

Database Programming

Practical Exercise #7

Student Name : Shreejana Shrestha

Student Id : C0930321

Question 1: nested skill_list creation

The screenshot displays the Oracle SQL Developer interface. The left pane shows the 'Connections' tree with 'ShreejanaDB' selected. The main window is the 'Query Builder' tab, showing the following SQL script:

```
CREATE OR REPLACE TYPE skill_list AS TABLE OF VARCHAR2(100);

CREATE TABLE employees_321(
  id number(10) GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY,
  name varchar2(100),
  position varchar2(50),
  salary number(10, 2),
  created_at TIMESTAMP,
  CONSTRAINT employeeSkill_pk FOREIGN KEY (skills) REFERENCES skill_list(id)
);

CREATE TABLE employees_321 (
  id NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY,
  name VARCHAR2(100),
  position VARCHAR2(50),
  salary NUMBER(10, 2),
  skills skill_list,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
NESTED TABLE skills STORE AS skills_nested_table;
/
```

The bottom pane shows the 'Script Output' tab with the following messages:

```
1 row updated.

Type SKILL_LIST compiled
```

Question 2: employees_321 table creation

The screenshot displays the Oracle SQL Developer interface. The left sidebar shows the 'Connections' pane with 'ShreejanaDB' selected. The main workspace is titled 'Worksheet' and contains the following SQL script:

```
CREATE OR REPLACE TYPE skill_list AS TABLE OF VARCHAR2(100);

CREATE TABLE employees_321 (
    id NUMBER GENERATED ALWAYS AS IDENTITY (START with 1 INCREMENT by 1) PRIMARY KEY,
    name VARCHAR2(100),
    position VARCHAR2(50),
    salary NUMBER(10, 2),
    skills skill_list,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
NESTED TABLE skills STORE AS skills_nested_table;

CREATE TABLE employees_audit (
    id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    employee_id NUMBER REFERENCES employees_UserID(id),
    action VARCHAR2(50),
    old_skills skill_list,
    new_skills skill_list,
    changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

Below the script, the 'Script Output' pane shows the message: 'Table EMPLOYEES_321 created.' The 'Query Result' pane is empty. The status bar at the bottom indicates 'Task completed in 0.235 seconds'.

Question 3: employees_audit table creation

The screenshot displays the Oracle SQL Developer interface. The left sidebar shows the 'Connections' pane with 'ShreejanaDB' selected, and the 'Reports' pane. The main workspace is the 'Query Builder' window, which contains the following SQL script:

```
name VARCHAR2(100),
position VARCHAR2(50),
salary NUMBER(10, 2),
skills skill_list,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
NESTED TABLE skills STORE AS skills_nested_table;

-- employees_audit table creation
CREATE TABLE employees_audit (
  id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  employee_id NUMBER REFERENCES employees_321(id),
  action VARCHAR2(50),
  old_skills skill_list,
  new_skills skill_list,
  changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
NESTED TABLE old_skills STORE AS old_skills_nested_table;
NESTED TABLE new_skills STORE AS new_skills_nested_table;

INSERT INTO employees_321(id, name, position, salary, created_at) VALUES('David',
```

Below the query editor, the 'Script Output' pane shows the execution result:

```
Table EMPLOYEES_AUDIT created.
```

The status bar at the bottom indicates 'Task completed in 0.097 seconds'.

Insertion of data for employees_321 table

The screenshot displays the Oracle SQL Developer interface. The left pane shows the 'Connections' tree with 'ShreejanaDB' selected. The main window is split into a 'Worksheet' and a 'Query Builder'. The 'Worksheet' contains a PL/SQL script that updates the 'employees_321' table and displays the old and new records. The 'Query Builder' is currently empty. Below the script, the 'Script Output' and 'Query Result' tabs are visible. The 'Query Result' tab shows the output of the 'select * from employees_321;' query, displaying two rows of data.

```
...sql | ShreejanaDB2.sql | ShreejanaDB2.sql | Welcome Page | ShreejanaDB
Connections
Oracle Connections
ShreejanaDB
Tables (Filter)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Table
Triggers
Types
Sequences
Materialized Views
Materialized Views
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports

Worksheet
Query Builder

:OLD.skills,
:NEW.skills,
SYSTIMESTAMP
);

-- Display the old and new records
DBMS_OUTPUT.PUT_LINE('Employee ID: ' || :OLD.id);
DBMS_OUTPUT.PUT_LINE('Old Skills: ' || TO_CHAR(:OLD.skills));
DBMS_OUTPUT.PUT_LINE('New Skills: ' || TO_CHAR(:NEW.skills));
DBMS_OUTPUT.PUT_LINE('Change Action: ' || v_action);
DBMS_OUTPUT.PUT_LINE('Change Timestamp: ' || SYSTIMESTAMP);
END IF;
END;
/

select * from employees_321;
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.008 seconds

ID	NAME	POSITION	SALARY	SKILLS
1	1 David Smith	Accountant	98754	SYSTEM.SKILL_LIST('Accounting', 'Book keeping')
2	2 Raman Neth	Programmer	85478	SYSTEM.SKILL_LIST('Python', 'React')

Question 4: after_employee_update trigger creation

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main 'Query Builder' window contains the following SQL code:

```
VALUES (  
    :OLD.id,  
    v_action,  
    :OLD.skills,  
    :NEW.skills,  
    SYSTIMESTAMP  
);  
  
-- Display the old and new records  
DBMS_OUTPUT.PUT_LINE('Employee ID: ' || :OLD.id);  
  
DBMS_OUTPUT.PUT_LINE('Old Skills: ');  
FOR i IN 1..:OLD.skills.COUNT LOOP  
    DBMS_OUTPUT.PUT_LINE(' - ' || :OLD.skills(i));  
END LOOP;  
  
DBMS_OUTPUT.PUT_LINE('New Skills: ');  
FOR i IN 1..:NEW.skills.COUNT LOOP  
    DBMS_OUTPUT.PUT_LINE(' - ' || :NEW.skills(i));  
END LOOP;  
  
DBMS_OUTPUT.PUT_LINE('Change Action: ' || v_action);  
DBMS_OUTPUT.PUT_LINE('Change Timestamp: ' || SYSTIMESTAMP);  
END IF;  
END;
```

At the bottom, the 'Script Output' pane shows the message: 'Trigger AFTER_EMPLOYEE_UPDATE compiled'. The 'Query Result' pane is empty. The status bar at the bottom indicates 'Task completed in 0.077 seconds'.

Connections x | ...sql | ShreejanaDB2.sql x | ShreejanaDB2.sql x | Welcome Page x | ShreejanaDB x

0.077 seconds

ShreejanaDB

Oracle Connections

- ShreejanaDB
 - Tables (Filter)
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues
 - Queues Task
 - Triggers
 - Types
 - Sequences
 - Materialized Views
 - Materialized Views

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Worksheet | Query Builder

```
-- QUESTION 4: creating trigger that will insert a row whenever an update to
-- the employees' skill are made
CREATE OR REPLACE TRIGGER after_employee_update
AFTER UPDATE ON employees_321
FOR EACH ROW
DECLARE
    v_action VARCHAR2(50) := 'UPDATE';
BEGIN
    IF :OLD.skills IS NOT NULL AND :NEW.skills IS NOT NULL AND :OLD.skills != :NEW.skills
    -- Insert a new record in the audit table
    INSERT INTO employees_audit (
        employee_id,
        action,
        old_skills,
        new_skills,
        changed_at
    )
    VALUES (
        :OLD.id,
        v_action,
        :OLD.skills,
        :NEW.skills,
        SYSTIMESTAMP
    );
END;
```

Script Output x | Query Result x

Task completed in 0.077 seconds

Trigger AFTER_EMPLOYEE_UPDATE compiled

execution of trigger while doing update to the record

The screenshot displays the SQL Developer interface. The left sidebar shows the 'Connections' pane with 'ShreejanaDB' selected. The main window is divided into a 'Worksheet' and a 'Query Builder'. The 'Worksheet' contains a PL/SQL script for a trigger. The 'Query Builder' shows a query result table with 2 rows and 5 columns: ID, NAME, POSITION, SALARY, and SKILLS.

```
DBMS_OUTPUT.PUT_LINE(' - ' || :OLD.skills(i));
END LOOP;

DBMS_OUTPUT.PUT_LINE('New Skills: ');
FOR i IN 1..:NEW.skills.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(' - ' || :NEW.skills(i));
END LOOP;

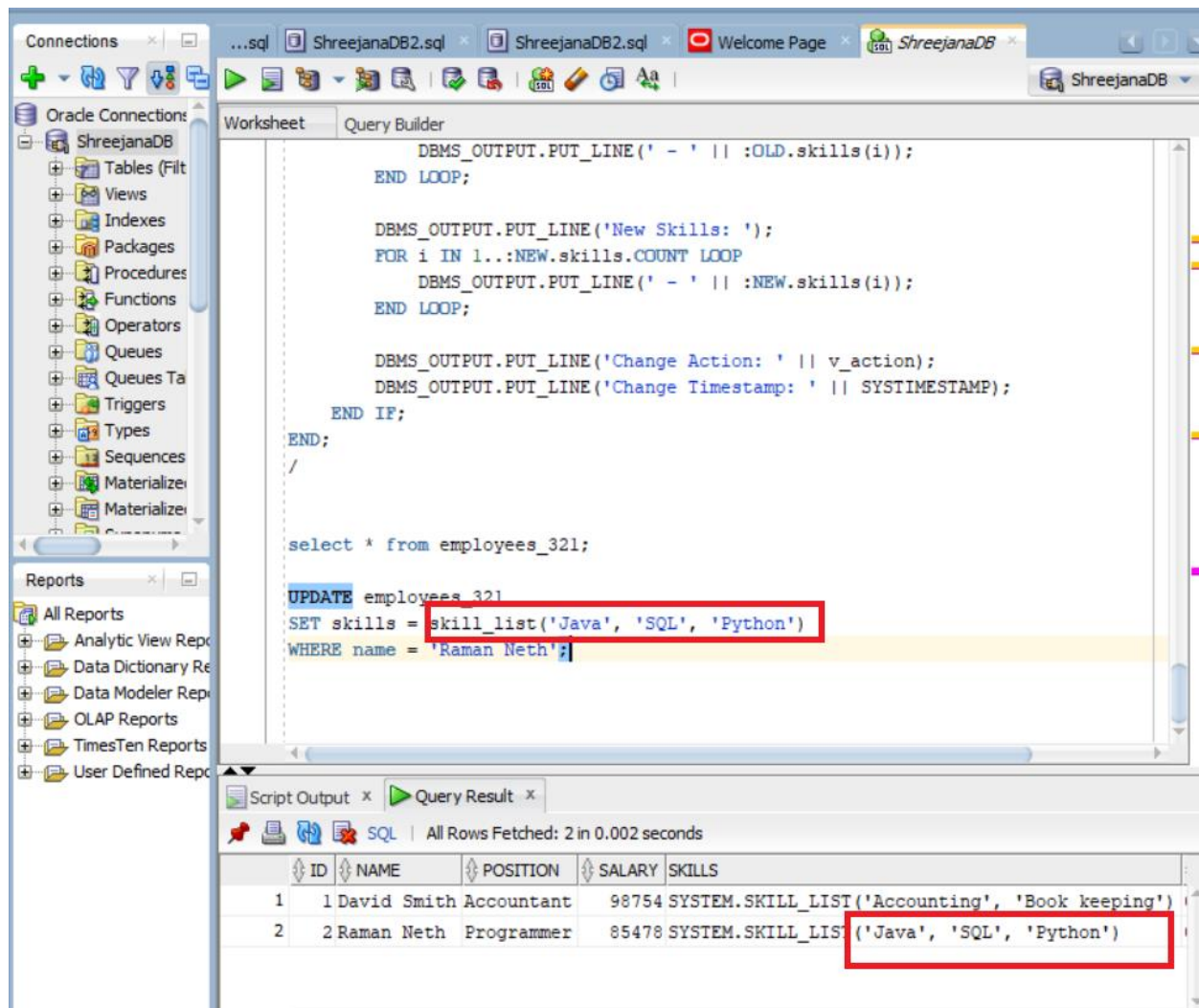
DBMS_OUTPUT.PUT_LINE('Change Action: ' || v_action);
DBMS_OUTPUT.PUT_LINE('Change Timestamp: ' || SYSTIMESTAMP);
END IF;
END;
/

select * from employees_321;

UPDATE employees_321
SET skills = skill_list('Java', 'SQL', 'Python')
WHERE name = 'Raman Neth';
```

ID	NAME	POSITION	SALARY	SKILLS
1	David Smith	Accountant	98754	SYSTEM.SKILL_LIST('Accounting', 'Book keeping')
2	Raman Neth	Programmer	85478	SYSTEM.SKILL_LIST('Python', 'React')

Output after the update of the record



The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree with 'ShreejanaDB' selected. The main window shows a PL/SQL script in the 'Query Builder' tab. The script includes a loop to output old skills, a loop to output new skills, and an update statement for the 'employees_321' table. The update statement is highlighted with a red box. Below the script, the 'Script Output' pane shows the results of the query, which is a table with columns ID, NAME, POSITION, SALARY, and SKILLS. The table contains two rows: one for David Smith and one for Raman Neth. The SKILLS column for Raman Neth is highlighted with a red box.

```
DBMS_OUTPUT.PUT_LINE(' - ' || :OLD.skills(i));
END LOOP;

DBMS_OUTPUT.PUT_LINE('New Skills: ');
FOR i IN 1..:NEW.skills.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(' - ' || :NEW.skills(i));
END LOOP;

DBMS_OUTPUT.PUT_LINE('Change Action: ' || v_action);
DBMS_OUTPUT.PUT_LINE('Change Timestamp: ' || SYSTIMESTAMP);
END IF;
END;
/

select * from employees_321;

UPDATE employees_321
SET skills = skill_list('Java', 'SQL', 'Python')
WHERE name = 'Raman Neth';
```

ID	NAME	POSITION	SALARY	SKILLS
1	David Smith	Accountant	98754	SYSTEM.SKILL_LIST('Accounting', 'Book keeping')
2	Raman Neth	Programmer	85478	SYSTEM.SKILL_LIST('Java', 'SQL', 'Python')

```
-- EXPLANATION
/* A row level trigger for employees_321 has been created that would fire after
   updating operation is performed on the table.

   Once the update query is run for the employee with name Raman Neth, the above
   trigger after_employee_update is fired and it will display the results
   updating the records
*/
```


Connections x | ShreejanaDB2.sql x | ShreejanaDB2.sql x | ShreejanaDB2.sql x | ShreejanaDB2.sql x | Welcome Page x

Oracle Connections
ShreejanaDB
Tables (Filter)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized Views

Worksheet | Query Builder

Once the update query is run for the employee with name Raman Neth, the above trigger after_employee_update is fired and it will display the results updating the records

```
*/  
  
select * from employees_321;  
  
UPDATE employees_321  
SET skills = skill_list('Java', 'SQL', 'Python')  
WHERE name = 'Raman Neth';
```

Script Output x | Query Result x

Task completed in 0.034 seconds

Employee ID: 2
Old Skills:
- Python
- React
New Skills:
- Java
- SQL
- Python
Change Action: UPDATE
Change Timestamp: 24-07-09 11:08:21.225000000 -04:00

1 row updated.