PHASE 6

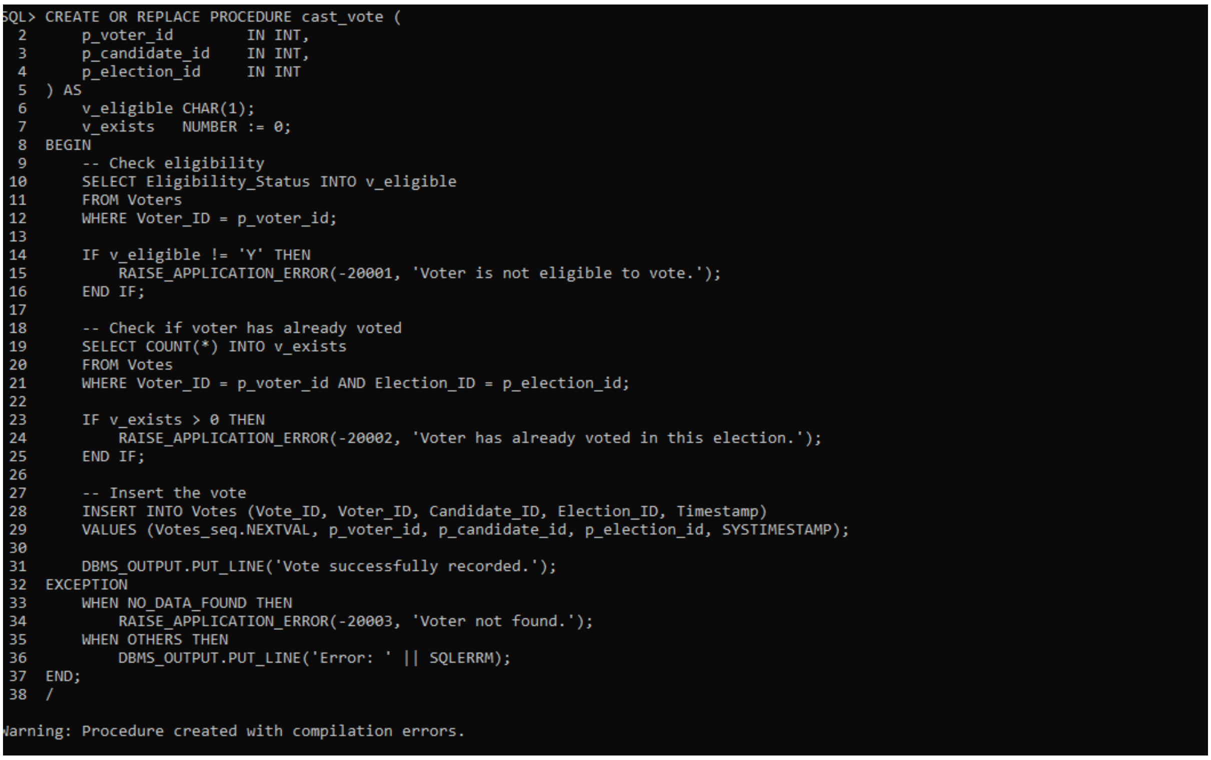
Use Case Example:

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

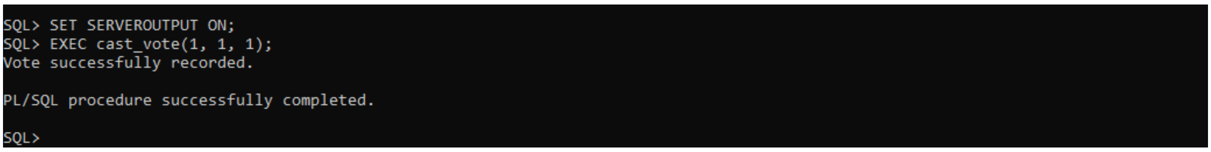
Procedure to Cast a Vote

```
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;
/
```



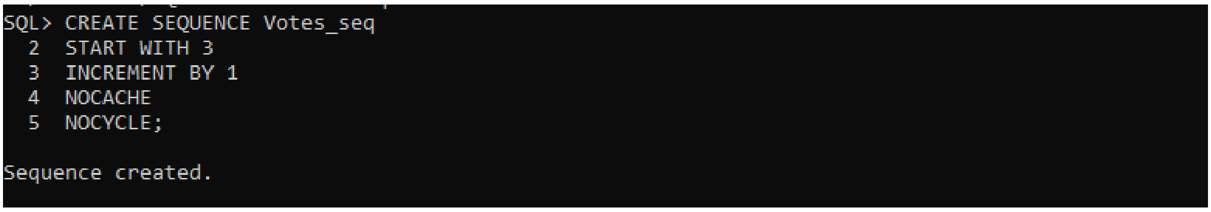
```
SQL> CREATE OR REPLACE PROCEDURE cast_vote (
         p_voter_id        IN INT,
         p_candidate_id    IN INT,
         p_election_id     IN INT
     ) AS
         v_eligible CHAR(1);
         v_exists   NUMBER := 0;
     BEGIN
         -- Check eligibility
         SELECT Eligibility_Status 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;
     /

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>
```

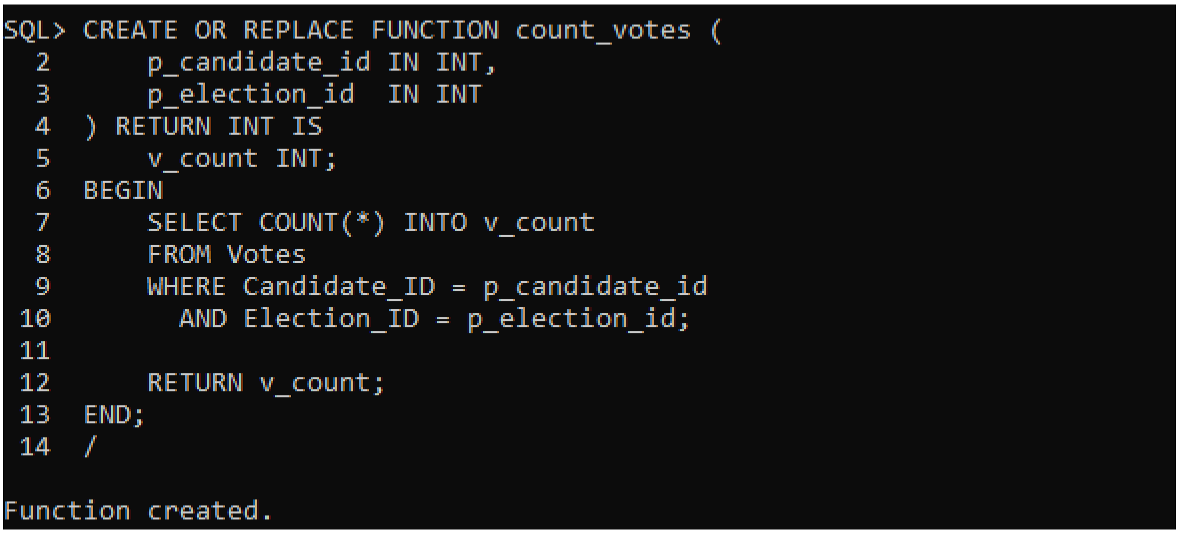
Supporting Sequence (for Vote_ID)

```
SQL> CREATE SEQUENCE Votes_seq

   START WITH 3

   INCREMENT BY 1

   NOCACHE

   NOCYCLE;
```



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

Sequence created.
```

Function Example – Count Votes for a Candidate



```
SQL> CREATE OR REPLACE FUNCTION count_votes (
         p_candidate_id IN INT,
         p_election_id  IN INT
     ) RETURN INT IS
         v_count INT;
     BEGIN
         SELECT COUNT(*) INTO v_count
         FROM Votes
         WHERE Candidate_ID = p_candidate_id
           AND Election_ID = p_election_id;

         RETURN v_count;
     END;
     /

Function created.
```