**BÀI TẬP THỰC HÀNH**

**1.1 Creating object types and object-relational tables**

CREATE TABLE people (name VARCHAR2(30), phone VARCHAR2(20));

CREATE TYPE person AS OBJECT (name VARCHAR2(30), phone VARCHAR2(20));/

CREATE TABLE person\_table OF person;

Select \* from people

Select p.last\_name, p.addr.city from person\_table p

DESCRIBE person\_table

**1.2 Inserting Values**

--CAU 3

INSERT INTO people VALUES ('Le Van A', '1-800-555-1212');

INSERT INTO people VALUES ('Nguyen Van B', '1-700-555-1414');/

INSERT INTO people VALUES ('Nguyen Van C', '0977586539');/

INSERT INTO people VALUES ('Le Van D', '036278489');/

INSERT INTO people VALUES ('Tran Van Quan', '0976587780');/

INSERT INTO person\_table VALUES ('John Smith', '1-800-555-1212' );

INSERT INTO person\_table VALUES ('Tran Van A', '0978586532');/

INSERT INTO person\_table VALUES ('Nguyen Bao Ngoc', '0977586538');/

INSERT INTO person\_table VALUES ('Le Anh Quan', '036278456');/

INSERT INTO person\_table VALUES ('Nguyen Phuong Nga', '0979588539');/

--CAU 4

CREATE OR REPLACE TYPE job AS OBJECT(job\_ID VARCHAR(20), jobtitle VARCHAR(20), salary\_amount INT, years\_of\_experience INT);/

CREATE TABLE job\_table OF job;

INSERT INTO job\_table VALUES (job('CV1', 'MC', 3000000, 10));/

INSERT INTO job\_table VALUES (job('CV2', 'Accounting', 1000000, 5));/

INSERT INTO job\_table VALUES (job('CV3', 'Singer', 5000000, 15));/

INSERT INTO job\_table VALUES (job('CV4', 'Developer', 10000000, 20));/

INSERT INTO job\_table VALUES (job('CV5', 'Actor', 7000000, 12));/

Select jb.\* from job\_table jb

**1.3 Select statements**

--CAU 5

SELECT p.name, p.phone FROM person\_table p

WHERE p.name = 'Nguyen Phuong Nga';

SELECT VALUE(p) FROM person\_table p

WHERE p.name = 'Nguyen Phuong Nga';

SELECT \* FROM person\_table p

WHERE p.name = 'Nguyen Phuong Nga';

SELECT name FROM person\_table p

WHERE p.name = 'Nguyen Phuong Nga';

**1.4 Object types as user-defined datatypes**

--CAU 6

CREATE TYPE street AS OBJECT (snumber NUMBER, sname VARCHAR2(30), flat\_number NUMBER);/

CREATE OR REPLACE TYPE address AS OBJECT (street\_and\_number street, city VARCHAR2(30), postal\_code VARCHAR2(8), country VARCHAR2(30));/

**1.5 Dropping types and tables**

--CAU 7

DROP TABLE person\_table

DROP TYPE person FORCE

--CAU 8,9,10

CREATE OR REPLACE TYPE person AS OBJECT (first\_name VARCHAR2(30), middle\_initial VARCHAR2(20), last\_name VARCHAR2(20), phone phone\_typ, addr address);/

CREATE OR REPLACE TYPE phone\_typ AS OBJECT (business VARCHAR2(30), home VARCHAR2(20), mobile VARCHAR2(12));/

CREATE TABLE person\_table OF person;

INSERT INTO person\_table VALUES ('Nguyen', 'Van', 'A', phone\_typ('7786-987-456', '0987654890', '0367543890'), address(street(1, 'Le Viet Thuat', 30), 'Vinh', '7654879', 'Nghe An'));/

INSERT INTO person\_table VALUES ('Nguyen', 'Van', 'B', phone\_typ('7787-999-456', '0987654567', '0367543567'), address(street(2, 'Nguyen Van Cu', 10), 'Vinh', '5467378', 'Nghe An'));/

INSERT INTO person\_table VALUES ('Nguyen', 'Van', 'C', phone\_typ('6545-987-789', '0956732682', '0945728165'), address(street(3, 'Nguyen Truong Thi', 12), 'HN', '8974928', 'HN'));/

INSERT INTO person\_table VALUES ('Nguyen', 'Van', 'D', phone\_typ('3425-646-342', '0784536536', '0923536743'), address(street(4, 'Le Duan', 02), 'Ha Noi', '56348924', 'HN'));/

INSERT INTO person\_table VALUES ('Nguyen', 'Van', 'E', phone\_typ('5424-643-231', '0345728165', '0983627582'), address(street(5, 'Phan Boi Chau', 19), 'Ho Chi Minh', '82645174', 'HCM'));/

**2.2 Subtypes**

-- CAU 11, 12

ALTER TYPE person NOT FINAL CASCADE;

CREATE TYPE employee UNDER person (emp\_ID INT) NOT FINAL ;/

CREATE TABLE employee\_table OF employee;

INSERT INTO employee\_table VALUES ('Nguyen', 'Van', 'A', phone\_typ('7786-987-456', '0987654890', '0367543890'), address(street(1, 'Tran Phu', 30), 'Hue', '7654879', 'Hue'), 100);/

INSERT INTO employee\_table VALUES ('Nguyen', 'Van', 'B', phone\_typ('7786-987-456', '0987654890', '0367543890'), address(street(2, 'Kim Dong', 30), 'Vinh', '7654879', 'Nghe An'), 101);/

INSERT INTO employee\_table VALUES ('Nguyen', 'Van', 'C', phone\_typ('7786-987-456', '0987654890', '0367543890'), address(street(3, 'Phan Chau Trinh', 30), 'Da Nang', '7654879', 'Da Nang'), 102);/

SELECT value(p).first\_name, value(p).middle\_initial, value(p).last\_name, value(p).addr.street\_and\_number.sname, value(p).addr.city, value(p).phone.home

FROM employee\_table p

WHERE value(p) IS OF (person);

-- CAU 13

select SYS\_NC\_OID$ from person\_table;

ALTER TABLE job\_table

ADD (CONSTRAINT jobID PRIMARY KEY (job\_ID));

ALTER TABLE employee\_table

ADD (CONSTRAINT empID PRIMARY KEY (emp\_ID));

**2.4 References or REFs**

-- CAU 14, 15, 17, 18, 19

CREATE TABLE employment (employee REF employee SCOPE IS employee\_table, position REF job SCOPE IS job\_table);

INSERT INTO employment

SELECT REF(e), REF(j)

FROM job\_table j, employee\_table e

WHERE e.emp\_ID = 101

AND j.job\_ID = 'CV1';

INSERT INTO employment

SELECT REF(e), REF(j)

FROM job\_table j, employee\_table e

WHERE e.emp\_ID = 100

AND j.job\_ID = 'CV2';

SELECT emp.employee.first\_name, emp.employee.middle\_initial, emp.employee.last\_name

FROM employment emp;

SELECT emp.employee.first\_name, emp.employee.middle\_initial, emp.employee.last\_name

FROM employment emp

WHERE emp.position.salary\_amount > 20000;

SELECT emp.employee.first\_name, emp.employee.middle\_initial, emp.employee.last\_name

FROM employment emp

WHERE emp.employee.addr.city = 'Hue';

CREATE OR REPLACE TYPE project AS OBJECT (project\_id NUMBER, project\_leader REF person, project\_title VARCHAR2(30));/

CREATE TABLE project\_table OF project;

ALTER TABLE project\_table ADD (CONSTRAINT projectID PRIMARY KEY (project\_id ));

INSERT INTO project\_table VALUES (1, (SELECT REF(e) FROM person\_table e WHERE e.last\_name= 'B'), 'Quan Ly Thu Tu');/

INSERT INTO project\_table VALUES (2, (SELECT REF(e) FROM person\_table e WHERE e.last\_name= 'A'), 'Quan Ly Thuoc');/

INSERT INTO project\_table VALUES (3, (SELECT REF(e) FROM person\_table e WHERE e.last\_name= 'C'), 'Data Mining');/

CREATE TABLE project\_membership (project\_id REF project SCOPE IS project\_table, member REF person SCOPE IS employee\_table);

INSERT INTO project\_membership

SELECT REF(p), REF(e)

FROM project\_table p, employee\_table e

WHERE p.project\_id = 1 AND e.emp\_ID = 100;

INSERT INTO project\_membership

SELECT REF(p), REF(e)

FROM project\_table p, employee\_table e

WHERE p.project\_id = 2 AND e.emp\_ID = 101;

INSERT INTO project\_membership

SELECT REF(p), REF(e)

FROM project\_table p, employee\_table e

WHERE p.project\_id = 3 AND e.emp\_ID = 102;

SELECT p.member.first\_name, p.member.middle\_initial, p.member.last\_name, p.project\_id.project\_title FROM project\_membership p

**3.2. Member Methods**

CREATE OR REPLACE TYPE job2 AS OBJECT (job\_id int, jobtitle varchar(20), salary\_amount int, years\_of\_experience int, MEMBER FUNCTION evaluate\_qualification RETURN STRING);

CREATE OR REPLACE TYPE BODY job2 AS

MEMBER FUNCTION evaluate\_qualification RETURN STRING IS

BEGIN

IF self.years\_of\_experience < 2 THEN

RETURN 'Too bad';

ELSIF self.years\_of\_experience = 2 THEN

RETURN 'OK';

ELSE

RETURN 'Great!';

END IF;

END evaluate\_qualification;

END;

**CÂU 20:**

ALTER TYPE job2

ADD MEMBER FUNCTION salary\_fraction(N REAL) RETURN REAL

CASCADE;

CREATE OR REPLACE TYPE BODY job2 AS

MEMBER FUNCTION evaluate\_qualification RETURN STRING IS

BEGIN

IF self.years\_of\_experience < 2 THEN

RETURN 'Too bad';

ELSIF self.years\_of\_experience = 2 THEN

RETURN 'OK';

ELSE

RETURN 'Great!';

END IF;

END evaluate\_qualification;

MEMBER FUNCTION salary\_fraction(N REAL) RETURN REAL IS

BEGIN

RETURN (self.salary\_amount/N);

END salary\_fraction;

END;

**CÂU 21:**

CREATE TABLE job2\_table OF job2;

INSERT INTO job2\_table VALUES (10, 'Teacher', 5000000, 3);/

INSERT INTO job2\_table VALUES (11, 'Lawyer', 7000000, 2);/

INSERT INTO job2\_table VALUES (12, 'Electrician', 100000, 2);/

INSERT INTO job2\_table VALUES (13, 'IT', 10000000, 5);/

INSERT INTO job2\_table VALUES (14, 'Accounting', 600000, 1);/

SELECT j.evaluate\_qualification(), j.salary\_fraction(12) FROM job2\_table j;

SELECT j.jobtitle FROM job2\_table j

WHERE j.evaluate\_qualification() = 'OK' AND j.salary\_fraction(6) > 10000;

**3.3. Adding methods to existing types**

**CÂU 22:**

ALTER TYPE person

ADD MEMBER FUNCTION print\_name RETURN STRING CASCADE;

CREATE OR REPLACE TYPE BODY person AS

MEMBER FUNCTION print\_name RETURN STRING IS

BEGIN

RETURN self.last\_name || ', ' || self.first\_name || ' ' || self.middle\_initial;

END print\_name;

END;

SELECT p.print\_name() FROM person\_table p;

**CÂU 23:**

ALTER TYPE person

ADD MEMBER FUNCTION count\_numberphone RETURN NUMBER

CASCADE;

CREATE OR REPLACE TYPE BODY person AS

MEMBER FUNCTION print\_name RETURN STRING IS

BEGIN

RETURN self.last\_name || ', ' || self.first\_name || ' ' || self.middle\_initial;

END print\_name;

MEMBER FUNCTION count\_numberphone RETURN NUMBER IS

N REAL;

BEGIN

N := 0;

IF self.phone.business IS NOT NULL THEN

N:= N + 1;

END IF;

IF self.phone.home IS NOT NULL THEN

N:= N + 1;

END IF;

IF self.phone.mobile IS NOT NULL THEN

N:= N + 1;

END IF;

RETURN N;

END count\_numberphone;

END;

SELECT p.print\_name(), p.count\_numberphone()

FROM person\_table p;

**3.4. Map Methods**

ALTER TYPE job

ADD MAP MEMBER FUNCTION sorting RETURN NUMBER

CASCADE;

CREATE OR REPLACE TYPE BODY job AS

MAP MEMBER FUNCTION sorting RETURN NUMBER IS

BEGIN

RETURN self.years\_of\_experience \* self.salary\_amount;

END sorting;

END;

SELECT \* FROM job\_table j ORDER BY j.sorting() DESC;

**CÂU 24:**

ALTER TYPE person

ADD MAP MEMBER FUNCTION orderBy\_name RETURN STRING

CASCADE;

CREATE OR REPLACE TYPE BODY person AS

MEMBER FUNCTION print\_name RETURN STRING IS

BEGIN

RETURN self.last\_name || ', ' || self.first\_name || ' ' || self.middle\_initial;

END print\_name;

MEMBER FUNCTION count\_numberphone RETURN NUMBER IS

N REAL;

BEGIN

N := 0;

IF self.phone.business IS NOT NULL THEN

N:= N + 1;

END IF;

IF self.phone.home IS NOT NULL THEN

N:= N + 1;

END IF;

IF self.phone.mobile IS NOT NULL THEN

N:= N + 1;

END IF;

RETURN N;

END count\_numberphone;

MAP MEMBER FUNCTION orderBy\_name RETURN STRING IS

BEGIN

RETURN self.last\_name || ', ' || self.first\_name || ' ' || self.middle\_initial;

END orderBy\_name;

END;

SELECT p.orderBy\_name()

FROM person\_table p ORDER BY p.orderBy\_name() ASC;

**4.1. About attribute and type names**

CREATE TYPE somename (person person);

**4.2. Collection Types**

CREATE TYPE phone\_array AS VARRAY(10) OF VARCHAR2(12);

CREATE TABLE company1 (name VARCHAR2(20), phone phone\_array);

INSERT INTO company1 VALUES ('abc', phone\_array('243-4758','485-2534'));

CREATE TYPE phone\_nested AS TABLE OF VARCHAR2(12);

CREATE TABLE company2 (name VARCHAR2(20), phone phone\_nested)

NESTED TABLE phone STORE AS phone\_nr\_table;

INSERT into company2 VALUES ('abc', phone\_nested('243-4758','485-2534'));

SELECT c.name, p.\* FROM company1 c, TABLE(c.phone) p;

SELECT c.name, p.\* FROM company2 c, TABLE(c.phone) p;

**Exercises 25:**

CREATE TYPE phone\_code AS OBJECT(area\_code VARCHAR2(20), local\_number VARCHAR2(20));

CREATE TYPE phone\_numbers AS TABLE OF phone\_code;

CREATE TABLE company3 (name VARCHAR2(20), phone phone\_numbers)

NESTED TABLE phone STORE AS phone\_nested\_table;

**Exercises 26:**

INSERT INTO company3 VALUES ('abc', phone\_numbers(phone\_code('0131','243-4758'), phone\_code('0131','485-2534')));

INSERT INTO company3 VALUES ('def', phone\_numbers(phone\_code('0132','243-4768'), phone\_code('0130','485-6782')));

INSERT INTO company3 VALUES ('ghi', phone\_numbers(phone\_code('0133','243-4798'), phone\_code('0129','485-8263')));

SELECT c.name, p.\* FROM company3 c, TABLE(c.phone) p;

**4.3. Multi-level collection types and other advanced features**

**4.4. Select statements for nested tables**

SELECT \* FROM company1;

SELECT c.name, t.\* FROM company1 c, TABLE(c.phone) t;

SELECT name, t.\* FROM company1, TABLE(SELECT phone FROM company1) t;

SELECT t.COLUMN\_VALUE from company1 c, TABLE(c.phone) t;

**Exercises 28:**

SELECT c.name, t.local\_number FROM company3 c, TABLE(c.phone) t;

SELECT c.name, t.area\_code FROM company3 c, TABLE(c.phone) t WHERE t.area\_code = '0133';

SELECT t.local\_number FROM company3 c, TABLE(c.phone) t;

**5.1. Ordinary Relations and Object Relations**

CREATE TABLE people (ID number PRIMARY KEY, name VARCHAR2(30), phone VARCHAR2(20));

INSERT into people VALUES (0, 'Smith', '546-4364');

INSERT into people VALUES (1, 'Miller', '556-4374');

INSERT into people VALUES (2, 'Jones', '536-4386');

CREATE TYPE phone\_nested AS TABLE OF VARCHAR2(12);

CREATE TYPE people\_type AS OBJECT (person\_ID number, name VARCHAR2(30), phone\_list phone\_nested);

CREATE TABLE people\_object\_table OF people\_type

NESTED TABLE phone\_list STORE AS p\_table;

INSERT INTO people\_object\_table

SELECT ID, name, phone\_nested(phone) FROM people;

CREATE VIEW people\_object\_view OF people\_type WITH OBJECT IDENTIFIER (person\_ID) AS

SELECT ID, name, phone\_nested(phone) AS phone\_list from people;

SELECT p.person\_ID, p.name, ph.\* FROM people\_object\_view p, TABLE(p.phone\_list) ph;

**Exercises 29:**

CREATE TABLE department (dno NUMBER PRIMARY KEY, dname VARCHAR2(20), dstreet VARCHAR2(30), dstreetnumber VARCHAR2(20), dcity VARCHAR2(30), dpostalcode VARCHAR2(8));/

CREATE OR REPLACE TYPE address\_typ AS OBJECT (dstreet VARCHAR2(30), dstreetnumber VARCHAR2(20), dcity VARCHAR2(30), dpostalcode VARCHAR2(8));/

CREATE OR REPLACE TYPE dept\_type AS OBJECT (deptno NUMBER, deptname VARCHAR2(20), deptaddress address\_typ);/

CREATE VIEW dept\_view OF dept\_type WITH OBJECT IDENTIFIER (deptno) AS

SELECT dno, dname , address\_typ(d.dstreet, d.dstreetnumber, d.dcity, d.dpostalcode) AS address\_list from department d;

**Exercises 30:**

INSERT INTO department VALUES (1, 'Tran Thi A', 'Nguyen Van Cu', '02', 'TP.Vinh', '098-8372');/

INSERT INTO department VALUES (2, 'ST', '400 Oracle Pkwy', 'Redwood S', 'CA', '94065');/

INSERT INTO department VALUES (3, 'Apps', '310 Open', 'RedSan', 'TA', '73826');/

INSERT INTO dept\_view VALUES (4, 'Apple', address\_typ('300 Close', 'Footsan', 'HN', '089-2928'));/

SELECT v.deptno, v.deptname, v.deptaddress.dstreet, v.deptaddress.dcity FROM dept\_view v;

**5.2. Using Nested Tables in Views**

CREATE TABLE phone\_nrs (ID number, phone VARCHAR2(20));

INSERT INTO phone\_nrs VALUES (0, '546-4364');

INSERT INTO phone\_nrs VALUES (0, '546-4123');

INSERT INTO phone\_nrs VALUES (1, '556-4374');

INSERT INTO phone\_nrs VALUES (2, '536-4386');

CREATE VIEW people\_object\_view2 OF people\_type WITH OBJECT IDENTIFIER (person\_ID) AS

SELECT p.ID, p.name, CAST(MULTISET (SELECT phone FROM phone\_nrs n

WHERE n.ID = p.ID) AS phone\_nested)

FROM people p;

**Exercises 31:**

CREATE TABLE employees (empID NUMBER PRIMARY KEY, empname VARCHAR2(20), deptno NUMBER REFERENCES department(dno));/

CREATE OR REPLACE TYPE employee\_t AS OBJECT (eID NUMBER, ename VARCHAR2(20));/

CREATE OR REPLACE TYPE employee\_list\_t AS TABLE OF employee\_t ;/

CREATE OR REPLACE TYPE dept\_t AS OBJECT(deptno NUMBER, deptname VARCHAR2(20), address address\_typ, emp\_list employee\_list\_t);/

CREATE VIEW dept\_view1 OF dept\_t WITH OBJECT IDENTIFIER (deptno) AS

SELECT d.dno, d.dname, address\_typ(d.dstreet, d.dstreetnumber, d.dcity, d.dpostalcode) AS address\_list, CAST(MULTISET (SELECT e.empID, e.empname FROM

employees e WHERE e.deptno = d.dno) AS employee\_list\_t) AS emp\_list FROM department d;

INSERT INTO employees VALUES (100, 'John', 1);/

INSERT INTO employees VALUES (200, 'Robert', 2 );/

INSERT INTO employees VALUES (300, 'Mary', 3);/

SELECT dv.deptno, dv.deptname, dv.address.dstreet, e.\* FROM dept\_view1 dv, TABLE(dv.emp\_list) e WHERE dv.deptno = 1;/