PHASE 6

Use Case Example:

> "Allow a voter to cast a vote. The system must verify that the voter is eligible and hasn't already voted in the election."

Procedure to Cast a Vote

```
CREATE OR REPLACE PROCEDURE cast_vote (
    p_voter_id        IN INT,
    p_candidate_id    IN INT,
    p_election_id     IN INT
) AS
    v_eligible CHAR(1);
    v_exists   NUMBER := 0;
BEGIN
    -- Check eligibility
    SELECT Eligibility_Status INTO v_eligible
    FROM Voters
    WHERE Voter_ID = p_voter_id;

    IF v_eligible != 'Y' THEN
        RAISE_APPLICATION_ERROR(-20001, 'Voter is not eligible to vote.');
    END IF;

    -- Check if voter has already voted
```

```
    SELECT COUNT(*) INTO v_exists
    FROM Votes
    WHERE Voter_ID = p_voter_id AND Election_ID = p_election_id;

    IF v_exists > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Voter has already voted in this
election.');
    END IF;

    -- Insert the vote
    INSERT INTO Votes (Vote_ID, Voter_ID, Candidate_ID, Election_ID,
Timestamp)
    VALUES (Votes_seq.NEXTVAL, p_voter_id, p_candidate_id, p_election_id,
SYSTIMESTAMP);

    DBMS_OUTPUT.PUT_LINE('Vote successfully recorded.');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20003, 'Voter not found.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

```
SQL> CREATE OR REPLACE PROCEDURE cast_vote (
  2      p_voter_id        IN INT,
  3      p_candidate_id    IN INT,
  4      p_election_id     IN INT
  5  ) AS
  6      v_eligible CHAR(1);
  7      v_exists   NUMBER := 0;
  8  BEGIN
  9      -- Check eligibility
 10      SELECT Eligibility_Status INTO v_eligible
 11      FROM Voters
 12      WHERE Voter_ID = p_voter_id;
 13
 14      IF v_eligible != 'Y' THEN
 15          RAISE_APPLICATION_ERROR(-20001, 'Voter is not eligible to vote.');
 16      END IF;
 17
 18      -- Check if voter has already voted
 19      SELECT COUNT(*) INTO v_exists
 20      FROM Votes
 21      WHERE Voter_ID = p_voter_id AND Election_ID = p_election_id;
 22
 23      IF v_exists > 0 THEN
 24          RAISE_APPLICATION_ERROR(-20002, 'Voter has already voted in this election.');
 25      END IF;
 26
 27      -- Insert the vote
 28      INSERT INTO Votes (Vote_ID, Voter_ID, Candidate_ID, Election_ID, Timestamp)
 29      VALUES (Votes_seq.NEXTVAL, p_voter_id, p_candidate_id, p_election_id, SYSTIMESTAMP);
 30
 31      DBMS_OUTPUT.PUT_LINE('Vote successfully recorded.');
 32  EXCEPTION
 33      WHEN NO_DATA_FOUND THEN
 34          RAISE_APPLICATION_ERROR(-20003, 'Voter not found.');
 35      WHEN OTHERS THEN
 36          DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
 37  END;
 38  /

Warning: Procedure created with compilation errors.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> EXEC cast_vote(1, 1, 1);
Vote successfully recorded.

PL/SQL procedure successfully completed.

SQL>
```

Supporting Sequence (for Vote_ID)

SQL> CREATE SEQUENCE Votes_seq

START WITH 3

INCREMENT BY 1

NOCACHE

NOCYCLE;

```
SQL> CREATE SEQUENCE Votes_seq
  2  START WITH 3
  3  INCREMENT BY 1
  4  NOCACHE
  5  NOCYCLE;

Sequence created.
```

Function Example – Count Votes for a Candidate

```
SQL> CREATE OR REPLACE FUNCTION count_votes (
  2      p_candidate_id IN INT,
  3      p_election_id  IN INT
  4  ) RETURN INT IS
  5      v_count INT;
  6  BEGIN
  7      SELECT COUNT(*) INTO v_count
  8      FROM Votes
  9      WHERE Candidate_ID = p_candidate_id
 10        AND Election_ID = p_election_id;
 11
 12      RETURN v_count;
 13  END;
 14  /

Function created.
```