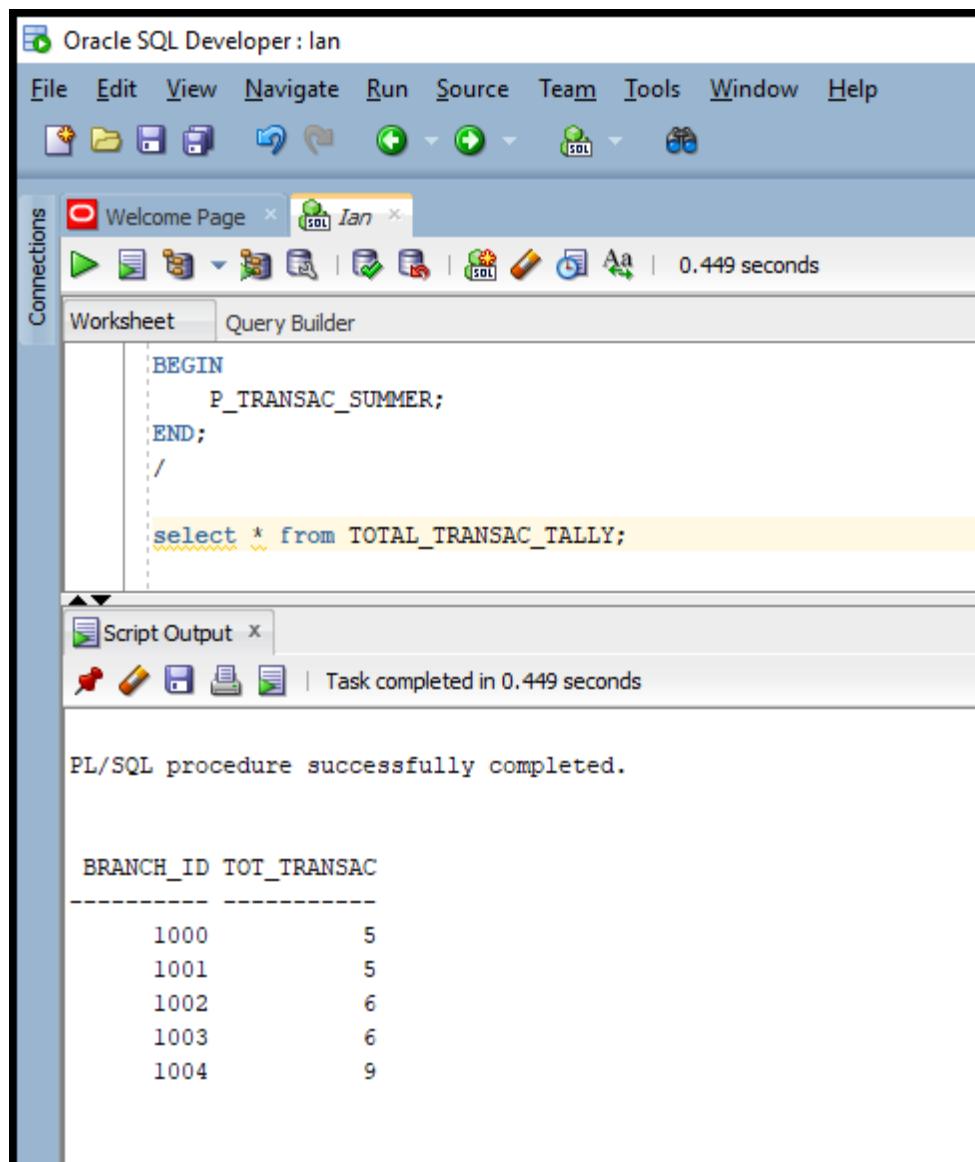
The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE P_TRANSAC_SUMMER
IS
ID_COUNTER NUMBER;

BEGIN
    ID_COUNTER := 1000;
LOOP
    EXIT WHEN ID_COUNTER = 1005;
    INSERT INTO TOTAL_TRANSAC_TALLY (BRANCH_ID, TOT_TRANSAC )
    SELECT ID_COUNTER, (CRED_TRANS_TALLY.COUNT + DEPO_TRANS_TALLY.COUNT)
    FROM CRED_TRANS_TALLY, DEPO_TRANS_TALLY
    WHERE ID_COUNTER = CRED_TRANS_TALLY."BRANCH ID" AND
    ID_COUNTER = DEPO_TRANS_TALLY."BRANCH ID" AND
    CRED_TRANS_TALLY."BRANCH ID" = DEPO_TRANS_TALLY."BRANCH ID";
    ID_COUNTER := ID_COUNTER + 1;
END LOOP;
END;
/
```

The code is highlighted with syntax coloring. The 'Script Output' tab at the bottom shows the message: "Procedure P\_TRANSAC\_SUMMER compiled".

## Utilize procedure to fill empty table total\_transac\_tally



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following PL/SQL code:

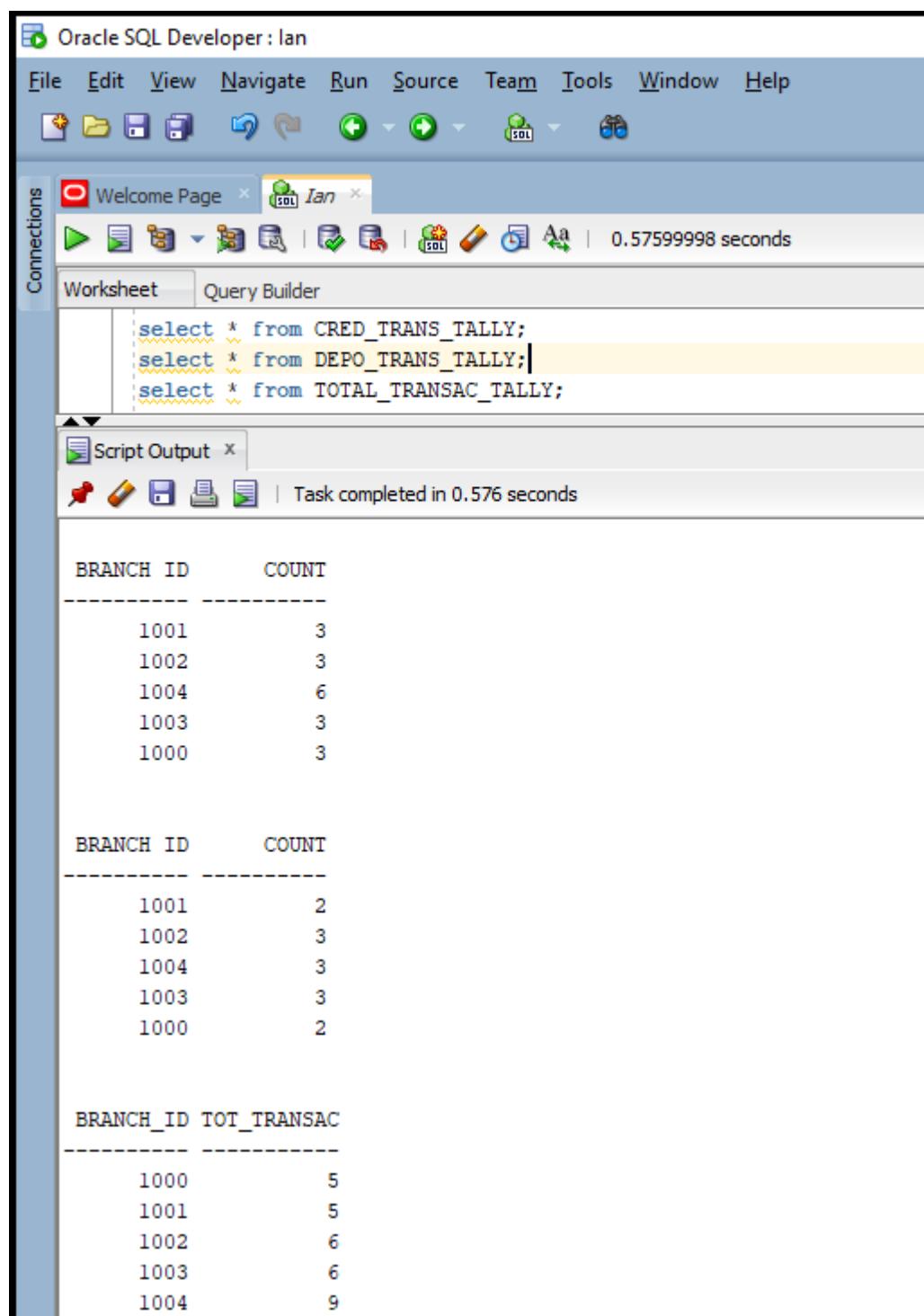
```
BEGIN
    P_TRANSAC_SUMMER;
END;
/
select * from TOTAL_TRANSAC_TALLY;
```

The 'Script Output' tab shows the results of the execution:

PL/SQL procedure successfully completed.

BRANCH_ID	TOT_TRANSAC
1000	5
1001	5
1002	6
1003	6
1004	9

## Verify how procedure filled empty table `total_transac_tally`



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying three SQL queries:

```
select * from CRED_TRANS_TALLY;
select * from DEPO_TRANS_TALLY;
select * from TOTAL_TRANSAC_TALLY;
```

The 'Script Output' tab shows the results of these queries:

BRANCH_ID	COUNT
1001	3
1002	3
1004	6
1003	3
1000	3

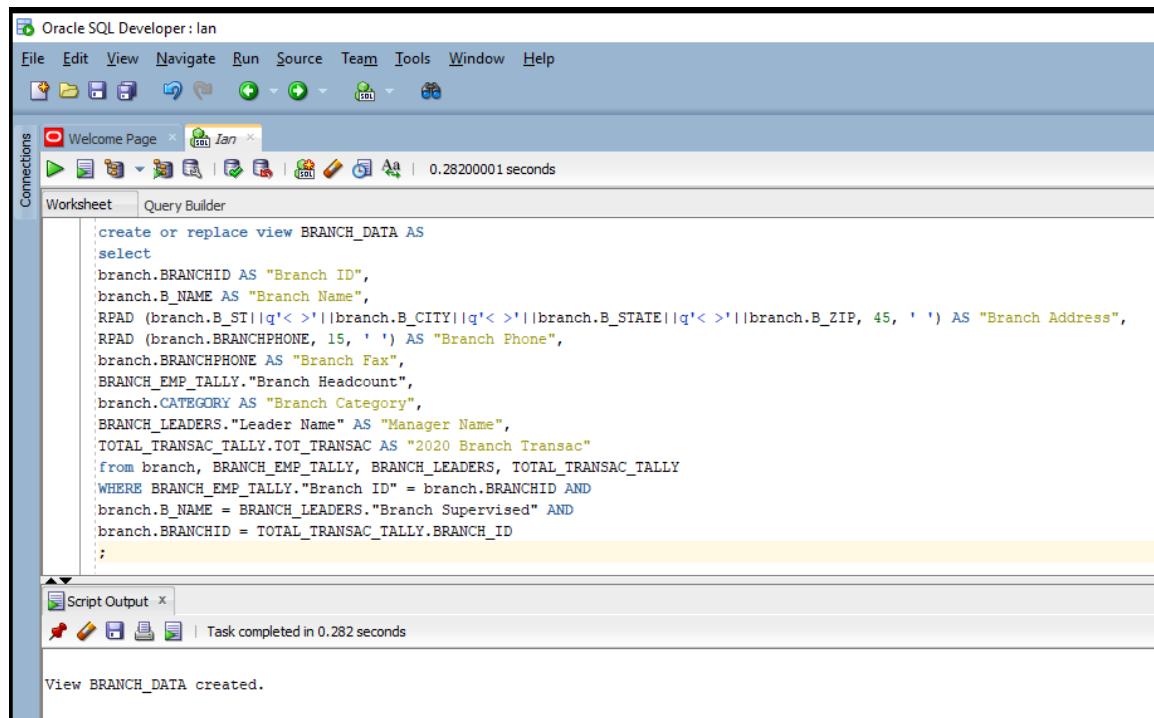
BRANCH_ID	COUNT
1001	2
1002	3
1004	3
1003	3
1000	2

BRANCH_ID	TOT_TRANSAC
1000	5
1001	5
1002	6
1003	6
1004	9

**Create view branch\_data with the following attributes from above views and tables:**

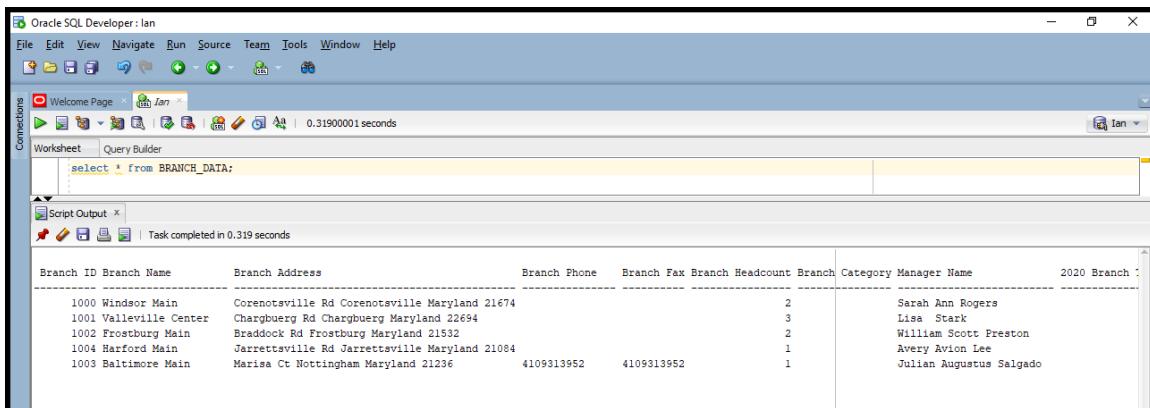
Branch ID
Branch Name
Address
Phone Number
Fax Number (to be inputted later)
Number of Employee at this Branch
Category
Manager Name
Total Transactions Done at this Branch for year 2020



```
create or replace view BRANCH_DATA AS
select
branch.BRANCHID AS "Branch ID",
branch.B_NAME AS "Branch Name",
RPAD (branch.B_ST||q'< >'||branch.B_CITY||q'< >'||branch.B_STATE||q'< >'||branch.B_ZIP, 45, ' ') AS "Branch Address",
RPAD (branch.BRANCHPHONE, 15, ' ') AS "Branch Phone",
branch.BRANCHPHONE AS "Branch Fax",
BRANCH_EMP_TALLY."Branch Headcount",
branch.CATEGORY AS "Branch Category",
branch.LEADERS."Leader Name" AS "Manager Name",
TOTAL_TRANSAC_TALLY.TOT_TRANSAC AS "2020 Branch Transac"
from branch, BRANCH_EMP_TALLY, BRANCH_LEADERS, TOTAL_TRANSAC_TALLY
WHERE BRANCH_EMP_TALLY."Branch ID" = branch.BRANCHID AND
branch.B_NAME = BRANCH_LEADERS."Branch Supervised" AND
branch.BRANCHID = TOTAL_TRANSAC_TALLY.BRANCH_ID
;
```

View BRANCH\_DATA created.

## Verify contents of view branch\_data



Oracle SQL Developer : Ian

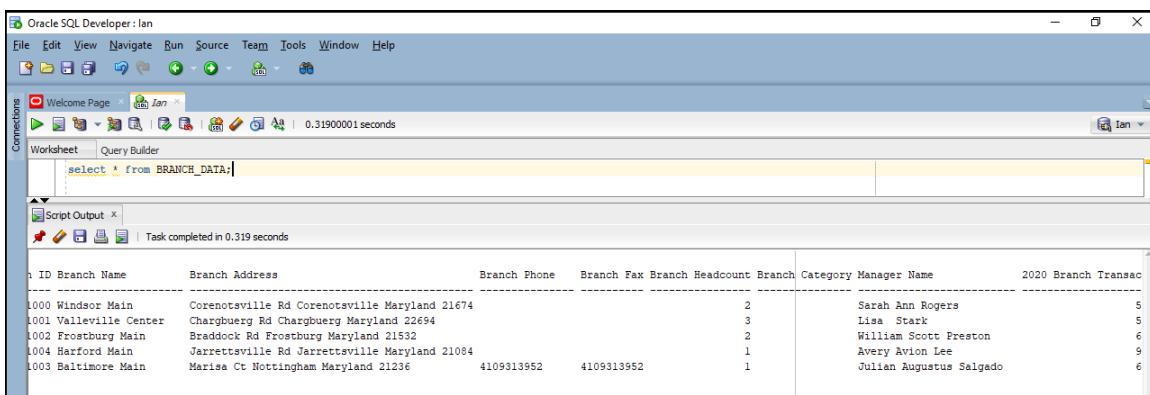
File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

select \* from BRANCH\_DATA;

Task completed in 0.319 seconds

Branch ID	Branch Name	Branch Address	Branch Phone	Branch Fax	Branch Headcount	Branch Category	Manager Name	2020 Branch
1000	Windsor Main	Corenotsville Rd Corenotsville Maryland 21674			2		Sarah Ann Rogers	
1001	Valleville Center	Chargburg Rd Chargburg Maryland 22694			3		Lisa Stark	
1002	Frostburg Main	Braddock Rd Frostburg Maryland 21532			2		William Scott Preston	
1004	Harford Main	Jarrettsville Rd Jarrettsville Maryland 21084			1		Avery Avion Lee	
1003	Baltimore Main	Marisa Ct Nottingham Maryland 21236	4109313952	4109313952	1		Julian Augustus Salgado	



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

select \* from BRANCH\_DATA;

Task completed in 0.319 seconds

Branch ID	Branch Name	Branch Address	Branch Phone	Branch Fax	Branch Headcount	Branch Category	Manager Name	2020 Branch Transac
1000	Windsor Main	Corenotsville Rd Corenotsville Maryland 21674			2		Sarah Ann Rogers	5
1001	Valleville Center	Chargburg Rd Chargburg Maryland 22694			3		Lisa Stark	5
1002	Frostburg Main	Braddock Rd Frostburg Maryland 21532			2		William Scott Preston	6
1004	Harford Main	Jarrettsville Rd Jarrettsville Maryland 21084			1		Avery Avion Lee	9
1003	Baltimore Main	Marisa Ct Nottingham Maryland 21236	4109313952	4109313952	1		Julian Augustus Salgado	6

^^^^^^^^^^^^^^^^^^^^^^^^

## **CHAPTER 1D Starts here**

^^^^^^^^^^^^^^^^^^^^^^^^

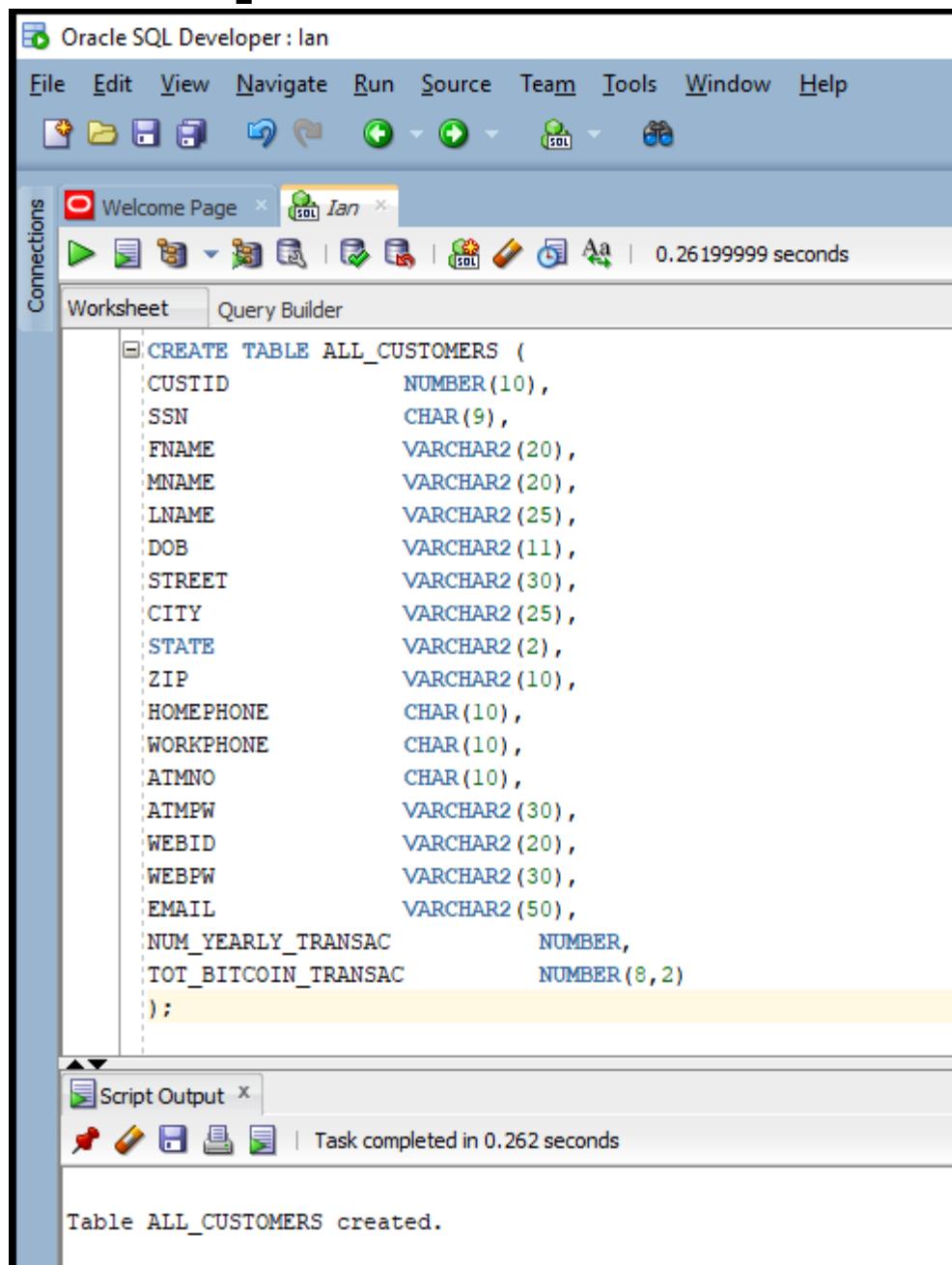
D. Valued\_Customers with the following attributes:

Describe how you select the important customers in this bank.

SSN
Name
Age
Home Phone
Work Phone
Address
Zip Code
Email <input type="checkbox"/> to be inputted later
State they live in
Total number of transactions the customer has done in a given year
Total amount (in bitcoin) of transactions the customer has done in a given year (you choose the year)

**=>Valued customers have > 65,000 total bitcoin transactions or > 6,000 yearly transactions.**

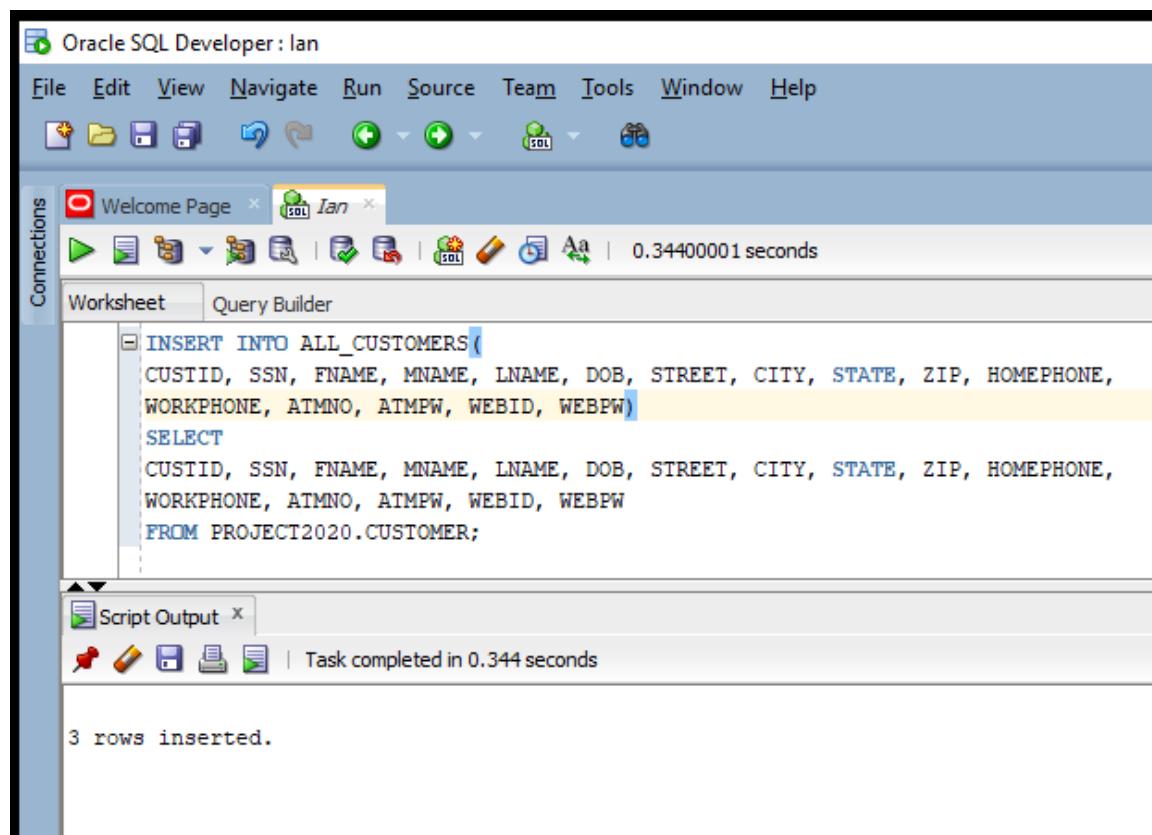
## Define table all\_customers and its attributes:



The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** A list of connections, with "Welcome Page" and "Ian" selected.
- Worksheet:** The main workspace where the SQL code is entered. The code defines the `ALL_CUSTOMERS` table with the following columns and data types:
  - `CUSTID`: NUMBER(10)
  - `SSN`: CHAR(9)
  - `FNAME`: VARCHAR2(20)
  - `MNAME`: VARCHAR2(20)
  - `LNAME`: VARCHAR2(25)
  - `DOB`: VARCHAR2(11)
  - `STREET`: VARCHAR2(30)
  - `CITY`: VARCHAR2(25)
  - `STATE`: VARCHAR2(2)
  - `ZIP`: VARCHAR2(10)
  - `HOMEPHONE`: CHAR(10)
  - `WORKPHONE`: CHAR(10)
  - `ATMNO`: CHAR(10)
  - `ATMPW`: VARCHAR2(30)
  - `WEBID`: VARCHAR2(20)
  - `WEBPW`: VARCHAR2(30)
  - `EMAIL`: VARCHAR2(50)
  - `NUM_YEARLY_TRANSAC`: NUMBER
  - `TOT_BITCOIN_TRANSAC`: NUMBER(8,2)
- Script Output:** A panel at the bottom showing the result of the table creation: "Table ALL\_CUSTOMERS created." and a completion time of "0.262 seconds".

**Insert values into table all\_customers from table project2020.customer**



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL execution. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The query editor contains the following SQL script:

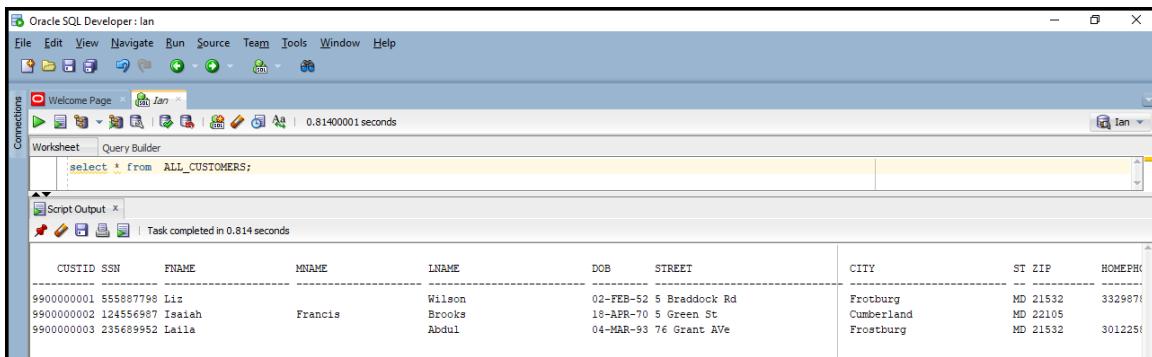
```
INSERT INTO ALL_CUSTOMERS(
  CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW)
SELECT
  CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW
FROM PROJECT2020.CUSTOMER;
```

The "Script Output" tab shows the result of the execution:

```
3 rows inserted.
```

The status bar at the bottom right indicates "Task completed in 0.344 seconds".

## Verify contents of table `all_customers`



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

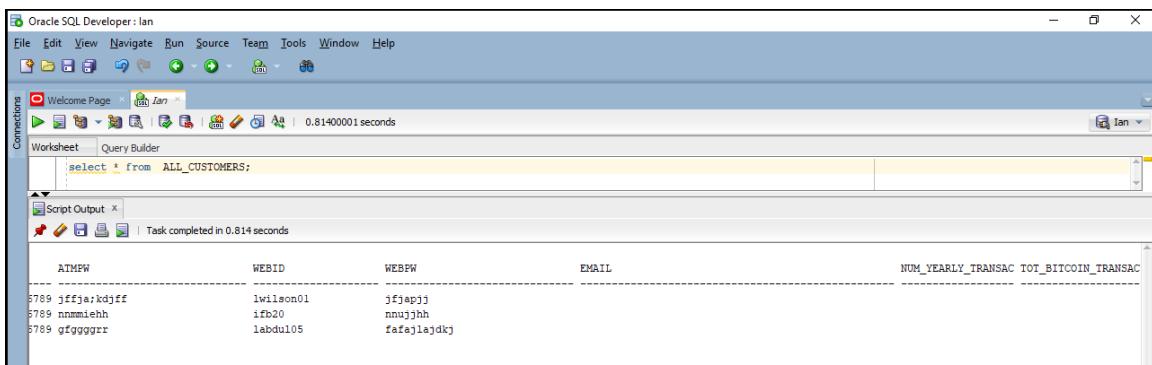
Connections Welcome Page Ian

Worksheet Query Builder

```
select * from ALL_CUSTOMERS;
```

Script Output Task completed in 0.814 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEPLK
9900000001	555887798	Liz		Wilson	02-FEB-52	5 Braddock Rd	Frostburg	MD	21532	3329871
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-70	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-93	76 Grant Ave	Frostburg	MD	21532	3012251

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

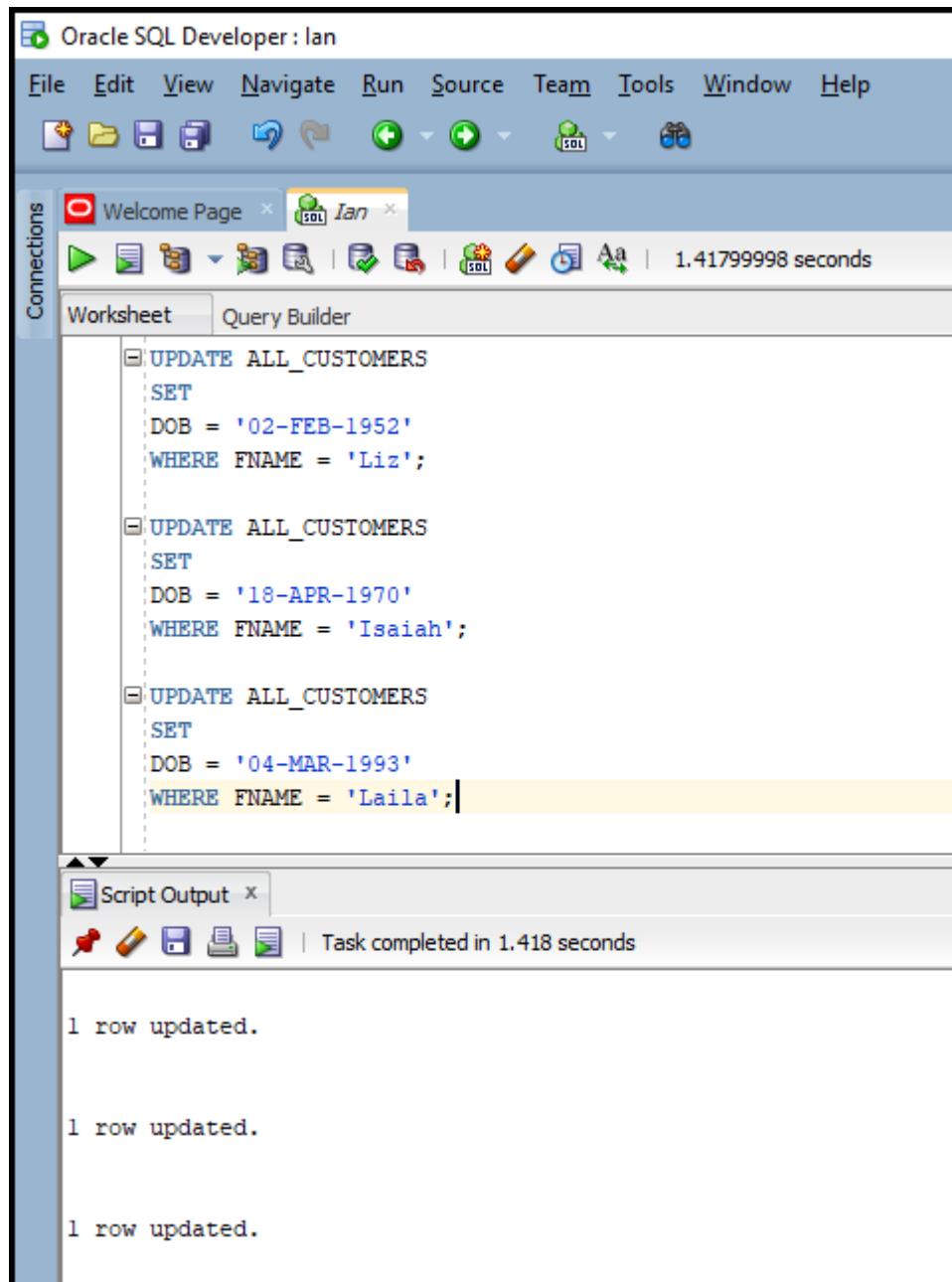
Worksheet Query Builder

```
select * from ALL_CUSTOMERS;
```

Script Output Task completed in 0.814 seconds

ATMPW	WEBID	WEBPW	EMAIL	NUM_YEARLY_TRANSAC	TOT_BITCOIN_TRANSAC
6789 jffja;kdjff	lwilson01	jfjapjj			
6789 nnmmeihh	ifb20	nnujjh			
6789 gfqgggr	labdul05	fafaqlajdkj			

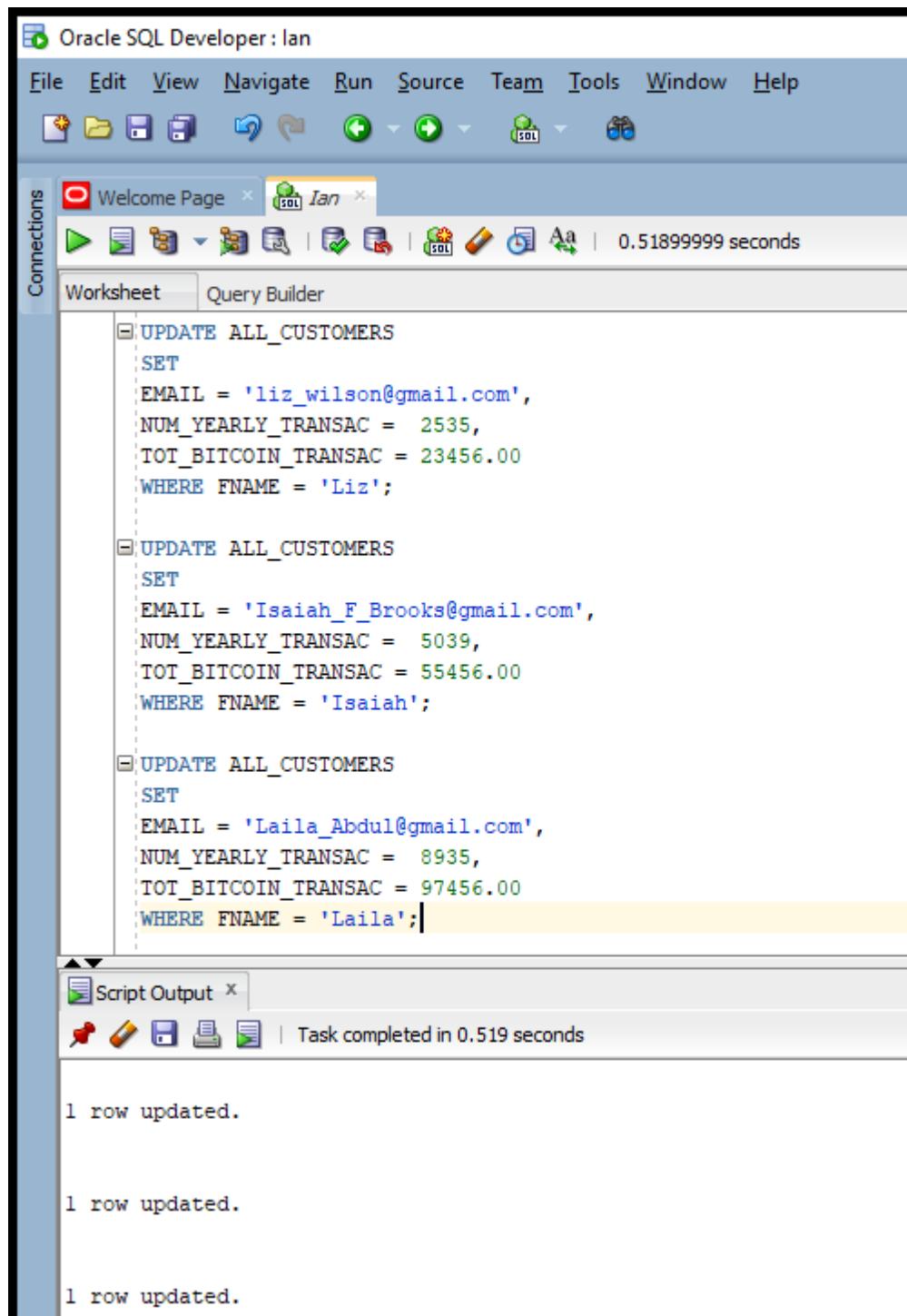
## Update contents of table all\_customers



The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** Welcome Page, Ian
- Worksheet:** Three update statements are listed:
  - UPDATE ALL\_CUSTOMERS  
SET  
DOB = '02-FEB-1952'  
WHERE FNAME = 'Liz';
  - UPDATE ALL\_CUSTOMERS  
SET  
DOB = '18-APR-1970'  
WHERE FNAME = 'Isaiah';
  - UPDATE ALL\_CUSTOMERS  
SET  
DOB = '04-MAR-1993'  
WHERE FNAME = 'Laila';
- Script Output:** The output shows three rows of text:
  - 1 row updated.
  - 1 row updated.
  - 1 row updated.Followed by the message: Task completed in 1.418 seconds.

## Further update contents of table all\_customers



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying three separate UPDATE statements for the 'all\_customers' table. Each statement sets the 'EMAIL', 'NUM\_YEARLY\_TRANSAC', and 'TOT\_BITCOIN\_TRANSAC' columns for a specific customer based on their 'FNAME'. The 'Script Output' tab at the bottom shows the confirmation '1 row updated.' for each of the three statements.

```
UPDATE ALL_CUSTOMERS
SET
EMAIL = 'liz_wilson@gmail.com',
NUM_YEARLY_TRANSAC = 2535,
TOT_BITCOIN_TRANSAC = 23456.00
WHERE FNAME = 'Liz';

UPDATE ALL_CUSTOMERS
SET
EMAIL = 'Isaiah_F_Brooks@gmail.com',
NUM_YEARLY_TRANSAC = 5039,
TOT_BITCOIN_TRANSAC = 55456.00
WHERE FNAME = 'Isaiah';

UPDATE ALL_CUSTOMERS
SET
EMAIL = 'Laila_Abdul@gmail.com',
NUM_YEARLY_TRANSAC = 8935,
TOT_BITCOIN_TRANSAC = 97456.00
WHERE FNAME = 'Laila';
```

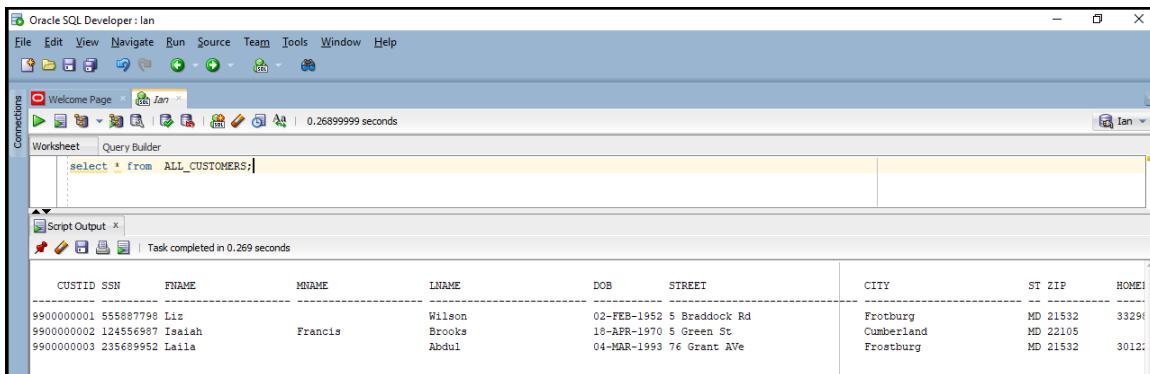
Script Output

```
1 row updated.

1 row updated.

1 row updated.
```

## Verify updated contents of table all\_customers



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

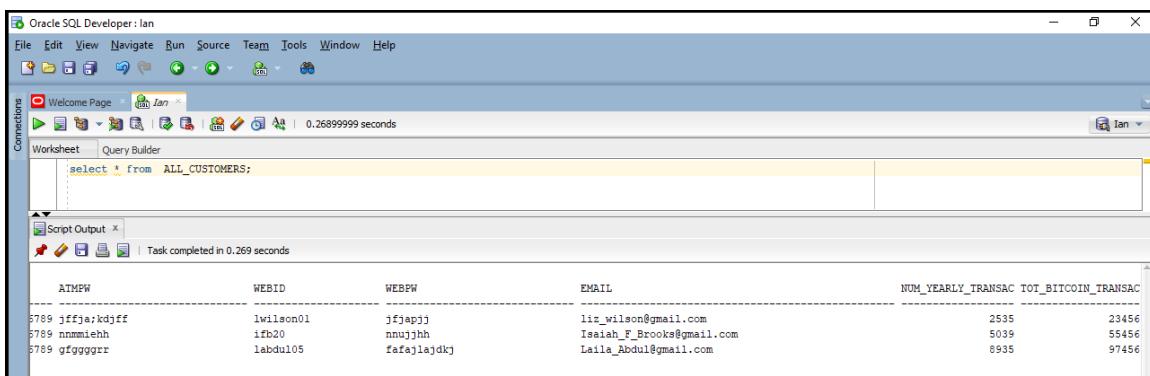
Connections Welcome Page Ian

Worksheet Query Builder

```
select * from ALL_CUSTOMERS;
```

Script Output Task completed in 0.269 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
5900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frostburg	MD	21532	33291
5900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
5900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30121



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

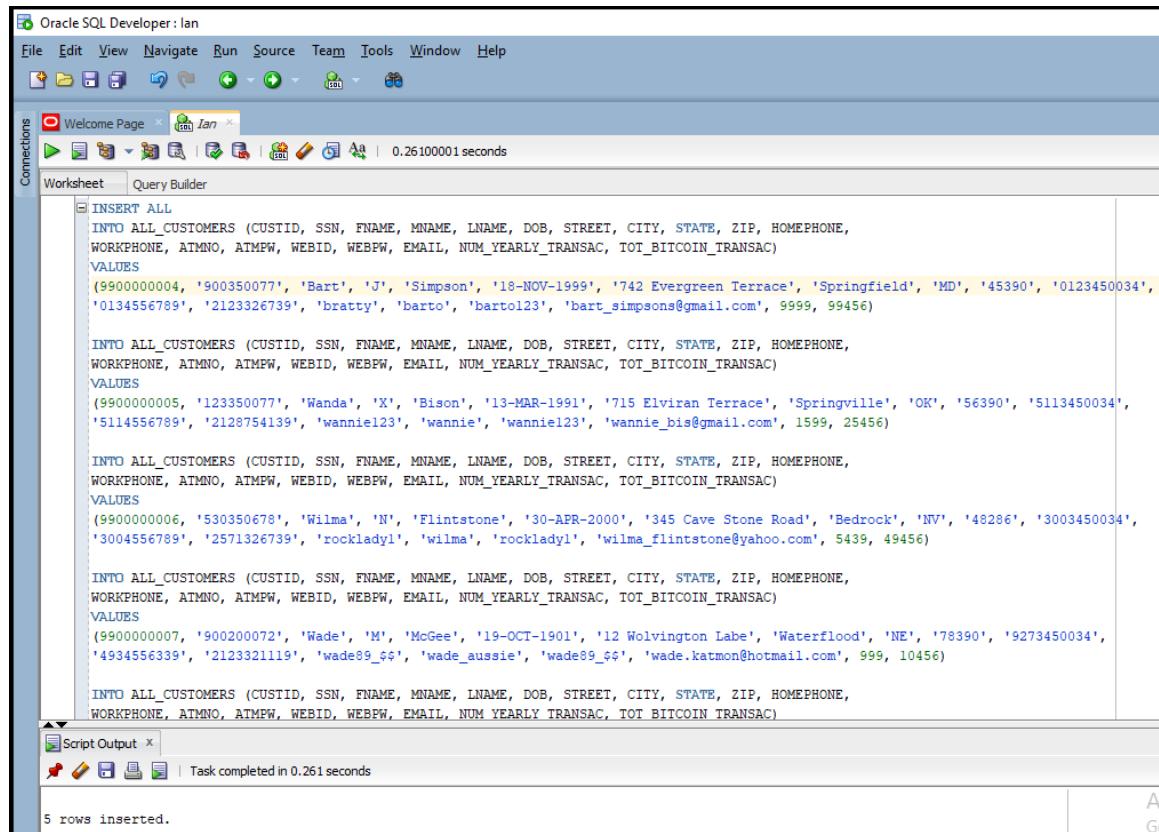
Worksheet Query Builder

```
select * from ALL_CUSTOMERS;
```

Script Output Task completed in 0.269 seconds

ATMPW	WEBID	WEBPW	EMAIL	NUM_YEARLY_TRANSAC	TOT_BITCOIN_TRANSAC
5789 jifjja:kdjff	1wilson01	jifjapjj	liz_wilson@gmail.com	2535	23456
5789 nnmmiehh	ifb20	nnujjh	Isaiah_F_Brooks@gmail.com	5039	55456
5789 gfggggrr	labdul105	fafajlaejdkj	Laila_Abdul@gmail.com	8935	97456

**Table all\_customers contains only three entries. Creatively add entries to all\_customers table.**



The screenshot shows the Oracle SQL Developer interface. The Worksheet tab contains a multi-line SQL script for inserting data into the 'all\_customers' table. The Script Output tab shows the result of the execution, indicating 5 rows inserted.

```
INSERT ALL
  INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (9900000004, '900350077', 'Bart', 'J', 'Simpson', '18-NOV-1999', '742 Evergreen Terrace', 'Springfield', 'MD', '45390', '0123450034',
  '0134556789', '2123326739', 'bratty', 'bart0123', 'bart_simpons@gmail.com', 9999, 99456)

  INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (9900000005, '123350077', 'Wanda', 'X', 'Bison', '13-MAR-1991', '715 Elviran Terrace', 'Springville', 'OK', '56390', '5113450034',
  '5114556789', '2128754139', 'wannie123', 'wannie', 'wannie123', 'wannie_bis@gmail.com', 1599, 25456)

  INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (9900000006, '530350678', 'Wilma', 'N', 'Flintstone', '30-APR-2000', '345 Cave Stone Road', 'Bedrock', 'NV', '48286', '3003450034',
  '3004556789', '2571326739', 'rocklady1', 'wilma', 'rocklady1', 'wilma_flintstone@yahoo.com', 5439, 49456)

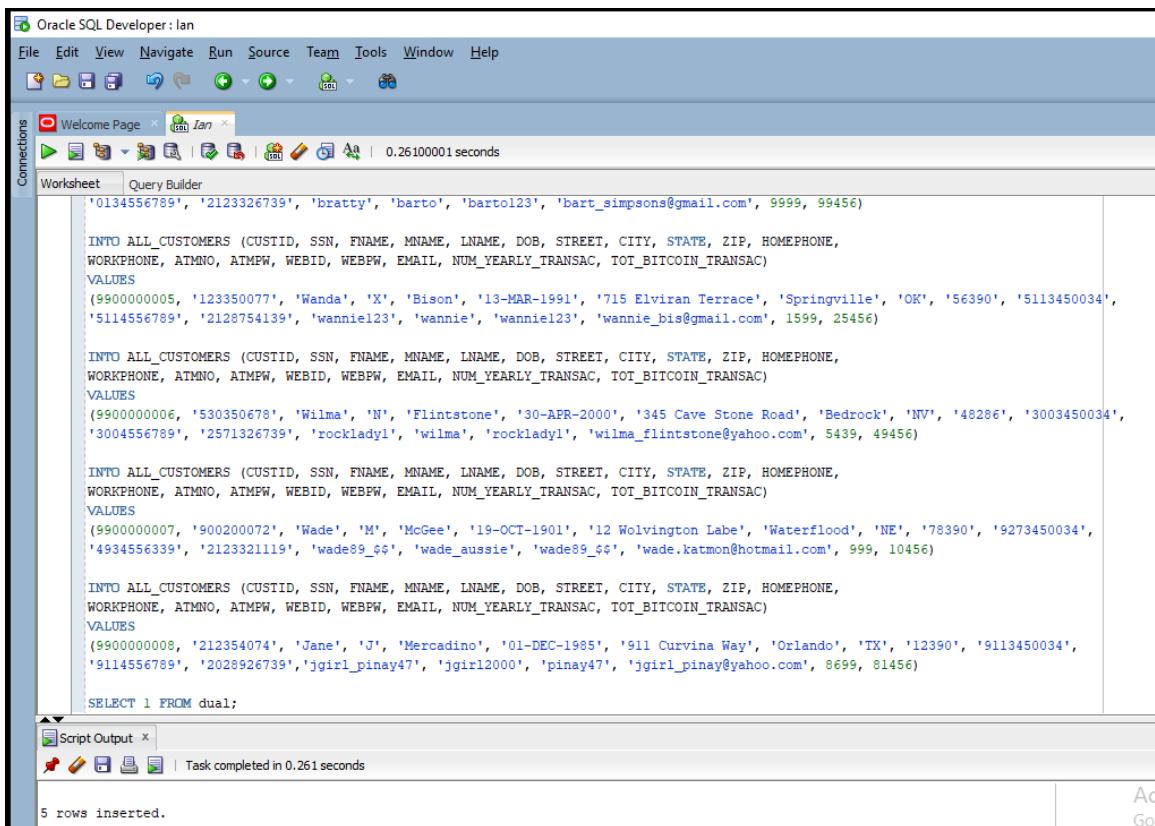
  INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (9900000007, '900200072', 'Wade', 'M', 'McGee', '19-OCT-1901', '12 Wolverton Labe', 'Waterflood', 'NE', '78390', '9273450034',
  '4934556339', '2123321119', 'wade89_66', 'wade_aussie', 'wade89_66', 'wade.katmon@hotmail.com', 999, 10456)

  INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
```

Script Output tab output:

```
5 rows inserted.
```

## Insert more creative entries into all\_customers table.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
'0134556789', '2123326739', 'bratty', 'bart', 'bart0123', 'bart_simpsons@gmail.com', 9999, 99456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000005, '123350077', 'Wanda', 'X', 'Bison', '13-MAR-1991', '715 Elviran Terrace', 'Springville', 'OK', '56390', '5113450034',
'5114556789', '2128754139', 'wannie123', 'wannie', 'wannie123', 'wannie_bis@gmail.com', 1599, 25456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000006, '530350678', 'Wilma', 'N', 'Flintstone', '30-APR-2000', '345 Cave Stone Road', 'Bedrock', 'NV', '48286', '3003450034',
'3004556789', '2571326739', 'rocklady1', 'wilma', 'rocklady1', 'wilma_flintstone@yahoo.com', 5439, 49456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000007, '900200072', 'Wade', 'M', 'McGee', '19-OCT-1901', '12 Wolverton Labe', 'Waterflood', 'NE', '78390', '9273450034',
'4934556339', '2123321119', 'wade89_55', 'wade_aussie', 'wade89_55', 'wade.katmon@hotmail.com', 999, 10456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000008, '212354074', 'Jane', 'J', 'Mercadino', '01-DEC-1985', '911 Curvina Way', 'Orlando', 'TX', '12390', '9113450034',
'9114556789', '2028926739', 'jgirl_pinay47', 'jgirl12000', 'pinay47', 'jgirl_pinay@yahoo.com', 8699, 81456)

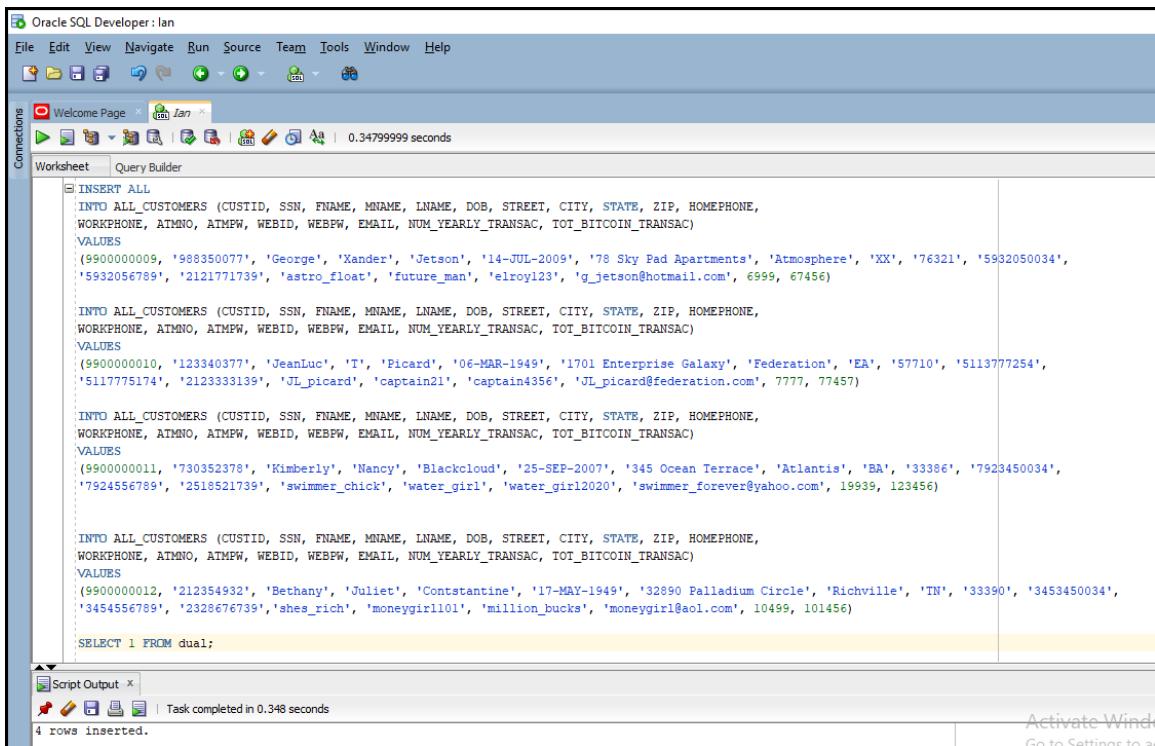
SELECT 1 FROM dual;
```

Script Output x

Task completed in 0.261 seconds

5 rows inserted.

## Still more creative entries into all\_customers table.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
INSERT ALL
INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000009, '988350077', 'George', 'Xander', 'Jetson', '14-JUL-2009', '78 Sky Pad Apartments', 'Atmosphere', 'XX', '76321', '5932050034',
'5932056789', '2121771739', 'astro_float', 'future_man', 'elroy123', 'g_jetson@hotmail.com', 6999, 67456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000010, '123340377', 'JeanLuc', 'T', 'Picard', '06-MAR-1949', '1701 Enterprise Galaxy', 'Federation', 'EA', '57710', '5113777254',
'5117775174', '212333139', 'JL_picard', 'captain21', 'captain4356', 'JL_picard@federation.com', 7777, 77457)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000011, '730352378', 'Kimberly', 'Nancy', 'Blackcloud', '25-SEP-2007', '345 Ocean Terrace', 'Atlantis', 'BA', '33386', '7923450034',
'7924556789', '2518521739', 'swimmer_chick', 'water_girl', 'water_girl2020', 'swimmer_forever@yahoo.com', 19939, 123456)

INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMMO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES
(9900000012, '212354932', 'Bethany', 'Juliet', 'Constantine', '17-MAY-1949', '32890 Palladium Circle', 'Richville', 'TN', '33390', '3453450034',
'3454556789', '2328676739', 'shes_rich', 'million_bucks', 'moneygirl101', 'moneygirl@aol.com', 10499, 101456)

SELECT 1 FROM dual;
```

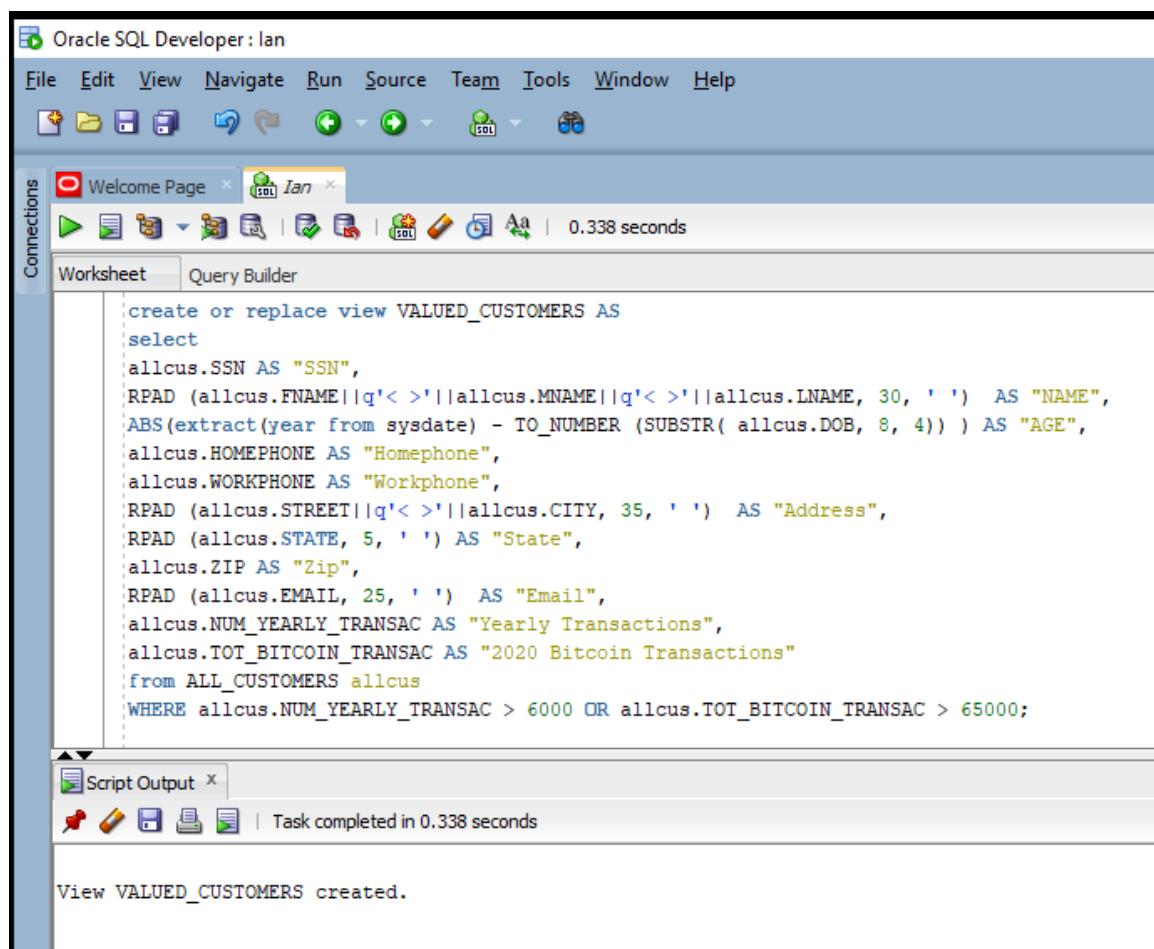
Script Output x

Task completed in 0.348 seconds

4 rows inserted.

**Finally, create view valued\_customers with the following attributes from above views and tables. As stated before, valued customers have > 65,000 total bitcoin transactions or > 6,000 yearly transactions.**

SSN
Name
Age
Home Phone
Work Phone
Address
Zip Code
Email <input type="checkbox"/> to be inputted later
State they live in
Total number of transactions the customer has done in a given year
Total amount (in bitcoin) of transactions the customer has done in a given year (you choose the year)



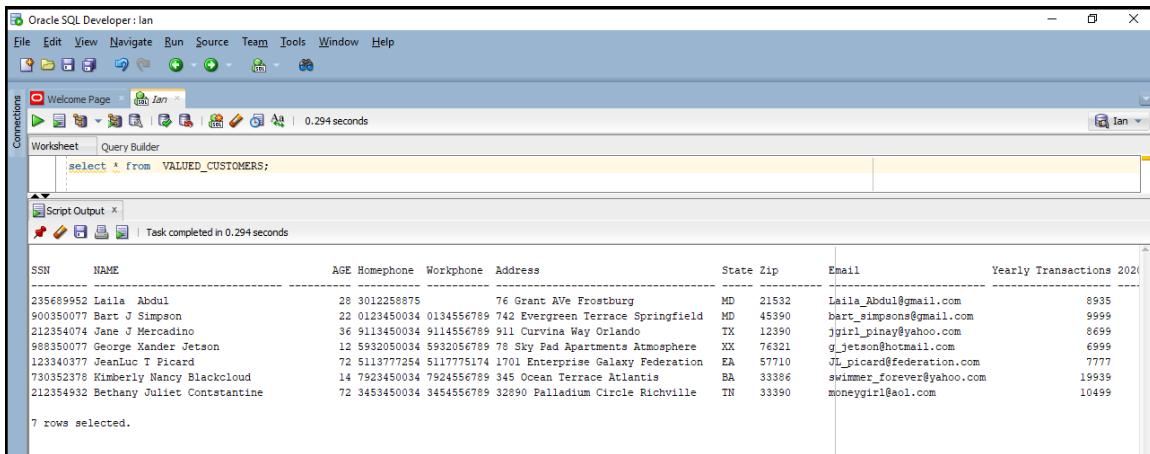
```

create or replace view VALUED_CUSTOMERS AS
select
    allcus.SSN AS "SSN",
    RPAD (allcus.FNAME||q'< >'||allcus.MNAME||q'< >'||allcus.LNAME, 30, ' ') AS "NAME",
    ABS(extract(year from sysdate) - TO_NUMBER (SUBSTR( allcus.DOB, 8, 4))) AS "AGE",
    allcus.HOMEPHONE AS "Homephone",
    allcus.WORKPHONE AS "Workphone",
    RPAD (allcus.STREET||q'< >'||allcus.CITY, 35, ' ') AS "Address",
    RPAD (allcus.STATE, 5, ' ') AS "State",
    allcus.ZIP AS "Zip",
    RPAD (allcus.EMAIL, 25, ' ') AS "Email",
    allcus.NUM_YEARLY_TRANSAC AS "Yearly Transactions",
    allcus.TOT_BITCOIN_TRANSAC AS "2020 Bitcoin Transactions"
from ALL_CUSTOMERS allcus
WHERE allcus.NUM_YEARLY_TRANSAC > 6000 OR allcus.TOT_BITCOIN_TRANSAC > 65000;

```

View VALUED\_CUSTOMERS created.

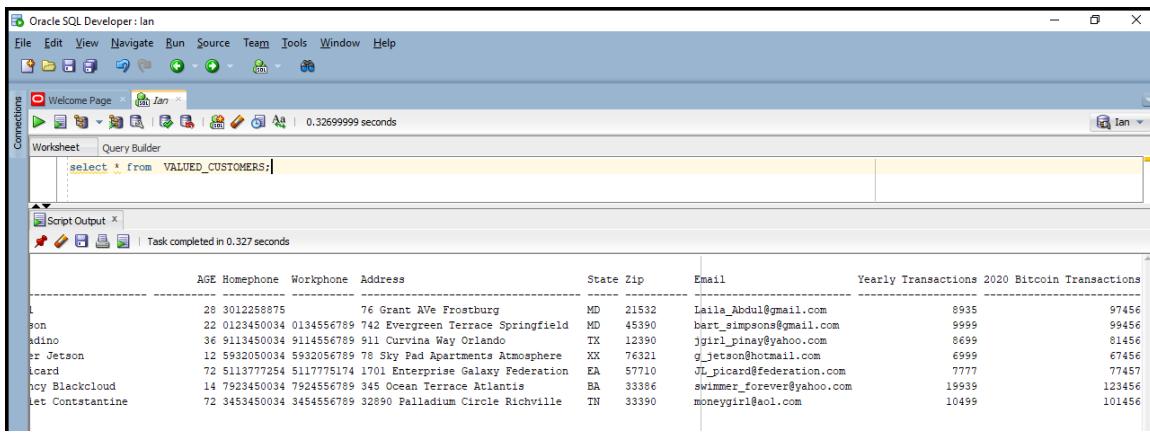
## Show contents of view valued\_customers



```
select * from VALUED_CUSTOMERS;
```

SSN	NAME	AGE	Homephone	Workphone	Address	State	Zip	Email	Yearly Transactions	2020 Bitcoin Transactions
235689952	Laila Abdul	28	30122358875		76 Grant Ave Frostburg	MD	21532	Laila_Abdul@gmail.com	8935	97456
600350077	Bert J Simpson	22	0123450034	0134556789	742 Evergreen Terrace Springfield	MD	45390	bart_simpons@gmail.com	9999	99456
212354074	Jane J Mercadino	36	9113450034	9114556789	911 Curvina Way Orlando	TX	12390	jgirl_pinay@yahoo.com	8699	81456
983350077	George Xander Jetson	12	5932050034	5932056789	78 Sky Pad Apartments Atmosphere	XX	76321	g_jetson@hotmail.com	6999	67456
123340377	JeanLuc T Picard	72	5113777254	5117775174	1701 Enterprise Galaxy Federation	EA	57710	JL_picard@federation.com	7777	77457
730352378	Kimberly Nancy Blackcloud	14	7923450034	7924556789	345 Ocean Terrace Atlantis	BA	33386	swimmer_forever@yahoo.com	19939	123456
212354932	Bethany Juliet Constantine	72	3453450034	3454556789	32890 Palladium Circle Richville	TN	33390	moneygirl@aol.com	10499	101456

7 rows selected.



```
select * from VALUED_CUSTOMERS;
```

SSN	NAME	AGE	Homephone	Workphone	Address	State	Zip	Email	Yearly Transactions	2020 Bitcoin Transactions
235689952	Laila Abdul	28	30122358875		76 Grant Ave Frostburg	MD	21532	Laila_Abdul@gmail.com	8935	97456
600350077	Bert J Simpson	22	0123450034	0134556789	742 Evergreen Terrace Springfield	MD	45390	bart_simpons@gmail.com	9999	99456
212354074	Jane J Mercadino	36	9113450034	9114556789	911 Curvina Way Orlando	TX	12390	jgirl_pinay@yahoo.com	8699	81456
983350077	George Xander Jetson	12	5932050034	5932056789	78 Sky Pad Apartments Atmosphere	XX	76321	g_jetson@hotmail.com	6999	67456
123340377	JeanLuc T Picard	72	5113777254	5117775174	1701 Enterprise Galaxy Federation	EA	57710	JL_picard@federation.com	7777	77457
730352378	Kimberly Nancy Blackcloud	14	7923450034	7924556789	345 Ocean Terrace Atlantis	BA	33386	swimmer_forever@yahoo.com	19939	123456
212354932	Bethany Juliet Constantine	72	3453450034	3454556789	32890 Palladium Circle Richville	TN	33390	moneygirl@aol.com	10499	101456

7 rows selected.

^^^^^^^^^^^^^^^^^^^^^^^^

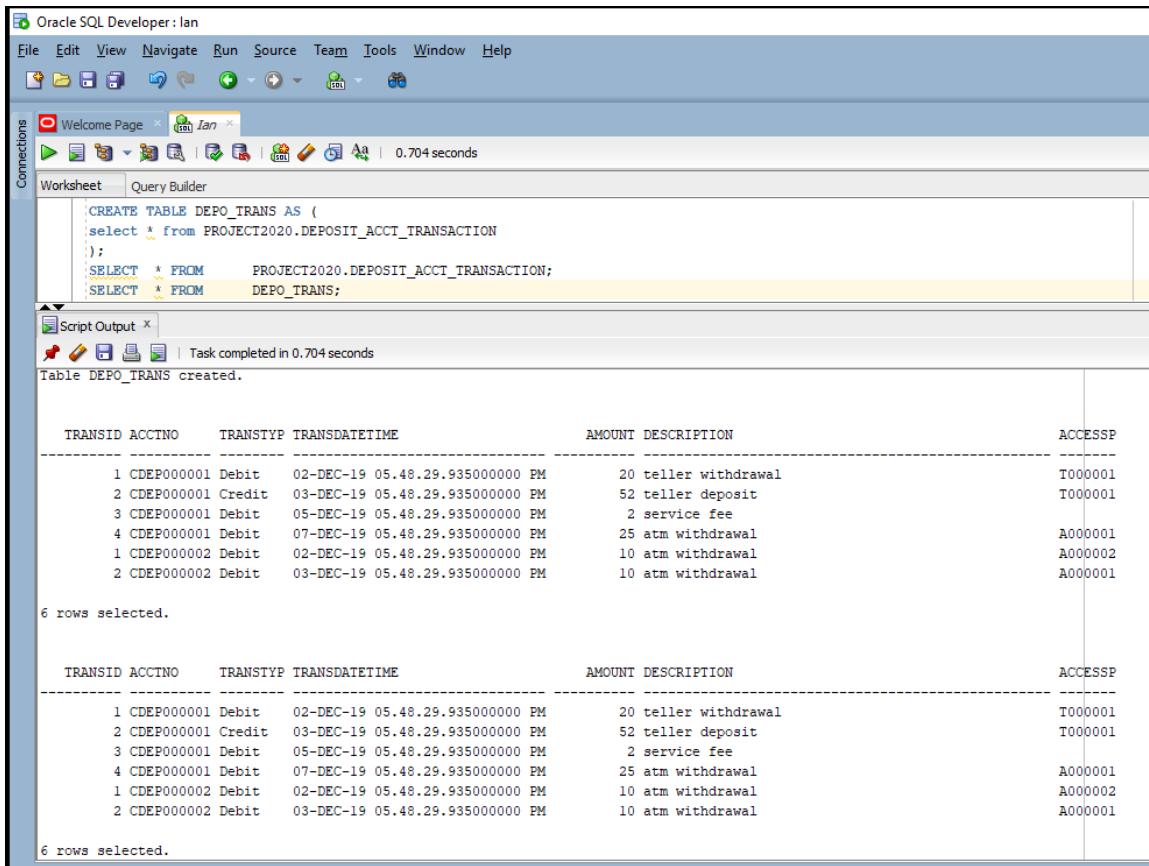
## CHAPTER 1E Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

E. Create view `statistics_by_branch` with the following attributes (Read only view):

Branch Id
Branch name
Year
Total deposit in that year for this location
Total number of transactions
Total number of employees at this branch

### Create table `depo_trans`



The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** Welcome Page, Ian
- Worksheet:** The code for creating the `DEPO_TRANS` table is pasted into the worksheet:CREATE TABLE DEPO\_TRANS AS (
 select \* from PROJECT2020.DEPOSIT\_ACCT\_TRANSACTION
);
SELECT \* FROM PROJECT2020.DEPOSIT\_ACCT\_TRANSACTION;
SELECT \* FROM DEPO\_TRANS;
- Script Output:** Task completed in 0.704 seconds. The message "Table DEPO\_TRANS created." is displayed.
- Data Output:** Two tables of transaction data are displayed. The first table shows 6 rows selected, and the second table shows 6 rows selected. Both tables have the following columns: TRANSID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSP.

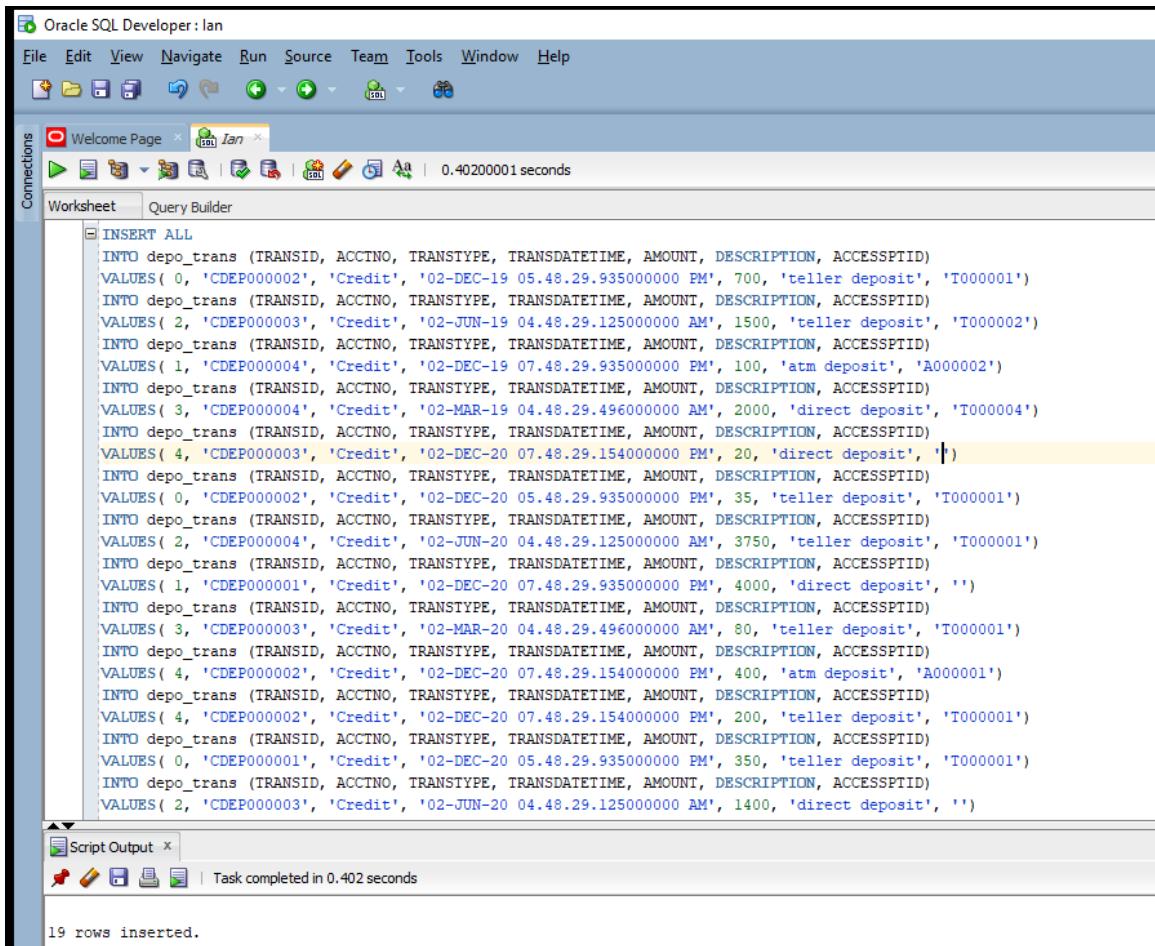
TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.93500000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.93500000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001

6 rows selected.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.93500000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.93500000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001

6 rows selected.

## Insert values into table depo\_trans

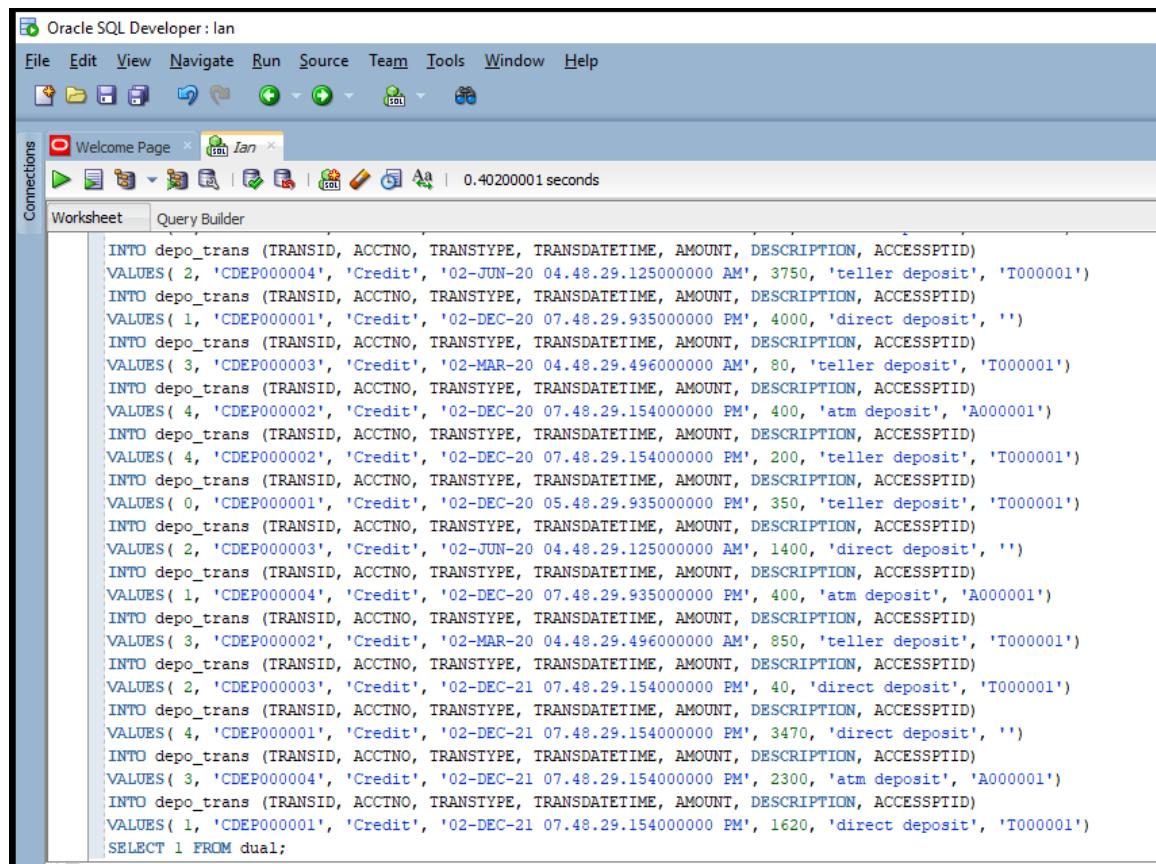


The screenshot shows the Oracle SQL Developer interface. The Worksheet tab contains a SQL script for inserting data into the 'depo\_trans' table. The Script Output tab shows the results of the execution.

```
INSERT ALL
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 0, 'CDEP000002', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 700, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 2, 'CDEP000003', 'Credit', '02-JUN-19 04.48.29.125000000 AM', 1500, 'teller deposit', 'T000002')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 1, 'CDEP000004', 'Credit', '02-DEC-19 07.48.29.935000000 PM', 100, 'atm deposit', 'A000002')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 3, 'CDEP000004', 'Credit', '02-MAR-19 04.48.29.496000000 AM', 2000, 'direct deposit', 'T000004')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 4, 'CDEP000003', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 20, 'direct deposit', '')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 0, 'CDEP000002', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 35, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 2, 'CDEP000004', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 3750, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 4000, 'direct deposit', '')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 3, 'CDEP000003', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 80, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 400, 'atm deposit', 'A000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 200, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 0, 'CDEP000001', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 350, 'teller deposit', 'T000001')
  INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES( 2, 'CDEP000003', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 1400, 'direct deposit', '')
```

19 rows inserted.

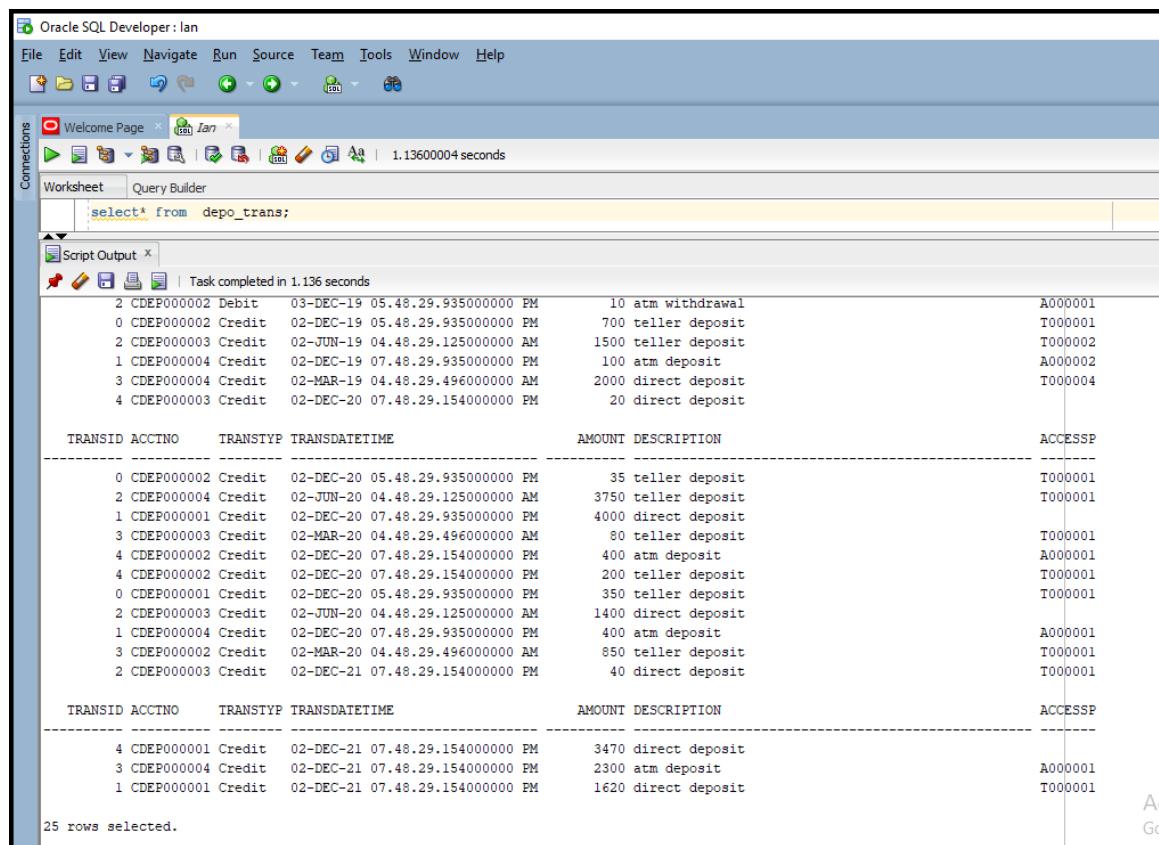
## Insert values into table depo\_trans



The screenshot shows the Oracle SQL Developer interface with a query window open. The title bar says 'Oracle SQL Developer : Ian'. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations. The connections pane shows a single connection named 'Ian'. The worksheet tab is selected, showing a SQL query. The query is a large block of INSERT statements into the 'depo\_trans' table, with a final SELECT 1 FROM dual; statement. The code uses placeholder values like 'CDEP00004' for transaction IDs and 'Credit' for transaction types. The SQL is color-coded for readability.

```
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP00004', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 3750, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP00001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 4000, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP00003', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 80, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP00002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 400, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP00002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 200, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 0, 'CDEP00001', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 350, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP00003', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 1400, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP00004', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 400, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP00002', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 850, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP00003', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 40, 'direct deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP00001', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 3470, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP00004', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 2300, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP00001', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 1620, 'direct deposit', 'T000001')
SELECT 1 FROM dual;
```

## Show contents of table depo\_trans



The screenshot shows the Oracle SQL Developer interface with a query window displaying the results of a SELECT statement on the 'depo\_trans' table. The results are presented in three horizontal sections, each with a different header. The first section has headers: TRANSID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSP. The second section has the same headers. The third section has the same headers. The data shows various transactions (Debit, Credit) with amounts ranging from 10 to 3470, and descriptions like 'atm withdrawal', 'teller deposit', 'atm deposit', 'direct deposit', and 'teller deposit'. The ACCESSP column consistently shows values like A00001, T00001, or T00002.

```
select* from depo_trans;
```

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
2	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A00001
0	CDEP000002	Credit	02-DEC-19 05.48.29.93500000 PM	700	teller deposit	T00001
2	CDEP000003	Credit	02-JUN-19 04.48.29.12500000 AM	1500	teller deposit	T00002
1	CDEP000004	Credit	02-DEC-19 07.48.29.93500000 PM	100	atm deposit	A00002
3	CDEP000004	Credit	02-MAR-19 04.48.29.49600000 AM	2000	direct deposit	T00004
4	CDEP000003	Credit	02-DEC-20 07.48.29.15400000 PM	20	direct deposit	

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
0	CDEP000002	Credit	02-DEC-20 05.48.29.93500000 PM	35	teller deposit	T00001
2	CDEP000004	Credit	02-JUN-20 04.48.29.12500000 AM	3750	teller deposit	T00001
1	CDEP000001	Credit	02-DEC-20 07.48.29.93500000 PM	4000	direct deposit	
3	CDEP000003	Credit	02-MAR-20 04.48.29.49600000 AM	80	teller deposit	T00001
4	CDEP000002	Credit	02-DEC-20 07.48.29.15400000 PM	400	atm deposit	A00001
4	CDEP000002	Credit	02-DEC-20 07.48.29.15400000 PM	200	teller deposit	T00001
0	CDEP000001	Credit	02-DEC-20 05.48.29.93500000 PM	350	teller deposit	T00001
2	CDEP000003	Credit	02-JUN-20 04.48.29.12500000 AM	1400	direct deposit	
1	CDEP000004	Credit	02-DEC-20 07.48.29.93500000 PM	400	atm deposit	A00001
3	CDEP000002	Credit	02-MAR-20 04.48.29.49600000 AM	850	teller deposit	T00001
2	CDEP000003	Credit	02-DEC-21 07.48.29.15400000 PM	40	direct deposit	T00001

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
4	CDEP000001	Credit	02-DEC-21 07.48.29.15400000 PM	3470	direct deposit	
3	CDEP000004	Credit	02-DEC-21 07.48.29.15400000 PM	2300	atm deposit	A00001
1	CDEP000001	Credit	02-DEC-21 07.48.29.15400000 PM	1620	direct deposit	T00001

25 rows selected.

## Show contents of two relevant feeder tables

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

```
select * from branch;
select * from BRANCH_EMP_TALLY;
```

Task completed in 0.551seconds

B_NAME	B_ST	B_CITY	B_STATE	B_ZIP	BRANCHPHON	BRANCH_MANAGER_ID	BRANCHURL	BRANCHID	CATEGORY
Harford Main	Jarrettsville Rd	Jarrettsville	Maryland	21084		12344	HarfMain.com	1004	
Frostburg Main	Braddock Rd	Frostburg	Maryland	21532		12345	FrostMain.com	1002	
Baltimore Main	Marisa Ct	Nottingham	Maryland	21236	4109313952	12343	BaltMain.com	1003	
Windsor Main	Corenotsville Rd	Corenotsville	Maryland	21674				1000	
Valleville Center	Chargburg Rd	Chargburg	Maryland	22694				1001	

Branch Name	Branch ID	Branch Headcount
Valleville Center	1001	3
Baltimore Main	1003	1
Frostburg Main	1002	2
Windsor Main	1000	2
Harford Main	1004	1

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

```
select
    dt.TRANSID+1000 AS "BRANCH_ID",
    branch.B_NAME AS "Branch Name",
    RPAD( SUBSTR(dt.TRANSDEATETIME,8,2), 5, ' ') AS "YEAR",
    SUM(dt.AMOUNT) AS "Yearly Total Deposits",
    COUNT(dt.TRANSID) AS "Yearly Transactions",
    BRANCH_EMP_TALLY."Branch Headcount"
from depo_trans dt, branch, BRANCH_EMP_TALLY
WHERE branch.BRANCHID = dt.TRANSID+1000 AND branch.B_NAME = BRANCH_EMP_TALLY."Branch Name"
GROUP BY SUBSTR(TRANSDEATETIME,8,2), TRANSID, branch.B_NAME, BRANCH_EMP_TALLY."Branch Headcount"
ORDER BY SUBSTR(TRANSDEATETIME,8,2) ASC;
```

Task completed in 0.634seconds

BRANCH ID	Branch Name	YEAR	Yearly Total Deposits	Yearly Transactions	Branch Headcount
1000	Windsor Main	19	700	1	2
1001	Valleville Center	19	130	3	3
1002	Frostburg Main	19	1562	3	2
1003	Baltimore Main	19	2002	2	1
1004	Harford Main	19	25	1	1
1000	Windsor Main	20	385	2	2
1001	Valleville Center	20	4400	2	3
1002	Frostburg Main	20	5150	2	2
1003	Baltimore Main	20	930	2	1
1004	Harford Main	20	620	3	1
1001	Valleville Center	21	1620	1	3

BRANCH ID	Branch Name	YEAR	Yearly Total Deposits	Yearly Transactions	Branch Headcount
1002	Frostburg Main	21	40	1	2
1003	Baltimore Main	21	2300	1	1
1004	Harford Main	21	3470	1	1

**Finally, create view statistics\_by\_Branch with the following attributes  
(Read only view) then show contents:**

Branch Id
Branch name
Year
Total deposit in that year for this location
Total number of transactions
Total number of employees at this branch

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page x Jan x 0.634 seconds

Worksheet Query Builder

```

select
    dt.TRANSID+1000 AS "Branch ID",
    branch.B_NAME AS "Branch Name",
    RPAD(SUBSTR(dt.TRANSDEATETIME,8,2), 5, ' ') AS "YEAR",
    SUM(dt.AMOUNT) AS "Yearly Total Deposits",
    COUNT(dt.TRANSID) AS "Yearly Transactions",
    BRANCH_EMP_TALLY."Branch Headcount"
    from depo_trans dt, branch, BRANCH_EMP_TALLY
    WHERE branch.BRANCHID = dt.TRANSID+1000 AND branch.B_NAME = BRANCH_EMP_TALLY."Branch Name"
    GROUP BY SUBSTR(TRANSDEATETIME,8,2), TRANSID, branch.B_NAME, BRANCH_EMP_TALLY."Branch Headcount"
    ORDER BY SUBSTR(TRANSDEATETIME,8,2) ASC;

```

Script Output x Task completed in 0.634 seconds

Branch ID	Branch Name	Year	Yearly Total Deposits	Yearly Transactions	Branch Headcount
1002 Frostburg Main	19	1562	3	2	
1003 Baltimore Main	19	2002	2	1	
1004 Harford Main	19	25	1	1	
1000 Windsor Main	20	385	2	2	
1001 Valleville Center	20	4400	2	3	
1002 Frostburg Main	20	5150	2	2	
1003 Baltimore Main	20	930	2	1	
1004 Harford Main	20	620	3	1	
1001 Valleville Center	21	1620	1	3	

Branch ID	Branch Name	Year	Yearly Total Deposits	Yearly Transactions	Branch Headcount
1002 Frostburg Main	21	40	1	2	
1003 Baltimore Main	21	2300	1	1	
1004 Harford Main	21	3470	1	1	

14 rows selected.

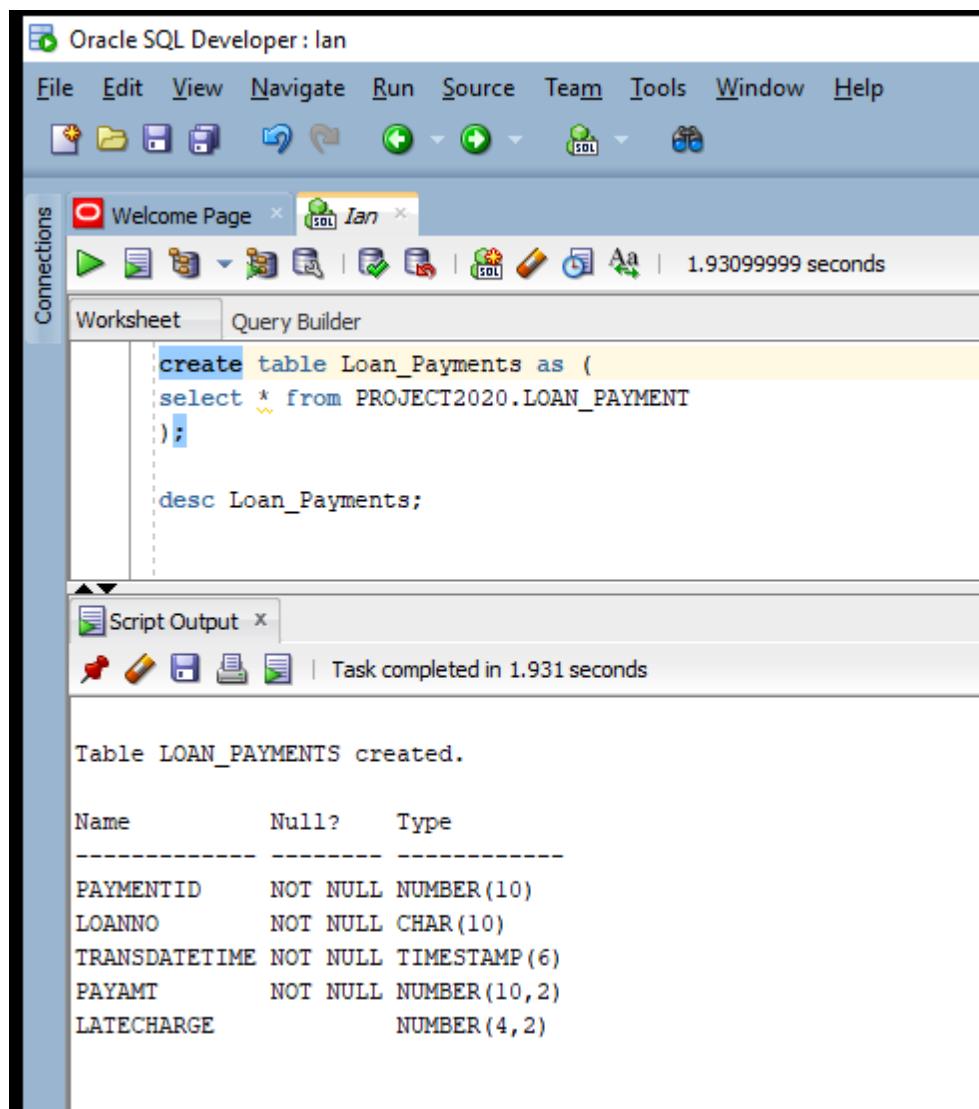
^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 1F,G Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

F,G. Create two more views that can be used by customers. (Make sure it is useful to the customers. You will be graded based on the usefulness of the views)

### Create views loan\_summary and loan\_payment\_summary



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
create table Loan_Payments as (
  select * from PROJECT2020.LOAN_PAYMENT
);

desc Loan_Payments;
```

Below the worksheet, the 'Script Output' tab shows the results of the execution:

```
Table LOAN_PAYMENTS created.

Name      Null?    Type
-----
PAYMENTID NOT NULL NUMBER(10)
LOANNO     NOT NULL CHAR(10)
TRANSDATETIME NOT NULL TIMESTAMP(6)
PAYAMT     NOT NULL NUMBER(10,2)
LATECHARGE          NUMBER(4,2)
```

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
INSERT ALL
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '05-FEB-19 12.21.16.836432000 PM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '05-FEB-19 12.21.16.836432000 PM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '07-APR-19 09.21.16.836432000 PM', 427.15, 20.00)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '05-MAY-19 09.21.16.836432000 AM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '04-JUN-19 12.21.16.836432000 PM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '05-JUL-19 10.21.16.836432000 PM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '05-AUG-19 06.21.16.836432000 PM', 427.15, NULL)
  INTO Loan_Payments (PAYMENTID, LOANNO, TRANSDATETIME, PAYAMT, LATECHARGE)
  VALUES(3, 'CL0000003', '03-SEP-19 04.21.16.836432000 PM', 427.15, NULL)
  SELECT 1 FROM dual;
```

Script Output

Task completed in 1.012 seconds

8 rows inserted.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
CREATE OR REPLACE VIEW LOAN_NUMBER AS
SELECT LOANNO, AMTBORROWED, MINMOPAY
CREATE OR REPLACE VIEW TOTAL_PAYMENTS AS
SELECT SUM(PAYAMT) AS "Total Paid"
CREATE OR REPLACE VIEW LOAN_DETAILS AS
SELECT DESCRIP, INTRATE, DURATION
FROM PROJECT2020.LOAN WHERE LOANNO = 'CL0000003';
FROM Loan_Payments WHERE LOANNO = 'CL0000003';
FROM PROJECT2020.LOAN_PRODUCT WHERE LOANPRODID = 'L2';
```

Script Output

Task completed in 0.635 seconds

View LOAN\_NUMBER created.

View TOTAL\_PAYMENTS created.

View LOAN\_DETAILS created.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
select * from LOAN_NUMBER;
select * from TOTAL_PAYMENTS;
select * from LOAN_DETAILS;
```

Script Output Task completed in 0.514 seconds

LOANNO	AMTBORROWED	MINMOPAY
CL0000003	18000	427.15

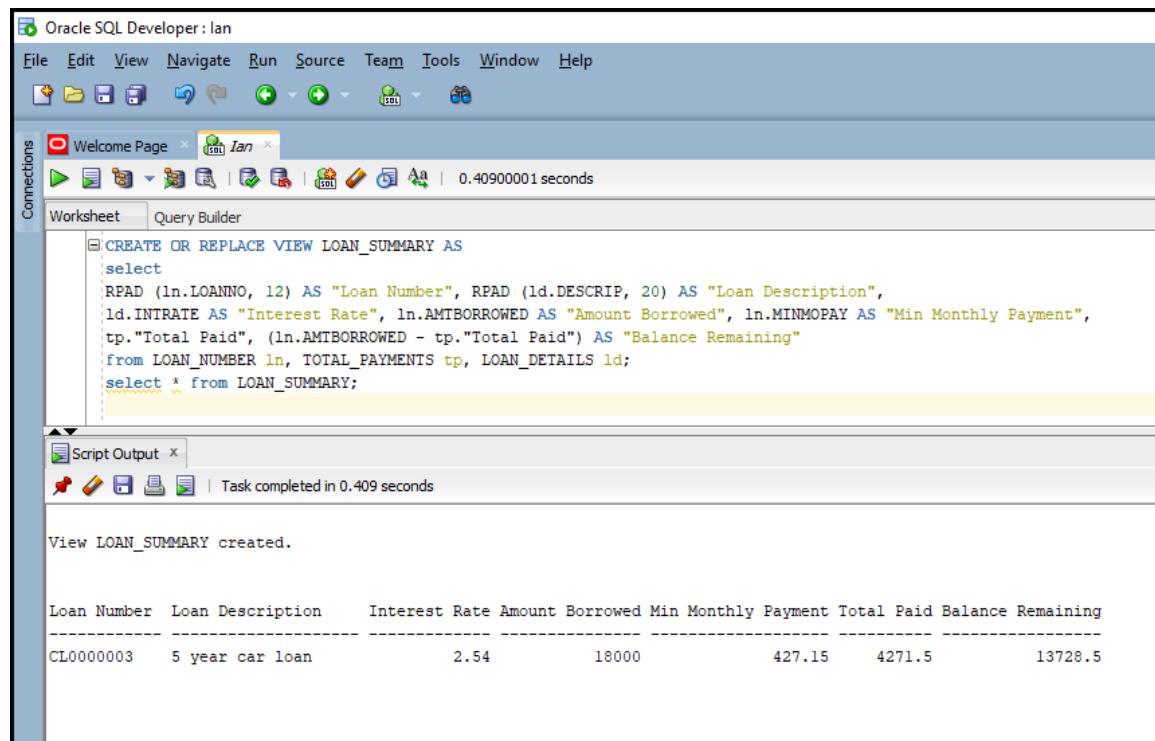
Total Paid

4271.5
--------

DESCRIP

INTRATE	DURATION
2.54	60
5 year car loan	

## Create and display view loan\_summary



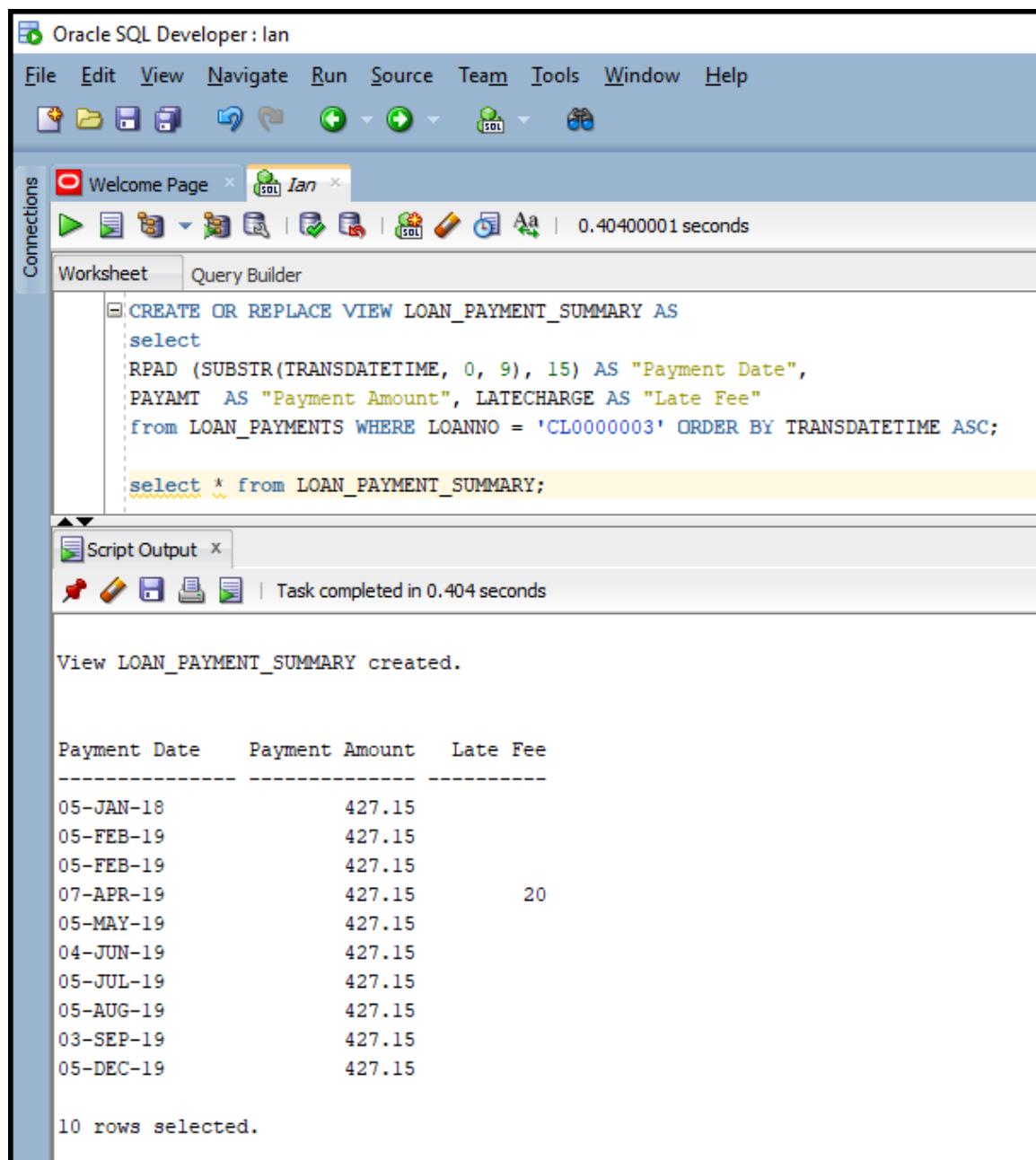
The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for connection, schema browser, and code editor. The Connections panel shows a connection named "Ian". The Worksheet tab is active, displaying the following SQL code:

```
CREATE OR REPLACE VIEW LOAN_SUMMARY AS
  select
    RPAD (ln.LOANNO, 12) AS "Loan Number", RPAD (ld.DESCRIPTION, 20) AS "Loan Description",
    ldINTRATE AS "Interest Rate", lnAMTBORROWED AS "Amount Borrowed", lnMINMOPAY AS "Min Monthly Payment",
    tp.Total Paid", (lnAMTBORROWED - tp.Total Paid) AS "Balance Remaining"
  from LOAN_NUMBER ln, TOTAL_PAYMENTS tp, LOAN_DETAILS ld;
  select * from LOAN_SUMMARY;
```

The Script Output tab shows the message "View LOAN\_SUMMARY created." and the output of the query:

Loan Number	Loan Description	Interest Rate	Amount Borrowed	Min Monthly Payment	Total Paid	Balance Remaining
CL0000003	5 year car loan	2.54	18000	427.15	4271.5	13728.5

## Create and display view loan\_payment\_summary



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the creation of a view:

```
CREATE OR REPLACE VIEW LOAN_PAYMENT_SUMMARY AS
select
  RPAD (SUBSTR(TRANSDATETIME, 0, 9), 15) AS "Payment Date",
  PAYAMT AS "Payment Amount", LATECHARGE AS "Late Fee"
from LOAN_PAYMENTS WHERE LOANNO = 'CL0000003' ORDER BY TRANSDATETIME ASC;

select * from LOAN_PAYMENT_SUMMARY;
```

The 'Script Output' tab shows the confirmation of view creation and the resulting data:

```
View LOAN_PAYMENT_SUMMARY created.

Payment Date      Payment Amount      Late Fee
-----          -----
05-JAN-18          427.15
05-FEB-19          427.15
05-FEB-19          427.15
07-APR-19          427.15          20
05-MAY-19          427.15
04-JUN-19          427.15
05-JUL-19          427.15
05-AUG-19          427.15
03-SEP-19          427.15
05-DEC-19          427.15

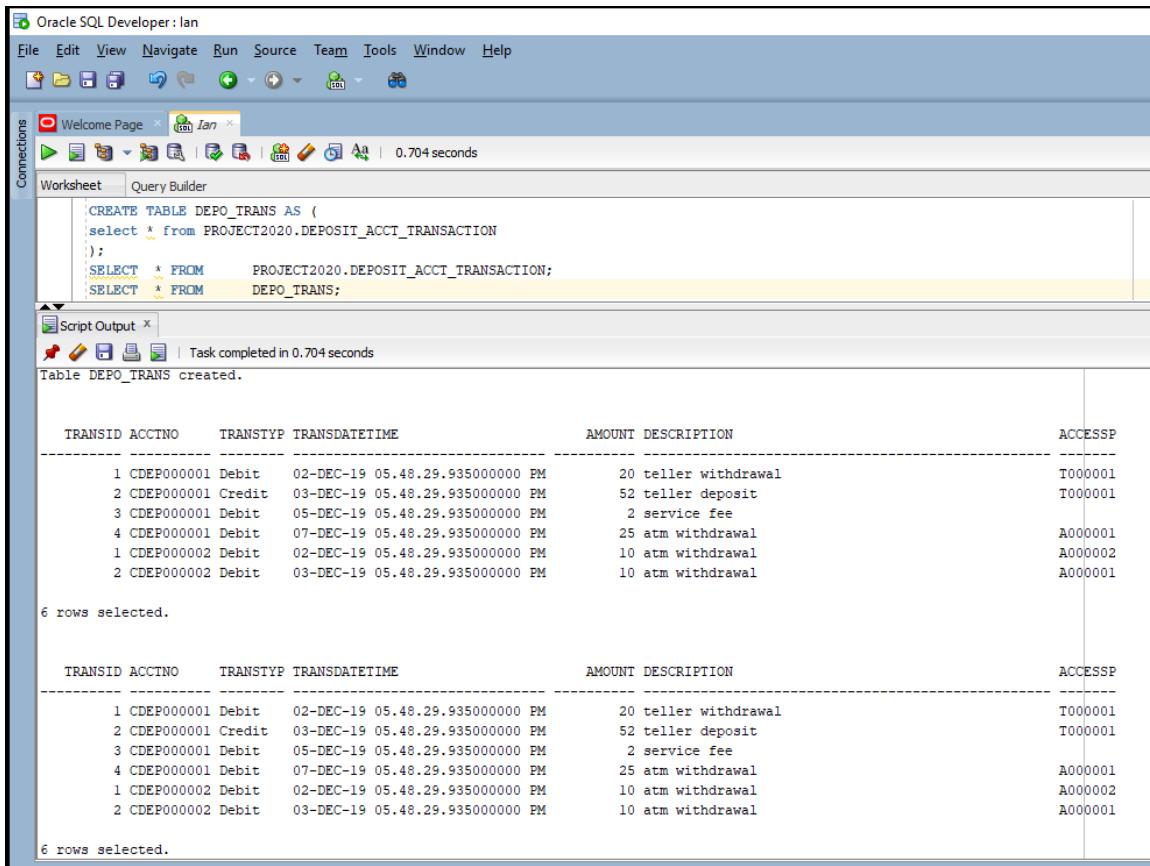
10 rows selected.
```

^^^^^^^^^^^^^^^^^^^^

## CHAPTER 1H,I Starts here

^^^^^^^^^^^^^^^^^^^^

H,I. Create two more views that can be used by management. (Make sure it is useful for managerial decisions)



The screenshot shows the Oracle SQL Developer interface with a query window containing the following SQL code:

```
CREATE TABLE DEPO_TRANS AS
select * from PROJECT2020.DEPOSIT_ACCT_TRANSACTION
);
SELECT * FROM PROJECT2020.DEPOSIT_ACCT_TRANSACTION;
SELECT * FROM DEPO_TRANS;
```

The "Script Output" pane shows the results of the execution:

Table DEPO\_TRANS created.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.935000000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000001

6 rows selected.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.935000000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000001

6 rows selected.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.40200001 seconds

Worksheet Query Builder

INSERT ALL

```
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 0, 'CDEP000002', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 700, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000003', 'Credit', '02-JUN-19 04.48.29.125000000 AM', 1500, 'teller deposit', 'T000002')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP000004', 'Credit', '02-DEC-19 07.48.29.935000000 PM', 100, 'atm deposit', 'A000002')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP000004', 'Credit', '02-MAR-19 04.48.29.496000000 AM', 2000, 'direct deposit', 'T000004')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000003', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 20, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 0, 'CDEP000002', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 35, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000004', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 3750, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 4000, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP000003', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 80, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 400, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 200, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 0, 'CDEP000001', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 350, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000003', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 1400, 'direct deposit', '')
```

Script Output Task completed in 0.402 seconds

19 rows inserted.

Oracle SQL Developer : Ian

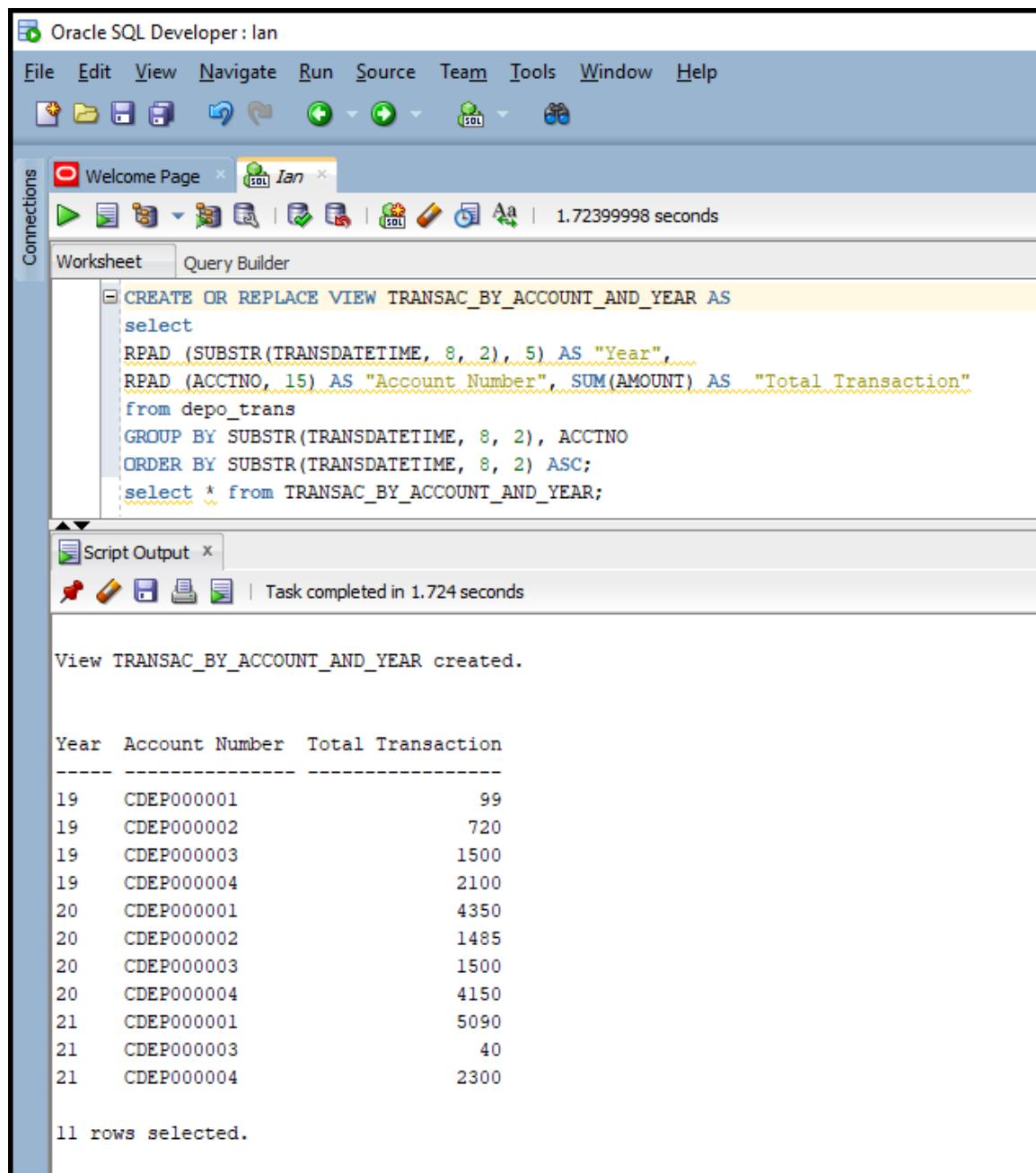
File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.40200001 seconds

Worksheet Query Builder

```
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000004', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 3750, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 4000, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP000003', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 80, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 400, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000002', 'Credit', '02-DEC-20 07.48.29.154000000 PM', 200, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 0, 'CDEP000001', 'Credit', '02-DEC-20 05.48.29.935000000 PM', 350, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000003', 'Credit', '02-JUN-20 04.48.29.125000000 AM', 1400, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP000004', 'Credit', '02-DEC-20 07.48.29.935000000 PM', 400, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP000002', 'Credit', '02-MAR-20 04.48.29.496000000 AM', 850, 'teller deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 2, 'CDEP000003', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 40, 'direct deposit', 'T000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 4, 'CDEP000001', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 3470, 'direct deposit', '')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 3, 'CDEP000004', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 2300, 'atm deposit', 'A000001')
INTO depo_trans (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES( 1, 'CDEP000001', 'Credit', '02-DEC-21 07.48.29.154000000 PM', 1620, 'direct deposit', 'T000001')
SELECT 1 FROM dual;
```

## Create and display view transac\_by\_account\_and\_year



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the creation of a view named 'TRANSAC\_BY\_ACCOUNT\_AND\_YEAR'. The code is as follows:

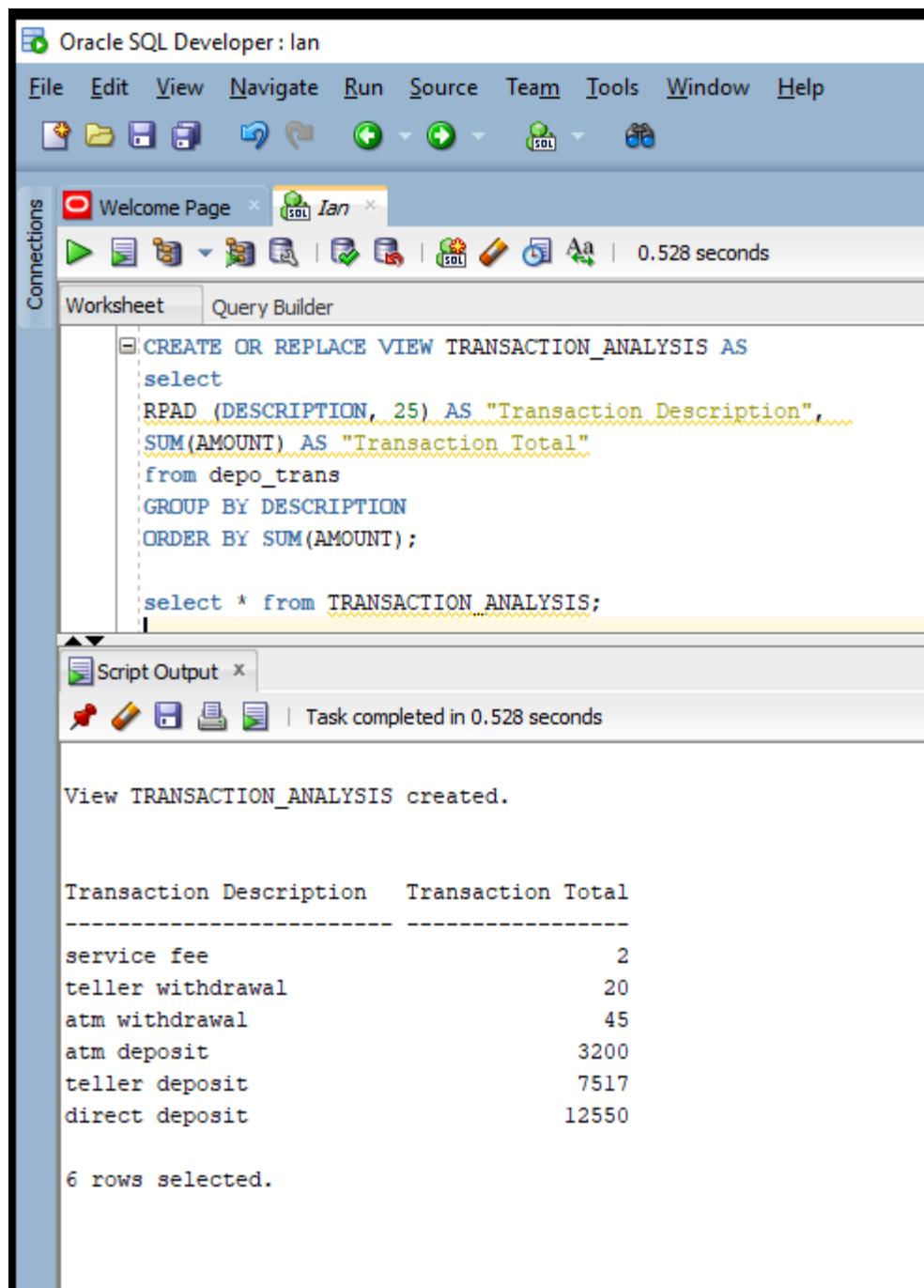
```
CREATE OR REPLACE VIEW TRANSAC_BY_ACCOUNT_AND_YEAR AS
select
  RPAD (SUBSTR(TRANSDATETIME, 8, 2), 5) AS "Year",
  RPAD (ACCTNO, 15) AS "Account Number", SUM(AMOUNT) AS "Total Transaction"
  from depo_trans
  GROUP BY SUBSTR(TRANSDATETIME, 8, 2), ACCTNO
  ORDER BY SUBSTR(TRANSDATETIME, 8, 2) ASC;
select * from TRANSAC_BY_ACCOUNT_AND_YEAR;
```

Below the code, the message 'View TRANSAC\_BY\_ACCOUNT\_AND\_YEAR created.' is displayed. The 'Script Output' tab shows the results of the query:

Year	Account Number	Total Transaction
19	CDEP000001	99
19	CDEP000002	720
19	CDEP000003	1500
19	CDEP000004	2100
20	CDEP000001	4350
20	CDEP000002	1485
20	CDEP000003	1500
20	CDEP000004	4150
21	CDEP000001	5090
21	CDEP000003	40
21	CDEP000004	2300

11 rows selected.

## Create and display view transaction\_analysis



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the creation of a view:

```
CREATE OR REPLACE VIEW TRANSACTION_ANALYSIS AS
  select
    RPAD (DESCRIPTION, 25) AS "Transaction Description",
    SUM(AMOUNT) AS "Transaction Total"
  from depo_trans
  GROUP BY DESCRIPTION
  ORDER BY SUM(AMOUNT);

  select * from TRANSACTION_ANALYSIS;
```

The 'Script Output' tab shows the results of the execution:

```
View TRANSACTION_ANALYSIS created.

Transaction Description      Transaction Total
-----  -----
service fee                      2
teller withdrawal                 20
atm withdrawal                     45
atm deposit                        3200
teller deposit                     7517
direct deposit                     12550

6 rows selected.
```

## **CHAPTER 2**

^^^^^^^^^^^^^^^^^^^^^^^^

### **CHAPTER 2A Starts here**

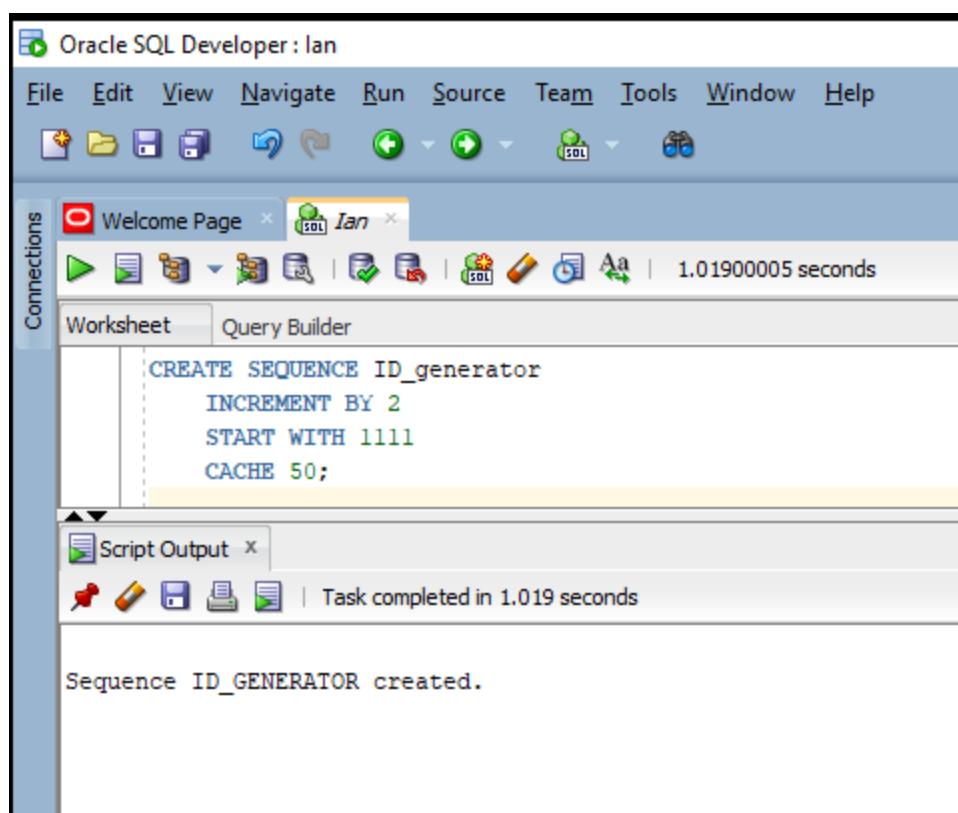
^^^^^^^^^^^^^^^^^^^^^^^^

A. Create a sequence called ID\_generator to be used for Account ID.

Start with 1111

Generate only odd numbers for security

Cache 50 numbers at a time



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations. The Connections sidebar shows "Welcome Page" and "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The Worksheet tab contains the following SQL code:

```
CREATE SEQUENCE ID_generator
INCREMENT BY 2
START WITH 1111
CACHE 50;
```

The Script Output tab shows the result of the execution:

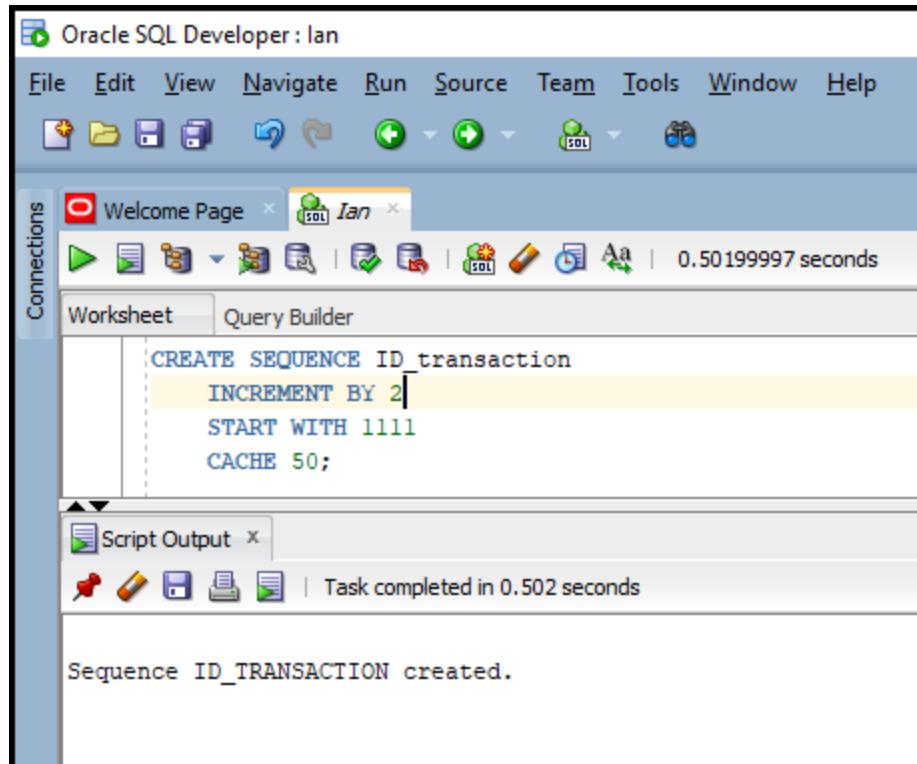
```
Sequence ID_GENERATOR created.
```

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 2B Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

B. Create a sequence to be used for the Transaction ID. (Make your own assumption).



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE SEQUENCE ID_transaction
  INCREMENT BY 2
  START WITH 1111
  CACHE 50;
```

The code is highlighted in yellow. Below the worksheet, the 'Script Output' tab shows the result of the execution:

```
Sequence ID_TRANSACTION created.
```

## **CHAPTER 3**

**Write Subprograms with exception handling: (make sure your subprograms have appropriate exception handling).**

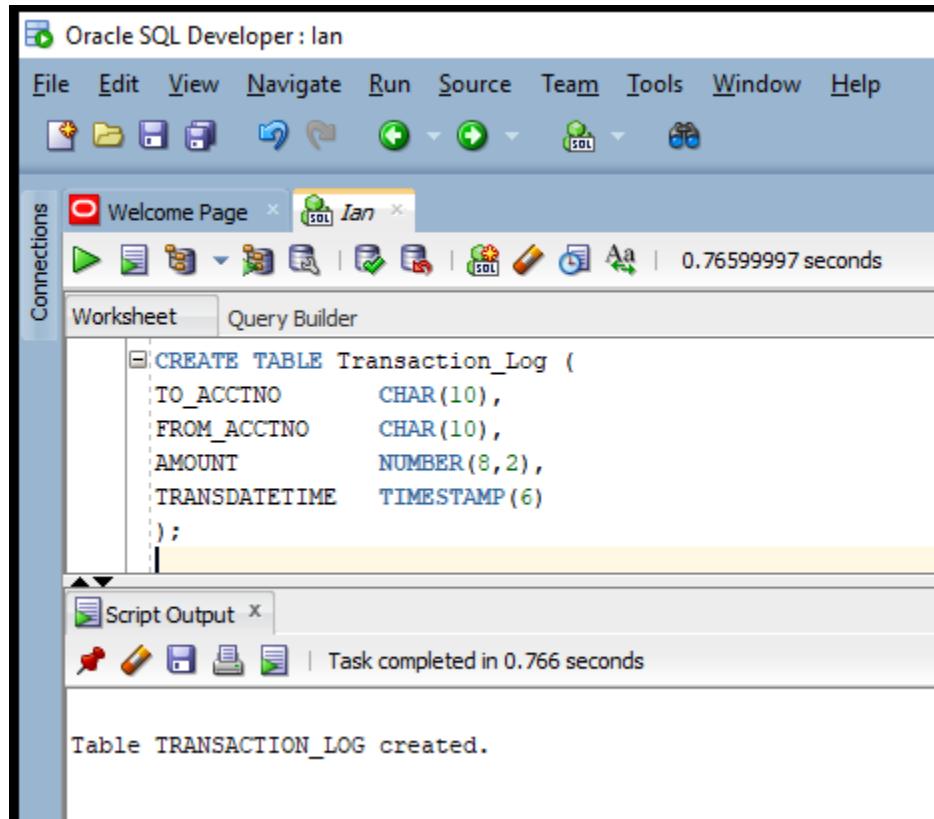
^^^^^^^^^^^^^^^^^^^^^^^^^^^^

## **CHAPTER 3A Starts here**

^^^^^^^^^^^^^^^^^^^^^^^^

A. Procedure to transfer \$x from one account to another account. You will pass the amount of transfer, and two account numbers. Also, write the FROM account number, to TO account number, the amount and the date of transaction into a Transaction\_Log file (Create this table). Show your work. You may use CREDIT\_ACCOUNT table for this procedure.

### **Create empty table transaction\_log with stated attributes**



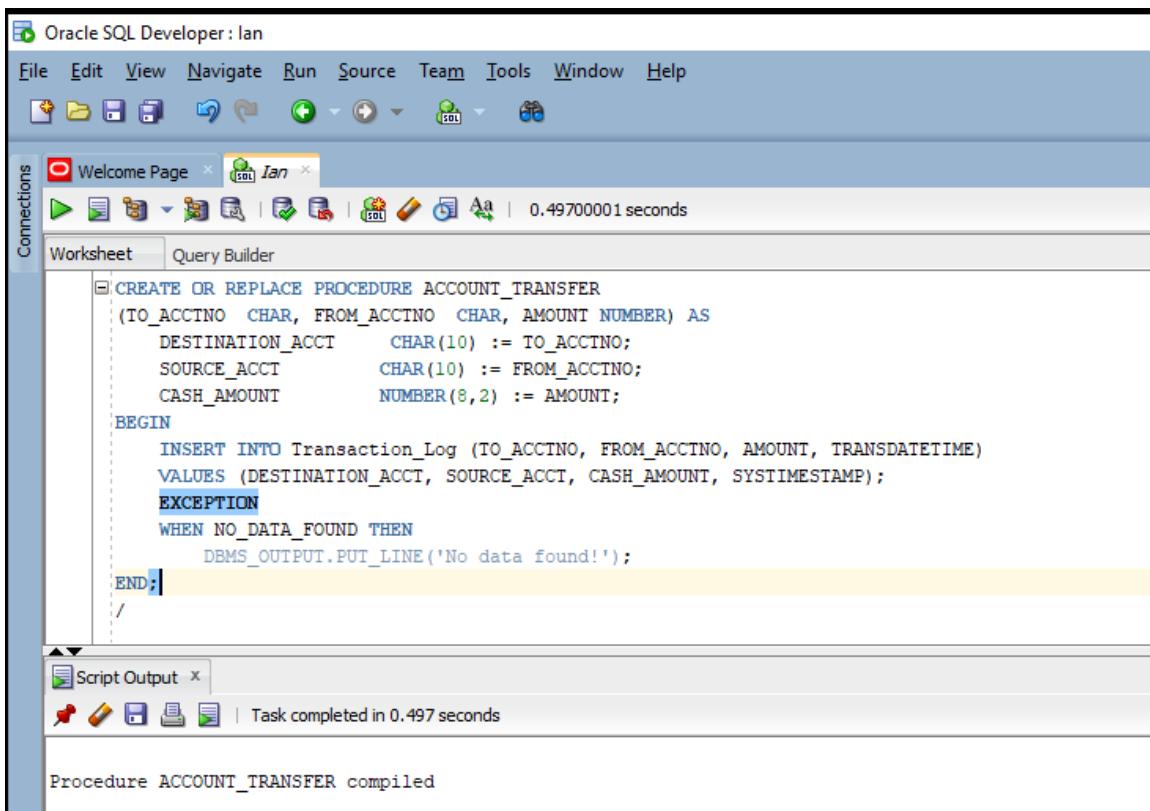
The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL code for creating the 'Transaction\_Log' table:

```
CREATE TABLE Transaction_Log (
    TO_ACCTNO      CHAR(10),
    FROM_ACCTNO    CHAR(10),
    AMOUNT         NUMBER(8,2),
    TRANSDATETIME  TIMESTAMP(6)
);
```

The 'Script Output' tab at the bottom shows the confirmation message:

```
Table TRANSACTION_LOG created.
```

## Create procedure to load attribute values into table transaction\_log with exception handing

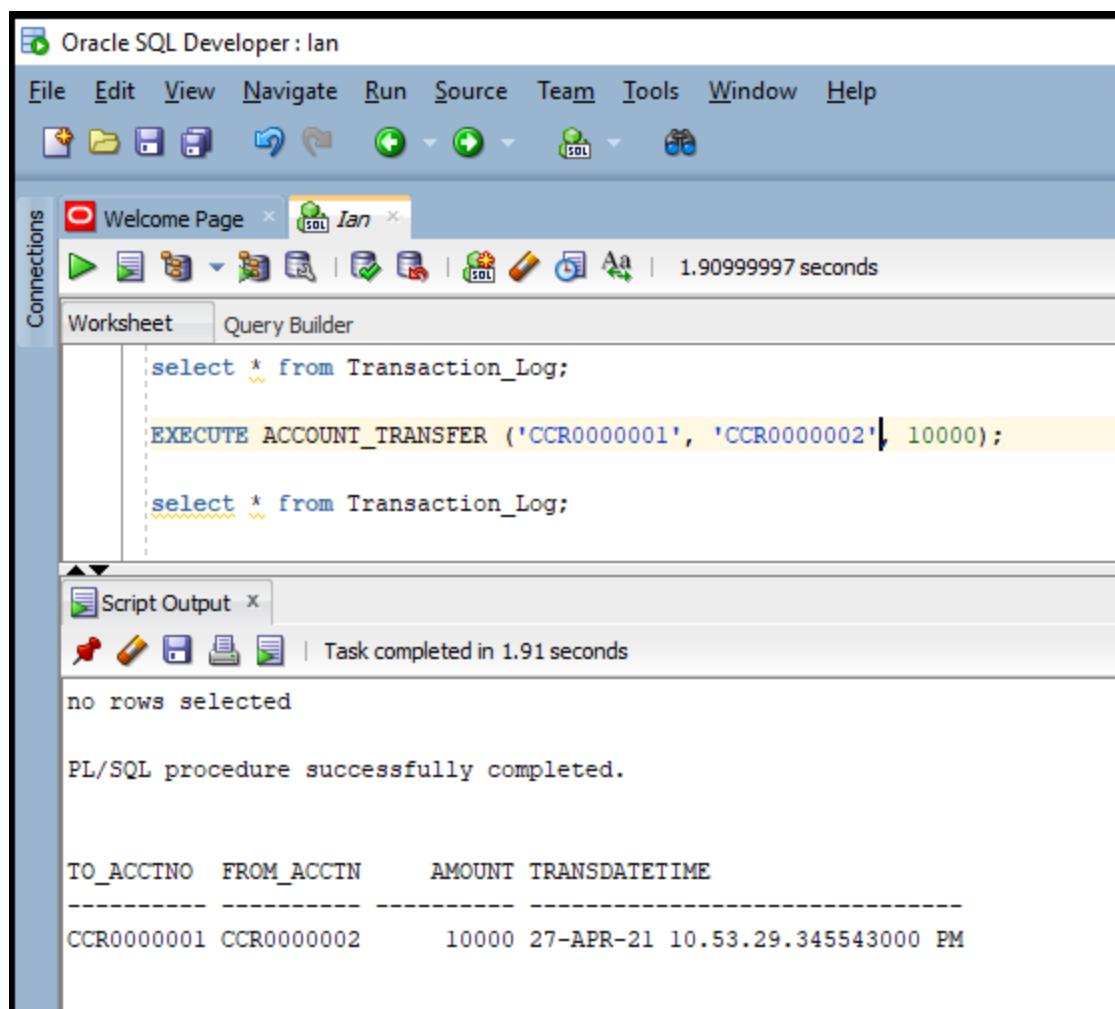


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE ACCOUNT_TRANSFER
  (TO_ACCTNO CHAR, FROM_ACCTNO CHAR, AMOUNT NUMBER) AS
  DESTINATION_ACCT      CHAR(10) := TO_ACCTNO;
  SOURCE_ACCT           CHAR(10) := FROM_ACCTNO;
  CASH_AMOUNT           NUMBER(8,2) := AMOUNT;
BEGIN
  INSERT INTO Transaction_Log (TO_ACCTNO, FROM_ACCTNO, AMOUNT, TRANSDATETIME)
  VALUES (DESTINATION_ACCT, SOURCE_ACCT, CASH_AMOUNT, SYSTIMESTAMP);
  EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data found!');
END;
/
```

The code is syntax-highlighted, with keywords in blue and identifiers in black. The 'Script Output' tab at the bottom shows the message: 'Procedure ACCOUNT\_TRANSFER compiled'.

## Execute procedure to load attribute values into table transaction\_log



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and database navigation. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The script area contains the following PL/SQL code:

```
select * from Transaction_Log;

EXECUTE ACCOUNT_TRANSFER ('CCR0000001', 'CCR0000002', 10000);

select * from Transaction_Log;
```

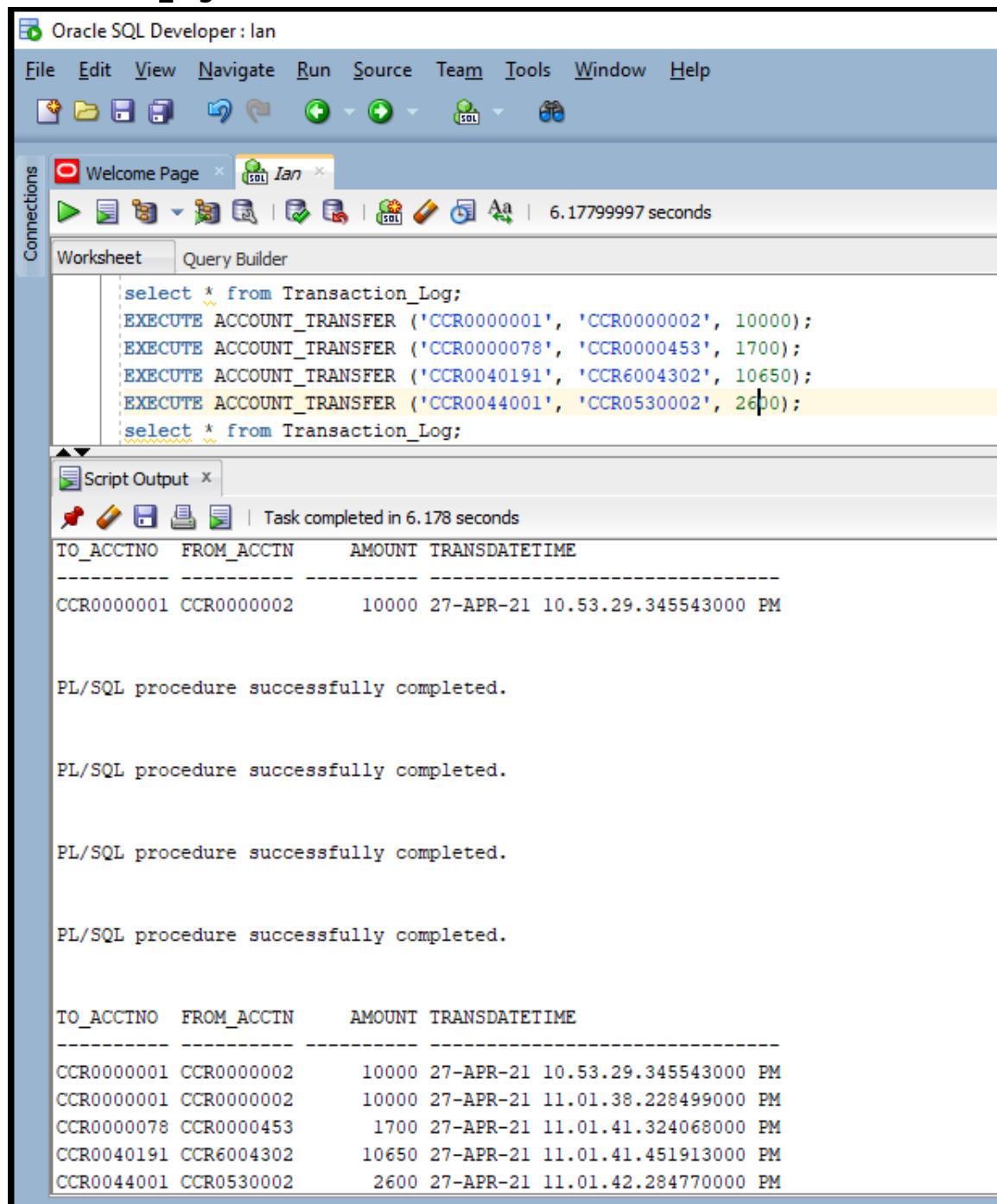
The "Script Output" window below shows the execution results:

```
no rows selected

PL/SQL procedure successfully completed.

TO_ACCTNO  FROM_ACCTN      AMOUNT TRANSDATETIME
-----  -----
CCR0000001  CCR0000002      10000 27-APR-21 10.53.29.345543000 PM
```

**Execute procedure four more times to load attribute values into table transaction\_log**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Welcome Page Ian 6.17799997 seconds

Connections Worksheet Query Builder

```
select * from Transaction_Log;
EXECUTE ACCOUNT_TRANSFER ('CCR0000001', 'CCR0000002', 10000);
EXECUTE ACCOUNT_TRANSFER ('CCR0000078', 'CCR0000453', 1700);
EXECUTE ACCOUNT_TRANSFER ('CCR0040191', 'CCR6004302', 10650);
EXECUTE ACCOUNT_TRANSFER ('CCR0044001', 'CCR0530002', 2600);
select * from Transaction_Log;
```

Script Output Task completed in 6.178 seconds

TO_ACCTNO	FROM_ACCTN	AMOUNT	TRANSDATETIME
CCR0000001	CCR0000002	10000	27-APR-21 10.53.29.345543000 PM

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

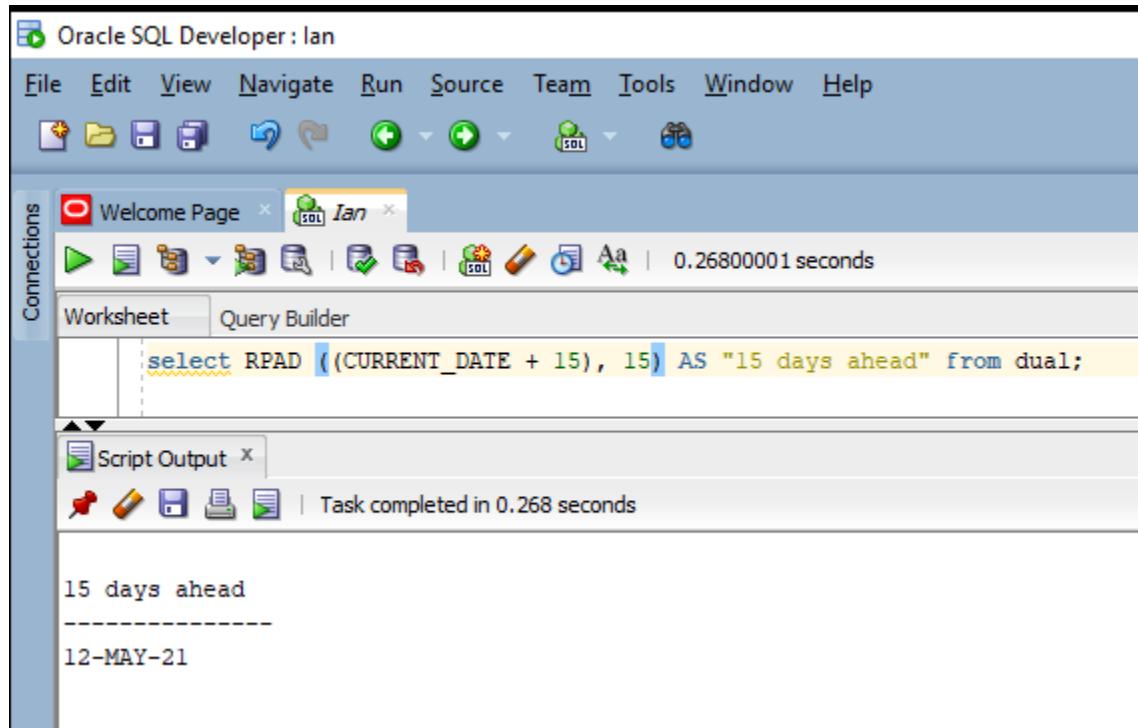
TO_ACCTNO	FROM_ACCTN	AMOUNT	TRANSDATETIME
CCR0000001	CCR0000002	10000	27-APR-21 10.53.29.345543000 PM
CCR0000001	CCR0000002	10000	27-APR-21 11.01.38.228499000 PM
CCR0000078	CCR0000453	1700	27-APR-21 11.01.41.324068000 PM
CCR0040191	CCR6004302	10650	27-APR-21 11.01.41.451913000 PM
CCR0044001	CCR0530002	2600	27-APR-21 11.01.42.284770000 PM

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3B Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

B. Create a subprogram called **Birthday\_sub** that accepts today's date as default and writes the first name, last name, and address of a customer whose birth date (day and month only) is 15 days from today's date. Write them into a file (create a table called **B\_C\_File**). You may add additional records to your customer table.



The screenshot shows the Oracle SQL Developer interface. The Worksheet pane contains the following SQL code:

```
select RPAD ((CURRENT_DATE + 15), 15) AS "15 days ahead" from dual;
```

The Script Output pane shows the results of the query:

```
15 days ahead
-----
12-MAY-21
```

## Display contents of table all\_customers and attribute data types

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
SELECT * FROM ALL_CUSTOMERS;
DESC ALL_CUSTOMERS;
```

Script Output x | Task completed in 1.6 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD	21532	33294
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30124
9900000004	900350077	Bart	J	Simpson	18-NOV-1989	742 Evergreen Terrace	Springfield	MO	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	30-APR-2000	345 Cave Stone Road	Bedrock	NV	48286	30034
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Lane	Waterflood	NE	78390	92734
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX	12390	91134
9900000009	988350077	George	Xander	Jetson	14-JUL-2009	78 Sky Pad Apartments	Atmosphere	XX	76321	59324
9900000010	123340377	JeanLuc	I	Picard	06-MAR-1949	1701 Enterprise Galaxy	Federation	EA	57710	51134
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	79234
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	34534

12 rows selected.

Name	Null?	Type
CUSTID		NUMBER(10)
SSN		CHAR(9)
FNAME		VARCHAR2(20)
MNAME		VARCHAR2(20)

Activate Windows  
Go to Settings to activate Windows.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
SELECT * FROM ALL_CUSTOMERS;
DESC ALL_CUSTOMERS;
```

Script Output x | Task completed in 1.6 seconds

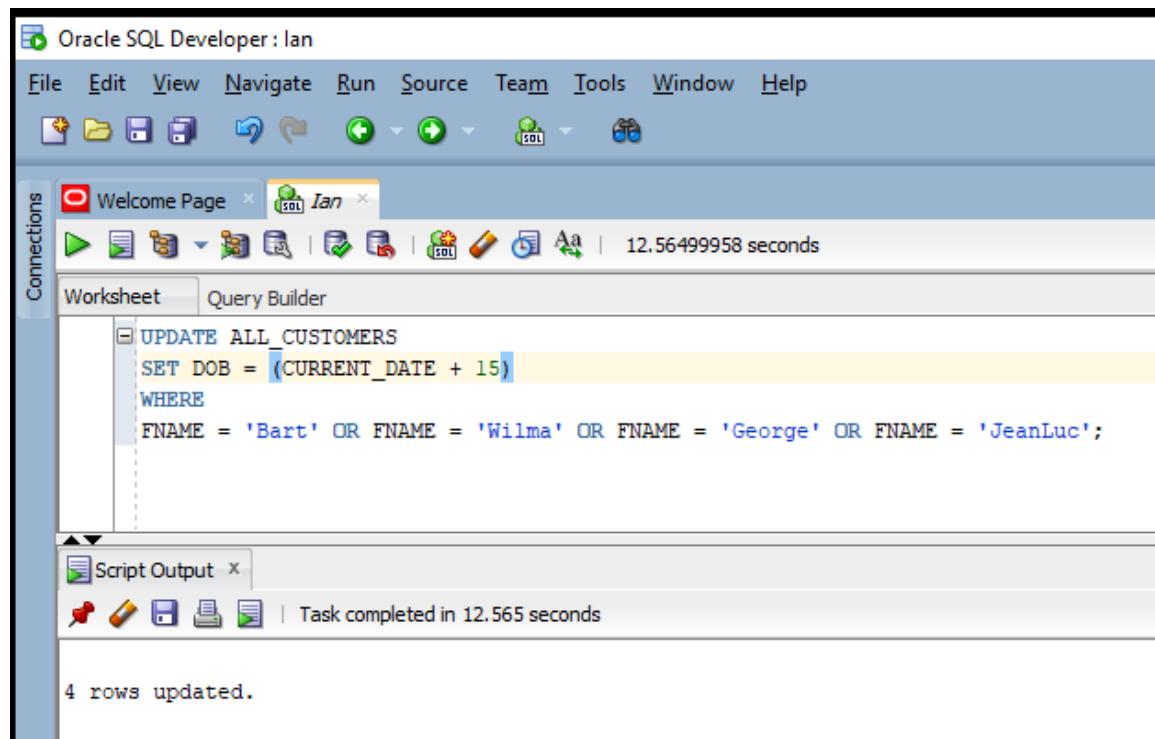
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	34534
------------	-----------	---------	--------	-------------	-------------	------------------------	-----------	----	-------	-------

12 rows selected.

Name	Null?	Type
CUSTID		NUMBER(10)
SSN		CHAR(9)
FNAME		VARCHAR2(20)
MNAME		VARCHAR2(20)
LNAME		VARCHAR2(25)
DOB		VARCHAR2(11)
STREET		VARCHAR2(30)
CITY		VARCHAR2(25)
STATE		VARCHAR2(2)
ZIP		VARCHAR2(10)
HOMEPHONE		CHAR(10)
WORKPHONE		CHAR(10)
ATMNO		CHAR(10)
ATMPW		VARCHAR2(30)
WEBID		VARCHAR2(20)
WEBPW		VARCHAR2(30)
EMAIL		VARCHAR2(50)
NUN_YEARLY_TRANSAC		NUMBER
TOT_BITCOIN_TRANSAC		NUMBER(8,2)

Activate Windows

## Update DOB for four rows



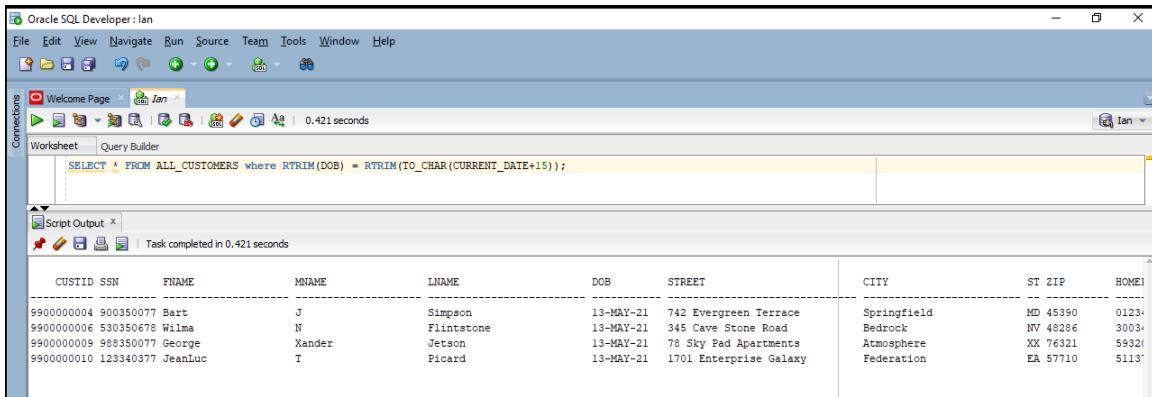
The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for connection management, navigation, and code editing. The Connections panel on the left shows a connection named "Ian". The main workspace has two tabs: "Worksheet" and "Query Builder". The "Worksheet" tab is active and contains the following SQL code:

```
UPDATE ALL_CUSTOMERS
SET DOB = (CURRENT_DATE + 15)
WHERE
  FNAME = 'Bart' OR FNAME = 'Wilma' OR FNAME = 'George' OR FNAME = 'JeanLuc';
```

The "Script Output" tab below shows the result of the execution:

```
4 rows updated.
```

## Verify updated DOBs



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.421 seconds

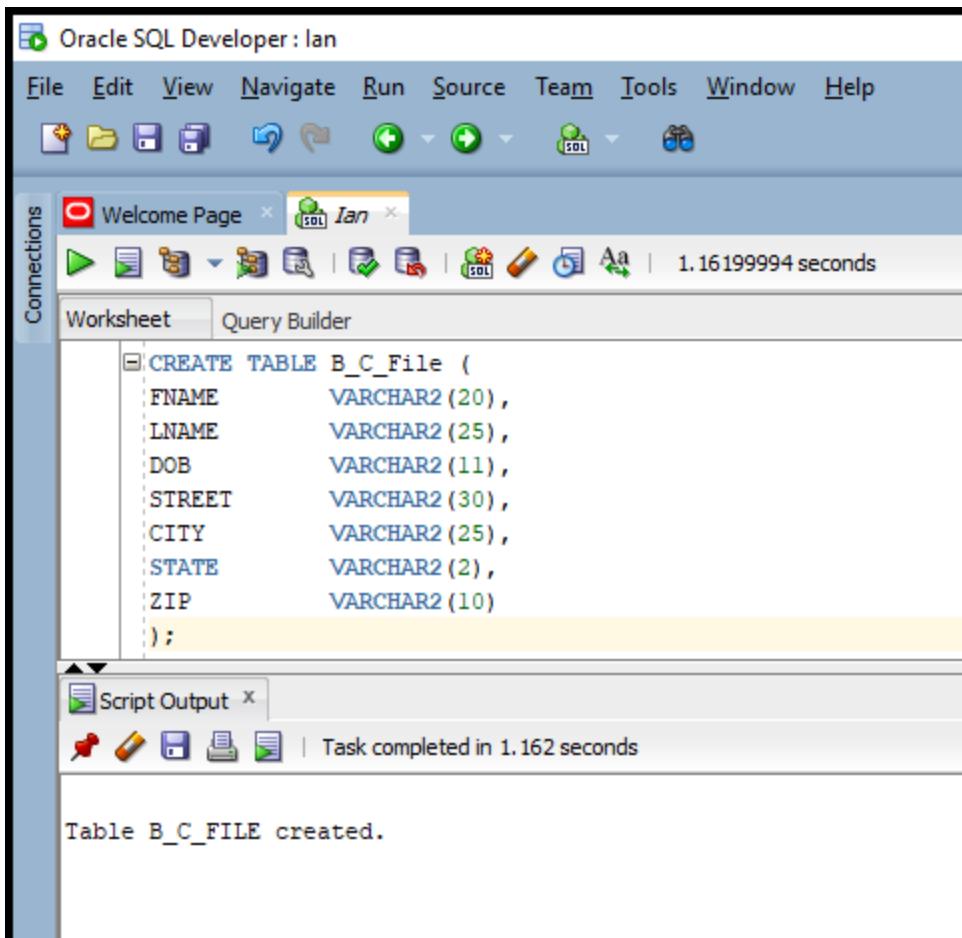
Worksheet Query Builder

```
SELECT * FROM ALL_CUSTOMERS where RTRIM(DOB) = RTRIM(TO_CHAR(CURRENT_DATE+15));
```

Script Output x Task completed in 0.421 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME1
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	48286	30034
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	59321
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	51137

## Create table b\_c\_file



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 1.16199994 seconds

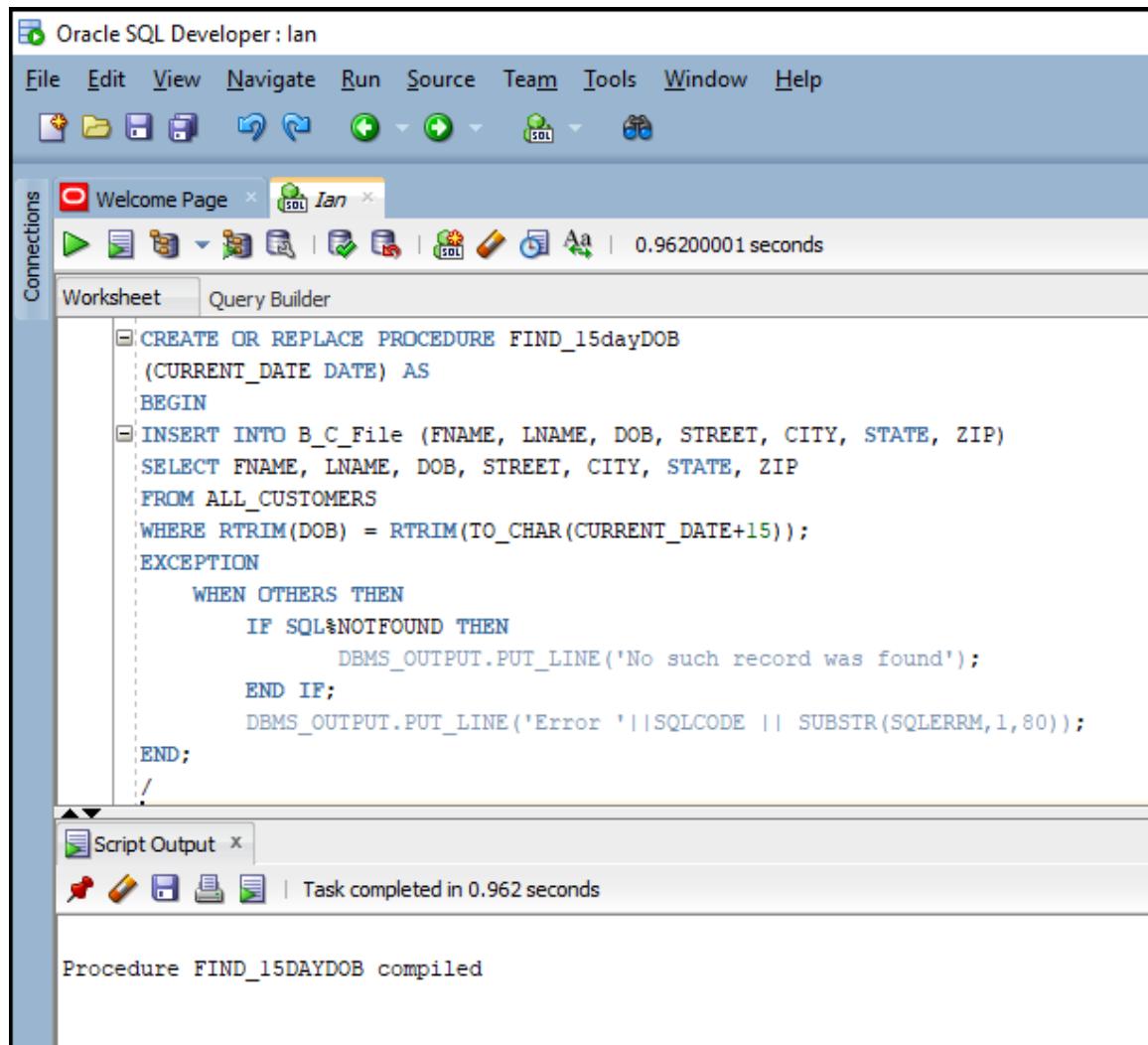
Worksheet Query Builder

```
CREATE TABLE B_C_File (
  FNAME      VARCHAR2(20),
  LNAME      VARCHAR2(25),
  DOB        VARCHAR2(11),
  STREET     VARCHAR2(30),
  CITY        VARCHAR2(25),
  STATE      VARCHAR2(2),
  ZIP        VARCHAR2(10)
);
```

Script Output x Task completed in 1.162 seconds

Table B\_C\_FILE created.

**Create procedure to locate rows with DOBs 15 days from today's date and insert them into table b\_c\_file**

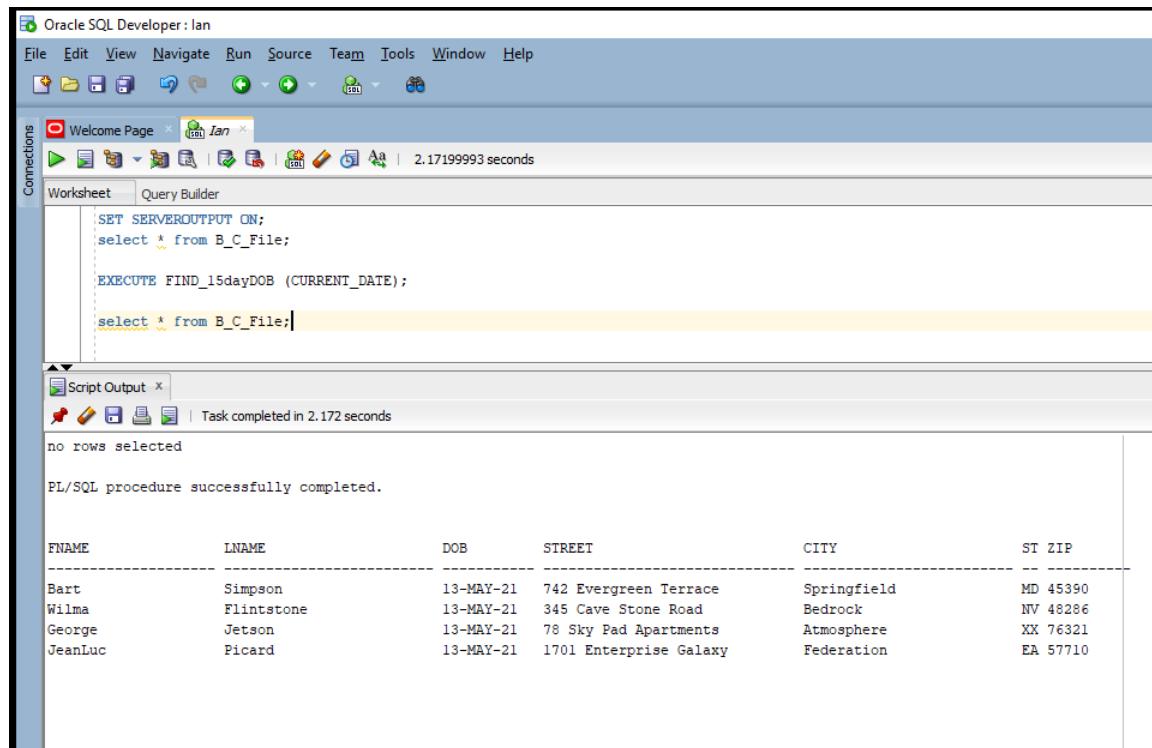


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE FIND_15dayDOB
  (CURRENT_DATE DATE) AS
BEGIN
  INSERT INTO B_C_File (FNAME, LNAME, DOB, STREET, CITY, STATE, ZIP)
  SELECT FNAME, LNAME, DOB, STREET, CITY, STATE, ZIP
  FROM ALL_CUSTOMERS
  WHERE RTRIM(DOB) = RTRIM(TO_CHAR(CURRENT_DATE+15));
  EXCEPTION
  WHEN OTHERS THEN
    IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('No such record was found');
    END IF;
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
  END;
/
```

The 'Script Output' tab at the bottom shows the message: "Procedure FIND\_15DAYDOB compiled".

**Execute procedure to locate rows with DOBs 15 days from today's date and insert them into table b\_c\_file.  
Display contents of table b\_c\_file.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
select * from B_C_File;

EXECUTE FIND_15dayDOB (CURRENT_DATE);

select * from B_C_File;
```

The 'Script Output' tab shows the results of the execution:

```
no rows selected

PL/SQL procedure successfully completed.
```

Below the output, the contents of the B\_C\_File table are displayed as a table:

FNAME	LNAME	DOB	STREET	CITY	ST ZIP
Bart	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD 45390
Wilma	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV 48286
George	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX 76321
JeanLuc	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA 57710

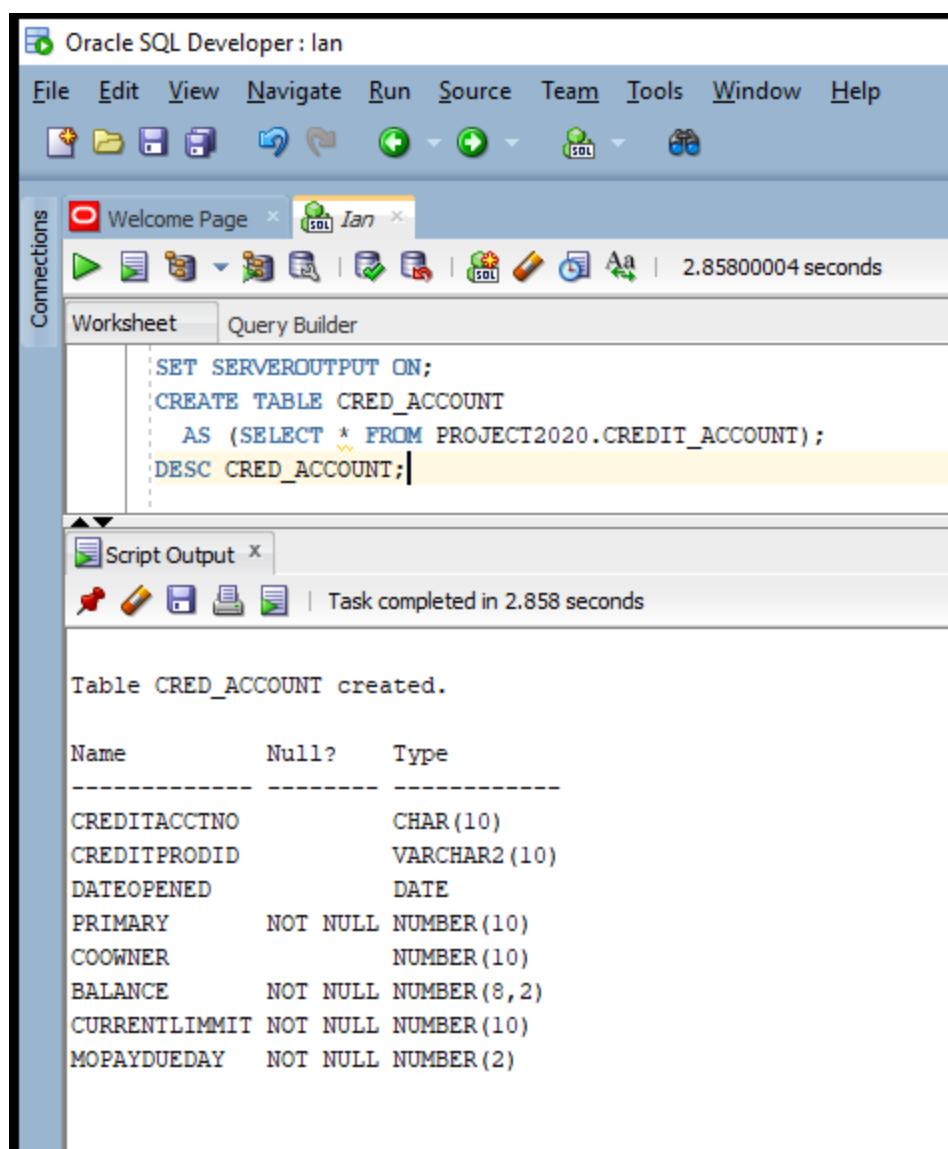
^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3C Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

C. Procedure to write all daily (today) transactions (credits and debits) into a database table called **Today\_Transaction**. (all deposits and withdraw for that day). You may use CREDIT\_ACCOUNT table. Use SYSDATE to get today's date. The structure of Today\_Transaction is: (Date & Time, Account number, Account Type, Amount, deposit/withdraw)

**Create cred\_account table using table PROJECT2020.CREDIT\_ACCOUNT.**  
**Display attribute types of cred\_account table.**



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. In the 'Worksheet' tab, the following SQL code is entered:

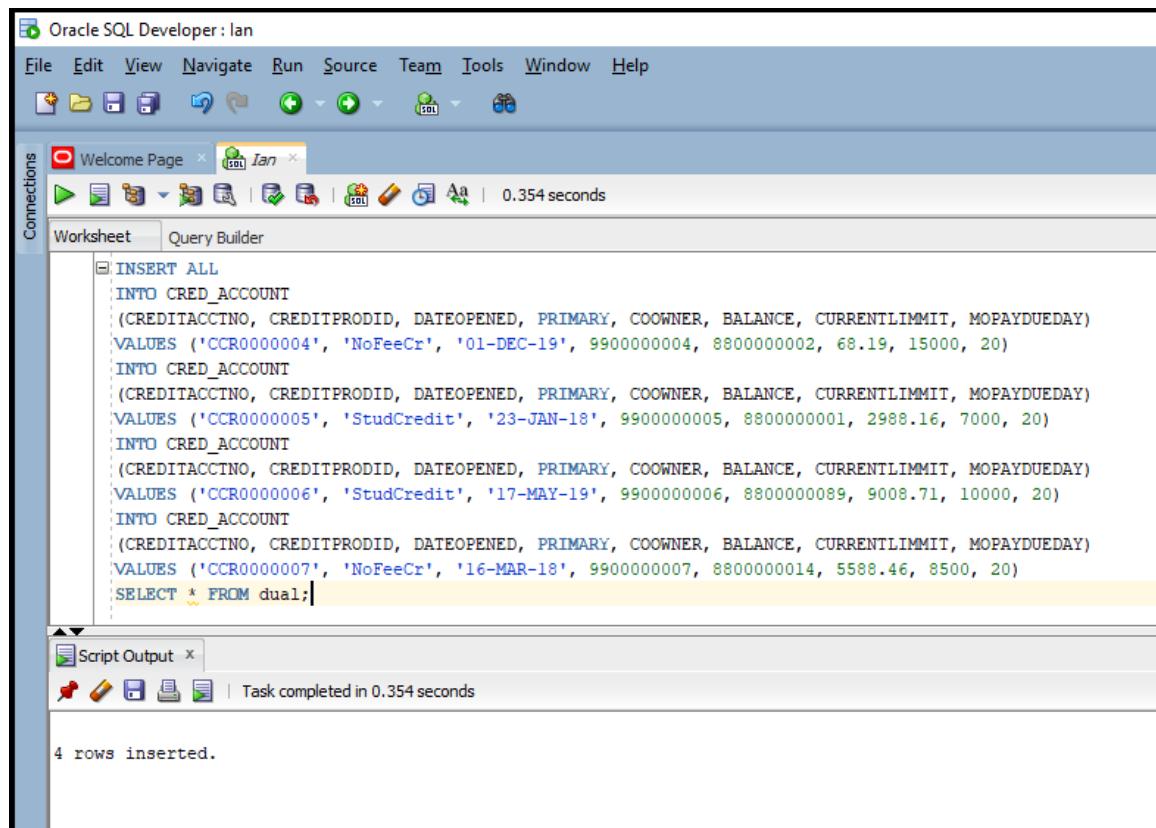
```
SET SERVEROUTPUT ON;
CREATE TABLE CRED_ACCOUNT
AS (SELECT * FROM PROJECT2020.CREDIT_ACCOUNT);
DESC CRED_ACCOUNT;
```

The 'Script Output' tab shows the result of the execution:

```
Table CRED_ACCOUNT created.

Name          Null?    Type
-----        -----
CREDITACCTNO      CHAR(10)
CREDITPRODID      VARCHAR2(10)
DATEOPENED        DATE
PRIMARY          NOT NULL NUMBER(10)
COOWNER           NUMBER(10)
BALANCE           NOT NULL NUMBER(8,2)
CURRENTLIMMIT    NOT NULL NUMBER(10)
MOPAYDUEDAY      NOT NULL NUMBER(2)
```

## Load entries into cred\_account table.



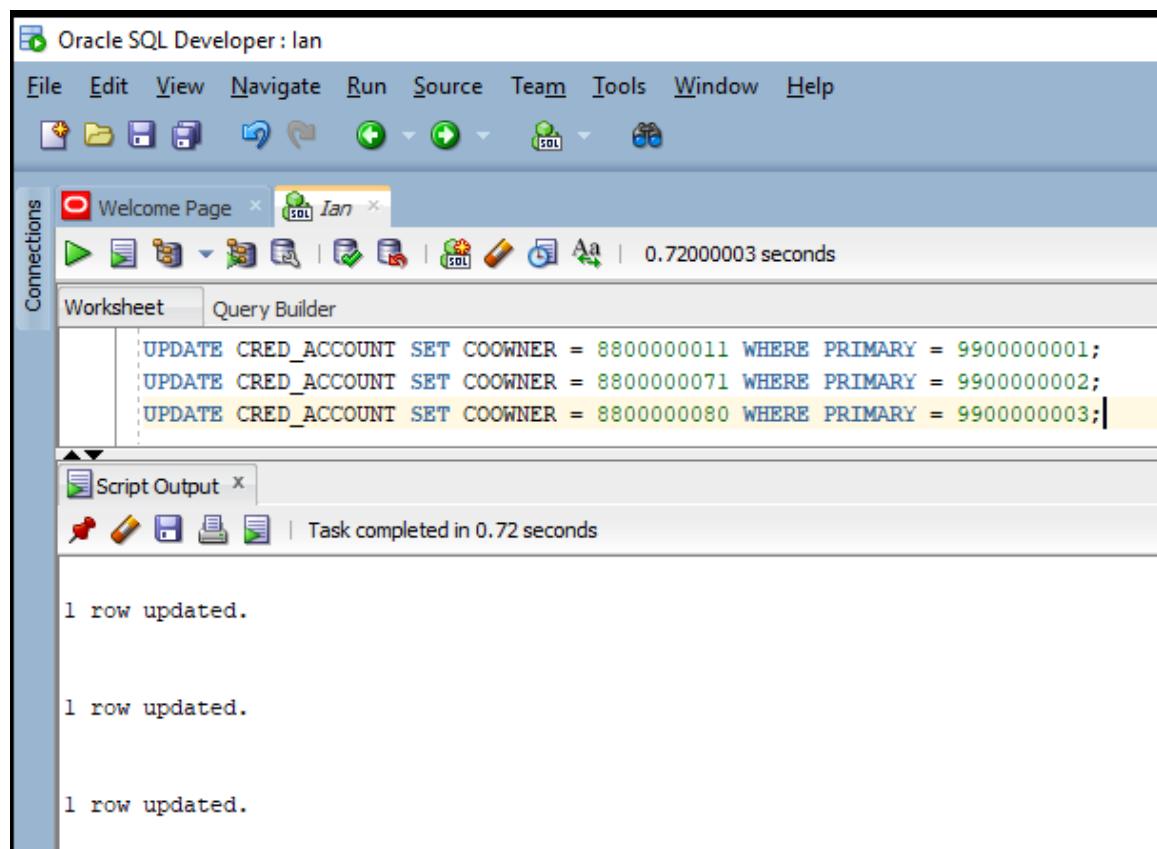
The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script. The script inserts four rows into the 'cred\_account' table. The 'Script Output' tab at the bottom shows the message '4 rows inserted.' indicating the successful execution of the query.

```
INSERT ALL
  INTO CRED_ACCOUNT
  (CREDITACCTNO, CREDITPRODID, DATEOPENED, PRIMARY, COOWNER, BALANCE, CURRENTLIMMIT, MOPAYUEDAY)
  VALUES ('CCR000004', 'NoFeeCr', '01-DEC-19', 9900000004, 8800000002, 68.19, 15000, 20)
  INTO CRED_ACCOUNT
  (CREDITACCTNO, CREDITPRODID, DATEOPENED, PRIMARY, COOWNER, BALANCE, CURRENTLIMMIT, MOPAYUEDAY)
  VALUES ('CCR000005', 'StudCredit', '23-JAN-18', 9900000005, 8800000001, 2988.16, 7000, 20)
  INTO CRED_ACCOUNT
  (CREDITACCTNO, CREDITPRODID, DATEOPENED, PRIMARY, COOWNER, BALANCE, CURRENTLIMMIT, MOPAYUEDAY)
  VALUES ('CCR000006', 'StudCredit', '17-MAY-19', 9900000006, 8800000089, 9008.71, 10000, 20)
  INTO CRED_ACCOUNT
  (CREDITACCTNO, CREDITPRODID, DATEOPENED, PRIMARY, COOWNER, BALANCE, CURRENTLIMMIT, MOPAYUEDAY)
  VALUES ('CCR000007', 'NoFeeCr', '16-MAR-18', 9900000007, 8800000014, 5588.46, 8500, 20)
  SELECT * FROM dual;
```

Script Output | Task completed in 0.354 seconds

4 rows inserted.

## Update a few entries in cred\_account table.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying three SQL UPDATE statements:

```
UPDATE CRED_ACCOUNT SET COOWNER = 8800000011 WHERE PRIMARY = 9900000001;
UPDATE CRED_ACCOUNT SET COOWNER = 8800000071 WHERE PRIMARY = 9900000002;
UPDATE CRED_ACCOUNT SET COOWNER = 8800000080 WHERE PRIMARY = 9900000003;
```

Below the worksheet, the 'Script Output' tab shows the results of the updates:

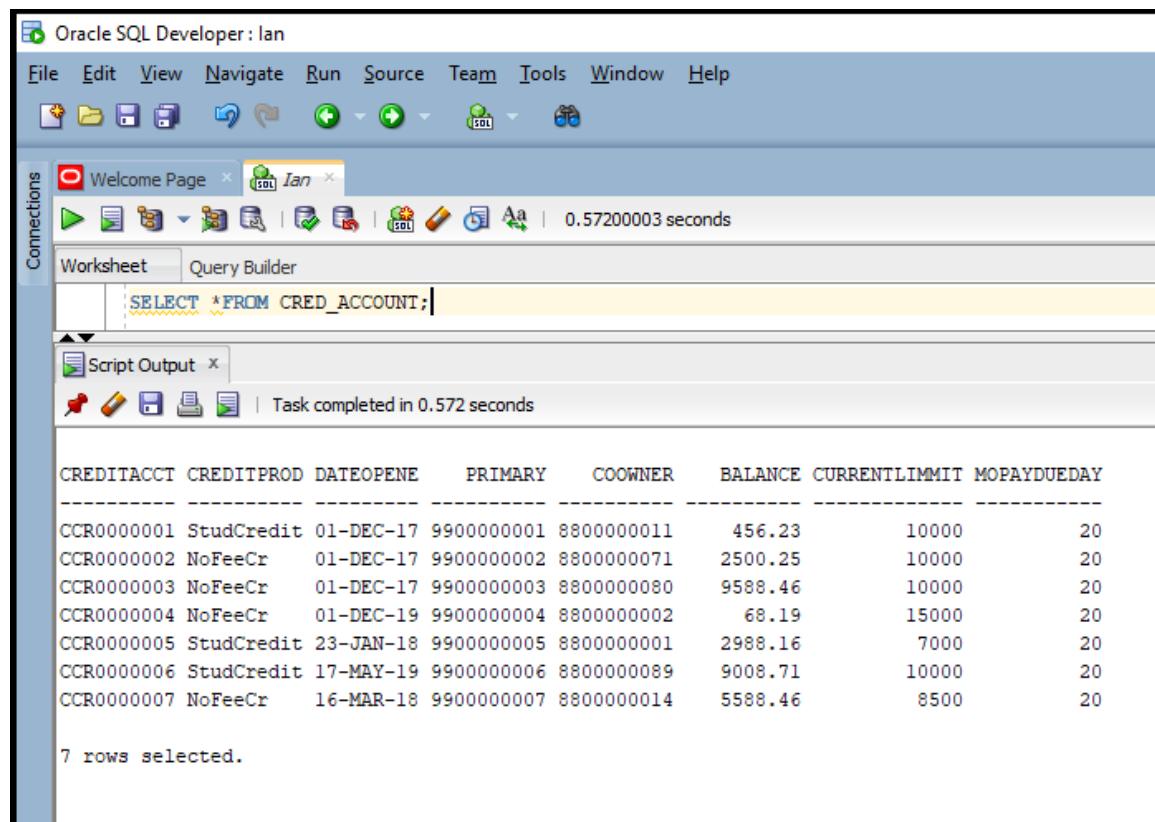
```
1 row updated.

1 row updated.

1 row updated.
```

The status bar at the bottom indicates the task completed in 0.72 seconds.

## Display contents of cred\_account table.



The screenshot shows the Oracle SQL Developer interface with a query window displaying the results of a SELECT statement on the cred\_account table.

Query:

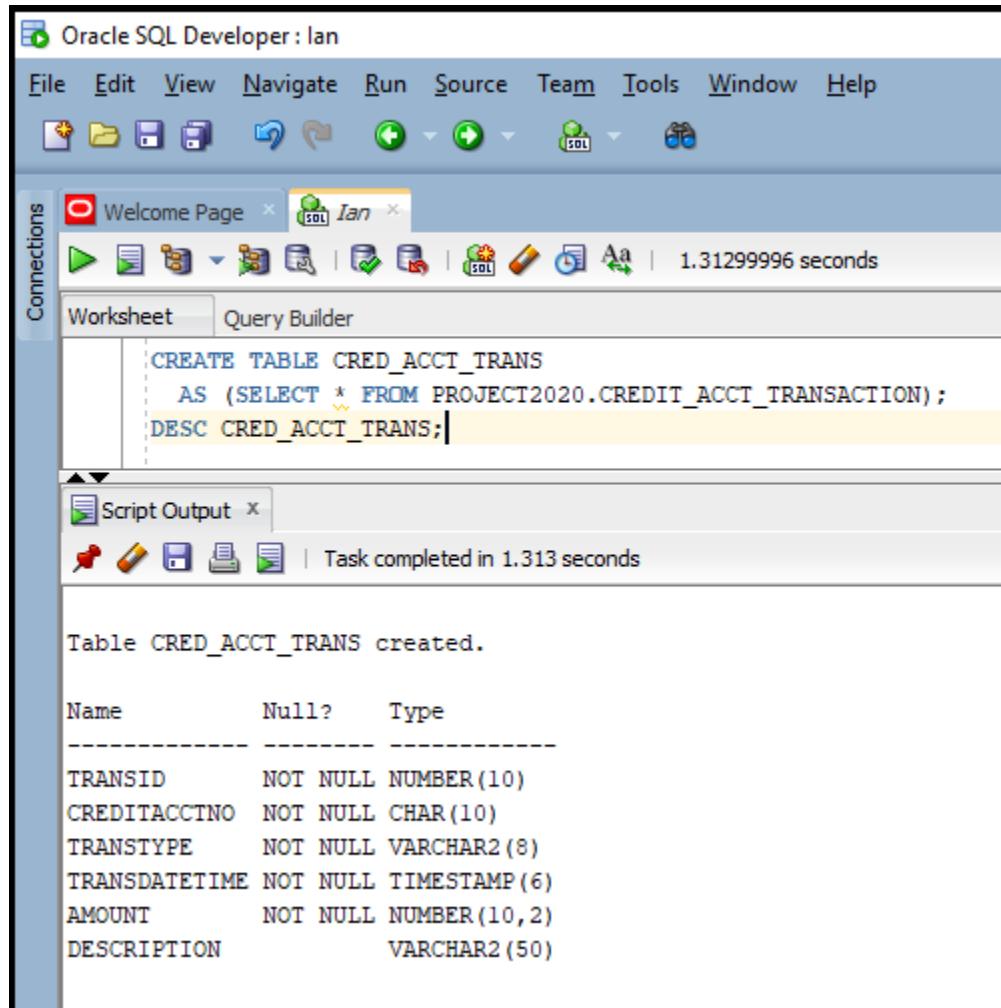
```
SELECT * FROM CRED_ACCOUNT;
```

Output:

CREDITACCT	CREDITPROD	DATEOPENE	PRIMARY	COOWNER	BALANCE	CURRENTLIMMIT	MOPAYDUEDAY
CCR0000001	StudCredit	01-DEC-17	9900000001	8800000011	456.23	10000	20
CCR0000002	NoFeeCr	01-DEC-17	9900000002	8800000071	2500.25	10000	20
CCR0000003	NoFeeCr	01-DEC-17	9900000003	8800000080	9588.46	10000	20
CCR0000004	NoFeeCr	01-DEC-19	9900000004	8800000002	68.19	15000	20
CCR0000005	StudCredit	23-JAN-18	9900000005	8800000001	2988.16	7000	20
CCR0000006	StudCredit	17-MAY-19	9900000006	8800000089	9008.71	10000	20
CCR0000007	NoFeeCr	16-MAR-18	9900000007	8800000014	5588.46	8500	20

7 rows selected.

**Create cred\_acct\_trans table using table  
PROJECT2020.CREDIT\_ACCT\_TRANSACTION.  
Display attribute types of cred\_acct\_trans table.**



The screenshot shows the Oracle SQL Developer interface with the following details:

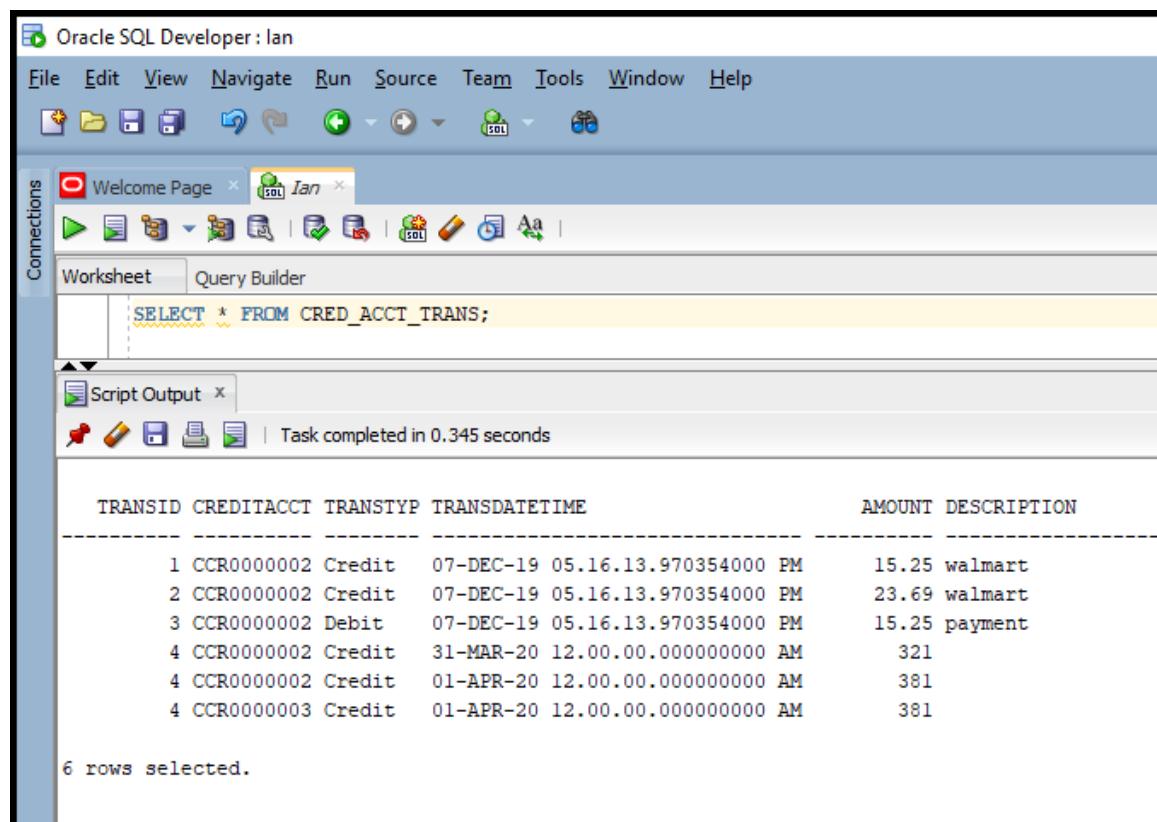
- Connections:** A list of connections, with "Ian" selected.
- Worksheet:** The main workspace where the following SQL code is entered:

```
CREATE TABLE CRED_ACCT_TRANS
AS (SELECT * FROM PROJECT2020.CREDIT_ACCT_TRANSACTION);
DESC CRED_ACCT_TRANS;
```
- Script Output:** A panel showing the execution results:
  - Message: "Table CRED\_ACCT\_TRANS created."
  - Table structure (described in the text below)
- Output:** A message indicating the task completed in 1.313 seconds.

**Table Structure:**

Name	Null?	Type
TRANSID	NOT NULL	NUMBER(10)
CREDITACCTNO	NOT NULL	CHAR(10)
TRANSTYPE	NOT NULL	VARCHAR2(8)
TRANSDATETIME	NOT NULL	TIMESTAMP(6)
AMOUNT	NOT NULL	NUMBER(10,2)
DESCRIPTION		VARCHAR2(50)

## Display contents of cred\_acct\_trans table.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. A query is run in the 'Worksheet' tab:

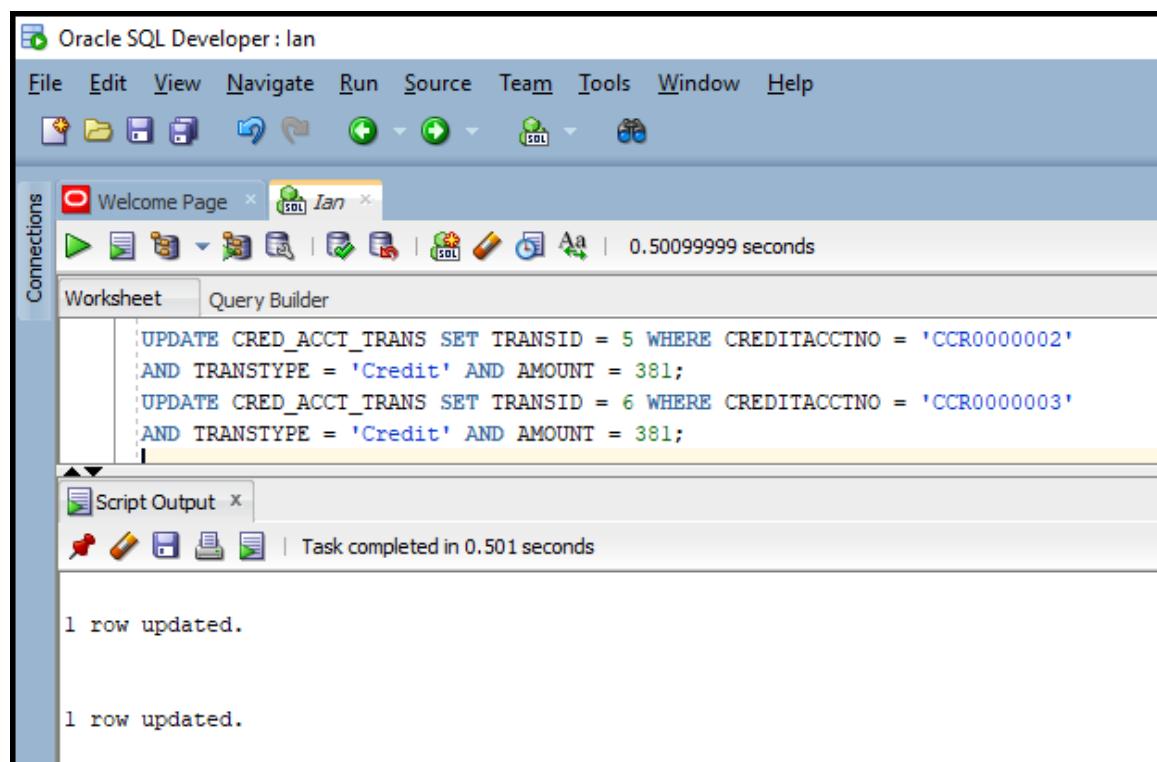
```
SELECT * FROM CRED_ACCT_TRANS;
```

The output in the 'Script Output' tab shows the results of the query:

TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
1	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	15.25	walmart
2	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	23.69	walmart
3	CCR0000002	Debit	07-DEC-19 05.16.13.970354000 PM	15.25	payment
4	CCR0000002	Credit	31-MAR-20 12.00.00.000000000 AM	321	
4	CCR0000002	Credit	01-APR-20 12.00.00.000000000 AM	381	
4	CCR0000003	Credit	01-APR-20 12.00.00.000000000 AM	381	

6 rows selected.

## Update a two entries in cred\_acct\_trans table.



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL execution. The Connections sidebar shows a connection named "Ian". The main workspace has two tabs: "Worksheet" and "Query Builder". The "Worksheet" tab contains the following SQL script:

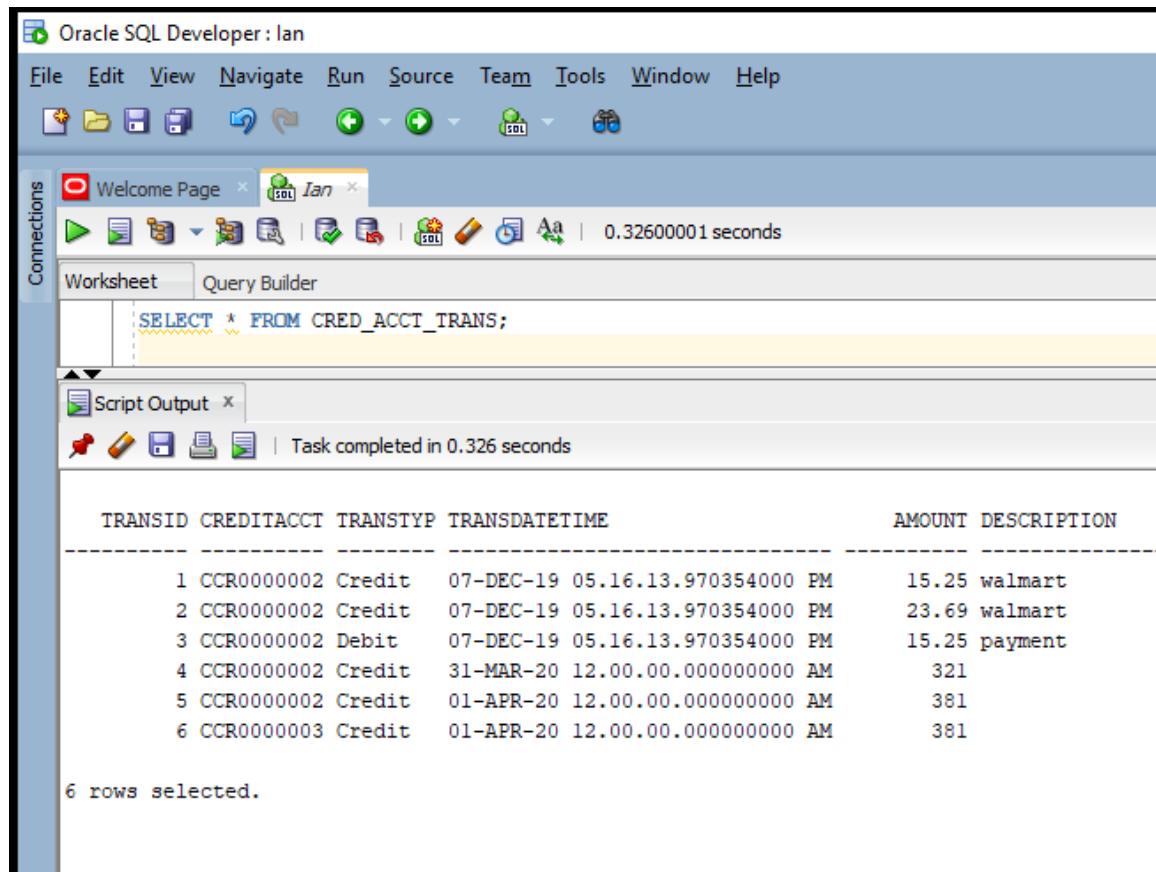
```
UPDATE CRED_ACCT_TRANS SET TRANSID = 5 WHERE CREDITACCTNO = 'CCR0000002'  
AND TRANSTYPE = 'Credit' AND AMOUNT = 381;  
UPDATE CRED_ACCT_TRANS SET TRANSID = 6 WHERE CREDITACCTNO = 'CCR0000003'  
AND TRANSTYPE = 'Credit' AND AMOUNT = 381;
```

The "Script Output" tab shows the results of the execution:

```
1 row updated.  
1 row updated.
```

The status bar at the bottom indicates "Task completed in 0.501 seconds".

## Verify updated entries in cred\_acct\_trans table.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL query:

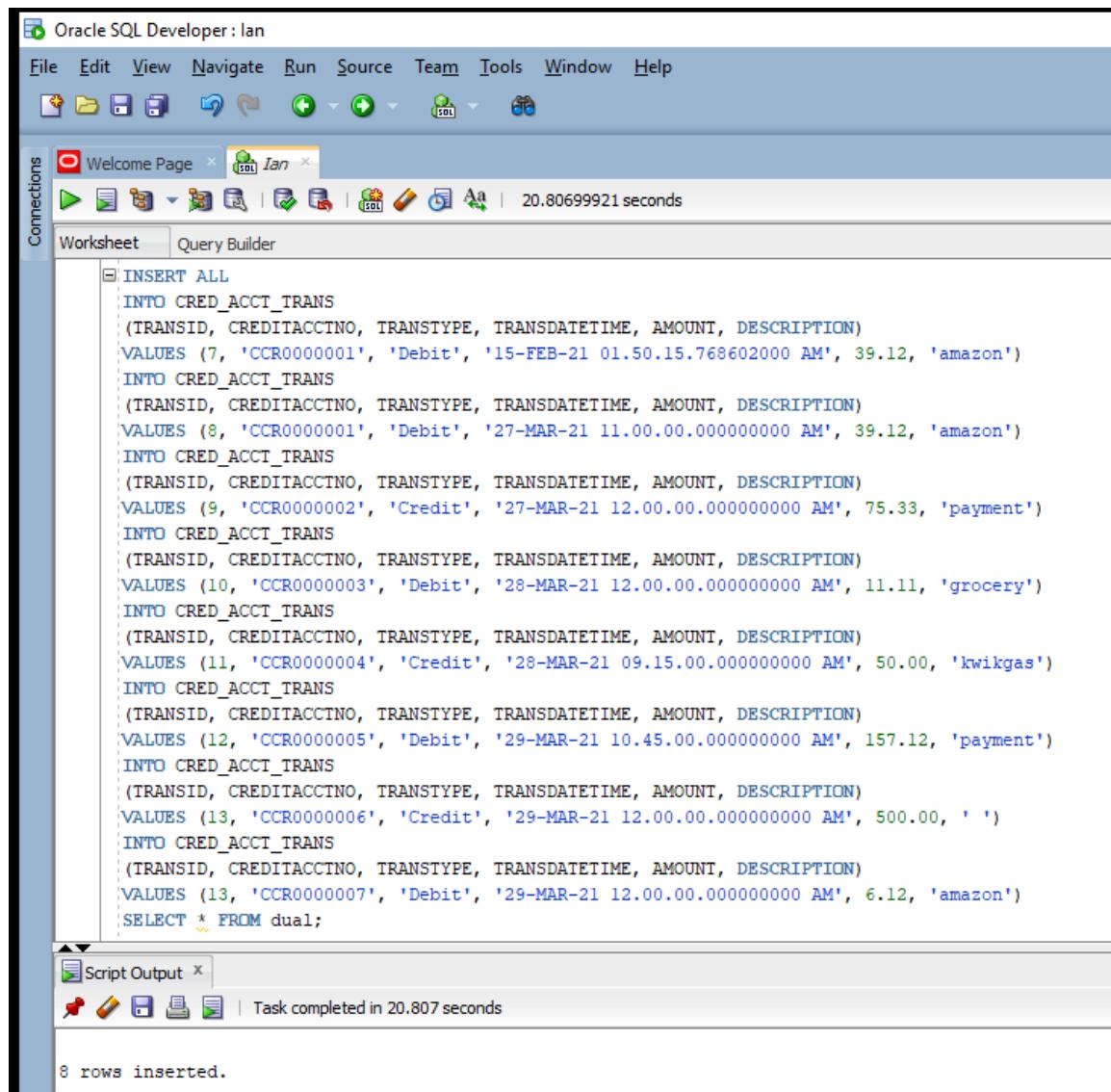
```
SELECT * FROM CRED_ACCT_TRANS;
```

The results are displayed in a table format:

TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
1	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	15.25	walmart
2	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	23.69	walmart
3	CCR0000002	Debit	07-DEC-19 05.16.13.970354000 PM	15.25	payment
4	CCR0000002	Credit	31-MAR-20 12.00.00.000000000 AM	321	
5	CCR0000002	Credit	01-APR-20 12.00.00.000000000 AM	381	
6	CCR0000003	Credit	01-APR-20 12.00.00.000000000 AM	381	

Below the table, the message '6 rows selected.' is displayed.

## Insert new rows into cred\_acct\_trans table.



The screenshot shows the Oracle SQL Developer interface with a query script in the Worksheet tab. The script inserts 8 rows into the cred\_acct\_trans table. The rows represent various transactions with different dates, amounts, and descriptions, such as 'amazon', 'payment', and 'kwikgas'. The script uses the INSERT ALL INTO ... SELECT \* FROM dual; syntax. The Script Output tab shows the message 'Task completed in 20.807 seconds' and '8 rows inserted.'

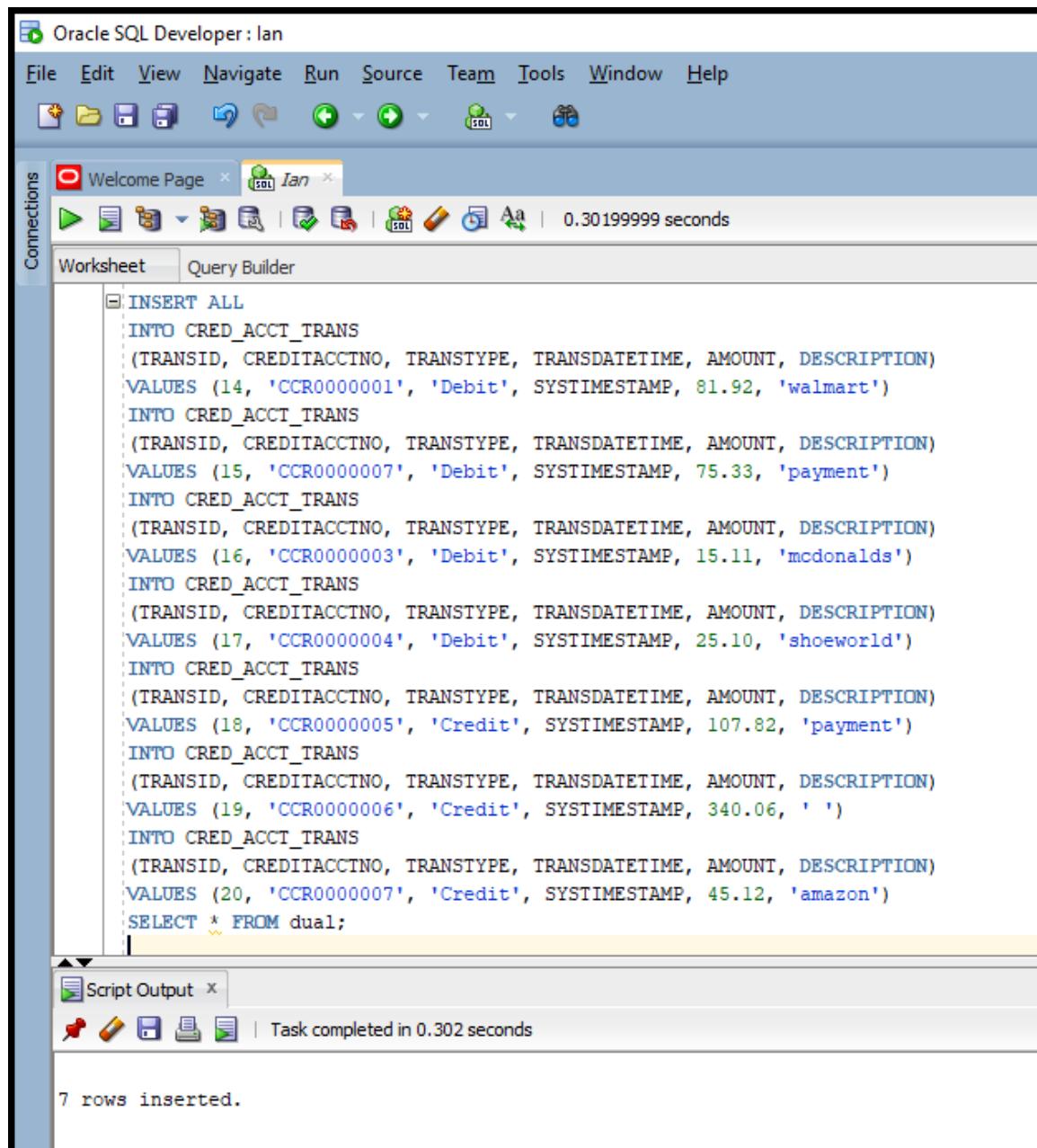
```
INSERT ALL
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (7, 'CCR0000001', 'Debit', '15-FEB-21 01.50.15.768602000 AM', 39.12, 'amazon')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (8, 'CCR0000001', 'Debit', '27-MAR-21 11.00.00.000000000 AM', 39.12, 'amazon')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (9, 'CCR0000002', 'Credit', '27-MAR-21 12.00.00.000000000 AM', 75.33, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (10, 'CCR0000003', 'Debit', '28-MAR-21 12.00.00.000000000 AM', 11.11, 'grocery')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (11, 'CCR0000004', 'Credit', '28-MAR-21 09.15.00.000000000 AM', 50.00, 'kwikgas')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (12, 'CCR0000005', 'Debit', '29-MAR-21 10.45.00.000000000 AM', 157.12, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (13, 'CCR0000006', 'Credit', '29-MAR-21 12.00.00.000000000 AM', 500.00, ' ')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (13, 'CCR0000007', 'Debit', '29-MAR-21 12.00.00.000000000 AM', 6.12, 'amazon')
  SELECT * FROM dual;
```

Script Output x

Task completed in 20.807 seconds

8 rows inserted.

## Insert more new rows into cred\_acct\_trans table.



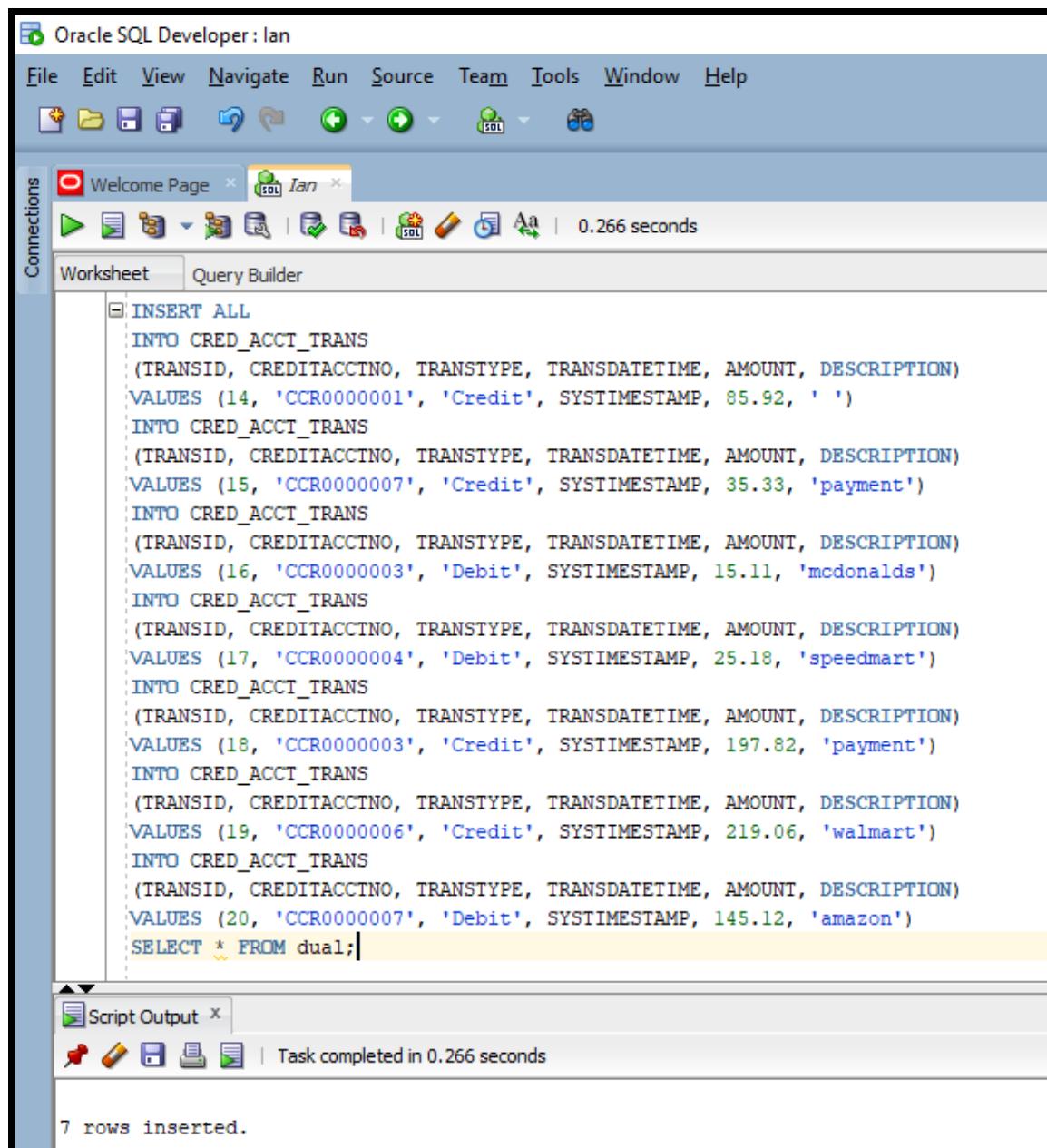
The screenshot shows the Oracle SQL Developer interface. The Worksheet tab contains a script for inserting data into the CRED\_ACCT\_TRANS table. The Script Output tab shows the results of the execution.

```
INSERT ALL
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (14, 'CCR0000001', 'Debit', SYSTIMESTAMP, 81.92, 'walmart')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (15, 'CCR0000007', 'Debit', SYSTIMESTAMP, 75.33, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (16, 'CCR0000003', 'Debit', SYSTIMESTAMP, 15.11, 'mcdonalds')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (17, 'CCR0000004', 'Debit', SYSTIMESTAMP, 25.10, 'shoeworld')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (18, 'CCR0000005', 'Credit', SYSTIMESTAMP, 107.82, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (19, 'CCR0000006', 'Credit', SYSTIMESTAMP, 340.06, ' ')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
    VALUES (20, 'CCR0000007', 'Credit', SYSTIMESTAMP, 45.12, 'amazon')
  SELECT * FROM dual;
```

Script Output tab output:

7 rows inserted.

## Insert still more new rows into cred\_acct\_trans table.

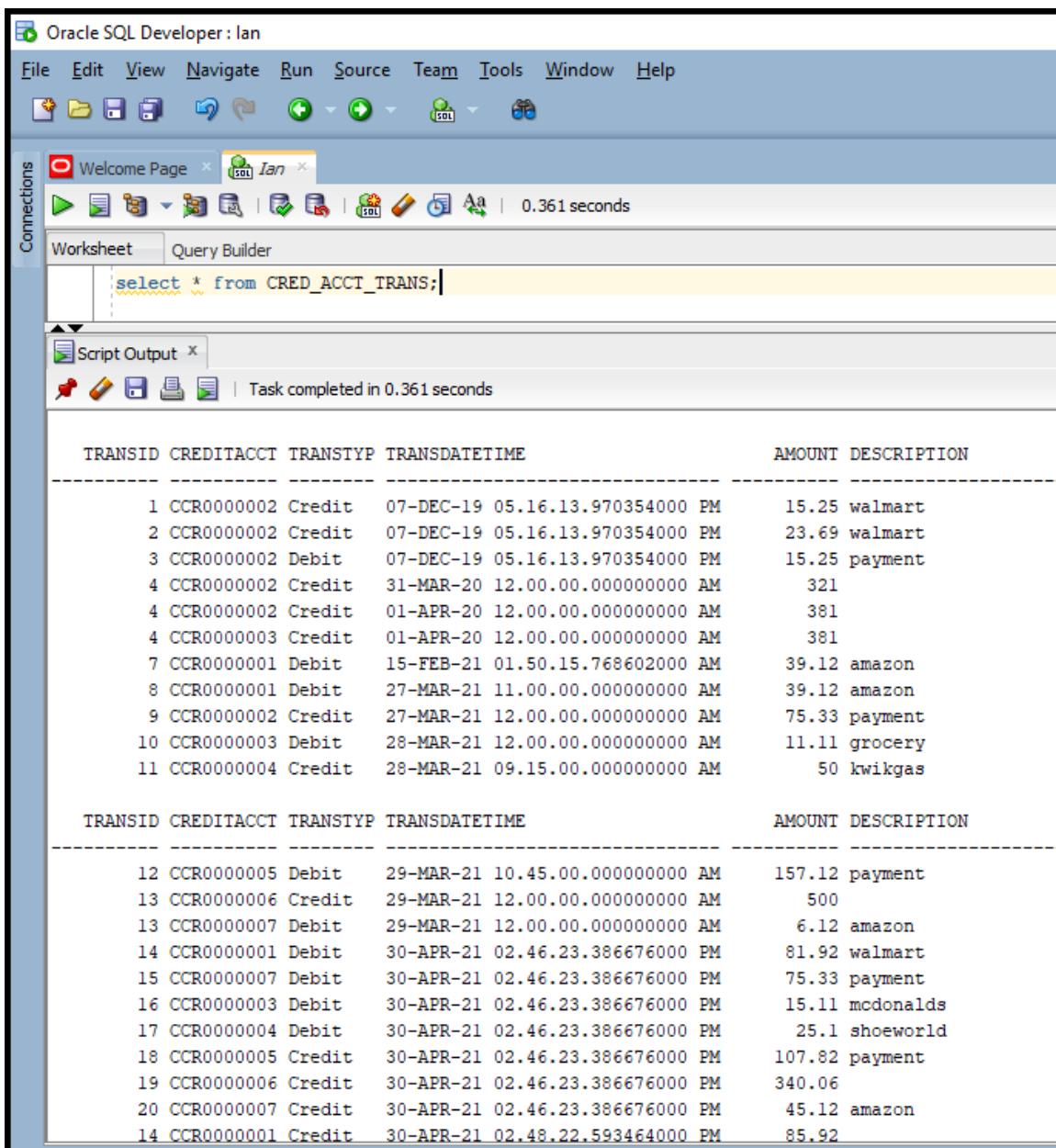


The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar below has various icons for file operations and SQL execution. The Connections sidebar shows a connection named 'Ian'. The main area has tabs for Worksheet and Query Builder, with the Worksheet tab active. The query window contains the following SQL script:

```
INSERT ALL
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (14, 'CCR0000001', 'Credit', SYSTIMESTAMP, 85.92, ' ')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (15, 'CCR0000007', 'Credit', SYSTIMESTAMP, 35.33, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (16, 'CCR0000003', 'Debit', SYSTIMESTAMP, 15.11, 'mcdonalds')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (17, 'CCR0000004', 'Debit', SYSTIMESTAMP, 25.18, 'speedmart')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (18, 'CCR0000003', 'Credit', SYSTIMESTAMP, 197.82, 'payment')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (19, 'CCR0000006', 'Credit', SYSTIMESTAMP, 219.06, 'walmart')
  INTO CRED_ACCT_TRANS
    (TRANSID, CREDITACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION)
  VALUES (20, 'CCR0000007', 'Debit', SYSTIMESTAMP, 145.12, 'amazon')
  SELECT * FROM dual;
```

The Script Output tab shows the result of the execution: "7 rows inserted." and "Task completed in 0.266 seconds".

## Verify new entries into cred\_acct\_trans table.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan

Worksheet Query Builder

select \* from CRED\_ACCT\_TRANS;

Script Output Task completed in 0.361 seconds

TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
1	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	15.25	walmart
2	CCR0000002	Credit	07-DEC-19 05.16.13.970354000 PM	23.69	walmart
3	CCR0000002	Debit	07-DEC-19 05.16.13.970354000 PM	15.25	payment
4	CCR0000002	Credit	31-MAR-20 12.00.00.000000000 AM	321	
4	CCR0000002	Credit	01-APR-20 12.00.00.000000000 AM	381	
4	CCR0000003	Credit	01-APR-20 12.00.00.000000000 AM	381	
7	CCR0000001	Debit	15-FEB-21 01.50.15.768602000 AM	39.12	amazon
8	CCR0000001	Debit	27-MAR-21 11.00.00.000000000 AM	39.12	amazon
9	CCR0000002	Credit	27-MAR-21 12.00.00.000000000 AM	75.33	payment
10	CCR0000003	Debit	28-MAR-21 12.00.00.000000000 AM	11.11	grocery
11	CCR0000004	Credit	28-MAR-21 09.15.00.000000000 AM	50	kwikgas
TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
12	CCR0000005	Debit	29-MAR-21 10.45.00.000000000 AM	157.12	payment
13	CCR0000006	Credit	29-MAR-21 12.00.00.000000000 AM	500	
13	CCR0000007	Debit	29-MAR-21 12.00.00.000000000 AM	6.12	amazon
14	CCR0000001	Debit	30-APR-21 02.46.23.386676000 PM	81.92	walmart
15	CCR0000007	Debit	30-APR-21 02.46.23.386676000 PM	75.33	payment
16	CCR0000003	Debit	30-APR-21 02.46.23.386676000 PM	15.11	mcdonalds
17	CCR0000004	Debit	30-APR-21 02.46.23.386676000 PM	25.1	shoeworld
18	CCR0000005	Credit	30-APR-21 02.46.23.386676000 PM	107.82	payment
19	CCR0000006	Credit	30-APR-21 02.46.23.386676000 PM	340.06	
20	CCR0000007	Credit	30-APR-21 02.46.23.386676000 PM	45.12	amazon
14	CCR0000001	Credit	30-APR-21 02.48.22.593464000 PM	85.92	

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections    Welcome Page    Ian

Worksheet    Query Builder

```
select * from CRED_ACCT_TRANS;
```

Script Output    Task completed in 0.361 seconds

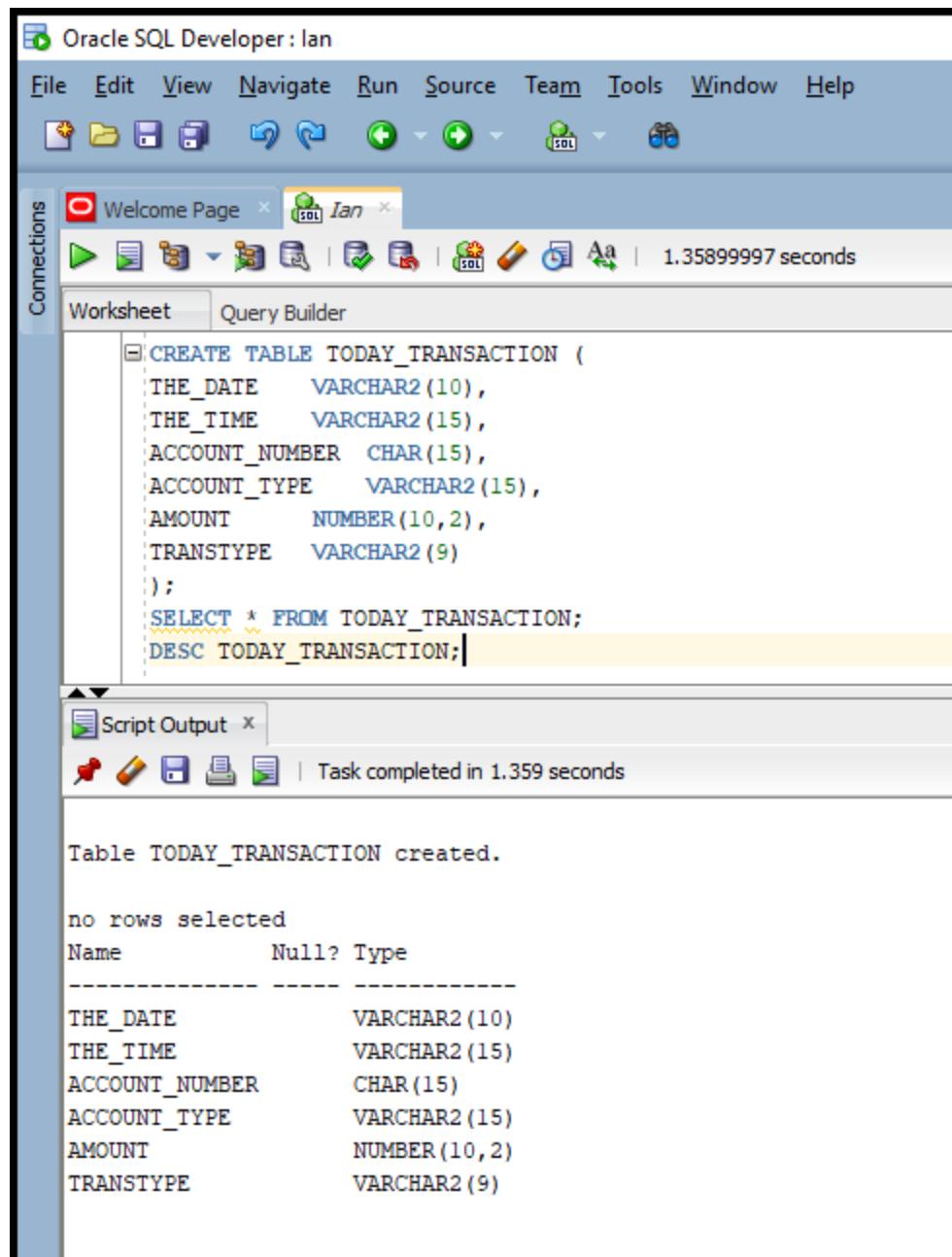
TRANSID	CREDITACCT	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION
11	CCR0000004	Credit	28-MAR-21 09.15.00.000000000 AM	50	kwikgas
12	CCR0000005	Debit	29-MAR-21 10.45.00.000000000 AM	157.12	payment
13	CCR0000006	Credit	29-MAR-21 12.00.00.000000000 AM	500	
13	CCR0000007	Debit	29-MAR-21 12.00.00.000000000 AM	6.12	amazon
14	CCR0000001	Debit	30-APR-21 02.46.23.386676000 PM	81.92	walmart
15	CCR0000007	Debit	30-APR-21 02.46.23.386676000 PM	75.33	payment
16	CCR0000003	Debit	30-APR-21 02.46.23.386676000 PM	15.11	mcdonalds
17	CCR0000004	Debit	30-APR-21 02.46.23.386676000 PM	25.1	shoeworld
18	CCR0000005	Credit	30-APR-21 02.46.23.386676000 PM	107.82	payment
19	CCR0000006	Credit	30-APR-21 02.46.23.386676000 PM	340.06	
20	CCR0000007	Credit	30-APR-21 02.46.23.386676000 PM	45.12	amazon
14	CCR0000001	Credit	30-APR-21 02.48.22.593464000 PM	85.92	
15	CCR0000007	Credit	30-APR-21 02.48.22.593464000 PM	35.33	payment
16	CCR0000003	Debit	30-APR-21 02.48.22.593464000 PM	15.11	mcdonalds
17	CCR0000004	Debit	30-APR-21 02.48.22.593464000 PM	25.18	speedmart
18	CCR0000003	Credit	30-APR-21 02.48.22.593464000 PM	197.82	payment
19	CCR0000006	Credit	30-APR-21 02.48.22.593464000 PM	219.06	walmart
20	CCR0000007	Debit	30-APR-21 02.48.22.593464000 PM	145.12	amazon

28 rows selected.

**Create table today\_transaction.**

**Show contents of today\_transaction table.**

**Show attribute data types today\_transaction table.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL script:

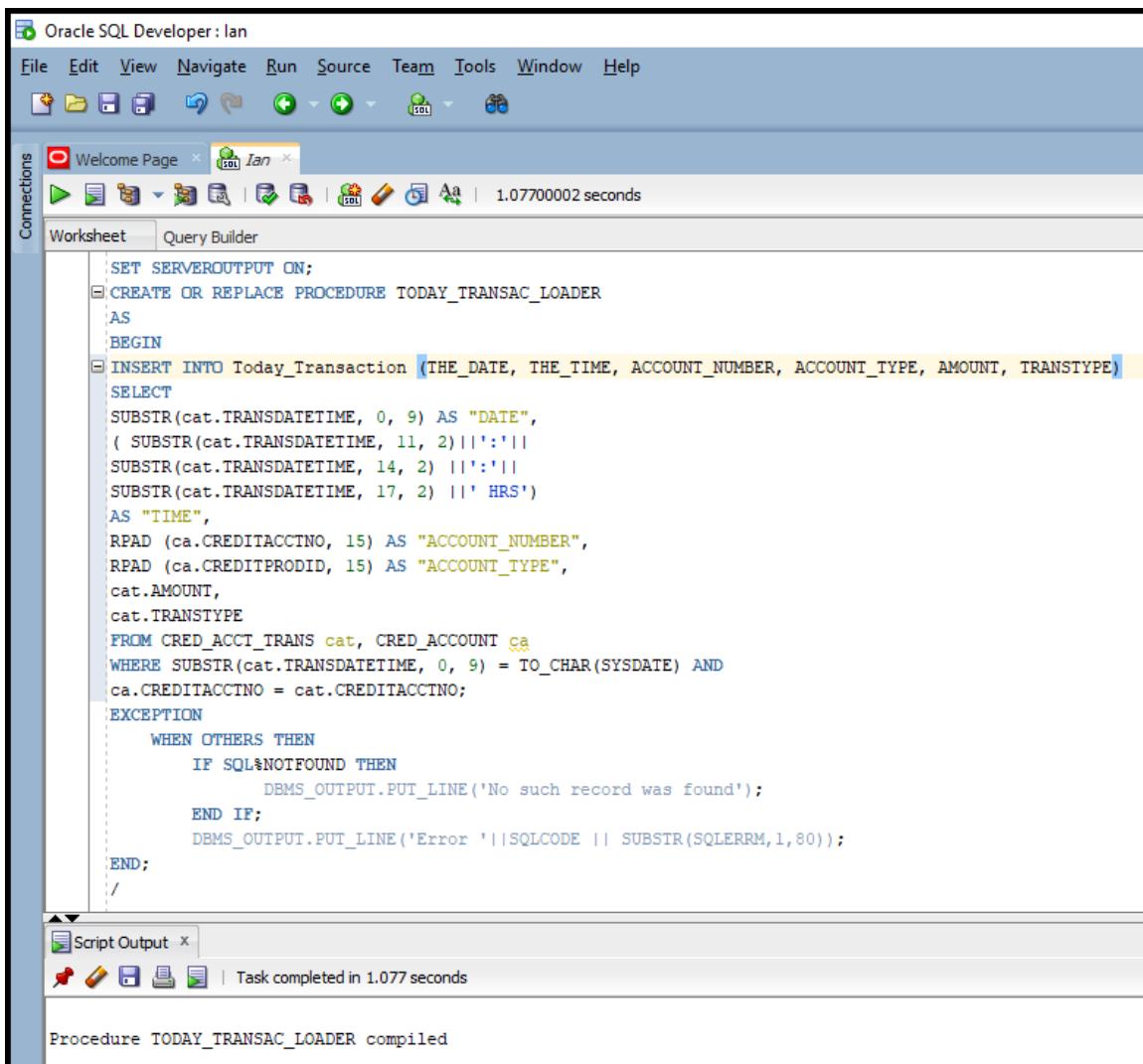
```
CREATE TABLE TODAY_TRANSACTION (
    THE_DATE      VARCHAR2(10),
    THE_TIME      VARCHAR2(15),
    ACCOUNT_NUMBER CHAR(15),
    ACCOUNT_TYPE   VARCHAR2(15),
    AMOUNT        NUMBER(10,2),
    TRANSTYPE     VARCHAR2(9)
);
SELECT * FROM TODAY_TRANSACTION;
DESC TODAY_TRANSACTION;
```

The 'Script Output' tab shows the results of the execution:

```
Table TODAY_TRANSACTION created.

no rows selected
Name      Null? Type
-----  -----
THE_DATE      VARCHAR2(10)
THE_TIME      VARCHAR2(15)
ACCOUNT_NUMBER CHAR(15)
ACCOUNT_TYPE   VARCHAR2(15)
AMOUNT        NUMBER(10,2)
TRANSTYPE     VARCHAR2(9)
```

## Create procedure to load today\_transaction table with exception handing.



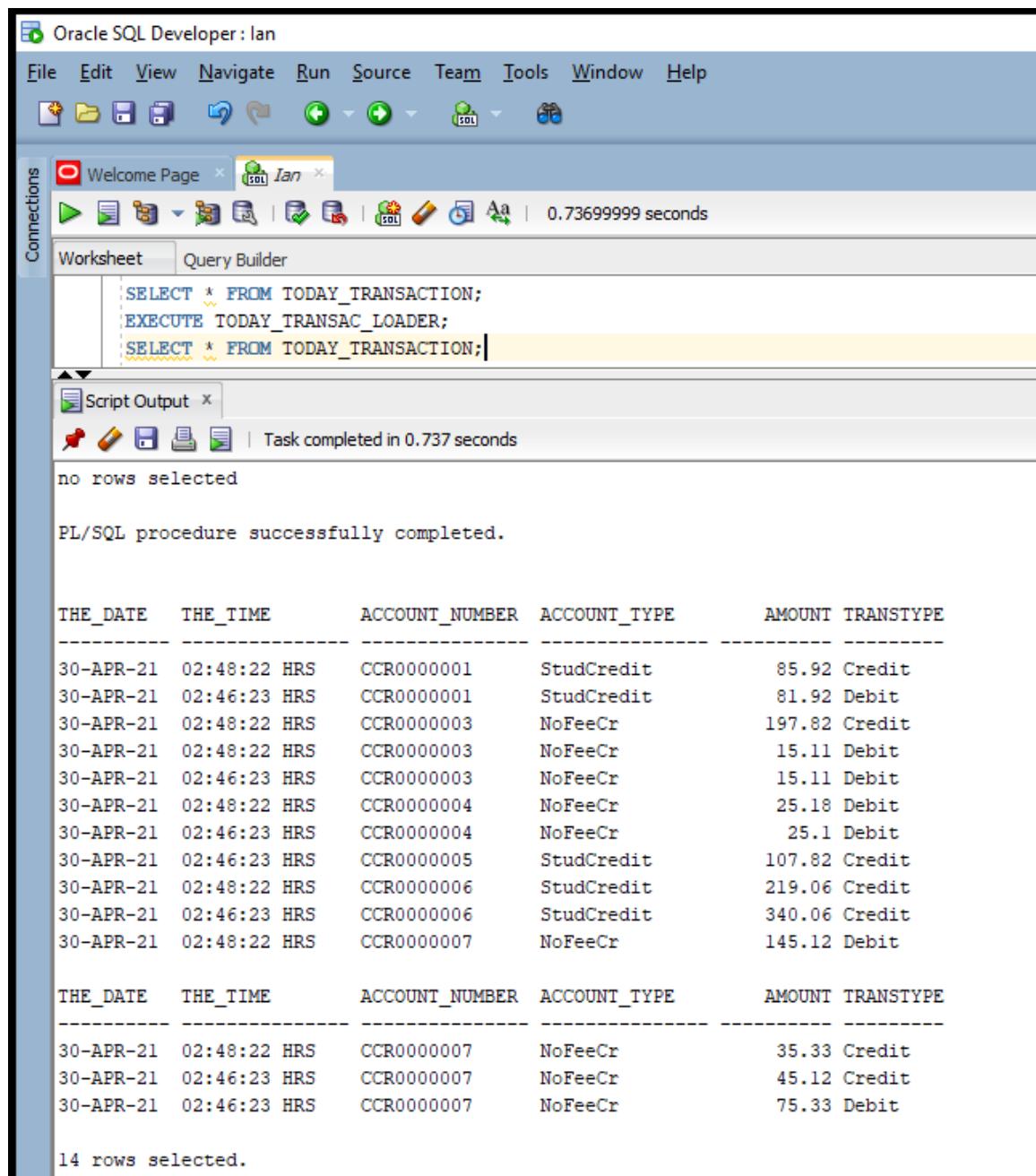
```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE TODAY_TRANSAC_LOADER
AS
BEGIN
  INSERT INTO Today_Transaction (THE_DATE, THE_TIME, ACCOUNT_NUMBER, ACCOUNT_TYPE, AMOUNT, TRANSTYPE)
  SELECT
    SUBSTR(cat.TRANSDATETIME, 0, 9) AS "DATE",
    ( SUBSTR(cat.TRANSDATETIME, 11, 2)||':'||)
    SUBSTR(cat.TRANSDATETIME, 14, 2)||':'||)
    SUBSTR(cat.TRANSDATETIME, 17, 2)||' HRS') AS "TIME",
    RPAD (ca.CREDITACCTNO, 15) AS "ACCOUNT_NUMBER",
    RPAD (ca.CREDITPRODID, 15) AS "ACCOUNT_TYPE",
    cat.AMOUNT,
    cat.TRANSTYPE
  FROM CRED_ACCT_TRANS cat, CRED_ACCOUNT ca
  WHERE SUBSTR(cat.TRANSDATETIME, 0, 9) = TO_CHAR(SYSDATE) AND
  ca.CREDITACCTNO = cat.CREDITACCTNO;
  EXCEPTION
    WHEN OTHERS THEN
      IF SQL$NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No such record was found');
      END IF;
      DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
  END;
  /

```

Script Output x | Task completed in 1.077 seconds

Procedure TODAY\_TRANSAC\_LOADER compiled

**Verify today\_transaction table is empty.**  
**Execute procedure to load today\_transaction table.**  
**Show new contents of today\_transaction table.**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
SELECT * FROM TODAY_TRANSACTION;
EXECUTE TODAY_TRANSAC_LOADER;
SELECT * FROM TODAY_TRANSACTION;
```

Script Output x

Task completed in 0.737 seconds

no rows selected

PL/SQL procedure successfully completed.

THE_DATE	THE_TIME	ACCOUNT_NUMBER	ACCOUNT_TYPE	AMOUNT	TRANSTYPE
30-APR-21	02:48:22 HRS	CCR0000001	StudCredit	85.92	Credit
30-APR-21	02:46:23 HRS	CCR0000001	StudCredit	81.92	Debit
30-APR-21	02:48:22 HRS	CCR0000003	NoFeeCr	197.82	Credit
30-APR-21	02:48:22 HRS	CCR0000003	NoFeeCr	15.11	Debit
30-APR-21	02:46:23 HRS	CCR0000003	NoFeeCr	15.11	Debit
30-APR-21	02:48:22 HRS	CCR0000004	NoFeeCr	25.18	Debit
30-APR-21	02:46:23 HRS	CCR0000004	NoFeeCr	25.1	Debit
30-APR-21	02:46:23 HRS	CCR0000005	StudCredit	107.82	Credit
30-APR-21	02:48:22 HRS	CCR0000006	StudCredit	219.06	Credit
30-APR-21	02:46:23 HRS	CCR0000006	StudCredit	340.06	Credit
30-APR-21	02:48:22 HRS	CCR0000007	NoFeeCr	145.12	Debit
THE_DATE	THE_TIME	ACCOUNT_NUMBER	ACCOUNT_TYPE	AMOUNT	TRANSTYPE
30-APR-21	02:48:22 HRS	CCR0000007	NoFeeCr	35.33	Credit
30-APR-21	02:46:23 HRS	CCR0000007	NoFeeCr	45.12	Credit
30-APR-21	02:46:23 HRS	CCR0000007	NoFeeCr	75.33	Debit

14 rows selected.

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3D Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

D. Create a function called CustomerInfo to accept a customer account number and return the total deposit for the customer account.

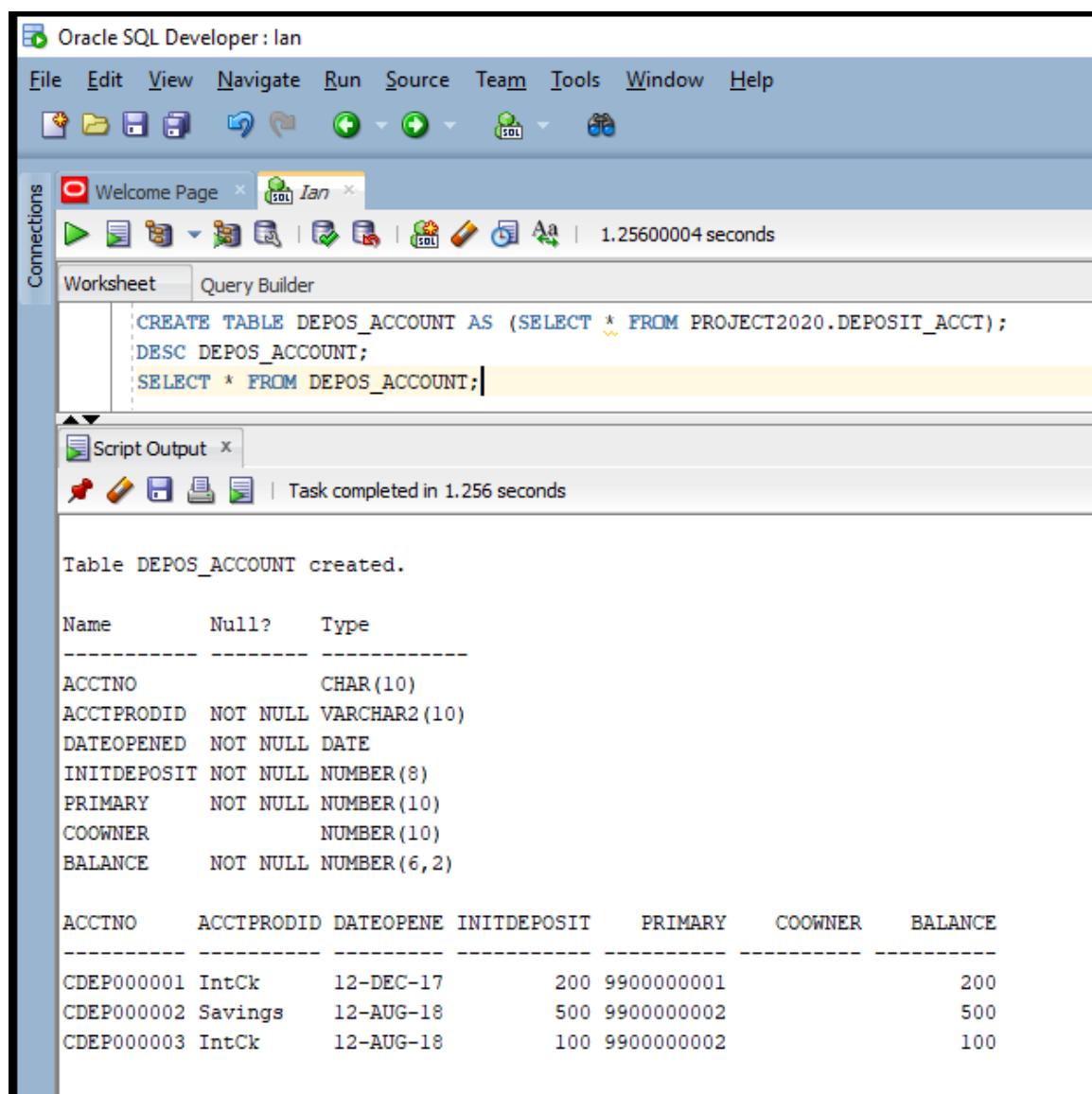
**PROCEDURE INPUT: CUSTOMER ACCOUNT NUMBER**

**PROCEDURE OUTPUT: TOTAL DEPOSIT**

**Create depos\_account table using table PROJECT2020.DEPOSIT\_ACCT.**

**Display attribute types of depos\_account table.**

**Display contents of depos\_account table.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE TABLE DEPOS_ACCOUNT AS (SELECT * FROM PROJECT2020.DEPOSIT_ACCT);
DESC DEPOS_ACCOUNT;
SELECT * FROM DEPOS_ACCOUNT;
```

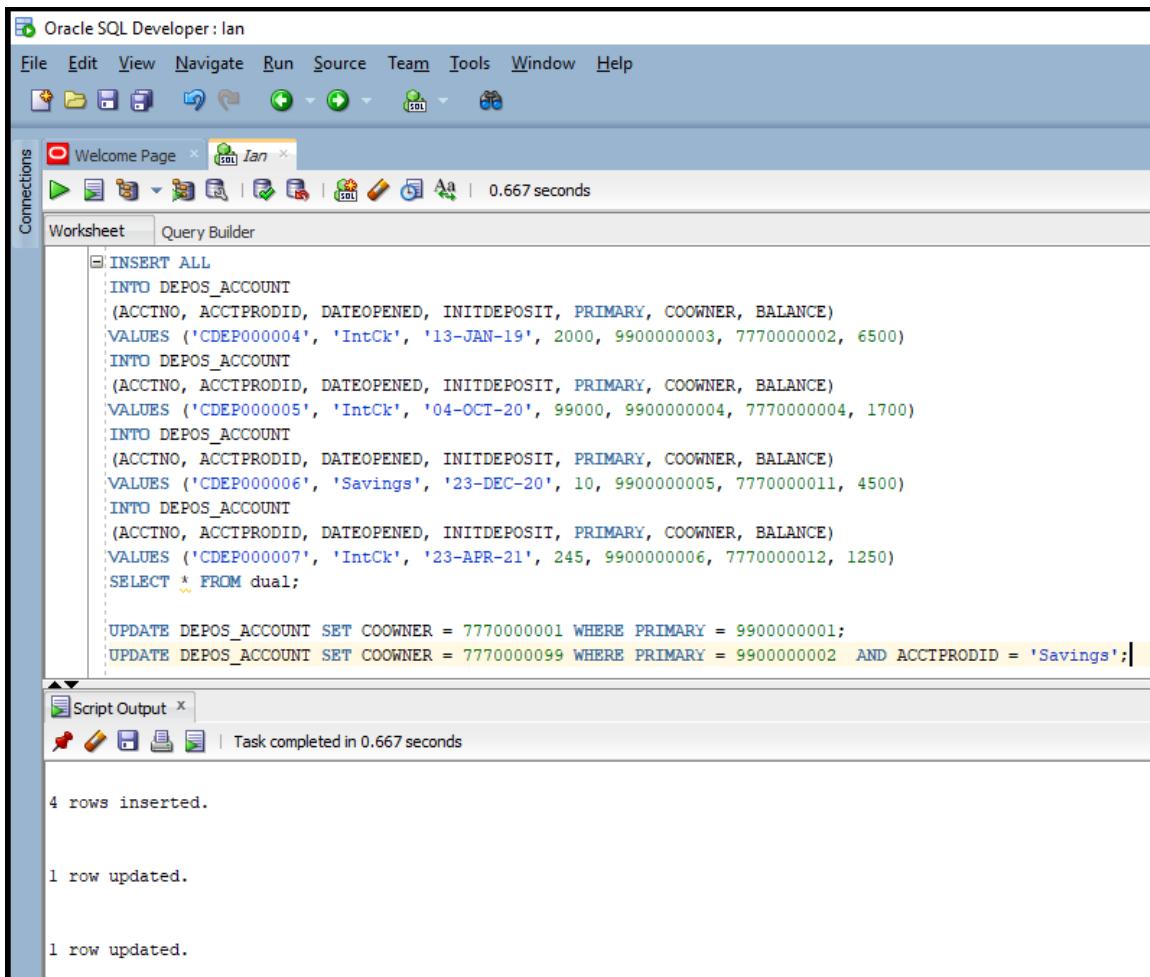
The 'Script Output' tab shows the results of the execution:

```
Table DEPOS_ACCOUNT created.

Name      Null?    Type
-----  -----
ACCTNO          CHAR(10)
ACCTPRODID    NOT NULL VARCHAR2(10)
DATEOPENED    NOT NULL DATE
INITDEPOSIT   NOT NULL NUMBER(8)
PRIMARY        NOT NULL NUMBER(10)
COOWNER        NUMBER(10)
BALANCE        NOT NULL NUMBER(6,2)

ACCTNO  ACCTPRODID DATEOPENED INITDEPOSIT  PRIMARY  COOWNER  BALANCE
-----  -----
CDEP000001  IntCk    12-DEC-17      200  9900000001          200
CDEP000002  Savings   12-AUG-18      500  9900000002          500
CDEP000003  IntCk    12-AUG-18      100  9900000002          100
```

**Load values into depos\_account table.  
Update entries in depos\_account table.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script. The script begins with an 'INSERT ALL' statement, inserting four rows into the 'DEPOS\_ACCOUNT' table. The first row is for an account with primary key '9900000003', opened on '13-JAN-19', with a balance of 6500. The second row is for an account with primary key '9900000004', opened on '04-OCT-20', with a balance of 1700. The third row is for an account with primary key '9900000005', opened on '23-DEC-20', with a balance of 4500. The fourth row is for an account with primary key '9900000006', opened on '23-APR-21', with a balance of 1250. Following the insertions, two UPDATE statements are executed: one setting the 'COOWNER' to 7770000001 for the primary key 9900000001, and another setting the 'COOWNER' to 7770000099 for the primary key 9900000002 and account type 'Savings'. The 'Script Output' tab shows the results of the execution: '4 rows inserted.', '1 row updated.', and '1 row updated.'.

```
INSERT ALL
INTO DEPOS_ACCOUNT
(ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
VALUES ('CDEP000004', 'IntCk', '13-JAN-19', 2000, 9900000003, 7770000002, 6500)
INTO DEPOS_ACCOUNT
(ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
VALUES ('CDEP000005', 'IntCk', '04-OCT-20', 99000, 9900000004, 7770000004, 1700)
INTO DEPOS_ACCOUNT
(ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
VALUES ('CDEP000006', 'Savings', '23-DEC-20', 10, 9900000005, 7770000011, 4500)
INTO DEPOS_ACCOUNT
(ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
VALUES ('CDEP000007', 'IntCk', '23-APR-21', 245, 9900000006, 7770000012, 1250)
SELECT * FROM dual;

UPDATE DEPOS_ACCOUNT SET COOWNER = 7770000001 WHERE PRIMARY = 9900000001;
UPDATE DEPOS_ACCOUNT SET COOWNER = 7770000099 WHERE PRIMARY = 9900000002 AND ACCTPRODID = 'Savings';|
```

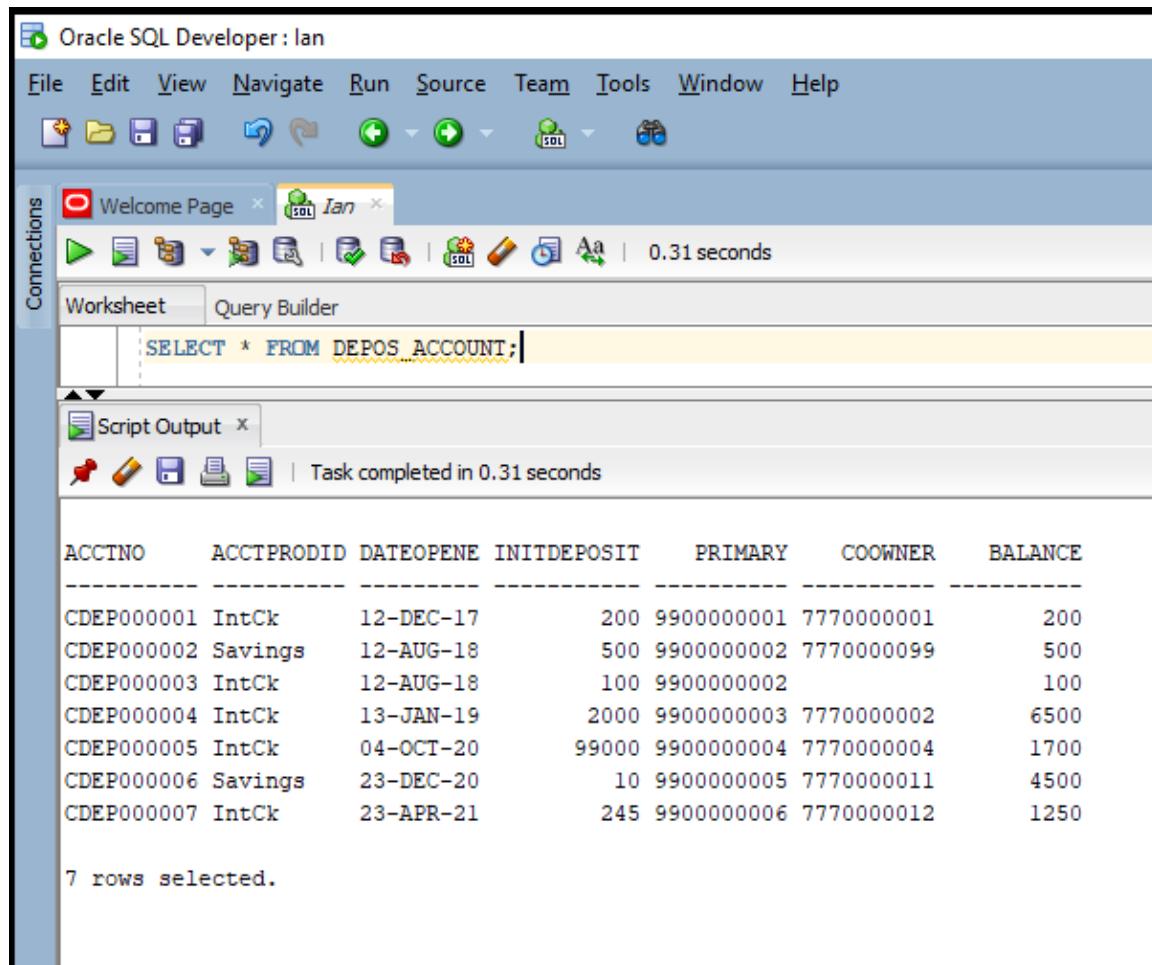
Script Output | Task completed in 0.667 seconds

4 rows inserted.

1 row updated.

1 row updated.

## Show contents of depos\_account table.



The screenshot shows the Oracle SQL Developer interface with a single connection named 'Ian'. The 'Worksheet' tab is active, displaying the following SQL query:

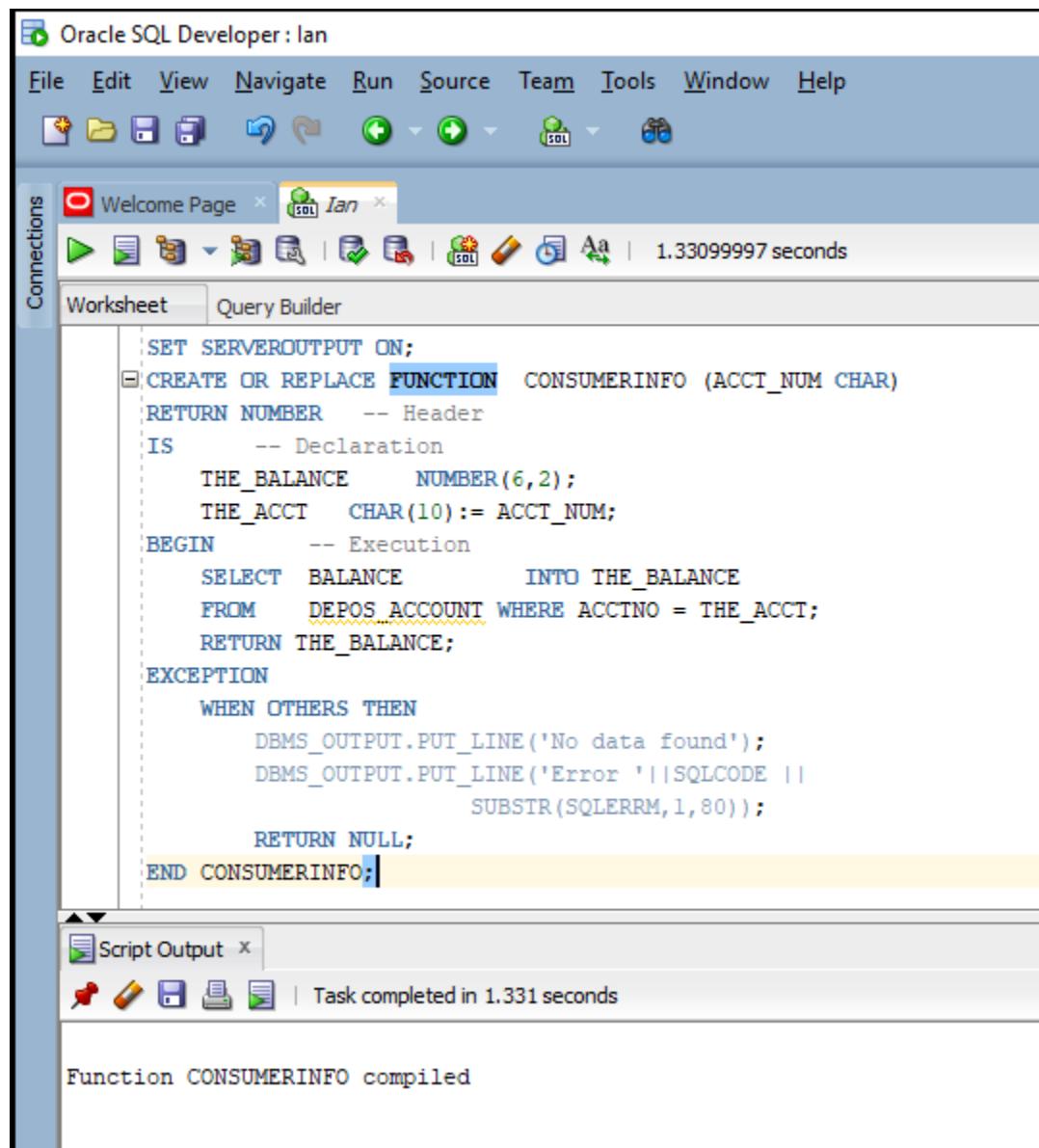
```
SELECT * FROM DEPOS_ACCOUNT;
```

The 'Script Output' tab shows the results of the query:

ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

Below the table, the message '7 rows selected.' is displayed.

**Create procedure to return account total balance from an account number in depos\_account table with exception handling.**

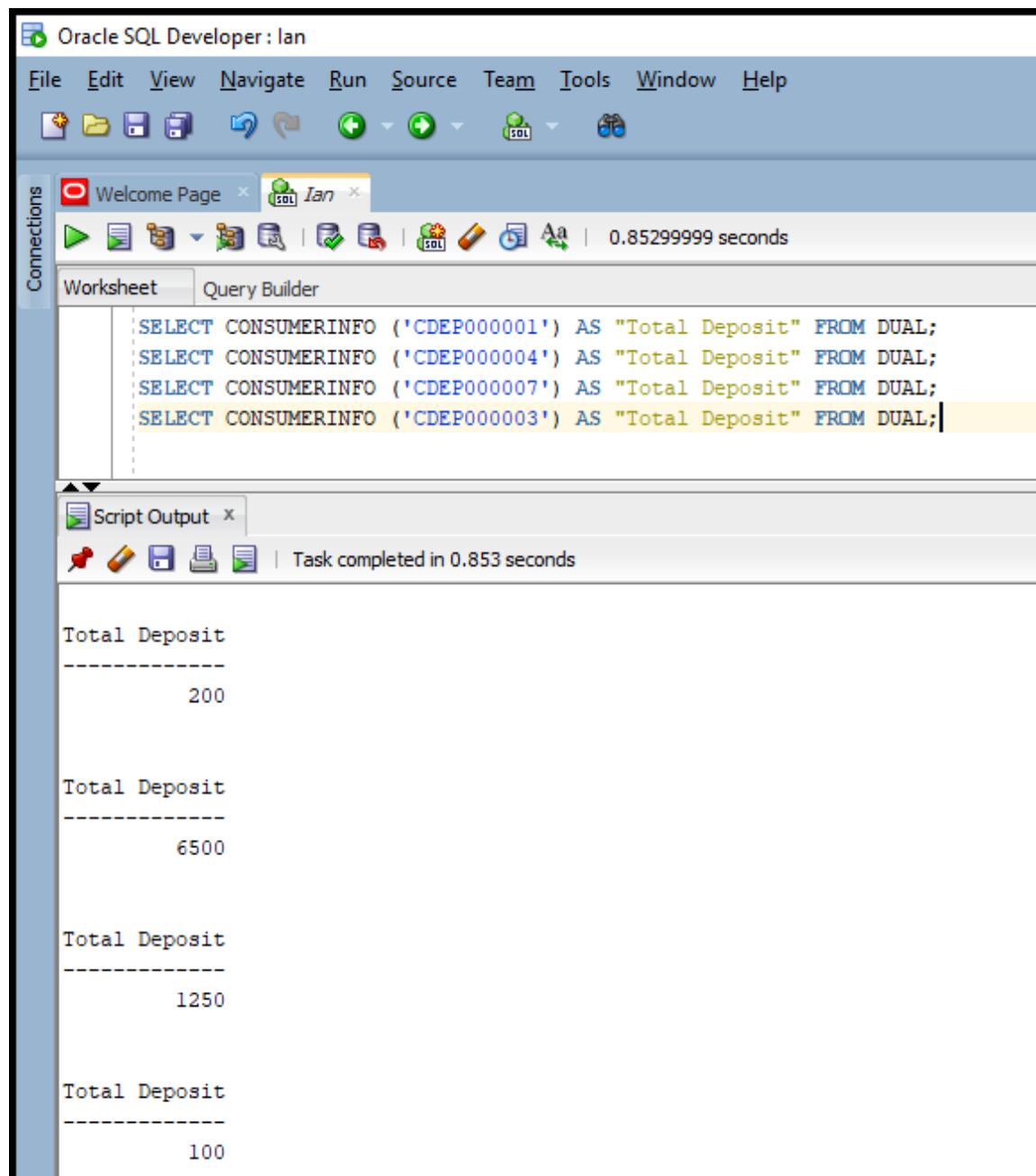


```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION CONSUMERINFO (ACCT_NUM CHAR)
  RETURN NUMBER -- Header
  IS -- Declaration
    THE_BALANCE    NUMBER(6,2);
    THE_ACCT      CHAR(10):= ACCT_NUM;
  BEGIN -- Execution
    SELECT BALANCE      INTO THE_BALANCE
    FROM DEPOS_ACCOUNT WHERE ACCTNO = THE_ACCT;
    RETURN THE_BALANCE;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('No data found');
      DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
      RETURN NULL;
  END CONSUMERINFO;
```

Script Output | Task completed in 1.331 seconds

Function CONSUMERINFO compiled

**Execute function to select an entry's account balance with “DUAL” used to represent a dummy table.**



The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main workspace has a 'Connections' sidebar on the left. A 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT CONSUMERINFO ('CDEP000001') AS "Total Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000004') AS "Total Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000007') AS "Total Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000003') AS "Total Deposit" FROM DUAL;
```

The 'Script Output' pane below shows the results of the execution:

```
Total Deposit
-----
200

Total Deposit
-----
6500

Total Deposit
-----
1250

Total Deposit
-----
100
```

The output pane also indicates "Task completed in 0.853 seconds".

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3E Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

E. Create a function with the same name CustomerInfo (overload) to accept a customer account number and a date; and return the total deposit of that customer account for that date.

**PROCEDURE INPUT: CUSTOMER ACCOUNT NUMBER, DATE**

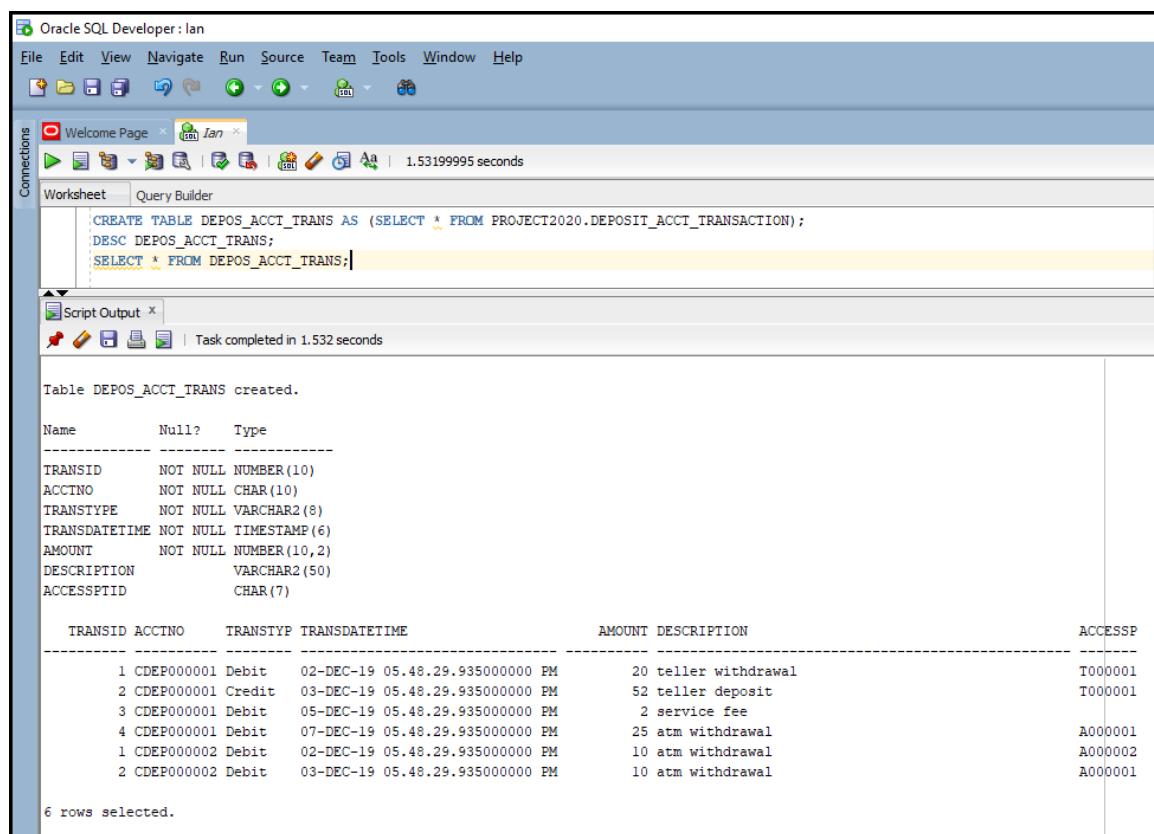
**PROCEDURE OUTPUT: TOTAL DEPOSIT on supplied DATE**

**Create depos\_acct\_trans table using table**

**PROJECT2020.DEPOSIT\_ACCT\_TRANSACTION.**

**Display attribute data types of depos\_acct\_trans table.**

**Display contents of depos\_acct\_trans table.**



The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet:** Contains the SQL code for creating the table and selecting from it:

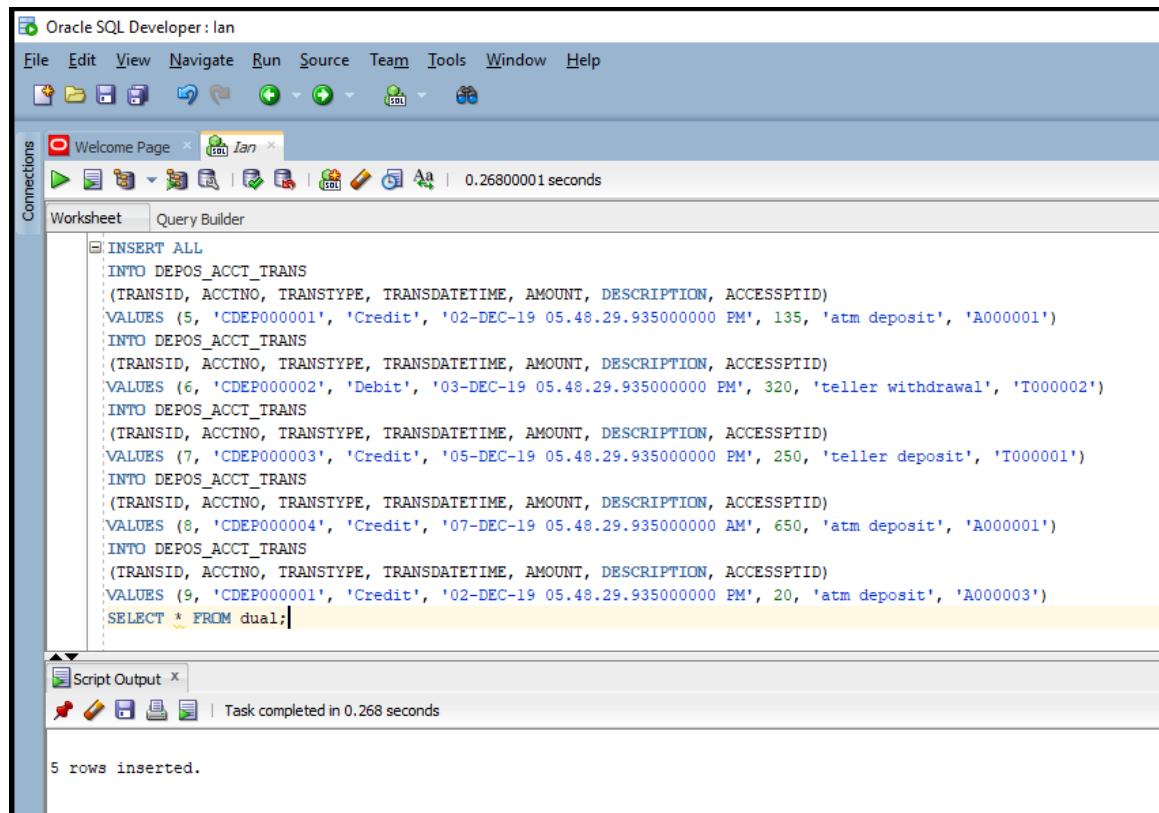
```
CREATE TABLE DEPOS_ACCT_TRANS AS (SELECT * FROM PROJECT2020.DEPOSIT_ACCT_TRANSACTION);
DESC DEPOS_ACCT_TRANS;
SELECT * FROM DEPOS_ACCT_TRANS;
```
- Script Output:** Shows the results of the creation and population of the table:

```
Table DEPOS_ACCT_TRANS created.

Name Null? Type
-----
TRANSID NOT NULL NUMBER(10)
ACCTNO NOT NULL CHAR(10)
TRANSTYP NOT NULL VARCHAR2(8)
TRANSDATETIME NOT NULL TIMESTAMP(6)
AMOUNT NOT NULL NUMBER(10,2)
DESCRIPTION VARCHAR2(50)
ACCESSPTID CHAR(7)

TRANSID ACCTNO TRANSTYP TRANSDATETIME AMOUNT DESCRIPTION ACCESSPTID
-----
1 CDEP00001 Debit 02-DEC-19 05.48.29.935000000 PM 20 teller withdrawal T000001
2 CDEP00001 Credit 03-DEC-19 05.48.29.935000000 PM 52 teller deposit T000001
3 CDEP00001 Debit 05-DEC-19 05.48.29.935000000 PM 2 service fee
4 CDEP00001 Debit 07-DEC-19 05.48.29.935000000 PM 25 atm withdrawal A000001
1 CDEP00002 Debit 02-DEC-19 05.48.29.935000000 PM 10 atm withdrawal A000002
2 CDEP00002 Debit 03-DEC-19 05.48.29.935000000 PM 10 atm withdrawal A000001
```
- Message:** Shows "6 rows selected."

## Load values into depos\_acct\_trans table.



Oracle SQL Developer : Ian

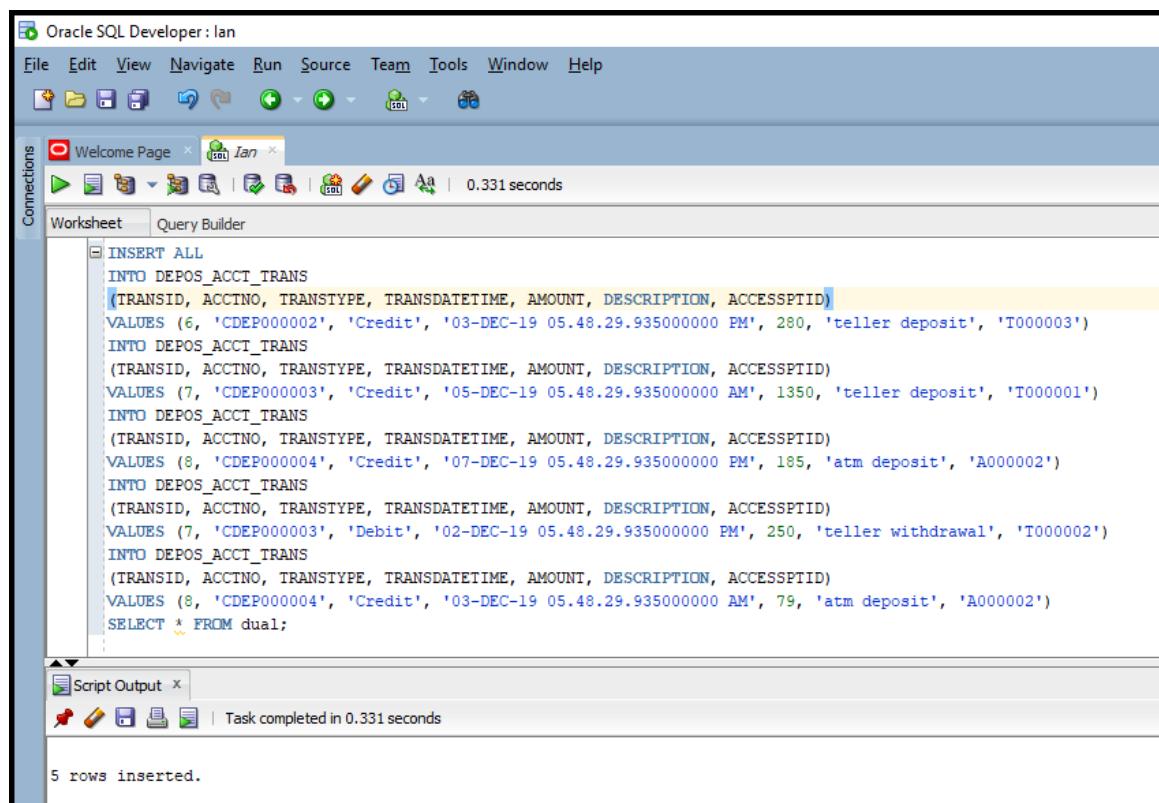
File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (5, 'CDEP000001', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 135, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000002', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 320, 'teller withdrawal', 'T000002')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000003', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 250, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000004', 'Credit', '07-DEC-19 05.48.29.935000000 AM', 650, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (9, 'CDEP000001', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 20, 'atm deposit', 'A000003')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.268 seconds

5 rows inserted.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

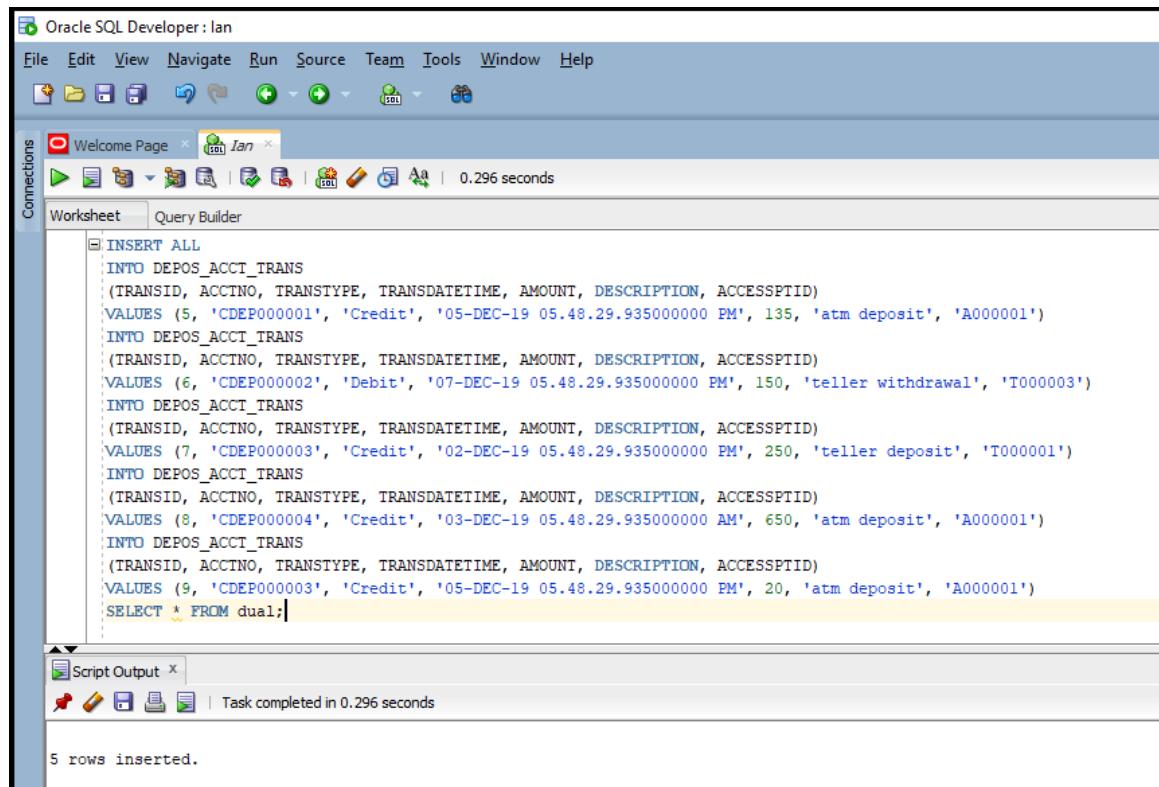
Connections Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000002', 'Credit', '03-DEC-19 05.48.29.935000000 PM', 280, 'teller deposit', 'T000003')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000003', 'Credit', '05-DEC-19 05.48.29.935000000 AM', 1350, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000004', 'Credit', '07-DEC-19 05.48.29.935000000 PM', 185, 'atm deposit', 'A000002')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000003', 'Debit', '02-DEC-19 05.48.29.935000000 PM', 250, 'teller withdrawal', 'T000002')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000004', 'Credit', '03-DEC-19 05.48.29.935000000 AM', 79, 'atm deposit', 'A000002')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.331 seconds

5 rows inserted.

## Load still more values into depos\_acct\_trans table.



Oracle SQL Developer: Ian

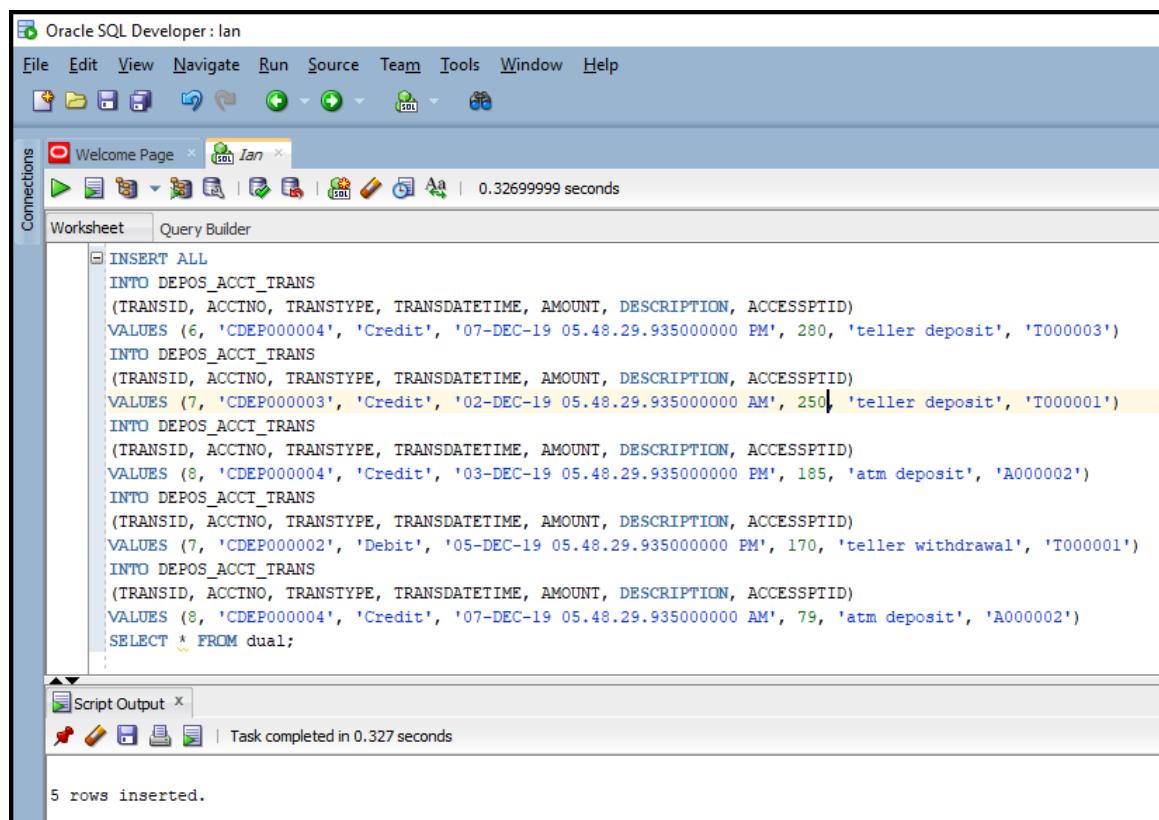
File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (5, 'CDEP000001', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 135, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000002', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 150, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000003', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 250, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000004', 'Credit', '03-DEC-19 05.48.29.935000000 AM', 650, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (9, 'CDEP000003', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 20, 'atm deposit', 'A000001')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.296 seconds

5 rows inserted.



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

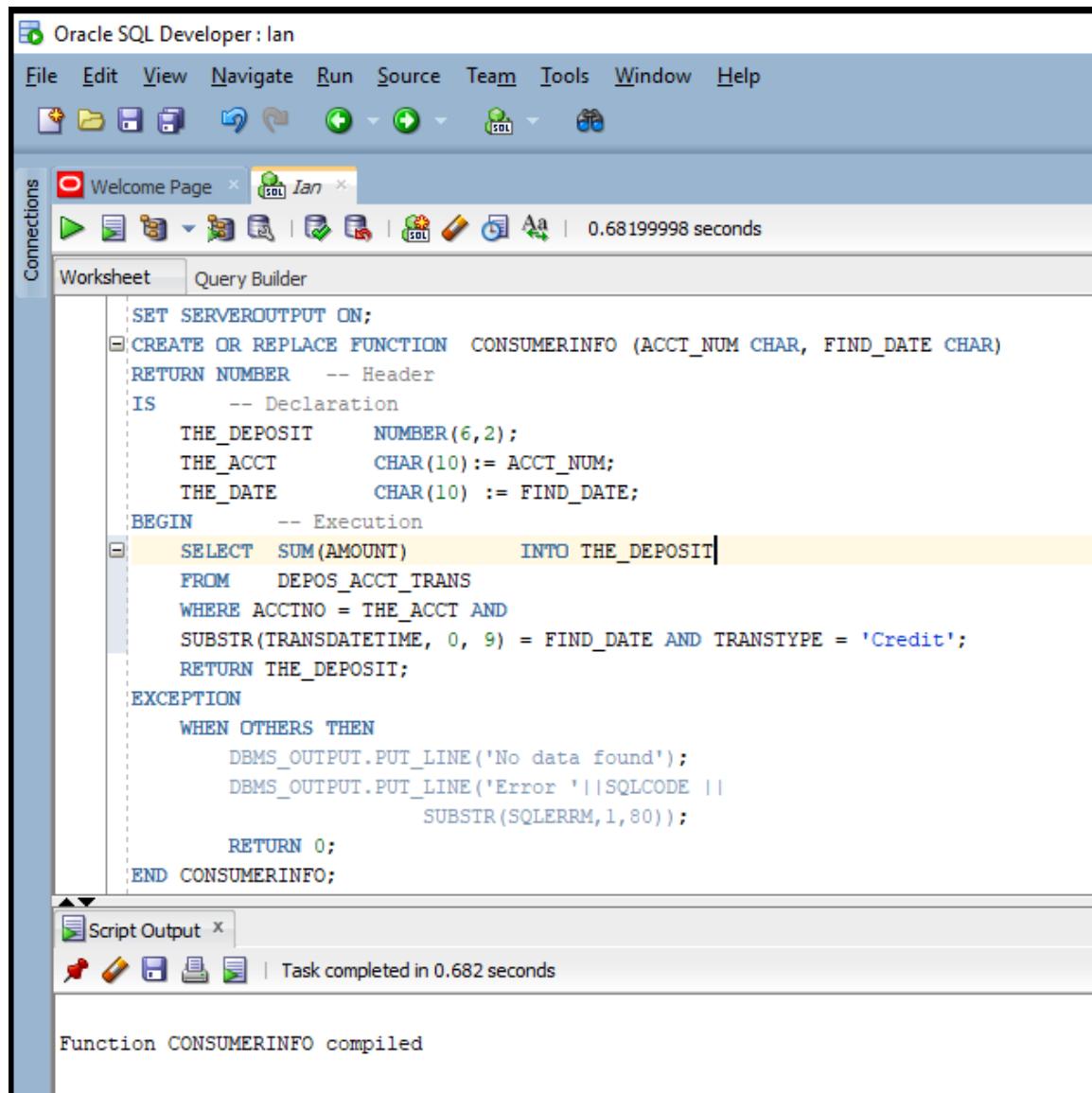
Connections Welcome Page Jan Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000004', 'Credit', '07-DEC-19 05.48.29.935000000 PM', 280, 'teller deposit', 'T000003')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000003', 'Credit', '02-DEC-19 05.48.29.935000000 AM', 250, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000004', 'Credit', '03-DEC-19 05.48.29.935000000 PM', 185, 'atm deposit', 'A000002')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000002', 'Debit', '05-DEC-19 05.48.29.935000000 PM', 170, 'teller withdrawal', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000004', 'Credit', '07-DEC-19 05.48.29.935000000 AM', 79, 'atm deposit', 'A000002')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.327 seconds

5 rows inserted.

**Create procedure to return account total balance from an account number on given date in depos\_acct\_trans table with exception handling.**



```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION CONSUMERINFO (ACCT_NUM CHAR, FIND_DATE CHAR)
RETURN NUMBER -- Header
IS -- Declaration
    THE_DEPOSIT    NUMBER(6,2);
    THE_ACCT      CHAR(10):= ACCT_NUM;
    THE_DATE      CHAR(10) := FIND_DATE;
BEGIN -- Execution
    SELECT SUM(AMOUNT)      INTO THE_DEPOSIT
    FROM DEPOS_ACCT_TRANS
    WHERE ACCTNO = THE_ACCT AND
    SUBSTR(TRANSDATETIME, 0, 9) = FIND_DATE AND TRANSTYPE = 'Credit';
    RETURN THE_DEPOSIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN 0;
END CONSUMERINFO;
```

Script Output | Task completed in 0.682 seconds

Function CONSUMERINFO compiled

## Show contents of depos\_acct\_trans table then execute function.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 0.88 seconds

Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '02-DEC-19';
SELECT CONSUMERINFO ('CDEP000004', '02-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000002', '02-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000003', '02-DEC-19') AS "Date Deposit" FROM DUAL;
```

Script Output x | Task completed in 0.88 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000000 PM	20	teller withdrawal	T000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000002
5	CDEP000001	Credit	02-DEC-19 05.48.29.935000000 PM	135	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.935000000 PM	20	atm deposit	A000003
7	CDEP000003	Debit	02-DEC-19 05.48.29.935000000 PM	250	teller withdrawal	T000002
7	CDEP000003	Credit	02-DEC-19 05.48.29.935000000 PM	250	teller deposit	T000001
7	CDEP000003	Credit	02-DEC-19 05.48.29.935000000 AM	250	teller deposit	T000001

7 rows selected.

Date Deposit

-----

Date Deposit

-----

Date Deposit

-----

500

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 1.1239999 seconds

Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '03-DEC-19';
SELECT CONSUMERINFO ('CDEP000001', '03-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000003', '03-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000002', '03-DEC-19') AS "Date Deposit" FROM DUAL;
```

Script Output x | Task completed in 1.124 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000000 PM	52	teller deposit	T000001
2	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000001
6	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	320	teller withdrawal	T000002
6	CDEP000002	Credit	03-DEC-19 05.48.29.935000000 PM	280	teller deposit	T000003
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 AM	650	atm deposit	A000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002

7 rows selected.

Date Deposit

-----

52

Date Deposit

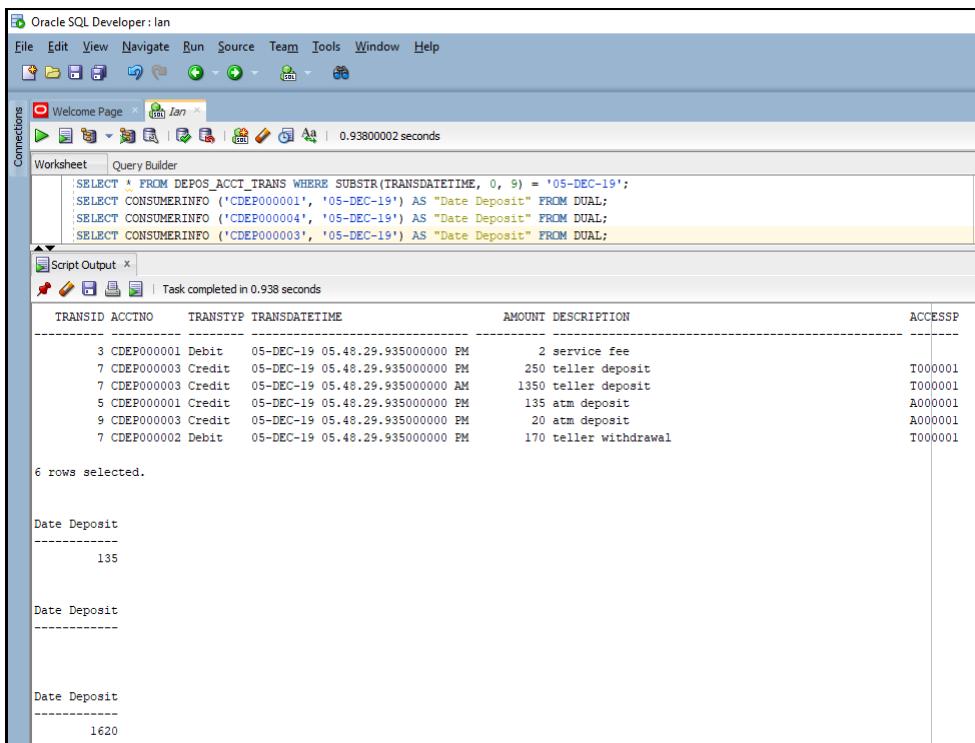
-----

Date Deposit

-----

280

## Execute function.



The screenshot shows the Oracle SQL Developer interface with a query window open. The query window displays a script and its execution results.

**Script:**

```
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '05-DEC-19';
SELECT CONSUMERINFO ('CDEP000001', '05-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000004', '05-DEC-19') AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO ('CDEP000003', '05-DEC-19') AS "Date Deposit" FROM DUAL;
```

**Execution Results:**

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
3	CDEP000001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	T000001
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	250	teller deposit	T000001
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 AM	1350	teller deposit	A000001
5	CDEP000001	Credit	05-DEC-19 05.48.29.93500000 PM	135	atm deposit	A000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	20	atm deposit	A000001
7	CDEP000002	Debit	05-DEC-19 05.48.29.93500000 PM	170	teller withdrawal	T000001

6 rows selected.

Date Deposit

-----

135

Date Deposit

-----

Date Deposit

-----

1620

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3F Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

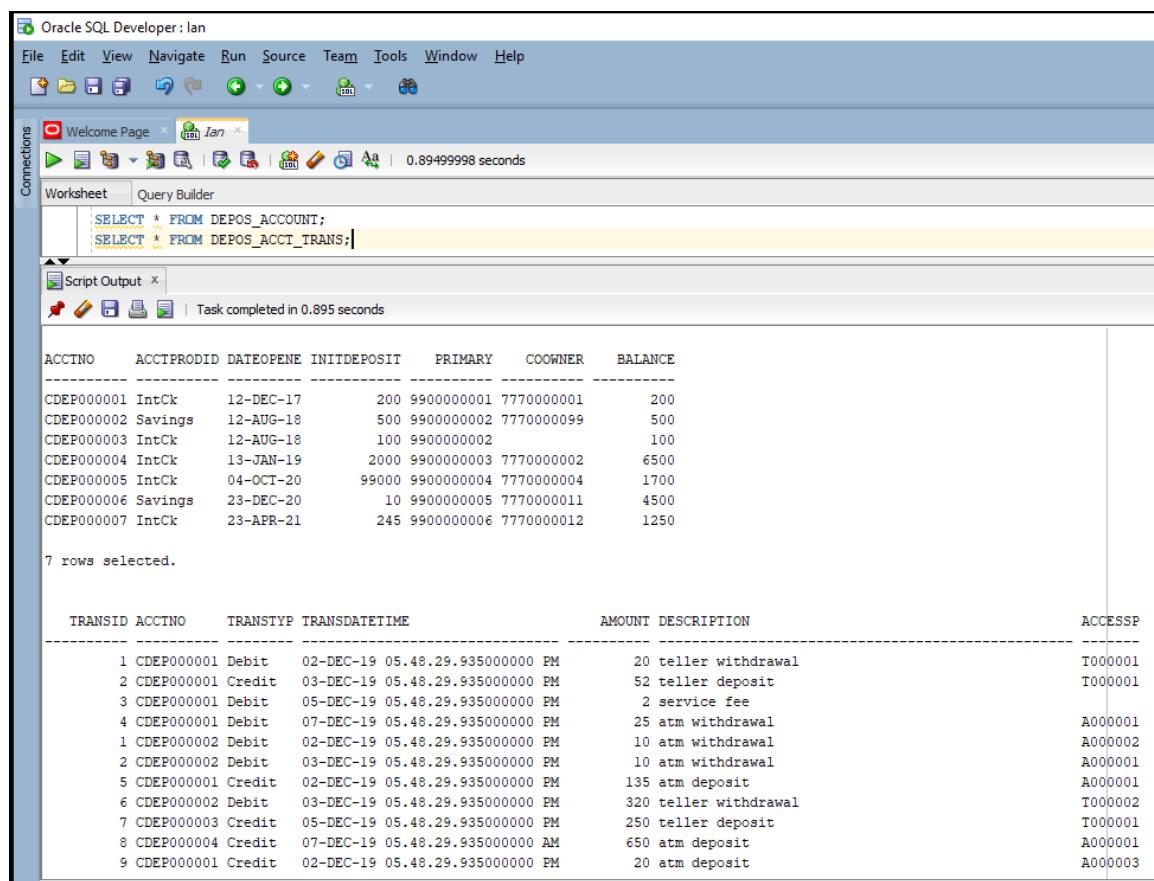
F. Create a function with the same name CustomerInfo (overload) to accept a customer ID, a date and a co-owner account number, and return the total deposit of that joint account for that date.

=> NOTE: Assumed Customer Id = Primary

**IN: CUSTOMER ID, DATE, COOWNER ACCOUNT NUMBER**

**OUT : TOTAL DEPOSIT on supplied DATE**

### Display relevant table contents



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains the following SQL code:

```
SELECT * FROM DEPOS_ACCOUNT;
SELECT * FROM DEPOS_ACCT_TRANS;
```

The 'Script Output' tab shows the results of the first query:

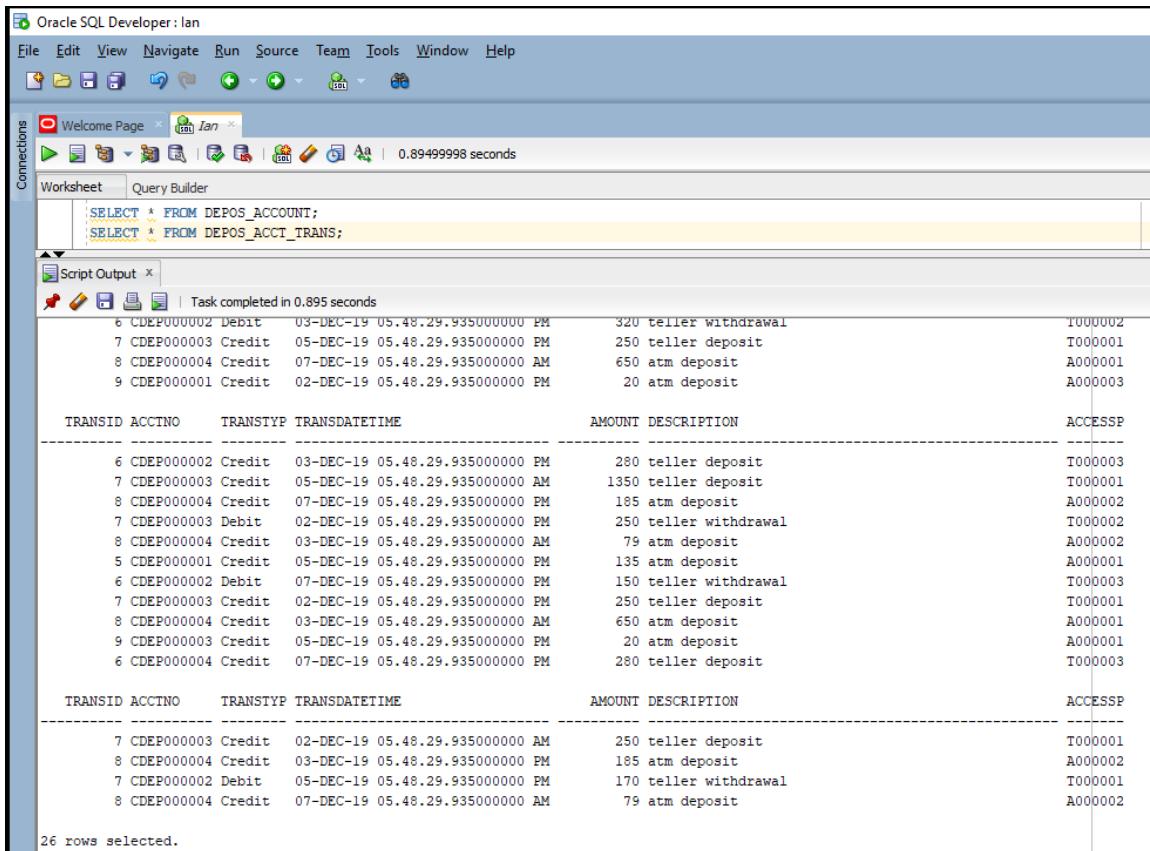
ACCTNO	ACCTPRODID	DATEOPENED	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

7 rows selected.

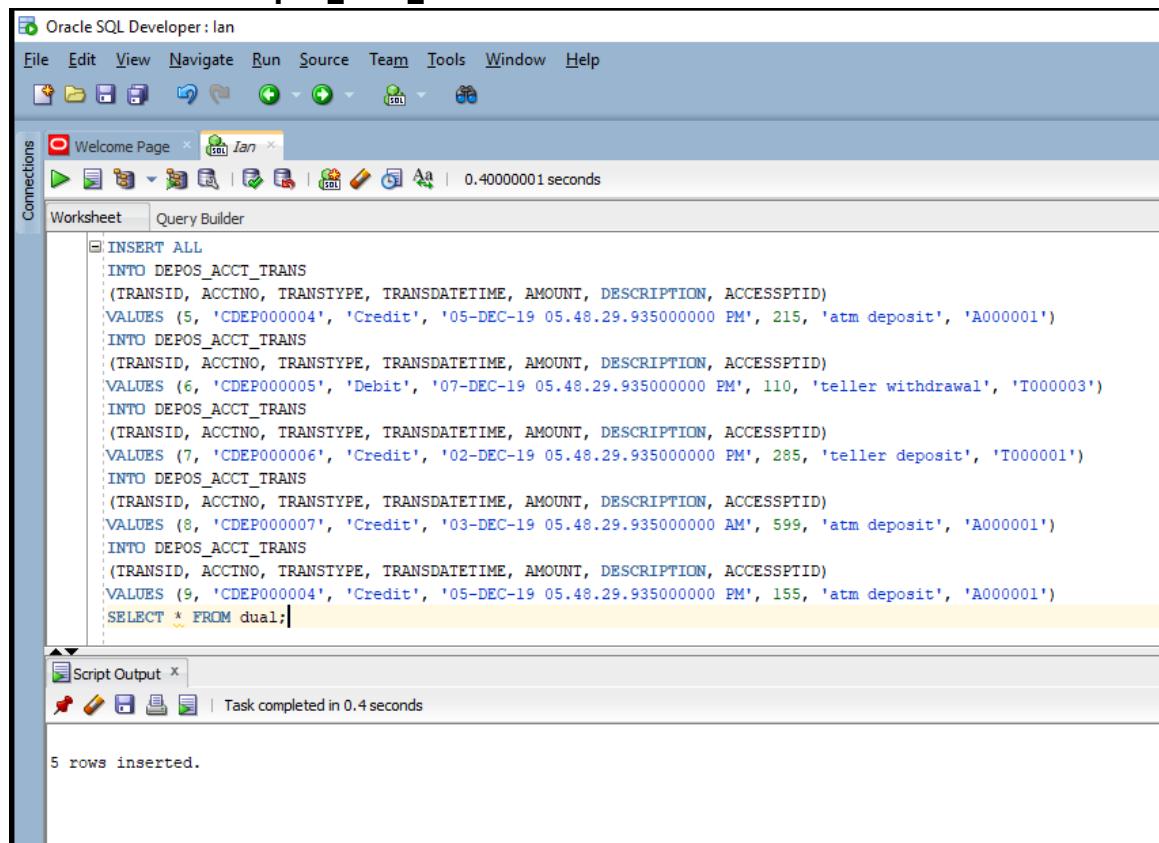
The 'Script Output' tab also shows the results of the second query:

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.93500000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.93500000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001
5	CDEP000001	Credit	02-DEC-19 05.48.29.93500000 PM	135	atm deposit	A000001
6	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	320	teller withdrawal	T000002
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.93500000 AM	650	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.93500000 PM	20	atm deposit	A000003

## Display relevant table contents



## Add entries to depos\_acct\_trans table.



Oracle SQL Developer : Ian

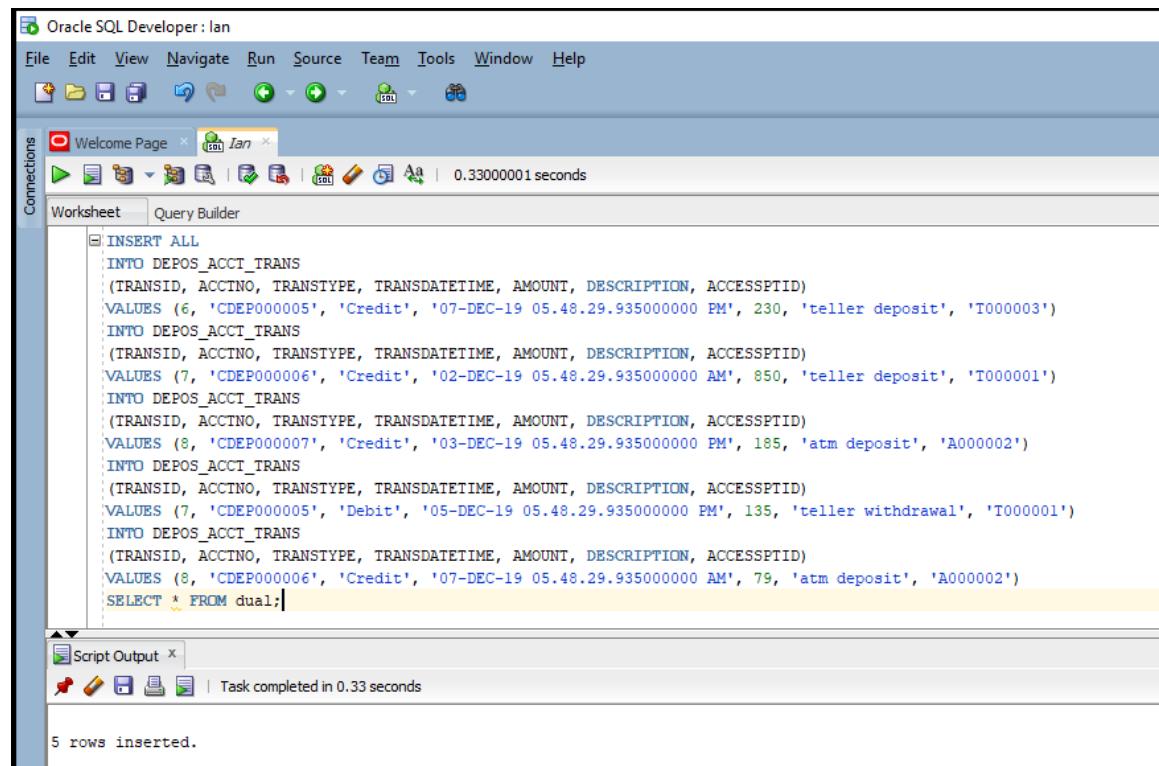
File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (5, 'CDEP000004', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 215, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000005', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 110, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000006', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 285, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000007', 'Credit', '03-DEC-19 05.48.29.935000000 AM', 599, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (9, 'CDEP000004', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 155, 'atm deposit', 'A000001')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.4 seconds

5 rows inserted.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

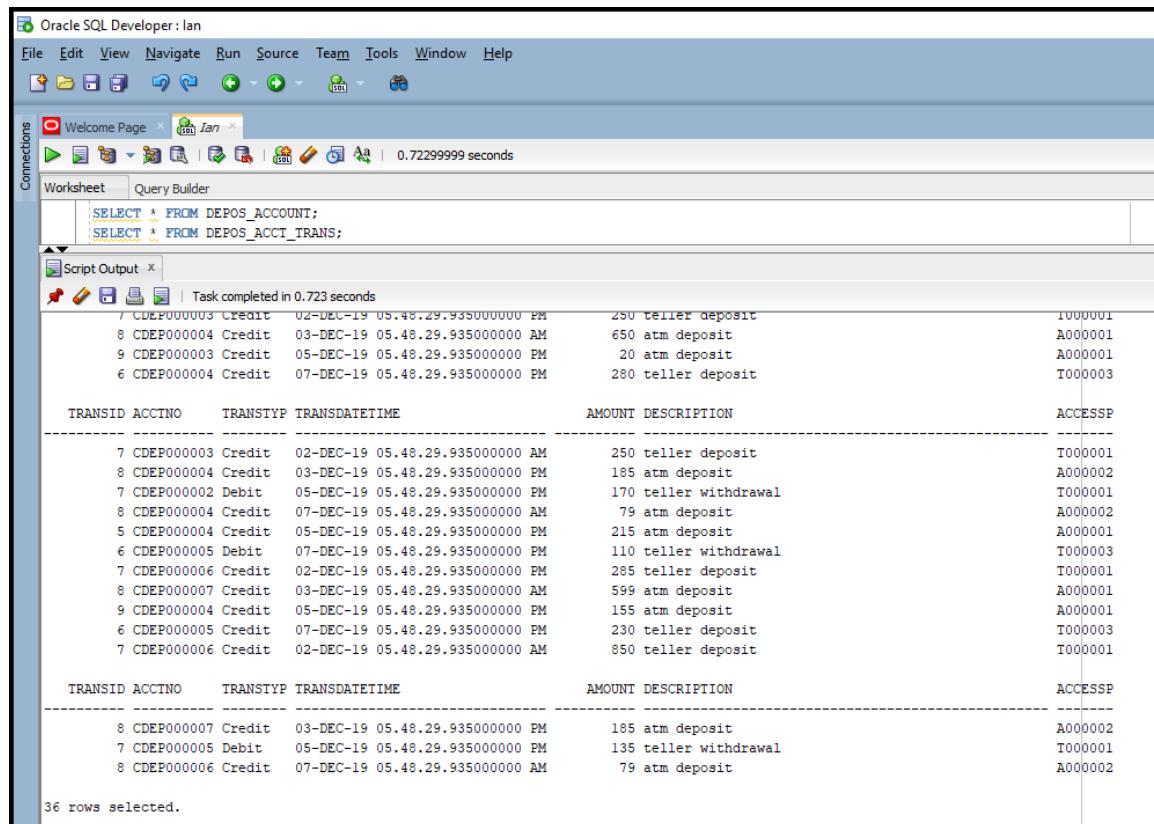
Connections Worksheet Query Builder

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (6, 'CDEP000005', 'Credit', '07-DEC-19 05.48.29.935000000 PM', 230, 'teller deposit', 'T000003')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000006', 'Credit', '02-DEC-19 05.48.29.935000000 AM', 850, 'teller deposit', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000007', 'Credit', '03-DEC-19 05.48.29.935000000 PM', 185, 'atm deposit', 'A000002')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (7, 'CDEP000005', 'Debit', '05-DEC-19 05.48.29.935000000 PM', 135, 'teller withdrawal', 'T000001')
  INTO DEPOS_ACCT_TRANS
  (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
  VALUES (8, 'CDEP000006', 'Credit', '07-DEC-19 05.48.29.935000000 AM', 79, 'atm deposit', 'A000002')
  SELECT * FROM dual;
```

Script Output | Task completed in 0.33 seconds

5 rows inserted.

## Add entries to depos\_acct\_trans table.



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCOUNT;
SELECT * FROM DEPOS_ACCT_TRANS;
```

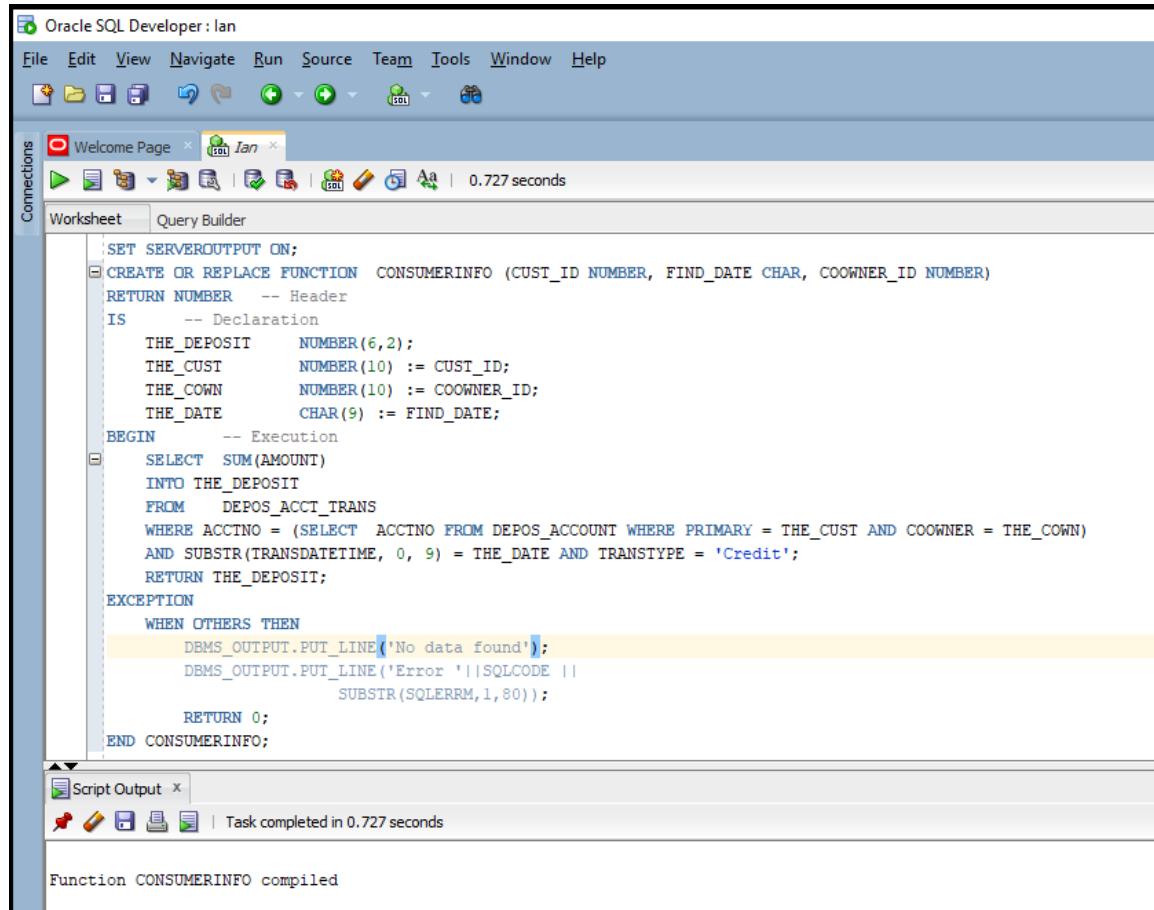
Script Output x

Task completed in 0.723 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
7	CDEP000003	Credit	02-DEC-19 05.48.29.93500000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.93500000 AM	650	atm deposit	A000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	20	atm deposit	A000001
6	CDEP000004	Credit	07-DEC-19 05.48.29.93500000 PM	280	teller deposit	T000003
7	CDEP000003	Credit	02-DEC-19 05.48.29.93500000 AM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002
7	CDEP000002	Debit	05-DEC-19 05.48.29.93500000 PM	170	teller withdrawal	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.93500000 AM	79	atm deposit	A000002
5	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	215	atm deposit	A000001
6	CDEP000005	Debit	07-DEC-19 05.48.29.93500000 PM	110	teller withdrawal	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 PM	285	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 AM	599	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	155	atm deposit	A000001
6	CDEP000005	Credit	07-DEC-19 05.48.29.93500000 PM	230	teller deposit	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 AM	850	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002
7	CDEP000005	Debit	05-DEC-19 05.48.29.93500000 PM	135	teller withdrawal	T000001
8	CDEP000006	Credit	07-DEC-19 05.48.29.93500000 AM	79	atm deposit	A000002

36 rows selected.

## Create procedure



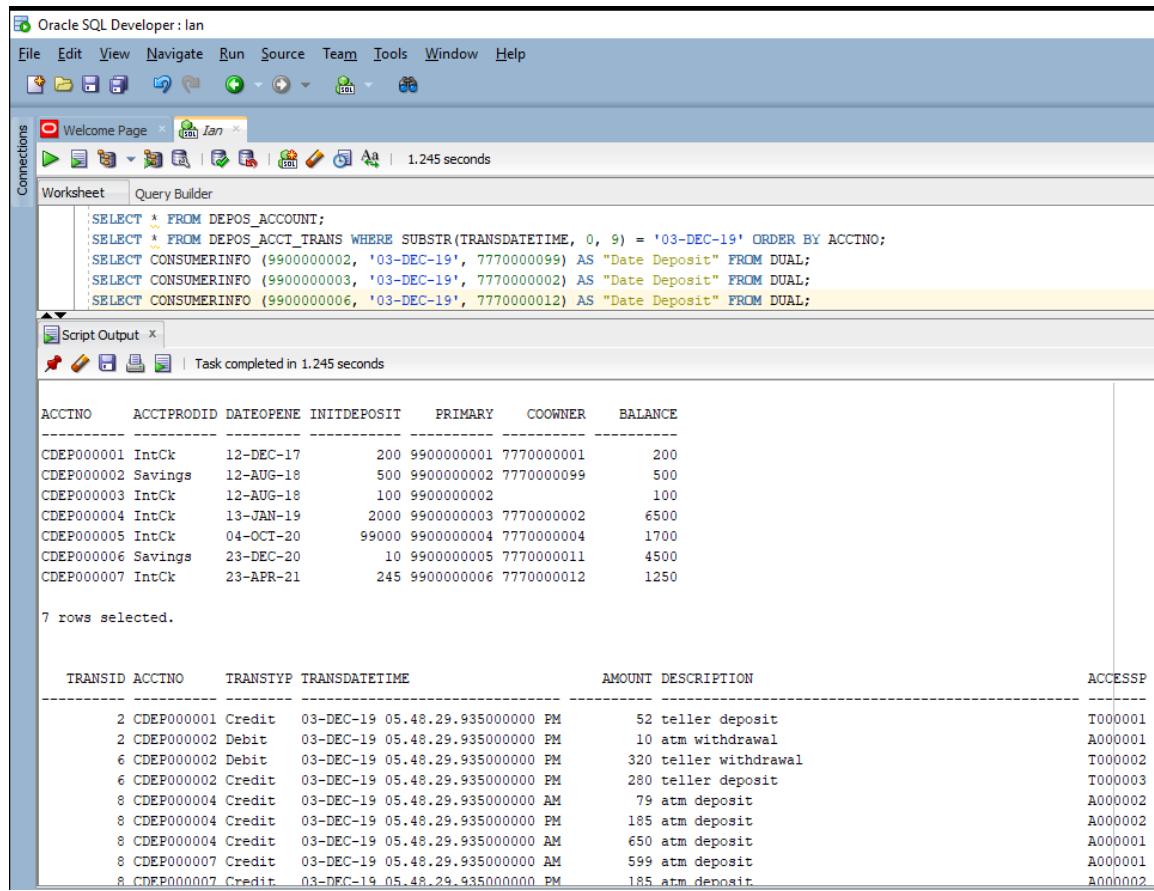
The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying a PL/SQL script to create a function named 'CONSUMERINFO'. The script includes declarations for variables and cursors, an execution block with a SELECT statement, and an exception handling block that outputs error messages to the script output window. The script is successfully compiled, as indicated by the message 'Function CONSUMERINFO compiled' in the output window.

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION CONSUMERINFO (CUST_ID NUMBER, FIND_DATE CHAR, COOWNER_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_DEPOSIT    NUMBER(6,2);
    THE_CUST       NUMBER(10) := CUST_ID;
    THE_COWN       NUMBER(10) := COOWNER_ID;
    THE_DATE       CHAR(9)  := FIND_DATE;
BEGIN -- Execution
    SELECT SUM(AMOUNT)
    INTO THE_DEPOSIT
    FROM DEPOS_ACCT_TRANS
    WHERE ACCTNO = (SELECT ACCTNO FROM DEPOS_ACCOUNT WHERE PRIMARY = THE_CUST AND COOWNER = THE_COWN)
    AND SUBSTR(TRANSDATETIME, 0, 9) = THE_DATE AND TRANSTYPE = 'Credit';
    RETURN THE_DEPOSIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE||SUBSTR(SQLERRM,1,80));
    RETURN 0;
END CONSUMERINFO;
```

Script Output | Task completed in 0.727 seconds

Function CONSUMERINFO compiled

## Execute procedure



The screenshot shows the Oracle SQL Developer interface with a query window open. The query is as follows:

```
SELECT * FROM DEPOS_ACCOUNT;
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '03-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (990000002, '03-DEC-19', 7770000099) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000003, '03-DEC-19', 7770000002) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000006, '03-DEC-19', 7770000012) AS "Date Deposit" FROM DUAL;
```

The results of the first query are:

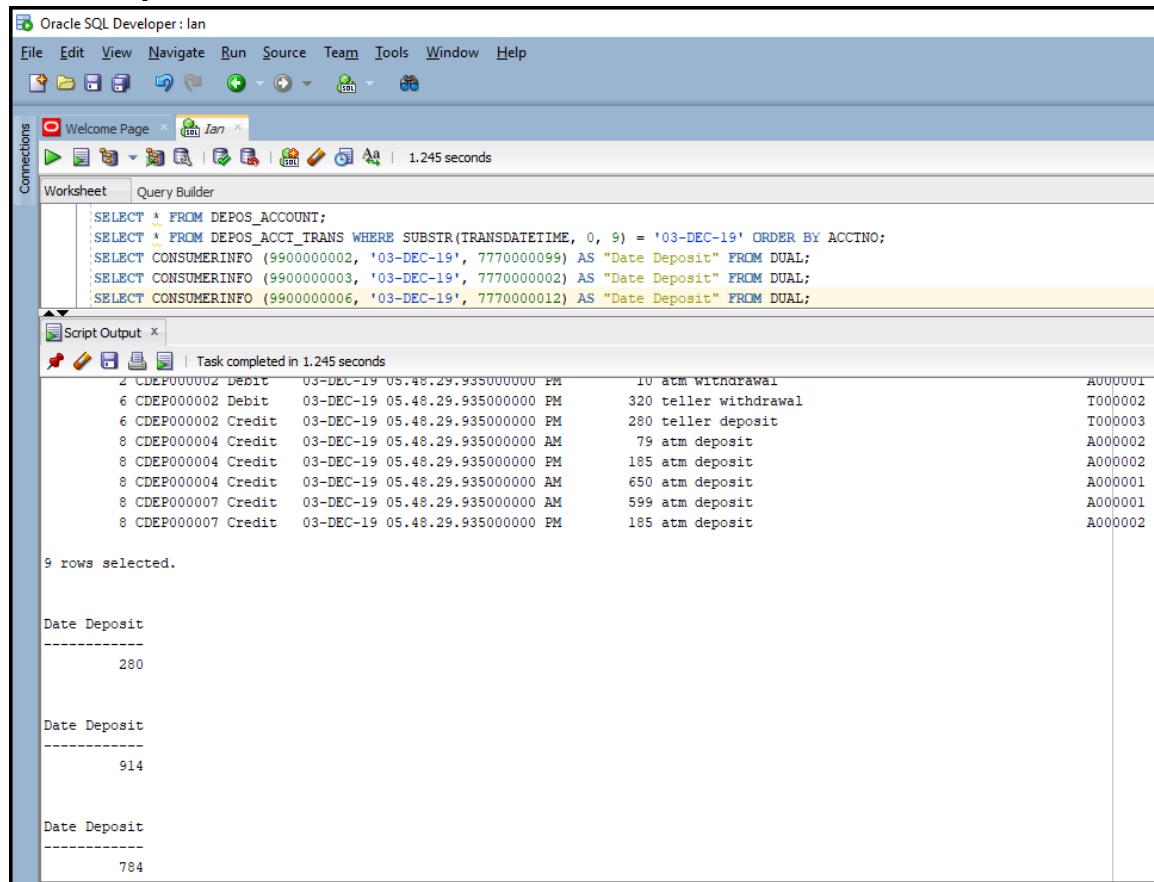
ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

7 rows selected.

The results of the second query are:

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
2	CDEP000001	Credit	03-DEC-19 05.48.29.93500000 PM	52	teller deposit	T000001
2	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001
6	CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	320	teller withdrawal	T000002
6	CDEP000002	Credit	03-DEC-19 05.48.29.93500000 PM	280	teller deposit	T000003
8	CDEP000004	Credit	03-DEC-19 05.48.29.93500000 AM	79	atm deposit	A000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.93500000 AM	650	atm deposit	A000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 AM	599	atm deposit	A000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002

## Execute procedure



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan

Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCOUNT;
SELECT FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '03-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (990000002, '03-DEC-19', 7770000099) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000003, '03-DEC-19', 7770000002) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000006, '03-DEC-19', 7770000012) AS "Date Deposit" FROM DUAL;
```

Script Output | Task completed in 1.245 seconds

Trans ID	Type	Date	Amount	Description	Account
2 CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000001
6 CDEP000002	Debit	03-DEC-19 05.48.29.93500000 PM	320	teller withdrawal	T000002
6 CDEP000002	Credit	03-DEC-19 05.48.29.93500000 PM	280	teller deposit	T000003
8 CDEP000004	Credit	03-DEC-19 05.48.29.93500000 AM	79	atm deposit	A000002
8 CDEP000004	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002
8 CDEP000004	Credit	03-DEC-19 05.48.29.93500000 AM	650	atm deposit	A000001
8 CDEP000007	Credit	03-DEC-19 05.48.29.93500000 AM	599	atm deposit	A000001
8 CDEP000007	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002

9 rows selected.

Date Deposit

-----

280

Date Deposit

-----

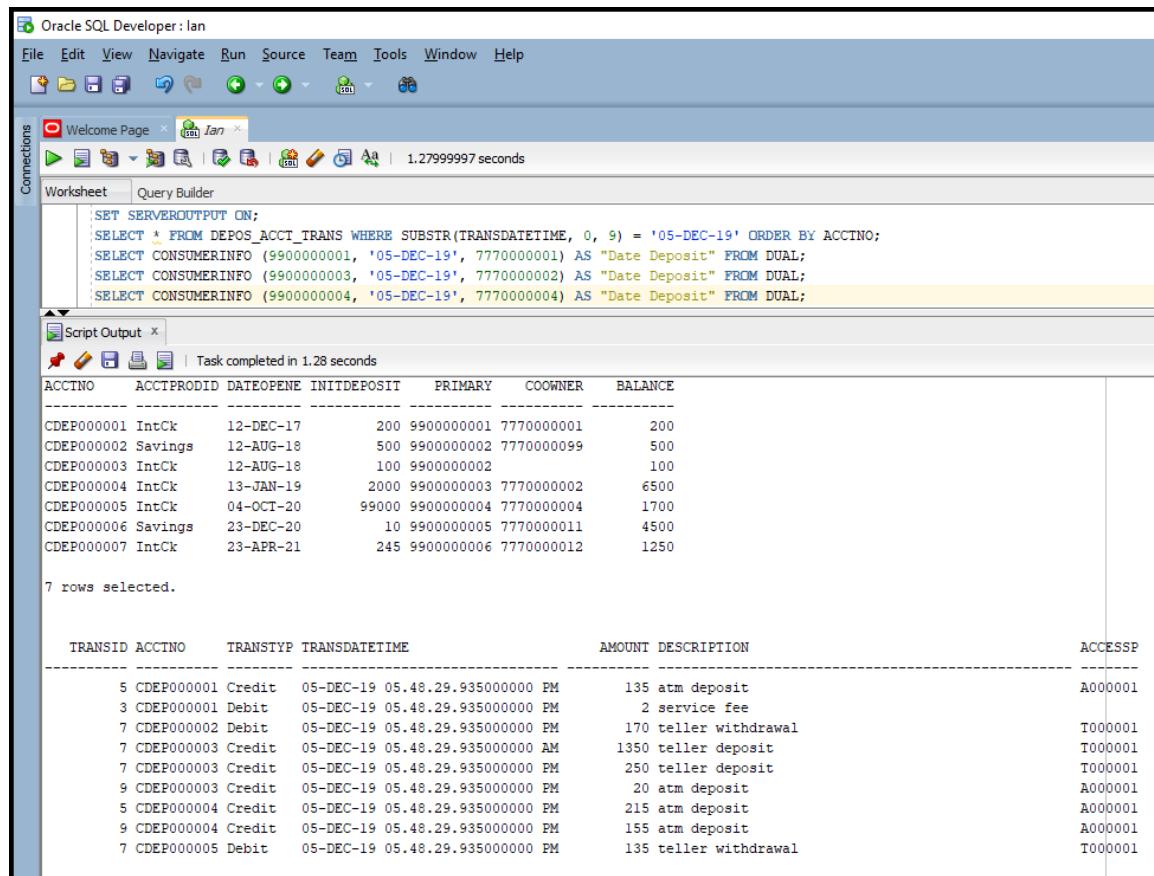
914

Date Deposit

-----

784

## Execute procedure



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script and its execution results.

Script content:

```
SET SERVEROUTPUT ON;
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '05-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (9900000001, '05-DEC-19', 7770000001) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000003, '05-DEC-19', 7770000002) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000004, '05-DEC-19', 7770000004) AS "Date Deposit" FROM DUAL;
```

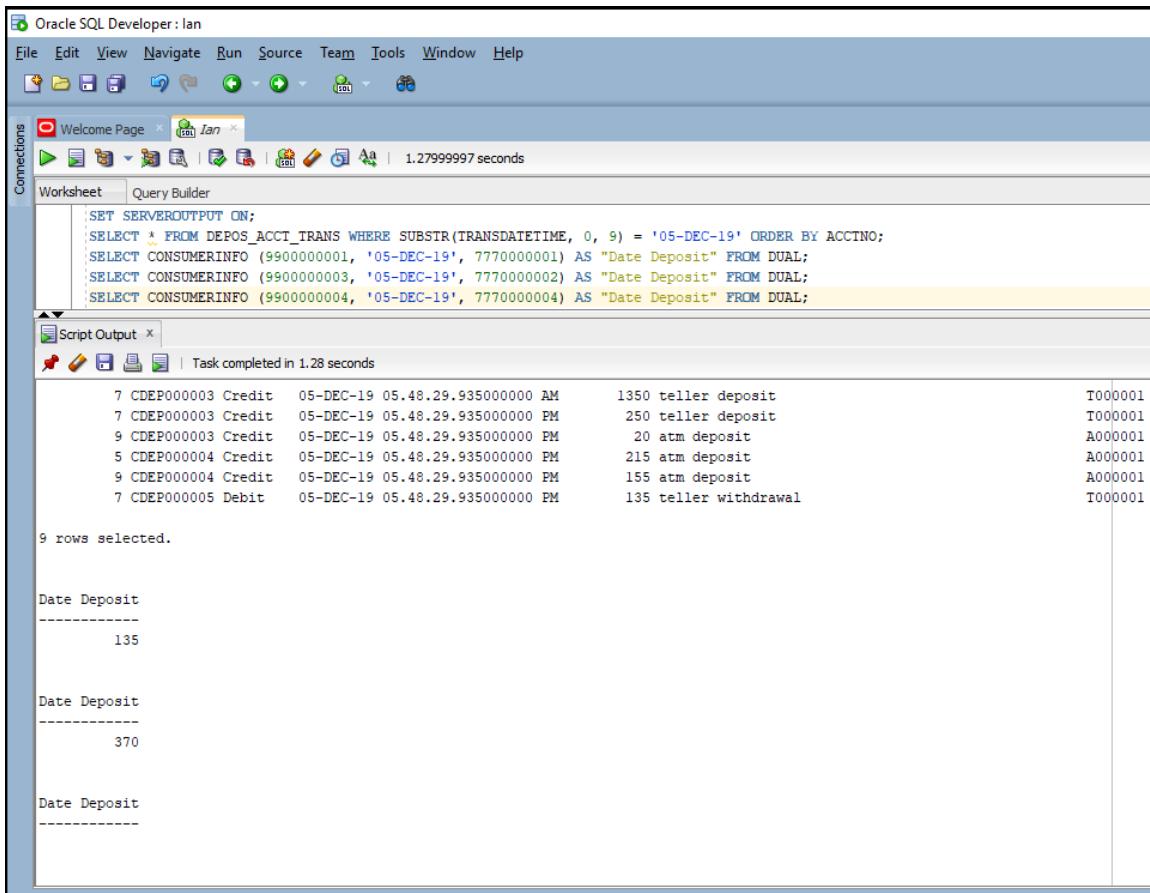
Execution results:

ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

7 rows selected.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
5	CDEP000001	Credit	05-DEC-19 05.48.29.93500000 PM	135	atm deposit	A000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.93500000 PM	2	service fee	
7	CDEP000002	Debit	05-DEC-19 05.48.29.93500000 PM	170	teller withdrawal	T000001
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 AM	1350	teller deposit	T000001
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	250	teller deposit	T000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 PM	20	atm deposit	A000001
5	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	215	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	155	atm deposit	A000001
7	CDEP000005	Debit	05-DEC-19 05.48.29.93500000 PM	135	teller withdrawal	T000001

## Execute procedure



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SET SERVEROUTPUT ON;
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '05-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (9900000001, '05-DEC-19', 7770000001) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000003, '05-DEC-19', 7770000002) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000004, '05-DEC-19', 7770000004) AS "Date Deposit" FROM DUAL;
```

The 'Script Output' tab shows the results of the execution:

Transaction ID	Type	Date	Time	Amount	Description	Account Number	
7 CDEP000003	Credit	05-DEC-19	05.48.29	9.35000000	AM	1350 teller deposit	T000001
7 CDEP000003	Credit	05-DEC-19	05.48.29	9.35000000	PM	250 teller deposit	T000001
9 CDEP000003	Credit	05-DEC-19	05.48.29	9.35000000	PM	20 atm deposit	A000001
5 CDEP000004	Credit	05-DEC-19	05.48.29	9.35000000	PM	215 atm deposit	A000001
9 CDEP000004	Credit	05-DEC-19	05.48.29	9.35000000	PM	155 atm deposit	A000001
7 CDEP000005	Debit	05-DEC-19	05.48.29	9.35000000	PM	135 teller withdrawal	T000001

9 rows selected.

Date Deposit

-----

135

Date Deposit

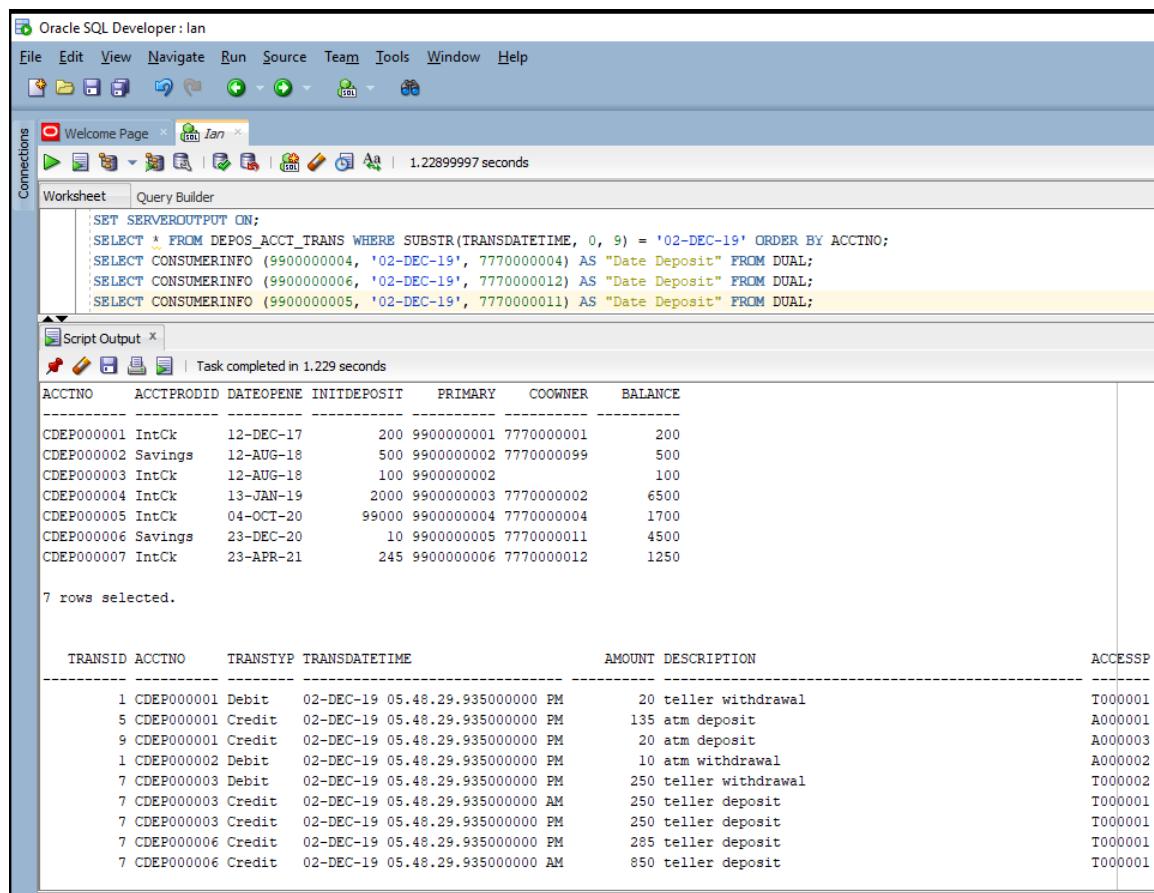
-----

370

Date Deposit

-----

## Execute procedure



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 1.2289997 seconds

Worksheet Query Builder

```
SET SERVEROUTPUT ON;
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '02-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (990000004, '02-DEC-19', 7770000004) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000006, '02-DEC-19', 7770000012) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (990000005, '02-DEC-19', 7770000011) AS "Date Deposit" FROM DUAL;
```

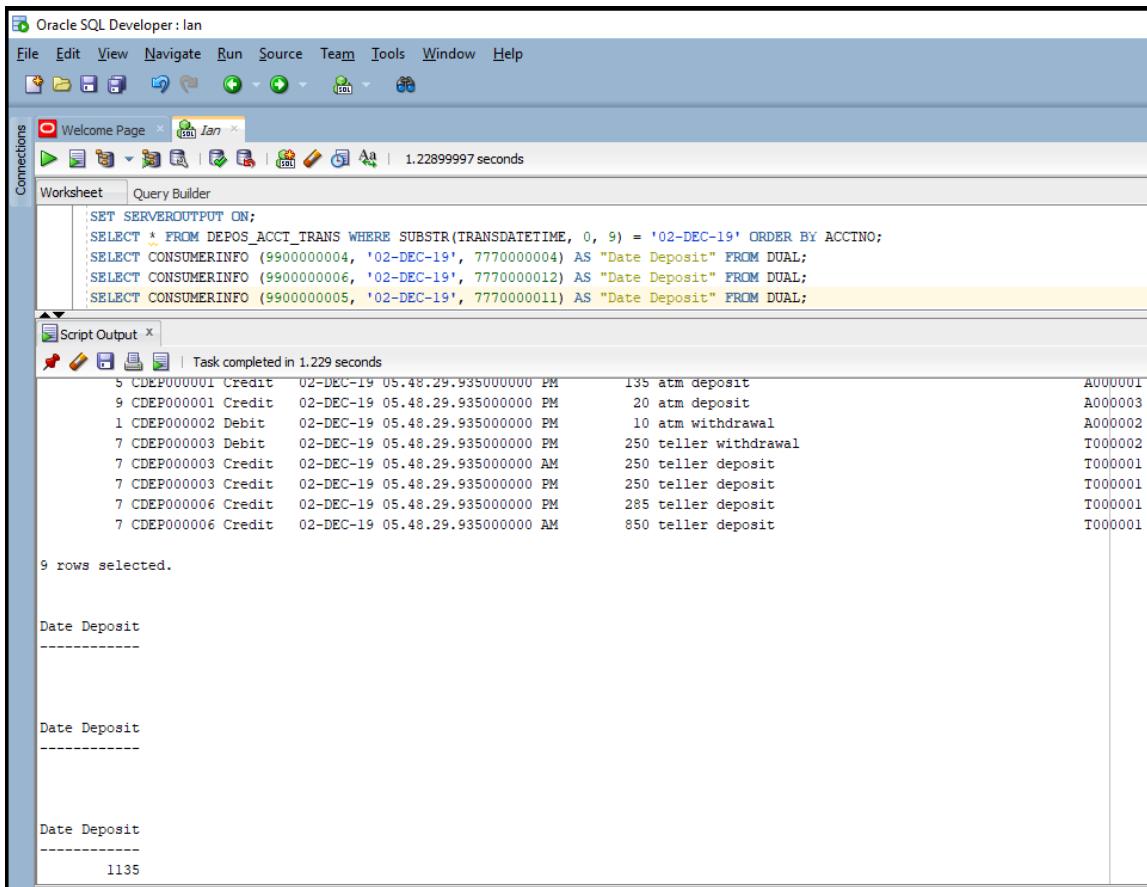
Script Output Task completed in 1.229 seconds

ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

7 rows selected.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
5	CDEP000001	Credit	02-DEC-19 05.48.29.93500000 PM	135	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.93500000 PM	20	atm deposit	A000003
1	CDEP000002	Debit	02-DEC-19 05.48.29.93500000 PM	10	atm withdrawal	A000002
7	CDEP000003	Debit	02-DEC-19 05.48.29.93500000 PM	250	teller withdrawal	T000002
7	CDEP000003	Credit	02-DEC-19 05.48.29.93500000 AM	250	teller deposit	T000001
7	CDEP000003	Credit	02-DEC-19 05.48.29.93500000 PM	250	teller deposit	T000001
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 PM	285	teller deposit	T000001
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 AM	850	teller deposit	T000001

## Execute procedure



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SET SERVEROUTPUT ON;
SELECT * FROM DEPOS_ACCT_TRANS WHERE SUBSTR(TRANSDATETIME, 0, 9) = '02-DEC-19' ORDER BY ACCTNO;
SELECT CONSUMERINFO (9900000004, '02-DEC-19', 7770000004) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000006, '02-DEC-19', 7770000012) AS "Date Deposit" FROM DUAL;
SELECT CONSUMERINFO (9900000005, '02-DEC-19', 7770000011) AS "Date Deposit" FROM DUAL;
```

The 'Script Output' tab shows the results of the execution:

ACCTNO	TRANSDATE	TRANSTIME	AMOUNT	TYPE	DEPOSIT/WITHDRAWAL	
5 CDEP000001	Credit	02-DEC-19	05.48.29.935000000 PM	135	atm deposit	A000001
9 CDEP000001	Credit	02-DEC-19	05.48.29.935000000 PM	20	atm deposit	A000003
1 CDEP000002	Debit	02-DEC-19	05.48.29.935000000 PM	10	atm withdrawal	A000002
7 CDEP000003	Debit	02-DEC-19	05.48.29.935000000 PM	250	teller withdrawal	T000002
7 CDEP000003	Credit	02-DEC-19	05.48.29.935000000 AM	250	teller deposit	T000001
7 CDEP000003	Credit	02-DEC-19	05.48.29.935000000 PM	250	teller deposit	T000001
7 CDEP000006	Credit	02-DEC-19	05.48.29.935000000 PM	285	teller deposit	T000001
7 CDEP000006	Credit	02-DEC-19	05.48.29.935000000 AM	850	teller deposit	T000001

9 rows selected.

Date Deposit

Date Deposit

Date Deposit

1135

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3G Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

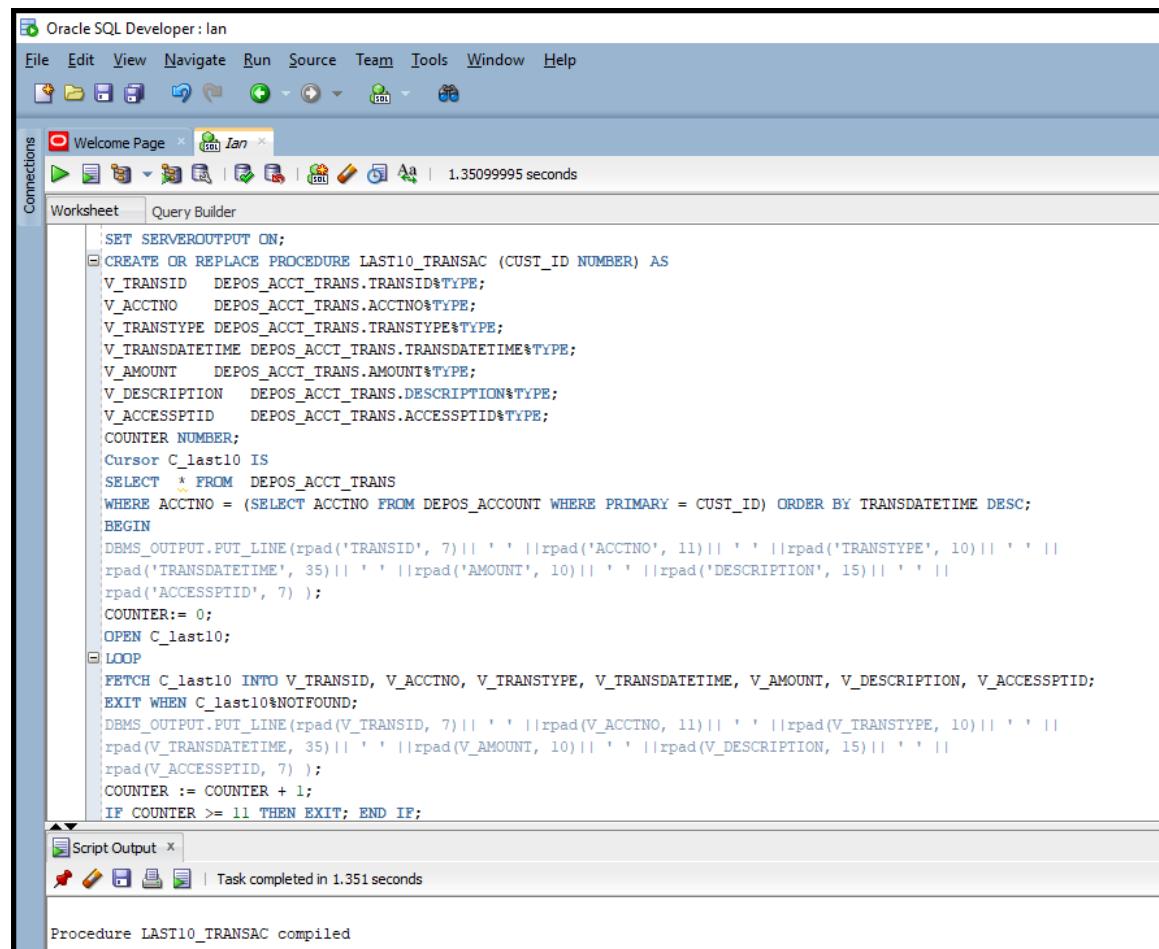
G. Procedure to list the last 10 transactions of a customer by passing Customer Id.

=> NOTE: Assumed Customer Id = Primary

**IN: CUSTOMER ID**

**OUT : LAST 10 TRANSACTIONS**

### Create procedure



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the PL/SQL code for the procedure. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE LAST10_TRANSAC (CUST_ID NUMBER) AS
  V_TRANSID  DEPOS_ACCT_TRANS.TRANSID%TYPE;
  V_ACCTNO  DEPOS_ACCT_TRANS.ACCTNO%TYPE;
  V_TRANSTYPE DEPOS_ACCT_TRANS.TRANSTYPE%TYPE;
  V_TRANSDATETIME DEPOS_ACCT_TRANS.TRANSDATETIME%TYPE;
  V_AMOUNT  DEPOS_ACCT_TRANS.AMOUNT%TYPE;
  V_DESCRIPTION  DEPOS_ACCT_TRANS.DESCRIPTION%TYPE;
  V_ACCESSPTID  DEPOS_ACCT_TRANS.ACCESSPTID%TYPE;
  COUNTER NUMBER;
  Cursor C_last10 IS
  SELECT * FROM DEPOS_ACCT_TRANS
  WHERE ACCTNO = (SELECT ACCTNO FROM DEPOS_ACCOUNT WHERE PRIMARY = CUST_ID) ORDER BY TRANSDATETIME DESC;
  BEGIN
  DBMS_OUTPUT.PUT_LINE(rpad('TRANSID', 7)|| ' ' ||rpad('ACCTNO', 11)|| ' ' ||rpad('TRANSTYPE', 10)|| ' ' ||
  rpad('TRANSDATETIME', 35)|| ' ' ||rpad('AMOUNT', 10)|| ' ' ||rpad('DESCRIPTION', 15)|| ' ' ||
  rpad('ACCESSPTID', 7) );
  COUNTER:= 0;
  OPEN C_last10;
  LOOP
  FETCH C_last10 INTO V_TRANSID, V_ACCTNO, V_TRANSTYPE, V_TRANSDATETIME, V_AMOUNT, V_DESCRIPTION, V_ACCESSPTID;
  EXIT WHEN C_last10%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(rpad(V_TRANSID, 7)|| ' ' ||rpad(V_ACCTNO, 11)|| ' ' ||rpad(V_TRANSTYPE, 10)|| ' ' ||
  rpad(V_TRANSDATETIME, 35)|| ' ' ||rpad(V_AMOUNT, 10)|| ' ' ||rpad(V_DESCRIPTION, 15)|| ' ' ||
  rpad(V_ACCESSPTID, 7) );
  COUNTER := COUNTER + 1;
  IF COUNTER >= 11 THEN EXIT; END IF;
  
```

The 'Script Output' tab at the bottom shows the message: 'Procedure LAST10\_TRANSAC compiled'.

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 1.35099995 seconds

Worksheet Query Builder

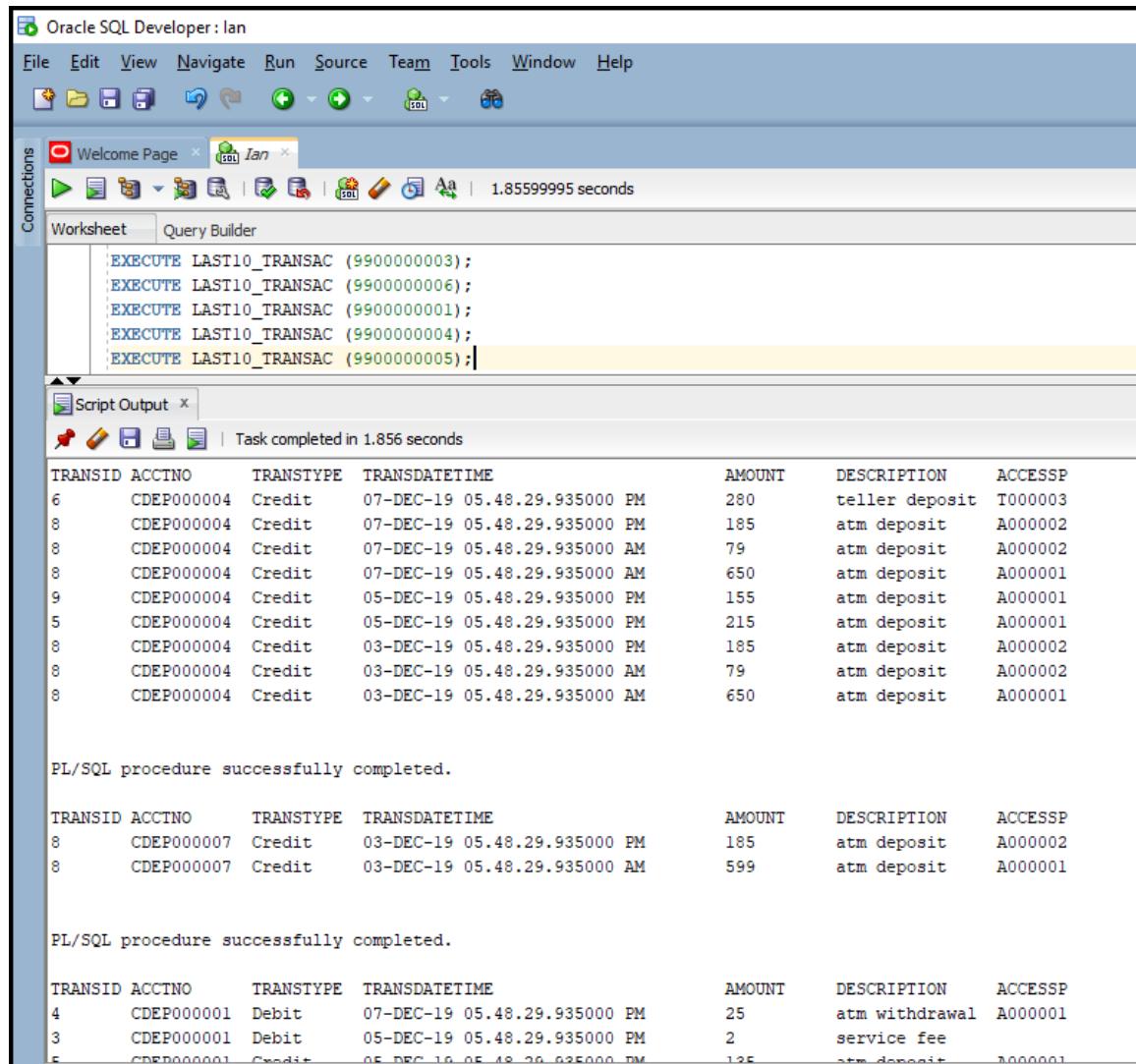
```
V_ACCESSPTID  DEPOS_ACCT_TRANS.ACCESSPTID%TYPE;
COUNTER NUMBER;
Cursor C_last10 IS
SELECT * FROM DEPOS_ACCT_TRANS
WHERE ACCTNO = (SELECT ACCTNO FROM DEPOS_ACCOUNT WHERE PRIMARY = CUST_ID) ORDER BY TRANSDATETIME DESC;
BEGIN
DBMS_OUTPUT.PUT_LINE(rpad('TRANSID', 7)|| ' ' ||rpad('ACCTNO', 11)|| ' ' ||rpad('TRANSTYPE', 10)|| ' ' ||
rpad('TRANSDATETIME', 35)|| ' ' ||rpad('AMOUNT', 10)|| ' ' ||rpad('DESCRIPTION', 15)|| ' ' ||
rpad('ACCESSPTID', 7) );
COUNTER:= 0;
OPEN C_last10;
LOOP
  FETCH C_last10 INTO V_TRANSID, V_ACCTNO, V_TRANSTYPE, V_TRANSDATETIME, V_AMOUNT, V_DESCRIPTION, V_ACCESSPTID;
  EXIT WHEN C_last10%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(rpad(V_TRANSID, 7)|| ' ' ||rpad(V_ACCTNO, 11)|| ' ' ||rpad(V_TRANSTYPE, 10)|| ' ' ||
rpad(V_TRANSDATETIME, 35)|| ' ' ||rpad(V_AMOUNT, 10)|| ' ' ||rpad(V_DESCRIPTION, 15)|| ' ' ||
rpad(V_ACCESSPTID, 7) );
  COUNTER := COUNTER + 1;
  IF COUNTER >= 11 THEN EXIT; END IF;
END LOOP;
CLOSE C_last10;
EXCEPTION
  WHEN OTHERS THEN
    IF SQL%NOTFOUND THEN DBMS_OUTPUT.PUT_LINE('No such record was found'); END IF;
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
END;
/

```

Script Output Task completed in 1.351 seconds

Procedure LAST10\_TRANSAC compiled

## Execute procedure



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. In the 'Worksheet' tab, a script is being run that executes a procedure named 'LAST10\_TRANSAC' five times with different parameters. The output shows the transaction details for each execution. The 'Script Output' tab displays the results of the procedure executions, including the transaction ID, account number, transaction type, date and time, amount, description, and access ID. The results are grouped by account number and transaction type.

```
EXECUTE LAST10_TRANSAC (9900000003);
EXECUTE LAST10_TRANSAC (9900000006);
EXECUTE LAST10_TRANSAC (9900000001);
EXECUTE LAST10_TRANSAC (9900000004);
EXECUTE LAST10_TRANSAC (9900000005);
```

TRANSACTIONID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
6	CDEP000004	Credit	07-DEC-19 05.48.29.935000 PM	280	teller deposit	T000003
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000 PM	185	atm deposit	A000002
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000 AM	79	atm deposit	A000002
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000 AM	650	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.935000 PM	155	atm deposit	A000001
5	CDEP000004	Credit	05-DEC-19 05.48.29.935000 PM	215	atm deposit	A000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000 PM	185	atm deposit	A000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000 AM	79	atm deposit	A000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000 AM	650	atm deposit	A000001

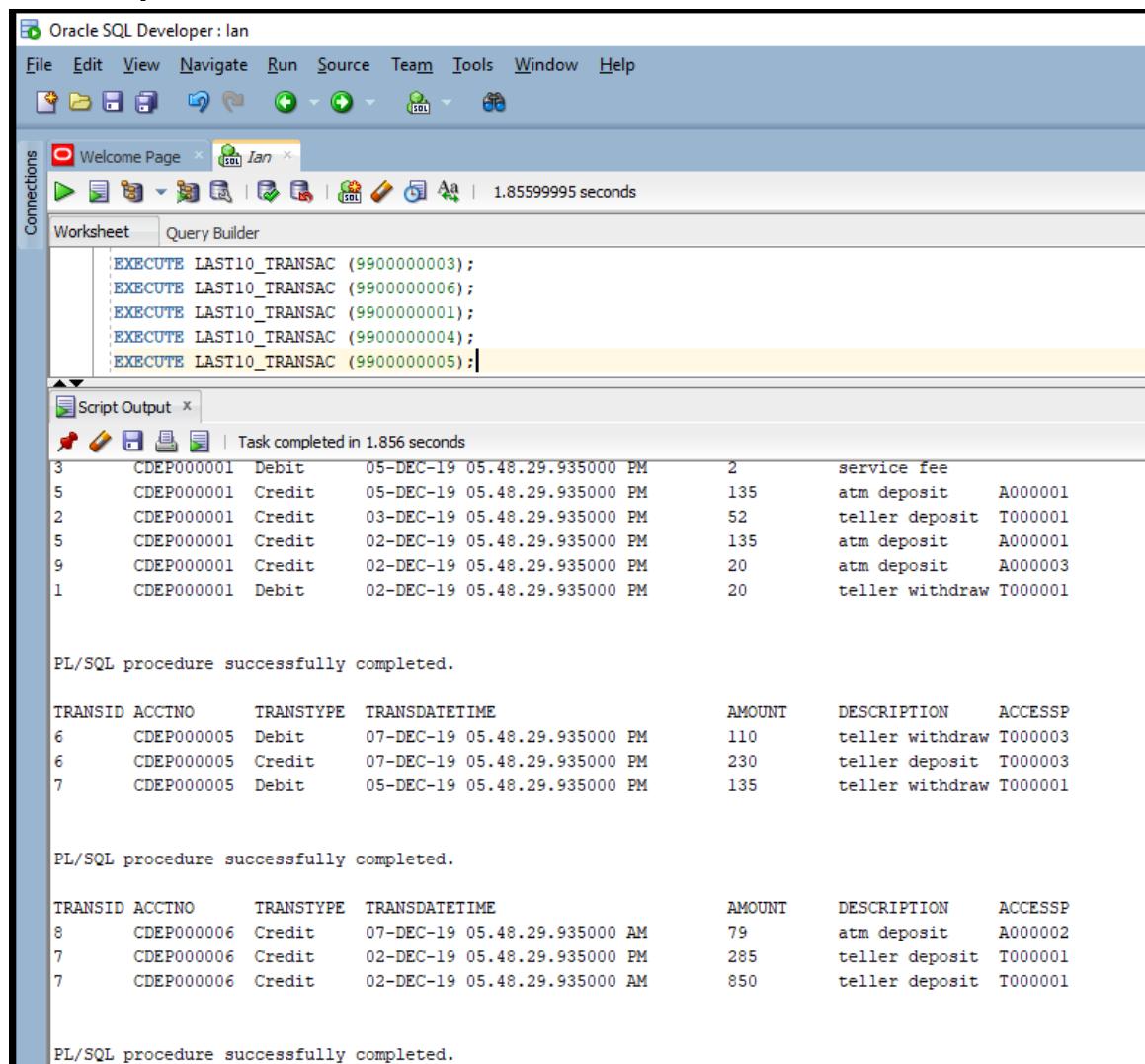
PL/SQL procedure successfully completed.

TRANSACTIONID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000 PM	185	atm deposit	A000002
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000 AM	599	atm deposit	A000001

PL/SQL procedure successfully completed.

TRANSACTIONID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
4	CDEP000001	Debit	07-DEC-19 05.48.29.935000 PM	25	atm withdrawal	A000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000 PM	2	service fee	
5	CDEP000001	Credit	05-DEC-19 05.48.29.935000 PM	125	atm deposit	A000001

## Execute procedure



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying a series of PL/SQL execute statements:

```
EXECUTE LAST10_TRANSAC (9900000003);
EXECUTE LAST10_TRANSAC (9900000006);
EXECUTE LAST10_TRANSAC (9900000001);
EXECUTE LAST10_TRANSAC (9900000004);
EXECUTE LAST10_TRANSAC (9900000005);
```

Below the worksheet, the 'Script Output' tab shows the results of the procedure execution. The first execution (id 3) produces the following log entries:

TRANSACTION_ID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000 PM	2	service fee	
5	CDEP000001	Credit	05-DEC-19 05.48.29.935000 PM	135	atm deposit	A000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000 PM	52	teller deposit	T000001
5	CDEP000001	Credit	02-DEC-19 05.48.29.935000 PM	135	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.935000 PM	20	atm deposit	A000003
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000 PM	20	teller withdraw	T000001

PL/SQL procedure successfully completed.

The second execution (id 6) produces the following log entries:

TRANSACTION_ID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
6	CDEP000005	Debit	07-DEC-19 05.48.29.935000 PM	110	teller withdraw	T000003
6	CDEP000005	Credit	07-DEC-19 05.48.29.935000 PM	230	teller deposit	T000003
7	CDEP000005	Debit	05-DEC-19 05.48.29.935000 PM	135	teller withdraw	T000001

PL/SQL procedure successfully completed.

The third execution (id 8) produces the following log entries:

TRANSACTION_ID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP000006	Credit	07-DEC-19 05.48.29.935000 AM	79	atm deposit	A000002
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000 PM	285	teller deposit	T000001
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000 AM	850	teller deposit	T000001

PL/SQL procedure successfully completed.

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 3H Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

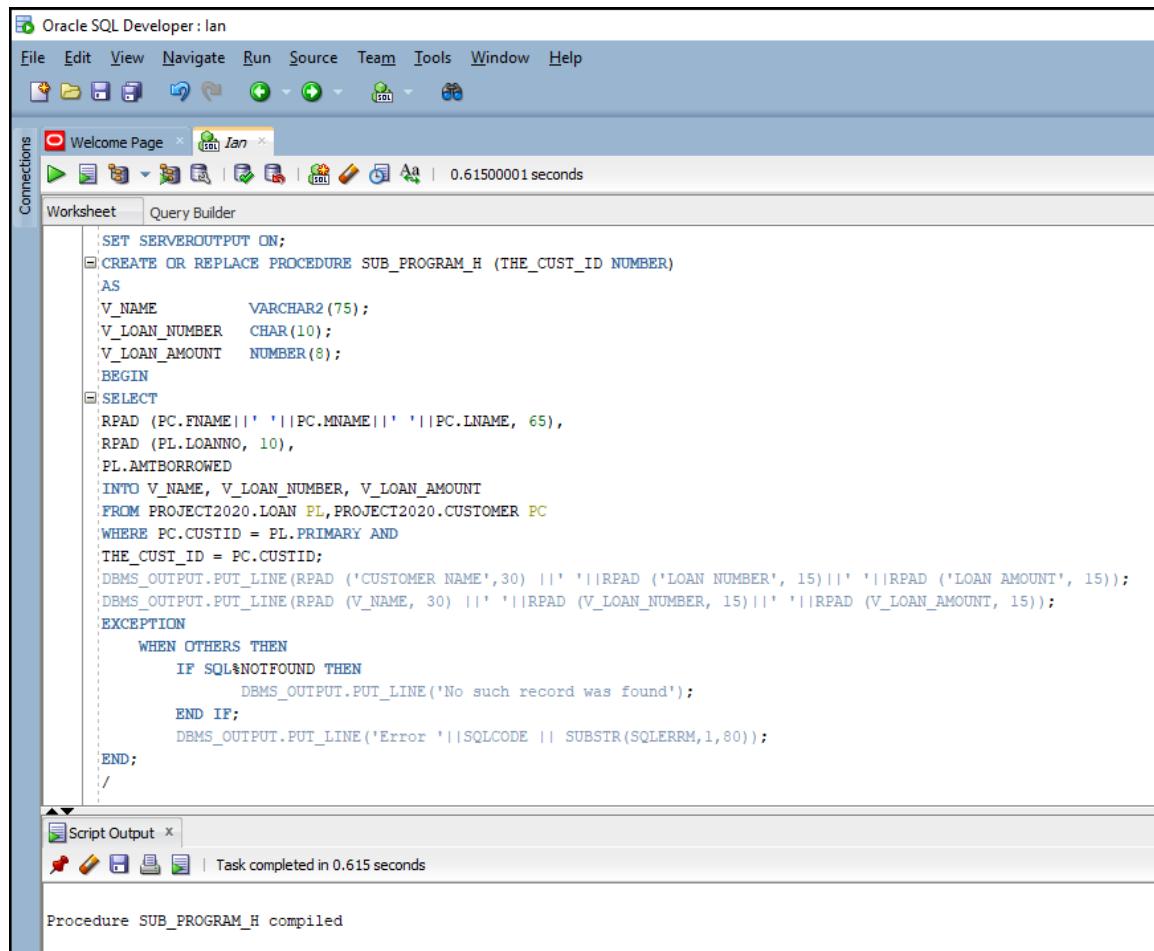
H. Create a subprogram to accept a customer ID and output name (first, mid, and last), loan no for this customer, and amount of loan. (a customer may have more than one loan).

=> NOTE: Assumed Customer Id = Primary

**IN: CUSTOMER ID**

**OUT : CUSTOMER NAME (first, mid, and last), LOAN NUMBER, LOAN AMOUNT**

### Create procedure

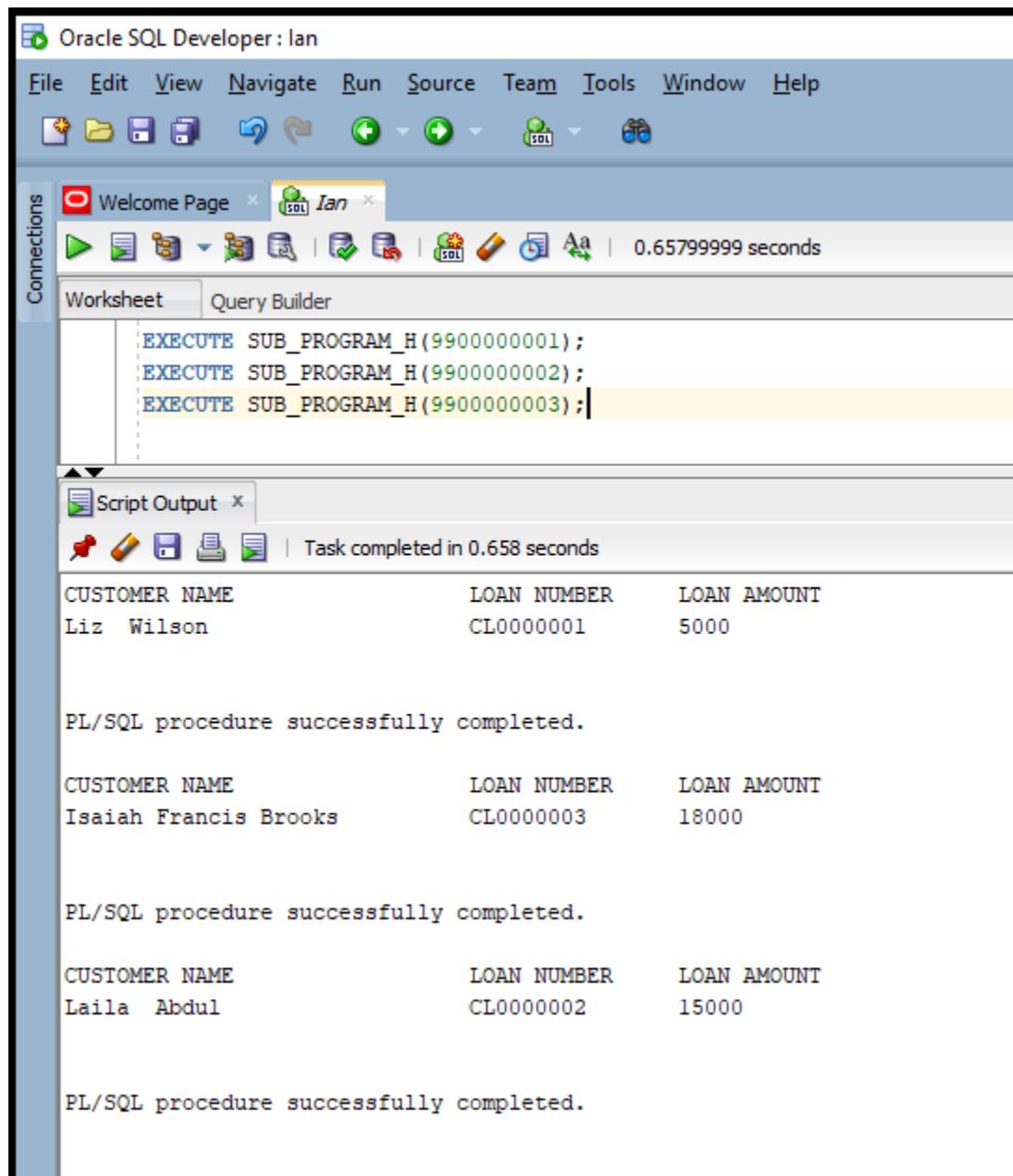


The screenshot shows the Oracle SQL Developer interface with the 'Welcome Page' selected. The 'Worksheet' tab is active, displaying the SQL code for creating a stored procedure. The code defines a procedure named 'SUB\_PROGRAM\_H' that takes a customer ID as input and outputs the customer's name (first, mid, and last), loan number, and loan amount. The procedure uses DBMS\_OUTPUT to print the results. The 'Script Output' tab at the bottom shows the message 'Procedure SUB\_PROGRAM\_H compiled'.

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE SUB_PROGRAM_H (THE_CUST_ID NUMBER)
AS
  V_NAME      VARCHAR2(75);
  V_LOAN_NUMBER CHAR(10);
  V_LOAN_AMOUNT NUMBER(8);
BEGIN
  SELECT
    RPAD (PC.FNAME||' '||PC.MNAME||' '||PC.LNAME, 65),
    RPAD (PL.LOANNO, 10),
    PL.AMTBORROWED
  INTO V_NAME, V_LOAN_NUMBER, V_LOAN_AMOUNT
  FROM PROJECT2020.LOAN PL,PROJECT2020.CUSTOMER PC
  WHERE PC.CUSTID = PL.PRIMARY AND
  THE_CUST_ID = PC.CUSTID;
  DBMS_OUTPUT.PUT_LINE(RPAD ('CUSTOMER NAME',30) ||' '||RPAD ('LOAN NUMBER', 15)||' '||RPAD ('LOAN AMOUNT', 15));
  DBMS_OUTPUT.PUT_LINE(RPAD (V_NAME, 30) ||' '||RPAD (V_LOAN_NUMBER, 15)||' '||RPAD (V_LOAN_AMOUNT, 15));
EXCEPTION
  WHEN OTHERS THEN
    IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE('No such record was found');
    END IF;
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
  END;
/
```

Procedure SUB\_PROGRAM\_H compiled

## Execute procedure.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
EXECUTE SUB_PROGRAM_H(9900000001);
EXECUTE SUB_PROGRAM_H(9900000002);
EXECUTE SUB_PROGRAM_H(9900000003);
```

The code is executed, and the 'Script Output' tab shows the results:

```
PL/SQL procedure successfully completed.

CUSTOMER NAME          LOAN NUMBER      LOAN AMOUNT
Liz Wilson             CL0000001       5000

PL/SQL procedure successfully completed.

CUSTOMER NAME          LOAN NUMBER      LOAN AMOUNT
Isaiah Francis Brooks CL0000003       18000

PL/SQL procedure successfully completed.

CUSTOMER NAME          LOAN NUMBER      LOAN AMOUNT
Laila Abdul            CL0000002       15000
```

The output indicates that three sub-programs were successfully executed, each returning a customer record with their name, loan number, and loan amount.

^^^^^^^^^^^^^^^^^^^^^^^^

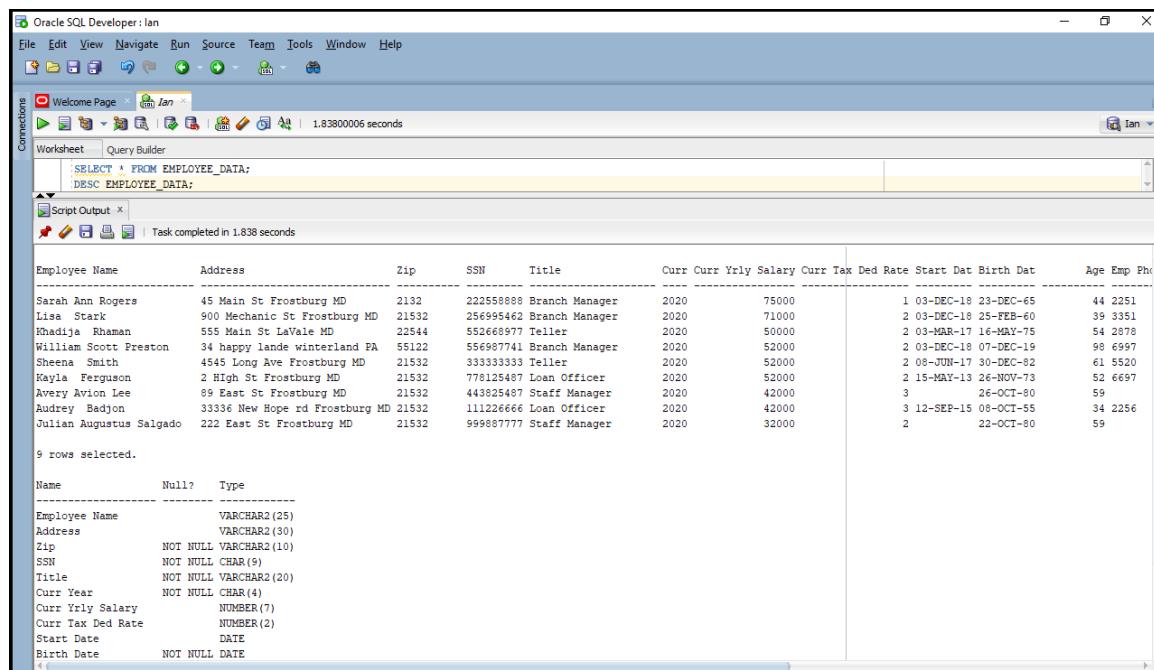
## CHAPTER 3I Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

**I. Create a subprogram to write the following information for each employee into a database called Emp\_list only for employees that older than 30 years old. Name (first, mid, and last), address, date of birth, and their salary.**

**Show contents of employee\_data table.**

**Show attribute variable data types of employee\_data table.**



```
SELECT * FROM EMPLOYEE_DATA;
DESC EMPLOYEE_DATA;
```

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Pho
Sarah Ann Rogers	45 Main St Frostburg MD	2132	2225588888	Branch Manager	2020 75000	1 03-DEC-18 23-DEC-65	44 2251	
Lisa Stark	900 Mechanic St Frostburg MD	21532	256995462	Branch Manager	2020 71000	2 03-DEC-18 25-FEB-60	39 3351	
Khadija Rhaman	555 Main St LaVale MD	22544	552665977	Teller	2020 50000	2 03-MAR-17 16-MAY-75	54 2878	
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020 52000	2 03-DEC-18 07-DEC-19	98 6997	
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020 52000	2 08-JUN-17 30-DEC-82	61 5520	
Kayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020 52000	2 15-MAY-13 26-NOV-73	52 6697	
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020 42000	3 26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020 42000	3 12-SEP-15 08-OCT-55	34 2256	
Julian Augustus Saligado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020 32000	2 22-OCT-80	59	

9 rows selected.

Name	Null?	Type
Employee Name		VARCHAR2 (25)
Address		VARCHAR2 (30)
Zip		NOT NULL VARCHAR2 (10)
SSN		NOT NULL CHAR (9)
Title		NOT NULL VARCHAR2 (20)
Curr Year		NOT NULL CHAR (4)
Curr Yrly Salary		NUMBER (7)
Curr Tax Ded Rate		NUMBER (2)
Start Date		DATE
Birth Date		NOT NULL DATE

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 1.8380006 seconds

Worksheet Query Builder

```
SELECT * FROM EMPLOYEE_DATA;
DESC EMPLOYEE_DATA;
```

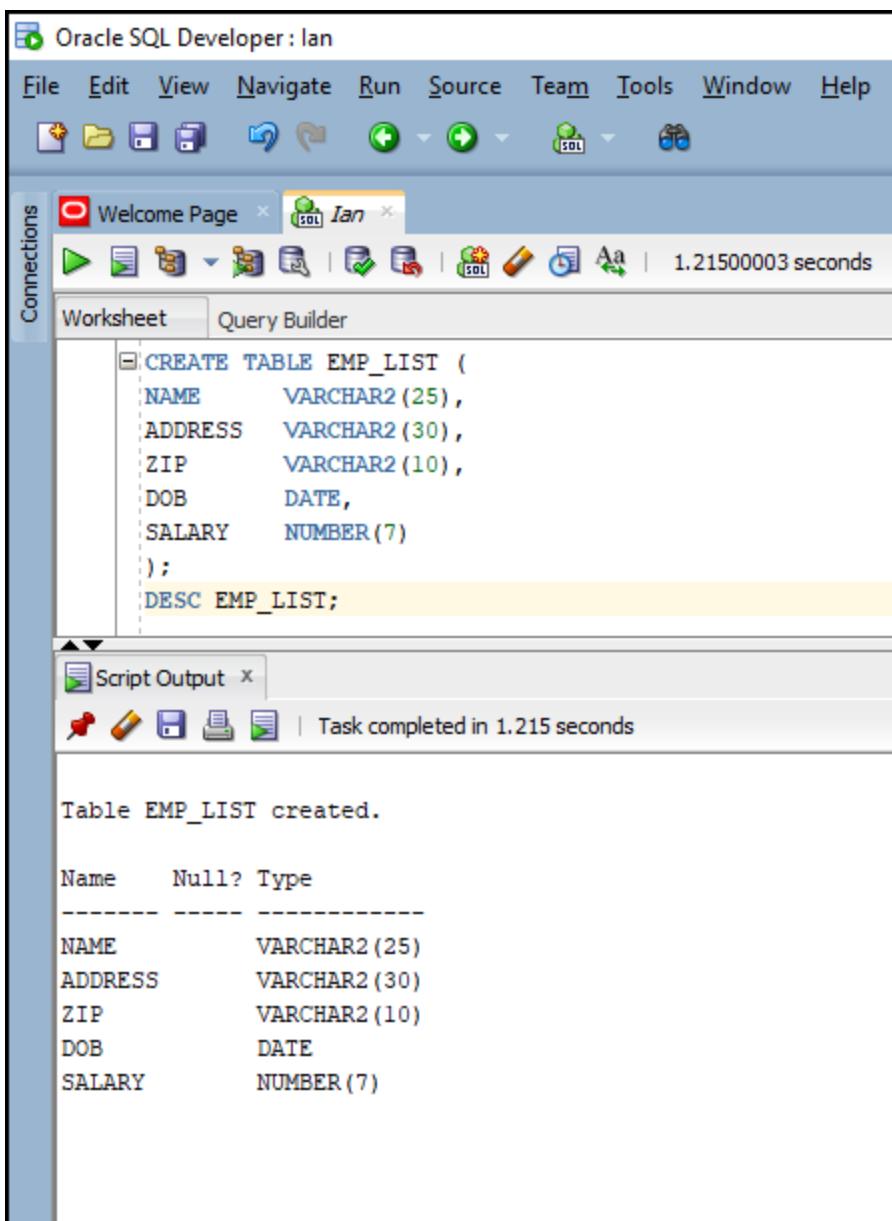
Script Output | Task completed in 1.838 seconds

```
Sheena Smith      4545 Long Ave Frostburg MD 21532 333333333 Teller      2020      52000      2 08-JUN-17 30-DEC-82      61 5520
Kayla Ferguson    2 High St Frostburg MD 21532 778125487 Loan Officer 2020      52000      2 15-MAY-13 26-NOV-73      52 6697
Avery Avion Lee   69 East St Frostburg MD 21532 443825487 Staff Manager 2020      42000      3 26-OCT-80      59
Audrey Badjon    33336 New Hope rd Frostburg MD 21532 111226666 Loan Officer 2020      42000      3 12-SEP-15 08-OCT-55      34 2256
Julian Augustus Salgado 222 East St Frostburg MD 21532 999887777 Staff Manager 2020      32000      2 22-OCT-80      59

9 rows selected.

Name      Null?      Type
-----      -----
Employee Name      VARCHAR2(25)
Address      VARCHAR2(30)
Zip      NOT NULL VARCHAR2(10)
SSN      NOT NULL CHAR(9)
Title      NOT NULL VARCHAR2(20)
Curr Year      NOT NULL CHAR(4)
Curr Yrly Salary      NUMBER(7)
Curr Tax Ded Rate      NUMBER(2)
Start Date      DATE
Birth Date      NOT NULL DATE
Age      NUMBER
Emp Phone Ext      VARCHAR2(15)
Branch Phone Num      VARCHAR2(15)
Branch Name      NOT NULL VARCHAR2(20)
Highest Degree      VARCHAR2(15)
Highest Degree date      VARCHAR2(20)
```

## Create table emp\_list.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

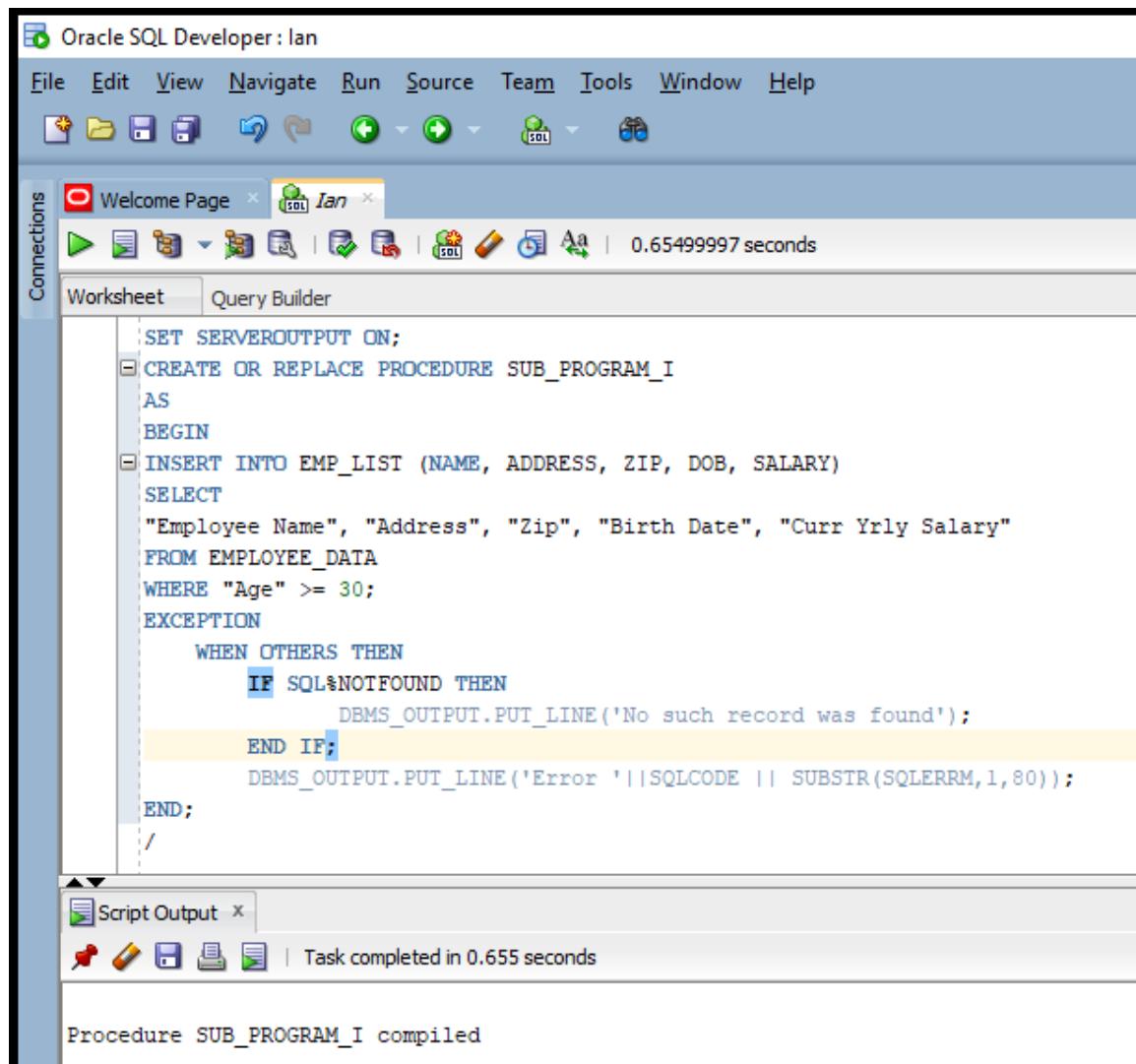
```
CREATE TABLE EMP_LIST (
  NAME      VARCHAR2 (25),
  ADDRESS   VARCHAR2 (30),
  ZIP       VARCHAR2 (10),
  DOB       DATE,
  SALARY    NUMBER (7)
);
DESC EMP_LIST;
```

Below the worksheet, the 'Script Output' tab shows the results of the execution:

```
Table EMP_LIST created.

Name Null? Type
----- -----
NAME      VARCHAR2 (25)
ADDRESS   VARCHAR2 (30)
ZIP       VARCHAR2 (10)
DOB       DATE
SALARY    NUMBER (7)
```

## Create procedure to load emp\_list table.



The screenshot shows the Oracle SQL Developer interface with the following details:

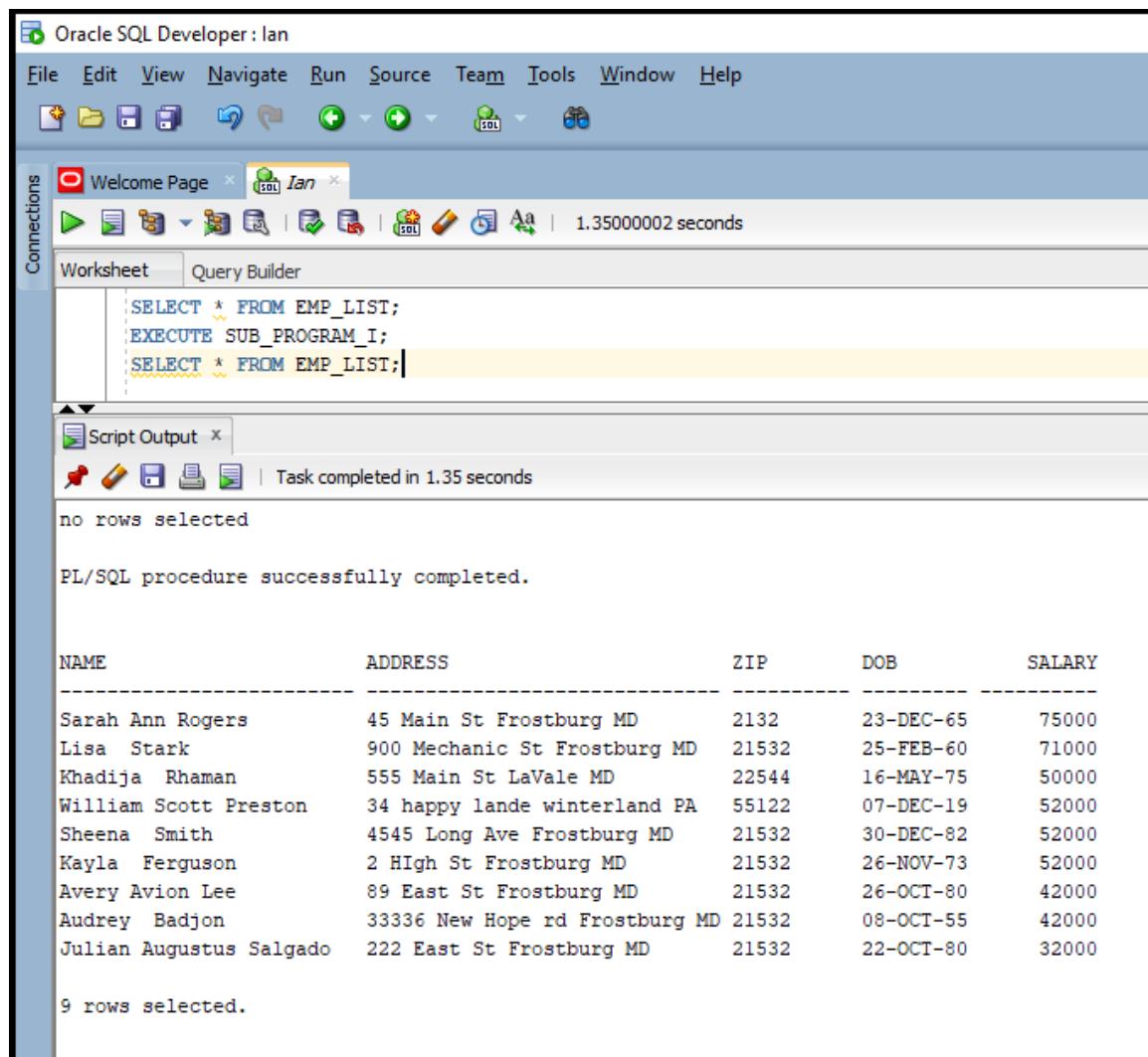
- Connections:** A list of connections, with "Ian" selected.
- Worksheet:** The main area where the PL/SQL code is entered. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE SUB_PROGRAM_I
AS
BEGIN
  INSERT INTO EMP_LIST (NAME, ADDRESS, ZIP, DOB, SALARY)
  SELECT
    "Employee Name", "Address", "Zip", "Birth Date", "Curr Yrly Salary"
  FROM EMPLOYEE_DATA
  WHERE "Age" >= 30;
  EXCEPTION
    WHEN OTHERS THEN
      IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No such record was found');
      END IF;
      DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
  END;
/
```

- Script Output:** A panel at the bottom showing the result of the compilation: "Procedure SUB\_PROGRAM\_I compiled".
- Toolbar:** Standard SQL Developer toolbar with icons for file operations, navigation, and tools.
- Header:** "Oracle SQL Developer : Ian" and a menu bar with File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.

**Execute procedure.**

**Verify contents of emp\_list table.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SELECT * FROM EMP_LIST;
EXECUTE SUB_PROGRAM_I;
SELECT * FROM EMP_LIST;
```

Below the code, the 'Script Output' tab shows the results of the execution:

```
no rows selected

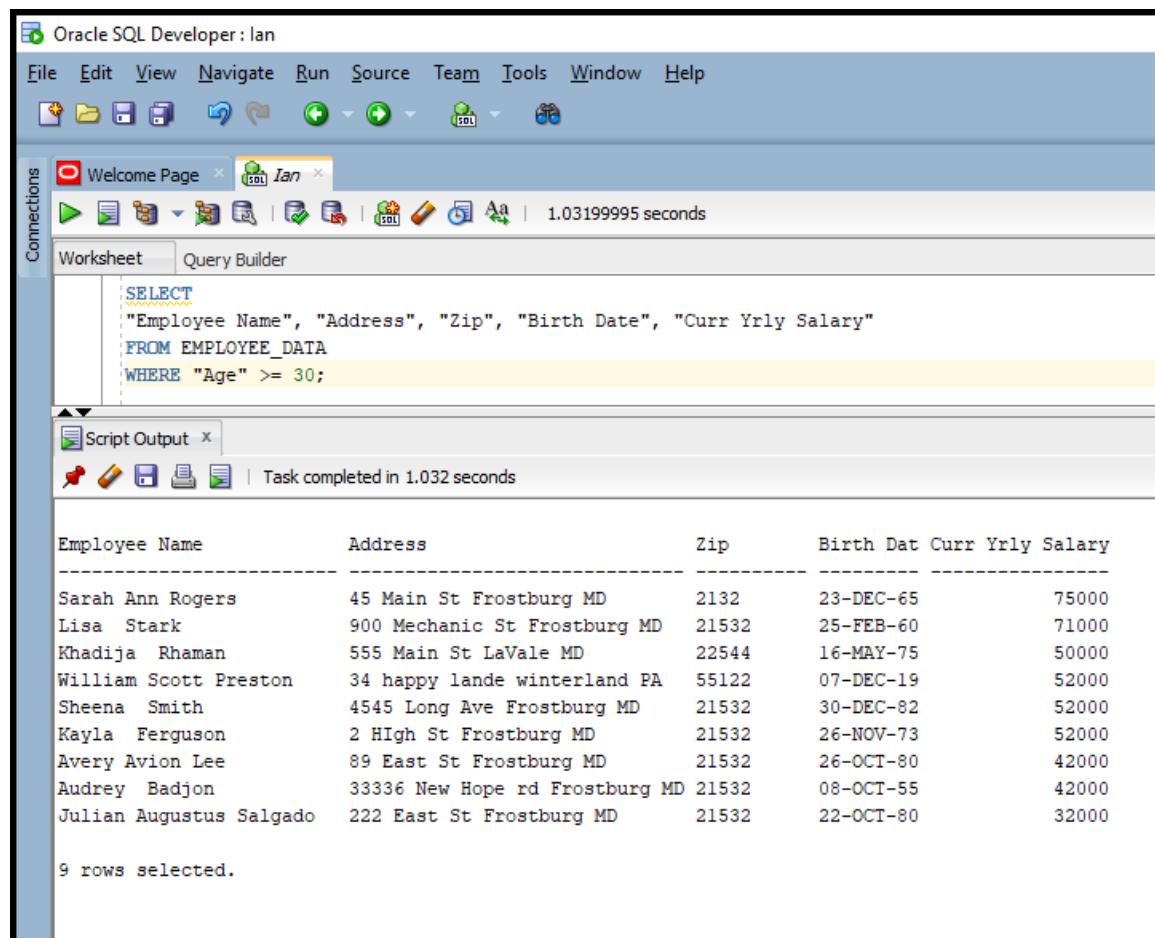
PL/SQL procedure successfully completed.
```

Following this, a table is displayed showing the contents of the EMP\_LIST table:

NAME	ADDRESS	ZIP	DOB	SALARY
Sarah Ann Rogers	45 Main St Frostburg MD	2132	23-DEC-65	75000
Lisa Stark	900 Mechanic St Frostburg MD	21532	25-FEB-60	71000
Khadija Rhaman	555 Main St LaVale MD	22544	16-MAY-75	50000
William Scott Preston	34 happy lande winterland PA	55122	07-DEC-19	52000
Sheena Smith	4545 Long Ave Frostburg MD	21532	30-DEC-82	52000
Kayla Ferguson	2 HIgh St Frostburg MD	21532	26-NOV-73	52000
Avery Avion Lee	89 East St Frostburg MD	21532	26-OCT-80	42000
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	08-OCT-55	42000
Julian Augustus Salgado	222 East St Frostburg MD	21532	22-OCT-80	32000

At the bottom of the table, it says '9 rows selected.'

## Verify contents of employee\_data table for age above 30.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL query:

```
SELECT
  "Employee Name", "Address", "Zip", "Birth Date", "Curr Yrly Salary"
FROM EMPLOYEE_DATA
WHERE "Age" >= 30;
```

The 'Script Output' tab shows the results of the query:

Employee Name	Address	Zip	Birth Dat	Curr Yrly Salary
Sarah Ann Rogers	45 Main St Frostburg MD	2132	23-DEC-65	75000
Lisa Stark	900 Mechanic St Frostburg MD	21532	25-FEB-60	71000
Khadija Rhaman	555 Main St LaVale MD	22544	16-MAY-75	50000
William Scott Preston	34 happy lande winterland PA	55122	07-DEC-19	52000
Sheena Smith	4545 Long Ave Frostburg MD	21532	30-DEC-82	52000
Kayla Ferguson	2 HIgh St Frostburg MD	21532	26-NOV-73	52000
Avery Avion Lee	89 East St Frostburg MD	21532	26-OCT-80	42000
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	08-OCT-55	42000
Julian Augustus Salgado	222 East St Frostburg MD	21532	22-OCT-80	32000

9 rows selected.

^^^^^^^^^^^^^^^^^^^^^^^^

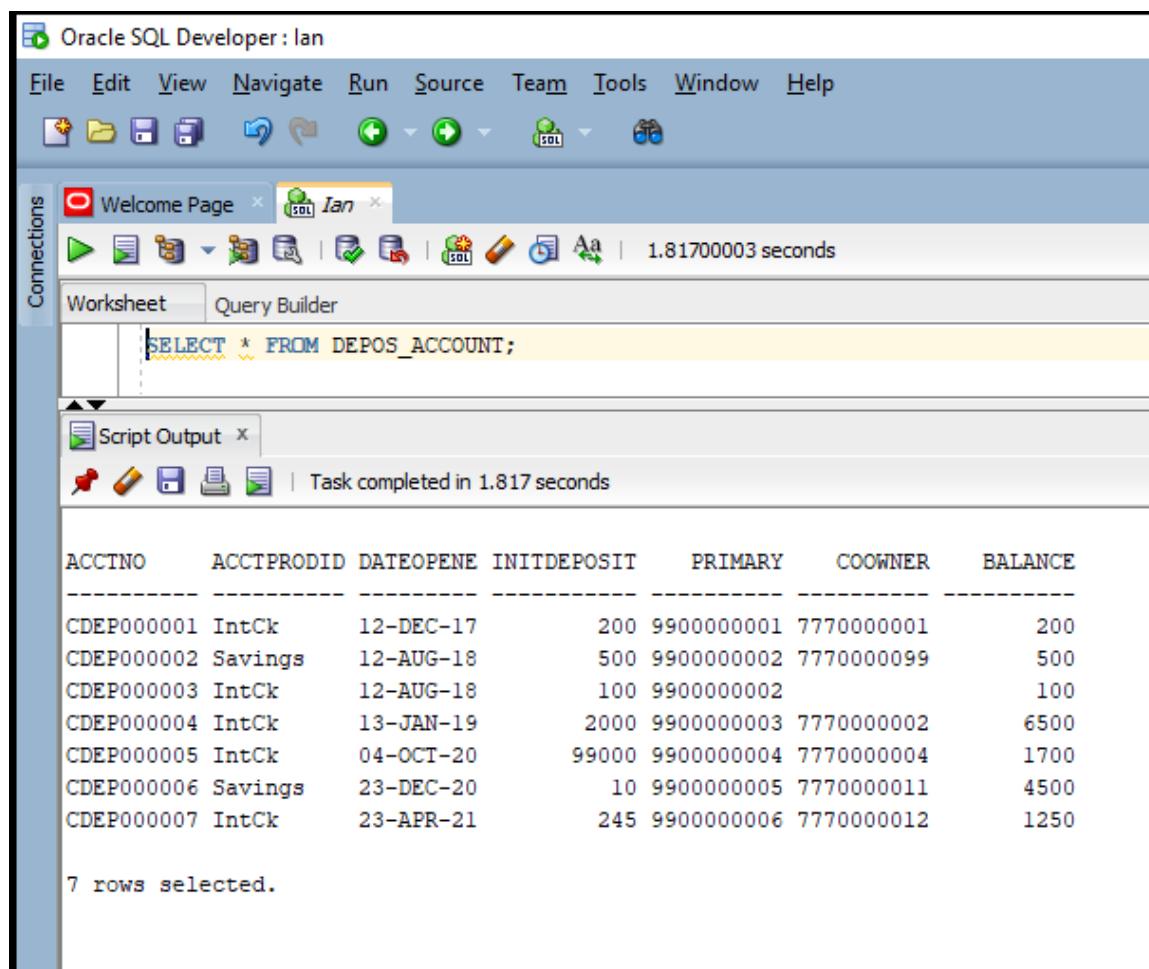
## CHAPTER 4A Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

A. Create a package with the following functions called **BankP** for each customer (customer id):

- a--Function to return the current balance.
- b--Function to return the last deposit from checking.
- c--Function to return the last deposit from saving
- d--Function to return the last withdraw from checking.
- e--Function to return the last withdraw from saving.

**Verify table depos\_account contents.**

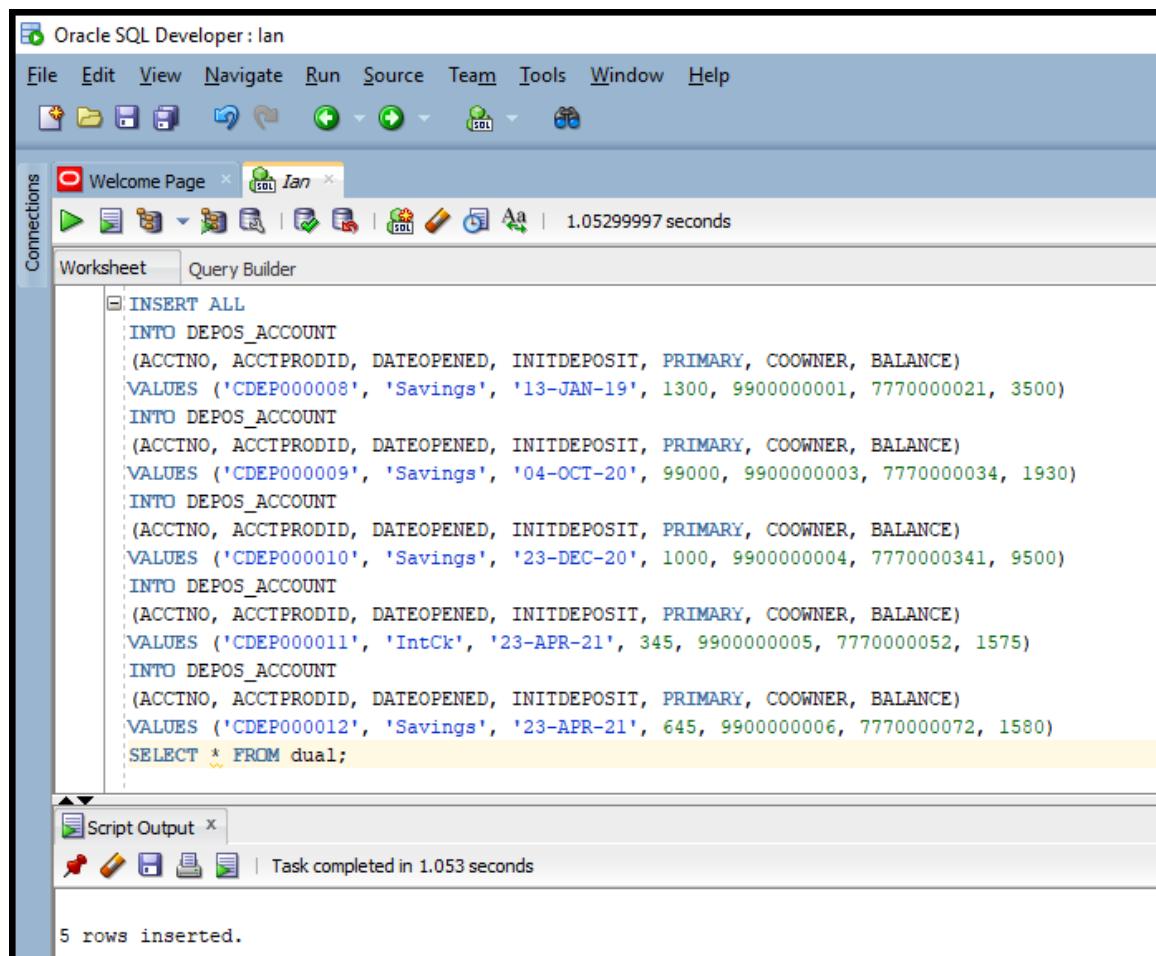


The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar below has various icons for file operations. The Connections sidebar shows a connection named 'Ian'. The main workspace has two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab contains the SQL query: 'SELECT \* FROM DEPOS\_ACCOUNT;'. The 'Script Output' tab shows the results of the query, which is a table with the following data:

ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

7 rows selected.

## Insert rows.

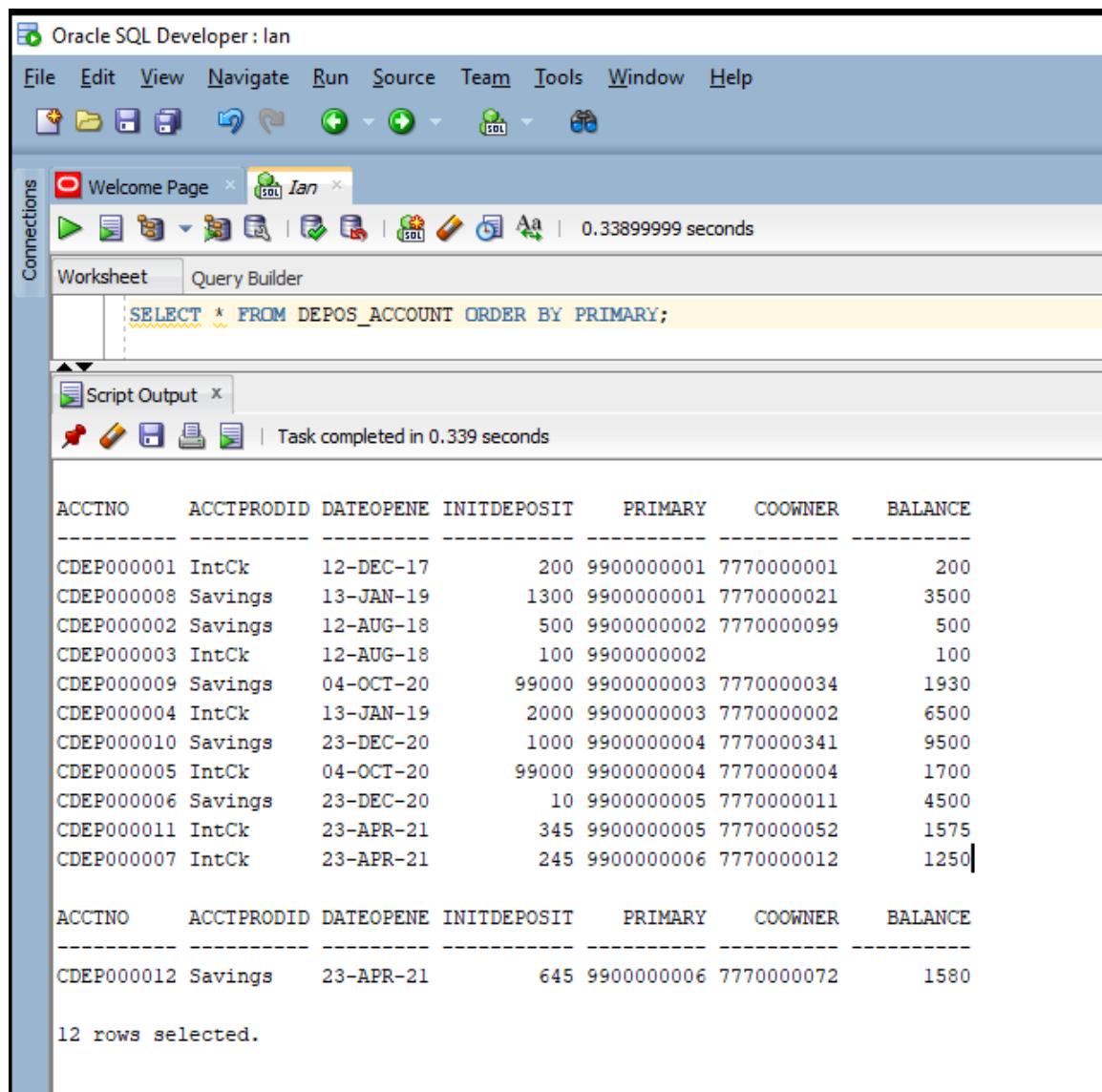


The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying an SQL script for inserting rows into the 'DEPOS\_ACCOUNT' table. The script includes five INSERT statements with specific values for columns like ACCTNO, ACCTPRODID, and INITDEPOSIT. A final SELECT \* FROM dual; statement is present. The 'Script Output' tab at the bottom shows a message indicating 5 rows inserted in 1.053 seconds.

```
INSERT ALL
  INTO DEPOS_ACCOUNT
    (ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
    VALUES ('CDEP000008', 'Savings', '13-JAN-19', 1300, 9900000001, 7770000021, 3500)
  INTO DEPOS_ACCOUNT
    (ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
    VALUES ('CDEP000009', 'Savings', '04-OCT-20', 99000, 9900000003, 7770000034, 1930)
  INTO DEPOS_ACCOUNT
    (ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
    VALUES ('CDEP000010', 'Savings', '23-DEC-20', 1000, 9900000004, 77700000341, 9500)
  INTO DEPOS_ACCOUNT
    (ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
    VALUES ('CDEP000011', 'IntCk', '23-APR-21', 345, 9900000005, 7770000052, 1575)
  INTO DEPOS_ACCOUNT
    (ACCTNO, ACCTPRODID, DATEOPENED, INITDEPOSIT, PRIMARY, COOWNER, BALANCE)
    VALUES ('CDEP000012', 'Savings', '23-APR-21', 645, 9900000006, 7770000072, 1580)
SELECT * FROM dual;

5 rows inserted.
```

## Check table contents.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. A query is being run in the 'Worksheet' tab:

```
SELECT * FROM DEPOS_ACCOUNT ORDER BY PRIMARY;
```

The results are displayed in the 'Script Output' tab:

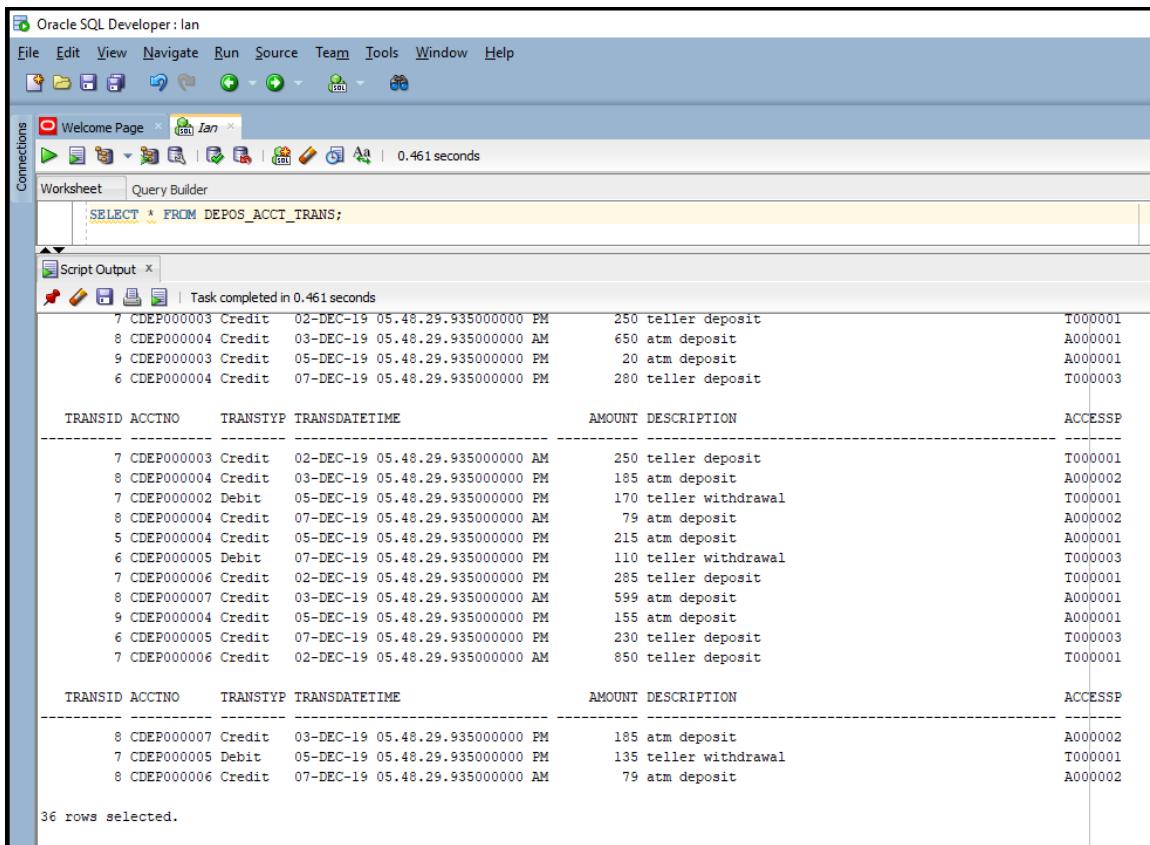
ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000001	IntCk	12-DEC-17	200	9900000001	7770000001	200
CDEP000008	Savings	13-JAN-19	1300	9900000001	7770000021	3500
CDEP000002	Savings	12-AUG-18	500	9900000002	7770000099	500
CDEP000003	IntCk	12-AUG-18	100	9900000002		100
CDEP000009	Savings	04-OCT-20	99000	9900000003	7770000034	1930
CDEP000004	IntCk	13-JAN-19	2000	9900000003	7770000002	6500
CDEP000010	Savings	23-DEC-20	1000	9900000004	7770000341	9500
CDEP000005	IntCk	04-OCT-20	99000	9900000004	7770000004	1700
CDEP000006	Savings	23-DEC-20	10	9900000005	7770000011	4500
CDEP000011	IntCk	23-APR-21	345	9900000005	7770000052	1575
CDEP000007	IntCk	23-APR-21	245	9900000006	7770000012	1250

ACCTNO	ACCTPRODID	DATEOPENE	INITDEPOSIT	PRIMARY	COOWNER	BALANCE
CDEP000012	Savings	23-APR-21	645	9900000006	7770000072	1580

12 rows selected.

## Check table contents.

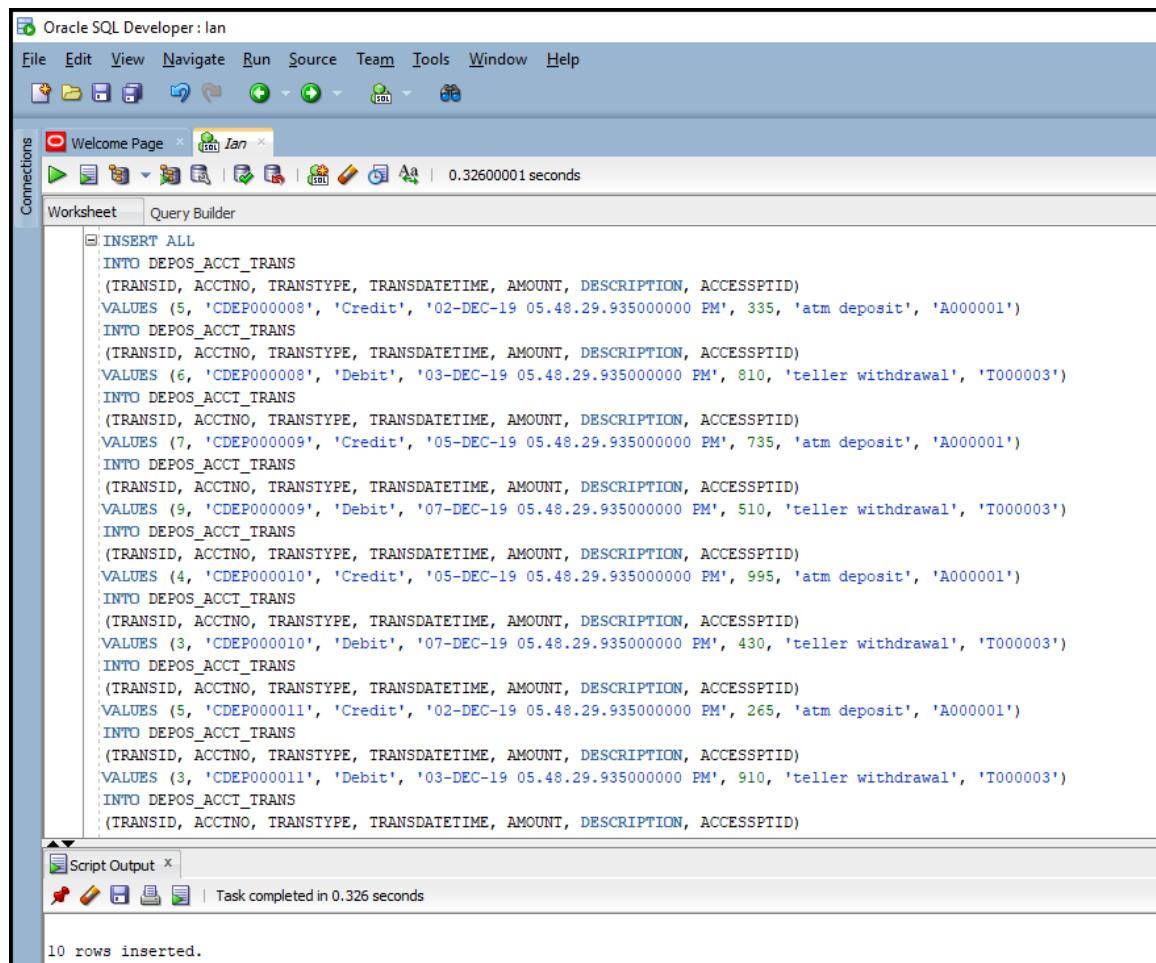


The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL query: `SELECT * FROM DEPOS_ACCT_TRANS;`. The 'Script Output' tab shows the results of the query execution. The results are presented in three separate tables, each with columns: TRANSID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSP. The first table has 7 rows, the second has 8 rows, and the third has 3 rows. The output is as follows:

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
7	CDEP000003	Credit	02-DEC-19 05.48.29.935000000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 AM	650	atm deposit	A000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.935000000 PM	20	atm deposit	A000001
6	CDEP000004	Credit	07-DEC-19 05.48.29.935000000 PM	280	teller deposit	T000003
7	CDEP000003	Credit	02-DEC-19 05.48.29.935000000 AM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002
7	CDEP000002	Debit	05-DEC-19 05.48.29.935000000 PM	170	teller withdrawal	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002
5	CDEP000004	Credit	05-DEC-19 05.48.29.935000000 PM	215	atm deposit	A000001
6	CDEP000005	Debit	07-DEC-19 05.48.29.935000000 PM	110	teller withdrawal	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000000 PM	285	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000000 AM	599	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.935000000 PM	155	atm deposit	A000001
6	CDEP000005	Credit	07-DEC-19 05.48.29.935000000 PM	230	teller deposit	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000000 AM	850	teller deposit	T000001
TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002
7	CDEP000005	Debit	05-DEC-19 05.48.29.935000000 PM	135	teller withdrawal	T000001
8	CDEP000006	Credit	07-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002

36 rows selected.

## Insert rows.



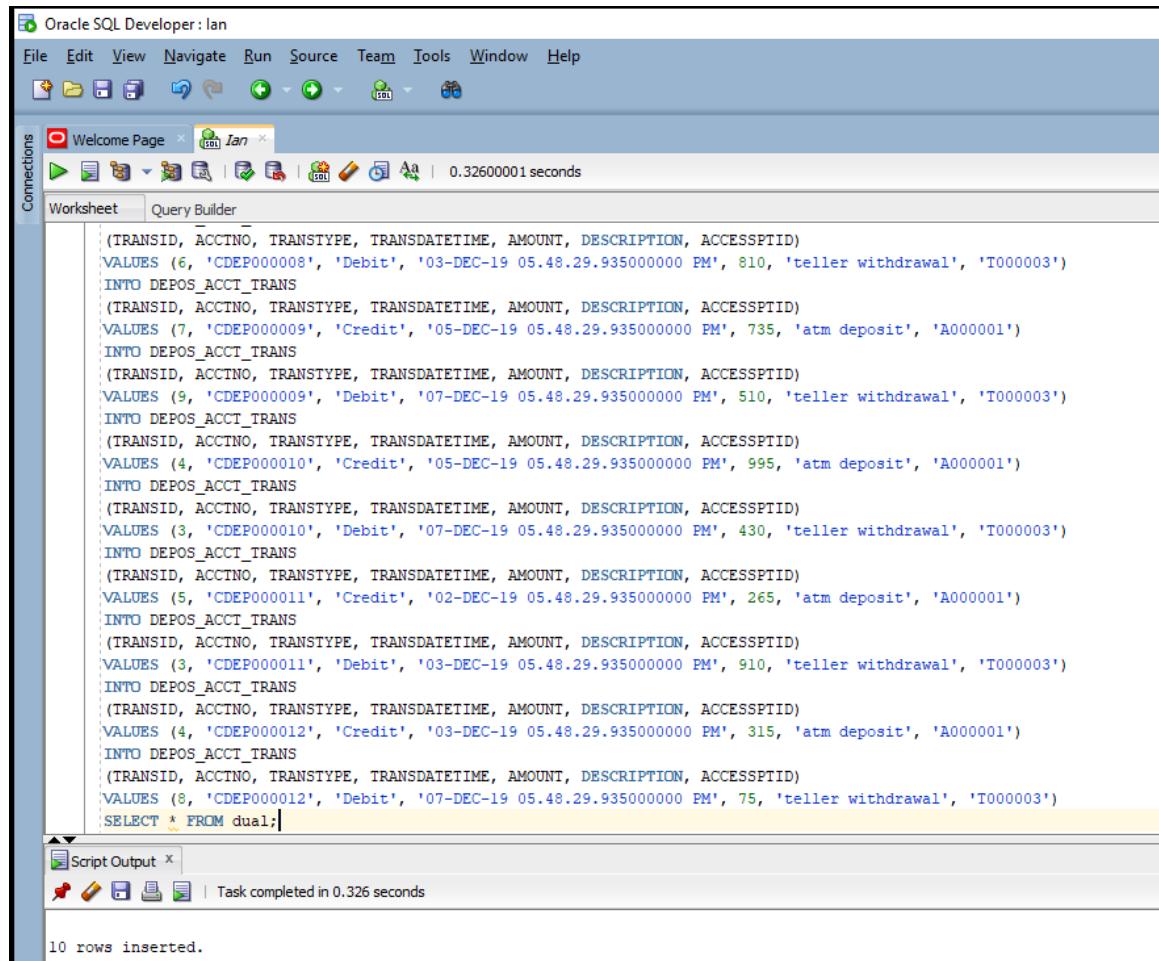
The screenshot shows the Oracle SQL Developer interface. The Worksheet tab contains a SQL script for inserting rows into the DEPOS\_ACCT\_TRANS table. The Script Output tab shows the execution results, indicating 10 rows inserted in 0.326 seconds.

```
INSERT ALL
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (5, 'CDEP000008', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 335, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (6, 'CDEP000008', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 810, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (7, 'CDEP000009', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 735, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (9, 'CDEP000009', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 510, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (4, 'CDEP000010', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 995, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (3, 'CDEP000010', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 430, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (5, 'CDEP000011', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 265, 'atm deposit', 'A000001')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES (3, 'CDEP000011', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 910, 'teller withdrawal', 'T000003')
  INTO DEPOS_ACCT_TRANS
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
```

Script Output tab: Task completed in 0.326 seconds

10 rows inserted.

## Insert rows.



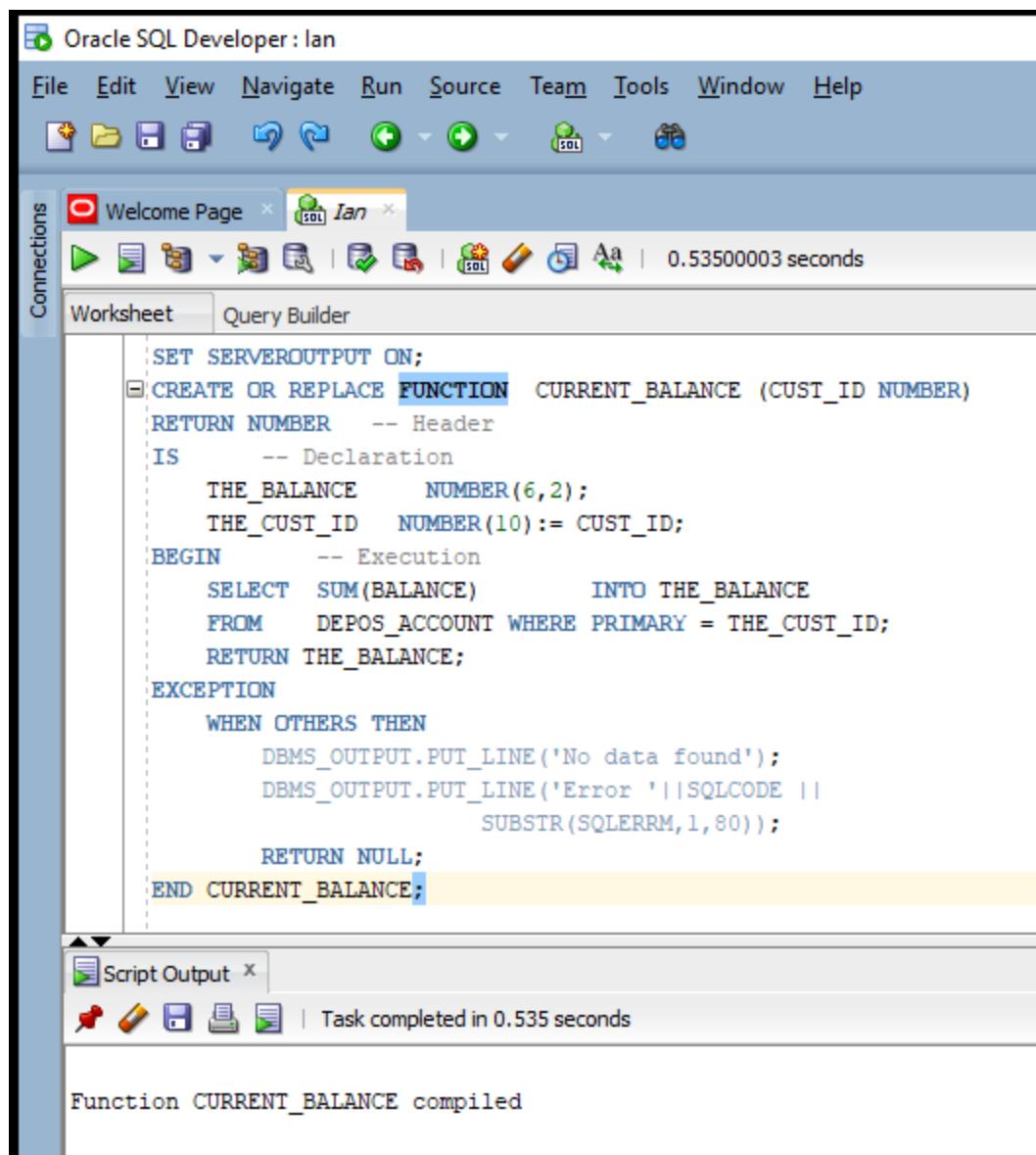
The screenshot shows the Oracle SQL Developer interface with a script execution window. The script inserts 10 rows into the DEPOS\_ACCT\_TRANS table. The rows represent various transactions (Debit and Credit) on different account numbers (ACCTNO) on different dates and times, with amounts ranging from 75 to 995. The script concludes with a SELECT \* FROM dual; statement. The execution completed in 0.326 seconds.

```
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (6, 'CDEP000008', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 810, 'teller withdrawal', 'T000003')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (7, 'CDEP000009', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 735, 'atm deposit', 'A000001')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (9, 'CDEP000009', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 510, 'teller withdrawal', 'T000003')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (4, 'CDEP000010', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 995, 'atm deposit', 'A000001')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (3, 'CDEP000010', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 430, 'teller withdrawal', 'T000003')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (5, 'CDEP000011', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 265, 'atm deposit', 'A000001')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (3, 'CDEP000011', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 910, 'teller withdrawal', 'T000003')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (4, 'CDEP000012', 'Credit', '03-DEC-19 05.48.29.935000000 PM', 315, 'atm deposit', 'A000001')
| INTO DEPOS_ACCT_TRANS
| (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
| VALUES (8, 'CDEP000012', 'Debit', '07-DEC-19 05.48.29.935000000 PM', 75, 'teller withdrawal', 'T000003')
| SELECT * FROM dual;
```

Script Output | Task completed in 0.326 seconds

10 rows inserted.

## Create function.

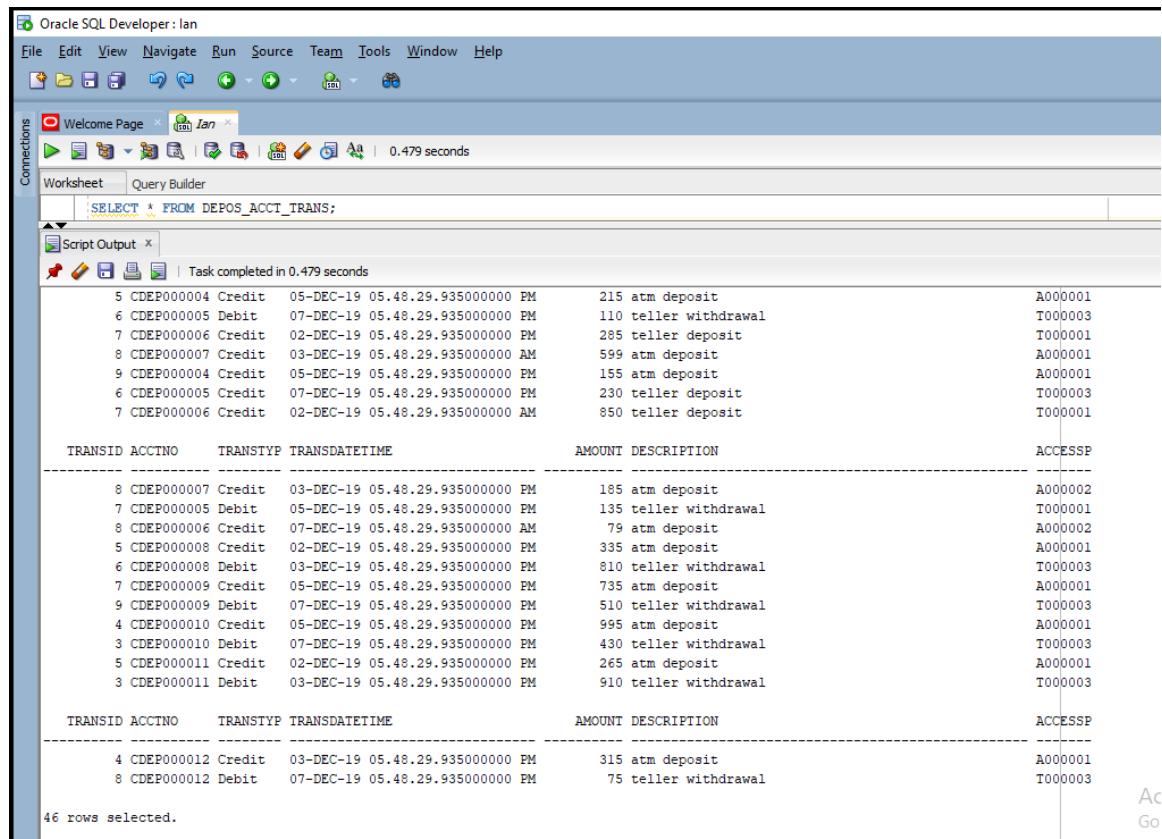


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION CURRENT_BALANCE (CUST_ID NUMBER)
  RETURN NUMBER -- Header
IS -- Declaration
  THE_BALANCE      NUMBER(6,2);
  THE_CUST_ID     NUMBER(10):= CUST_ID;
BEGIN -- Execution
  SELECT SUM(BALANCE)      INTO THE_BALANCE
  FROM DEPOS_ACCOUNT WHERE PRIMARY = THE_CUST_ID;
  RETURN THE_BALANCE;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
    RETURN NULL;
END CURRENT_BALANCE;
```

The code is highlighted with syntax coloring. The 'Script Output' tab at the bottom shows the message: 'Function CURRENT\_BALANCE compiled'.

## Check table contents.



The screenshot shows the Oracle SQL Developer interface with a query window displaying the results of a SELECT statement. The results are presented in three horizontal sections, each with a different header and a dashed line separator.

**Section 1:**

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
5	CDEP00004	Credit	05-DEC-19 05.48.29.935000000 PM	215	atm deposit	A000001
6	CDEP00005	Debit	07-DEC-19 05.48.29.935000000 PM	110	teller withdrawal	T000003
7	CDEP00006	Credit	02-DEC-19 05.48.29.935000000 PM	285	teller deposit	T000001
8	CDEP00007	Credit	03-DEC-19 05.48.29.935000000 AM	599	atm deposit	A000001
9	CDEP00004	Credit	05-DEC-19 05.48.29.935000000 PM	155	atm deposit	A000001
6	CDEP00005	Credit	07-DEC-19 05.48.29.935000000 PM	230	teller deposit	T000003
7	CDEP00006	Credit	02-DEC-19 05.48.29.935000000 AM	850	teller deposit	T000001

**Section 2:**

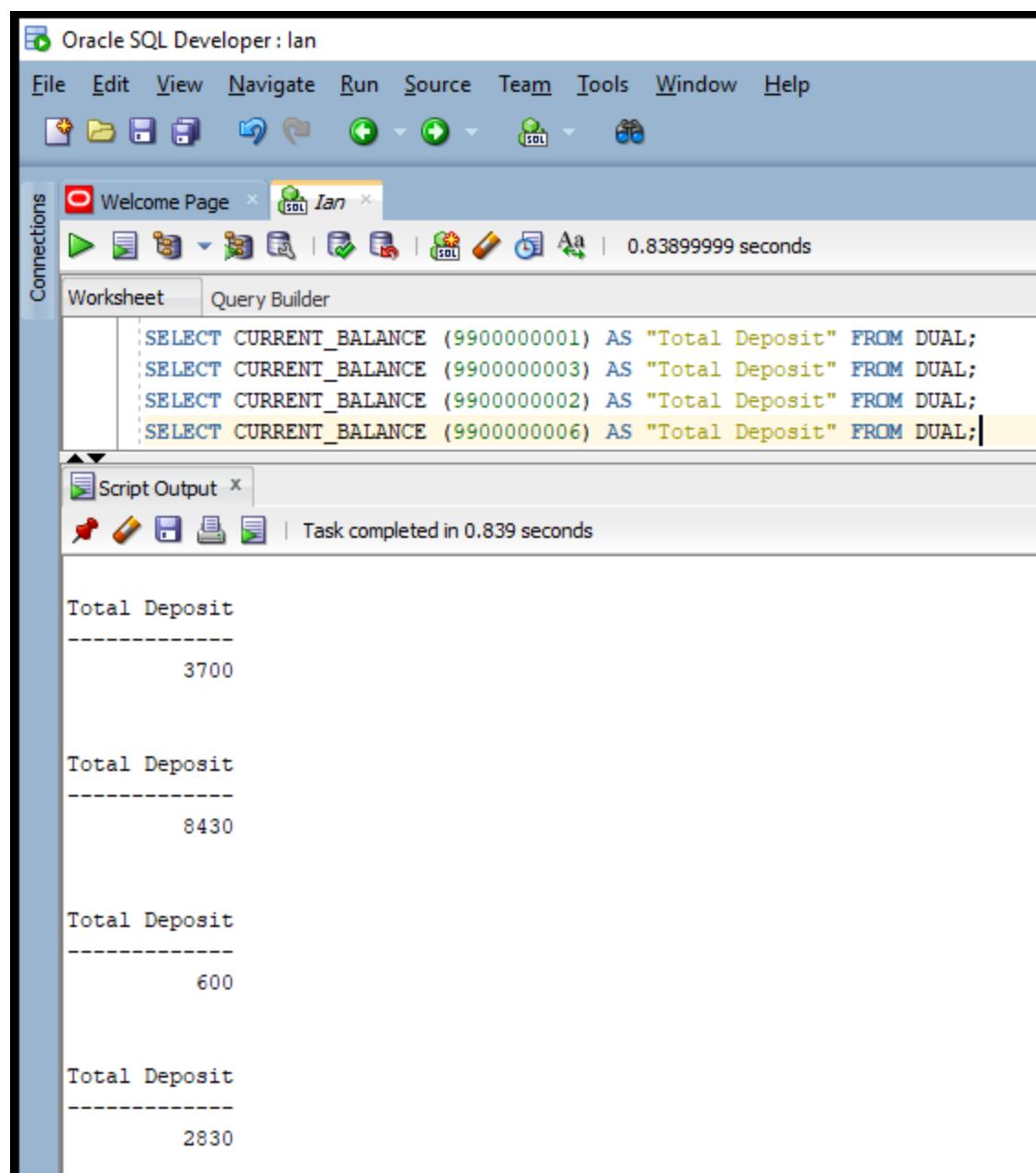
TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP00007	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002
7	CDEP00005	Debit	05-DEC-19 05.48.29.935000000 PM	135	teller withdrawal	T000001
8	CDEP00006	Credit	07-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002
5	CDEP00008	Credit	02-DEC-19 05.48.29.935000000 PM	335	atm deposit	A000001
6	CDEP00008	Debit	03-DEC-19 05.48.29.935000000 PM	810	teller withdrawal	T000003
7	CDEP00009	Credit	05-DEC-19 05.48.29.935000000 PM	735	atm deposit	A000001
9	CDEP00009	Debit	07-DEC-19 05.48.29.935000000 PM	510	teller withdrawal	T000003
4	CDEP00010	Credit	05-DEC-19 05.48.29.935000000 PM	995	atm deposit	A000001
3	CDEP00010	Debit	07-DEC-19 05.48.29.935000000 PM	430	teller withdrawal	T000003
5	CDEP00011	Credit	02-DEC-19 05.48.29.935000000 PM	265	atm deposit	A000001
3	CDEP00011	Debit	03-DEC-19 05.48.29.935000000 PM	910	teller withdrawal	T000003

**Section 3:**

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
4	CDEP00012	Credit	03-DEC-19 05.48.29.935000000 PM	315	atm deposit	A000001
8	CDEP00012	Debit	07-DEC-19 05.48.29.935000000 PM	75	teller withdrawal	T000003

46 rows selected.

## Execute function.



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL execution. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet area contains the following SQL code:

```
SELECT CURRENT_BALANCE (9900000001) AS "Total Deposit" FROM DUAL;
SELECT CURRENT_BALANCE (9900000003) AS "Total Deposit" FROM DUAL;
SELECT CURRENT_BALANCE (9900000002) AS "Total Deposit" FROM DUAL;
SELECT CURRENT_BALANCE (9900000006) AS "Total Deposit" FROM DUAL;
```

Below the worksheet is a "Script Output" window showing the results of the execution:

```
Total Deposit
-----
3700

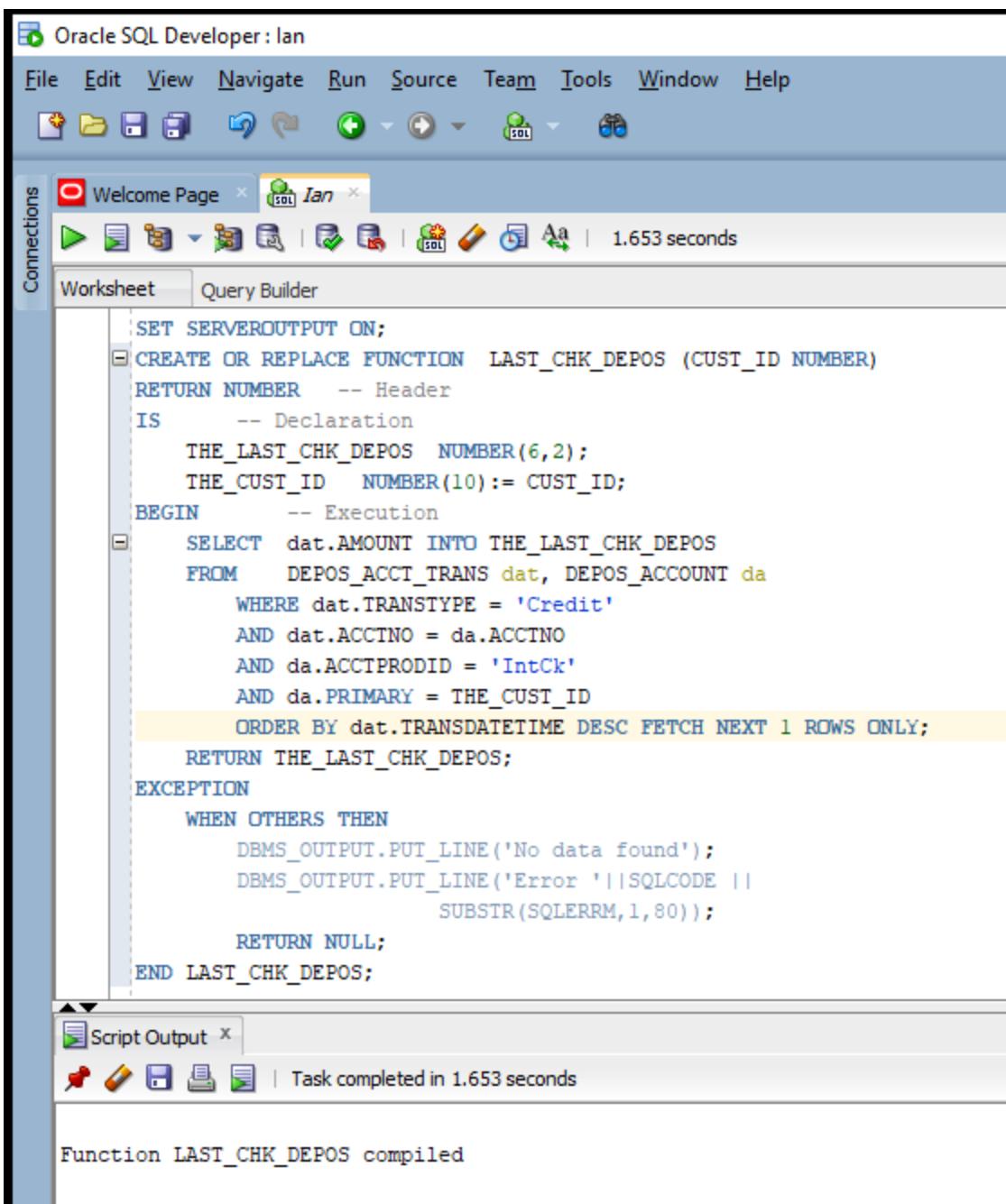
Total Deposit
-----
8430

Total Deposit
-----
600

Total Deposit
-----
2830
```

The status bar at the bottom of the interface indicates "0.838999999 seconds".

## Create function.

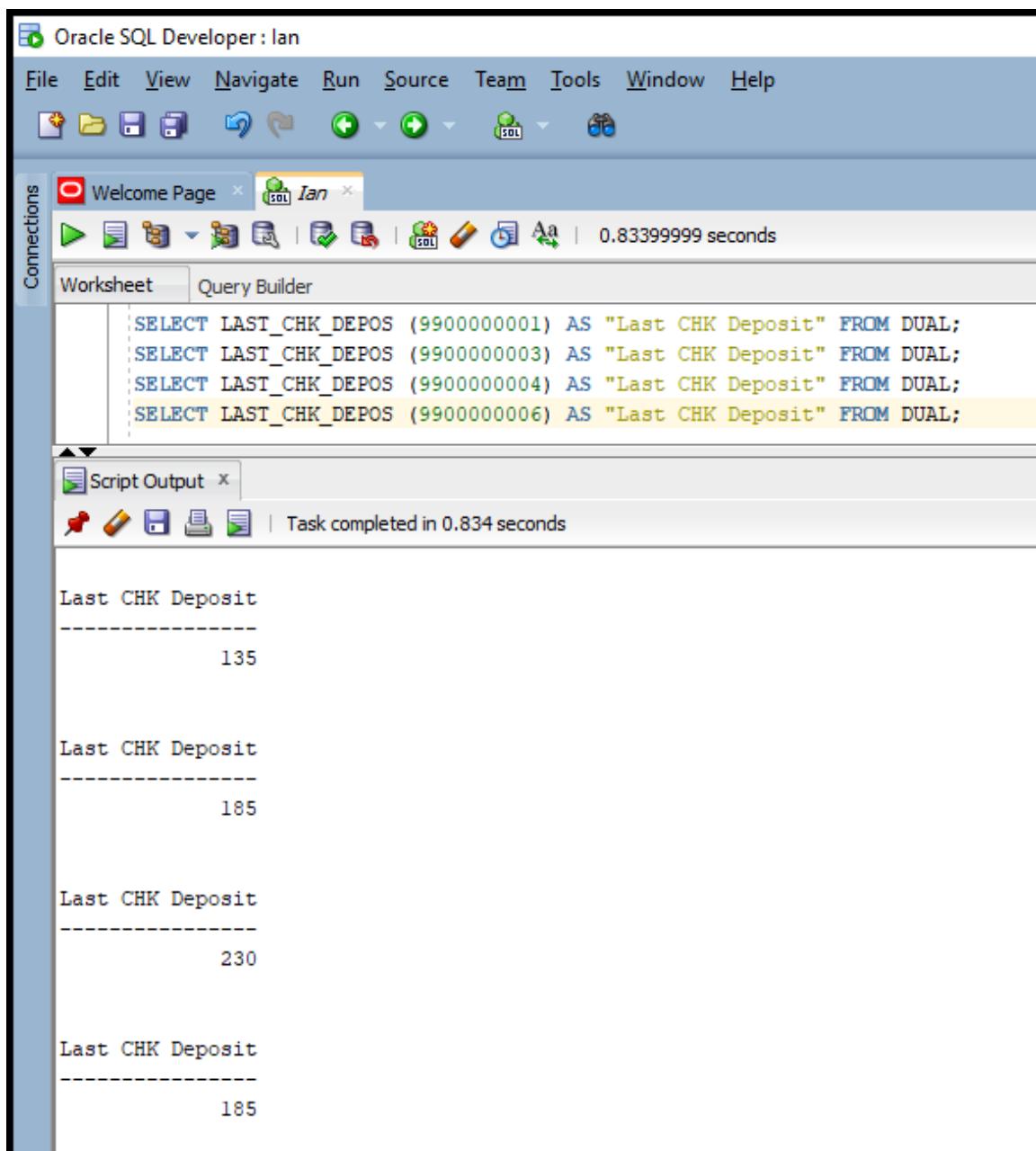


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION LAST_CHK_DEPOS (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_CHK_DEPOS NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_CHK_DEPOS
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Credit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'IntCk'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_CHK_DEPOS;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_CHK_DEPOS;
```

The 'Script Output' tab at the bottom shows the message: 'Function LAST\_CHK\_DEPOS compiled'.

## Execute function.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT LAST_CHK_DEPOS (9900000001) AS "Last CHK Deposit" FROM DUAL;
SELECT LAST_CHK_DEPOS (9900000003) AS "Last CHK Deposit" FROM DUAL;
SELECT LAST_CHK_DEPOS (9900000004) AS "Last CHK Deposit" FROM DUAL;
SELECT LAST_CHK_DEPOS (9900000006) AS "Last CHK Deposit" FROM DUAL;
```

The 'Script Output' tab shows the results of the execution:

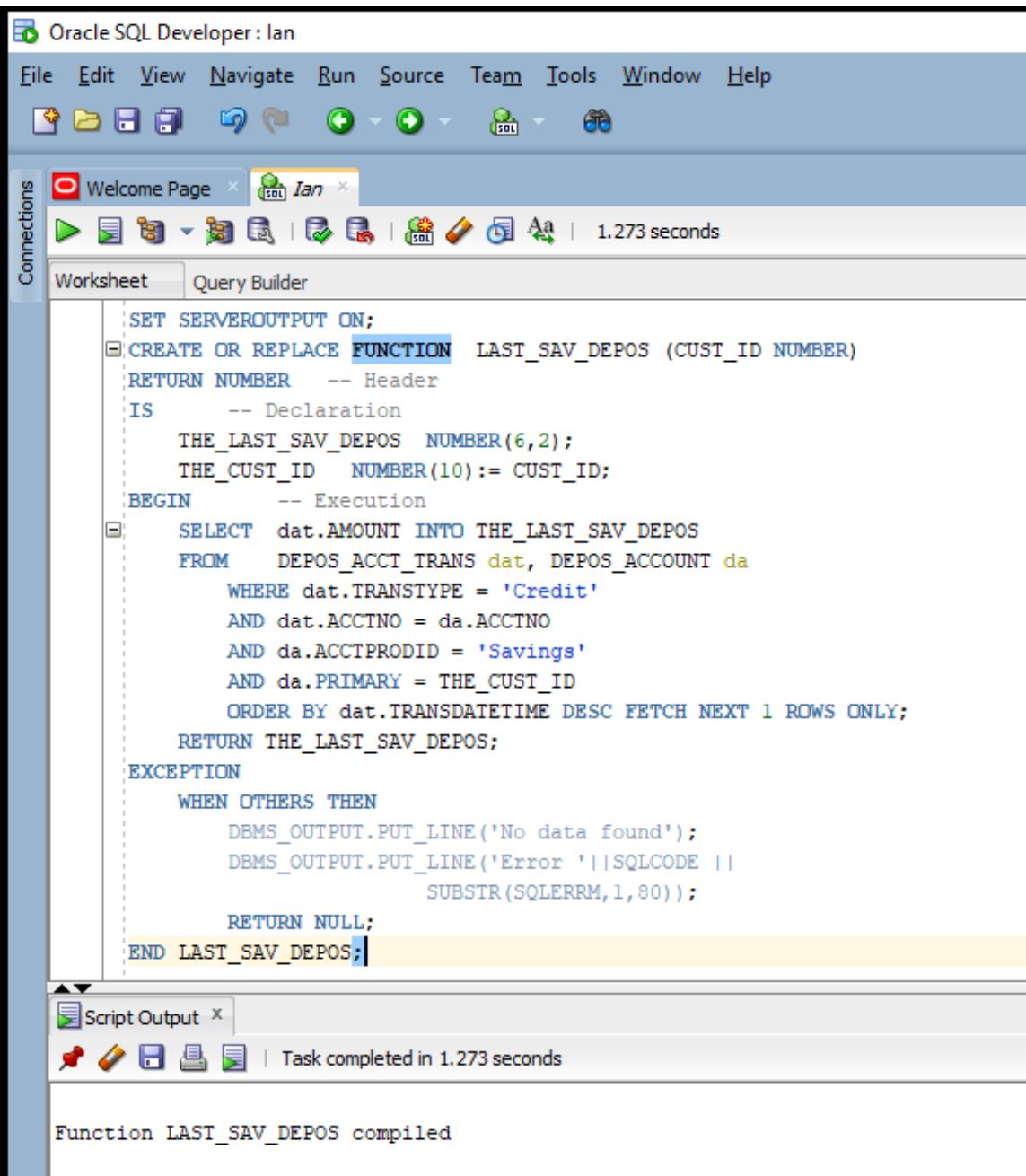
```
Last CHK Deposit
-----
135

Last CHK Deposit
-----
185

Last CHK Deposit
-----
230

Last CHK Deposit
-----
185
```

## Create function.

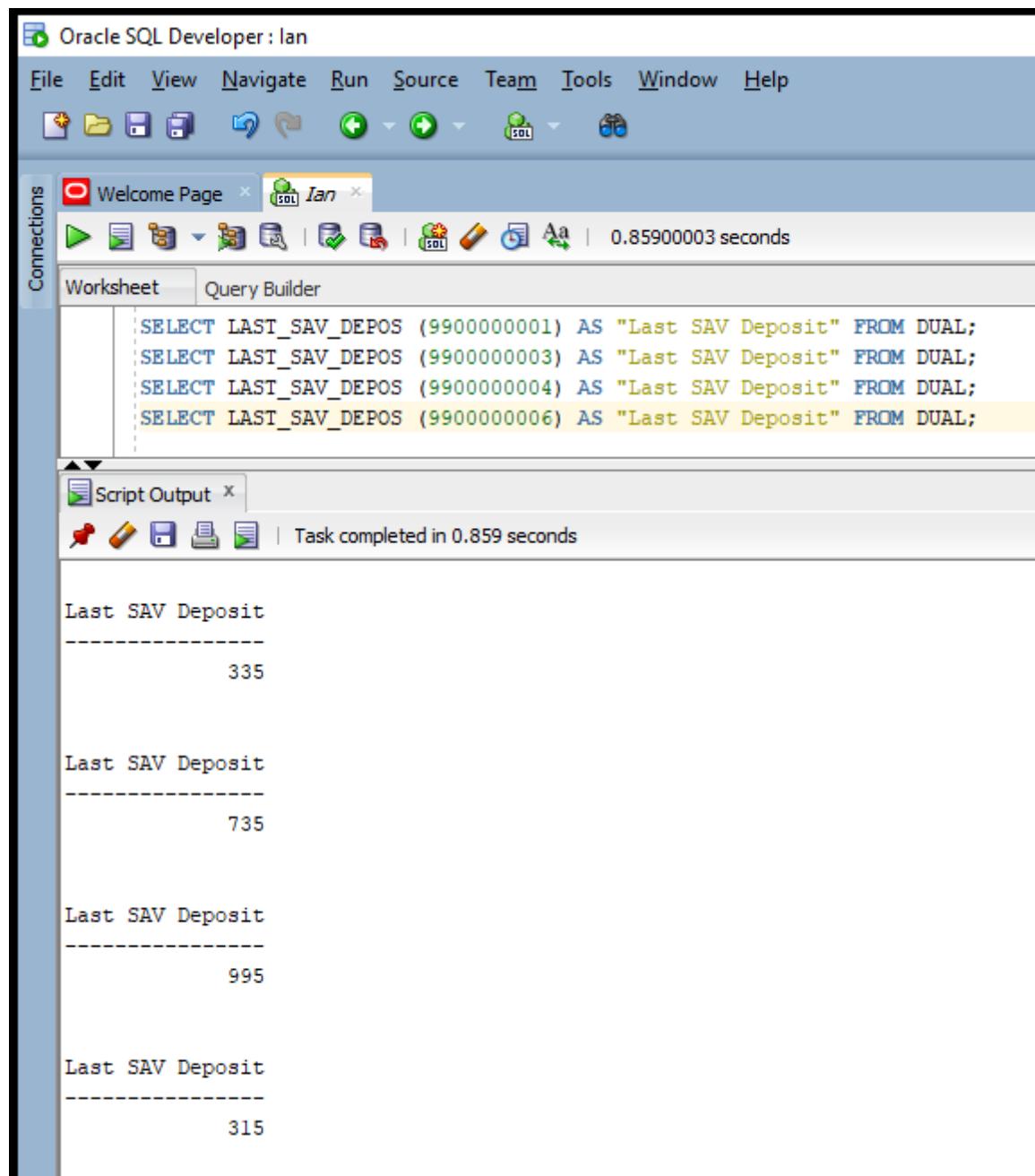


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION LAST_SAV_DEPOS (CUST_ID NUMBER)
  RETURN NUMBER -- Header
  IS -- Declaration
    THE_LAST_SAV_DEPOS NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
  BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_SAV_DEPOS
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Credit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'Savings'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_SAV_DEPOS;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('No data found');
      DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
      RETURN NULL;
  END LAST_SAV_DEPOS;
```

The 'Script Output' tab at the bottom shows the message: 'Function LAST\_SAV\_DEPOS compiled'.

## Execute function.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT LAST_SAV_DEPOS (9900000001) AS "Last SAV Deposit" FROM DUAL;
SELECT LAST_SAV_DEPOS (9900000003) AS "Last SAV Deposit" FROM DUAL;
SELECT LAST_SAV_DEPOS (9900000004) AS "Last SAV Deposit" FROM DUAL;
SELECT LAST_SAV_DEPOS (9900000006) AS "Last SAV Deposit" FROM DUAL;
```

The 'Script Output' tab shows the results of the execution:

```
Last SAV Deposit
-----
335

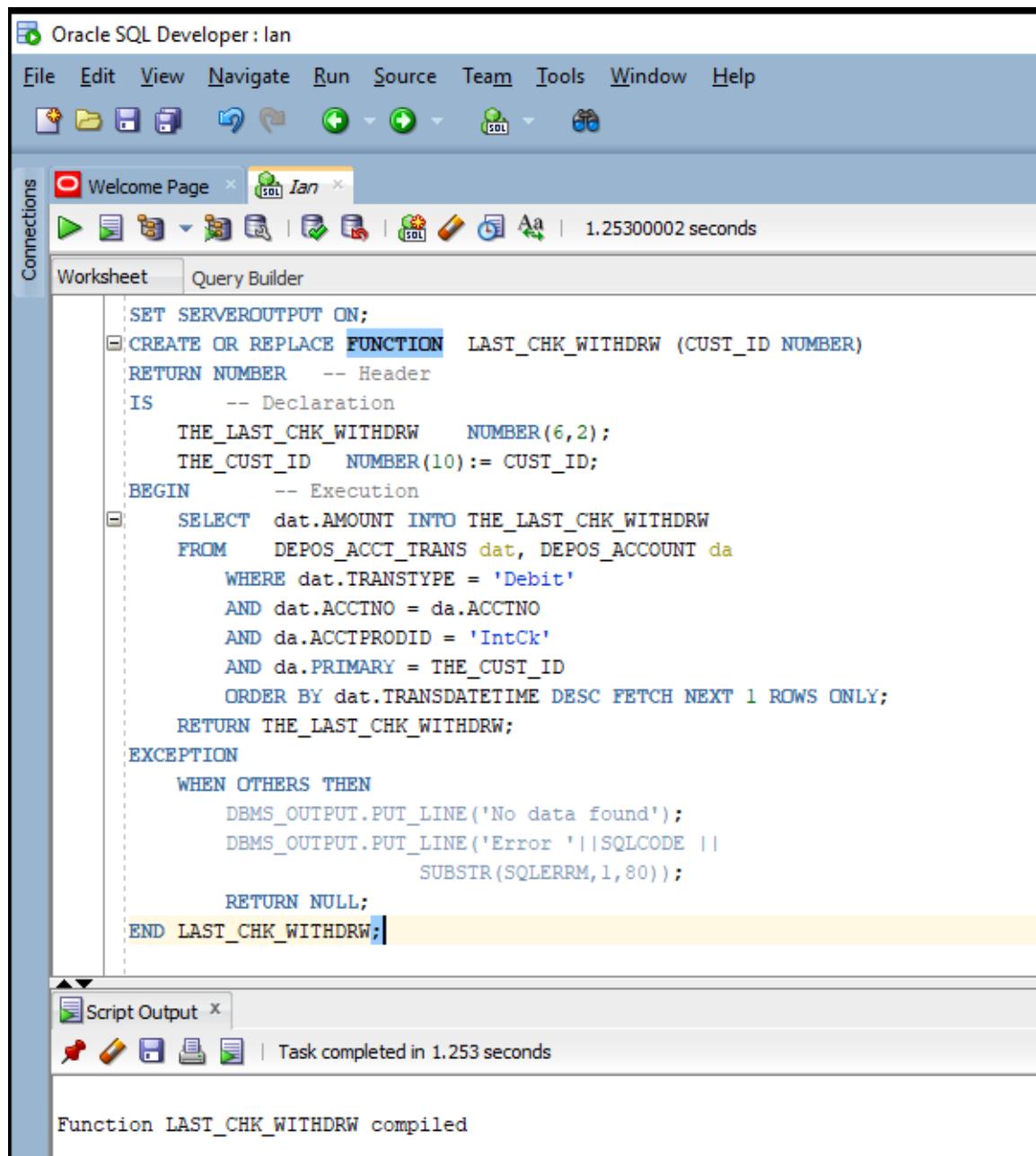
Last SAV Deposit
-----
735

Last SAV Deposit
-----
995

Last SAV Deposit
-----
315
```

The status bar at the bottom of the interface indicates "Task completed in 0.859 seconds".

## Create function.

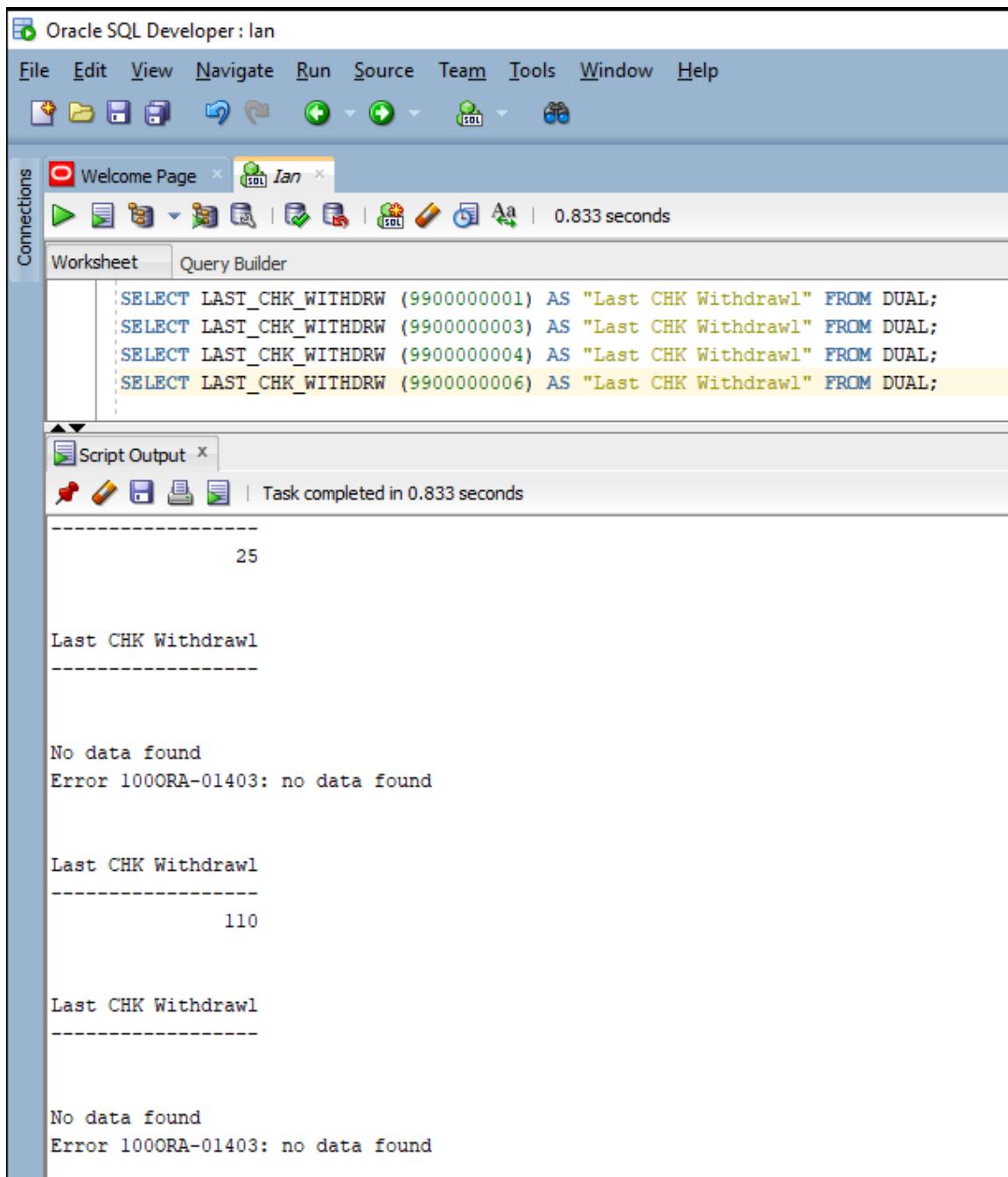


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION LAST_CHK_WITHDRW (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_CHK_WITHDRW NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_CHK_WITHDRW
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Debit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'IntCk'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_CHK_WITHDRW;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_CHK_WITHDRW;
```

The 'Script Output' tab at the bottom shows the message: 'Function LAST\_CHK\_WITHDRW compiled'.

## Execute function.



The screenshot shows the Oracle SQL Developer interface with a connection named 'lan'. The 'Worksheet' tab is active, displaying a series of SQL queries. The queries are as follows:

```
SELECT LAST_CHK_WITHDRW (9900000001) AS "Last CHK Withdrawl" FROM DUAL;
SELECT LAST_CHK_WITHDRW (9900000003) AS "Last CHK Withdrawl" FROM DUAL;
SELECT LAST_CHK_WITHDRW (9900000004) AS "Last CHK Withdrawl" FROM DUAL;
SELECT LAST_CHK_WITHDRW (9900000006) AS "Last CHK Withdrawl" FROM DUAL;
```

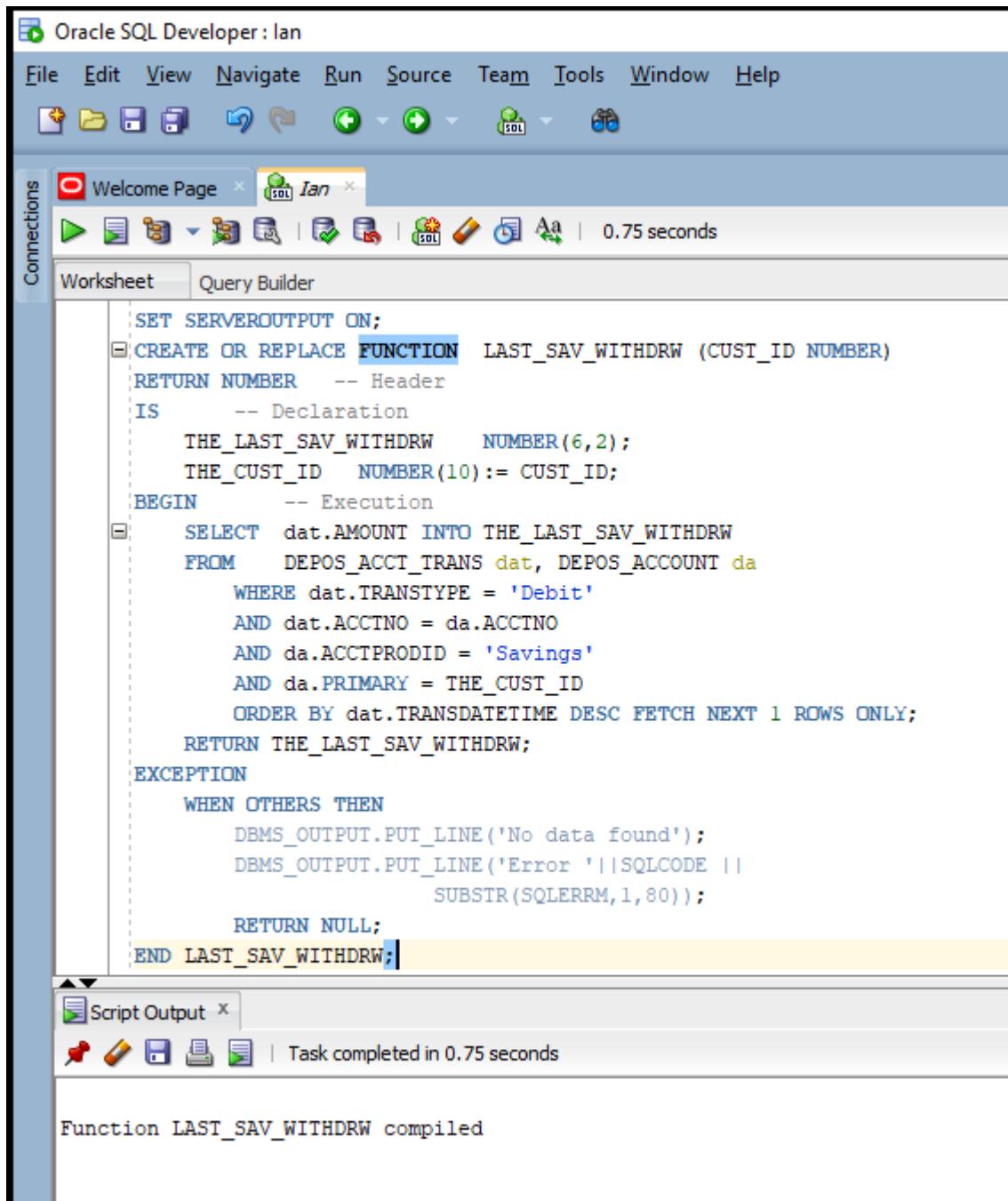
Below the worksheet, the 'Script Output' tab shows the results of the execution. The output is as follows:

```
25
Last CHK Withdrawl
-----
No data found
Error 1000ORA-01403: no data found

Last CHK Withdrawl
-----
110

Last CHK Withdrawl
-----
No data found
Error 1000ORA-01403: no data found
```

## Create function.



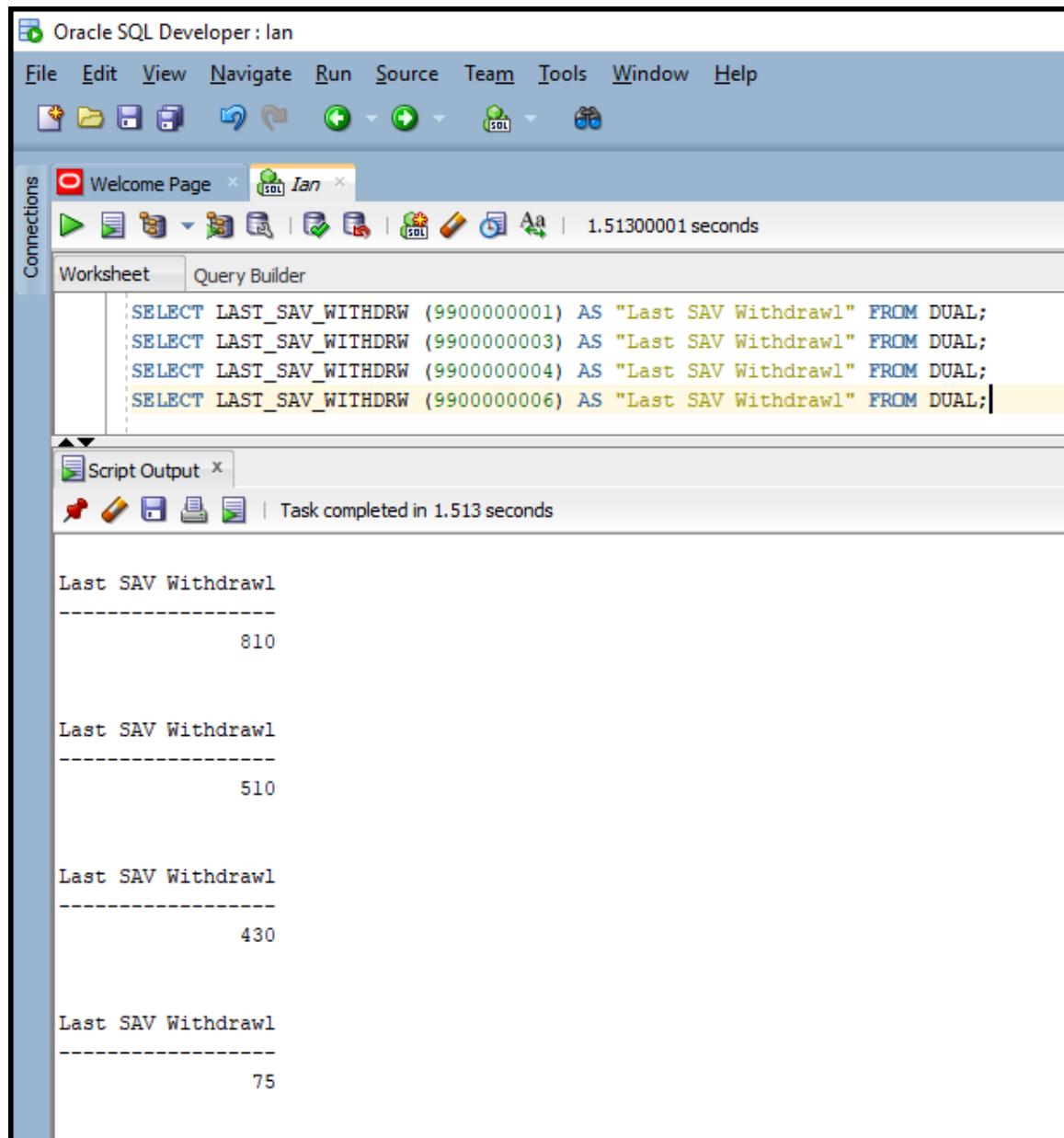
The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** A list of connections, with one named "Ian" selected.
- Worksheet:** The main area where the SQL code is entered. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION LAST_SAV_WITHDRW (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_SAV_WITHDRW    NUMBER(6,2);
    THE_CUST_ID   NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_SAV_WITHDRW
    FROM   DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Debit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'Savings'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_SAV_WITHDRW;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_SAV_WITHDRW;
```

- Script Output:** A panel at the bottom showing the result of the compilation: "Function LAST\_SAV\_WITHDRW compiled".
- Toolbar:** Standard SQL Developer toolbar with icons for file operations, navigation, and run.
- Header:** "Welcome Page" and "Ian" are shown in the title bar.

## Execute function.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT LAST_SAV_WITHDRW (9900000001) AS "Last SAV Withdrawl" FROM DUAL;
SELECT LAST_SAV_WITHDRW (9900000003) AS "Last SAV Withdrawl" FROM DUAL;
SELECT LAST_SAV_WITHDRW (9900000004) AS "Last SAV Withdrawl" FROM DUAL;
SELECT LAST_SAV_WITHDRW (9900000006) AS "Last SAV Withdrawl" FROM DUAL;
```

The 'Script Output' tab shows the results of the execution:

```
Last SAV Withdrawl
-----
810

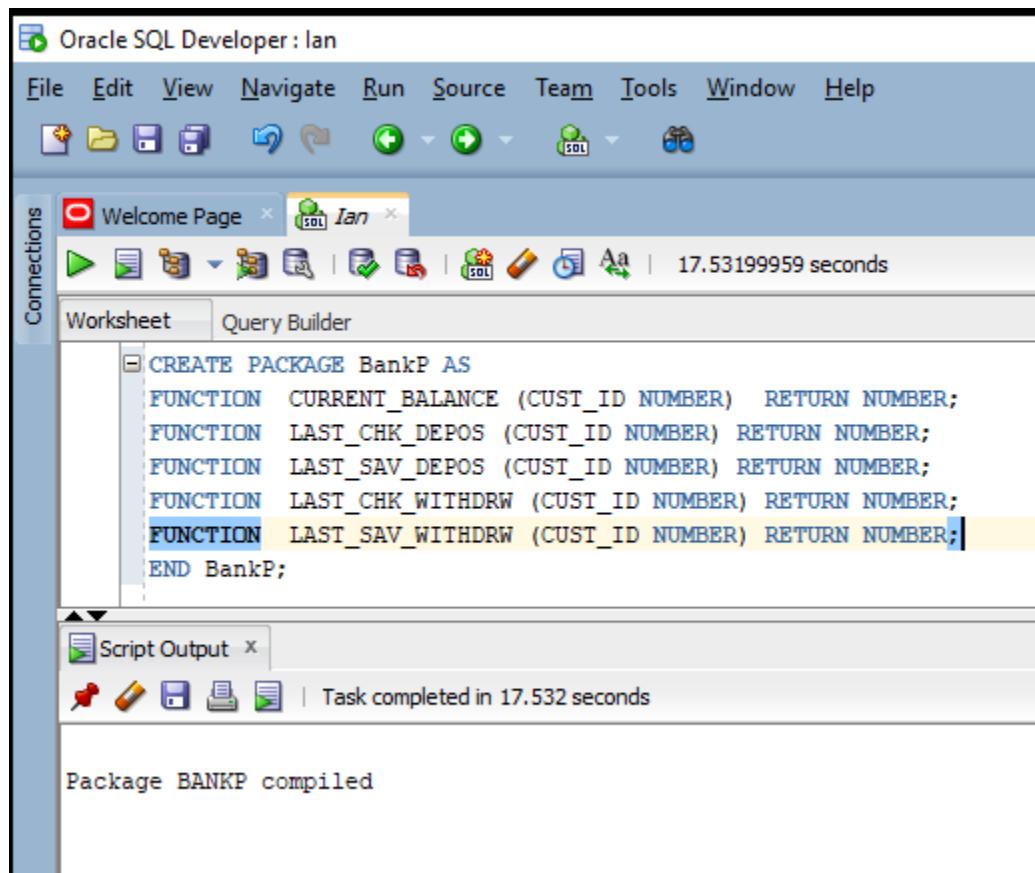
Last SAV Withdrawl
-----
510

Last SAV Withdrawl
-----
430

Last SAV Withdrawl
-----
75
```

The status bar at the bottom of the interface indicates "Task completed in 1.513 seconds".

## Create package containing several functions



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and navigation. The Connections sidebar shows a single connection named "Ian". The main workspace has two tabs: "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet area contains the following PL/SQL code:

```
CREATE PACKAGE BankP AS
  FUNCTION CURRENT_BALANCE (CUST_ID NUMBER) RETURN NUMBER;
  FUNCTION LAST_CHK_DEPOS (CUST_ID NUMBER) RETURN NUMBER;
  FUNCTION LAST_SAV_DEPOS (CUST_ID NUMBER) RETURN NUMBER;
  FUNCTION LAST_CHK_WITHDRW (CUST_ID NUMBER) RETURN NUMBER;
  FUNCTION LAST_SAV_WITHDRW (CUST_ID NUMBER) RETURN NUMBER;
END BankP;
```

The "LAST\_SAV\_WITHDRW" function is currently selected. Below the worksheet is a "Script Output" window showing the message "Task completed in 17.532 seconds". The status bar at the bottom of the interface displays "Package BANKP compiled".

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page × Ian ×

Worksheet Query Builder

```
CREATE PACKAGE BODY BankP AS
FUNCTION CURRENT_BALANCE (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_BALANCE    NUMBER(6,2);
    THE_CUST_ID   NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT SUM(BALANCE)      INTO THE_BALANCE
    FROM DEPOS_ACCOUNT WHERE PRIMARY = THE_CUST_ID;
    RETURN THE_BALANCE;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END CURRENT_BALANCE;
-----
FUNCTION LAST_CHK_DEPOS (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_CHK_DEPOS NUMBER(6,2);
    THE_CUST_ID   NUMBER(10):= CUST_ID;
BEGIN -- Execution

    SELECT dat.AMOUNT INTO THE_LAST_CHK_DEPOS

```

Script Output ×

Task completed in 1.557 seconds

Package Body BANKP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
END CURRENT_BALANCE;

FUNCTION LAST_CHK_DEPOS (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_CHK_DEPOS NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_CHK_DEPOS
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Credit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'IntCk'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_CHK_DEPOS;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                            SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_CHK_DEPOS;

FUNCTION LAST_SAV_DEPOS (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_SAV_DEPOS NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_SAV_DEPOS
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Save'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'IntCk'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_SAV_DEPOS;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                            SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_SAV_DEPOS;
```

Script Output

Task completed in 1.557 seconds

Package Body BANKP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
        RETURN NULL;
END LAST_CHK_DEPOS;
-----
FUNCTION LAST_SAV_DEPOS (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_SAV_DEPOS NUMBER(6,2);
    THE_CUST_ID NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_SAV_DEPOS
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
        WHERE dat.TRANSTYPE = 'Credit'
        AND dat.ACCTNO = da.ACCTNO
        AND da.ACCTPRODID = 'Savings'
        AND da.PRIMARY = THE_CUST_ID
        ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_SAV_DEPOS;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
            SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_SAV_DEPOS;
-----
FUNCTION LAST_CHK_WITHDRW (CUST_ID NUMBER)
RETURN NUMBER -- Header
```

Script Output

Task completed in 1.557 seconds

Package Body BANKP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
END LAST_SAV_DEPOS;

-----
FUNCTION LAST_CHK_WITHDRW (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
    THE_LAST_CHK_WITHDRW    NUMBER(6,2);
    THE_CUST_ID   NUMBER(10):= CUST_ID;
BEGIN -- Execution
    SELECT dat.AMOUNT INTO THE_LAST_CHK_WITHDRW
    FROM   DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da
    WHERE dat.TRANSTYPE = 'Debit'
    AND dat.ACCTNO = da.ACCTNO
    AND da.ACCTPRODID = 'IntCk'
    AND da.PRIMARY = THE_CUST_ID
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;
    RETURN THE_LAST_CHK_WITHDRW;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
        RETURN NULL;
END LAST_CHK_WITHDRW;
-----
FUNCTION LAST_SAV_WITHDRW (CUST_ID NUMBER)
RETURN NUMBER -- Header
IS -- Declaration
```

Script Output X

Task completed in 1.557 seconds

Package Body BANKP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||  
SUBSTR(SQLERRM,1,80));  
RETURN NULL;  
END LAST_CHK_WITHDRW;  
-----  
FUNCTION LAST_SAV_WITHDRW (CUST_ID NUMBER)  
RETURN NUMBER -- Header  
IS -- Declaration  
    THE_LAST_SAV_WITHDRW    NUMBER(6,2);  
    THE_CUST_ID   NUMBER(10):= CUST_ID;  
BEGIN -- Execution  
    SELECT dat.AMOUNT INTO THE_LAST_SAV_WITHDRW  
    FROM DEPOS_ACCT_TRANS dat, DEPOS_ACCOUNT da  
    WHERE dat.TRANSTYPE = 'Debit'  
    AND dat.ACCTNO = da.ACCTNO  
    AND da.ACCTPRODID = 'Savings'  
    AND da.PRIMARY = THE_CUST_ID  
    ORDER BY dat.TRANSDATETIME DESC FETCH NEXT 1 ROWS ONLY;  
    RETURN THE_LAST_SAV_WITHDRW;  
EXCEPTION  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('No data found');  
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||  
SUBSTR(SQLERRM,1,80));  
        RETURN NULL;  
END LAST_SAV_WITHDRW;  
END BankP;
```

Script Output | Task completed in 1.557 seconds

Package Body BANKP compiled

^^^^^^^^^^^^^^^^^^^^^^^^

## **CHAPTER 4B Starts here**

^^^^^^^^^^^^^^^^^^^^^^^^

B. Create a package with the following functions & procedures called **BranchP** by branch id.

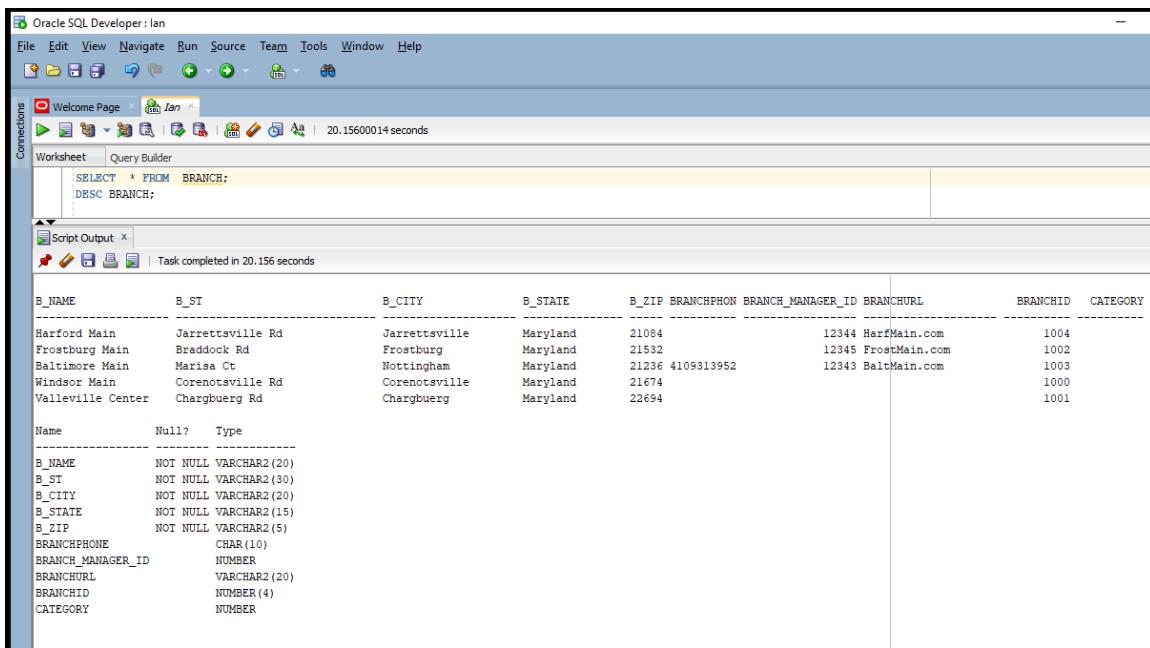
a--Function to return the current branch address.

b--Function to return the current branch phone number.

c--Procedure to output the name of employees working at that branch.

## Check table contents

### Check table attribute data types



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

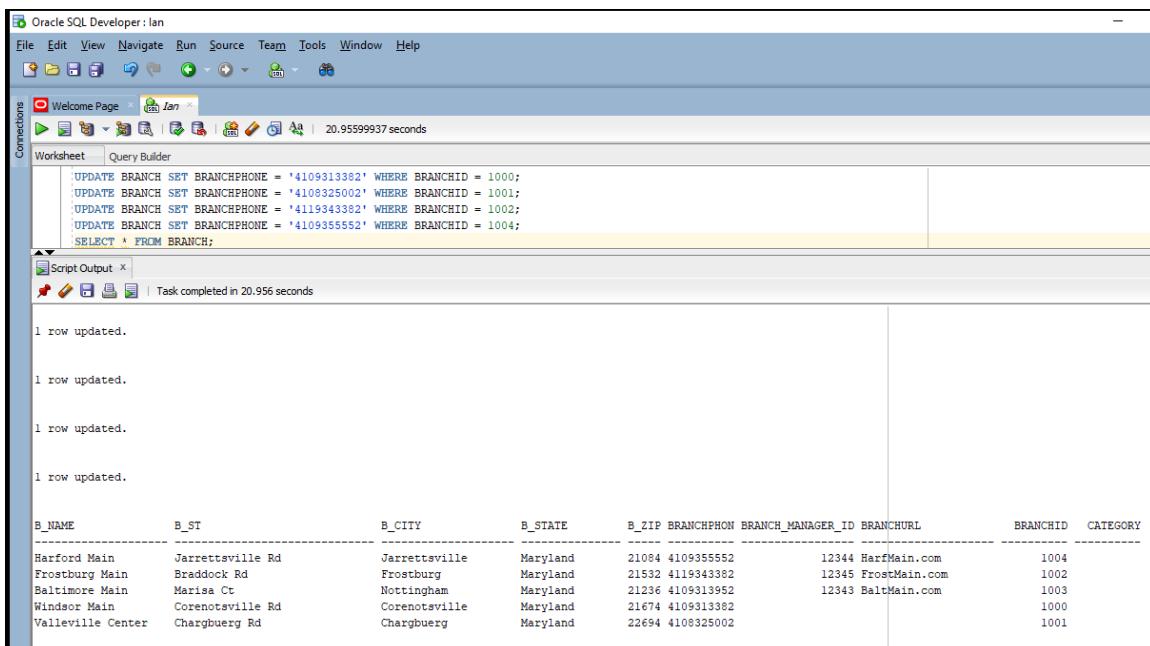
SELECT \* FROM BRANCH;  
DESC BRANCH;

Task completed in 20.156 seconds

B_NAME	B_ST	B_CITY	B_STATE	B_ZIP	BRANCHPHON	BRANCH_MANAGER_ID	BRANCHURL	BRANCHID	CATEGORY
Harford Main	Jarrettsville Rd	Jarrettsville	Maryland	21084		12344	HarfMain.com	1004	
Frostburg Main	Braddock Rd	Frostburg	Maryland	21532		12345	FrostMain.com	1002	
Baltimore Main	Marisa Ct	Nottingham	Maryland	21236	4109313952	12343	BaltMain.com	1003	
Windsor Main	Corenotsville Rd	Corenotsville	Maryland	21674				1000	
Valleville Center	Chargbuerg Rd	Chargbuerg	Maryland	22694				1001	

Name Null? Type

B\_NAME NOT NULL VARCHAR2(20)  
B\_ST NOT NULL VARCHAR2(30)  
B\_CITY NOT NULL VARCHAR2(20)  
B\_STATE NOT NULL VARCHAR2(15)  
B\_ZIP NOT NULL VARCHAR2(5)  
BRANCHPHONE CHAR(10)  
BRANCH\_MANAGER\_ID NUMBER  
BRANCHURL VARCHAR2(20)  
BRANCHID NUMBER(4)  
CATEGORY NUMBER



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

UPDATE BRANCH SET BRANCHPHONE = '4109313382' WHERE BRANCHID = 1000;  
UPDATE BRANCH SET BRANCHPHONE = '4108325002' WHERE BRANCHID = 1001;  
UPDATE BRANCH SET BRANCHPHONE = '4119343382' WHERE BRANCHID = 1002;  
UPDATE BRANCH SET BRANCHPHONE = '4109355552' WHERE BRANCHID = 1004;  
SELECT \* FROM BRANCH;

Task completed in 20.95599937 seconds

1 row updated.

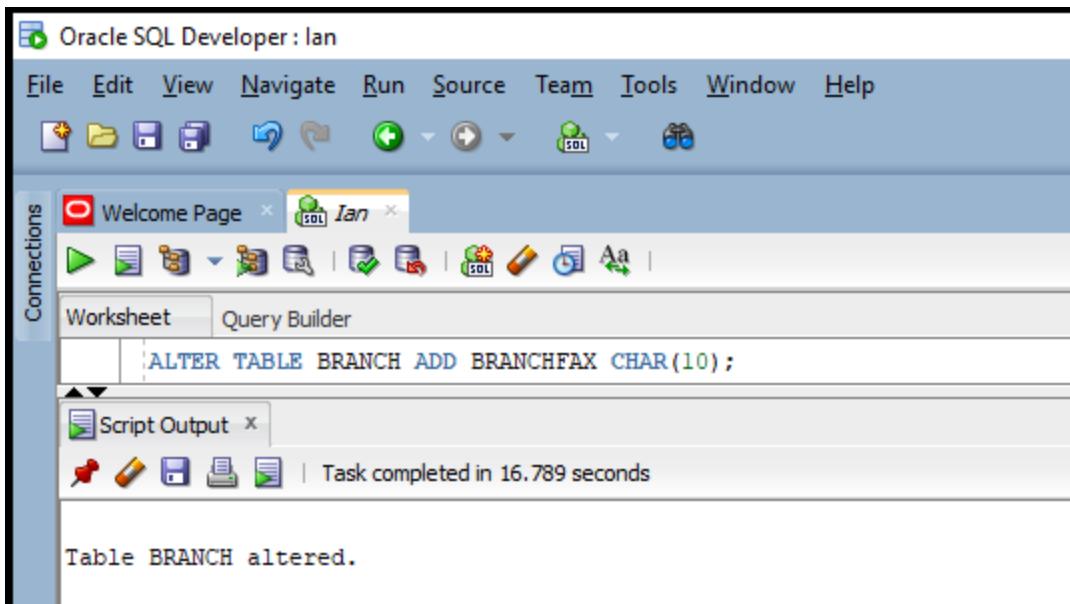
1 row updated.

1 row updated.

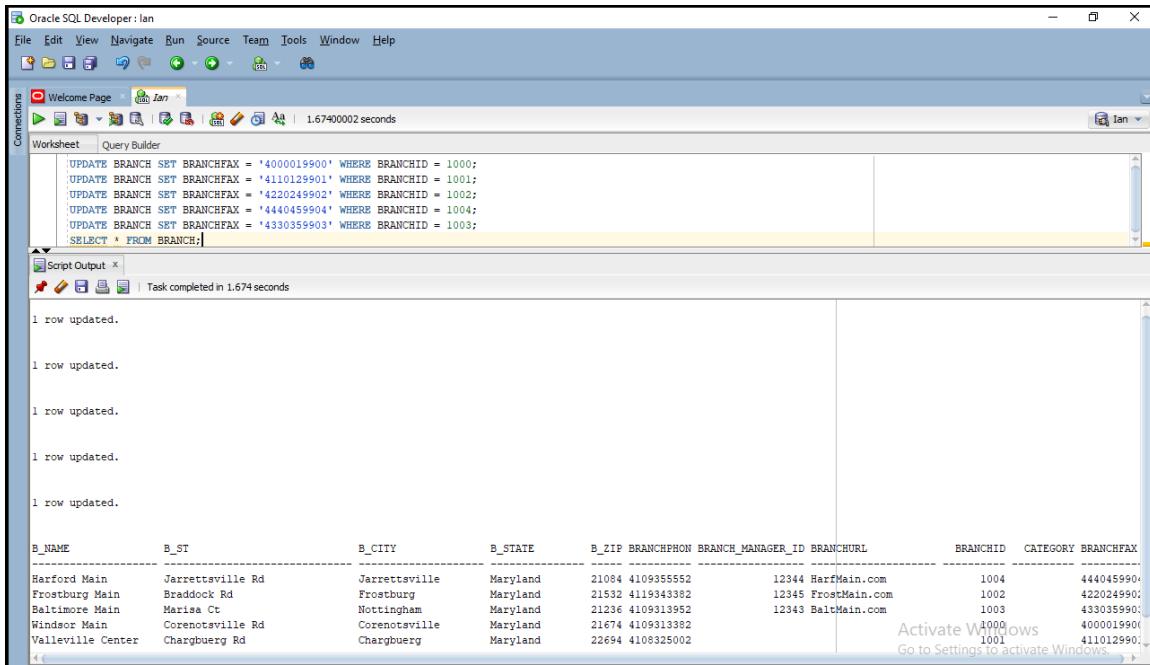
1 row updated.

B_NAME	B_ST	B_CITY	B_STATE	B_ZIP	BRANCHPHON	BRANCH_MANAGER_ID	BRANCHURL	BRANCHID	CATEGORY
Harford Main	Jarrettsville Rd	Jarrettsville	Maryland	21084	4109355552	12344	HarfMain.com	1004	
Frostburg Main	Braddock Rd	Frostburg	Maryland	21532	4119343382	12345	FrostMain.com	1002	
Baltimore Main	Marisa Ct	Nottingham	Maryland	21236	4109313952	12343	BaltMain.com	1003	
Windsor Main	Corenotsville Rd	Corenotsville	Maryland	21674	4109313382			1000	
Valleville Center	Chargbuerg Rd	Chargbuerg	Maryland	22694	4108325002			1001	

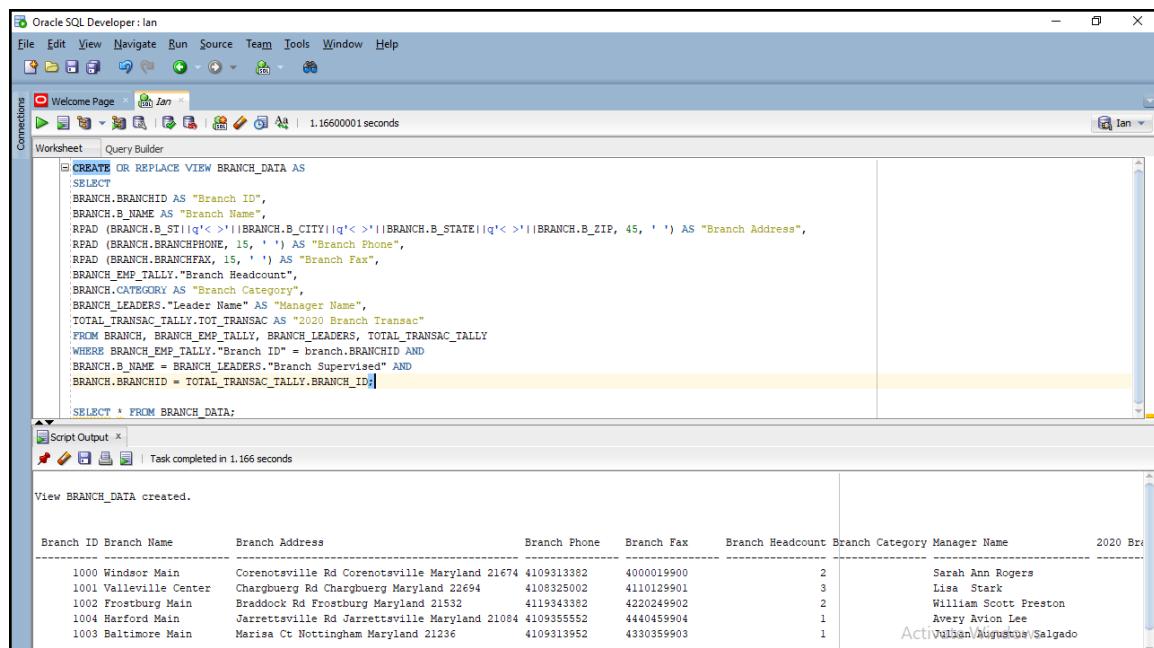
## Alter table branch with new attribute.



## Update table branch.



## Modify view branch\_data



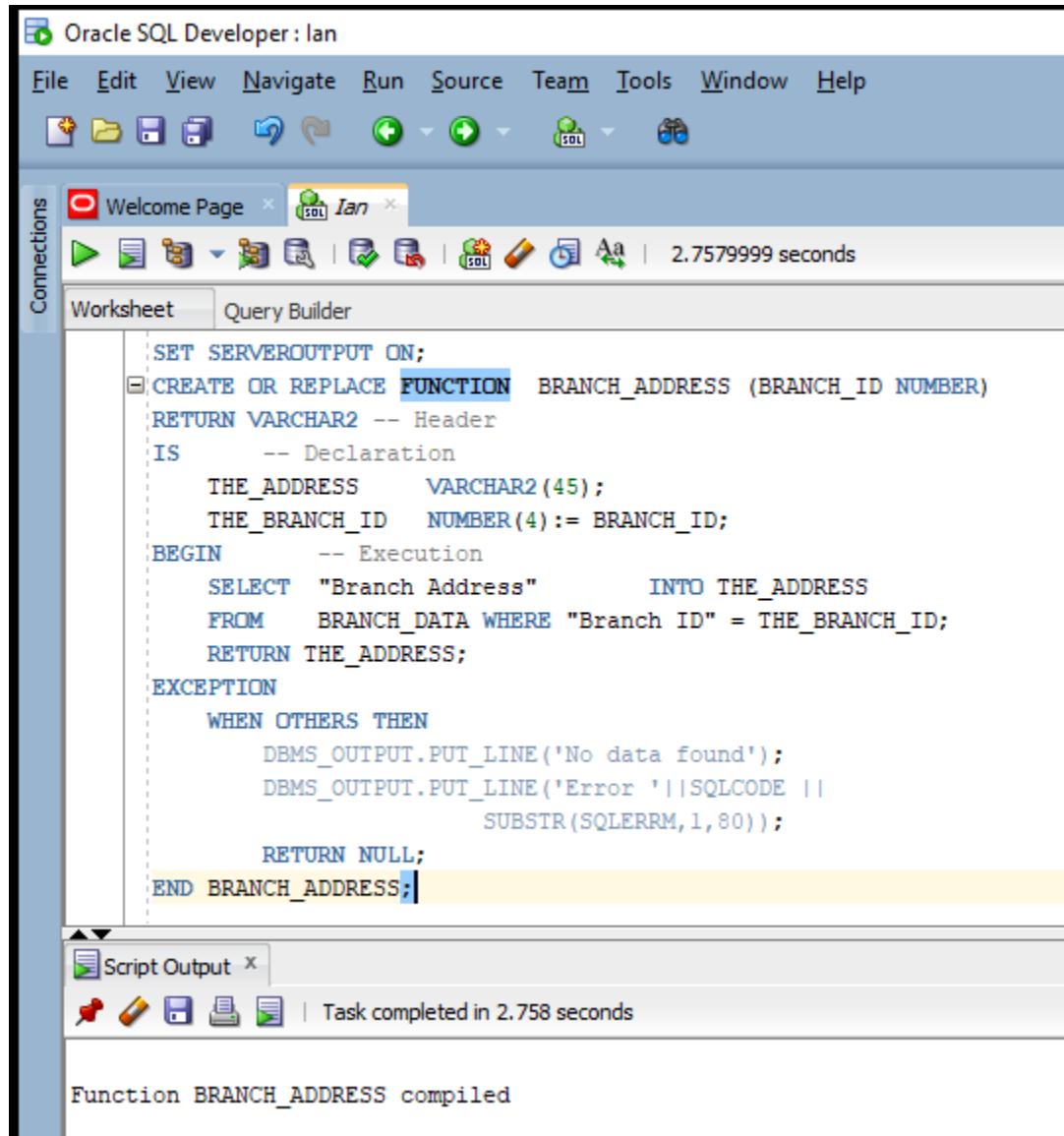
The screenshot shows the Oracle SQL Developer interface with a connection named 'ian'. The 'Worksheet' tab is active, displaying the SQL code for creating a view named 'branch\_data'. The code includes a SELECT statement that joins several tables: 'branch', 'branch\_emp\_tally', 'branch\_leaders', and 'total\_transac\_tally'. It retrieves columns such as Branch ID, Name, Address, Phone, Fax, Headcount, Category, and Manager Name. The 'Script Output' tab shows the message 'View branch\_data created.' and the execution time 'Task completed in 1.166 seconds'. Below the output, a table displays the data for the newly created view.

```
CREATE OR REPLACE VIEW branch_data AS
SELECT
  branch.BRANCHID AS "Branch ID",
  branch.B_NAME AS "Branch Name",
  RPAD (branch.B_ST||q'<'||branch.B_CITY||q'<'||branch.B_STATE||q'<'||branch.B_ZIP, 45, ' ') AS "Branch Address",
  RPAD (branch.BRANCHPHONE, 15, ' ') AS "Branch Phone",
  RPAD (branch.BRANCHFAX, 15, ' ') AS "Branch Fax",
  branch.BRANCHCATEGORY AS "Branch Category",
  branch.CATEGORY AS "Branch Category",
  branch.LEADERS."Leader Name" AS "Manager Name",
  total_transac_tally.TOT_TRANSAC AS "2020 Branch Transac"
FROM branch, branch_emp_tally, branch_leaders, total_transac_tally
WHERE branch.BRANCHID = branch_emp_tally."Branch ID" AND
branch.B_NAME = branch_leaders."Branch Supervised" AND
branch.BRANCHID = total_transac_tally.BRANCH_ID;

SELECT * FROM branch_data;
```

Branch ID	Branch Name	Branch Address	Branch Phone	Branch Fax	Branch Headcount	Branch Category	Manager Name	2020 Br
1000	Windsor Main	Corensville Rd Corensville Maryland 21674	4109313382	4000019900	2	Sarah Ann Rogers		
1001	Valleville Center	Chargburg Rd Chargburg Maryland 22694	4108325002	4110129901	3	Lisa Stark		
1002	Frostburg Main	Braddock Rd Frostburg Maryland 21532	4119343382	4220249902	2	William Scott Preston		
1004	Harford Main	Jarrettsville Rd Jarrettsville Maryland 21084	4109355552	4440459904	1	Avery Avion Lee		
1003	Baltimore Main	Marisa Ct Nottingham Maryland 21236	4109313952	4330359903	1	Julian Augustus Salgado		

## Create function.

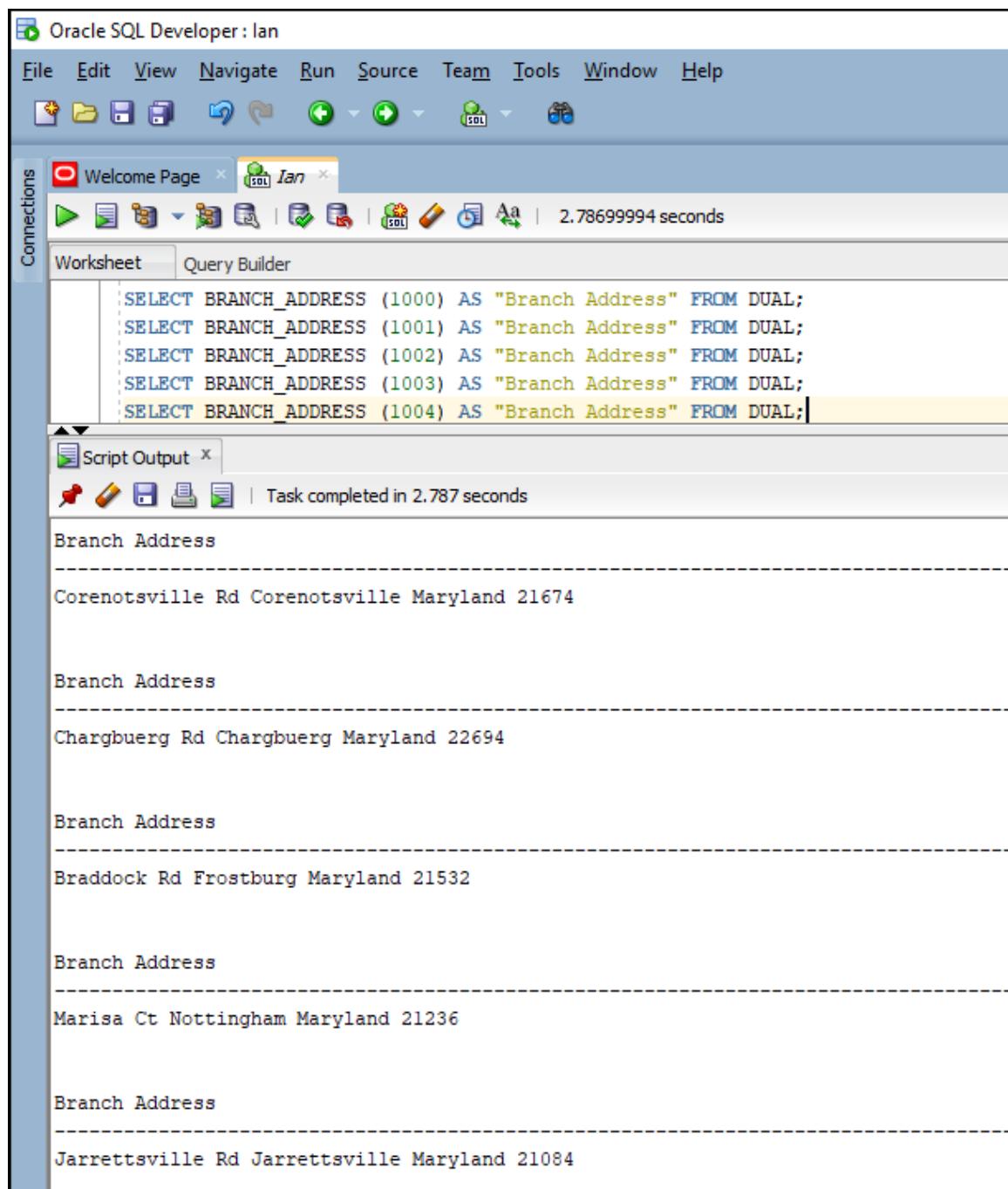


The screenshot shows the Oracle SQL Developer interface with a connection named 'lan'. The 'Worksheet' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION BRANCH_ADDRESS (BRANCH_ID NUMBER)
  RETURN VARCHAR2 -- Header
IS
  -- Declaration
  THE_ADDRESS      VARCHAR2(45);
  THE_BRANCH_ID    NUMBER(4):= BRANCH_ID;
BEGIN
  -- Execution
  SELECT "Branch Address"      INTO THE_ADDRESS
  FROM   BRANCH_DATA WHERE "Branch ID" = THE_BRANCH_ID;
  RETURN THE_ADDRESS;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
    RETURN NULL;
END BRANCH_ADDRESS;
```

The 'Script Output' tab at the bottom shows the message: 'Function BRANCH\_ADDRESS compiled'.

## Execute function.



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL execution. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The query editor contains the following SQL code:

```
SELECT BRANCH_ADDRESS (1000) AS "Branch Address" FROM DUAL;
SELECT BRANCH_ADDRESS (1001) AS "Branch Address" FROM DUAL;
SELECT BRANCH_ADDRESS (1002) AS "Branch Address" FROM DUAL;
SELECT BRANCH_ADDRESS (1003) AS "Branch Address" FROM DUAL;
SELECT BRANCH_ADDRESS (1004) AS "Branch Address" FROM DUAL;
```

The "Script Output" tab shows the results of the execution:

```
Branch Address
Corenotsville Rd Corenotsville Maryland 21674

Branch Address
Chargbuerg Rd Chargbuerg Maryland 22694

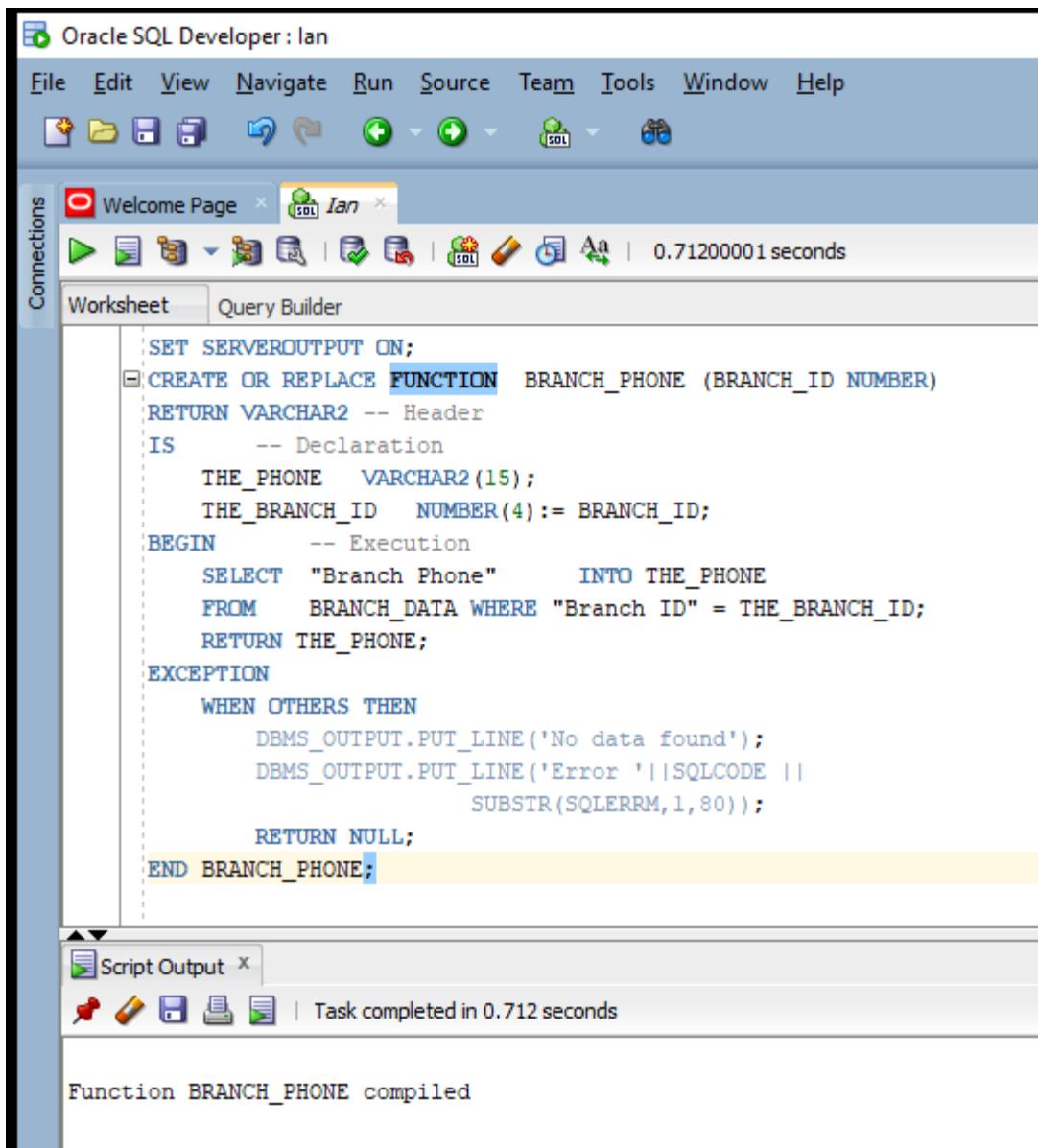
Branch Address
Braddock Rd Frostburg Maryland 21532

Branch Address
Marisa Ct Nottingham Maryland 21236

Branch Address
Jarrettsville Rd Jarrettsville Maryland 21084
```

The status bar at the bottom of the interface indicates "Task completed in 2.787 seconds".

## Create function.

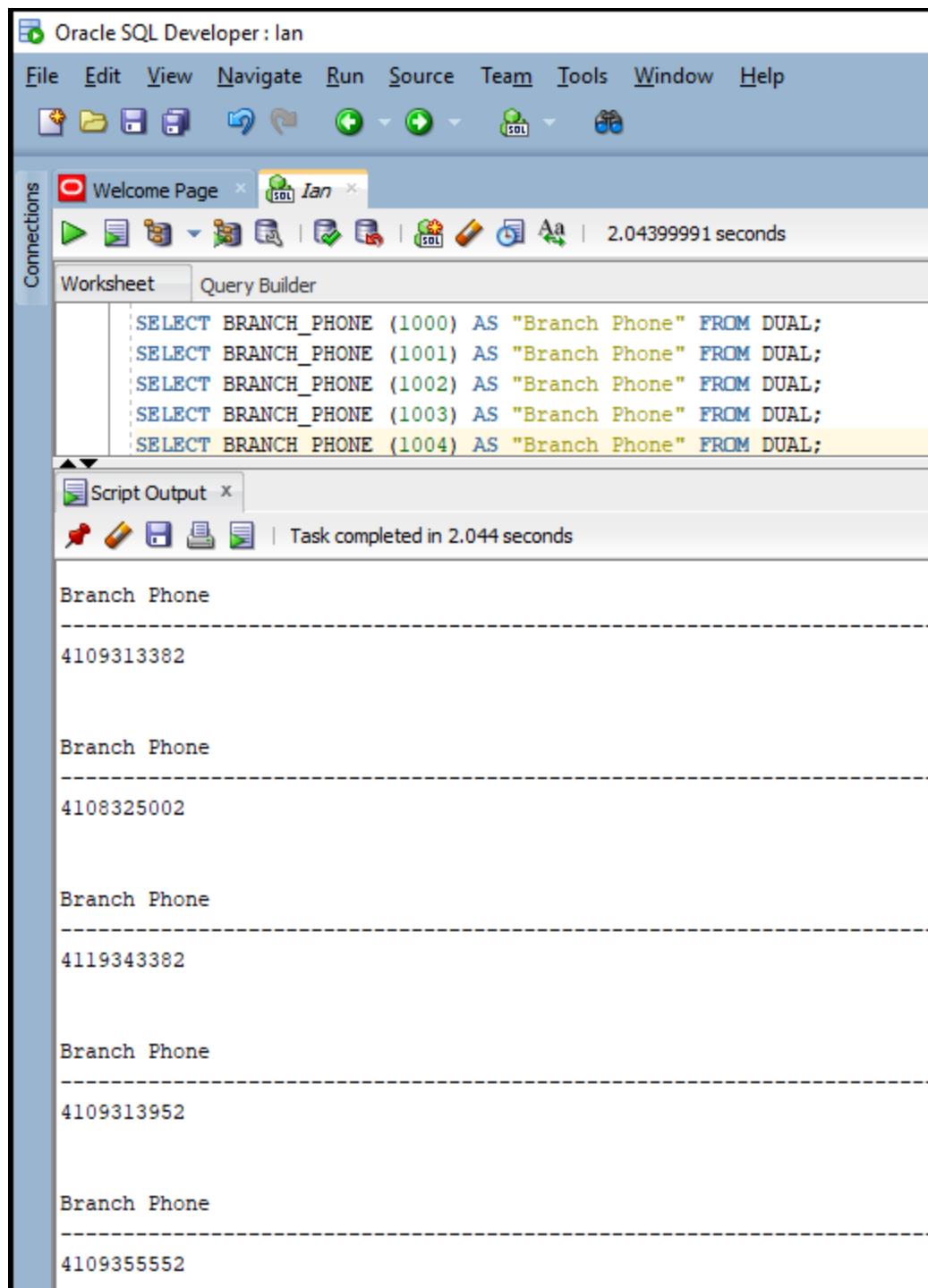


The screenshot shows the Oracle SQL Developer interface with a connection named 'lan'. The 'Worksheet' tab is active, displaying the SQL code for creating a function. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION BRANCH_PHONE (BRANCH_ID NUMBER)
  RETURN VARCHAR2 -- Header
IS
  -- Declaration
  THE_PHONE  VARCHAR2(15);
  THE_BRANCH_ID  NUMBER(4):= BRANCH_ID;
BEGIN
  -- Execution
  SELECT "Branch Phone"      INTO THE_PHONE
  FROM   BRANCH_DATA WHERE "Branch ID" = THE_BRANCH_ID;
  RETURN THE_PHONE;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                         SUBSTR(SQLERRM,1,80));
  RETURN NULL;
END BRANCH_PHONE;
```

The 'Script Output' tab at the bottom shows the message: 'Function BRANCH\_PHONE compiled'.

## Execute function.



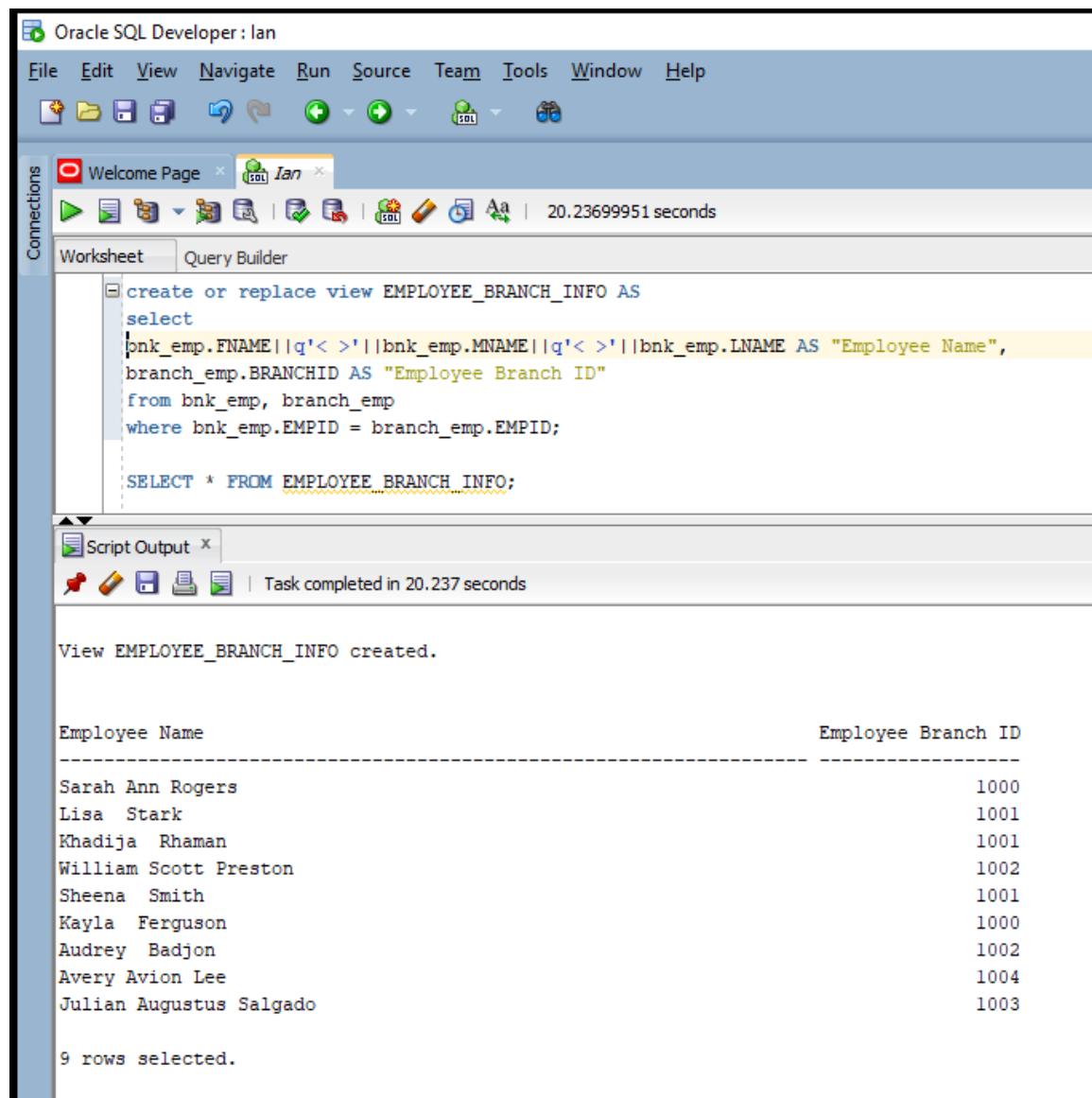
The screenshot shows the Oracle SQL Developer interface with a query window displaying a series of SELECT statements and their results. The query window has tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' selected. The query itself is as follows:

```
SELECT BRANCH_PHONE (1000) AS "Branch Phone" FROM DUAL;
SELECT BRANCH_PHONE (1001) AS "Branch Phone" FROM DUAL;
SELECT BRANCH_PHONE (1002) AS "Branch Phone" FROM DUAL;
SELECT BRANCH_PHONE (1003) AS "Branch Phone" FROM DUAL;
SELECT BRANCH_PHONE (1004) AS "Branch Phone" FROM DUAL;
```

The results are displayed in the 'Script Output' window, showing five rows of data, each consisting of the label 'Branch Phone' followed by a dashed line and the corresponding phone number. The results are as follows:

Branch Phone
4109313382
4108325002
4119343382
4109313952
4109355552

## Create/modify view



```
create or replace view EMPLOYEE_BRANCH_INFO AS
select
bnk_emp.FNAME||q'< >'||bnk_emp.MNAME||q'< >'||bnk_emp.LNAME AS "Employee Name",
branch_emp.BRANCHID AS "Employee Branch ID"
from bnk_emp, branch_emp
where bnk_emp.EMPID = branch_emp.EMPID;

SELECT * FROM EMPLOYEE_BRANCH_INFO;
```

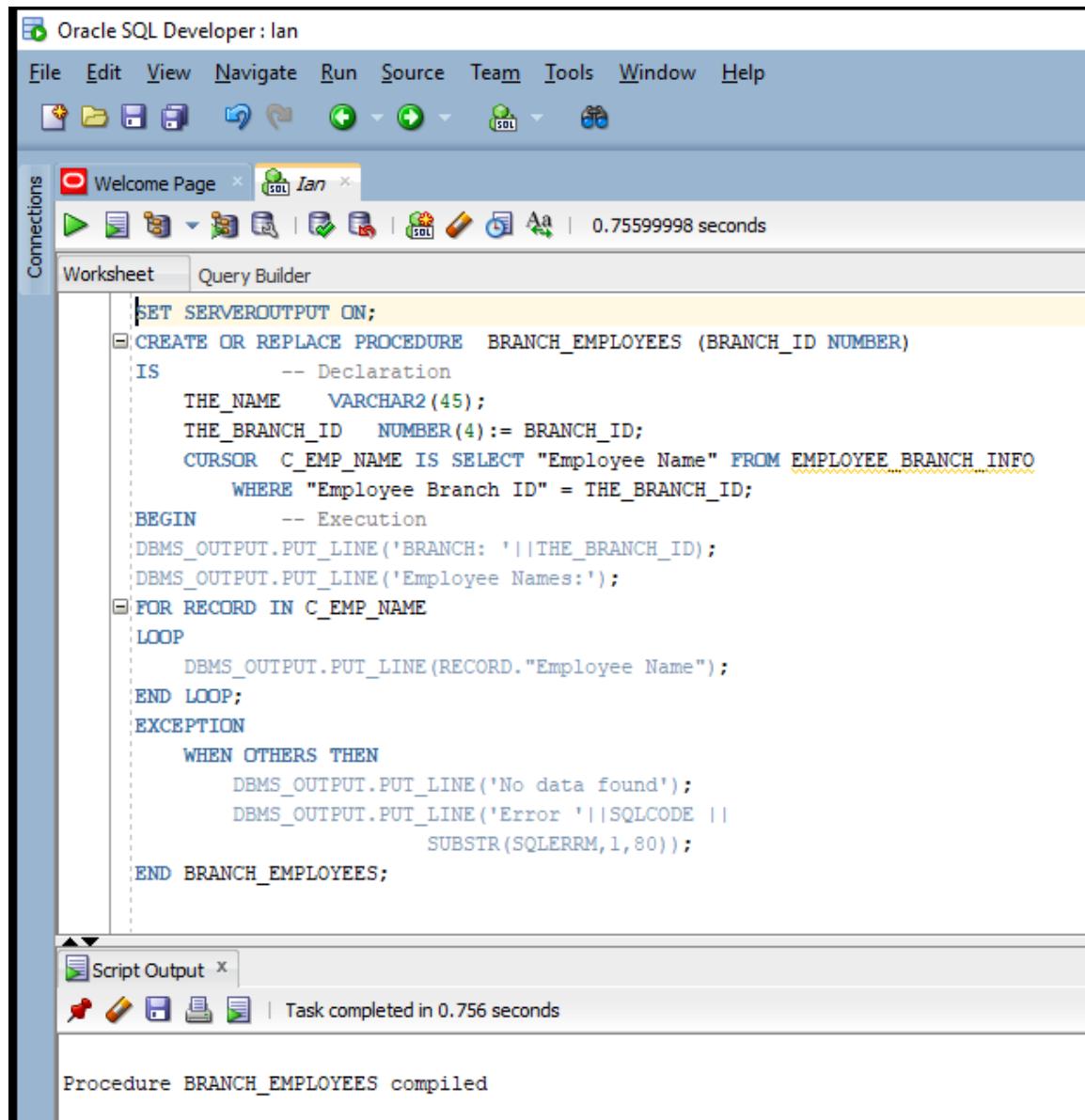
Script Output | Task completed in 20.237 seconds

View EMPLOYEE\_BRANCH\_INFO created.

Employee Name	Employee Branch ID
Sarah Ann Rogers	1000
Lisa Stark	1001
Khadija Rhaman	1001
William Scott Preston	1002
Sheena Smith	1001
Kayla Ferguson	1000
Audrey Badjon	1002
Avery Avion Lee	1004
Julian Augustus Salgado	1003

9 rows selected.

## Create procedure.

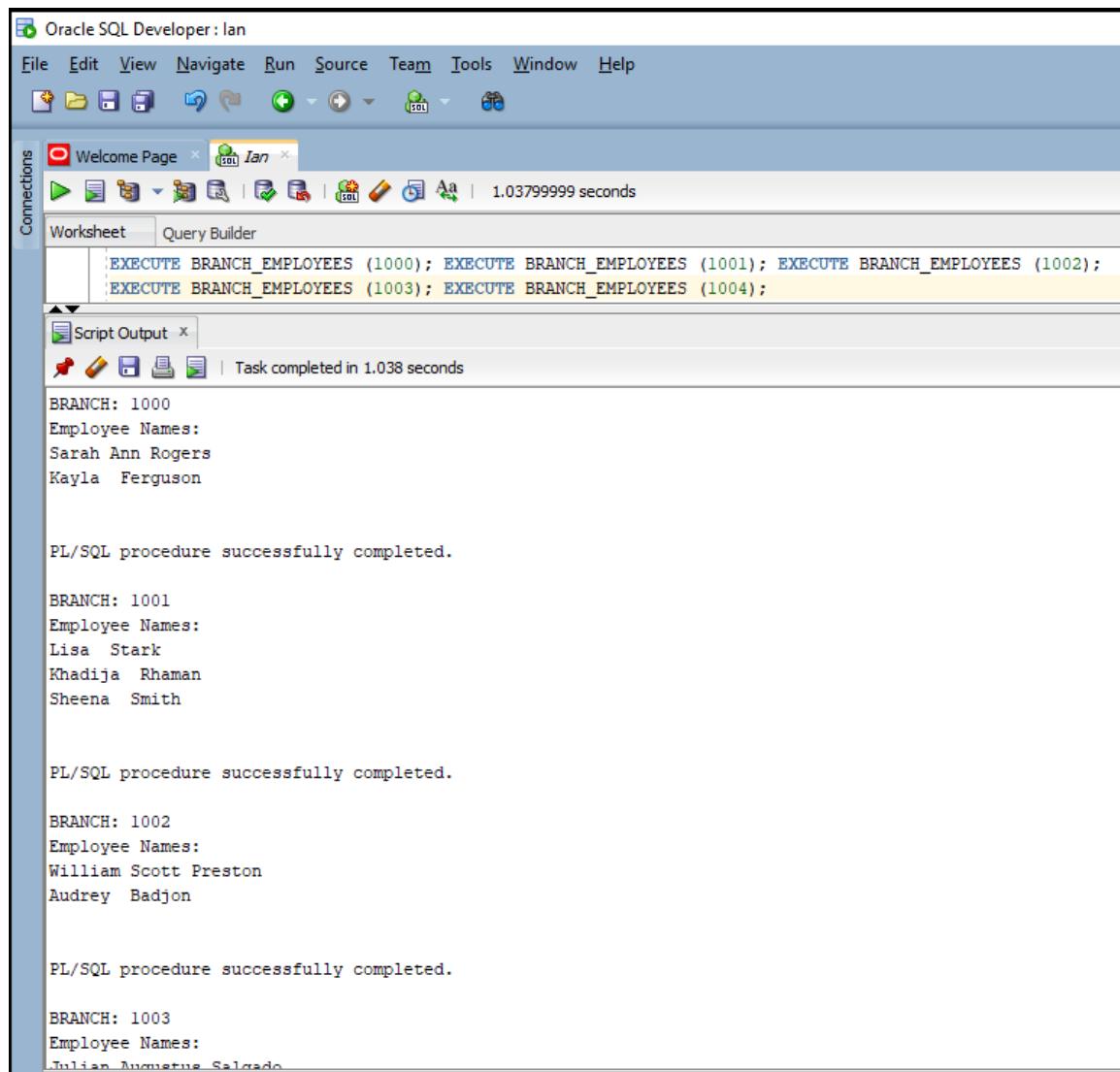


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE BRANCH_EMPLOYEES (BRANCH_ID NUMBER)
IS
    -- Declaration
    THE_NAME      VARCHAR2(45);
    THE_BRANCH_ID  NUMBER(4):= BRANCH_ID;
    CURSOR C_EMP_NAME IS SELECT "Employee Name" FROM EMPLOYEE_BRANCH_INFO
                         WHERE "Employee Branch ID" = THE_BRANCH_ID;
BEGIN
    -- Execution
    DBMS_OUTPUT.PUT_LINE('BRANCH: '||THE_BRANCH_ID);
    DBMS_OUTPUT.PUT_LINE('Employee Names:');
    FOR RECORD IN C_EMP_NAME
    LOOP
        DBMS_OUTPUT.PUT_LINE(RECORD."Employee Name");
    END LOOP;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('No data found');
            DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                               SUBSTR(SQLERRM,1,80));
    END BRANCH_EMPLOYEES;
```

The 'Script Output' tab at the bottom shows the message: "Procedure BRANCH\_EMPLOYEES compiled".

## Execute procedure.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. In the 'Worksheet' tab, the following PL/SQL code is executed:

```
EXECUTE BRANCH_EMPLOYEES (1000); EXECUTE BRANCH_EMPLOYEES (1001); EXECUTE BRANCH_EMPLOYEES (1002);
EXECUTE BRANCH_EMPLOYEES (1003); EXECUTE BRANCH_EMPLOYEES (1004);
```

The 'Script Output' tab displays the results of the procedure execution:

```
BRANCH: 1000
Employee Names:
Sarah Ann Rogers
Kayla Ferguson

PL/SQL procedure successfully completed.

BRANCH: 1001
Employee Names:
Lisa Stark
Khadija Rhaman
Sheena Smith

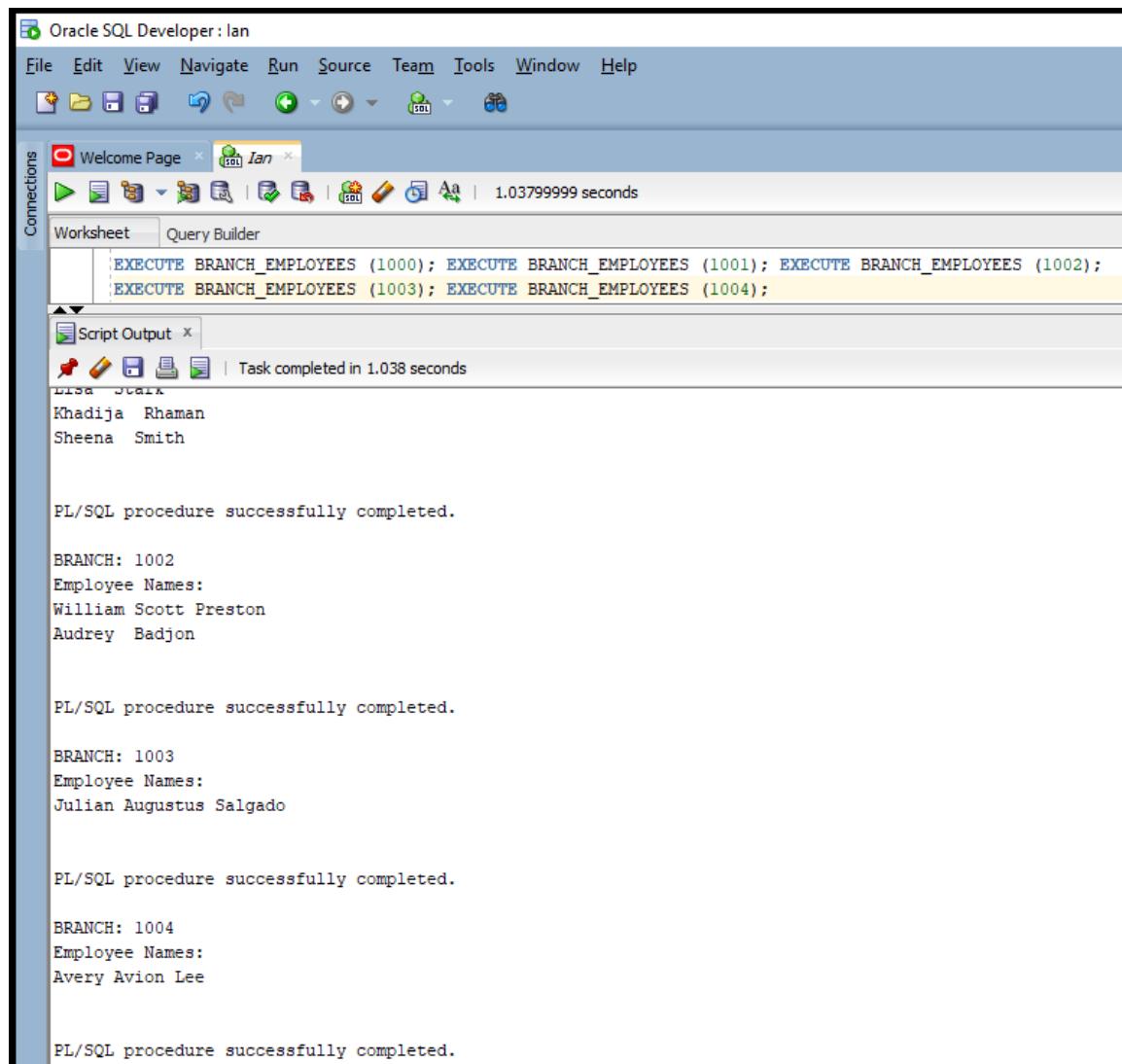
PL/SQL procedure successfully completed.

BRANCH: 1002
Employee Names:
William Scott Preston
Audrey Badjon

PL/SQL procedure successfully completed.

BRANCH: 1003
Employee Names:
Julian Augustus Salcedo
```

## Execute procedure.



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for connection, schema browser, and code editor. The Connections panel shows a connection named "Jan". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet contains the following PL/SQL code:

```
EXECUTE BRANCH_EMPLOYEES (1000); EXECUTE BRANCH_EMPLOYEES (1001); EXECUTE BRANCH_EMPLOYEES (1002);
EXECUTE BRANCH_EMPLOYEES (1003); EXECUTE BRANCH_EMPLOYEES (1004);
```

Below the worksheet is a "Script Output" window showing the results of the execution. The output is as follows:

```
Lisa Stark
Khadija Rhaman
Sheena Smith

PL/SQL procedure successfully completed.

BRANCH: 1002
Employee Names:
William Scott Preston
Audrey Badjon

PL/SQL procedure successfully completed.

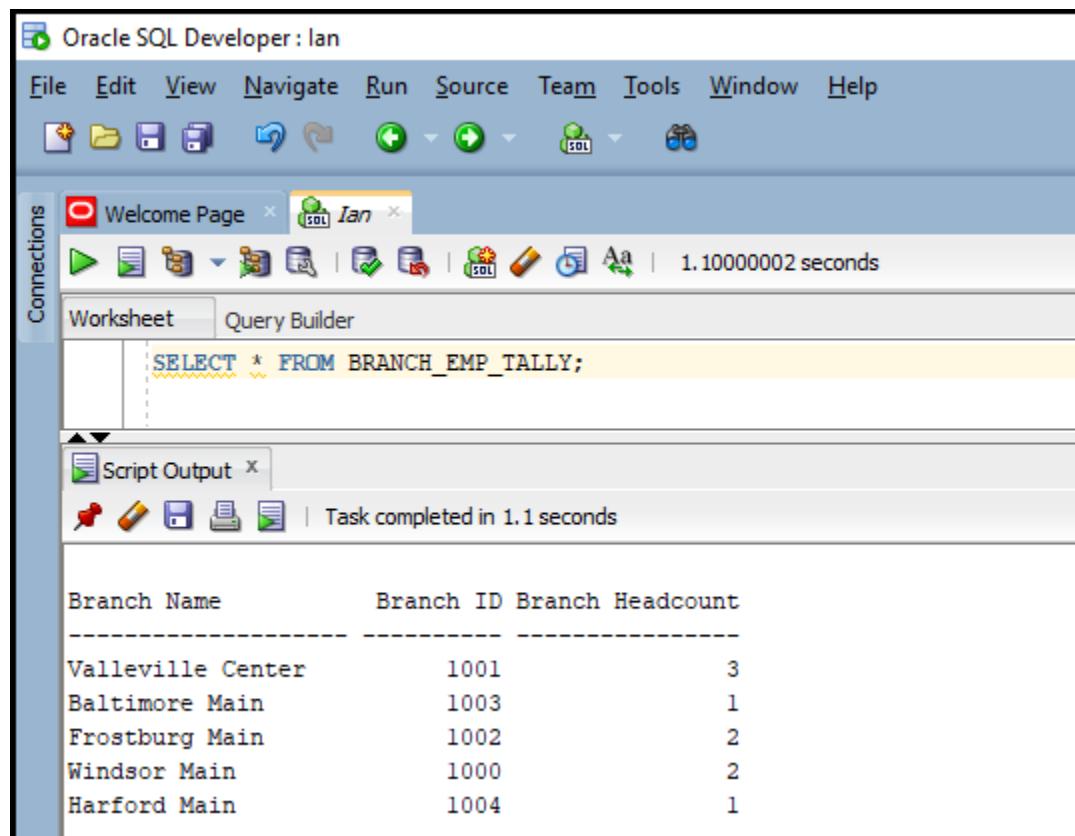
BRANCH: 1003
Employee Names:
Julian Augustus Salgado

PL/SQL procedure successfully completed.

BRANCH: 1004
Employee Names:
Avery Avion Lee

PL/SQL procedure successfully completed.
```

## Check table contents

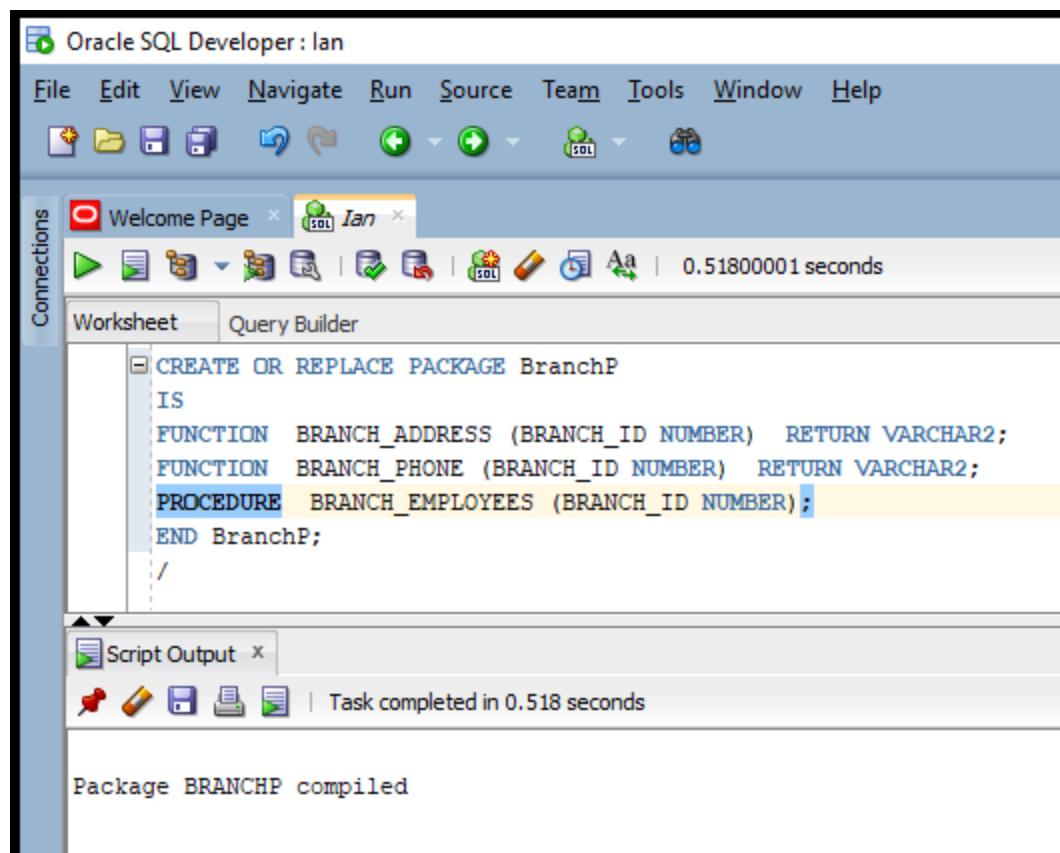


The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for connection management and query execution. The Connections panel shows a connection named "Ian". The Worksheet tab is active, displaying the SQL query: "SELECT \* FROM BRANCH\_EMP\_TALLY;". The Script Output tab shows the execution results:

Branch Name	Branch ID	Branch Headcount
Valleeville Center	1001	3
Baltimore Main	1003	1
Frostburg Main	1002	2
Windsor Main	1000	2
Harford Main	1004	1

The output indicates the task completed in 1.1 seconds.

## Create package of functions and procedure



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL tasks. The Connections sidebar shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet area contains the following PL/SQL code:

```
CREATE OR REPLACE PACKAGE BranchP
IS
  FUNCTION  BRANCH_ADDRESS (BRANCH_ID NUMBER) RETURN VARCHAR2;
  FUNCTION  BRANCH_PHONE (BRANCH_ID NUMBER) RETURN VARCHAR2;
  PROCEDURE  BRANCH_EMPLOYEES (BRANCH_ID NUMBER);
END BranchP;
/
```

The "Script Output" tab at the bottom shows the message "Package BRANCHP compiled" and "Task completed in 0.518 seconds".

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
CREATE OR REPLACE PACKAGE BODY BranchP
IS
FUNCTION BRANCH_ADDRESS (BRANCH_ID NUMBER)
RETURN VARCHAR2 -- Header
IS
-- Declaration
THE_ADDRESS VARCHAR2(45);
THE_BRANCH_ID NUMBER(4):= BRANCH_ID;
BEGIN
-- Execution
SELECT "Branch Address" INTO THE_ADDRESS
FROM BRANCH_DATA WHERE "Branch ID" = THE_BRANCH_ID;
RETURN THE_ADDRESS;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('No data found');
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
SUBSTR(SQLERRM,1,80));
RETURN NULL;
END BRANCH_ADDRESS;
-----
FUNCTION BRANCH_PHONE (BRANCH_ID NUMBER)
RETURN VARCHAR2 -- Header
IS
-- Declaration
THE_PHONE VARCHAR2(15);
THE_BRANCH_ID NUMBER(4):= BRANCH_ID;
BEGIN
-- Execution
SELECT "Branch Phone" INTO THE_PHONE
FROM BRANCH_DATA WHERE "Branch ID" = THE_BRANCH_ID;
```

Script Output Task completed in 0.564 seconds

Package Body BRANCHP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
SUBSTR(SQLERRM,1,80)) ;  
    RETURN NULL;  
END BRANCH_ADDRESS;  
---  
FUNCTION BRANCH_PHONE (BRANCH_ID NUMBER)  
RETURN VARCHAR2 -- Header  
IS  
    -- Declaration  
    THE_PHONE    VARCHAR2(15);  
    THE_BRANCH_ID NUMBER(4):= BRANCH_ID;  
BEGIN  
    -- Execution  
    SELECT "Branch Phone"      INTO THE_PHONE  
    FROM   BRANCH_DATA WHERE "Branch ID" = THE_BRANCH_ID;  
    RETURN THE_PHONE;  
EXCEPTION  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('No data found');  
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE||  
                           SUBSTR(SQLERRM,1,80));  
        RETURN NULL;  
END BRANCH_PHONE;  
---  
PROCEDURE BRANCH_EMPLOYEES (BRANCH_ID NUMBER)  
IS  
    -- Declaration  
    THE_NAME    VARCHAR2(45);  
    THE_BRANCH_ID NUMBER(4):= BRANCH_ID;  
    CURSOR C_EMP_NAME IS SELECT "Employee Name" FROM EMPLOYEE_BRANCH_INFO  
                      WHERE "Employee Branch ID" = THE_BRANCH_ID;
```

Script Output Task completed in 0.564 seconds

Package Body BRANCHP compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.56400001 seconds

Worksheet Query Builder

```
DBMS_OUTPUT.PUT_LINE('No data found');
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                     SUBSTR(SQLERRM,1,80));
RETURN NULL;
END BRANCH_PHONE;
-----
PROCEDURE BRANCH_EMPLOYEES (BRANCH_ID NUMBER)
IS
    -- Declaration
    THE_NAME      VARCHAR2(45);
    THE_BRANCH_ID  NUMBER(4):= BRANCH_ID;
    CURSOR C_EMP_NAME IS SELECT "Employee Name" FROM EMPLOYEE_BRANCH_INFO
        WHERE "Employee Branch ID" = THE_BRANCH_ID;
    BEGIN
        -- Execution
        DBMS_OUTPUT.PUT_LINE('BRANCH: '||THE_BRANCH_ID);
        DBMS_OUTPUT.PUT_LINE('Employee Names:');
    FOR RECORD IN C_EMP_NAME
    LOOP
        DBMS_OUTPUT.PUT_LINE(RECORD."Employee Name");
    END LOOP;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('No data found');
            DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                     SUBSTR(SQLERRM,1,80));
    END BRANCH_EMPLOYEES;
END BranchP;
/
```

Script Output Task completed in 0.564 seconds

Package Body BRANCHP compiled

^^^^^^^^^^^^^^^^^^^^^^^^

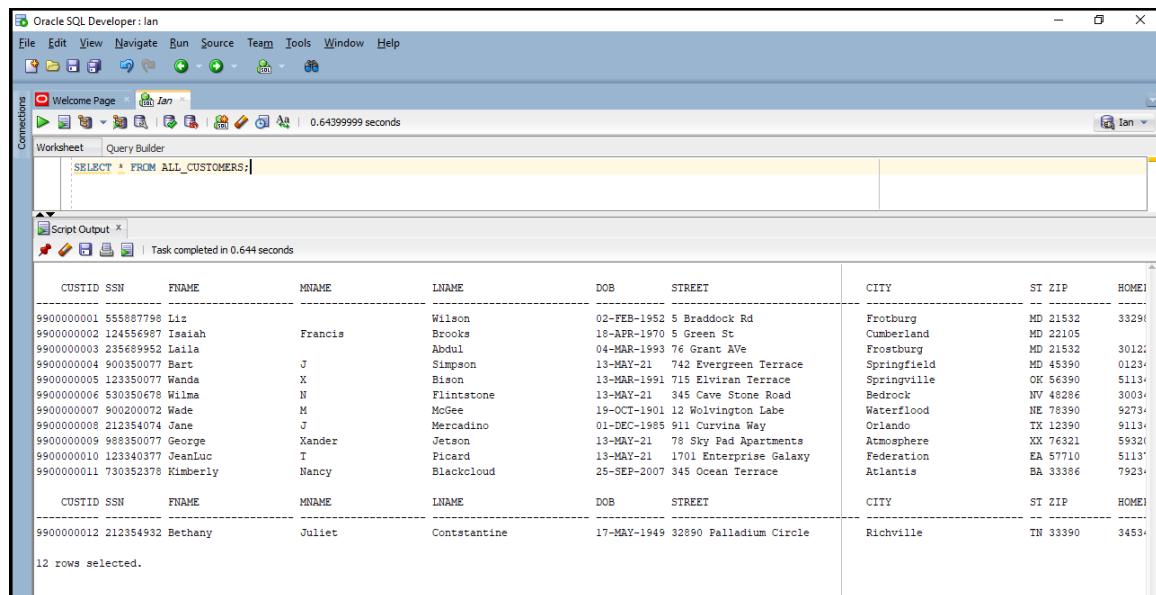
## CHAPTER 4C Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

C. Create a package called **Insert\_pkg** with subprograms to automate the insert, delete, and update of data in your database.

a--Subprogram to insert a row in table customer (make sure you use the sequence for generating ID's)

### Check table contents.

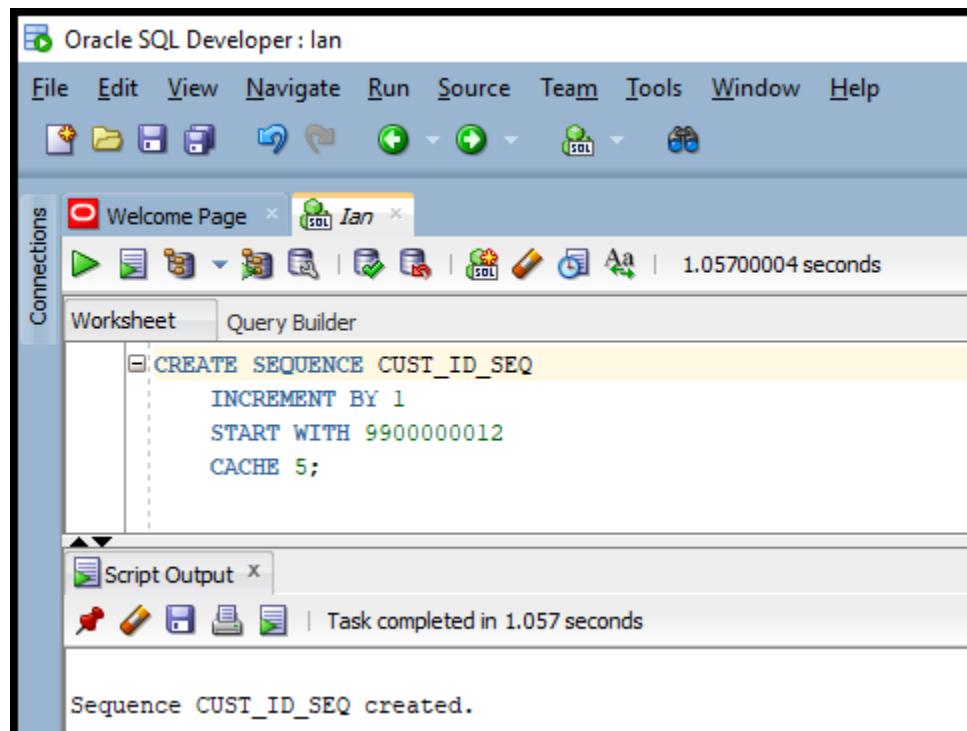


The screenshot shows the Oracle SQL Developer interface with a query window displaying the results of a SELECT query on the ALL\_CUSTOMERS table. The results are presented in two tables, each with 12 rows of data. The columns for the first table are CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, ST, ZIP, and HOME. The columns for the second table are the same. The data includes various names like Liz, Isaiah, Francis, Abdul, Simpson, Bison, Flintstone, McGe, Mercadino, Xander, Picard, and Blackcloud, along with their respective addresses, cities, states, and zip codes. The query window also shows a note: 'Task completed in 0.644 seconds'.

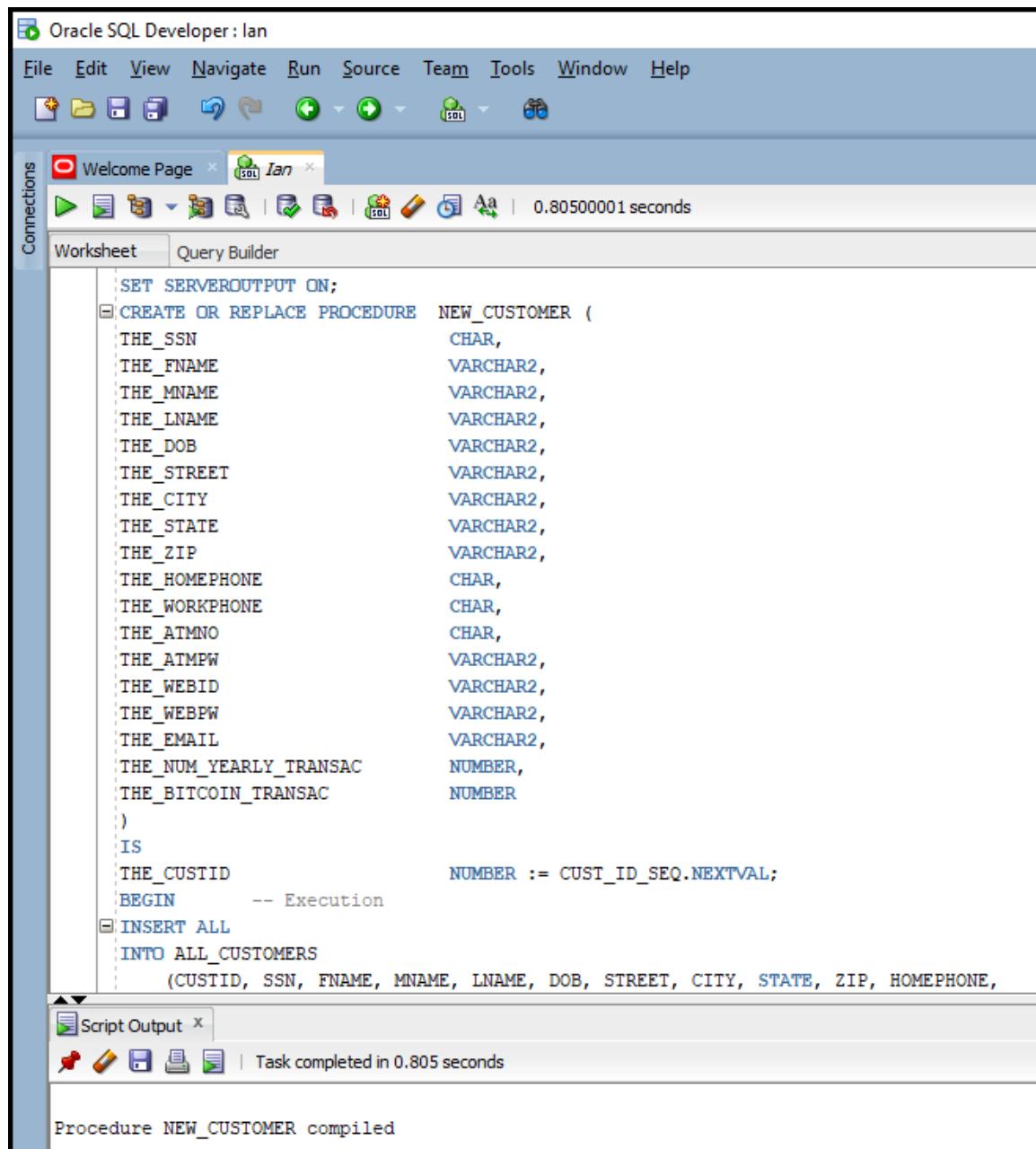
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD	21532	33294
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30124
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	48286	30034
9900000007	900200072	Wade	M	McGe	19-OCT-1901	12 Wolverton Lake	Waterflood	NV	78390	92734
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX	12390	91134
9900000009	9883550077	George		Xander	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	59324
9900000010	123340377	JeanLuc	T	Jetson	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	51134
9900000011	730352378	Kimberly	Nancy	Picard	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	79234
				Blackcloud						
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	34534

12 rows selected.

## Create Sequence



## Create procedure.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE NEW_CUSTOMER (
    THE_SSN           CHAR,
    THE_FNAME         VARCHAR2,
    THE_MNAME         VARCHAR2,
    THE_LNAME         VARCHAR2,
    THE_DOB           VARCHAR2,
    THE_STREET        VARCHAR2,
    THE_CITY          VARCHAR2,
    THE_STATE         VARCHAR2,
    THE_ZIP           VARCHAR2,
    THE_HOMEPHONE     CHAR,
    THE_WORKPHONE     CHAR,
    THE_ATMNO         CHAR,
    THE_ATMPW         VARCHAR2,
    THE_WEBID         VARCHAR2,
    THE_WEBPW         VARCHAR2,
    THE_EMAIL         VARCHAR2,
    THE_NUM_YEARLY_TRANSAC NUMBER,
    THE_BITCOIN_TRANSAC NUMBER
)
IS
    THE_CUSTID        NUMBER := CUST_ID_SEQ.NEXTVAL;
BEGIN
    -- Execution
    INSERT ALL
    INTO ALL_CUSTOMERS
        (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,

```

The code is partially cut off at the bottom. The 'Script Output' tab at the bottom shows the message: 'Procedure NEW\_CUSTOMER compiled'.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan

Worksheet Query Builder

```
THE_HOMEPHONE      CHAR,
THE_WORKPHONE      CHAR,
THE_ATMNO          CHAR,
THE_ATMPW          VARCHAR2,
THE_WEBID          VARCHAR2,
THE_WEBPW          VARCHAR2,
THE_EMAIL          VARCHAR2,
THE_NUM_YEARLY_TRANSAC NUMBER,
THE_BITCOIN_TRANSAC NUMBER
)
IS
THE_CUSTID          NUMBER := CUST_ID_SEQ.NEXTVAL;
BEGIN
-- Execution

INSERT ALL
INTO ALL CUSTOMERS
(CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES (THE_CUSTID, THE_SSN, THE_FNAME, THE_MNAME, THE_LNAME, THE_DOB, THE_STREET, THE_CITY, THE_STATE, THE_ZIP, THE_HOMEPHONE,
THE_WORKPHONE, THE_ATMNO, THE_ATMPW, THE_WEBID, THE_WEBPW, THE_EMAIL, THE_NUM_YEARLY_TRANSAC, THE_BITCOIN_TRANSAC)
SELECT * FROM dual;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('No data found');
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
SUBSTR(SQLERRM,1,80));
END NEW_CUSTOMER;
/



```

Script Output | Task completed in 0.805 seconds

Procedure NEW\_CUSTOMER compiled

**Check table contents.  
Execute procedure.  
Recheck table contents.**

```

SELECT * FROM ALL_CUSTOMERS;
EXECUTE NEW_CUSTOMER ('900355577', 'Somebody', 'J', 'A_person', '23-DEC-1999', '742 Someplace Terrace', 'Somewhereville', 'MD', '45260', '0123236734', '0136758289';
SELECT * FROM ALL_CUSTOMERS;

```

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frotnburg	MD	21532	33294
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila	Abdul		04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30124
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	48286	30034
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Lake	Waterflood	NE	78390	92734
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX	12390	91134
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	59324
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	51134
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	79234
<hr/>										
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	34534
12 rows selected.										
PL/SQL procedure successfully completed.										

```

SELECT * FROM ALL_CUSTOMERS;
EXECUTE NEW_CUSTOMER ('900355577', 'Somebody', 'J', 'A_person', '23-DEC-1999', '742 Someplace Terrace', 'Somewhereville', 'MD', '45260', '0123236734', '0136758289';
SELECT * FROM ALL_CUSTOMERS;

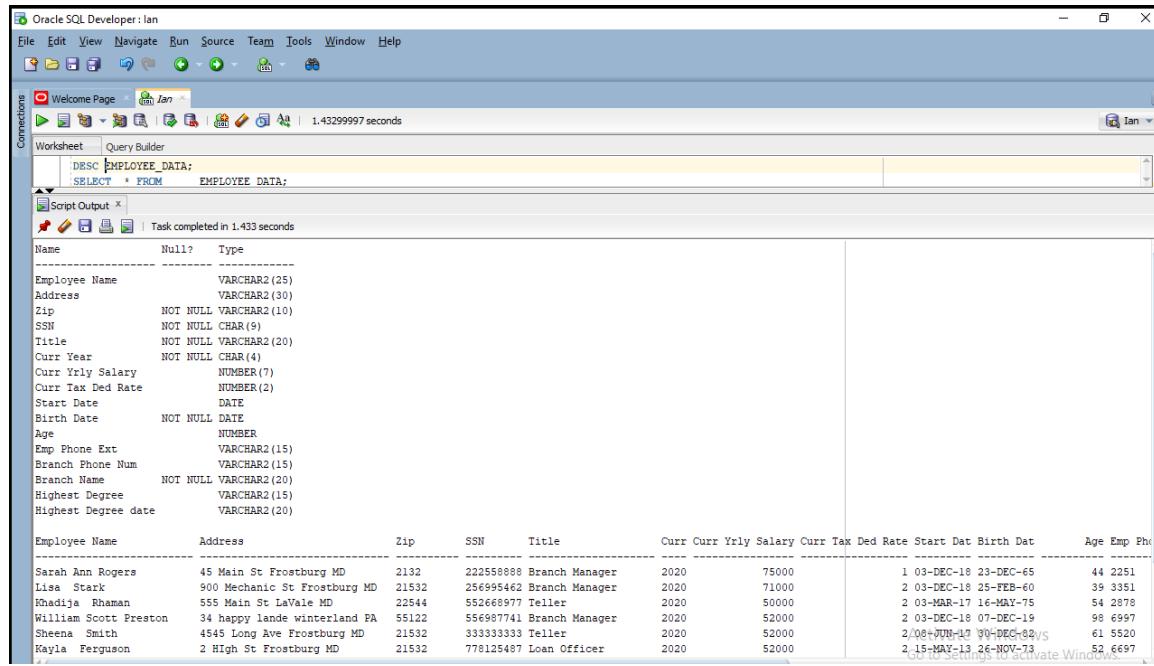
```

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frotnburg	MD	21532	33294
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila	Abdul		04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30124
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	48286	30034
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Lake	Waterflood	NE	78390	92734
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX	12390	91134
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	59324
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	51134
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	79234
<hr/>										
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	34534
9900000013	900355577	Somebody	J	A_person	23-DEC-1999	742 Someplace Terrace	Somewhereville	MD	45260	01234
13 rows selected.										
PL/SQL procedure successfully completed.										

b-Subprogram to insert a row in table bank employee

**Check table contents.**

**Check table attribute data types**



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

DESC EMPLOYEE\_DATA;  
SELECT \* FROM EMPLOYEE DATA;

Script Output | Task completed in 1.433 seconds

Name	Null?	Type
Employee Name		VARCHAR2(25)
Address		VARCHAR2(30)
Zip	NOT NULL	VARCHAR2(10)
SSN	NOT NULL	CHAR(9)
Title	NOT NULL	VARCHAR2(20)
Curr Year	NOT NULL	CHAR(4)
Curr Yrly Salary		NUMBER(7)
Curr Tax Ded Rate		NUMBER(2)
Start Date		DATE
Birth Date	NOT NULL	DATE
Age		NUMBER
Emp Phone Ext		VARCHAR2(15)
Branch Phone Num		VARCHAR2(15)
Branch Name	NOT NULL	VARCHAR2(20)
Highest Degree		VARCHAR2(15)
Highest Degree date		VARCHAR2(20)

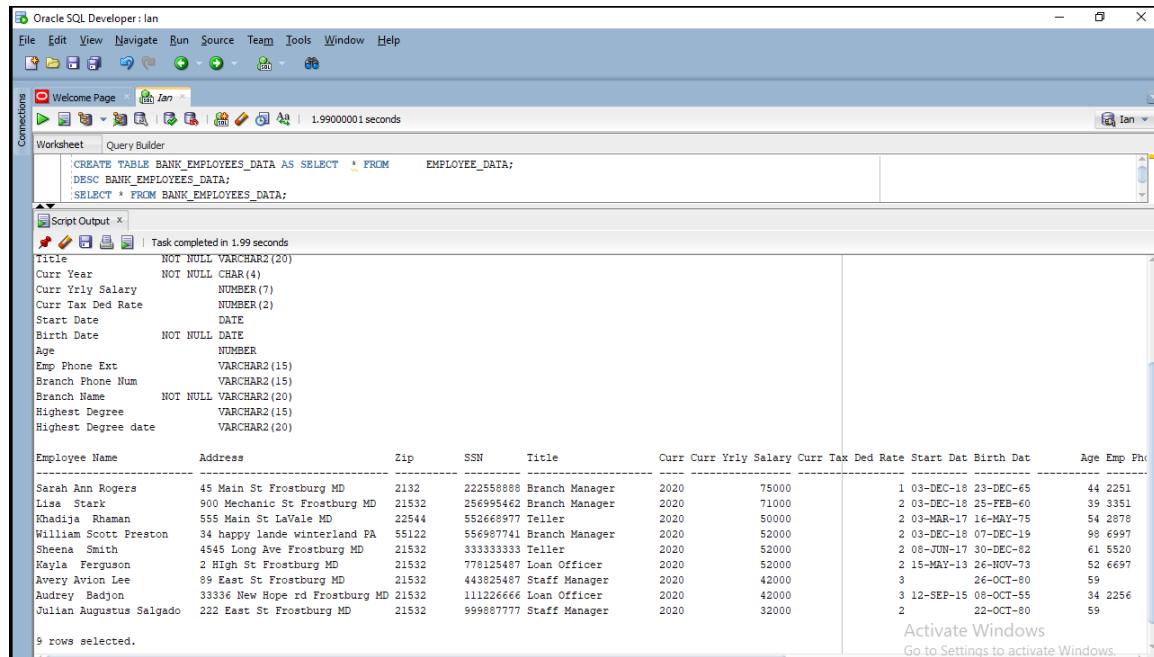
  

Employee Name	Address	Zip	SSN	Title	Curr Yrly Salary	Curr Tax Ded Rate	Start Date	Birth Date	Age	Emp Photo
Sarah Ann Rogers	45 Main St Frostburg MD	21532	222558888	Branch Manager	2020	75000	1 03-DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	25695462	Branch Manager	2020	71000	2 03-DEC-18	25-FEB-60	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000	2 03-MAR-17	16-MAY-75	54	2878
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020	52000	2 03-DEC-18	07-DEC-19	98	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000	2 08-JUN-17	30-DEC-82	61	5520
Mayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000	2 15-MAY-13	26-NOV-73	52	6697

**Create new table.**

**Check new table attribute data types.**

**Check new table contents.**



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

CREATE TABLE BANK\_EMPLOYEES\_DATA AS SELECT \* FROM EMPLOYEE\_DATA;  
DESC BANK\_EMPLOYEES\_DATA;  
SELECT \* FROM BANK\_EMPLOYEES\_DATA;

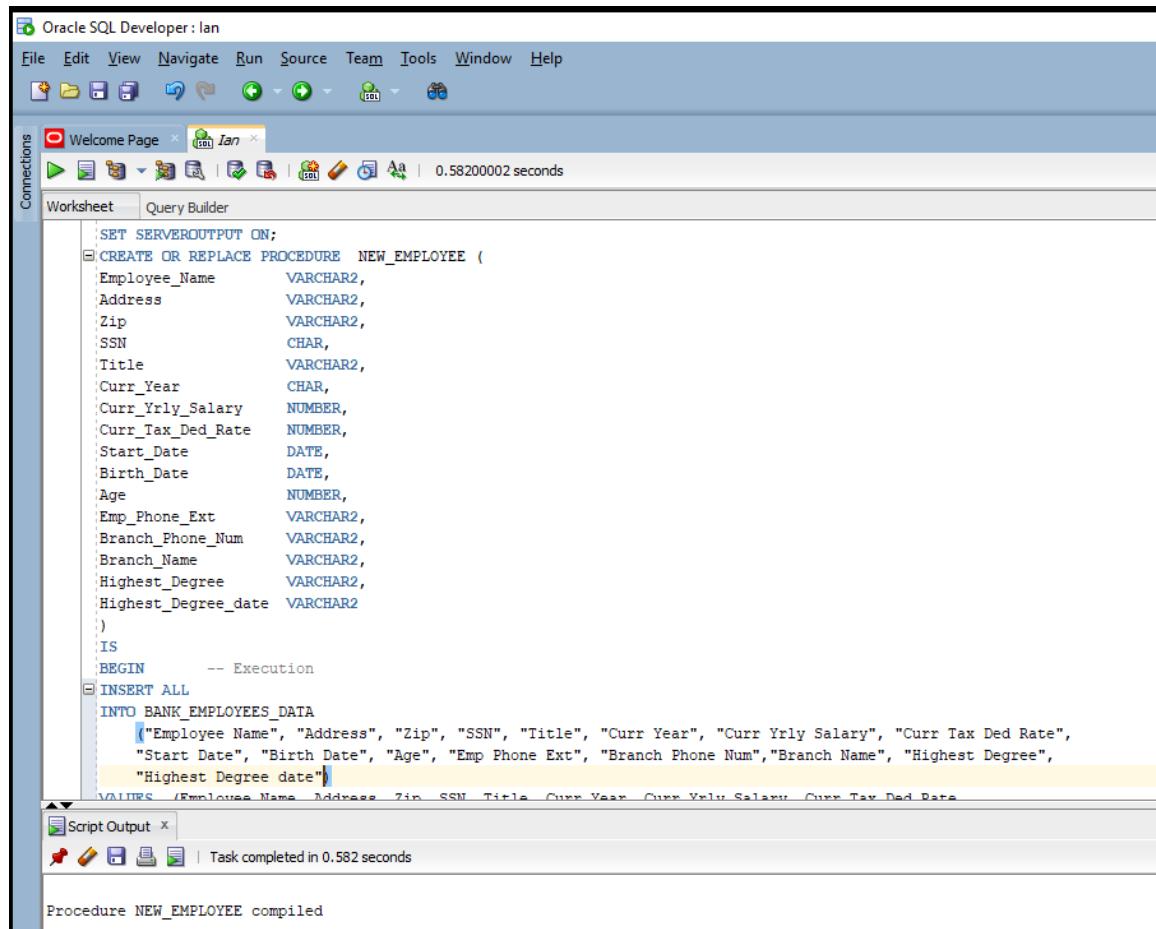
Script Output | Task completed in 1.99000001 seconds

Name	Null?	Type
Title	NOT NULL	VARCHAR2(20)
Curr Year	NOT NULL	CHAR(4)
Curr Yrly Salary		NUMBER(7)
Curr Tax Ded Rate		NUMBER(2)
Start Date		DATE
Birth Date	NOT NULL	DATE
Age		NUMBER
Emp Phone Ext		VARCHAR2(15)
Branch Phone Num		VARCHAR2(15)
Branch Name	NOT NULL	VARCHAR2(20)
Highest Degree		VARCHAR2(15)
Highest Degree date		VARCHAR2(20)

Employee Name	Address	Zip	SSN	Title	Curr Yrly Salary	Curr Tax Ded Rate	Start Date	Birth Date	Age	Emp Photo
Sarah Ann Rogers	45 Main St Frostburg MD	21532	222558888	Branch Manager	2020	75000	1 03-DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	25695462	Branch Manager	2020	71000	2 03-DEC-18	25-FEB-60	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000	2 03-MAR-17	16-MAY-75	54	2878
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020	52000	2 03-DEC-18	07-DEC-19	98	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000	2 08-JUN-17	30-DEC-82	61	5520
Mayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000	2 15-MAY-13	26-NOV-73	52	6697
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020	42000	3	26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020	42000	3 12-SEP-15	08-OCT-55	34	2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020	32000	2	22-OCT-80	59	

## Create procedure.



The screenshot shows the Oracle SQL Developer interface with a procedure creation script in the Worksheet tab. The script creates a procedure named NEW\_EMPLOYEE that inserts data into the BANK\_EMPLOYEES\_DATA table. The table has columns: Employee\_Name, Address, Zip, SSN, Title, Curr\_Year, Curr\_Yrly\_Salary, Curr\_Tax\_Ded\_Rate, Start\_Date, Birth\_Date, Age, Emp\_Phone\_Ext, Branch\_Phone\_Num, Branch\_Name, Highest\_Degree, and Highest\_Degree\_date. The script uses a dynamic SQL INSERT statement with a VALUES clause containing all these columns. The procedure is successfully compiled.

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE NEW_EMPLOYEE (
    Employee_Name      VARCHAR2,
    Address            VARCHAR2,
    Zip                VARCHAR2,
    SSN                CHAR,
    Title              VARCHAR2,
    Curr_Year          CHAR,
    Curr_Yrly_Salary   NUMBER,
    Curr_Tax_Ded_Rate  NUMBER,
    Start_Date         DATE,
    Birth_Date         DATE,
    Age                NUMBER,
    Emp_Phone_Ext     VARCHAR2,
    Branch_Phone_Num  VARCHAR2,
    Branch_Name        VARCHAR2,
    Highest_Degree     VARCHAR2,
    Highest_Degree_date VARCHAR2
)
IS
BEGIN      -- Execution
  INSERT ALL
  INTO BANK_EMPLOYEES_DATA
    ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
     "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
     "Highest Degree date")
    VALUES (Employee_Name, Address, Zip, SSN, Title, Curr_Year, Curr_Yrly_Salary, Curr_Tax_Ded_Rate,
            Start_Date, Birth_Date, Age, Emp_Phone_Ext, Branch_Phone_Num, Branch_Name, Highest_Degree,
            Highest_Degree_date);
END;
/
Procedure NEW_EMPLOYEE compiled
```

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 0.58200002 seconds

Worksheet Query Builder

```
CREATE OR REPLACE PROCEDURE NEW_EMPLOYEE AS
  Curr_Tax_Ded_Rate NUMBER,
  Start_Date DATE,
  Birth_Date DATE,
  Age NUMBER,
  Emp_Phone_Ext VARCHAR2,
  Branch_Phone_Num VARCHAR2,
  Branch_Name VARCHAR2,
  Highest_Degree VARCHAR2,
  Highest_Degree_date VARCHAR2
)
IS
BEGIN -- Execution
  INSERT ALL
  INTO BANK_EMPLOYEES_DATA
    ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
     "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
     "Highest Degree date")
    VALUES (Employee_Name, Address, Zip, SSN, Title, Curr_Year, Curr_Yrly_Salary, Curr_Tax_Ded_Rate,
            Start_Date, Birth_Date, Age, Emp_Phone_Ext, Branch_Phone_Num, Branch_Name, Highest_Degree, Highest_Degree_date)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE || SUBSTR(SQLERRM,1,80));
  END NEW_EMPLOYEE;
/

```

Script Output Task completed in 0.582 seconds

Procedure NEW\_EMPLOYEE compiled

**Check table contents.  
Execute procedure.  
Recheck table contents.**

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian | 1.35800004seconds

Worksheet Query Builder

```
SELECT * FROM BANK_EMPLOYEES_DATA;
EXECUTE NEW_EMPLOYEE ('Bugs Bunny', '1234 Hoppy Street MD', '23944', '888668977', 'Teller', '2019', 75000, 2, '07-MAR-15', '16-MAY-75',
54, '2338', '4108325002', 'Valleeville Center', 'Mast', '12-DEC-95');
SELECT * FROM BANK_EMPLOYEES_DATA;
```

Script Output X | Task completed in 1.358 seconds

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	21532	222558888	Branch Manager	2020 75000	1 03-DEC-18 23-DEC-65	44 2251	
Lisa Stark	900 Mechanic St Frostburg MD	21532	256959462	Branch Manager	2020 71000	2 03-DEC-18 25-FEB-60	39 3351	
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020 50000	2 03-MAR-17 16-MAY-75	54 2878	
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020 52000	2 03-DEC-18 07-DEC-19	98 6997	
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020 52000	2 08-JUN-17 30-DEC-82	61 5520	
Kayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020 52000	2 15-MAY-13 26-NOV-73	52 6697	
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020 42000	3 26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020 42000	3 12-SEP-15 08-OCT-55	34 2256	
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020 32000	2 22-OCT-80	59	

9 rows selected.

PL/SQL procedure successfully completed.

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	21532	222558888	Branch Manager	2020 75000	1 03-DEC-18 23-DEC-65	44 2251	
Lisa Stark	900 Mechanic St Frostburg MD	21532	256959462	Branch Manager	2020 71000	2 03-DEC-18 25-FEB-60	39 3351	
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020 50000	2 03-MAR-17 16-MAY-75	54 2878	
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020 52000	2 03-DEC-18 07-DEC-19	98 6997	
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020 52000	2 08-JUN-17 30-DEC-82	61 5520	

PL/SQL procedure successfully completed.

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020 42000	3 26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020 42000	3 12-SEP-15 08-OCT-55	34 2256	
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020 32000	2 22-OCT-80	59	

9 rows selected.

PL/SQL procedure successfully completed.

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	21532	222558888	Branch Manager	2020 75000	1 03-DEC-18 23-DEC-65	44 2251	
Lisa Stark	900 Mechanic St Frostburg MD	21532	256959462	Branch Manager	2020 71000	2 03-DEC-18 25-FEB-60	39 3351	
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020 50000	2 03-MAR-17 16-MAY-75	54 2878	
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020 52000	2 03-DEC-18 07-DEC-19	98 6997	
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020 52000	2 08-JUN-17 30-DEC-82	61 5520	
Kayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020 52000	2 15-MAY-13 26-NOV-73	52 6697	
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020 42000	3 26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020 42000	3 12-SEP-15 08-OCT-55	34 2256	
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020 32000	2 22-OCT-80	59	
Bugs Bunny	1234 Hoppy Street MD	23944	888668977	Teller	2019 75000	2 07-MAR-15 16-MAY-75	54 2338	

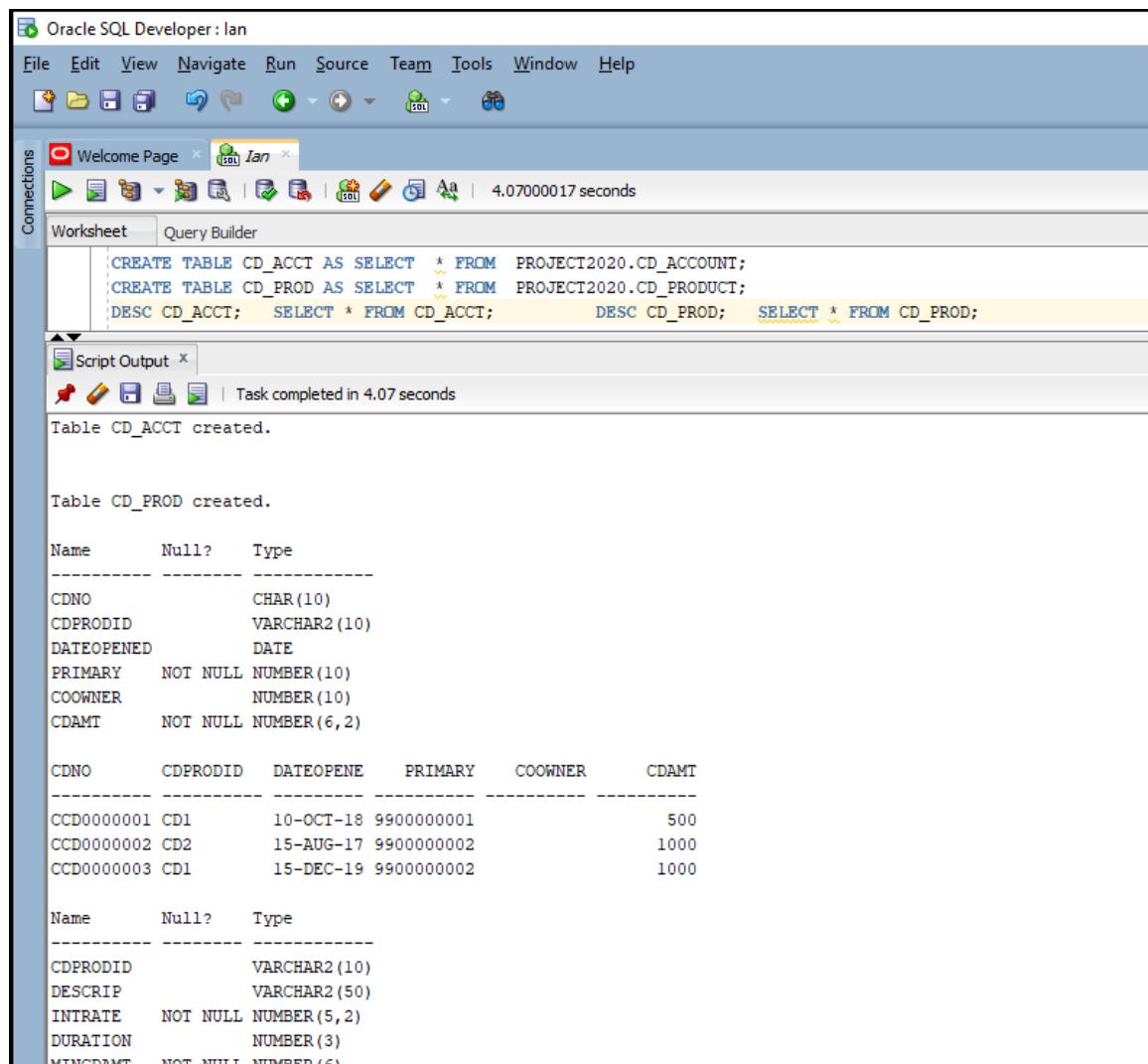
10 rows selected.

Activate Windows  
Go to Settings to activate Windows

c--Subprogram to insert a row in table CD

**Check two new tables.**

**Check new table attribute data types.**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page x Oracle SQL Developer : Ian x

Worksheet Query Builder

```
CREATE TABLE CD_ACCT AS SELECT * FROM PROJECT2020.CD_ACCOUNT;
CREATE TABLE CD_PROD AS SELECT * FROM PROJECT2020.CD_PRODUCT;
DESC CD_ACCT;   SELECT * FROM CD_ACCT;      DESC CD_PROD;   SELECT * FROM CD_PROD;
```

Script Output x

Table CD\_ACCT created.

Table CD\_PROD created.

Name	Null?	Type
CDNO		CHAR(10)
CDPRODID		VARCHAR2(10)
DATEOPENED		DATE
PRIMARY	NOT NULL	NUMBER(10)
COOWNER		NUMBER(10)
CDAMT	NOT NULL	NUMBER(6,2)

CDNO	CDPRODID	DATEOPENED	PRIMARY	COOWNER	CDAMT
CCD00000001	CD1	10-OCT-18	9900000001		500
CCD00000002	CD2	15-AUG-17	9900000002		1000
CCD00000003	CD1	15-DEC-19	9900000002		1000

Name	Null?	Type
CDPRODID		VARCHAR2(10)
DESCRIP		VARCHAR2(50)
INTRATE	NOT NULL	NUMBER(5,2)
DURATION		NUMBER(3)
MTMCDAMT	NOT NULL	NUMBER(14)

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan

Worksheet Query Builder

```
CREATE TABLE CD_ACCT AS SELECT * FROM PROJECT2020.CD_ACCOUNT;
CREATE TABLE CD_PROD AS SELECT * FROM PROJECT2020.CD_PRODUCT;
DESC CD_ACCT; SELECT * FROM CD_ACCT; DESC CD_PROD; SELECT * FROM CD_PROD;
```

Script Output x

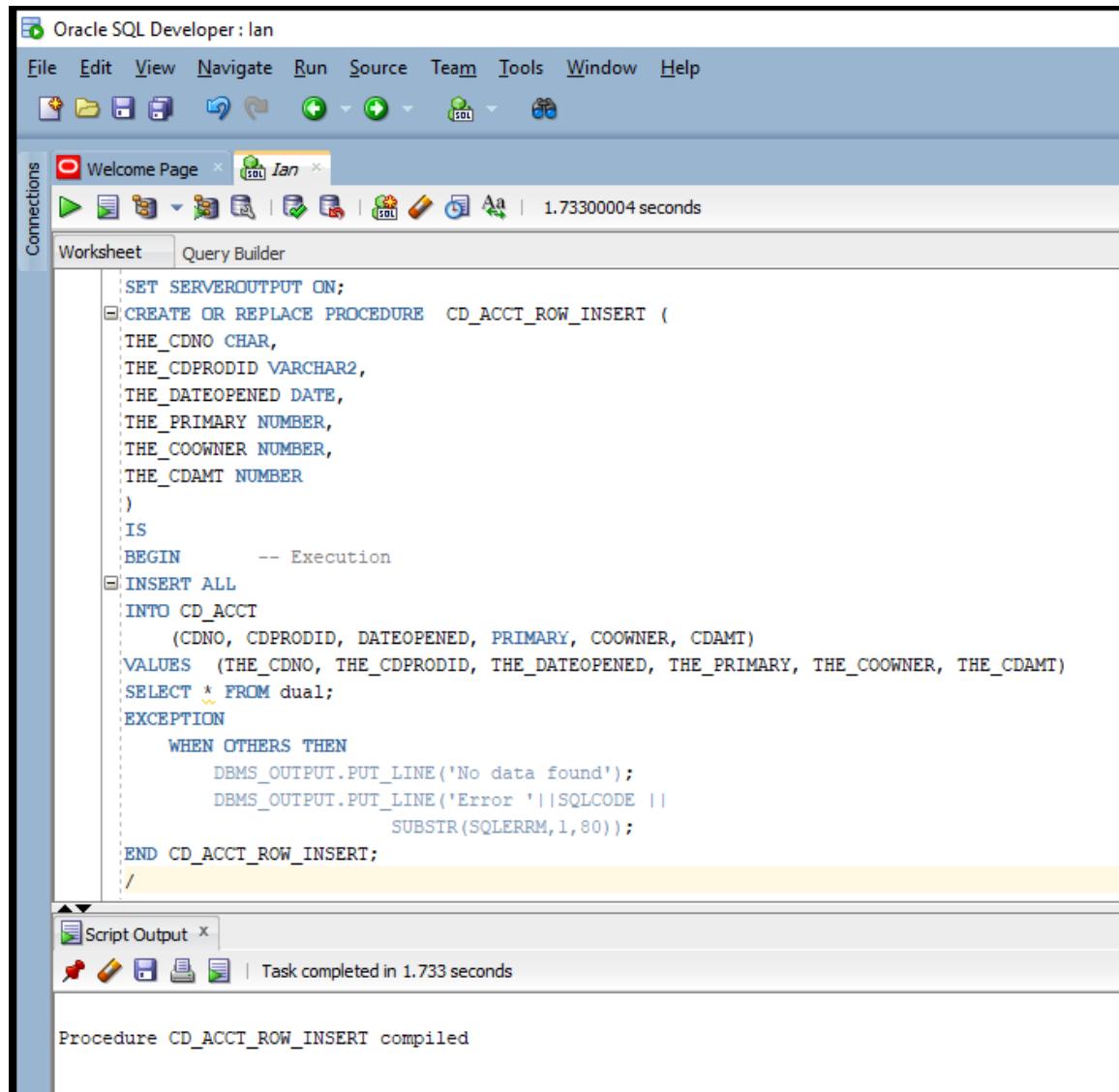
Task completed in 4.07 seconds

CDNO	CDPRODID	DATEOPENE	PRIMARY	COOWNER	CDAMT
CCD0000001	CD1	10-OCT-18	9900000001		500
CCD0000002	CD2	15-AUG-17	9900000002		1000
CCD0000003	CD1	15-DEC-19	9900000002		1000

Name	Null?	Type
CDPRODID		VARCHAR2(10)
DESCRIP		VARCHAR2(50)
INTRATE	NOT NULL	NUMBER(5,2)
DURATION		NUMBER(3)
MINCDAMT	NOT NULL	NUMBER(6)
PENALTYAMT		NUMBER(4,2)

CDPRODID	DESCRIP	INTRATE	DURATION	MINCDAMT	PENALTYAMT
CD1	9 month CD		2	9	500
CD2	5 year CD		10.25	60	1000
CD3	2 year CD		2.5	12	750

## Create procedure.

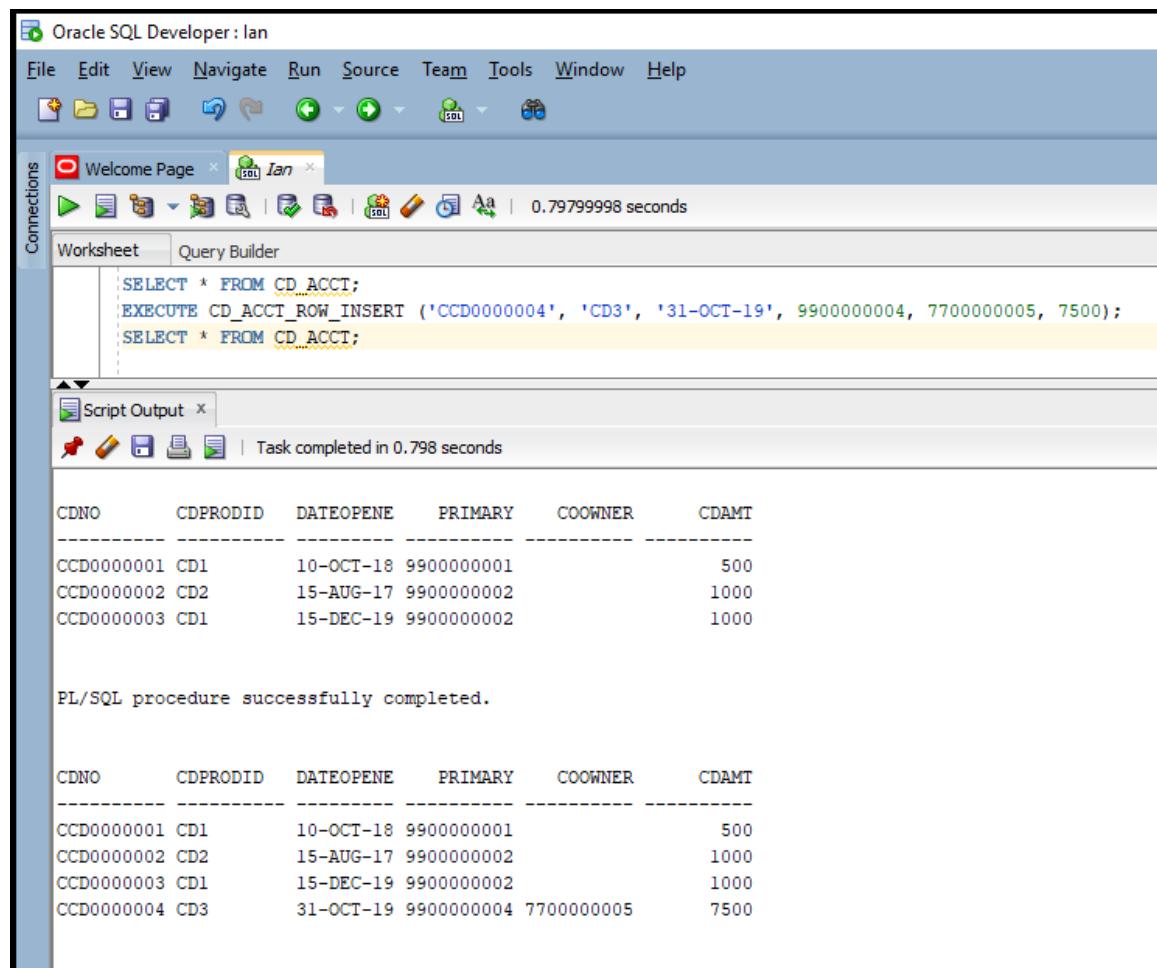


The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE CD_ACCT_ROW_INSERT (
  THE_CDNO CHAR,
  THE_CDPRODID VARCHAR2,
  THE_DATEOPENED DATE,
  THE_PRIMARY NUMBER,
  THE_COOWNER NUMBER,
  THE_CDAMT NUMBER
)
IS
BEGIN      -- Execution
  INSERT ALL
  INTO CD_ACCT
  (CDNO, CDPRODID, DATEOPENED, PRIMARY, COOWNER, CDAMT)
  VALUES (THE_CDNO, THE_CDPRODID, THE_DATEOPENED, THE_PRIMARY, THE_COOWNER, THE_CDAMT)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                         SUBSTR(SQLERRM,1,80));
END CD_ACCT_ROW_INSERT;
/
```

The 'Script Output' tab at the bottom shows the message: "Procedure CD\_ACCT\_ROW\_INSERT compiled".

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL functions. The Connections panel shows a connection named "Ian". The Worksheet tab is active, displaying the following SQL script:

```
SELECT * FROM CD_ACCT;
EXECUTE CD_ACCT_ROW_INSERT ('CCD0000004', 'CD3', '31-OCT-19', 9900000004, 7700000005, 7500);
SELECT * FROM CD_ACCT;
```

The Script Output tab shows the results of the execution:

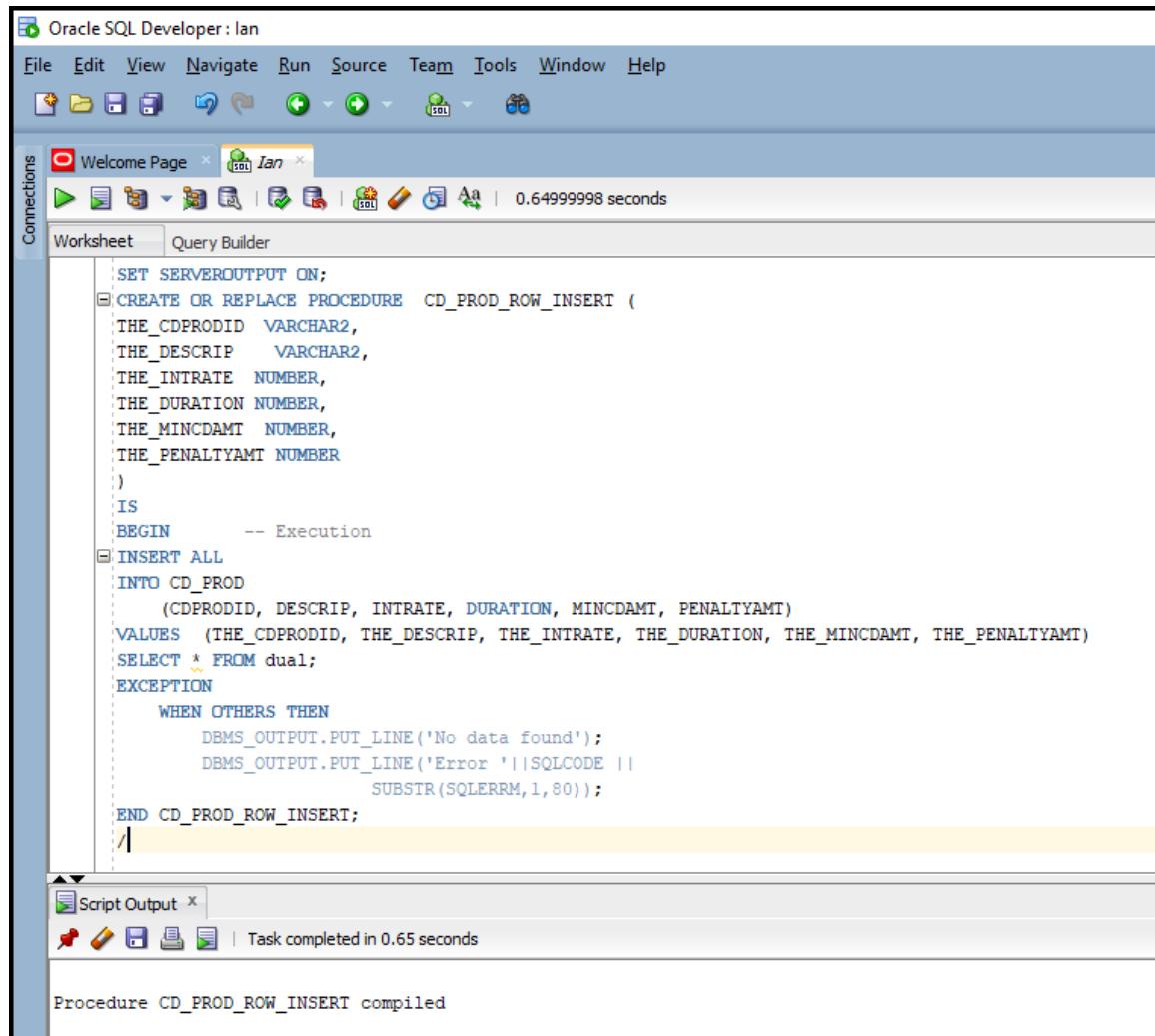
```
PL/SQL procedure successfully completed.
```

CDNO	CDPRODID	DATEOPENE	PRIMARY	COOWNER	CDAMT
CCD0000001	CD1	10-OCT-18	9900000001		500
CCD0000002	CD2	15-AUG-17	9900000002		1000
CCD0000003	CD1	15-DEC-19	9900000002		1000

CDNO	CDPRODID	DATEOPENE	PRIMARY	COOWNER	CDAMT
CCD0000001	CD1	10-OCT-18	9900000001		500
CCD0000002	CD2	15-AUG-17	9900000002		1000
CCD0000003	CD1	15-DEC-19	9900000002		1000
CCD0000004	CD3	31-OCT-19	9900000004	7700000005	7500

## Create procedure.



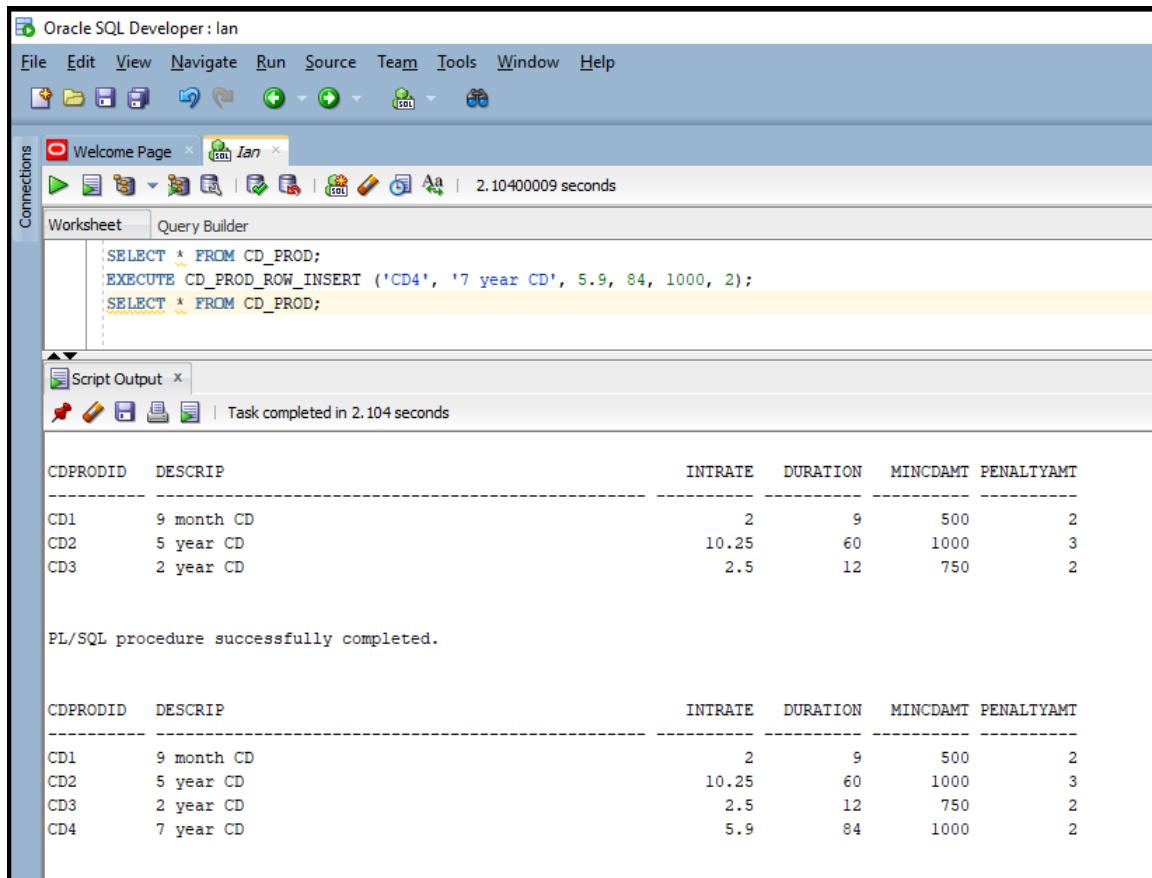
The screenshot shows the Oracle SQL Developer interface with a procedure creation script in the Worksheet tab. The script creates a procedure named CD\_PROD\_ROW\_INSERT with parameters for CDPRODID, DESCRIPT, INTRATE, DURATION, MINCDAMT, and PENALTYAMT. It inserts data into the CD\_PROD table and handles errors by outputting messages to the DBMS\_OUTPUT. The procedure is successfully compiled.

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE CD_PROD_ROW_INSERT (
    THE_CDPROID  VARCHAR2,
    THE_DESCRIP  VARCHAR2,
    THE_INTRATE  NUMBER,
    THE_DURATION NUMBER,
    THE_MINCDAMT NUMBER,
    THE_PENALTYAMT NUMBER
)
IS
BEGIN      -- Execution
    INSERT ALL
    INTO CD_PROD
        (CDPRODID, DESCRIPT, INTRATE, DURATION, MINCDAMT, PENALTYAMT)
    VALUES (THE_CDPROID, THE_DESCRIP, THE_INTRATE, THE_DURATION, THE_MINCDAMT, THE_PENALTYAMT)
    SELECT * FROM dual;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
END CD_PROD_ROW_INSERT;
/
```

Script Output x | Task completed in 0.65 seconds

Procedure CD\_PROD\_ROW\_INSERT compiled

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface with a query being run against a database. The query is as follows:

```
SELECT * FROM CD_PROD;
EXECUTE CD_PROD_ROW_INSERT ('CD4', '7 year CD', 5.9, 84, 1000, 2);
SELECT * FROM CD_PROD;
```

The results of the first query (before the procedure execution) are:

CDPRODID	DESCRIP	INTRATE	DURATION	MINCDAMT	PENALTYAMT
CD1	9 month CD	2	9	500	2
CD2	5 year CD	10.25	60	1000	3
CD3	2 year CD	2.5	12	750	2

The results of the second query (after the procedure execution) are:

CDPRODID	DESCRIP	INTRATE	DURATION	MINCDAMT	PENALTYAMT
CD1	9 month CD	2	9	500	2
CD2	5 year CD	10.25	60	1000	3
CD3	2 year CD	2.5	12	750	2
CD4	7 year CD	5.9	84	1000	2

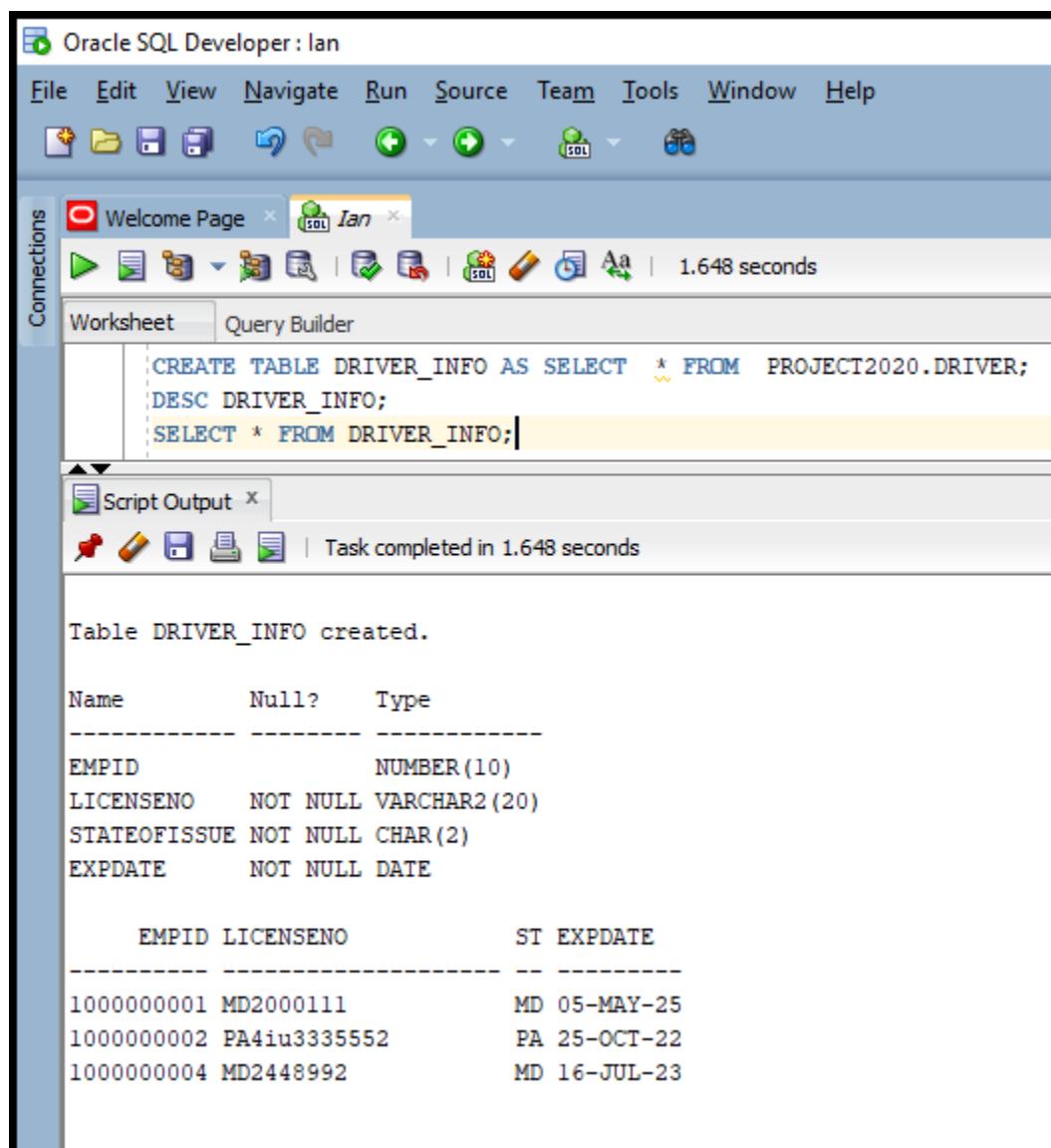
Below the tables, a message states: "PL/SQL procedure successfully completed."

d--Subprogram to insert a row in table driver.

**Create new table.**

**Check new table attribute data types.**

**Check new table contents.**



The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** A list of connections, with "Welcome Page" and "Ian" selected.
- Worksheet:** The main workspace where the following SQL code was run:

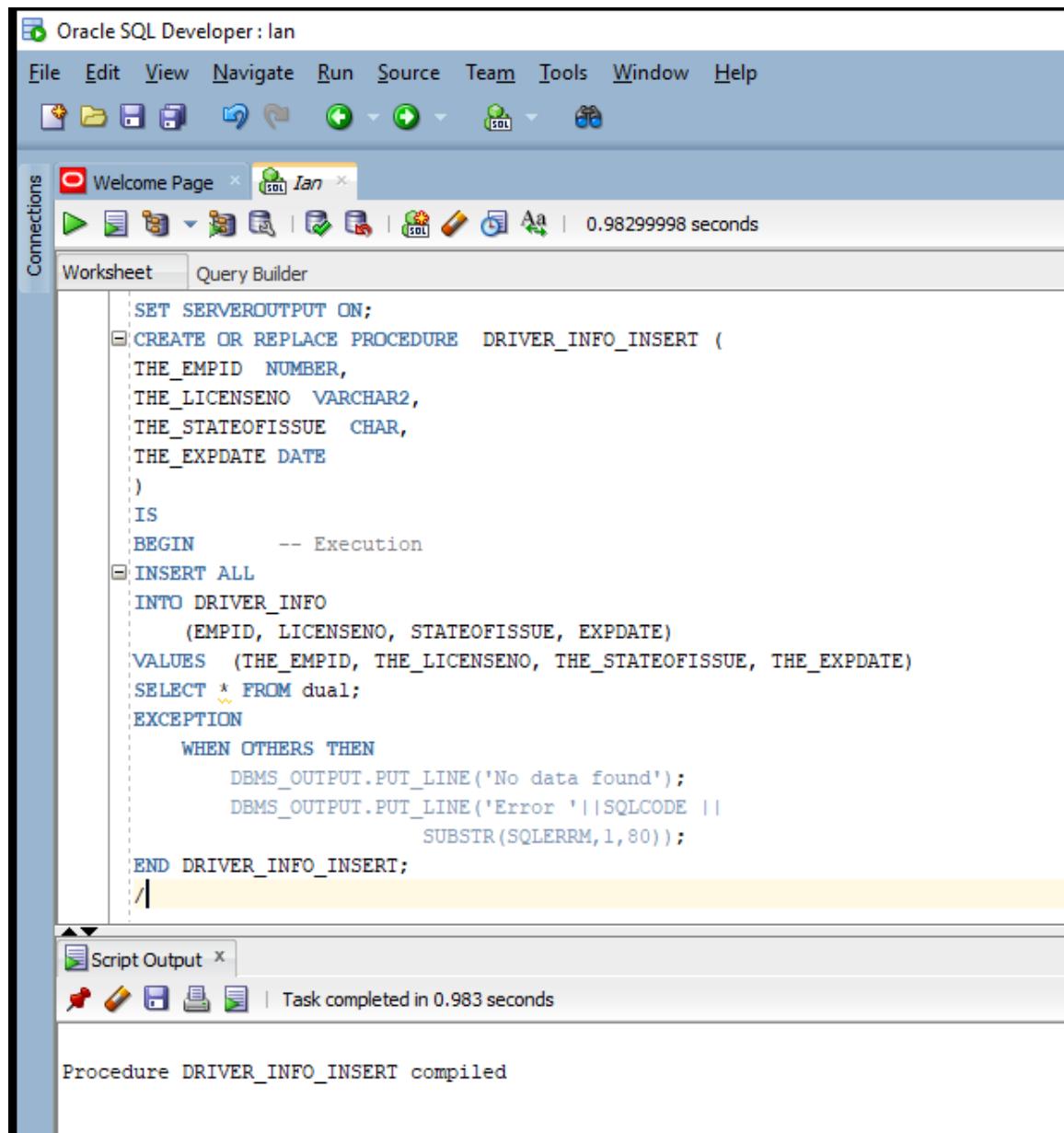
```
CREATE TABLE DRIVER_INFO AS SELECT * FROM PROJECT2020.DRIVER;
DESC DRIVER_INFO;
SELECT * FROM DRIVER_INFO;
```
- Script Output:** A panel showing the results of the execution:

```
Table DRIVER_INFO created.

Name      Null?    Type
-----  -----
EMPID          NUMBER(10)
LICENSENO      NOT NULL VARCHAR2(20)
STATEOFSUITE  NOT NULL CHAR(2)
EXPPDATE      NOT NULL DATE

EMPID LICENSENO      ST EXPPDATE
-----  -----
1000000001 MD2000111      MD 05-MAY-25
1000000002 PA4iu3335552      PA 25-OCT-22
1000000004 MD2448992      MD 16-JUL-23
```

## Create procedure.



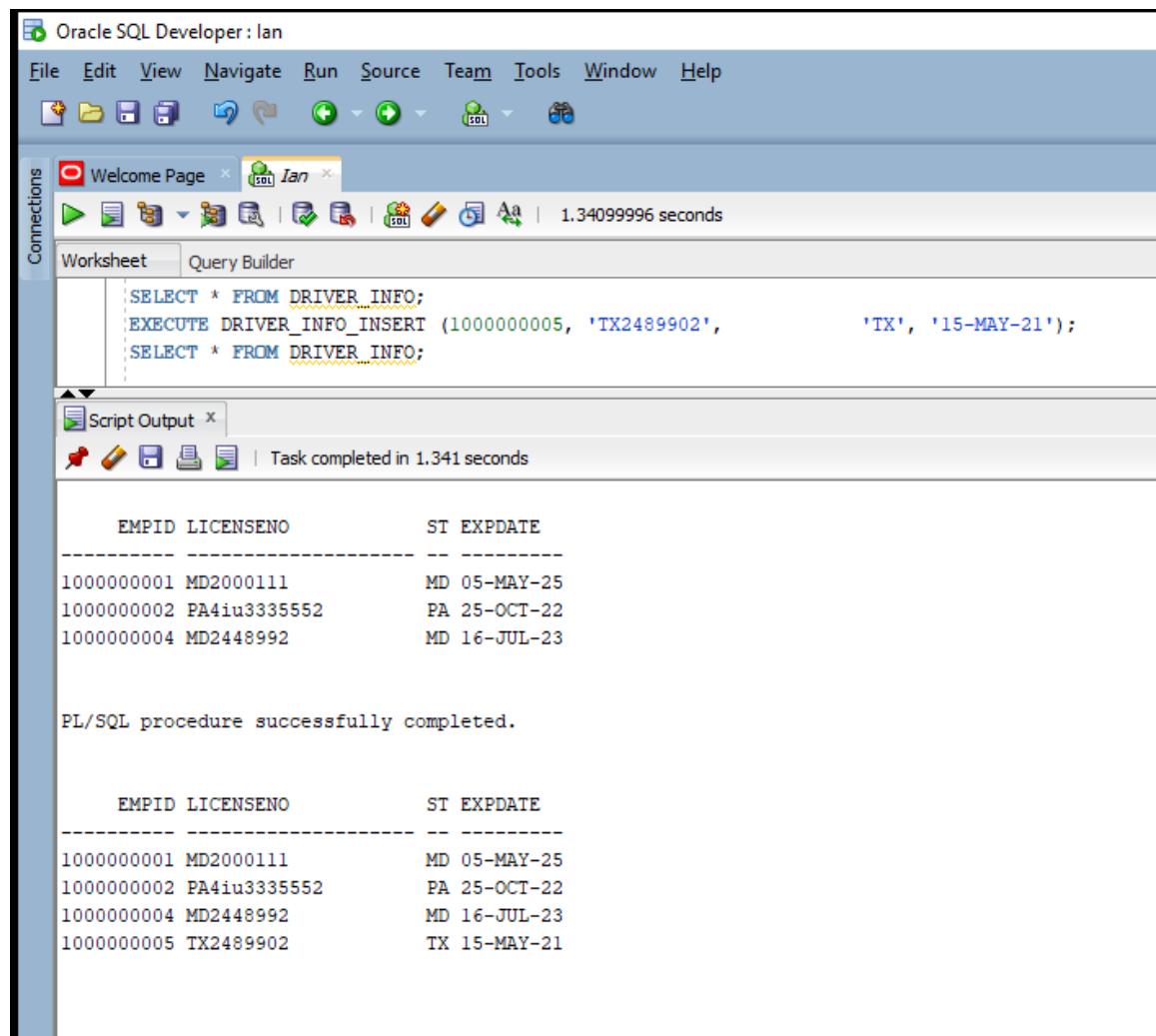
The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE DRIVER_INFO_INSERT (
    THE_EMPID NUMBER,
    THE_LICENSENO VARCHAR2,
    THE_STATEOFISSUE CHAR,
    THE_EXPDATE DATE
)
IS
BEGIN      -- Execution
INSERT ALL
    INTO DRIVER_INFO
        (EMPID, LICENSENO, STATEOFISSUE, EXPDATE)
    VALUES (THE_EMPID, THE_LICENSENO, THE_STATEOFISSUE, THE_EXPDATE)
    SELECT * FROM dual;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
            SUBSTR(SQLERRM,1,80));
END DRIVER_INFO_INSERT;
/
```

The code is a stored procedure named 'DRIVER\_INFO\_INSERT' that takes four parameters: 'THE\_EMPID', 'THE\_LICENSENO', 'THE\_STATEOFISSUE', and 'THE\_EXPDATE'. It begins with an 'IS' block containing a 'BEGIN' block for execution. Inside the 'BEGIN' block, an 'INSERT ALL' statement is used to insert data into the 'DRIVER\_INFO' table. The 'VALUES' clause uses the parameter names. A 'SELECT \* FROM dual' statement is included after the insert. An 'EXCEPTION' block handles errors, printing a message if none are caught. The procedure ends with an 'END' block and a final slash ('/').

The 'Script Output' tab at the bottom shows the message: 'Procedure DRIVER\_INFO\_INSERT compiled'.

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface with a single connection named 'Ian'. The 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT * FROM DRIVER_INFO;
EXECUTE DRIVER_INFO_INSERT (1000000005, 'TX2489902', 'TX', '15-MAY-21');
SELECT * FROM DRIVER_INFO;
```

The 'Script Output' tab shows the results of the execution:

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	PA4iu3335552	PA	25-OCT-22
1000000004	MD2448992	MD	16-JUL-23

Below this, a message indicates the procedure was successfully completed:

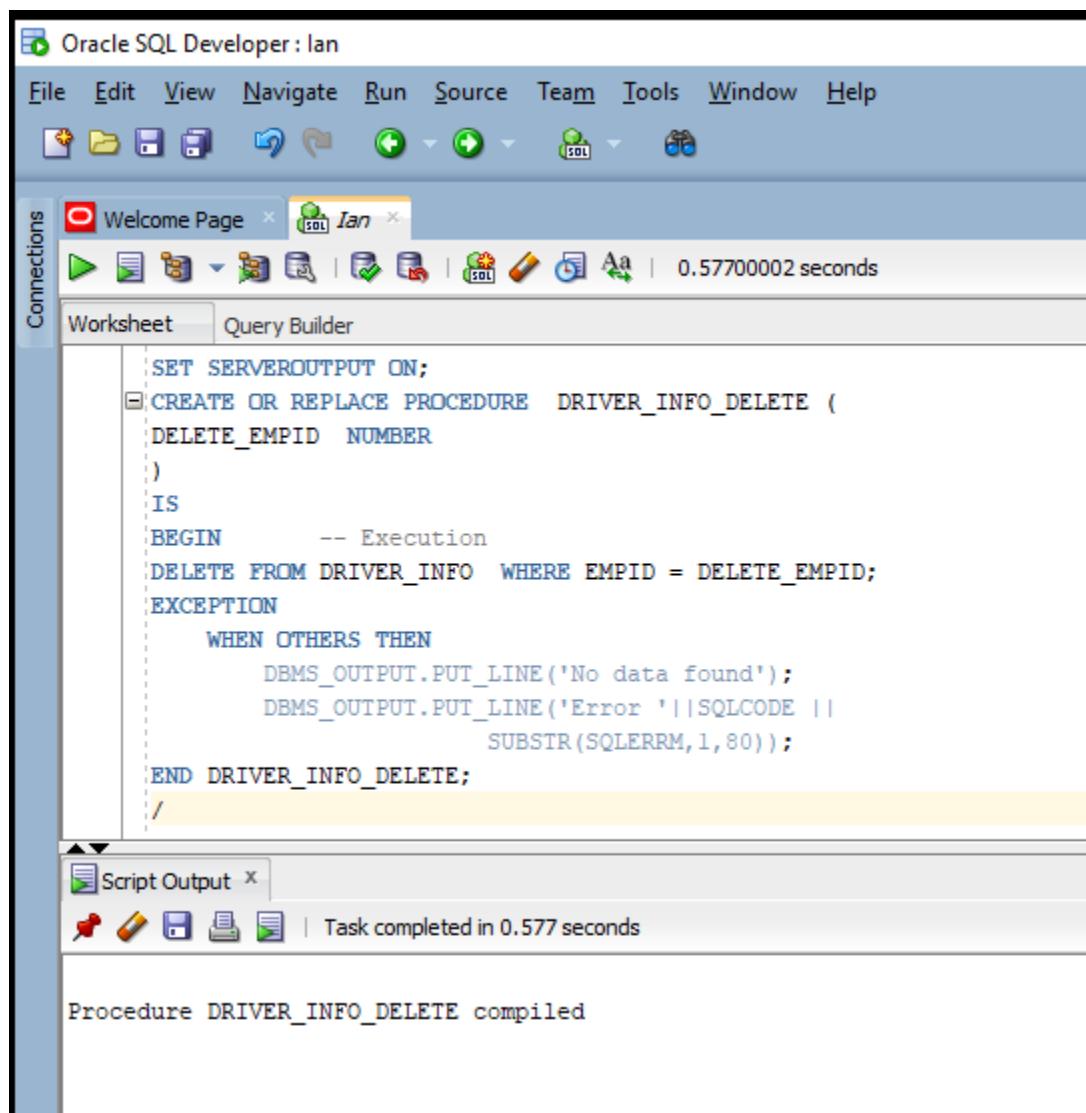
```
PL/SQL procedure successfully completed.
```

When the code is run again, the output shows the new row has been added:

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	PA4iu3335552	PA	25-OCT-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	TX2489902	TX	15-MAY-21

e--Subprogram to delete a row from table driver.

**Create procedure.**

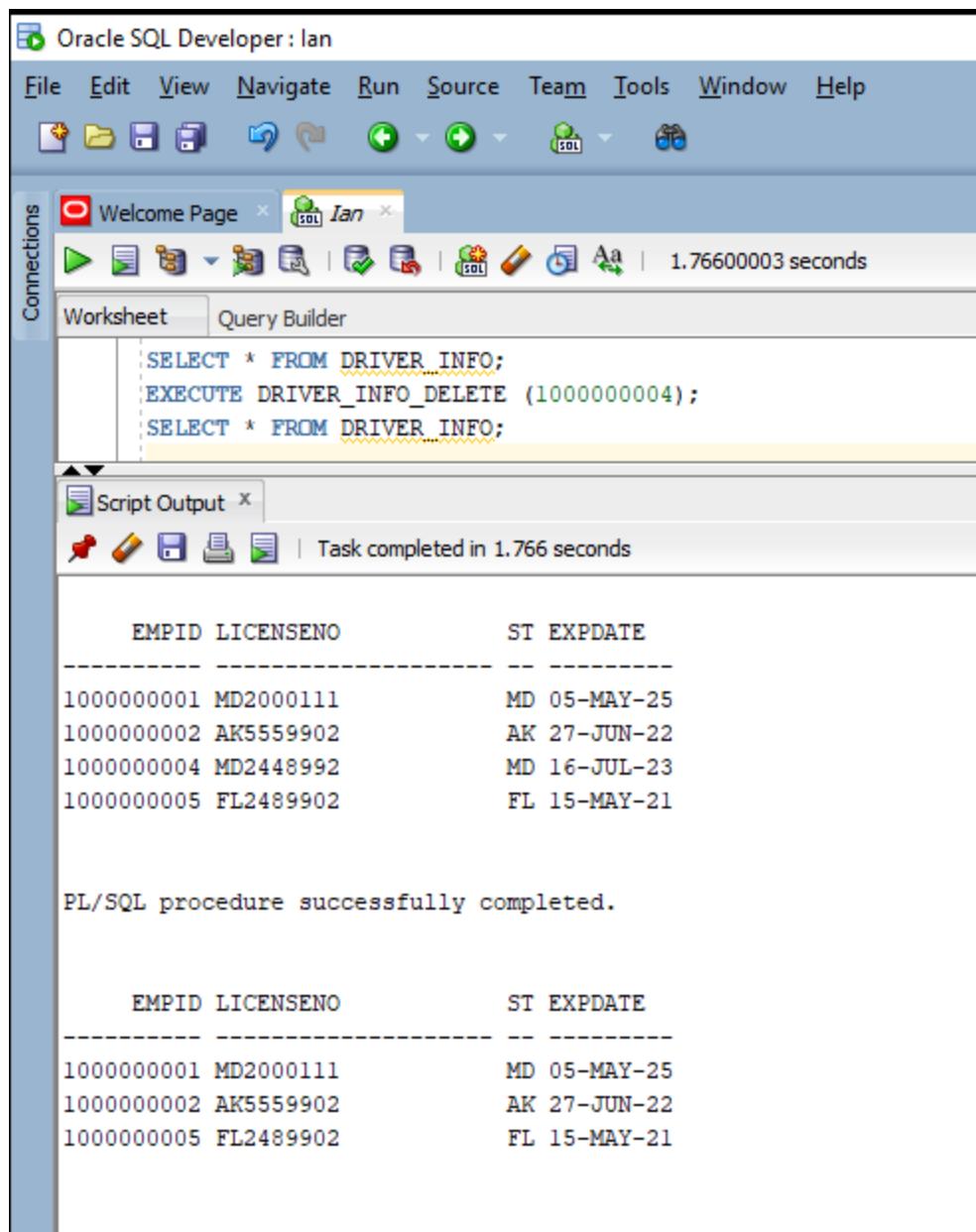


The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for connection, file, and code navigation. The Connections panel shows a connection named "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The code area contains the following PL/SQL procedure:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE DRIVER_INFO_DELETE (
  DELETE_EMPID NUMBER
)
IS
BEGIN      -- Execution
DELETE FROM DRIVER_INFO WHERE EMPID = DELETE_EMPID;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                         SUBSTR(SQLERRM,1,80));
END DRIVER_INFO_DELETE;
/
```

The "Script Output" panel at the bottom shows the message: "Procedure DRIVER\_INFO\_DELETE compiled".

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and navigation. The Connections sidebar shows a single connection named "Ian". The main workspace has two tabs: "Worksheet" and "Query Builder", with "Worksheet" selected. The worksheet contains the following PL/SQL code:

```
SELECT * FROM DRIVER_INFO;
EXECUTE DRIVER_INFO_DELETE (1000000004);
SELECT * FROM DRIVER_INFO;
```

The "Script Output" tab shows the results of the execution:

```
PL/SQL procedure successfully completed.
```

Below this, the output of the SELECT statements is displayed:

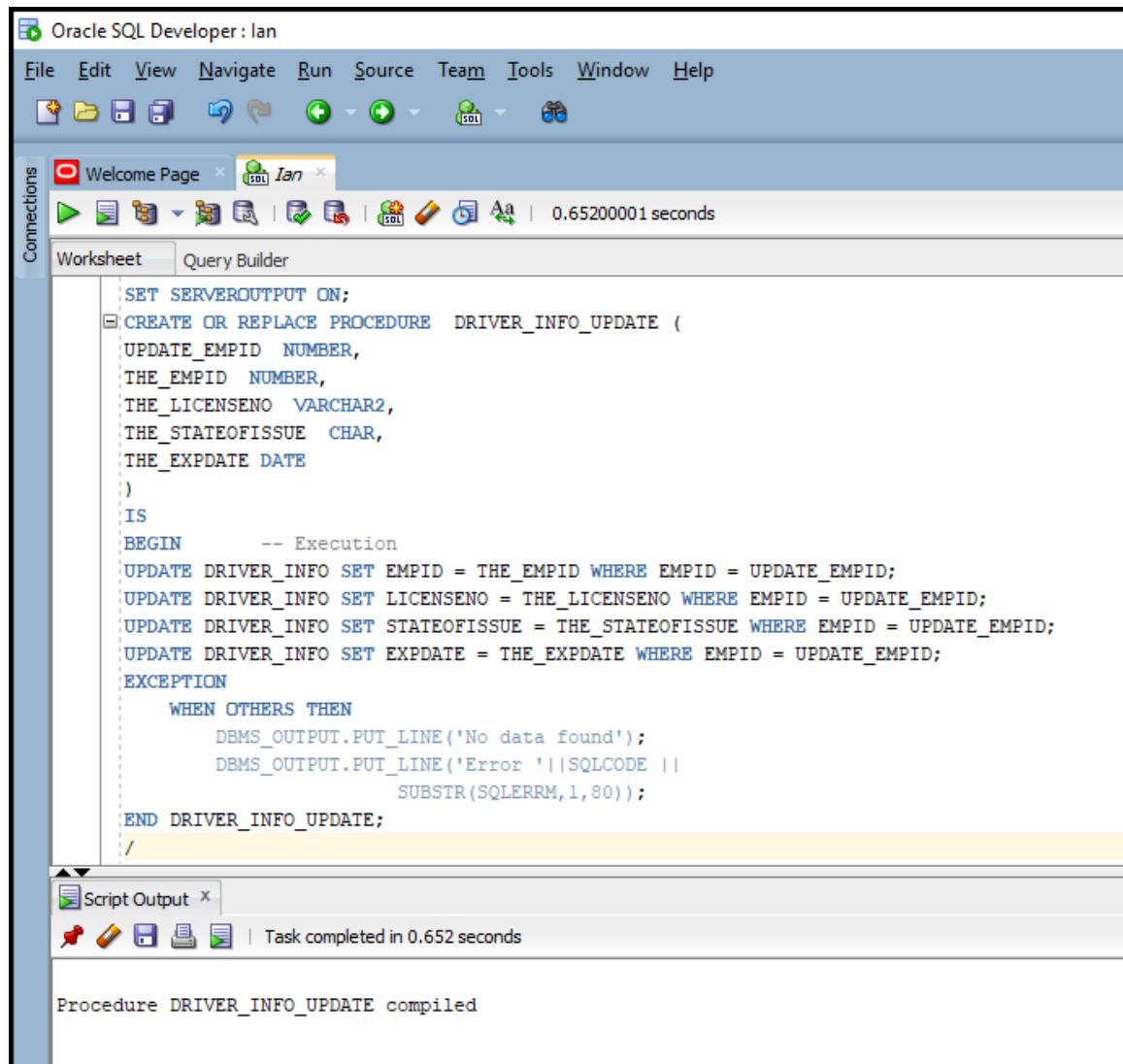
EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	AK5559902	AK	27-JUN-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	FL2489902	FL	15-MAY-21

PL/SQL procedure successfully completed.

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	AK5559902	AK	27-JUN-22
1000000005	FL2489902	FL	15-MAY-21

--Subprogram to update a row in table driver.

### Create procedure

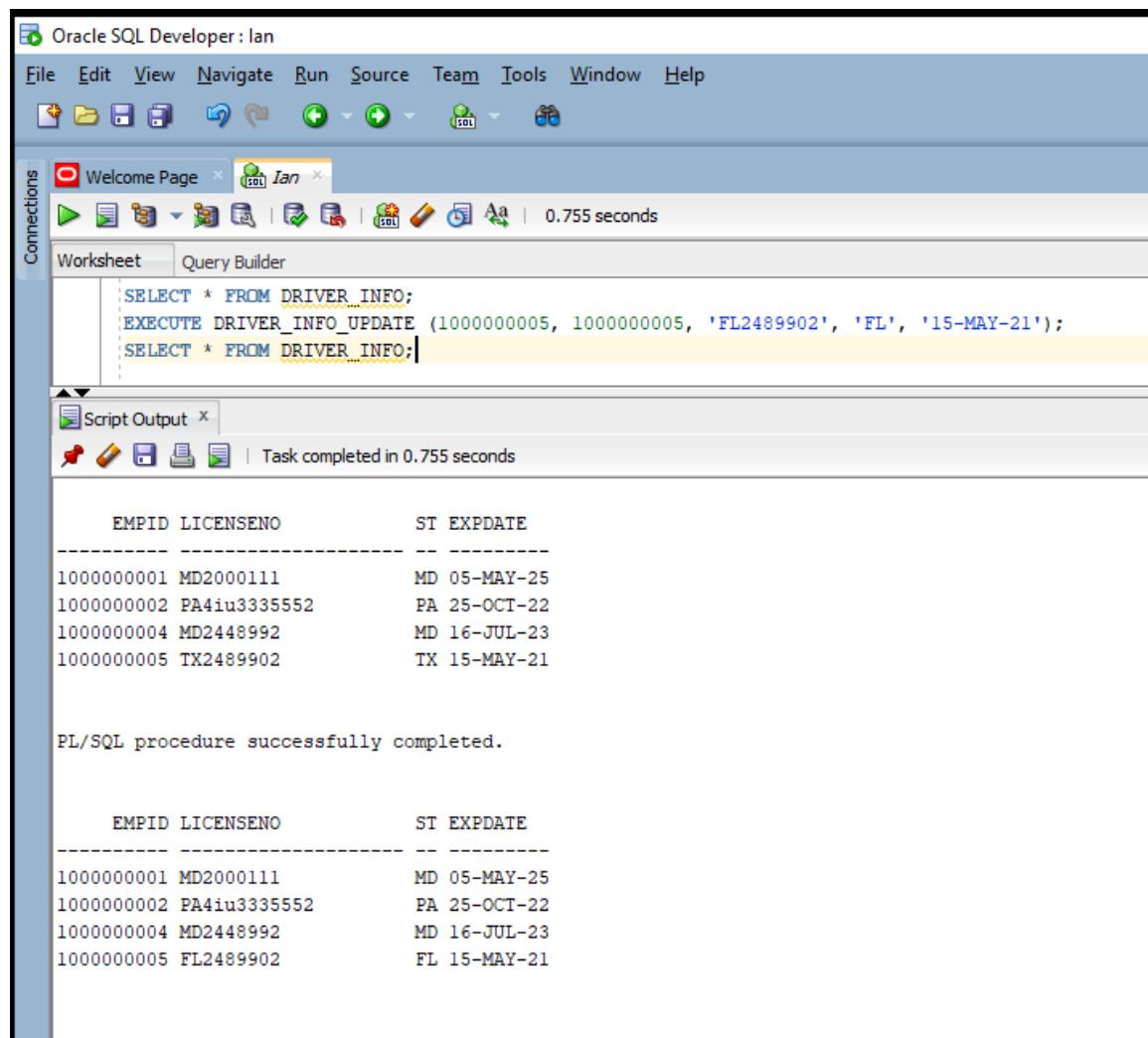


The screenshot shows the Oracle SQL Developer interface with a connection named 'lan'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE DRIVER_INFO_UPDATE (
    UPDATE_EMPID NUMBER,
    THE_EMPID NUMBER,
    THE_LICENSENO VARCHAR2,
    THE_STATEOFSISSION CHAR,
    THE_EXPDATE DATE
)
IS
BEGIN
    -- Execution
    UPDATE DRIVER_INFO SET EMPID = THE_EMPID WHERE EMPID = UPDATE_EMPID;
    UPDATE DRIVER_INFO SET LICENSENO = THE_LICENSENO WHERE EMPID = UPDATE_EMPID;
    UPDATE DRIVER_INFO SET STATEOFSISSION = THE_STATEOFSISSION WHERE EMPID = UPDATE_EMPID;
    UPDATE DRIVER_INFO SET EXPDATE = THE_EXPDATE WHERE EMPID = UPDATE_EMPID;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                            SUBSTR(SQLERRM,1,80));
END DRIVER_INFO_UPDATE;
/
```

The 'Script Output' tab at the bottom shows the message: 'Procedure DRIVER\_INFO\_UPDATE compiled'.

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : lan". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and SQL execution. The Connections panel shows a connection named "Jan". The Worksheet tab is active, displaying the following PL/SQL script:

```
SELECT * FROM DRIVER_INFO;
EXECUTE DRIVER_INFO_UPDATE (1000000005, 1000000005, 'FL2489902', 'FL', '15-MAY-21');
SELECT * FROM DRIVER_INFO;
```

The Script Output tab shows the results of the execution. The first part of the output displays the table contents before the update:

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	PA4iu3335552	PA	25-OCT-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	TX2489902	TX	15-MAY-21

Below this, a message states "PL/SQL procedure successfully completed." The second part of the output displays the table contents after the update:

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	PA4iu3335552	PA	25-OCT-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	FL2489902	FL	15-MAY-21

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
SELECT * FROM DRIVER_INFO;
EXECUTE DRIVER_INFO_UPDATE (1000000002, 1000000002, 'AK5559902', 'AK', '27-JUN-22');
SELECT * FROM DRIVER_INFO;
```

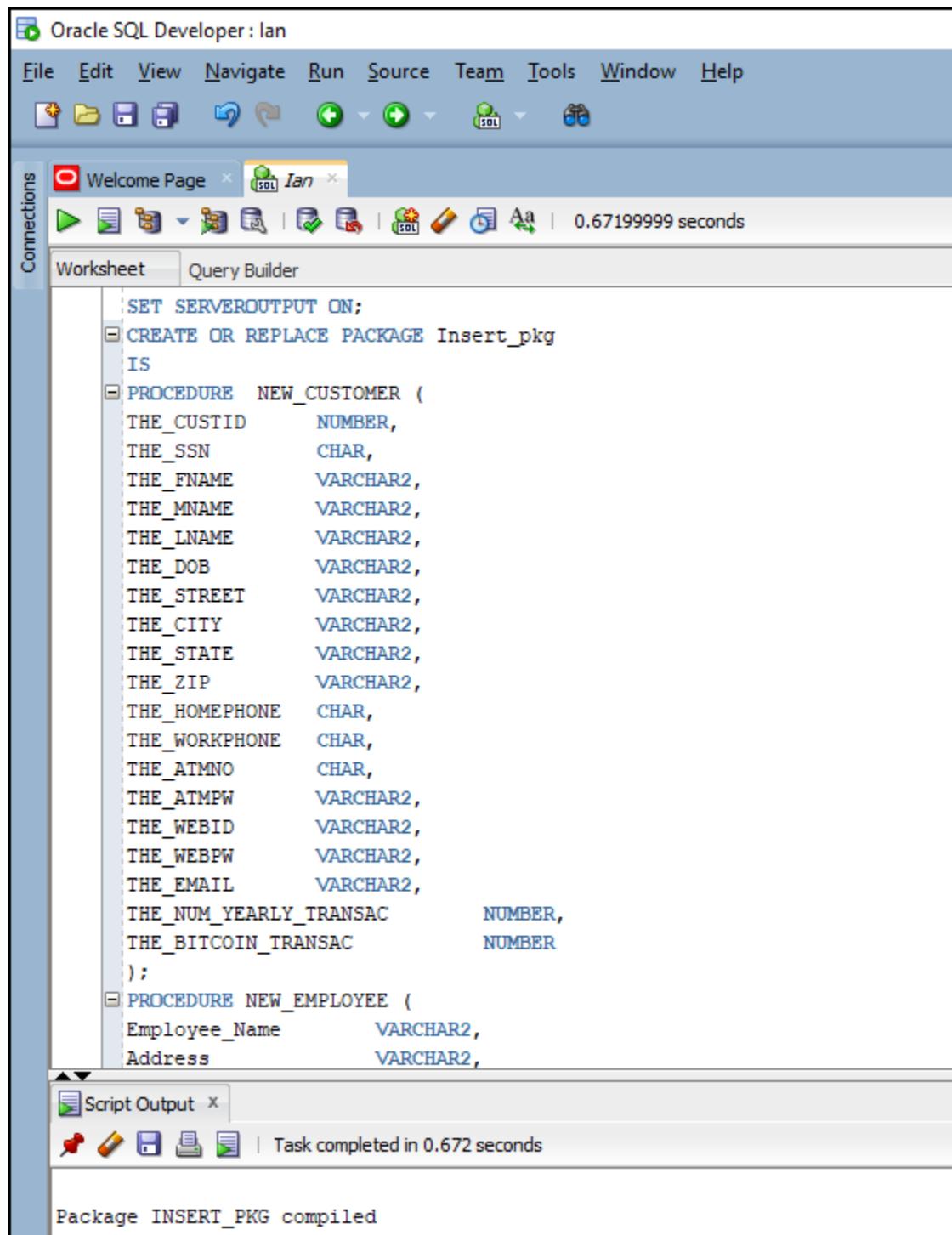
Script Output Task completed in 0.684 seconds

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	PA4iu3335552	PA	25-OCT-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	FL2489902	FL	15-MAY-21

PL/SQL procedure successfully completed.

EMPID	LICENSENO	ST	EXPDATE
1000000001	MD2000111	MD	05-MAY-25
1000000002	AK5559902	AK	27-JUN-22
1000000004	MD2448992	MD	16-JUL-23
1000000005	FL2489902	FL	15-MAY-21

## Create package



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PACKAGE Insert_pkg
IS
  PROCEDURE NEW_CUSTOMER (
    THE_CUSTID      NUMBER,
    THE_SSN         CHAR,
    THE_FNAME       VARCHAR2,
    THE_MNAME       VARCHAR2,
    THE_LNAME       VARCHAR2,
    THE_DOB         VARCHAR2,
    THE_STREET      VARCHAR2,
    THE_CITY        VARCHAR2,
    THE_STATE       VARCHAR2,
    THE_ZIP         VARCHAR2,
    THE_HOMEPHONE   CHAR,
    THE_WORKPHONE   CHAR,
    THE_ATMNO       CHAR,
    THE_ATMPW       VARCHAR2,
    THE_WEBID       VARCHAR2,
    THE_WEBPW       VARCHAR2,
    THE_EMAIL       VARCHAR2,
    THE_NUM_YEARLY_TRANSAC  NUMBER,
    THE_BITCOIN_TRANSAC  NUMBER
  );
  PROCEDURE NEW_EMPLOYEE (
    Employee_Name    VARCHAR2,
    Address          VARCHAR2,
  );

```

The code defines a package 'Insert\_pkg' with two procedures: 'NEW\_CUSTOMER' and 'NEW\_EMPLOYEE'. The 'NEW\_CUSTOMER' procedure takes 15 parameters of various data types. The 'NEW\_EMPLOYEE' procedure takes 2 parameters. The 'Worksheet' tab also shows a 'Script Output' section with the message 'Task completed in 0.672 seconds'.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections    Welcome Page    Ian

Worksheet    Query Builder

```
THE_NUM_YEARLY_TRANSAC      NUMBER,
THE_BITCOIN_TRANSAC        NUMBER
);
PROCEDURE NEW_EMPLOYEE (
Employee_Name      VARCHAR2,
Address           VARCHAR2,
Zip               VARCHAR2,
SSN               CHAR,
Title              VARCHAR2,
Curr_Year          CHAR,
Curr_Yrly_Salary   NUMBER,
Curr_Tax_Ded_Rate NUMBER,
Start_Date         DATE,
Birth_Date         DATE,
Age                NUMBER,
Emp_Phone_Ext     VARCHAR2,
Branch_Phone_Num  VARCHAR2,
Branch_Name        VARCHAR2,
Highest_Degree    VARCHAR2,
Highest_Degree_date VARCHAR2
);
PROCEDURE CD_ACCT_ROW_INSERT (
THE_CDNO CHAR,
THE_CDPRODID VARCHAR2,
THE_DATEOPENED DATE,
THE_PRIMARY NUMBER,
THE_COOWNER NUMBER,
```

Script Output    Task completed in 0.672 seconds

Package INSERT\_PKG compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
Highest_Degree      VARCHAR2,  
Highest_Degree_date VARCHAR2  
);  
PROCEDURE CD_ACCT_ROW_INSERT (  
  THE_CDNO CHAR,  
  THE_CDPRODID VARCHAR2,  
  THE_DATEOPENED DATE,  
  THE_PRIMARY NUMBER,  
  THE_COOWNER NUMBER,  
  THE_CDAMT NUMBER  
);  
PROCEDURE CD_PROD_ROW_INSERT (  
  THE_CDPRODID  VARCHAR2,  
  THE_DESCRIP    VARCHAR2,  
  THE_INTRATE   NUMBER,  
  THE_DURATION  NUMBER,  
  THE_MINCDAMT NUMBER,  
  THE_PENALTYAMT NUMBER  
);  
PROCEDURE DRIVER_INFO_INSERT (  
  THE_EMPID  NUMBER,  
  THE_LICENSENO VARCHAR2,  
  THE_STATEOFISSUE CHAR,  
  THE_EXPDATE DATE  
);  
PROCEDURE DRIVER_INFO_DELETE (  
  DELETE_EMPID NUMBER
```

Script Output

Task completed in 0.672 seconds

Package INSERT\_PKG compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
):
PROCEDURE CD_PROD_ROW_INSERT (
  THE_CDPRODID  VARCHAR2,
  THE_DESCRIP    VARCHAR2,
  THE_INTRATE   NUMBER,
  THE_DURATION  NUMBER,
  THE_MINCDAMT NUMBER,
  THE_PENALTYAMT NUMBER
);
PROCEDURE DRIVER_INFO_INSERT (
  THE_EMPIID  NUMBER,
  THE_LICENSENO  VARCHAR2,
  THE_STATEOFSISSION  CHAR,
  THE_EXPDATE DATE
);
PROCEDURE DRIVER_INFO_DELETE (
  DELETE_EMPIID  NUMBER
);
PROCEDURE DRIVER_INFO_UPDATE (
  UPDATE_EMPIID  NUMBER,
  THE_EMPIID  NUMBER,
  THE_LICENSENO  VARCHAR2,
  THE_STATEOFSISSION  CHAR,
  THE_EXPDATE DATE
);
END Insert_pkg;
/
```

Script Output Task completed in 0.672 seconds

Package INSERT\_PKG compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 1.3710005 seconds

Worksheet Query Builder

```
INSERT ALL
  INTO ALL_CUSTOMERS
    (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
     WORKPHONE, ATMMNO, ATMFW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES  (THE_CUSTID, THE_SSN, THE_FNAME, THE_MNAME, THE_LNAME, THE_DOB, THE_STREET, THE_CITY, THE_STATE, THE_ZIP, THE_HOMEPHONE,
           THE_WORKPHONE, THE_ATMMNO, THE_ATMFW, THE_WEBID, THE_WEBPW, THE_EMAIL, THE_NUM_YEARLY_TRANSAC, THE_BITCOIN_TRANSAC)
SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE||SUBSTR(SQLERRM,1,80));
  END NEW_CUSTOMER;
PROCEDURE NEW_EMPLOYEE (
  Employee_Name      VARCHAR2,
  Address            VARCHAR2,
  Zip                VARCHAR2,
  SSN                CHAR,
  Title              VARCHAR2,
  Curr_Year          CHAR,
  Curr_Yrly_Salary   NUMBER,
  Curr_Tax_Ded_Rate  NUMBER,
  Start_Date         DATE,
  Birth_Date         DATE,
  Age                NUMBER,
  Emp_Phone_Ext     VARCHAR2,
  Branch_Phone_Num  VARCHAR2,
```

Script Output x | Task completed in 1.371 seconds

Package Body INSERT\_PKG compiled

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page (Jan) 1.3710005 seconds

Worksheet Query Builder

```
})
IS
BEGIN      -- Execution
INSERT ALL
  INTO BANK_EMPLOYEES_DATA
    ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
     "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
     "Highest Degree date")
   VALUES  (Employee_Name, Address, Zip, SSN, Title, Curr_Year, Curr_Yrly_Salary, Curr_Tax_Ded_Rate,
            Start_Date, Birth_Date, Age, Emp_Phone_Ext, Branch_Phone_Num, Branch_Name, Highest_Degree, Highest_Degree_date)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                         SUBSTR(SQLERRM,1,80));
END NEW_EMPLOYEE;
PROCEDURE CD_ACCT_ROW_INSERT (
  THE_CDNO CHAR,
  THE_CDPROID VARCHAR2,
  THE_DATEOPENED DATE,
  THE_PRIMARY NUMBER,
  THE_COOWNER NUMBER,
  THE_CDAMT NUMBER
)
IS
BEGIN      -- Execution

```

Script Output Task completed in 1.371 seconds

Package Body INSERT\_PKG compiled

Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Edit Ian x

Worksheet Query Builder

```
INSERT ALL
  INTO CD_ACCT
    (CDNO, CDPROID, DATEOPENED, PRIMARY, COOWNER, CDAMT)
  VALUES  (THE_CDNO, THE_CDPROID, THE_DATEOPENED, THE_PRIMARY, THE_COOWNER, THE_CDAMT)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
      SUBSTR(SQLERRM,1,80));
END CD_ACCT_ROW_INSERT;
PROCEDURE CD_PROD_ROW_INSERT (
  THE_CDPROID VARCHAR2,
  THE_DESCRIP  VARCHAR2,
  THE_INTRATE  NUMBER,
  THE_DURATION NUMBER,
  THE_MINCDAMT NUMBER,
  THE_PENALTYAMT NUMBER
)
IS
BEGIN      -- Execution
INSERT ALL
  INTO CD_PROD
    (CDPROID, DESCRIPT, INTRATE, DURATION, MINCDAMT, PENALTYAMT)
  VALUES  (THE_CDPROID, THE_DESCRIP, THE_INTRATE, THE_DURATION, THE_MINCDAMT, THE_PENALTYAMT)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
      SUBSTR(SQLERRM,1,80));
END CD_PROD_ROW_INSERT;
```

Script Output x

Task completed in 1.371 seconds

Package Body INSERT\_PKG compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
      SUBSTR(SQLERRM,1,80));
END DRIVER_INFO_DELETE;
PROCEDURE DRIVER_INFO_UPDATE (
  UPDATE_EMPID NUMBER,
  THE_EMPID NUMBER,
  THE_LICENSENO VARCHAR2,
  THE_STATEOFISSUE CHAR,
  THE_EXPDATE DATE
)
IS
BEGIN      -- Execution
  UPDATE DRIVER_INFO SET EMPID = THE_EMPID WHERE EMPID = UPDATE_EMPID;
  UPDATE DRIVER_INFO SET LICENSENO = THE_LICENSENO WHERE EMPID = UPDATE_EMPID;
  UPDATE DRIVER_INFO SET STATEOFISSUE = THE_STATEOFISSUE WHERE EMPID = UPDATE_EMPID;
  UPDATE DRIVER_INFO SET EXPDATE = THE_EXPDATE WHERE EMPID = UPDATE_EMPID;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
      SUBSTR(SQLERRM,1,80));
END DRIVER_INFO_UPDATE;
END Insert_pkg;
/
```

Script Output Task completed in 1.371 seconds

Package Body INSERT\_PKG compiled

^^^^^^^^^^^^^^^^^^^^^^^^

## CHAPTER 4D Starts here

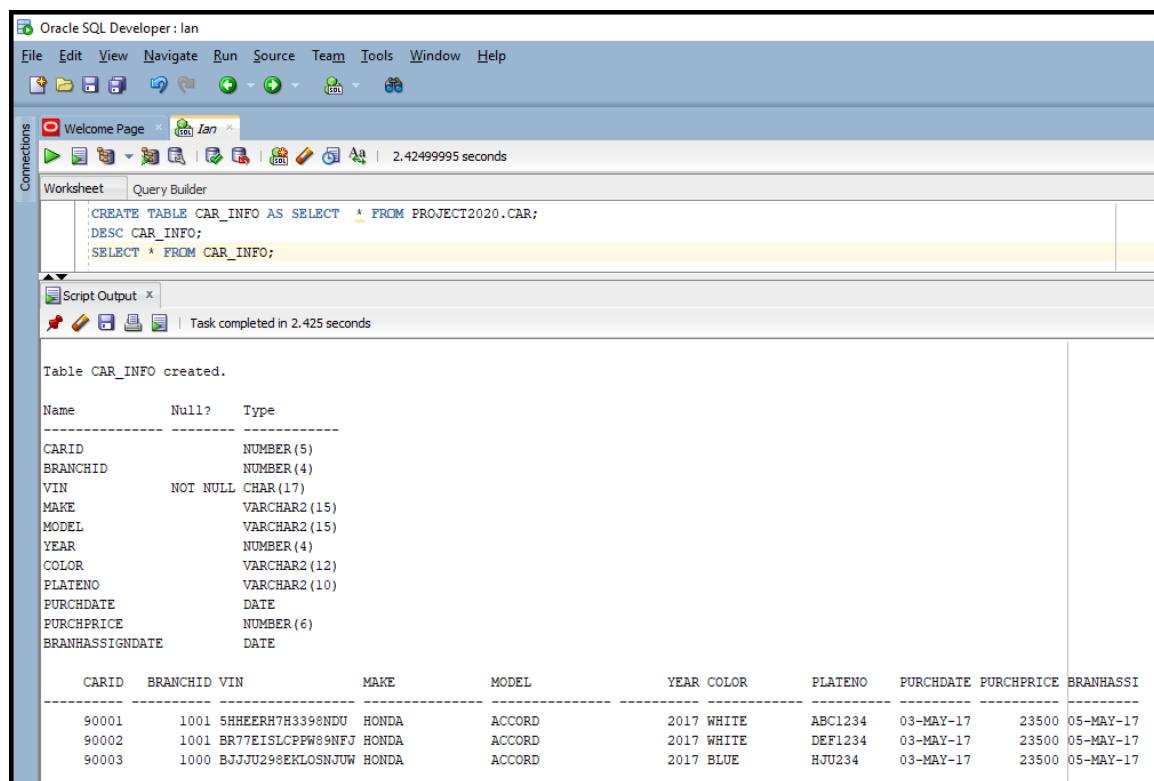
^^^^^^^^^^^^^^^^^^^^^^^^

**D. Create a useful package of your choice with functions, procedures, and datatype of your own.**

**Create new table.**

**Check new table attribute data types.**

**Check new table contents.**



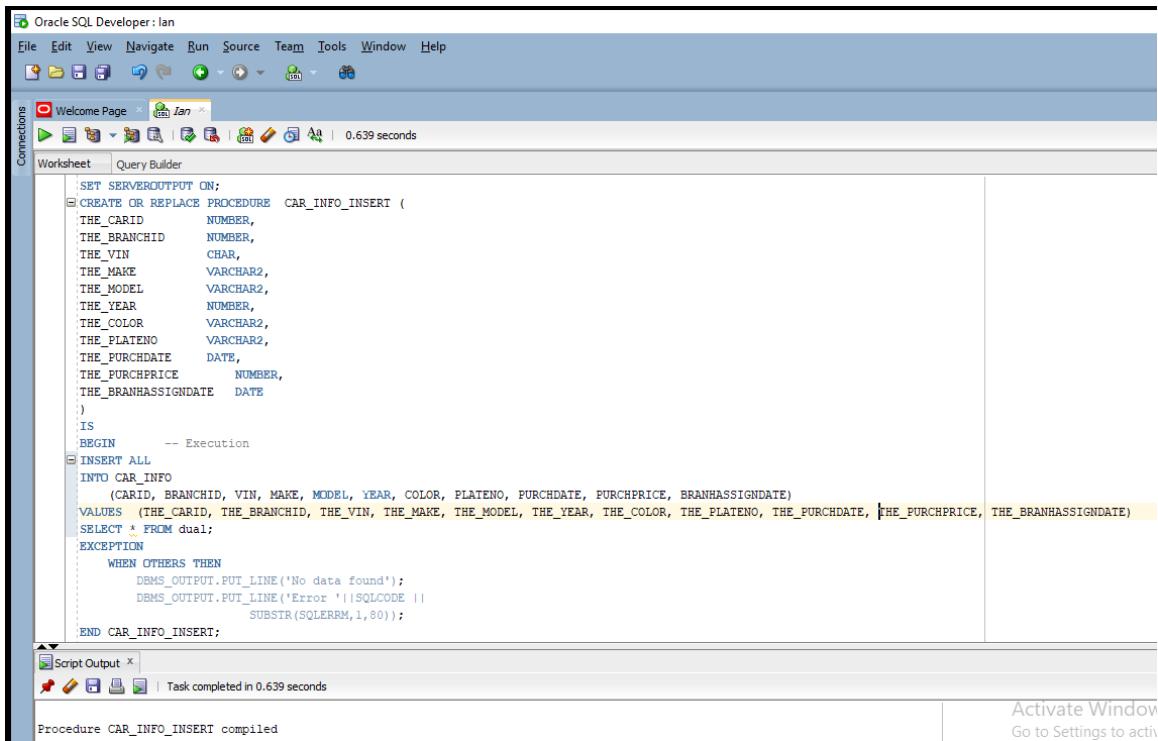
```
CREATE TABLE CAR_INFO AS SELECT * FROM PROJECT2020.CAR;
DESC CAR_INFO;
SELECT * FROM CAR_INFO;
```

Table CAR\_INFO created.

Name	Null?	Type
CARID		NUMBER (5)
BRANCHID		NUMBER (4)
VIN	NOT NULL	CHAR (17)
MAKE		VARCHAR2 (15)
MODEL		VARCHAR2 (15)
YEAR		NUMBER (4)
COLOR		VARCHAR2 (12)
PLATENO		VARCHAR2 (10)
PURCHDATE		DATE
PURCHPRICE		NUMBER (6)
BRANHASSIGNDATE		DATE

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHDATE	PURCHPRICE	BRANHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77E1S1CPFW89NFJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17

## Create procedure.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL code for creating a procedure. The code is as follows:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE CAR_INFO_INSERT (
  THE_CARID      NUMBER,
  THE_BRANCHID   NUMBER,
  THE_VIN        CHAR,
  THE_MAKE       VARCHAR2,
  THE_MODEL      VARCHAR2,
  THE_YEAR       NUMBER,
  THE_COLOR      VARCHAR2,
  THE_PLATENO    VARCHAR2,
  THE_PURCHASEDATE DATE,
  THE_PURCHPRICE NUMBER,
  THE_BRANHASSIGNDATE DATE
)
IS
BEGIN      -- Execution
  INSERT ALL
  INTO CAR_INFO
  (CARID, BRANCHID, VIN, MAKE, MODEL, YEAR, COLOR, PLATENO, PURCHASEDATE, PURCHPRICE, BRANHASSIGNDATE)
  VALUES (THE_CARID, THE_BRANCHID, THE_VIN, THE_MAKE, THE_MODEL, THE_YEAR, THE_COLOR, THE_PLATENO, THE_PURCHASEDATE, THE_PURCHPRICE, THE_BRANHASSIGNDATE)
  SELECT * FROM dual;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
    SUBSTR(SQLERRM,1,80));
END CAR_INFO_INSERT;
```

The 'Script Output' tab shows the message: 'Procedure CAR\_INFO\_INSERT compiled'. The status bar at the bottom indicates 'Task completed in 0.639 seconds'.

**Check table contents.**  
**Execute procedure.**  
**Recheck table contents.**

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 2.6059995 seconds

Worksheet Query Builder

```
SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_INSERT (90004, 1002, 'BTYUU298EKLOSNJUW', 'TOYOTA', 'COROLLA', 2019, 'TAN', 'HGN234', '14-MAY-17', 24000, '15-MAY-17');
SELECT * FROM CAR_INFO;

SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_INSERT (90005, 1003, 'SABUU098VVLOSNJUW', 'NISSAN', 'SENTRA', 2020, 'TAN', 'MLK777', '18-JUN-17', 22700, '19-JUN-17');
SELECT * FROM CAR_INFO;
```

Script Output Task completed in 2.606 seconds

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77E1SLCPW89NTJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJUW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17

PL/SQL procedure successfully completed.

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77E1SLCPW89NTJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJUW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1002	BTYUU298EKLOSNJUW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17

Activ Go to S

**Check table contents.  
Execute procedure.  
Recheck table contents.**

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 2.60599995 seconds

Worksheet Query Builder

```
SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_INSERT (90004, 1002, 'BYUU298EKLOSNJUW', 'TOYOTA', 'COROLLA', 2019, 'TAN', 'HGN234', '14-MAY-17', 24000, '15-MAY-17');

SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_INSERT (90005, 1003, 'SABUU098VVLOSNJUW', 'NISSAN', 'SENTRA', 2020, 'TAN', 'MLK777', '18-JUN-17', 22700, '19-JUN-17');
SELECT * FROM CAR_INFO;
```

Script Output Task completed in 2.606 seconds

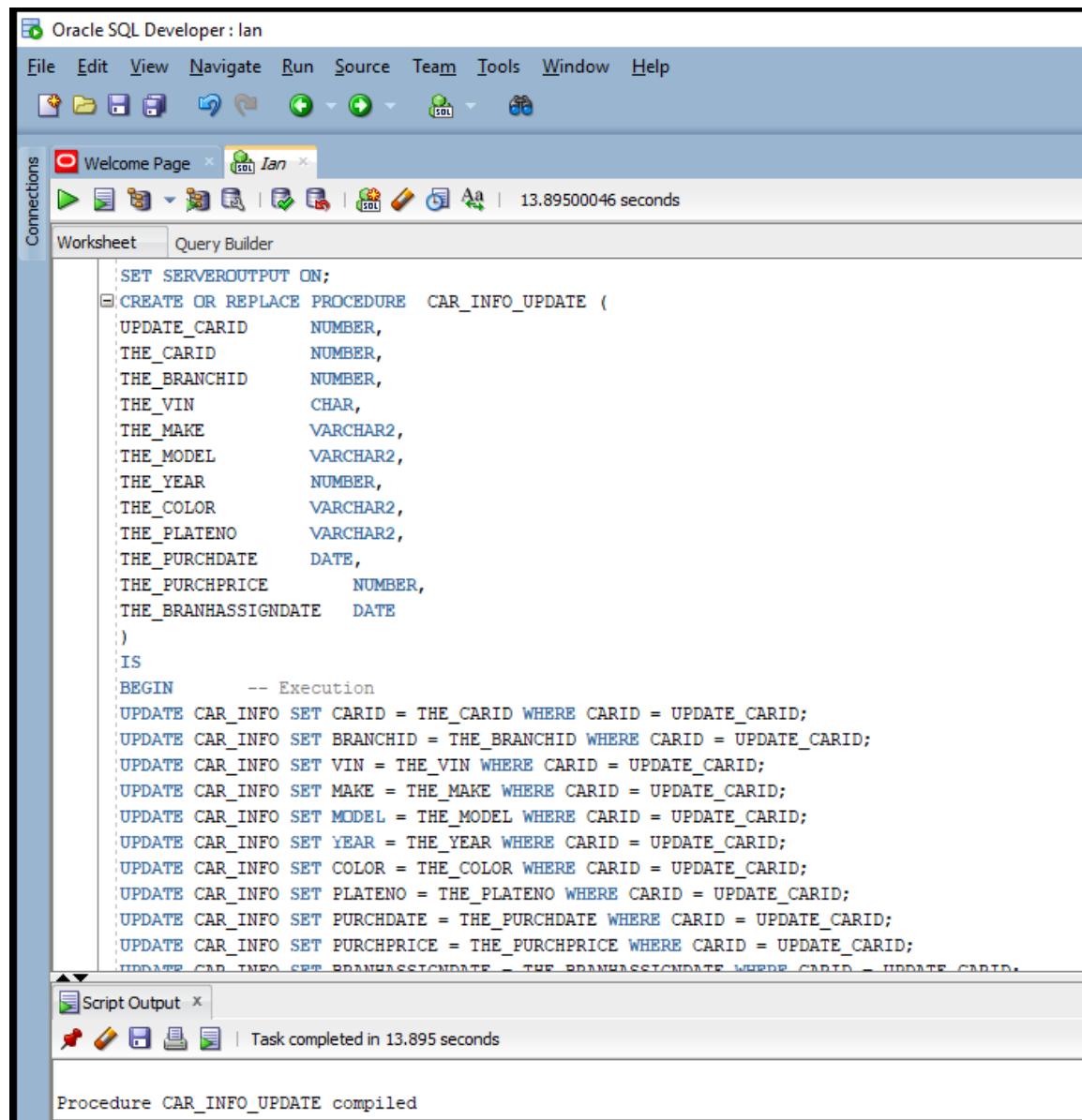
CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77EISLCPPW89NFJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJU298EKLOSNJUW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1002	BYUU298EKLOSNJUW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17

PL/SQL procedure successfully completed.

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77EISLCPPW89NFJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJU298EKLOSNJUW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1002	BYUU298EKLOSNJUW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17
90005	1003	SABUU098VVLOSNJUW	NISSAN	SENTRA	2020	TAN	MLK777	18-JUN-17	22700	19-JUN-17

Activ  
Go to

## Create procedure.



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the creation of a stored procedure named 'CAR\_INFO\_UPDATE'. The procedure takes several parameters: UPDATE\_CARID (NUMBER), THE\_CARID (NUMBER), THE\_BRANCHID (NUMBER), THE\_VIN (CHAR), THE\_MAKE (VARCHAR2), THE\_MODEL (VARCHAR2), THE\_YEAR (NUMBER), THE\_COLOR (VARCHAR2), THE\_PLATENO (VARCHAR2), THE\_PURCHDATE (DATE), THE\_PURCHPRICE (NUMBER), and THE\_BRANHASSIGNDATE (DATE). The code then executes multiple UPDATE statements for the 'CAR\_INFO' table, setting various fields (CARID, BRANCHID, VIN, MAKE, MODEL, YEAR, COLOR, PLATENO, PURCHDATE, PURCHPRICE, and BRANHASSIGNDATE) to the corresponding parameter values. The 'Script Output' tab at the bottom shows the message 'Procedure CAR\_INFO\_UPDATE compiled' and a completion time of 13.895 seconds.

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE CAR_INFO_UPDATE (
  UPDATE_CARID      NUMBER,
  THE_CARID         NUMBER,
  THE_BRANCHID      NUMBER,
  THE_VIN           CHAR,
  THE_MAKE          VARCHAR2,
  THE_MODEL         VARCHAR2,
  THE_YEAR          NUMBER,
  THE_COLOR         VARCHAR2,
  THE_PLATENO       VARCHAR2,
  THE_PURCHDATE     DATE,
  THE_PURCHPRICE    NUMBER,
  THE_BRANHASSIGNDATE DATE
)
IS
BEGIN      -- Execution
  UPDATE CAR_INFO SET CARID = THE_CARID WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET BRANCHID = THE_BRANCHID WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET VIN = THE_VIN WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET MAKE = THE_MAKE WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET MODEL = THE_MODEL WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET YEAR = THE_YEAR WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET COLOR = THE_COLOR WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET PLATENO = THE_PLATENO WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET PURCHDATE = THE_PURCHDATE WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET PURCHPRICE = THE_PURCHPRICE WHERE CARID = UPDATE_CARID;
  UPDATE CAR_INFO SET BRANHASSIGNDATE = THE_BRANHASSIGNDATE WHERE CARID = UPDATE_CARID;

```

Script Output | Task completed in 13.895 seconds

Procedure CAR\_INFO\_UPDATE compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page (SQL Ian)

Worksheet Query Builder

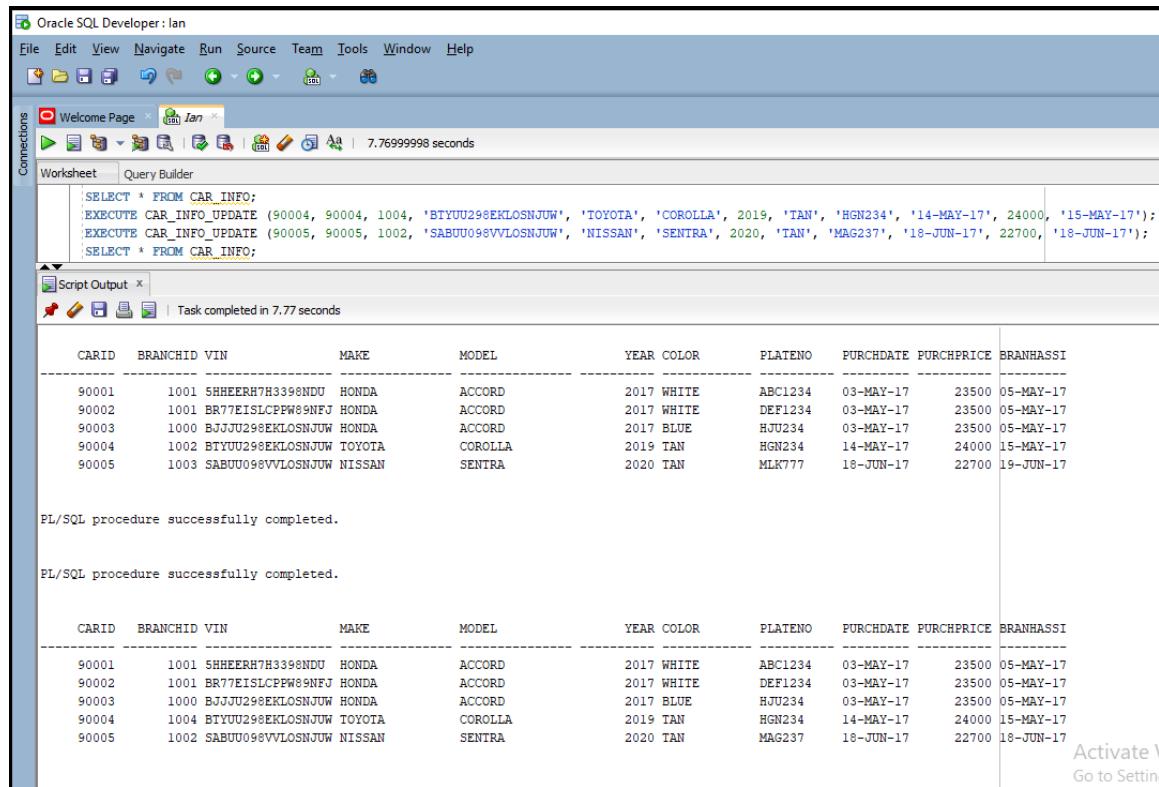
```
THE_COLOR      VARCHAR2,
THE_PLATENO    VARCHAR2,
THE_PURCHASEDATE DATE,
THE_PURCHASEPRICE NUMBER,
THE_BRANHASSIGNDATE DATE
)
IS
BEGIN      -- Execution
UPDATE CAR_INFO SET CARID = THE_CARID WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET BRANCHID = THE_BRANCHID WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET VIN = THE_VIN WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET MAKE = THE_MAKE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET MODEL = THE_MODEL WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET YEAR = THE_YEAR WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET COLOR = THE_COLOR WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PLATENO = THE_PLATENO WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PURCHASEDATE = THE_PURCHASEDATE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PURCHASEPRICE = THE_PURCHASEPRICE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET BRANHASSIGNDATE = THE_BRANHASSIGNDATE WHERE CARID = UPDATE_CARID;

EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('No data found');
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
SUBSTR(SQLERRM,1,80));
END CAR_INFO_UPDATE;
/
```

Script Output Task completed in 13.895 seconds

Procedure CAR\_INFO\_UPDATE compiled

## Create procedure.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains a script block with the following PL/SQL code:

```
SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_UPDATE (90004, 90004, 1004, 'BYUU298EKLOSNJUW', 'TOYOTA', 'COROLLA', 2019, 'TAN', 'HGN234', '14-MAY-17', 24000, '15-MAY-17');
EXECUTE CAR_INFO_UPDATE (90005, 90005, 1002, 'SABUU098VVLOSNJUW', 'NISSAN', 'SENTRA', 2020, 'TAN', 'MAG237', '18-JUN-17', 22700, '18-JUN-17');
SELECT * FROM CAR_INFO;
```

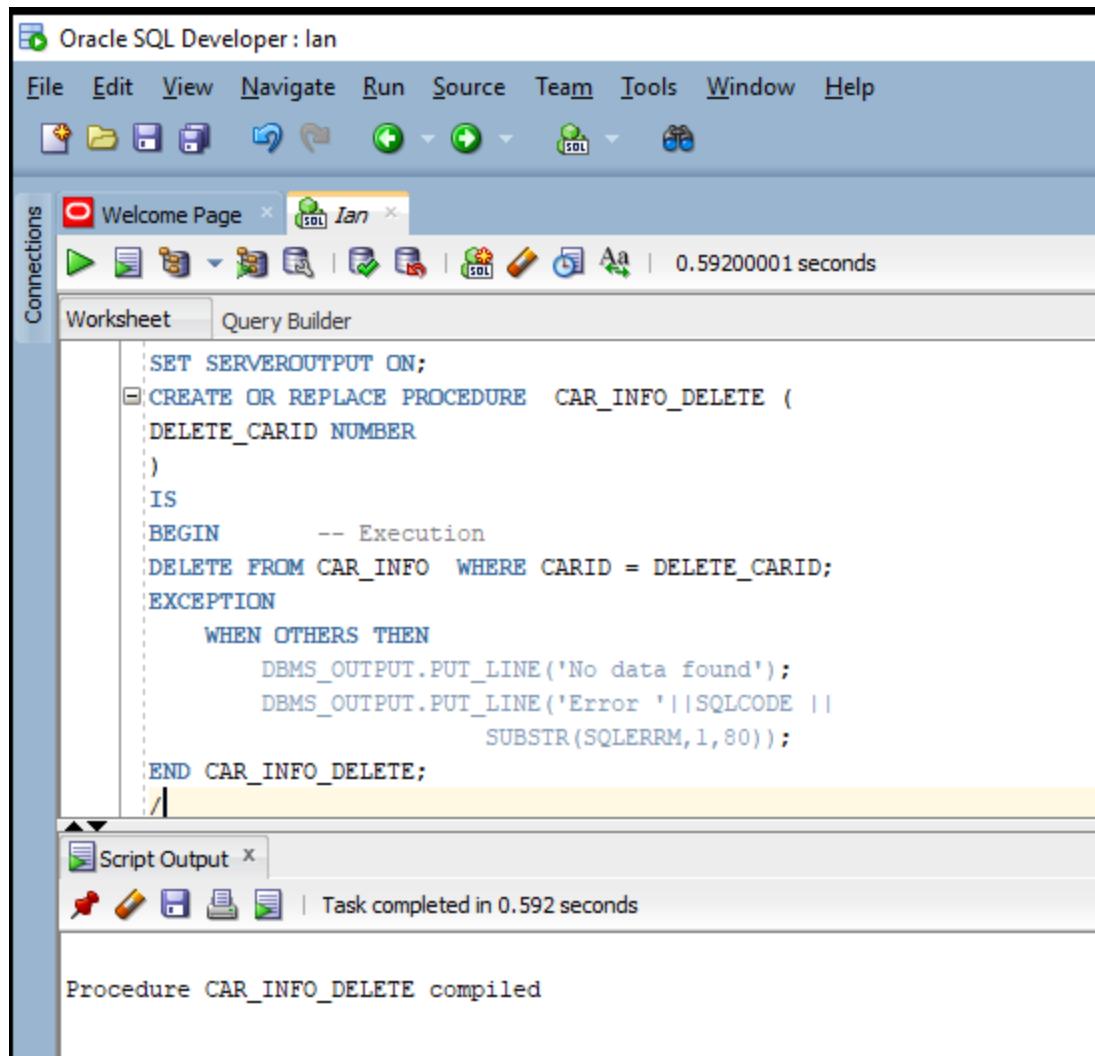
The 'Script Output' tab shows the results of the execution:

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77EISLCPPW89NFU	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJUW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1002	BYUU298EKLOSNJUW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17
90005	1003	SABUU098VVLOSNJUW	NISSAN	SENTRA	2020	TAN	MLK777	18-JUN-17	22700	18-JUN-17

Below the table, the message 'PL/SQL procedure successfully completed.' is displayed twice.

At the bottom right of the interface, there are 'Activate \ Go to Settings' buttons.

## Create procedure.

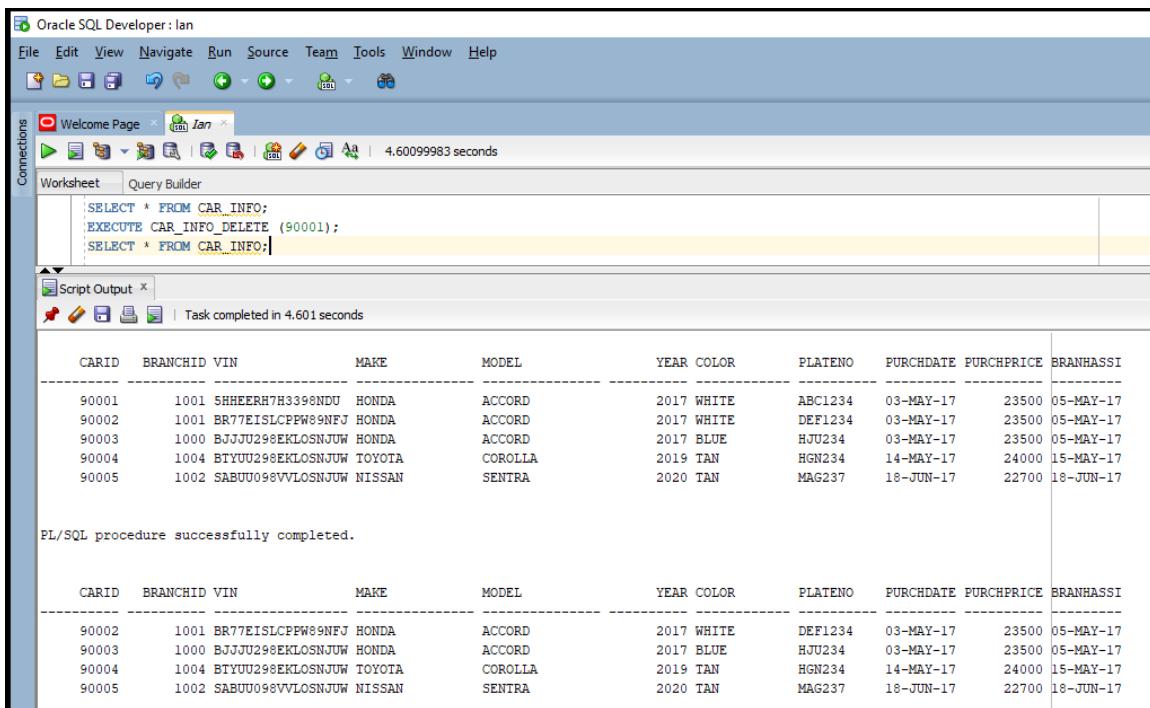


The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations and navigation. The Connections sidebar shows a connection to "Ian". The main workspace has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The code area contains the following PL/SQL procedure:

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE CAR_INFO_DELETE (
  DELETE_CARID NUMBER
)
IS
BEGIN      -- Execution
DELETE FROM CAR_INFO WHERE CARID = DELETE_CARID;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('No data found');
    DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                         SUBSTR(SQLERRM,1,80));
END CAR_INFO_DELETE;
/
```

The "Script Output" tab at the bottom shows the message: "Procedure CAR\_INFO\_DELETE compiled".

**Check table contents.  
Execute procedure.  
Recheck table contents.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following PL/SQL script:

```
SELECT * FROM CAR_INFO;
EXECUTE CAR_INFO_DELETE (90001);
SELECT * FROM CAR_INFO;
```

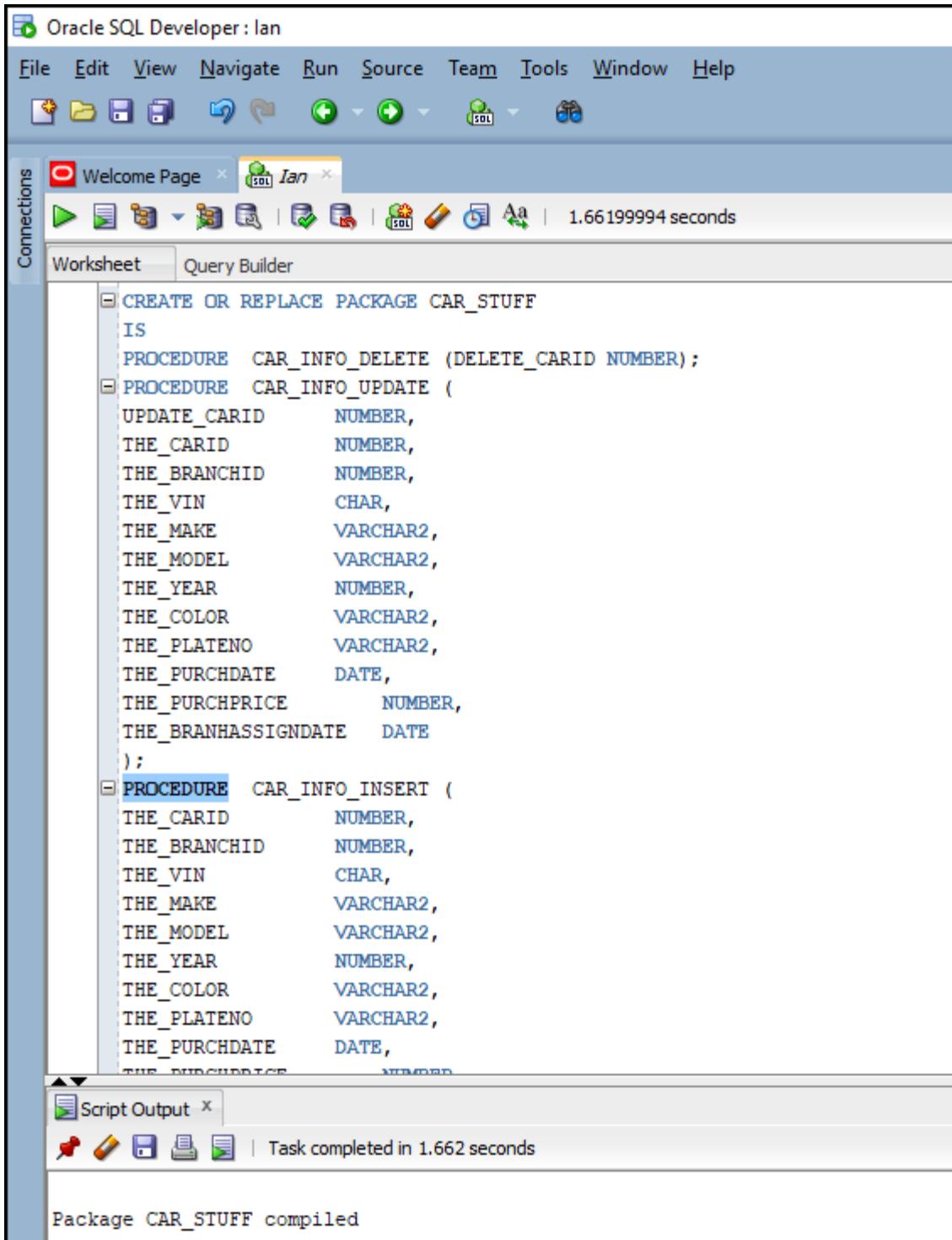
The 'Script Output' tab shows the results of the execution. It first displays the table structure and data before the delete operation:

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90001	1001	5HHEERH7H3398NDU	HONDA	ACCORD	2017	WHITE	ABC1234	03-MAY-17	23500	05-MAY-17
90002	1001	BR77EISLCPPW9NFJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1004	BYUU298EKLOSNJW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17
90005	1002	SABUU098VVL0SNJW	NISSAN	SENTRA	2020	TAN	MAG237	18-JUN-17	22700	18-JUN-17

Following this, the output shows the message 'PL/SQL procedure successfully completed.' and the table structure and data again, identical to the first display:

CARID	BRANCHID	VIN	MAKE	MODEL	YEAR	COLOR	PLATENO	PURCHASEDATE	PURCHASEPRICE	BRANCHASSI
90002	1001	BR77EISLCPPW9NFJ	HONDA	ACCORD	2017	WHITE	DEF1234	03-MAY-17	23500	05-MAY-17
90003	1000	BJJJU298EKLOSNJW	HONDA	ACCORD	2017	BLUE	HJU234	03-MAY-17	23500	05-MAY-17
90004	1004	BYUU298EKLOSNJW	TOYOTA	COROLLA	2019	TAN	HGN234	14-MAY-17	24000	15-MAY-17
90005	1002	SABUU098VVL0SNJW	NISSAN	SENTRA	2020	TAN	MAG237	18-JUN-17	22700	18-JUN-17

## Create package



The screenshot shows the Oracle SQL Developer interface with a connection named 'Jan'. The 'Worksheet' tab is active, displaying a package creation script. The script defines a package 'CAR\_STUFF' with three procedures: 'CAR\_INFO\_DELETE', 'CAR\_INFO\_UPDATE', and 'CAR\_INFO\_INSERT'. Each procedure takes several parameters of various data types. The 'CAR\_INFO\_UPDATE' and 'CAR\_INFO\_INSERT' procedures also have a 'THE\_PURCHDATE' parameter, which is missing in the 'CAR\_INFO\_DELETE' procedure. The 'Script Output' tab at the bottom shows the message 'Task completed in 1.662 seconds'.

```
CREATE OR REPLACE PACKAGE CAR_STUFF
IS
  PROCEDURE CAR_INFO_DELETE (DELETE_CARID NUMBER);
  PROCEDURE CAR_INFO_UPDATE (
    UPDATE_CARID      NUMBER,
    THE_CARID        NUMBER,
    THE_BRANCHID     NUMBER,
    THE_VIN          CHAR,
    THE_MAKE         VARCHAR2,
    THE_MODEL        VARCHAR2,
    THE_YEAR         NUMBER,
    THE_COLOR        VARCHAR2,
    THE_PLATENO     VARCHAR2,
    THE_PURCHDATE    DATE,
    THE_PURCHPRICE   NUMBER,
    THE_BRANHASSIGNDATE DATE
  );
  PROCEDURE CAR_INFO_INSERT (
    THE_CARID        NUMBER,
    THE_BRANCHID     NUMBER,
    THE_VIN          CHAR,
    THE_MAKE         VARCHAR2,
    THE_MODEL        VARCHAR2,
    THE_YEAR         NUMBER,
    THE_COLOR        VARCHAR2,
    THE_PLATENO     VARCHAR2,
    THE_PURCHDATE    DATE,
    THE_PURCHPRICE   NUMBER
  );
END;
/
```

Script Output x

Task completed in 1.662 seconds

Package CAR\_STUFF compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
PROCEDURE CAR_INFO_UPDATE (
    UPDATE_CARID      NUMBER,
    THE_CARID         NUMBER,
    THE_BRANCHID      NUMBER,
    THE_VIN           CHAR,
    THE_MAKE          VARCHAR2,
    THE_MODEL         VARCHAR2,
    THE_YEAR          NUMBER,
    THE_COLOR         VARCHAR2,
    THE_PLATENO       VARCHAR2,
    THE_PURCHDATE     DATE,
    THE_PURCHPRICE    NUMBER,
    THE_BRANHASSIGNDATE DATE
);
PROCEDURE CAR_INFO_INSERT (
    THE_CARID         NUMBER,
    THE_BRANCHID      NUMBER,
    THE_VIN           CHAR,
    THE_MAKE          VARCHAR2,
    THE_MODEL         VARCHAR2,
    THE_YEAR          NUMBER,
    THE_COLOR         VARCHAR2,
    THE_PLATENO       VARCHAR2,
    THE_PURCHDATE     DATE,
    THE_PURCHPRICE    NUMBER,
    THE_BRANHASSIGNDATE DATE
);
END CAR_STUFF;
```

Script Output Task completed in 1.662 seconds

Package CAR\_STUFF compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
CREATE OR REPLACE PACKAGE BODY CAR_STUFF
IS
PROCEDURE CAR_INFO_DELETE (DELETE_CARID NUMBER)
IS
BEGIN      -- Execution
DELETE FROM CAR_INFO WHERE CARID = DELETE_CARID;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('No data found');
DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
SUBSTR(SQLERRM,1,80));
END CAR_INFO_DELETE;
PROCEDURE CAR_INFO_UPDATE (
UPDATE_CARID      NUMBER,
THE_CARID        NUMBER,
THE_BRANCHID     NUMBER,
THE_VIN          CHAR,
THE_MAKE         VARCHAR2,
THE_MODEL        VARCHAR2,
THE_YEAR         NUMBER,
THE_COLOR        VARCHAR2,
THE_PLATENO      VARCHAR2,
THE_PURCHASEDATE DATE,
THE_PURCHASEPRICE NUMBER,
THE_BRANCHASSIGNDATE DATE
)
```

Script Output Task completed in 0.448 seconds

Package Body CAR\_STUFF compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian

Worksheet Query Builder

```
BEGIN -- Execution
UPDATE CAR_INFO SET CARID = THE_CARID WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET BRANCHID = THE_BRANCHID WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET VIN = THE_VIN WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET MAKE = THE_MAKE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET MODEL = THE_MODEL WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET YEAR = THE_YEAR WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET COLOR = THE_COLOR WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PLATENO = THE_PLATENO WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PURCHASEDATE = THE_PURCHASEDATE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET PURCHASEPRICE = THE_PURCHASEPRICE WHERE CARID = UPDATE_CARID;
UPDATE CAR_INFO SET BRANCHASSIGNDATE = THE_BRANCHASSIGNDATE WHERE CARID = UPDATE_CARID;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('No data found');
        DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
END CAR_INFO_UPDATE;

```

PROCEDURE CAR\_INFO\_INSERT (

THE_CARID	NUMBER,
THE_BRANCHID	NUMBER,
THE_VIN	CHAR,
THE_MAKE	VARCHAR2,
THE_MODEL	VARCHAR2,
THE_YEAR	NUMBER,

Script Output | Task completed in 0.448 seconds

Package Body CAR\_STUFF compiled

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 0.44800001 seconds

Worksheet Query Builder

```
THE_BRANCHID NUMBER,
THE_VIN CHAR,
THE_MAKE VARCHAR2,
THE_MODEL VARCHAR2,
THE_YEAR NUMBER,
THE_COLOR VARCHAR2,
THE_PLATENO VARCHAR2,
THE_PURCHASEDATE DATE,
THE_PURCHASEPRICE NUMBER,
THE_BRANHASSIGNDATE DATE
)
IS
BEGIN -- Execution
  INSERT ALL
    INTO CAR_INFO
      (CARID, BRANCHID, VIN, MAKE, MODEL, YEAR, COLOR, PLATENO, PURCHASEDATE, PURCHASEPRICE, BRANHASSIGNDATE)
    VALUES (THE_CARID, THE_BRANCHID, THE_VIN, THE_MAKE, THE_MODEL, THE_YEAR, THE_COLOR, THE_PLATENO, THE_PURCHASEDATE,
            THE_PURCHASEPRICE, THE_BRANHASSIGNDATE)
    SELECT * FROM dual;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('No data found');
      DBMS_OUTPUT.PUT_LINE('Error '||SQLCODE ||
                           SUBSTR(SQLERRM,1,80));
  END CAR_INFO_INSERT;
END CAR_STUFF;
```

Script Output Task completed in 0.448 seconds

Package Body CAR\_STUFF compiled

^^^^^^^^^^^^^^^^^^^^

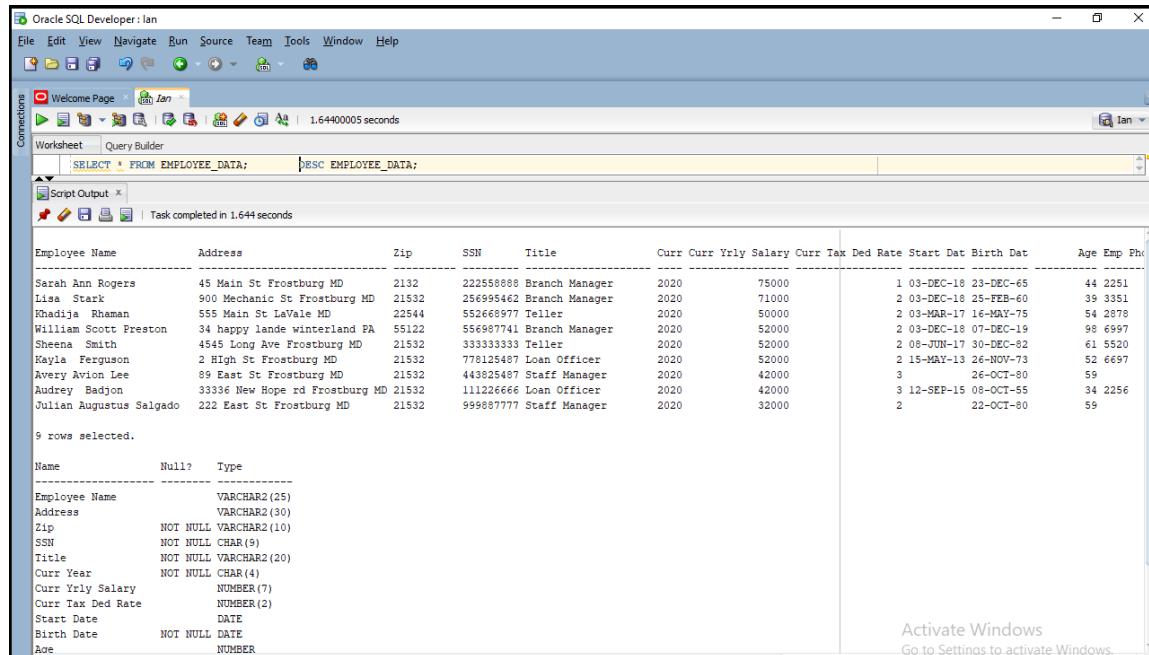
## CHAPTER 5A Starts here

^^^^^^^^^^^^^^^^^^^^

### Create the following triggers:

A. Any deletion from employee file, trigger to write OLD attributes into an **Employee\_History** file.

Check table contents.

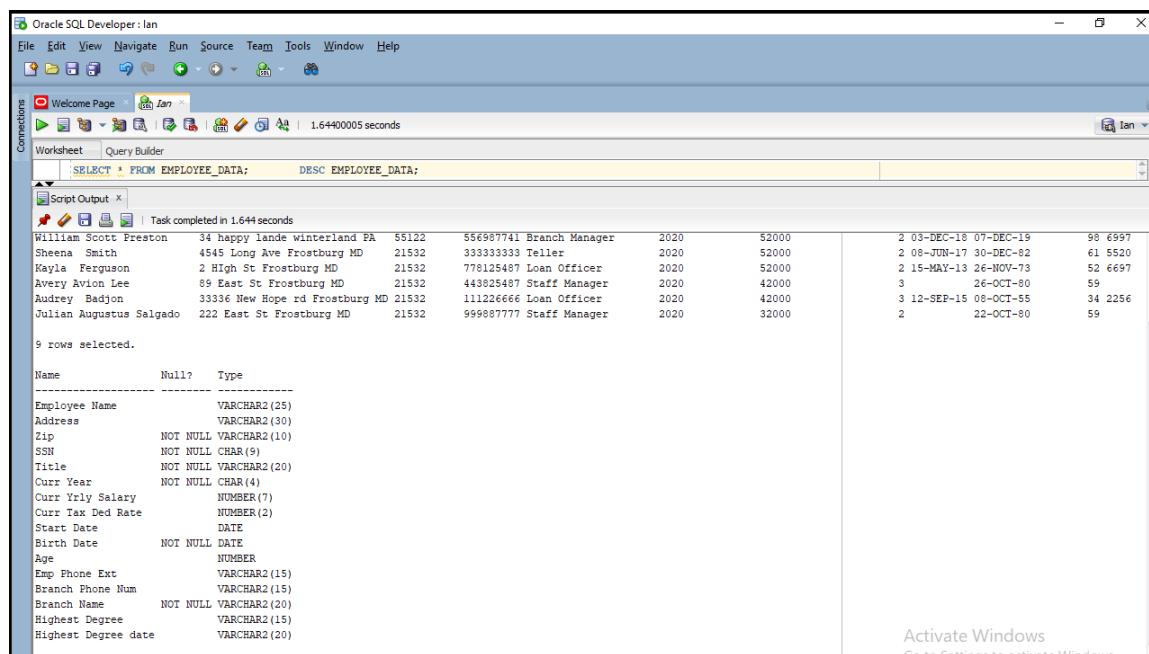


```
Employee Name      Address      Zip      SSN      Title      Curr Yrly Salary      Curr Tax Ded Rate      Start Date      Birth Date      Age      Emp Phc
-----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----
Sarah Ann Rogers  45 Main St Frostburg MD 21532  222558888 Branch Manager  2020      75000      1 03-DEC-18 23-DEC-65  44 2251
Lisa Stark        900 Mechanic St Frostburg MD 21532  25695462 Branch Manager  2020      71000      2 03-DEC-18 25-FEB-60  39 3351
Khadija Rhaman   555 Main St LaVale MD 22544  552668977 Teller      2020      50000      2 03-MAR-17 16-MAY-75  54 2878
William Scott Preston  34 happy lande Winterland PA 55122  556987741 Branch Manager  2020      52000      2 03-DEC-18 07-DEC-19  98 6997
Sheena Smith     4545 Long Ave Frostburg MD 21532  333333333 Teller      2020      52000      2 08-JUN-17 30-DEC-62  61 5520
Kayla Ferguson   2 High St Frostburg MD 21532  778125487 Loan Officer  2020      52000      2 15-MAY-13 26-NOV-73  52 6697
Avery Avion Lee  89 East St Frostburg MD 21532  443825487 Staff Manager  2020      42000      3 26-OCT-80          59
Audrey Badjon   33336 New Hope rd Frostburg MD 21532  111226666 Loan Officer  2020      42000      3 12-SEP-15 08-OCT-55  34 2256
Julian Augustus Salgado  222 East St Frostburg MD 21532  999887777 Staff Manager  2020      32000      2 22-OCT-80          59

9 rows selected.

Name      Null?      Type
-----      -----      -----
Employee Name      VARCHAR2(25)
Address      VARCHAR2(30)
Zip      NOT NULL VARCHAR2(10)
SSN      NOT NULL CHAR(9)
Title      NOT NULL VARCHAR2(20)
Curr Year      NOT NULL CHAR(4)
Curr Yrly Salary      NUMBER(7)
Curr Tax Ded Rate      NUMBER(2)
Start Date      DATE
Birth Date      NOT NULL DATE
Age      NUMBER(4)

Activate Windows
Go to Settings to activate Windows.
```



```
Employee Name      Address      Zip      SSN      Title      Curr Yrly Salary      Curr Tax Ded Rate      Start Date      Birth Date      Age      Emp Phc
-----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----
William Scott Preston  34 happy lande Winterland PA 55122  556987741 Branch Manager  2020      52000      2 03-DEC-18 07-DEC-19  98 6997
Sheena Smith     4545 Long Ave Frostburg MD 21532  333333333 Teller      2020      52000      2 08-JUN-17 30-DEC-62  61 5520
Kayla Ferguson   2 High St Frostburg MD 21532  778125487 Loan Officer  2020      52000      2 15-MAY-13 26-NOV-73  52 6697
Avery Avion Lee  89 East St Frostburg MD 21532  443825487 Staff Manager  2020      42000      3 26-OCT-80          59
Audrey Badjon   33336 New Hope rd Frostburg MD 21532  111226666 Loan Officer  2020      42000      3 12-SEP-15 08-OCT-55  34 2256
Julian Augustus Salgado  222 East St Frostburg MD 21532  999887777 Staff Manager  2020      32000      2 22-OCT-80          59

9 rows selected.

Name      Null?      Type
-----      -----      -----
Employee Name      VARCHAR2(25)
Address      VARCHAR2(30)
Zip      NOT NULL VARCHAR2(10)
SSN      NOT NULL CHAR(9)
Title      NOT NULL VARCHAR2(20)
Curr Year      NOT NULL CHAR(4)
Curr Yrly Salary      NUMBER(7)
Curr Tax Ded Rate      NUMBER(2)
Start Date      DATE
Birth Date      NOT NULL DATE
Age      NUMBER(4)
Emp Phone Ext      VARCHAR2(15)
Branch Phone Num      VARCHAR2(15)
Branch Name      NOT NULL VARCHAR2(20)
Highest Degree      VARCHAR2(15)
Highest Degree date      VARCHAR2(20)

Activate Windows
Go to Settings to activate Windows.
```

**Create new table.**  
**Check new table attribute data types.**  
**Check new table contents.**

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Ian 1.472 seconds

Worksheet Query Builder

```
CREATE TABLE ALL_EMPLOYEES AS SELECT * FROM EMPLOYEE_DATA;
DESC ALL_EMPLOYEES;
SELECT * FROM ALL_EMPLOYEES;
```

Script Output x | Task completed in 1.472 seconds

Table ALL\_EMPLOYEES created.

Name Null? Type

---

Name	Address	Zip	SSN	Title	Curr Yrly Salary	Curr Tax Ded Rate	Start Dat	Birth Dat	Age	Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222558888	Branch Manager	2020	75000	1/03/DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256995462	Branch Manager	2020	71000	2/03/DEC-18	25-FEB-60VS	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000	2/03/MAR-17	16-MAY-75	54	2878

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Ian 1.472 seconds

Worksheet Query Builder

```
CREATE TABLE ALL_EMPLOYEES AS SELECT * FROM EMPLOYEE_DATA;
DESC ALL_EMPLOYEES;
SELECT * FROM ALL_EMPLOYEES;
```

Script Output x | Task completed in 1.472 seconds

Table ALL\_EMPLOYEES created.

Name Null? Type

---

Name	Address	Zip	SSN	Title	Curr Yrly Salary	Curr Tax Ded Rate	Start Dat	Birth Dat	Age	Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222558888	Branch Manager	2020	75000	1/03/DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256995462	Branch Manager	2020	71000	2/03/DEC-18	25-FEB-60	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000	2/03/MAR-17	16-MAY-75	54	2878
William Scott Preston	34 happy lande Winterland PA	55122	556987741	Branch Manager	2020	52000	2/03/DEC-18	07-DEC-19	98	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000	2/08/JUN-17	30-DEC-82	61	5520
Kaylin Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000	2/15/MAY-13	26-NOV-73	52	6697
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020	42000	3	26-OCT-80	59	
Audrey Badjon	3336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020	42000	3/12-SEP-15	08-OCT-55	34	2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020	32000	2	22-OCT-80	59	

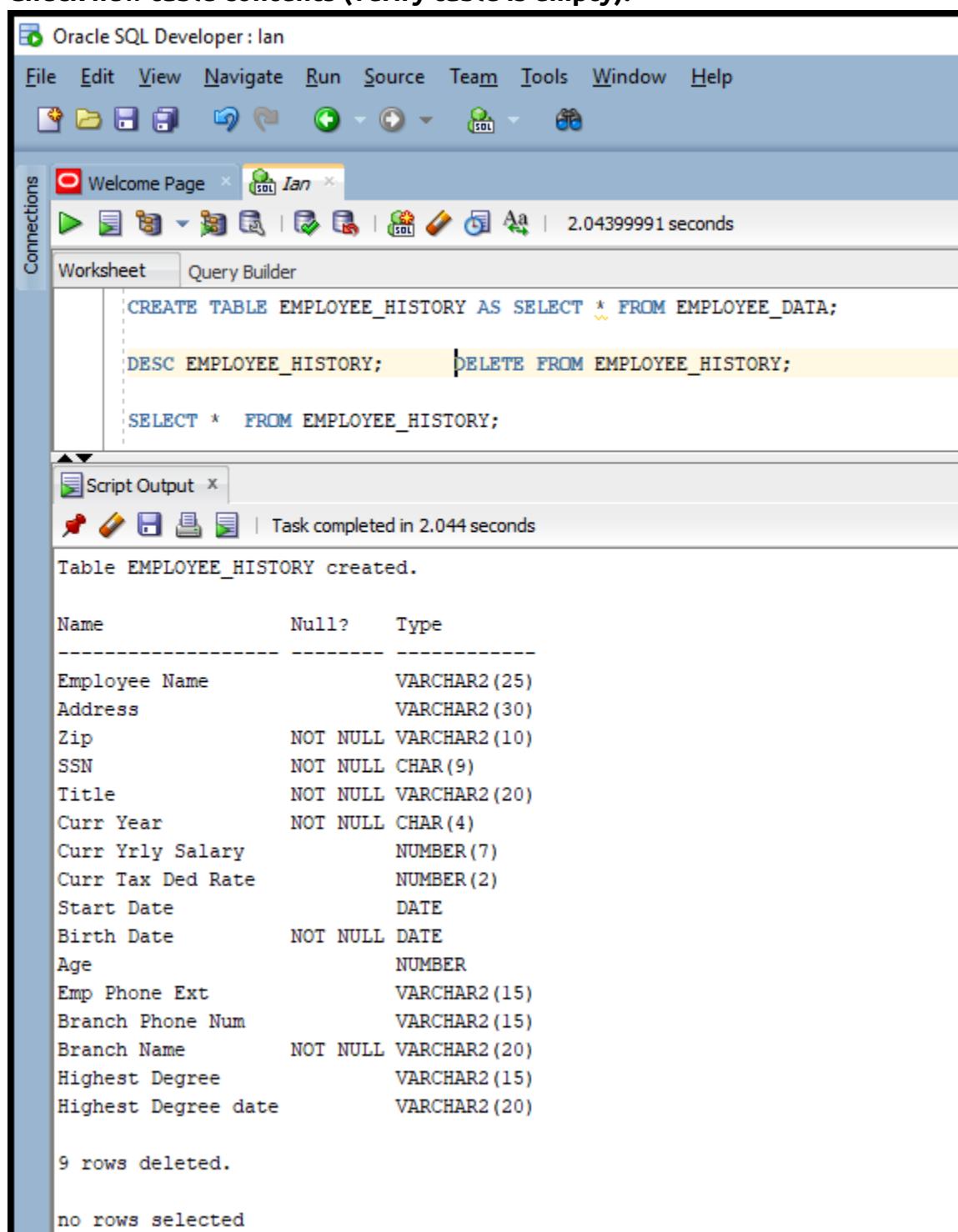
9 rows selected.

Activate Windows  
Go to Settings to activate Windows.

**Create new table.**

**Check new table attribute data types.**

**Check new table contents (verify table is empty).**



The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE TABLE EMPLOYEE_HISTORY AS SELECT * FROM EMPLOYEE_DATA;  
DESC EMPLOYEE_HISTORY;      DELETE FROM EMPLOYEE_HISTORY;  
SELECT * FROM EMPLOYEE_HISTORY;
```

The 'Script Output' tab shows the results of the table creation:

```
Table EMPLOYEE_HISTORY created.
```

A detailed description of the table attributes is provided:

Name	Null?	Type
Employee Name		VARCHAR2 (25)
Address		VARCHAR2 (30)
Zip	NOT NULL	VARCHAR2 (10)
SSN	NOT NULL	CHAR (9)
Title	NOT NULL	VARCHAR2 (20)
Curr Year	NOT NULL	CHAR (4)
Curr Yrly Salary		NUMBER (7)
Curr Tax Ded Rate		NUMBER (2)
Start Date		DATE
Birth Date	NOT NULL	DATE
Age		NUMBER
Emp Phone Ext		VARCHAR2 (15)
Branch Phone Num		VARCHAR2 (15)
Branch Name	NOT NULL	VARCHAR2 (20)
Highest Degree		VARCHAR2 (15)
Highest Degree date		VARCHAR2 (20)

Below the table description, the output shows:

```
9 rows deleted.  
no rows selected
```

## Insert test employee entry to test trigger.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.56199998 seconds

Worksheet Query Builder

```

SELECT * FROM ALL_EMPLOYEES;
INSERT ALL
INTO ALL_EMPLOYEES
  ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
  "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
  "Highest Degree date")
VALUES ('Test Employee', '1234 Test Lane Frostburg MD', '21327', '222999988', 'Any Position', '2021',
50001, 3, '17-OCT-15', '25-FEB-60', 39, '2991', '4109313382', 'Windsor Main', 'Mast', '11-MAY-88')
SELECT * FROM dual;
SELECT * FROM ALL_EMPLOYEES;

```

Script Output x Task completed in 0.562 seconds

Employee Name	Address	Zip	SSN	Title	Curr Year	Curr Yrly Salary	Curr Tax Ded Rate	Start Date	Birth Date	Age	Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222558888	Branch Manager	2020	75000		1-03-DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256959462	Branch Manager	2020	71000		2-03-DEC-18	25-FEB-60	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000		2-03-MAR-17	16-MAY-75	54	2878
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020	52000		2-03-DEC-18	07-DEC-19	96	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000		2-08-JUN-17	30-DEC-82	61	5520
Mayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000		2-15-MAY-13	26-NOV-73	52	6697
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020	42000		3	26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020	42000		3-12-SEP-15	08-OCT-55	34	2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020	32000		2	22-OCT-80	59	

9 rows selected.

1 row inserted.

Activate Windows  
Go to Settings to activate Windows.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Ian 0.56199998 seconds

Worksheet Query Builder

```

SELECT * FROM ALL_EMPLOYEES;
INSERT ALL
INTO ALL_EMPLOYEES
  ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
  "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
  "Highest Degree date")
VALUES ('Test Employee', '1234 Test Lane Frostburg MD', '21327', '222999988', 'Any Position', '2021',
50001, 3, '17-OCT-15', '25-FEB-60', 39, '2991', '4109313382', 'Windsor Main', 'Mast', '11-MAY-88')
SELECT * FROM dual;
SELECT * FROM ALL_EMPLOYEES;

```

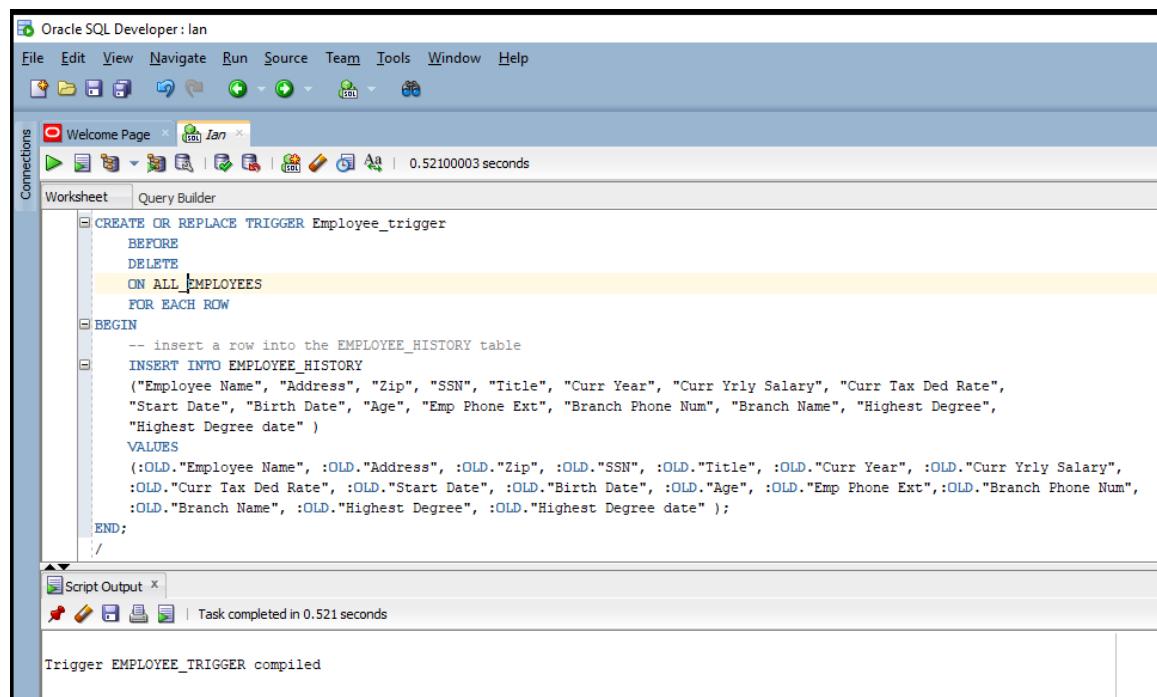
Script Output x Task completed in 0.562 seconds

Employee Name	Address	Zip	SSN	Title	Curr Year	Curr Yrly Salary	Curr Tax Ded Rate	Start Date	Birth Date	Age	Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222558888	Branch Manager	2020	75000		1-03-DEC-18	23-DEC-65	44	2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256959462	Branch Manager	2020	71000		2-03-DEC-18	25-FEB-60	39	3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020	50000		2-03-MAR-17	16-MAY-75	54	2878
William Scott Preston	34 happy lande winterland PA	55122	556987741	Branch Manager	2020	52000		2-03-DEC-18	07-DEC-19	96	6997
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020	52000		2-08-JUN-17	30-DEC-82	61	5520
Mayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020	52000		2-15-MAY-13	26-NOV-73	52	6697
Avery Avion Lee	89 East St Frostburg MD	21532	443825487	Staff Manager	2020	42000		3	26-OCT-80	59	
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020	42000		3-12-SEP-15	08-OCT-55	34	2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020	32000		2	22-OCT-80	59	
Test Employee	1234 Test Lane Frostburg MD	21327	222999988	Any Position	2021	50001		3-17-OCT-15	25-FEB-60	39	2991

10 rows selected.

Activate Windows  
Go to Settings to activate Windows.

## Create trigger.



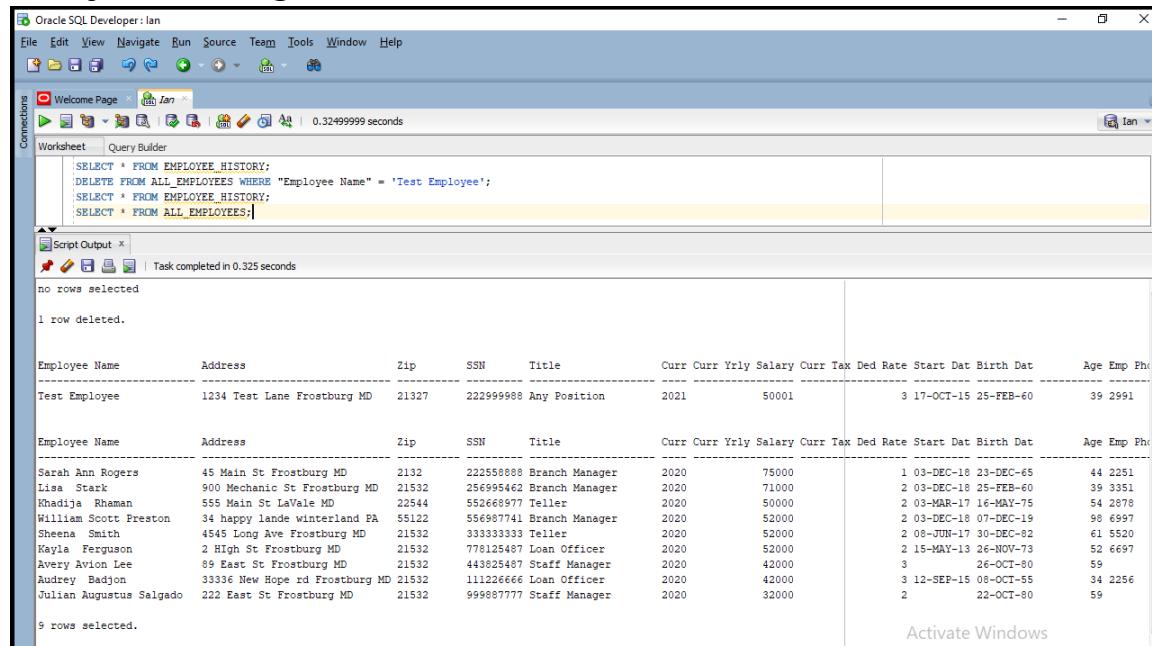
The screenshot shows the Oracle SQL Developer interface with a trigger creation script in the Worksheet tab. The trigger, named 'Employee\_trigger', is defined as follows:

```
CREATE OR REPLACE TRIGGER Employee_trigger
  BEFORE
  DELETE
  ON ALL EMPLOYEES
  FOR EACH ROW
BEGIN
  -- insert a row into the EMPLOYEE_HISTORY table
  INSERT INTO EMPLOYEE_HISTORY
  ("Employee Name", "Address", "Zip", "SSN", "Title", "Curr Year", "Curr Yrly Salary", "Curr Tax Ded Rate",
  "Start Date", "Birth Date", "Age", "Emp Phone Ext", "Branch Phone Num", "Branch Name", "Highest Degree",
  "Highest Degree date" )
  VALUES
  (:OLD."Employee Name", :OLD."Address", :OLD."Zip", :OLD."SSN", :OLD."Title", :OLD."Curr Year", :OLD."Curr Yrly Salary",
  :OLD."Curr Tax Ded Rate", :OLD."Start Date", :OLD."Birth Date", :OLD."Age", :OLD."Emp Phone Ext", :OLD."Branch Phone Num",
  :OLD."Branch Name", :OLD."Highest Degree", :OLD."Highest Degree date" );
END;
/
```

The trigger is successfully compiled, as indicated in the Script Output tab:

```
Trigger EMPLOYEE_TRIGGER compiled
```

**Test trigger.  
Verify table changes and table contents.**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Ian

Worksheet Query Builder

```
SELECT * FROM EMPLOYEE_HISTORY;
DELETE FROM ALL_EMPLOYEES WHERE "Employee Name" = 'Test Employee';
SELECT * FROM EMPLOYEE_HISTORY;
SELECT * FROM ALL_EMPLOYEES;
```

Script Output | Task completed in 0.325 seconds

no rows selected

1 row deleted.

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Test Employee	1234 Test Lane Frostburg MD	21327	222999988	Any Position	2021 50001		3 17-OCT-15 25-FEB-60	39 2991

Employee Name	Address	Zip	SSN	Title	Curr Curr Yrly Salary	Curr Tax Ded Rate	Start Dat Birth Dat	Age Emp Ph
Sarah Ann Rogers	45 Main St Frostburg MD	2132	222558888	Branch Manager	2020 75000		1 03-DEC-18 23-DEC-65	44 2251
Lisa Stark	900 Mechanic St Frostburg MD	21532	256995462	Branch Manager	2020 71000		2 03-DEC-18 25-FEB-60	39 3351
Khadija Rhaman	555 Main St LaVale MD	22544	552668977	Teller	2020 50000		2 03-MAR-17 16-MAY-75	54 2878
William Scott Preston	34 happy lande winterland PA	55122	556887741	Branch Manager	2020 52000		2 03-DEC-18 07-DEC-19	58 6987
Sheena Smith	4545 Long Ave Frostburg MD	21532	333333333	Teller	2020 52000		2 08-JUN-17 30-DEC-62	61 5520
Kayla Ferguson	2 High St Frostburg MD	21532	778125487	Loan Officer	2020 52000		2 15-MAY-13 26-NOV-73	52 6697
Avery Avion Lee	89 East St Frostburg MD	21532	443285487	Staff Manager	2020 42000		3 26-OCT-80	59
Audrey Badjon	33336 New Hope rd Frostburg MD	21532	111226666	Loan Officer	2020 42000		3 12-SEP-15 08-OCT-55	34 2256
Julian Augustus Salgado	222 East St Frostburg MD	21532	999887777	Staff Manager	2020 32000		2 22-OCT-80	59

9 rows selected.

Activate Windows

^^^^^^^^^^^^^^^^^^^^^^^^

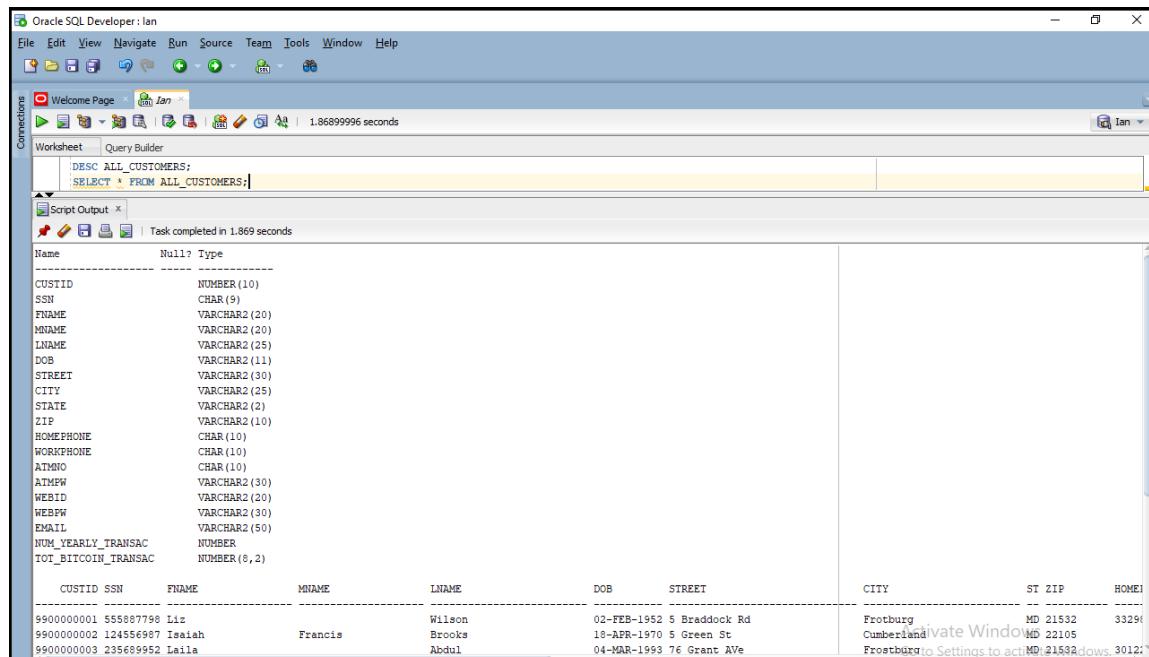
## CHAPTER 5B Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

B. Any modification to the customer accounts, write who, date, and the nature of (OLD and NEW) modification into a **Cust\_Mod\_Log** file.

**Check table contents.**

**Check table attribute data types.**



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

1.86899996 seconds

Worksheet

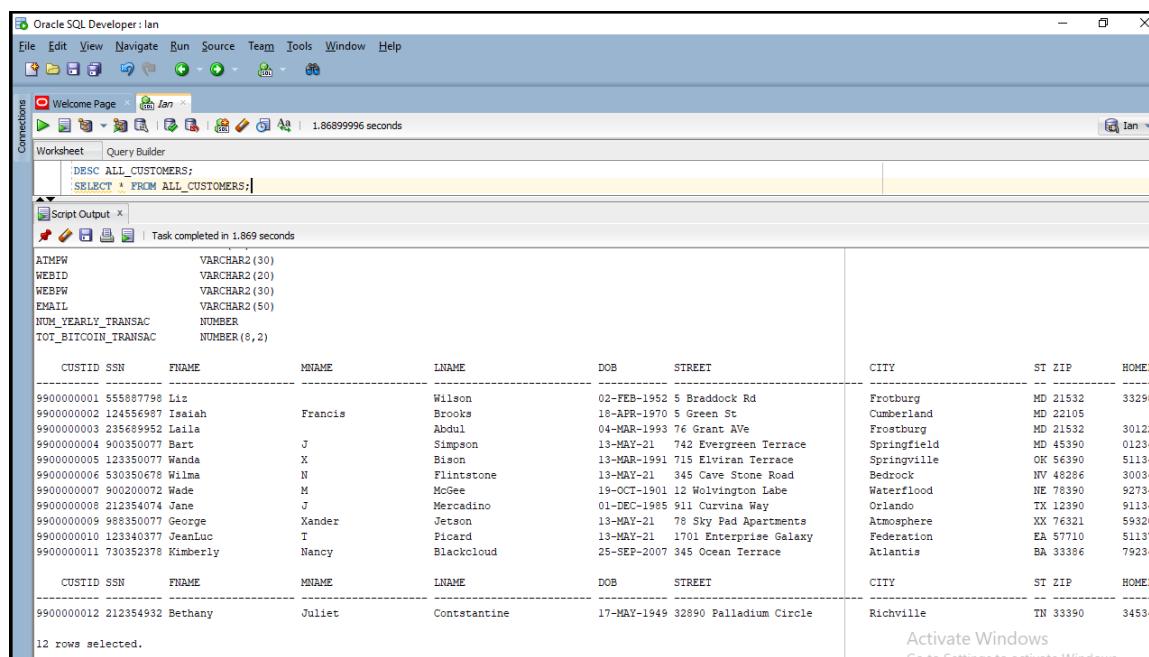
```
DESC ALL_CUSTOMERS;
SELECT * FROM ALL_CUSTOMERS;
```

Script Output

Task completed in 1.869 seconds

Name	Null?	Type
CUSTID		NUMBER(10)
SSN		CHAR(9)
FNAME		VARCHAR2(20)
MNAME		VARCHAR2(20)
LNAME		VARCHAR2(25)
DOB		VARCHAR2(11)
STREET		VARCHAR2(30)
CITY		VARCHAR2(25)
STATE		VARCHAR2(2)
ZIP		VARCHAR2(10)
HOMEPHONE		CHAR(10)
WORKPHONE		CHAR(10)
ATMNO		CHAR(10)
ATMPW		VARCHAR2(30)
WEBID		VARCHAR2(20)
WEBPW		VARCHAR2(30)
EMAIL		VARCHAR2(50)
NUM_YEARLY_TRANSAC		NUMBER
TOT_BITCOIN_TRANSAC		NUMBER(8,2)

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frostburg	MD	21532	3329
9900000002	124556987	Izaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	3012



Oracle SQL Developer: Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Script Output

1.86899996 seconds

Worksheet

```
DESC ALL_CUSTOMERS;
SELECT * FROM ALL_CUSTOMERS;
```

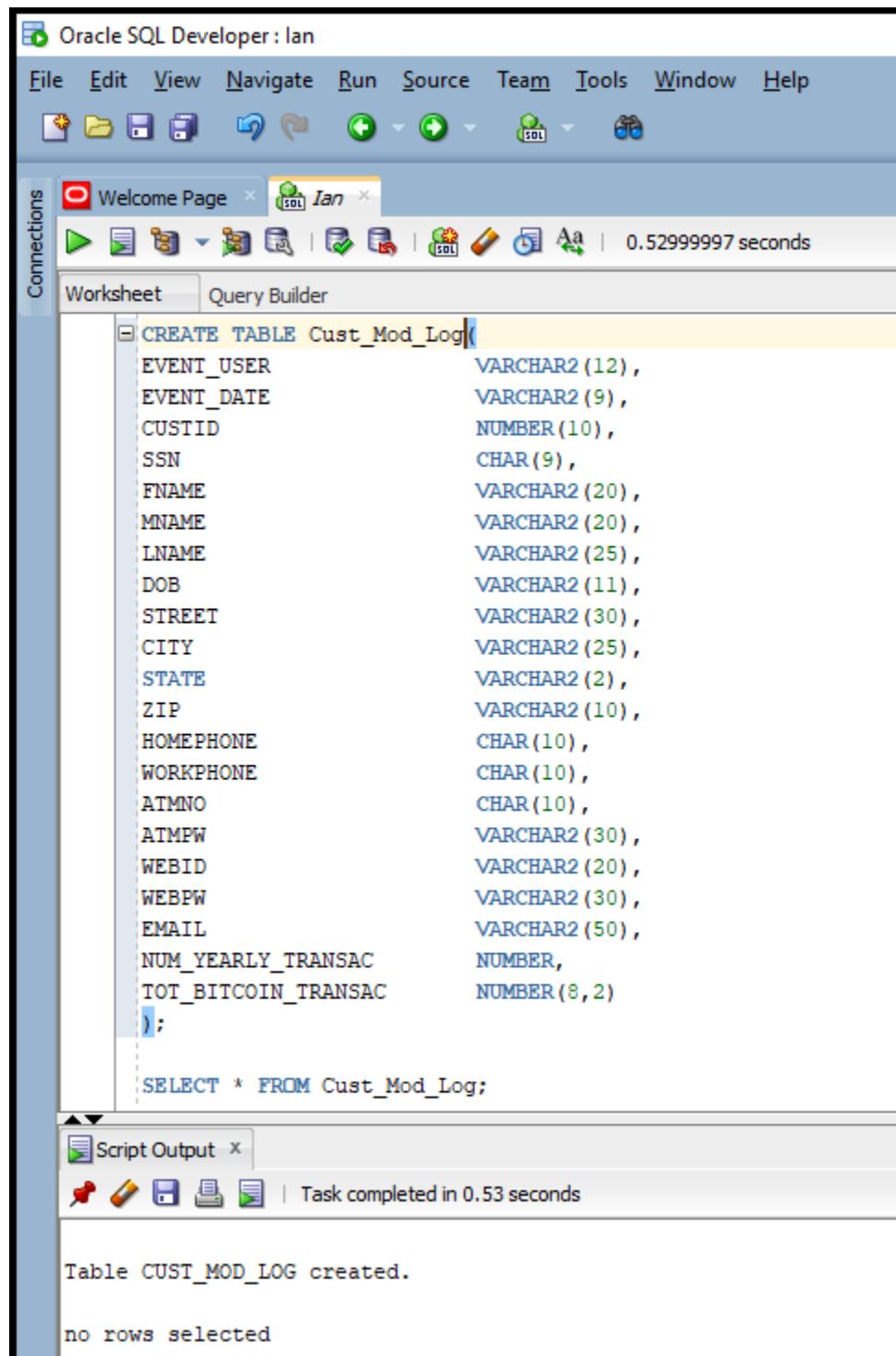
Script Output

Task completed in 1.869 seconds

ATMPW	VARCHAR2(30)									
WEBID	VARCHAR2(20)									
WEBPW	VARCHAR2(30)									
EMAIL	VARCHAR2(50)									
NUM_YEARLY_TRANSAC	NUMBER									
TOT_BITCOIN_TRANSAC	NUMBER(8,2)									
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frostburg	MD	21532	3329
9900000002	124556987	Izaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	3012
9900000004	900350077	Bert	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	0123
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elvira Terrace	Springville	OK	56390	5113
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	48286	3003
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Lake	Waterflood	NE	78390	9273
9900000008	212354074	Jane	J	Mercadino	01-DEC-1965	911 Curvina Way	Orlando	TX	12390	9113
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	5932
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	5113
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	7923
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN	33390	3453

**Create new table contents.**

**Check table contents.**



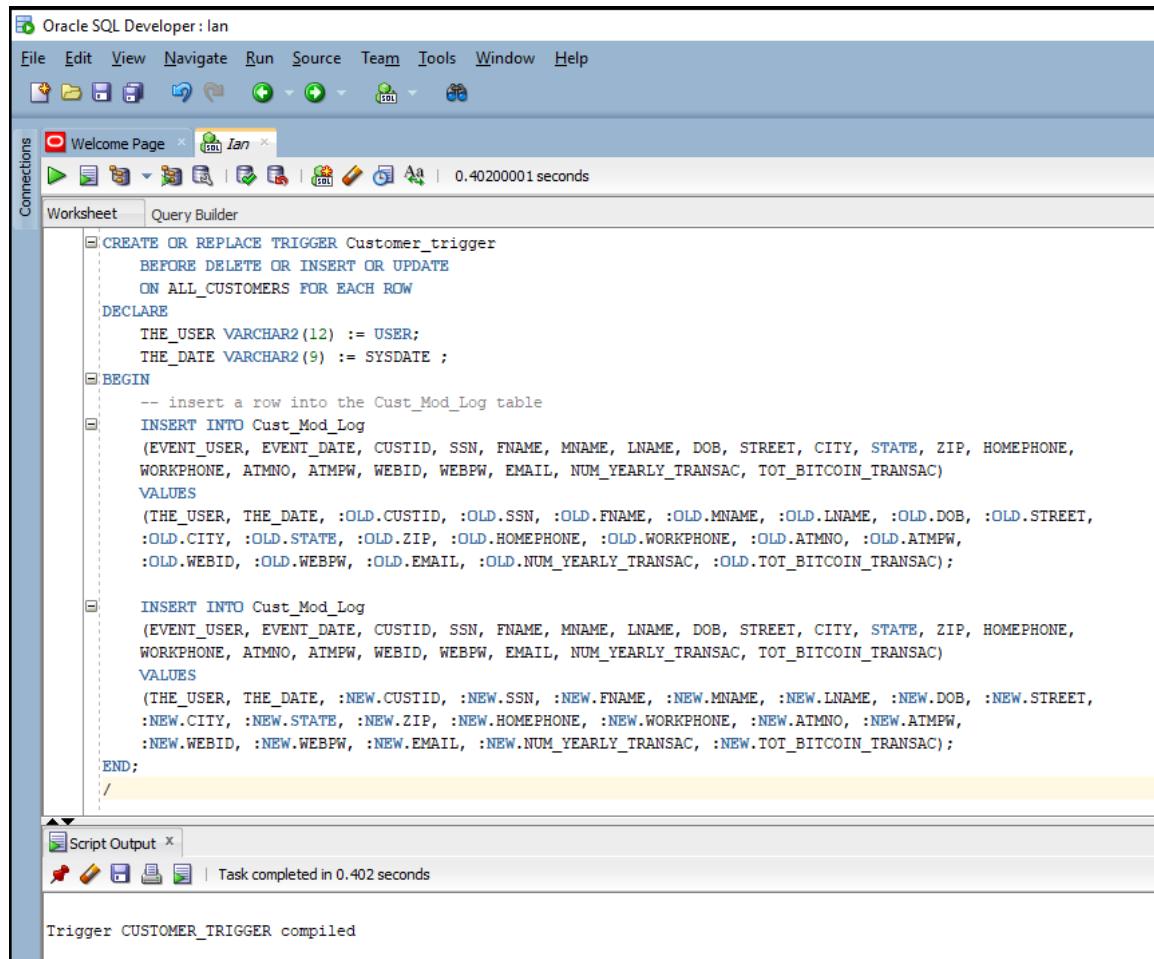
The screenshot shows the Oracle SQL Developer interface with a connection named 'Ian'. The 'Worksheet' tab is active, displaying SQL code for creating a table and performing a select operation. The 'Script Output' tab at the bottom shows the execution results.

```
CREATE TABLE Cust_Mod_Log(
  EVENT_USER          VARCHAR2(12),
  EVENT_DATE          VARCHAR2(9),
  CUSTID              NUMBER(10),
  SSN                 CHAR(9),
  FNAME               VARCHAR2(20),
  MNAME               VARCHAR2(20),
  LNAME               VARCHAR2(25),
  DOB                 VARCHAR2(11),
  STREET              VARCHAR2(30),
  CITY                VARCHAR2(25),
  STATE               VARCHAR2(2),
  ZIP                 VARCHAR2(10),
  HOMEPHONE           CHAR(10),
  WORKPHONE           CHAR(10),
  ATMNO               CHAR(10),
  ATMPW               VARCHAR2(30),
  WEBID               VARCHAR2(20),
  WEBPW               VARCHAR2(30),
  EMAIL               VARCHAR2(50),
  NUM_YEARLY_TRANSAC NUMBER,
  TOT_BITCOIN_TRANSAC NUMBER(8,2)
);

SELECT * FROM Cust_Mod_Log;
```

Script Output: Table CUST\_MOD\_LOG created.  
no rows selected

## Create trigger.



The screenshot shows the Oracle SQL Developer interface with a trigger creation script in the Worksheet tab. The trigger, named 'Customer\_trigger', is defined to fire BEFORE DELETE OR INSERT OR UPDATE ON ALL\_CUSTOMERS FOR EACH ROW. It uses local variables THE\_USER and THE\_DATE, and declares OLD and NEW record types for logging changes to the Cust\_Mod\_Log table. The trigger body includes two INSERT INTO statements for the log table, one for each row operation (DELETE and INSERT). The trigger is successfully compiled, as indicated by the message in the Script Output tab.

```
CREATE OR REPLACE TRIGGER Customer_trigger
  BEFORE DELETE OR INSERT OR UPDATE
  ON ALL_CUSTOMERS FOR EACH ROW
DECLARE
  THE_USER VARCHAR2(12) := USER;
  THE_DATE VARCHAR2(9) := SYSDATE ;
BEGIN
  -- insert a row into the Cust_Mod_Log table
  INSERT INTO Cust_Mod_Log
  (EVENT_USER, EVENT_DATE, CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (THE_USER, THE_DATE, :OLD.CUSTID, :OLD.SSN, :OLD.FNAME, :OLD.MNAME, :OLD.LNAME, :OLD.DOB, :OLD.STREET,
  :OLD.CITY, :OLD.STATE, :OLD.ZIP, :OLD.HOMEPHONE, :OLD.WORKPHONE, :OLD.ATMNO, :OLD.ATMPW,
  :OLD.WEBID, :OLD.WEBPW, :OLD.EMAIL, :OLD.NUM_YEARLY_TRANSAC, :OLD.TOT_BITCOIN_TRANSAC);

  INSERT INTO Cust_Mod_Log
  (EVENT_USER, EVENT_DATE, CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
  WORKPHONE, ATMNO, ATMPW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
  VALUES
  (THE_USER, THE_DATE, :NEW.CUSTID, :NEW.SSN, :NEW.FNAME, :NEW.MNAME, :NEW.LNAME, :NEW.DOB, :NEW.STREET,
  :NEW.CITY, :NEW.STATE, :NEW.ZIP, :NEW.HOMEPHONE, :NEW.WORKPHONE, :NEW.ATMNO, :NEW.ATMPW,
  :NEW.WEBID, :NEW.WEBPW, :NEW.EMAIL, :NEW.NUM_YEARLY_TRANSAC, :NEW.TOT_BITCOIN_TRANSAC);
END;
/
```

Trigger CUSTOMER\_TRIGGER compiled

**Insert rows to test trigger.  
Verify trigger's changes to tables.**

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```

SELECT * FROM ALL_CUSTOMERS;           SELECT * FROM Cust_Mod_Log;
INSERT ALL
INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATTNNO, ATTNW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES (9999999999, '900359999', 'Test', 'T', 'Testerton', '18-NOV-1999', '999 Tester Terrace', 'Testerfield', 'MD', '45390', '99999990034',
'0134556789', '2123326739', 'testy', 'testo', 'testol23', 'testy_testerton@gmail.com', 9999, 99456)
SELECT 1 FROM dual;
SELECT * FROM Cust_Mod_Log;           SELECT * FROM ALL_CUSTOMERS;
```

Script Output x

Task completed in 2.332 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD	21532	33291
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30121
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	40286	30034
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Lake	Waterflood	NY	78390	92734
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX	12390	91134
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX	76321	59324
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA	57710	51134
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA	33386	79234
CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME
9900000012	212354932	Bethany		Constantine	17-MAY-1949	32890 Palladium Circle	Kichville	TN	33390	34534

12 rows selected.

no rows selected

Activate Windows  
Go to Settings to activate Windows.

Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Worksheet Query Builder

```

SELECT * FROM ALL_CUSTOMERS;           SELECT * FROM Cust_Mod_Log;
INSERT ALL
INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATTNNO, ATTNW, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES (9999999999, '900359999', 'Test', 'T', 'Testerton', '18-NOV-1999', '999 Tester Terrace', 'Testerfield', 'MD', '45390', '99999990034',
'0134556789', '2123326739', 'testy', 'testo', 'testol23', 'testy_testerton@gmail.com', 9999, 99456)
SELECT 1 FROM dual;
SELECT * FROM Cust_Mod_Log;           SELECT * FROM ALL_CUSTOMERS;
```

Script Output x

Task completed in 2.332 seconds

12 rows selected.

no rows selected

1 row inserted.

EVENT_USER	EVENT_DAT	CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY
CS640F2010	03-MAY-21								
CS640F2010	03-MAY-21	999999999999	900359999	Test	T	Testerton	18-NOV-1999	999 Tester Terrace	Testerfield

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST	ZIP	HOME
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD	21532	33291
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD	22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD	21532	30121
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD	45390	01234
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK	56390	51134
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV	40286	30034

Activate Windows  
Go to Settings to activate Windows.

## Update rows to test trigger. Verify trigger's changes to tables.

Oracle SQL Developer: Ian

```

SELECT * FROM Cust_Mod_Log;
UPDATE ALL_CUSTOMERS SET STREET = '109090 Experiment Lane'
WHERE FNAME = 'Test';
SELECT * FROM Cust_Mod_Log;
SELECT * FROM ALL_CUSTOMERS;

```

Script Output x

Task completed in 2.284 seconds

EVENT_USER	EVENT_DAT	CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY
CS640F2010	03-MAY-21								
CS640F2010	03-MAY-21	9999999999	900359999	Test	T	Testerton	18-NOV-1999	999 Tester Terrace	Testerfield

1 row updated.

EVENT_USER	EVENT_DAT	CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY
CS640F2010	03-MAY-21								
CS640F2010	03-MAY-21	9999999999	900359999	Test	T	Testerton	18-NOV-1999	999 Tester Terrace	Testerfield
CS640F2010	03-MAY-21	9999999999	900359999	Test	T	Testerton	18-NOV-1999	999 Tester Terrace	Testerfield
CS640F2010	03-MAY-21	9999999999	900359999	Test	T	Testerton	18-NOV-1999	109090 Experiment Lane	Testerfield

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD 21532	3329
9900000002	124569697	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD 22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD 21532	3012
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD 45390	0123

Oracle SQL Developer: Ian

```

SELECT * FROM ALL_CUSTOMERS;           SELECT * FROM Cust_Mod_Log;
INSERT ALL
INTO ALL_CUSTOMERS (CUSTID, SSN, FNAME, MNAME, LNAME, DOB, STREET, CITY, STATE, ZIP, HOMEPHONE,
WORKPHONE, ATINNO, ATINWF, WEBID, WEBPW, EMAIL, NUM_YEARLY_TRANSAC, TOT_BITCOIN_TRANSAC)
VALUES (9999999999, '900359999', 'Test', 'T', 'Testerton', '18-NOV-1999', '999 Tester Terrace', 'Testerfield', 'MD', '45390', '99999990034',
'0134556789', '2123326739', 'testy', 'testo', 'testo123', 'testy_testerton@gmail.com', 9999, 99456)
SELECT 1 FROM dual;
SELECT * FROM Cust_Mod_Log;           SELECT * FROM ALL_CUSTOMERS;

```

Script Output x

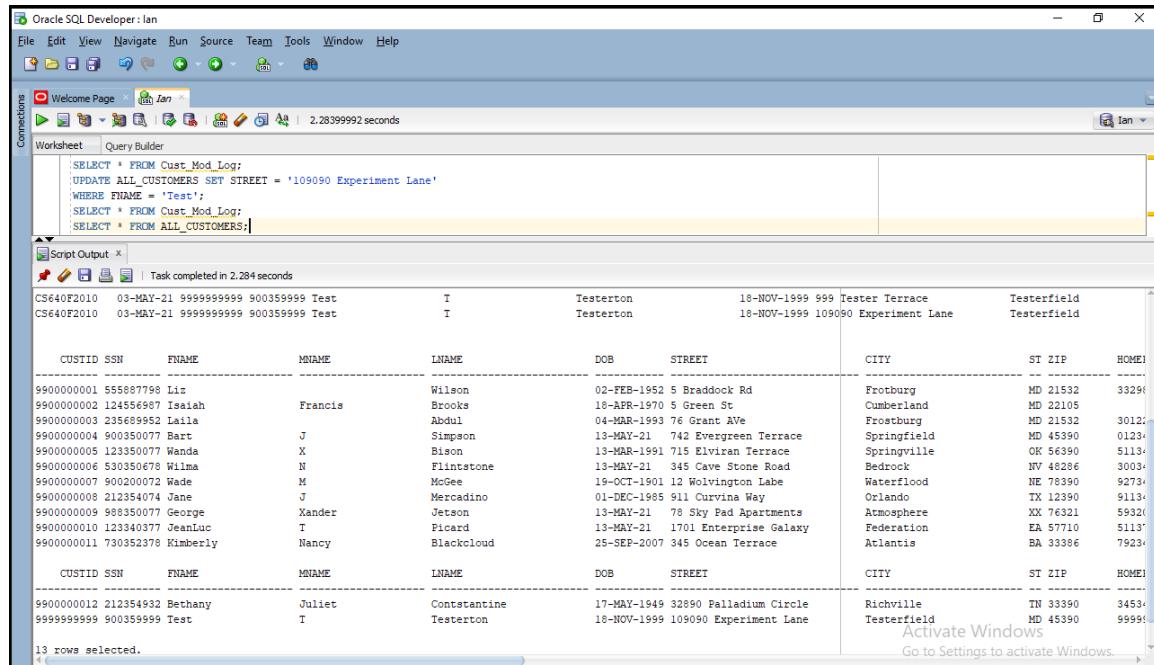
Task completed in 2.332 seconds

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST ZIP	HOMEI
9900000001	555887798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frothburg	MD 21532	3329
9900000002	124569697	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD 22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD 21532	3012
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD 45390	0123
9900000005	123350077	Wanda	X	Bison	13-MAR-1991	715 Elviran Terrace	Springville	OK 56390	5113
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NV 48286	3003
9900000007	900200072	Wade	M	McGee	19-OCT-1901	12 Wolverton Labe	Waterflood	NE 78390	9273
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curvina Way	Orlando	TX 12390	9113
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX 76321	5932
9900000010	123340377	JeanLuc	T	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA 57710	5113
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA 33386	7923

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN 33390	3453
9999999999	900359999	Test	T	Testerton	18-NOV-1999	999 Tester Terrace	Testerfield	MD 45390	9999

13 rows selected.

**Update rows to test trigger.  
Verify trigger's changes to tables.**



Oracle SQL Developer: Ian

Worksheet

```

SELECT * FROM Cust_Mod_Log;
UPDATE ALL_CUSTOMERS SET STREET = '109090 Experiment Lane'
WHERE FNAME = 'Test';
SELECT * FROM Cust_Mod_Log;
SELECT * FROM ALL_CUSTOMERS;

```

Script Output

```

Task completed in 2.284 seconds

CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 999 Tester Terrace      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 109090 Experiment Lane      Testerfield

CUSTID SSN      FNAME      MNAME      LNAME      DOB      STREET      CITY      ST ZIP      HOMEI
-----      -----
9900000001 555887798 Liz      Wilson      02-FEB-1952 5 Braddock Rd      Frothburg      MD 21532      3329
9900000002 124556987 Isaiah      Francis      Brooks      18-APR-1970 5 Green St      Cumberland      MD 22105
9900000003 235689952 Laila      Abdul      Simpson      04-MAR-1993 76 Grant Ave      Frostburg      MD 21532      3012
9900000004 900350077 Bart      J      Simpson      13-MAY-21 742 Evergreen Terrace      Springfield      MD 45390      0123
9900000005 123550077 Wanda      X      Bison      13-MAR-1991 715 Elviran Terrace      Springville      OH 56390      5113
9900000006 530350678 Wilma      N      Flintstone      13-MAY-21 345 Cave Stone Road      Bedrock      NV 48286      3003
9900000007 900200072 Wade      M      McGee      19-OCT-1901 12 Wolverton Lake      Waterflood      NE 78390      9273
9900000008 212354074 Jane      J      Mercadino      01-DEC-1985 911 Curvina Way      Orlando      TX 12390      9113
9900000009 988350077 George      Xander      Jetson      13-MAY-21 78 Sky Pad Apartments      Atmosphere      XX 76321      5932
9900000010 123340377 JeanLuc      T      Picard      13-MAY-21 1701 Enterprise Galaxy      Federation      EA 57710      9113
9900000011 730352378 Kimberly      Nancy      Blackcloud      25-SEP-2007 345 Ocean Terrace      Atlantis      BA 33986      7923

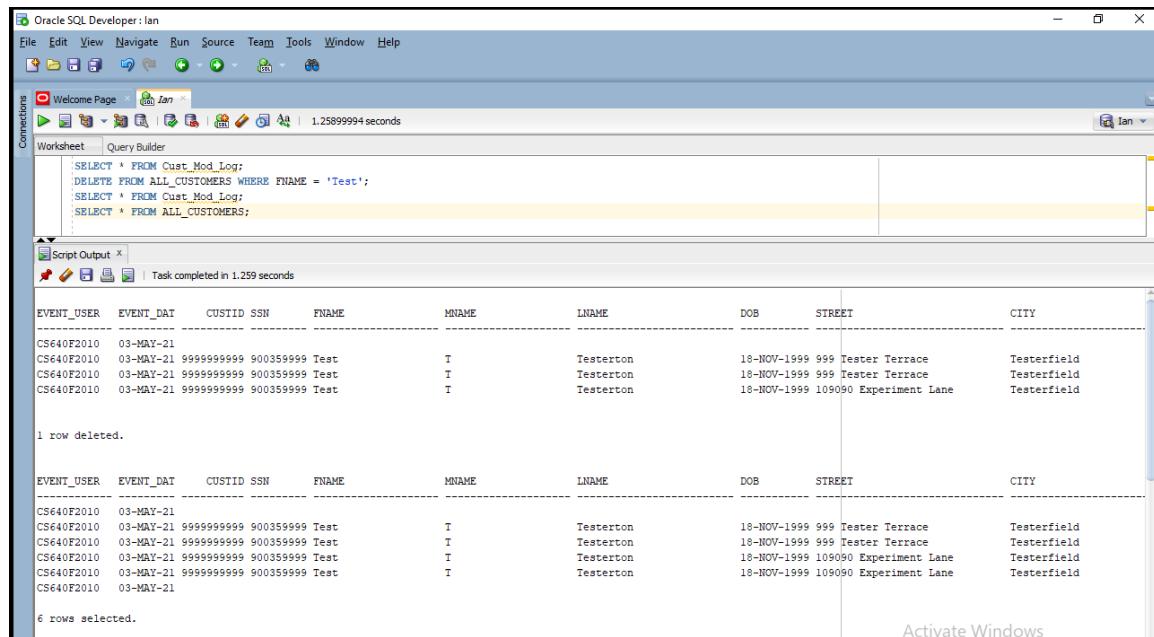
CUSTID SSN      FNAME      MNAME      LNAME      DOB      STREET      CITY      ST ZIP      HOMEI
-----      -----
9900000012 212354932 Bethany      Juliet      Constantine      17-MAY-1949 32890 Palladium Circle      Richville      TN 33390      3453
9999999999 900359999 Test      T      Testerton      18-NOV-1999 109090 Experiment Lane      Testerfield      MD 45390      9999

13 rows selected.

```

Activate Windows  
Go to Settings to activate Windows.

**Delete rows to test trigger.  
Verify trigger's changes to tables.**



Oracle SQL Developer: Ian

Worksheet

```

SELECT * FROM Cust_Mod_Log;
DELETE FROM ALL_CUSTOMERS WHERE FNAME = 'Test';
SELECT * FROM Cust_Mod_Log;
SELECT * FROM ALL_CUSTOMERS;

```

Script Output

```

Task completed in 1.259 seconds

EVENT_USER      EVENT_DATE      CUSTID SSN      FNAME      MNAME      LNAME      DOB      STREET      CITY
-----      -----
CS640F2010 03-MAY-21
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 999 Tester Terrace      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 999 Tester Terrace      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 109090 Experiment Lane      Testerfield

1 row deleted.

EVENT_USER      EVENT_DATE      CUSTID SSN      FNAME      MNAME      LNAME      DOB      STREET      CITY
-----      -----
CS640F2010 03-MAY-21
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 999 Tester Terrace      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 999 Tester Terrace      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 109090 Experiment Lane      Testerfield
CS640F2010 03-MAY-21 9999999999 900359999 Test      T      Testerton      18-NOV-1999 109090 Experiment Lane      Testerfield

6 rows selected.

```

Activate Windows

**Delete rows to test trigger.  
Verify trigger's changes to tables.**

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for database navigation and management. The Connections panel on the left shows a single connection named 'Jan'. The Worksheet tab is active, displaying a script in the editor:

```
SELECT * FROM Cust_Mod_Log;
DELETE FROM ALL_CUSTOMERS WHERE FNAME = 'Test';
SELECT * FROM CUST_MOD_LOG;
SELECT * FROM ALL_CUSTOMERS;
```

The Script Output window shows the results of the executed script:

```
Task completed in 1.259 seconds
CS640F2010 03-MAY-21 9999999999 9003599999 Test
CS640F2010 03-MAY-21

6 rows selected.
```

The main pane displays the data from the 'ALL\_CUSTOMERS' table:

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST ZIP	HOMEI
9900000001	555807798	Liz		Wilson	02-FEB-1952	5 Braddock Rd	Frotnburg	MD 21532	3329
9900000002	124556987	Isaiah	Francis	Brooks	18-APR-1970	5 Green St	Cumberland	MD 22105	
9900000003	235689952	Laila		Abdul	04-MAR-1993	76 Grant Ave	Frostburg	MD 21532	3012
9900000004	900350077	Bart	J	Simpson	13-MAY-21	742 Evergreen Terrace	Springfield	MD 45390	0123
9900000005	123500777	Wanda	X	Bison	13-MAR-1991	115 Elviran Terrace	Springville	OK 65390	5113
9900000006	530350678	Wilma	N	Flintstone	13-MAY-21	345 Cave Stone Road	Bedrock	NE 48286	3003
9900000007	900200072	Wade	M	McGee	19-OCT-1901	901 Wolfgang Lake	Waterflood	TX 76390	9273
9900000008	212354074	Jane	J	Mercadino	01-DEC-1985	911 Curving Way	Orlando	TX 12390	9113
9900000009	988350077	George	Xander	Jetson	13-MAY-21	78 Sky Pad Apartments	Atmosphere	XX 76321	5932
9900000010	123340377	JeanLuc	I	Picard	13-MAY-21	1701 Enterprise Galaxy	Federation	EA 57710	5113
9900000011	730352378	Kimberly	Nancy	Blackcloud	25-SEP-2007	345 Ocean Terrace	Atlantis	BA 33386	7923

The main pane also shows the results of a second query:

CUSTID	SSN	FNAME	MNAME	LNAME	DOB	STREET	CITY	ST ZIP	HOMEI
9900000012	212354932	Bethany	Juliet	Constantine	17-MAY-1949	32890 Palladium Circle	Richville	TN 33390	3453

At the bottom, a message reads: 'Activate Windows Go to Settings to activate Windows.'

^^^^^^^^^^^^^^^^^^^^^^^^

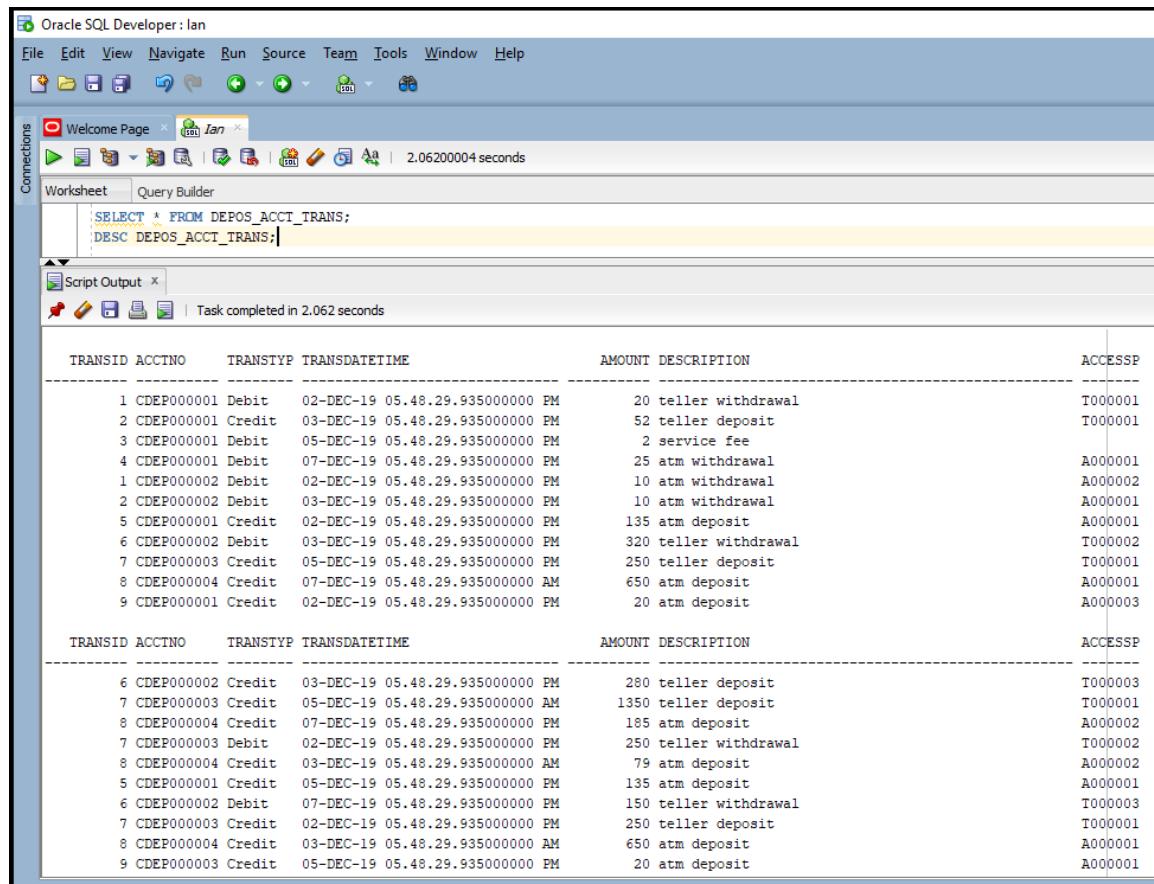
## CHAPTER 5C Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

C. Any deposit larger than \$5000 to any account will also be written in a **Large\_Dep\_Log** file.

**Check table contents.**

**Check table attribute data types.**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 2.06200004 seconds

Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCT_TRANS;
DESC DEPOS_ACCT_TRANS;
```

Script Output Task completed in 2.062 seconds

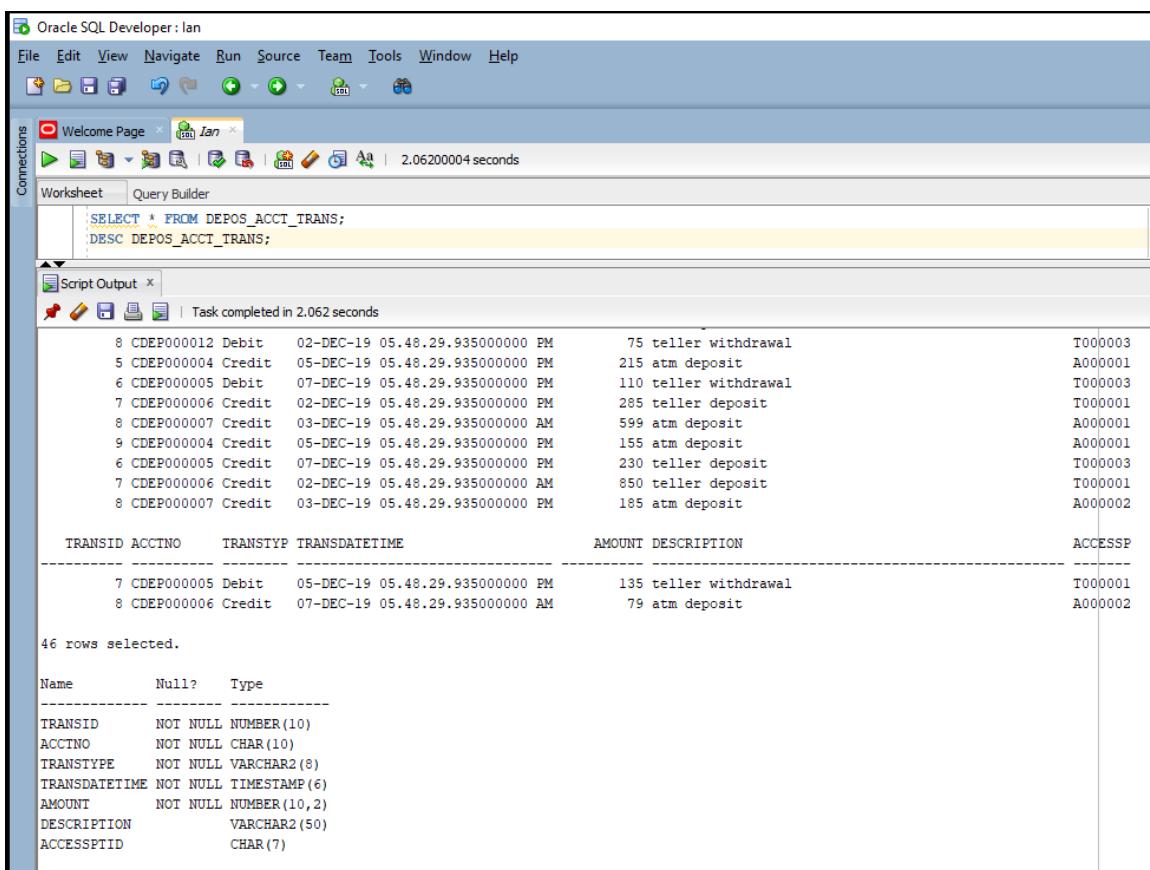
TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.93500000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.9350000000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.9350000000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.9350000000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.9350000000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.9350000000 PM	10	atm withdrawal	A000001
5	CDEP000001	Credit	02-DEC-19 05.48.29.9350000000 PM	135	atm deposit	A000001
6	CDEP000002	Debit	03-DEC-19 05.48.29.9350000000 PM	320	teller withdrawal	T000002
7	CDEP000003	Credit	05-DEC-19 05.48.29.9350000000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.9350000000 AM	650	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.9350000000 PM	20	atm deposit	A000003

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
6	CDEP000002	Credit	03-DEC-19 05.48.29.9350000000 PM	280	teller deposit	T000003
7	CDEP000003	Credit	05-DEC-19 05.48.29.9350000000 AM	1350	teller deposit	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.9350000000 PM	185	atm deposit	A000002
7	CDEP000003	Debit	02-DEC-19 05.48.29.9350000000 PM	250	teller withdrawal	T000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.9350000000 AM	79	atm deposit	A000002
5	CDEP000001	Credit	05-DEC-19 05.48.29.9350000000 PM	135	atm deposit	A000001
6	CDEP000002	Debit	07-DEC-19 05.48.29.9350000000 PM	150	teller withdrawal	T000003
7	CDEP000003	Credit	02-DEC-19 05.48.29.9350000000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.9350000000 AM	650	atm deposit	A000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.9350000000 PM	20	atm deposit	A000001

**Check table contents.**

**Check table attribute data types.**



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
SELECT * FROM DEPOS_ACCT_TRANS;
DESC DEPOS_ACCT_TRANS;
```

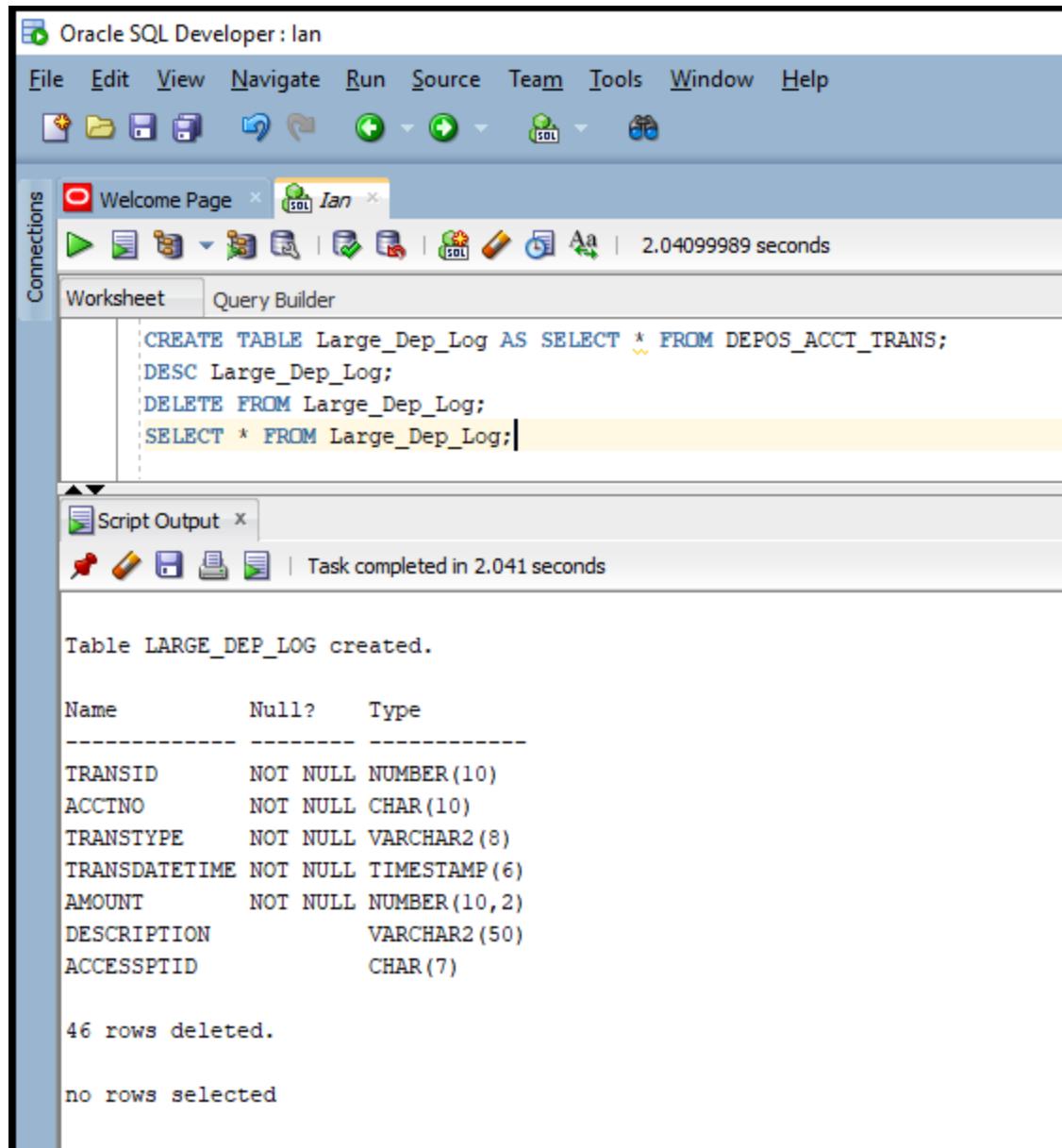
The 'Script Output' tab shows the results of the query:

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP000012	Debit	02-DEC-19 05.48.29.93500000 PM	75	teller withdrawal	T000003
5	CDEP000004	Credit	05-DEC-19 05.48.29.935000000 PM	215	atm deposit	A000001
6	CDEP000005	Debit	07-DEC-19 05.48.29.935000000 PM	110	teller withdrawal	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000000 PM	285	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000000 AM	599	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.935000000 PM	155	atm deposit	A000001
6	CDEP000005	Credit	07-DEC-19 05.48.29.935000000 PM	230	teller deposit	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000000 AM	850	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002

Below the table, the message '46 rows selected.' is displayed. The 'Table Structure' tab shows the table definition:

Name	Null?	Type
TRANSID	NOT NULL	NUMBER(10)
ACCTNO	NOT NULL	CHAR(10)
TRANSTYPE	NOT NULL	VARCHAR2(8)
TRANSDATETIME	NOT NULL	TIMESTAMP(6)
AMOUNT	NOT NULL	NUMBER(10,2)
DESCRIPTION		VARCHAR2(50)
ACCESSPTID		CHAR(7)

## Create empty table.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL code:

```
CREATE TABLE Large_Dep_Log AS SELECT * FROM DEPOS_ACCT_TRANS;
DESC Large_Dep_Log;
DELETE FROM Large_Dep_Log;
SELECT * FROM Large_Dep_Log;
```

The 'Script Output' tab shows the results of the execution:

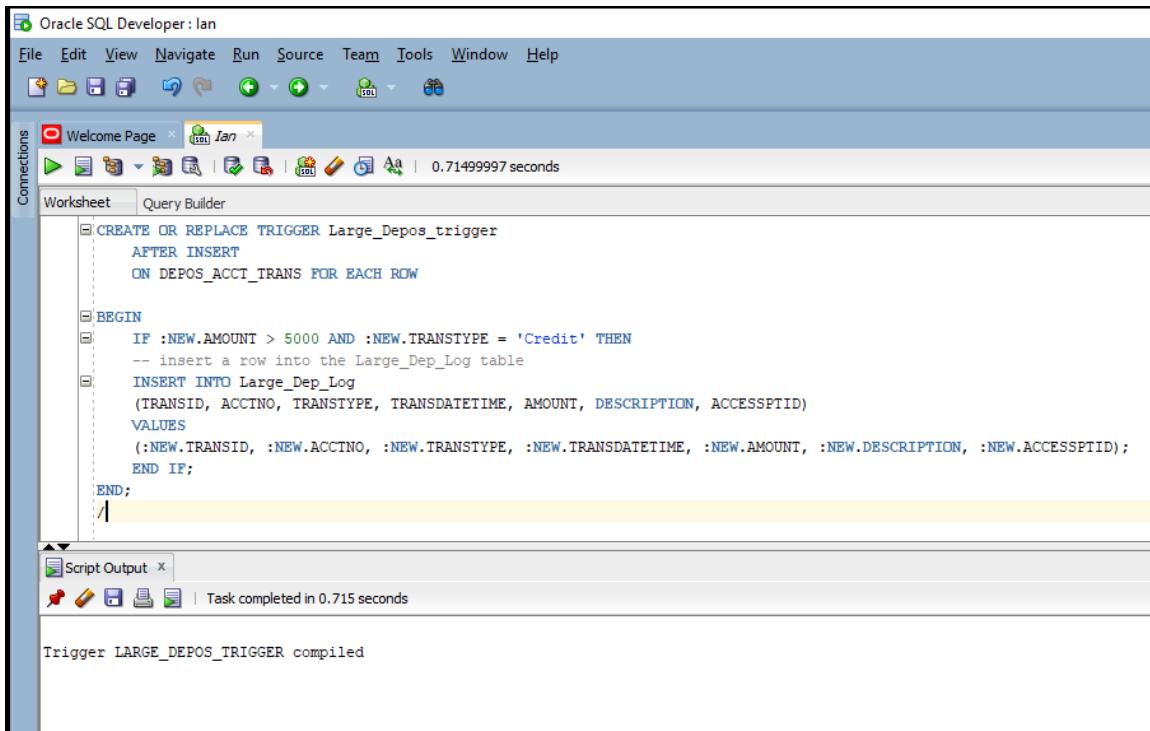
```
Table LARGE_DEP_LOG created.

Name          Null?    Type
-----        -----
TRANSID       NOT NULL NUMBER(10)
ACCTNO        NOT NULL CHAR(10)
TRANSTYPE     NOT NULL VARCHAR2(8)
TRANSDATETIME NOT NULL TIMESTAMP(6)
AMOUNT        NOT NULL NUMBER(10,2)
DESCRIPTION    VARCHAR2(50)
ACCESSPTID    CHAR(7)

46 rows deleted.

no rows selected
```

## Create trigger.



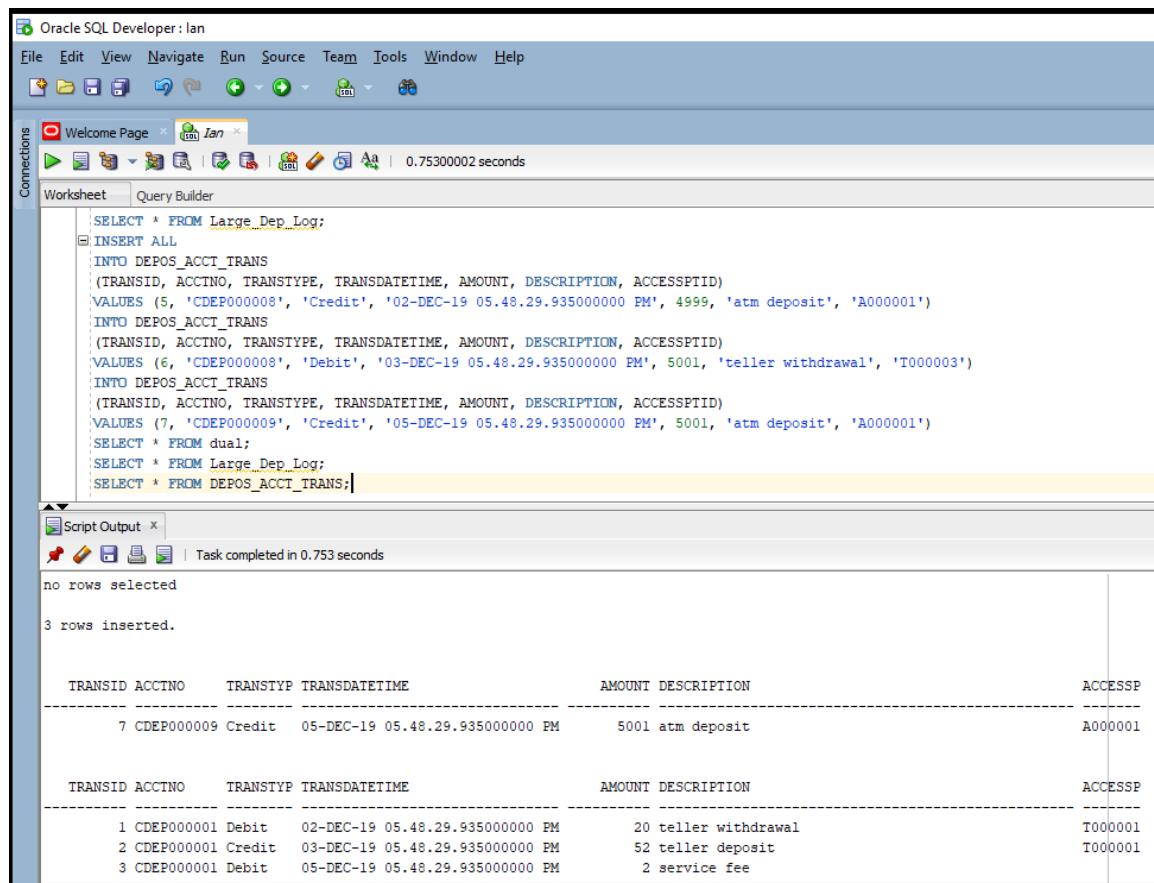
The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is selected, displaying the SQL code for creating a trigger. The trigger, named 'Large\_Deps\_trigger', is defined to fire AFTER an INSERT on the 'DEPOS\_ACCT\_TRANS' table for each row. The trigger body contains an IF block that checks if the new amount is greater than 5000 and the transaction type is 'Credit'. If true, it inserts a row into the 'Large\_Dep\_Log' table with the new transaction details. The code is syntax-highlighted, and the 'Script Output' tab at the bottom shows the message 'Trigger LARGE\_DEPOS\_TRIGGER compiled'.

```
CREATE OR REPLACE TRIGGER Large_Deps_trigger
  AFTER INSERT
  ON DEPOS_ACCT_TRANS FOR EACH ROW

  BEGIN
    IF :NEW.AMOUNT > 5000 AND :NEW.TRANSTYPE = 'Credit' THEN
      -- insert a row into the Large_Dep_Log table
      INSERT INTO Large_Dep_Log
      (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
      VALUES
      (:NEW.TRANSID, :NEW.ACCTNO, :NEW.TRANSTYPE, :NEW.TRANSDATETIME, :NEW.AMOUNT, :NEW.DESCRIPTION, :NEW.ACCESSPTID);
    END IF;
  END;
```

Trigger LARGE\_DEPOS\_TRIGGER compiled

## Insert rows to test trigger's changes to tables.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script. The script includes a select statement from 'Large\_Dep\_Log', an 'INSERT ALL' statement into 'DEPOS\_ACCT\_TRANS' table with three rows of data (Credit, Debit, Credit), and a final select statement from 'Large\_Dep\_Log'. Below the worksheet, the 'Script Output' tab shows the results: 'no rows selected' followed by '3 rows inserted.' Below this, two tables are displayed showing the inserted data. The first table shows a single credit transaction (Credit, 05-DEC-19, 5001, atm deposit). The second table shows three transactions: a debit (Debit, 02-DEC-19, 20), a credit (Credit, 03-DEC-19, 52), and a service fee (Debit, 05-DEC-19, 2).

```
SELECT * FROM Large_Dep_Log;
INSERT ALL
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (5, 'CDEP000008', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 4999, 'atm deposit', 'A000001')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (6, 'CDEP000008', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 5001, 'teller withdrawal', 'T000003')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (7, 'CDEP000009', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 5001, 'atm deposit', 'A000001')
SELECT * FROM dual;
SELECT * FROM Large_Dep_Log;
SELECT * FROM DEPOS_ACCT_TRANS;
```

Script Output

Task completed in 0.753 seconds

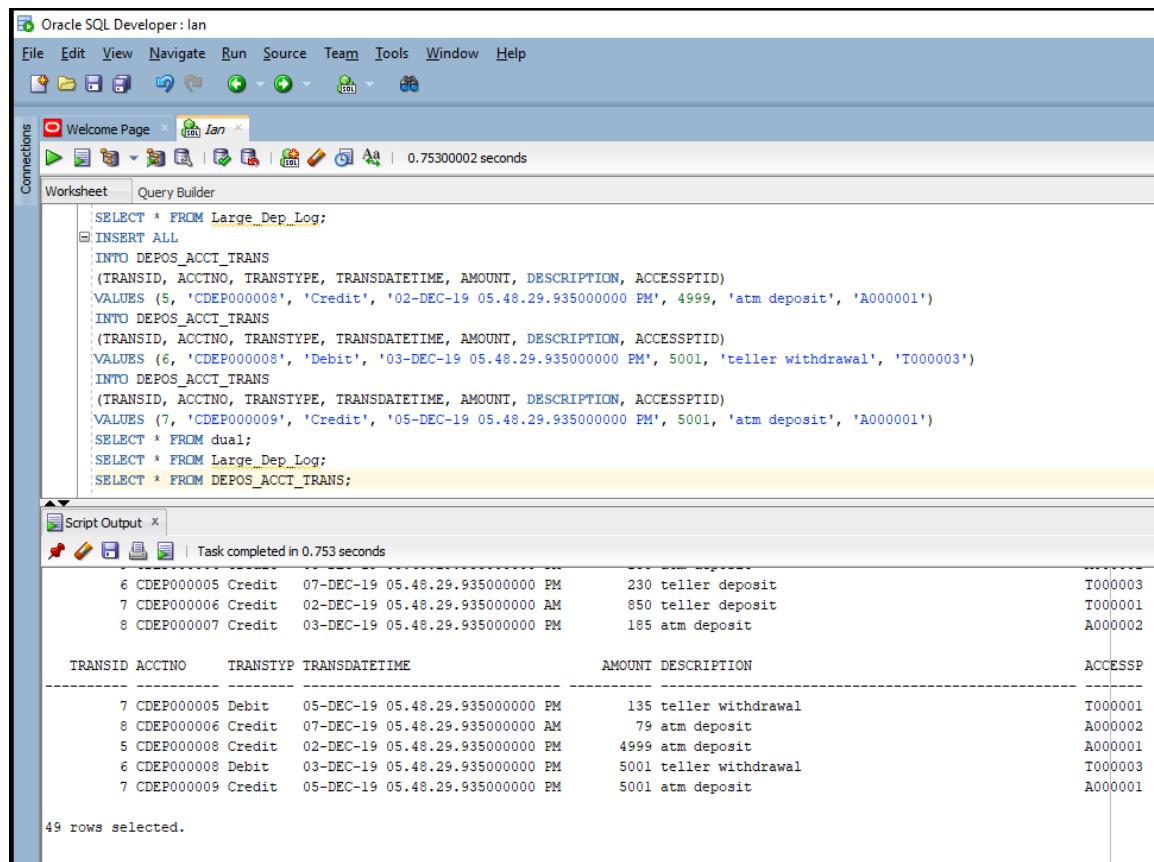
no rows selected

3 rows inserted.

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
7	CDEP000009	Credit	05-DEC-19 05.48.29.935000000 PM	5001	atm deposit	A000001

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000000 PM	2	service fee	

## Insert rows to test trigger's changes to tables.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script. The script starts with a comment `/\*` followed by `SELECT \* FROM Large\_Dep\_Log;` and then an `INSERT ALL` block. The `INSERT ALL` block contains four `INTO` statements for the `DEPOS\_ACCT\_TRANS` table, each with a `VALUES` clause. The first three `VALUES` clauses represent deposits (Credit), and the fourth represents a withdrawal (Debit). After the `INSERT ALL` block, there are two `SELECT \*` statements: one from the `dual` table and one from the `Large\_Dep\_Log` table. The 'Script Output' tab shows the results of the execution. It displays the inserted data in a table format with columns: TRANSID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSPTID. The results show 49 rows selected, with various transaction details like teller deposits, ATM deposits, and teller withdrawals.

```
/* SELECT * FROM Large_Dep_Log;
INSERT ALL
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (5, 'CDEP000008', 'Credit', '02-DEC-19 05.48.29.935000000 PM', 4999, 'atm deposit', 'A000001')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (6, 'CDEP000008', 'Debit', '03-DEC-19 05.48.29.935000000 PM', 5001, 'teller withdrawal', 'T000003')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (7, 'CDEP000009', 'Credit', '05-DEC-19 05.48.29.935000000 PM', 5001, 'atm deposit', 'A000001')
SELECT * FROM dual;
SELECT * FROM Large_Dep_Log;
SELECT * FROM DEPOS_ACCT_TRANS;
```

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSPTID
6	CDEP000005	Credit	07-DEC-19 05.48.29.935000000 PM	230	teller deposit	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.935000000 AM	850	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002
7	CDEP000005	Debit	05-DEC-19 05.48.29.935000000 PM	135	teller withdrawal	T000001
8	CDEP000006	Credit	07-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002
5	CDEP000008	Credit	02-DEC-19 05.48.29.935000000 PM	4999	atm deposit	A000001
6	CDEP000008	Debit	03-DEC-19 05.48.29.935000000 PM	5001	teller withdrawal	T000003
7	CDEP000009	Credit	05-DEC-19 05.48.29.935000000 PM	5001	atm deposit	A000001

49 rows selected.

^^^^^^^^^^^^^^^^^^^^^^^^

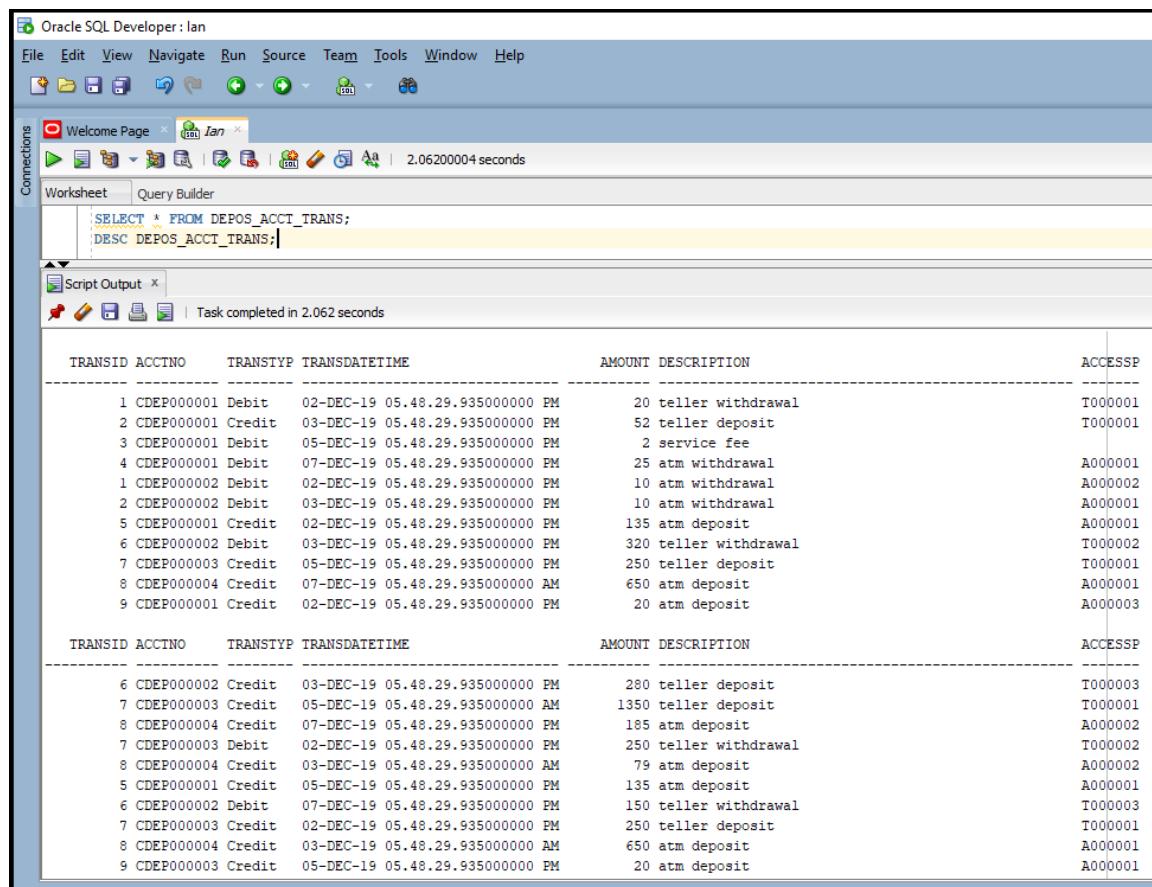
## CHAPTER 5D Starts here

^^^^^^^^^^^^^^^^^^^^^^^^

**D. Any withdraw larger than \$10000 from any account also will be written in a Large\_With\_Log file.**

**Check table contents.**

**Check table attribute data types.**



Oracle SQL Developer : Ian

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page Jan 2.06200004 seconds

Worksheet Query Builder

```
SELECT * FROM DEPOS_ACCT_TRANS;
DESC DEPOS_ACCT_TRANS;
```

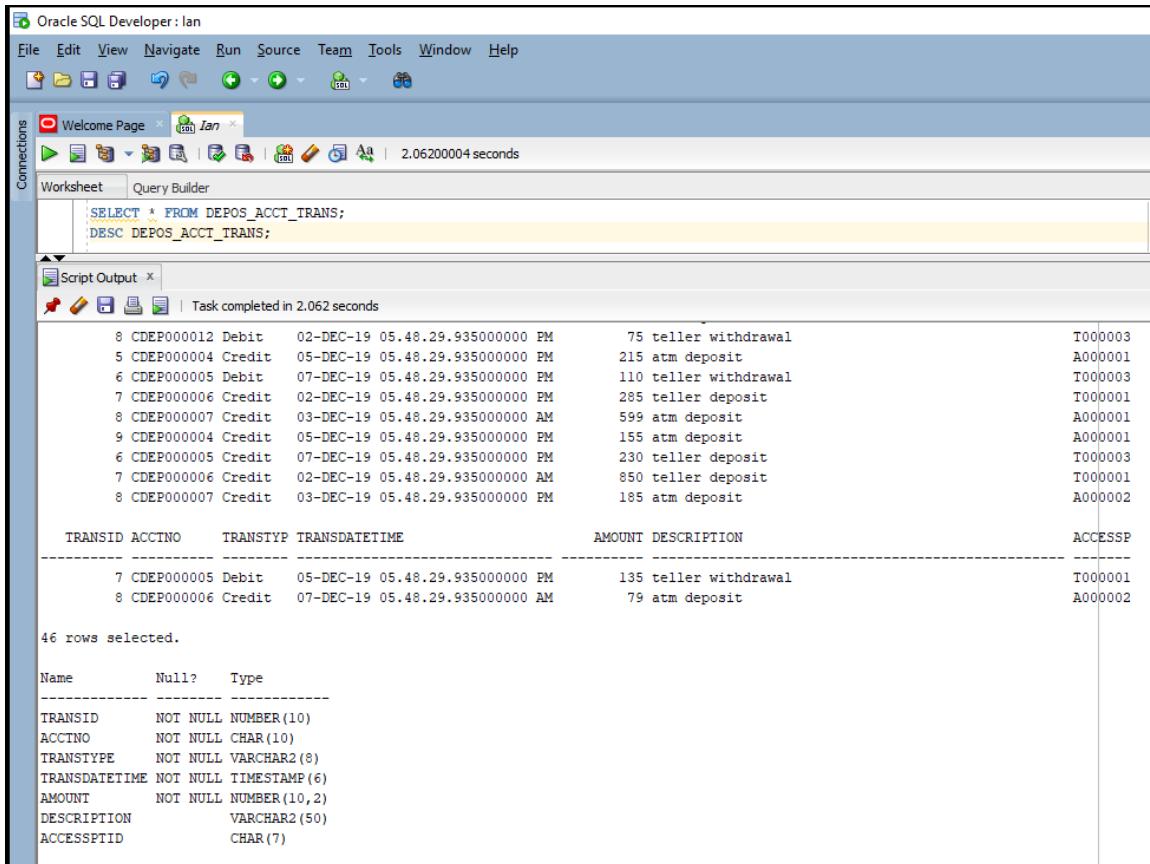
Script Output Task completed in 2.062 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
1	CDEP000001	Debit	02-DEC-19 05.48.29.935000000 PM	20	teller withdrawal	T000001
2	CDEP000001	Credit	03-DEC-19 05.48.29.935000000 PM	52	teller deposit	T000001
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000000 PM	2	service fee	
4	CDEP000001	Debit	07-DEC-19 05.48.29.935000000 PM	25	atm withdrawal	A000001
1	CDEP000002	Debit	02-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000001
5	CDEP000001	Credit	02-DEC-19 05.48.29.935000000 PM	135	atm deposit	A000001
6	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	320	teller withdrawal	T000002
7	CDEP000003	Credit	05-DEC-19 05.48.29.935000000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000000 AM	650	atm deposit	A000001
9	CDEP000001	Credit	02-DEC-19 05.48.29.935000000 PM	20	atm deposit	A000003

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
6	CDEP000002	Credit	03-DEC-19 05.48.29.935000000 PM	280	teller deposit	T000003
7	CDEP000003	Credit	05-DEC-19 05.48.29.935000000 AM	1350	teller deposit	T000001
8	CDEP000004	Credit	07-DEC-19 05.48.29.935000000 PM	185	atm deposit	A000002
7	CDEP000003	Debit	02-DEC-19 05.48.29.935000000 PM	250	teller withdrawal	T000002
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 AM	79	atm deposit	A000002
5	CDEP000001	Credit	05-DEC-19 05.48.29.935000000 PM	135	atm deposit	A000001
6	CDEP000002	Debit	07-DEC-19 05.48.29.935000000 PM	150	teller withdrawal	T000003
7	CDEP000003	Credit	02-DEC-19 05.48.29.935000000 PM	250	teller deposit	T000001
8	CDEP000004	Credit	03-DEC-19 05.48.29.935000000 AM	650	atm deposit	A000001
9	CDEP000003	Credit	05-DEC-19 05.48.29.935000000 PM	20	atm deposit	A000001

**Check table contents.  
Check table attribute data types.**



The screenshot shows the Oracle SQL Developer interface with a query being run against a database. The query is:

```
SELECT * FROM DEPOS_ACCT_TRANS;  
DESC DEPOS_ACCT_TRANS;
```

The results of the query are displayed in the 'Script Output' tab. The output shows 46 rows of transaction data and the table structure definition.

**Script Output**

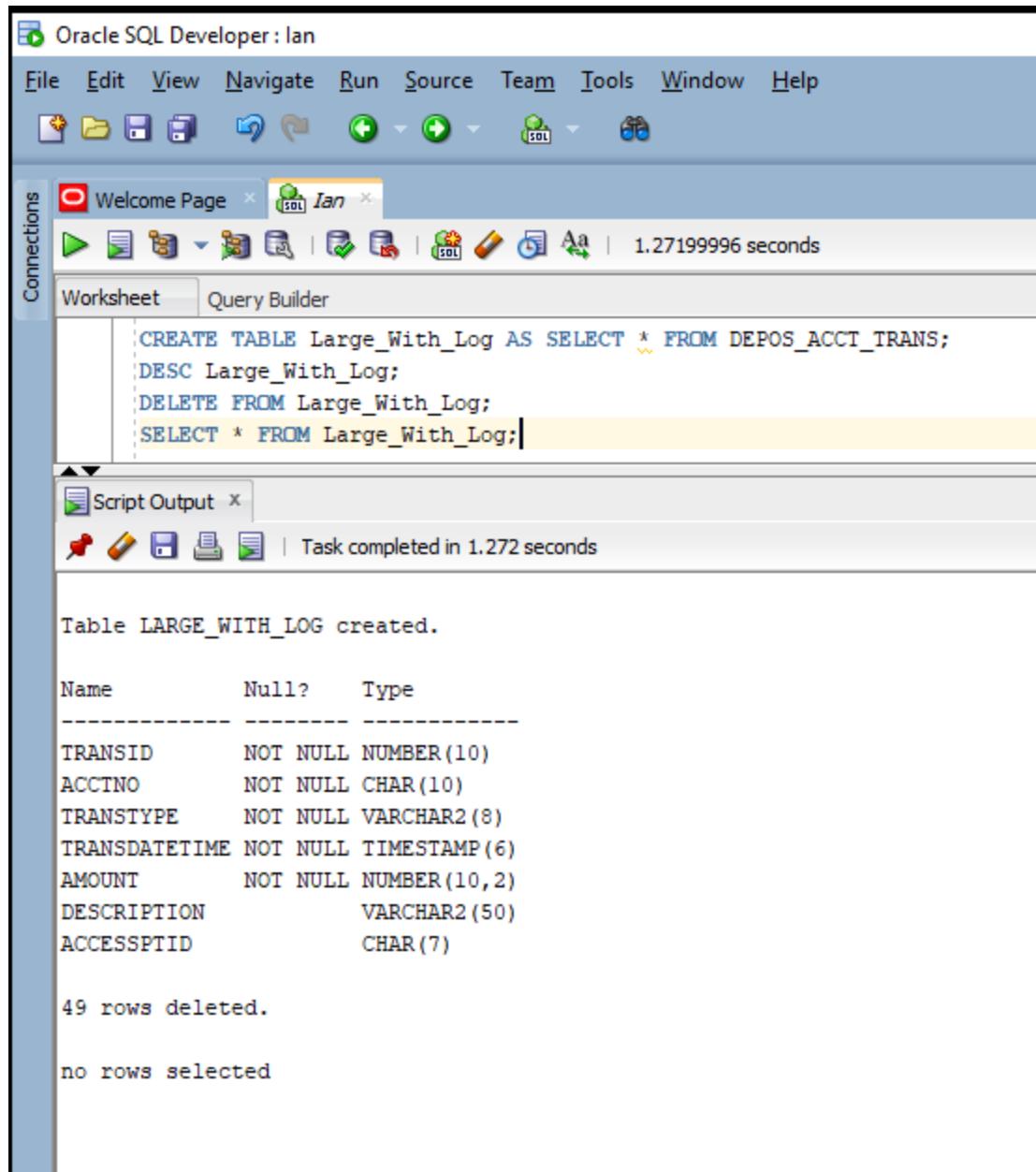
Task completed in 2.062 seconds

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSP
8	CDEP000012	Debit	02-DEC-19 05.48.29.93500000 PM	75	teller withdrawal	T000003
5	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	215	atm deposit	A000001
6	CDEP000005	Debit	07-DEC-19 05.48.29.93500000 PM	110	teller withdrawal	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 PM	285	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 AM	599	atm deposit	A000001
9	CDEP000004	Credit	05-DEC-19 05.48.29.93500000 PM	155	atm deposit	A000001
6	CDEP000005	Credit	07-DEC-19 05.48.29.93500000 PM	230	teller deposit	T000003
7	CDEP000006	Credit	02-DEC-19 05.48.29.93500000 AM	850	teller deposit	T000001
8	CDEP000007	Credit	03-DEC-19 05.48.29.93500000 PM	185	atm deposit	A000002

46 rows selected.

Name	Null?	Type
TRANSID	NOT NULL	NUMBER(10)
ACCTNO	NOT NULL	CHAR(10)
TRANSTYPE	NOT NULL	VARCHAR2(8)
TRANSDATETIME	NOT NULL	TIMESTAMP(6)
AMOUNT	NOT NULL	NUMBER(10,2)
DESCRIPTION		VARCHAR2(50)
ACCESSPTID		CHAR(7)

## Create empty table.



The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the following SQL script:

```
CREATE TABLE Large_With_Log AS SELECT * FROM DEPOS_ACCT_TRANS;
DESC Large_With_Log;
DELETE FROM Large_With_Log;
SELECT * FROM Large_With_Log;
```

The 'Script Output' tab shows the results of the execution:

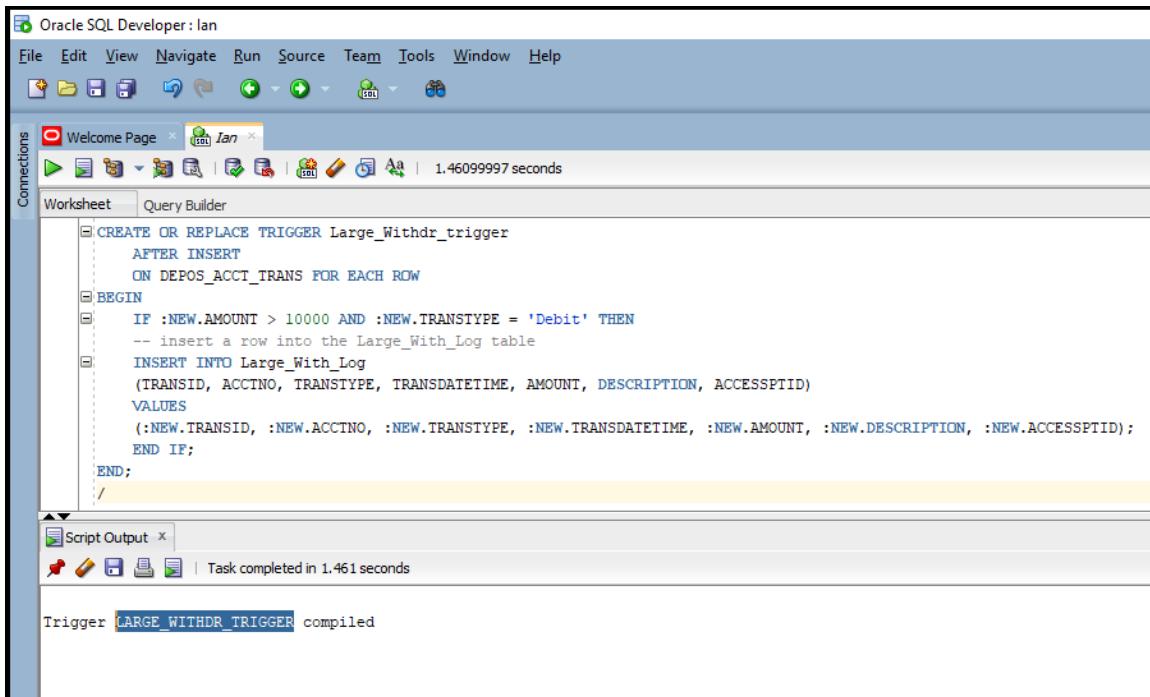
```
Table LARGE_WITH_LOG created.

Name          Null?    Type
-----
TRANSID      NOT NULL NUMBER(10)
ACCTNO       NOT NULL CHAR(10)
TRANSTYPE    NOT NULL VARCHAR2(8)
TRANSDATETIME NOT NULL TIMESTAMP(6)
AMOUNT        NOT NULL NUMBER(10,2)
DESCRIPTION   VARCHAR2(50)
ACCESSPTID   CHAR(7)

49 rows deleted.

no rows selected
```

## Create trigger.

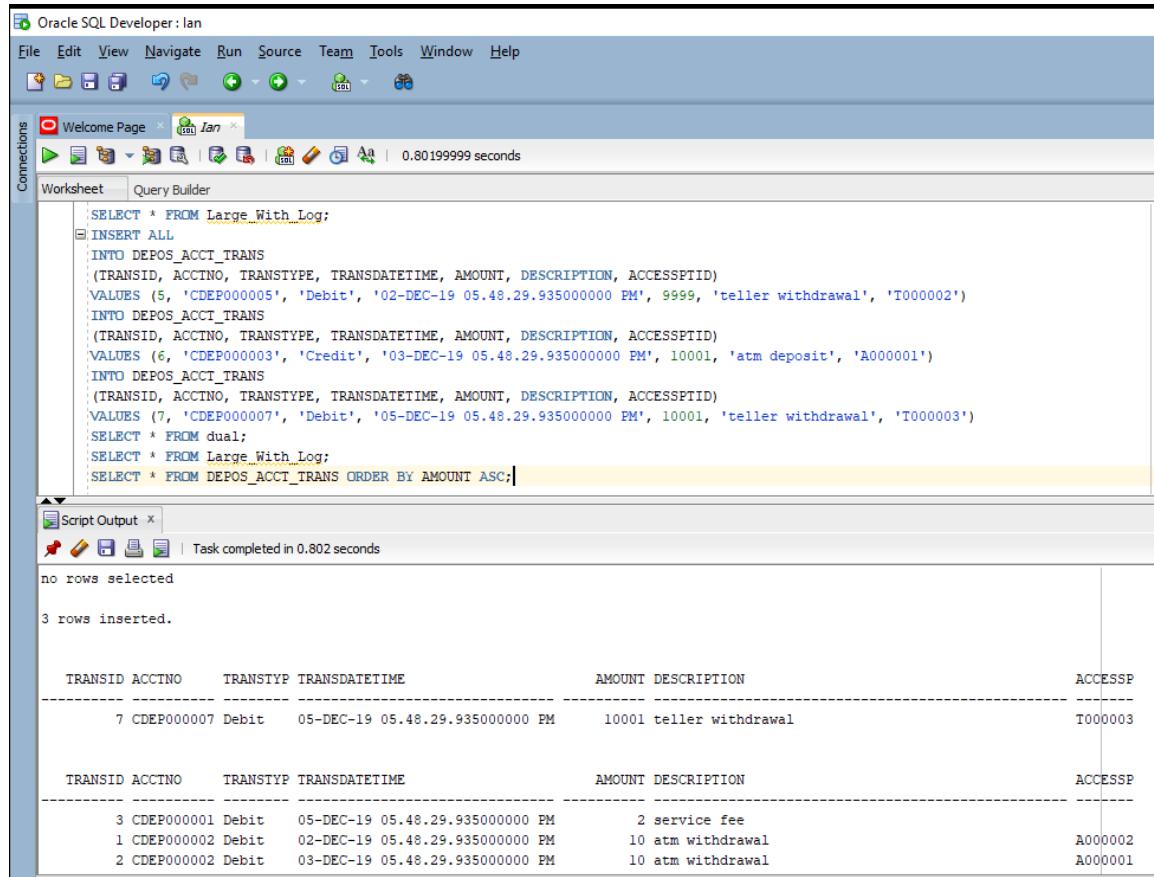


The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Ian". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for file operations. The Connections panel shows a connection to "Welcome Page" and "Jan". The Worksheet tab is active, displaying the following PL/SQL code:

```
CREATE OR REPLACE TRIGGER Large_Withdr_trigger
  AFTER INSERT
  ON DEPOS_ACCT_TRANS FOR EACH ROW
BEGIN
  IF :NEW.AMOUNT > 10000 AND :NEW.TRANSTYPE = 'Debit' THEN
    -- insert a row into the Large_With_Log table
    INSERT INTO Large_With_Log
    (TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
    VALUES
    (:NEW.TRANSID, :NEW.ACCTNO, :NEW.TRANSTYPE, :NEW.TRANSDATETIME, :NEW.AMOUNT, :NEW.DESCRIPTION, :NEW.ACCESSPTID);
  END IF;
END;
```

The Script Output panel shows the message "Trigger `LARGE_WITHDR_TRIGGER` compiled" and "Task completed in 1.461 seconds".

## Insert rows to test trigger.



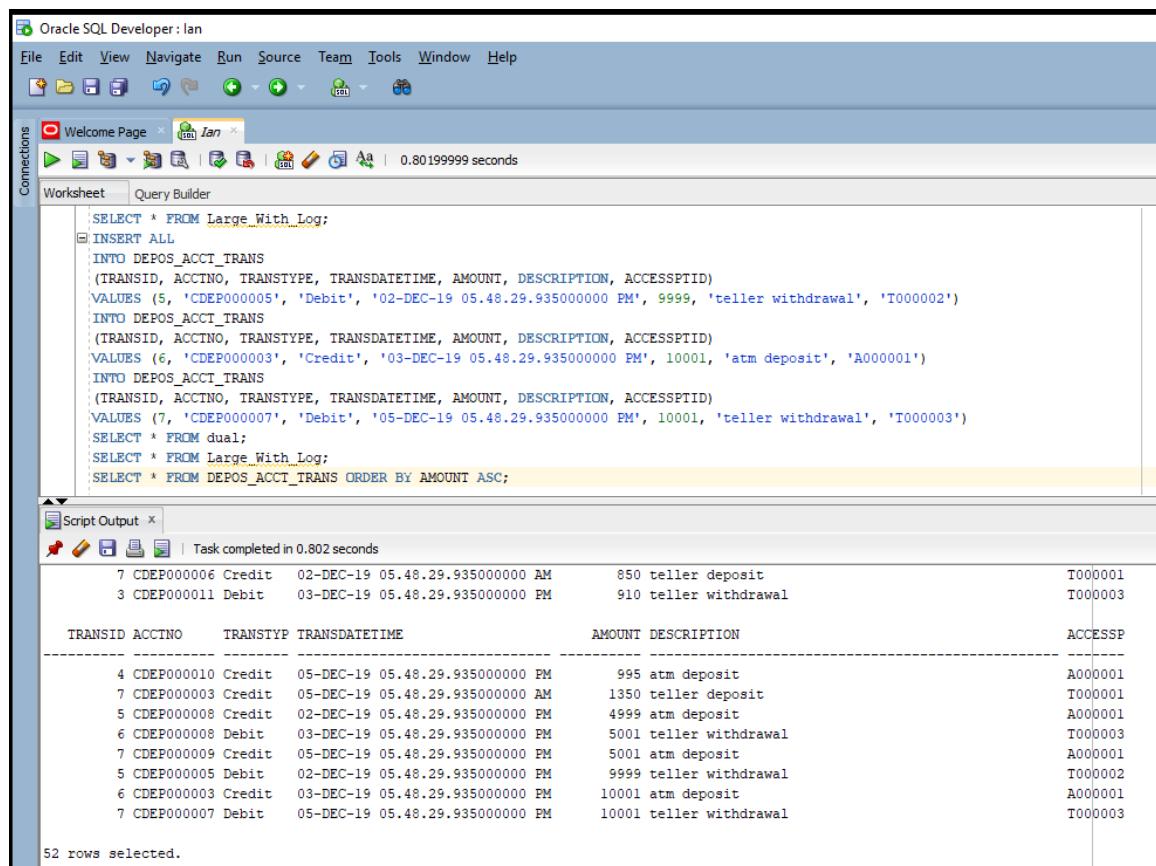
The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a SQL script. The script inserts three rows into the 'DEPOS\_ACCT\_TRANS' table and then selects all rows from the 'Large\_With\_Log' table. The 'Script Output' tab shows the results of the execution: 'no rows selected' and '3 rows inserted.' Below this, two tables show the inserted data. The first table shows a single debit transaction for account CDEP000007. The second table shows three transactions: a debit for account CDEP000001 (2 service fee), a debit for account CDEP000002 (10 atm withdrawal), and a debit for account CDEP000002 (10 atm withdrawal).

```
SELECT * FROM Large_With_Log;
INSERT ALL
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (5, 'CDEP000005', 'Debit', '02-DEC-19 05.48.29.935000000 PM', 9999, 'teller withdrawal', 'T000002')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (6, 'CDEP000003', 'Credit', '03-DEC-19 05.48.29.935000000 PM', 10001, 'atm deposit', 'A000001')
INTO DEPOS_ACCT_TRANS
(TRANSID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (7, 'CDEP000007', 'Debit', '05-DEC-19 05.48.29.935000000 PM', 10001, 'teller withdrawal', 'T000003')
SELECT * FROM dual;
SELECT * FROM Large_With_Log;
SELECT * FROM DEPOS_ACCT_TRANS ORDER BY AMOUNT ASC;
```

TRANSID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSPT
7	CDEP000007	Debit	05-DEC-19 05.48.29.935000000 PM	10001	teller withdrawal	T000003

TRANSID	ACCTNO	TRANSTYPE	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSPT
3	CDEP000001	Debit	05-DEC-19 05.48.29.935000000 PM	2	service fee	
1	CDEP000002	Debit	02-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000002
2	CDEP000002	Debit	03-DEC-19 05.48.29.935000000 PM	10	atm withdrawal	A000001

## Insert rows to test trigger.



The screenshot shows the Oracle SQL Developer interface. The Worksheet pane contains a SQL script to insert data into a table named 'Large\_With\_Log'. The script includes several INSERT statements with specific values for columns like TRANSDID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSPTID. It also includes a SELECT \* FROM Large\_With\_Log; statement at the end. The Script Output pane shows the results of the execution, displaying 52 rows selected. The output table has columns: TRANSID, ACCTNO, TRANSTYP, TRANSDATETIME, AMOUNT, DESCRIPTION, and ACCESSPTID. The data includes various transactions like 'Credit' and 'Debit' with amounts ranging from 999 to 10001, and descriptions like 'teller deposit' and 'teller withdrawal'.

```
SELECT * FROM Large_With_Log;
INSERT ALL
INTO DEPOS_ACCT_TRANS
(TRANSDID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (5, 'CDEP000005', 'Debit', '02-DEC-19 05.48.29.93500000 PM', 999, 'teller withdrawal', 'T000002')
INTO DEPOS_ACCT_TRANS
(TRANSDID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (6, 'CDEP000003', 'Credit', '03-DEC-19 05.48.29.93500000 PM', 10001, 'atm deposit', 'A000001')
INTO DEPOS_ACCT_TRANS
(TRANSDID, ACCTNO, TRANSTYPE, TRANSDATETIME, AMOUNT, DESCRIPTION, ACCESSPTID)
VALUES (7, 'CDEP000007', 'Debit', '05-DEC-19 05.48.29.93500000 PM', 10001, 'teller withdrawal', 'T000003')
SELECT * FROM dual;
SELECT * FROM Large_With_Log;
SELECT * FROM DEPOS_ACCT_TRANS ORDER BY AMOUNT ASC;
```

TRANSID	ACCTNO	TRANSTYP	TRANSDATETIME	AMOUNT	DESCRIPTION	ACCESSPTID
4	CDEP000010	Credit	05-DEC-19 05.48.29.93500000 PM	995	atm deposit	A000001
7	CDEP000003	Credit	05-DEC-19 05.48.29.93500000 AM	1350	teller deposit	T000001
5	CDEP000008	Credit	02-DEC-19 05.48.29.93500000 PM	4999	atm deposit	A000001
6	CDEP000008	Debit	03-DEC-19 05.48.29.93500000 PM	5001	teller withdrawal	T000003
7	CDEP000009	Credit	05-DEC-19 05.48.29.93500000 PM	5001	atm deposit	A000001
5	CDEP000005	Debit	02-DEC-19 05.48.29.93500000 PM	9999	teller withdrawal	T000002
6	CDEP000003	Credit	03-DEC-19 05.48.29.93500000 PM	10001	atm deposit	A000001
7	CDEP000007	Debit	05-DEC-19 05.48.29.93500000 PM	10001	teller withdrawal	T000003

52 rows selected.