IT3020
Database Systems
3rd Year, 2nd Semester

Laboratory Worksheet 03

Submitted to

Sri Lanka Institute of Information Technology

IT Number: IT21172878

Name: D.P. Liyanagama

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

2024/03/07

1. **The tables of information shown below are the same as in Practical 1. Create suitable object types and tables for recording this information.**
   Define address as an object type consisting of street number, street name, suburb, state and pin. Use a VARRAY type for exchanges in the Stocks table. Create INVESTMENTS as a nested table type in the object type of the Clients table.
   You may include client number and make it the primary key in the clients table, or treat (firstname, lastname) as the primary key. Though names are in general not unique, it is ok for this exercise. Make the names of these tables different from the ones you used in Practical 1.

```
CREATE TYPE exchange_varray AS VARRAY(3) OF VARCHAR(20)
/
CREATE TYPE stock_type AS OBJECT(
    companyName VARCHAR2(15),
    currentPrice NUMBER(6,2),
    exchanges exchange_varray,
    lastDividend NUMBER(4,2),
    eps NUMBER(4,2)
)
/
CREATE TYPE address_type AS OBJECT(
    streetNumber CHAR(10),
    streetName CHAR(15),
    suburb CHAR(20),
    state CHAR(15),
    pin CHAR(10)
)
/
CREATE TYPE investment_type AS OBJECT(
    company REF stock_type,
    purchasePrice NUMBER(6,2),
    purchaseDate DATE,
    qty NUMBER(10)
)
/

CREATE TYPE investment_list AS TABLE OF investment_type
/
```

```
CREATE TYPE client_type AS OBJECT(
    name VARCHAR(20),
    address address_type,
    investment investment_list
)
/
```

2. Create the required types and object tables in your database, and insert the given sample data.

```
CREATE TABLE stock_table OF stock_type(
CONSTRAINT stock_PK PRIMARY KEY(companyName)
)
/
CREATE TABLE client_table OF client_type(
CONSTRAINT client_PK PRIMARY KEY(name)
)
NESTED TABLE investment STORE AS investment_table
/
ALTER TABLE investment_table
ADD SCOPE FOR(company) IS stock_table
/

--insert data to stock table
INSERT INTO stock_table
VALUES(stock_type('BHP',10.50,exchange_varray('Sydney','New
York'),1.50,3.20))
/
INSERT INTO stock_table
VALUES(stock_type('IBM',70.00,exchange_varray('New
York','London','Tokyo'),4.25,10.00))
/
INSERT INTO stock_table
VALUES(stock_type('INTEL',76.50,exchange_varray('New
York','London'),5.00,12.40))
/
```

```
INSERT INTO stock_table
VALUES(stock_type('FORD',40.00,exchange_varray('New
York'),2.00,8.50))
/
INSERT INTO stock_table
VALUES(stock_type('GM',60.00,exchange_varray('New York'),2.50,9.20))
/
INSERT INTO stock_table
VALUES(stock_type('INFOSYS',45.00,exchange_varray('New
York'),3.00,7.80))
/

--insert data to client table
INSERT INTO client_table VALUES(client_type('John
Smith',address_type('3','East Av','Bentley','WA','6102'),
   investment_list(
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'BHP'),12.00,'02-OCT-2001',1000),
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'BHP'),10.50,'08-JUN-2002',2000),
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'IBM'),58.00,'12-FEB-2000',500),
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'IBM'),65.00,'10-APR-2001',1200),
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'INFOSYS'),64.00,'11-AUG-2001',1000)
      )
   )
)
/
INSERT INTO client_table VALUES(client_type('Jill
Brody',address_type('42','Bent St','Perth','WA','6001'),
   investment_list(
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'INTEL'),35.00,'30-JAN-2000',300),
      investment_type((SELECT REF(s) FROM stock_table s WHERE
s.companyName = 'INTEL'),54.00,'30-JAN-2001',400),
```

```
        investment_type((SELECT REF(s) FROM stock_table s WHERE
    s.companyName = 'INTEL'),60.00,'2-OCT-2001',200),
        investment_type((SELECT REF(s) FROM stock_table s WHERE
    s.companyName = 'FORD'),40.00,'5-OCT-1999',300),
        investment_type((SELECT REF(s) FROM stock_table s WHERE
    s.companyName = 'GM'),55.50,'12-DEC-2000',500)
        )
      )
    )
    /
```

3. The queries given below are the same as in Practical 1. Answer the queries using the object relational tables you have created in the previous step. Use dot expressions instead of joins wherever possible.

a) For each client, get the client's name, and the list of the client's investments with stock name, current price, last dividend and earnings per share.

```
SELECT DISTINCT c.name, i.company.companyName, i.company.currentPrice,
i.company.lastDividend, i.company.eps
FROM client_table c, TABLE(c.investment) i
/
```

b) Get the list of all clients and their share investments, showing the client name, and for each stock held by the client, the name of the stock, total number of shares held, and the average purchase price paid by the client for the stock. Average price is the total purchase value paid by a client for a given stock (value=qty*price) divided by the total quantity held by the client.

```
SELECT c.name, i.company.companyName, SUM(i.qty) total_qty,
SUM(i.qty*i.purchasePrice)/SUM(i.qty) Average_Price
FROM client_table c, TABLE(c.investment) i
GROUP BY c.name, i.company.companyName
/
```

c) For each stock traded in New York, find the quantity held by each client, and its current value (value=qty*price). Display stock (company) name, client name, number of shares held, and the current value of the shares.

```
SELECT c.name, i.company.companyName, SUM(i.qty),
SUM(i.qty*i.company.currentPrice) current_value, i.company.currentPrice
FROM client_table c, TABLE(c.investment) i, TABLE(i.company.exchanges) e
WHERE e.column_value = 'New York'
```

```
GROUP BY c.name, i.company.companyName,i.company.currentPrice
/
```

d) Find the total purchase value of investments for all clients. Display client name and total purchase value of the client's portfolio.

```
SELECT c.name, SUM(i.qty*i.purchasePrice) total_price
FROM client_table c, TABLE(c.investment) i
GROUP BY c.name
/
```

e) For each client, list the book profit (or loss) on the total share investment. Book profit is the total value of all stocks based on the current prices less the total amount paid for purchasing them.

```
SELECT c.name, SUM(i.qty * (i.company.currentPrice - i.purchasePrice))
book_profit
FROM client_table c, TABLE(c.investment) i
GROUP BY c.name
/
```

4. Suppose John sold all his INFOSYS stocks to Jill, and Jill sold all her GM stocks to John today at the current prices. Update the database for these two transactions. Use the query 3(a) to check if the update worked correctly. (Here, the term update does not necessarily mean using the update statement of SQL.)

```
INSERT INTO TABLE (SELECT c.investment
    FROM client_table c
    WHERE c.name = 'Jill Brody')
SELECT i.company,i.company.currentPrice,sysdate,i.qty
FROM client_table c, TABLE(c.investment) i
WHERE c.name = 'John Smith' AND i.company.companyName = 'INFOSYS'
/
INSERT INTO TABLE (SELECT c.investment
    FROM client_table c
    WHERE c.name = 'John Smith')
SELECT i.company,i.company.currentPrice,sysdate,i.qty
FROM client_table c, TABLE(c.investment) i
WHERE c.name = 'Jill Brody' AND i.company.companyName = 'GM'
/
DELETE TABLE (SELECT c.investment
    FROM client_table c WHERE c.name = 'Jill Brody') i
WHERE i.company.companyName = 'GM'
/
```

```
DELETE TABLE (SELECT c.investment
     FROM client_table c WHERE  c.name = 'John Smith') i
WHERE i.company.companyName = 'INFOSYS'
/
```