Experiment 2

Name : Omkar kore **Class** : BTech C

Roll NO : BTC19 **Batch** : C1

Q1. Create a super type person

```
-- STEP 1: Create the Super Type Person
CREATE OR REPLACE TYPE Person AS OBJECT (
   fname   VARCHAR2(20),
   lname   VARCHAR2(20),
   dob     DATE,

   MEMBER FUNCTION FullName RETURN VARCHAR2,
   MEMBER FUNCTION OnDate RETURN DATE
) NOT FINAL;
/

-- TYPE BODY for Person
CREATE OR REPLACE TYPE BODY Person AS
   MEMBER FUNCTION FullName RETURN VARCHAR2 IS
   BEGIN
      RETURN fname || ' ' || lname;
   END;
   MEMBER FUNCTION OnDate RETURN DATE IS
   BEGIN
      RETURN dob;
   END;
```

```
END;
/

-- STEP 2: Create Sub Type EmpObj
CREATE OR REPLACE TYPE EmpObj UNDER Person (
   job   VARCHAR2(20),
   sal   NUMBER(10,2),
   da    NUMBER(10,2),
   doj   DATE,

   MEMBER FUNCTION Earn RETURN NUMBER,
   OVERRIDING MEMBER FUNCTION OnDate RETURN DATE
);
/

-- TYPE BODY for EmpObj
CREATE OR REPLACE TYPE BODY EmpObj AS
   MEMBER FUNCTION Earn RETURN NUMBER IS
   BEGIN
      RETURN sal + da;
   END;

   OVERRIDING MEMBER FUNCTION OnDate RETURN DATE IS
   BEGIN
      RETURN doj;
   END;
END;
/
```

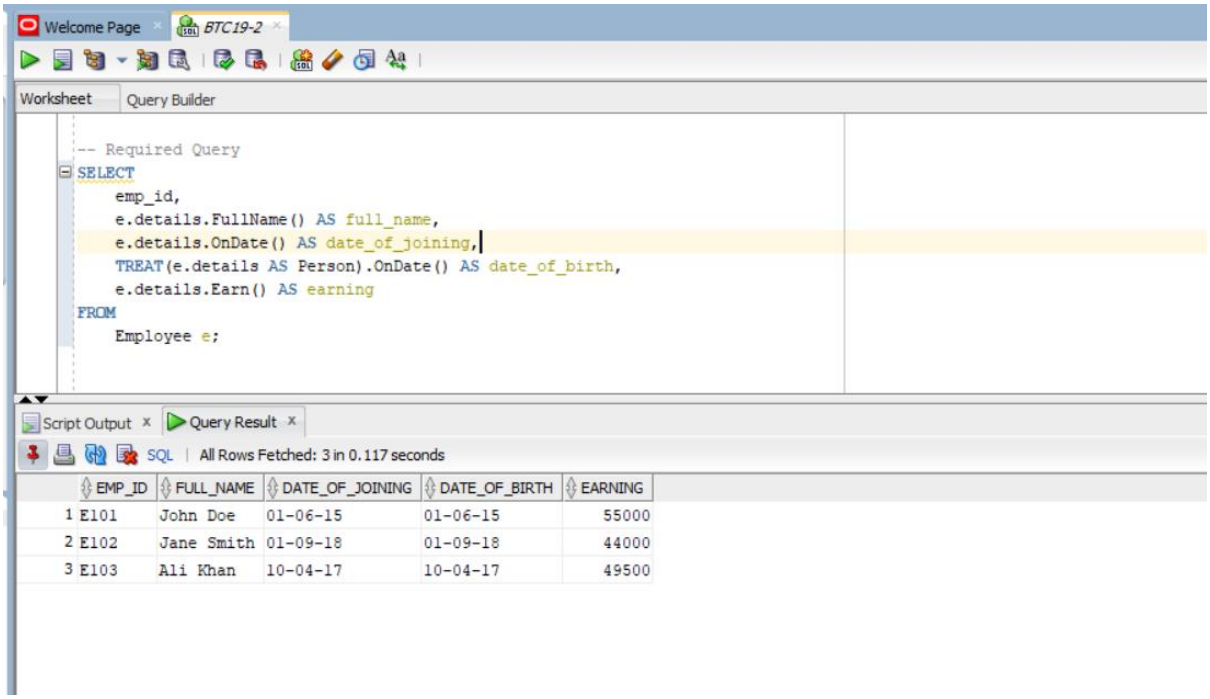
```
-- STEP 3: Create Table to Store Employees
CREATE TABLE Employee (
   emp_id   VARCHAR2(10) PRIMARY KEY,
   details  EmpObj
);

-- STEP 4: Insert Sample Data
INSERT INTO Employee VALUES ('E101',
   EmpObj('John', 'Doe', DATE '1990-01-01', 'Manager', 50000, 5000, DATE '2015-06-01')
);

INSERT INTO Employee VALUES ('E102',
   EmpObj('Jane', 'Smith', DATE '1992-02-14', 'Analyst', 40000, 4000, DATE '2018-09-01')
);

INSERT INTO Employee VALUES ('E103',
   EmpObj('Ali', 'Khan', DATE '1988-07-20', 'Developer', 45000, 4500, DATE '2017-04-10')
);

-- Required Query
SELECT
   emp_id,
   e.details.FullName() AS full_name,
   e.details.OnDate() AS date_of_joining,
   TREAT(e.details AS Person).OnDate() AS date_of_birth,
   e.details.Earn() AS earning
FROM
   Employee e;
```



Q2. Implementing Table Inheritance in SQL Server

```
CREATE SEQUENCE person_seq START WITH 1 INCREMENT BY 1;


CREATE TABLE People (
    person_id INT PRIMARY KEY,
    first_name VARCHAR2(50),
    middle_name VARCHAR2(50),
    last_name VARCHAR2(50),
    birth_date DATE,
    person_type VARCHAR2(10) -- 'Student', 'Teacher', or 'Parent'
);


CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    grade VARCHAR2(10),
    class VARCHAR2(20),
    parent_name VARCHAR2(100),
```

```
   CONSTRAINT fk_student_person FOREIGN KEY (student_id) REFERENCES
People(person_id)
);



CREATE TABLE Teachers (
   teacher_id INT PRIMARY KEY,
   subject VARCHAR2(50),
   employment_date DATE,
   skill_set VARCHAR2(100),
   CONSTRAINT fk_teacher_person FOREIGN KEY (teacher_id) REFERENCES
People(person_id)
);



CREATE TABLE Parents (
   parent_id INT PRIMARY KEY,
   occupation VARCHAR2(50),
   no_of_children INT,
   CONSTRAINT fk_parent_person FOREIGN KEY (parent_id) REFERENCES
People(person_id)
);


INSERT INTO People (person_id, first_name, middle_name, last_name, birth_date,
person_type)
VALUES (person_seq.NEXTVAL, 'Rajesh', NULL, 'Sharma', TO_DATE('1980-05-10',
'YYYY-MM-DD'), 'Parent');


INSERT INTO Parents (parent_id, occupation, no_of_children)
VALUES (person_seq.CURRVAL, 'Engineer', 2);
```

INSERT INTO People (person_id, first_name, middle_name, last_name, birth_date, person_type)

VALUES (person_seq.NEXTVAL, 'Amit', 'Kumar', 'Singh', TO_DATE('2008-08-15', 'YYYY-MM-DD'), 'Student');

INSERT INTO Students (student_id, grade, class, parent_name)

VALUES (person_seq.CURRVAL, '8th', '8A', 'Rajesh Sharma');

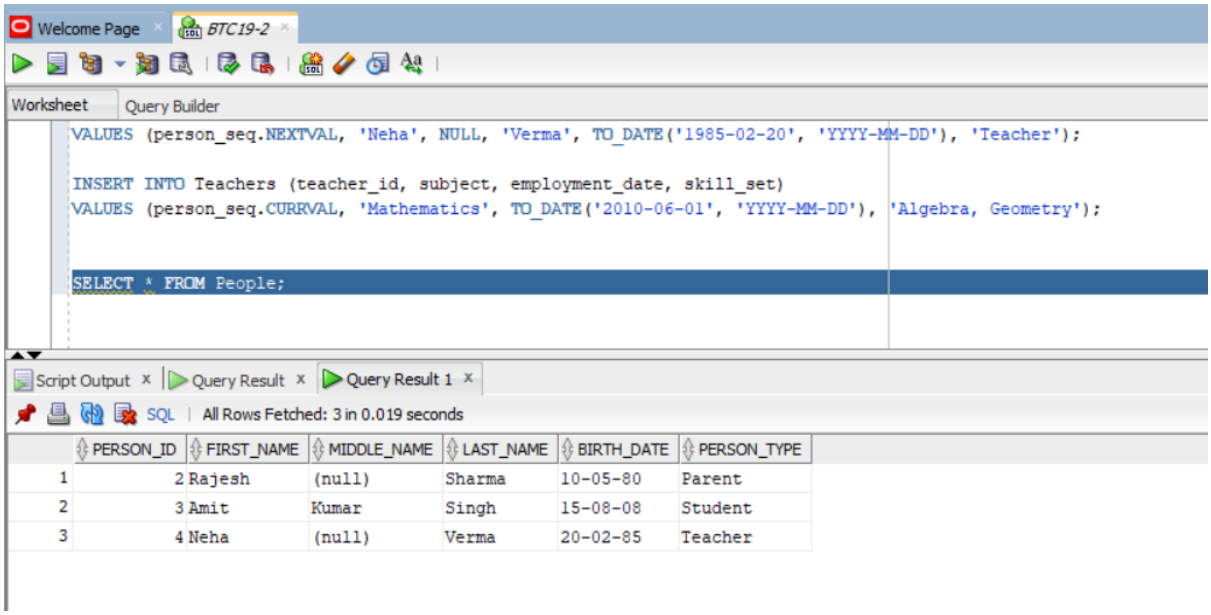
INSERT INTO People (person_id, first_name, middle_name, last_name, birth_date, person_type)

VALUES (person_seq.NEXTVAL, 'Neha', NULL, 'Verma', TO_DATE('1985-02-20', 'YYYY-MM-DD'), 'Teacher');

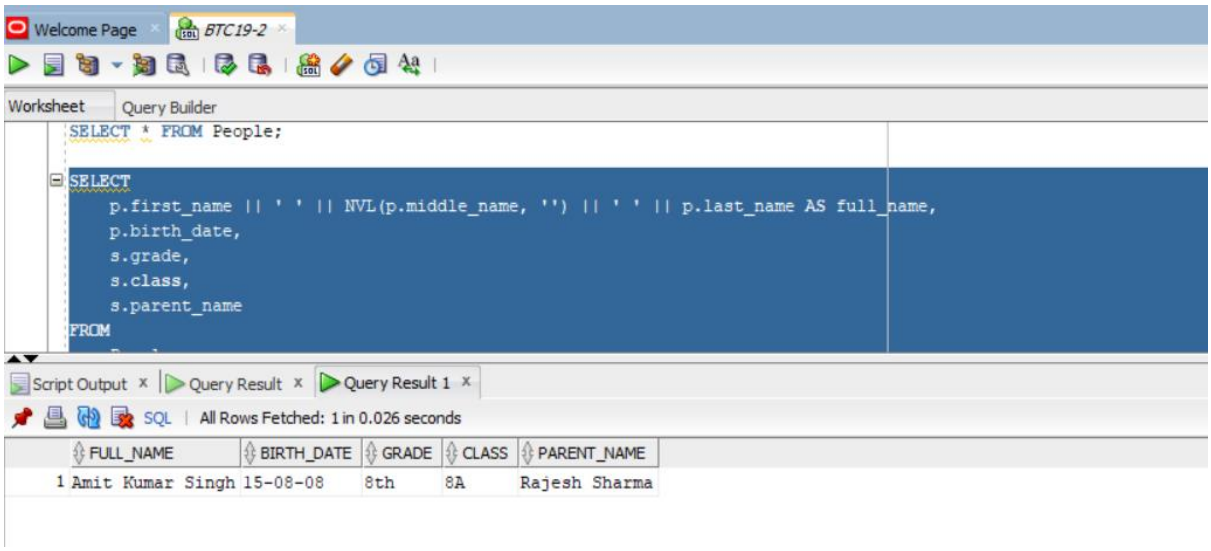
INSERT INTO Teachers (teacher_id, subject, employment_date, skill_set)

VALUES (person_seq.CURRVAL, 'Mathematics', TO_DATE('2010-06-01', 'YYYY-MM-DD'), 'Algebra, Geometry');

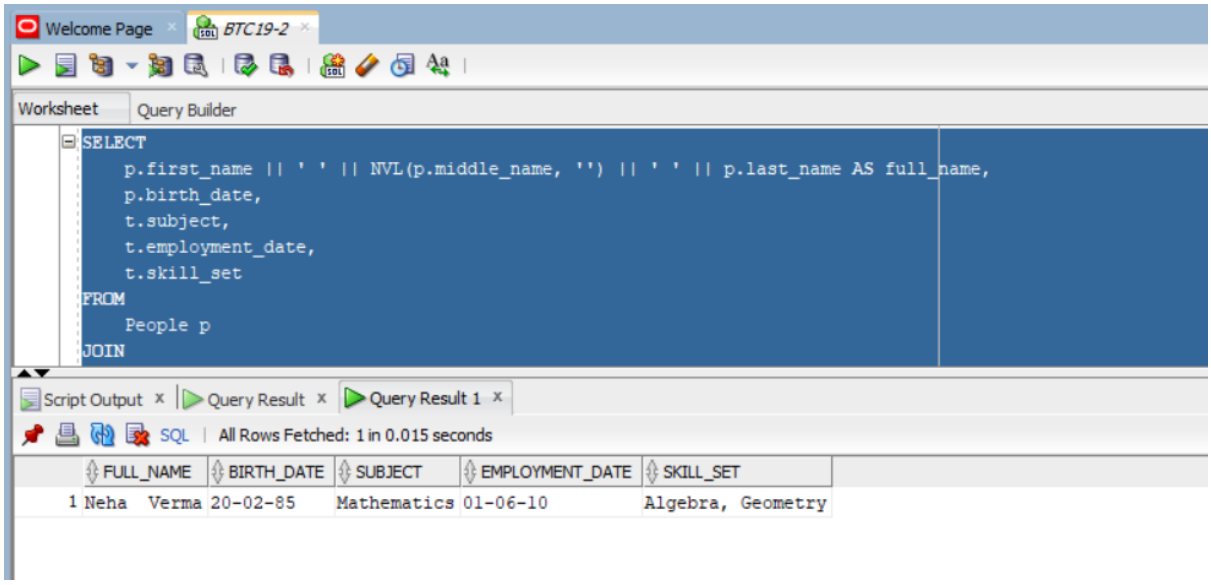
SELECT * FROM People;



```
SELECT
    p.first_name || ' ' || NVL(p.middle_name, '') || ' ' || p.last_name AS full_name,
    p.birth_date,
    s.grade,
    s.class,
    s.parent_name
FROM
    People p
JOIN
    Students s ON p.person_id = s.student_id;
```




```
SELECT
    p.first_name || ' ' || NVL(p.middle_name, '') || ' ' || p.last_name AS full_name,
    p.birth_date,
    t.subject,
    t.employment_date,
    t.skill_set
FROM
    People p
JOIN
    Teachers t ON p.person_id = t.teacher_id;
```




```
SELECT
    p.first_name || ' ' || NVL(p.middle_name, '') || ' ' || p.last_name AS full_name,
    p.birth_date,
    pa.occupation,
    pa.no_of_children
FROM
    People p
JOIN
    Parents pa ON p.person_id = pa.parent_id;
```