

# **Assignment 2 Report**

**Group 2**

**Members: Noah Stanich, Rei Uyeno**

## Syntax Error Identification

### Syntax Error 1

Code Section in Error:

```
IF (r_gggs.data_type = k_customer) THEN
```

Error Message Received:

Reason for Error:

- The initial if statement in this block is not closed.

Correction:

- Add an END IF clause at the end of the block.

```
ELSE  
RAISE_APPLICATION_ERROR(-20000, r_gggs.data_type || ' is not a valid type of data to process');  
END IF;
```

### Syntax Error 2

Code Section in Error:

```
ELSE IF (r_gggs.process_type = k_change) THEN
```

Error Message Received:

Reason for Error:

- This is invalid syntax for an else-if clause in PL/SQL.

Correction:

- The correct syntax for an else-if clause in PL/SQL is ELSIF (condition)  
THEN...

```
ELSIF (r_gggs.process_type = k_change) THEN
```

### Syntax Error 3

Code Section in Error

```
contact_phone_number = NVL2(r_gggs.column6, r_gggs.column6, contact_phone_number)
```

Error Message Received:

```
ERROR at line 70:  
ORA-06550: line 70, column 41:  
PL/SQL: ORA-00909: invalid number of arguments  
ORA-06550: line 66, column 11:  
PL/SQL: SQL Statement ignored
```

Reason for Error:

- Function has too many parameters

Correction

- Take out the duplicate item

```
phone_number = NVL(r_gggs.column6, phone_number)
```

### Syntax Error 4

Code Section in Error

```
WHERE name = r_gggs.column1
```

Reason for Error:

- Lack of semicolon

Error message:

```
ERROR at line 41:
ORA-06550: line 41, column 9:
PL/SQL: ORA-00933: SQL command not properly ended
ORA-06550: line 37, column 11:
PL/SQL: SQL Statement ignored
```

Correction

- Give it a semicolon

```
phone_number = NVL(r_gggs.phone_number, 0)
WHERE name = r_gggs.column1;
--
```

### Syntax Error 5

Code Section in Error

```
ELSIF (r_gggs.process_type = k_stats) THEN
    UPDATE gggs_vendor
    SET status = r_gggs.column2
    WHERE name = r_gggs.column1;
```

Error Message Received:

```
148 /
      ELSIF (r_gggs.process_type = k_stats) THEN
      *
ERROR at line 61:
ORA-06550: line 61, column 38:
PLS-00201: identifier 'K_STATS' must be declared
ORA-06550: line 55, column 9:
PL/SQL: Statement ignored
```

Reason for Error:

- Identifier is misspelled

Correction

- Identifier should be k\_status, not k\_stats

```
ELSIF (r_gggs.process_type = k_status) THEN
  UPDATE gggs_vendor
  SET status = r_gggs.column2
  WHERE name = r_gggs.column1;
```

### Logic Error Identification

### Syntax Error Identification

#### Logical Error 1

Expected Result:

- Code terminates after exception block

Actual Result:

- Code fails to terminate after exception block

Code Section in Error:

```
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;

    v_message := SQLERRM;

    INSERT INTO gggs_error_log_table
    VALUES
    (r_gggs.data_type, r_gggs.process_type, v_message);

    COMMIT;
```

Reason for Error:

- Lack of end block under exception

Correction:

- Add END statement at end of exception block

```
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;

    v_message := SQLERRM;

    INSERT INTO gggs_error_log_table
    VALUES
    (r_gggs.data_type, r_gggs.process_type, v_message);

    COMMIT;
END;
```

## Logical Error 2

Expected Result:

- 7 rows should be selected from gggs\_vendor.

Actual Result:

- 6 rows are selected from gggs\_vendor.

Code Section in Error:

```
ELSIF (r_gggs.data_type = k_vendor) THEN

  IF (r_gggs.process_type = k_status) THEN
    UPDATE gggs_vendor
    SET status = r_gggs.column2
    WHERE name = r_gggs.column1;

  ELSIF (r_gggs.process_type = k_change) THEN
    UPDATE gggs_vendor
    SET description = DECODE(r_gggs.column2, k_no_change_char, description, r_gggs.column2),
        contact_first_name = DECODE(r_gggs.column3, k_no_change_char, contact_first_name, r_gggs.column3),
        contact_last_name = DECODE(r_gggs.column4, k_no_change_char, contact_last_name, r_gggs.column4),
        contact_phone_number = NVL2(r_gggs.column6, r_gggs.column6, contact_phone_number)
    WHERE name = r_gggs.column1;
  ELSE
    RAISE_APPLICATION_ERROR(-20001, r_gggs.process_type || ' is not a valid process request for ' || r_gggs.data_type || ' data');
  END IF;
```

Reason for Error:

- Last item in ending data missing

Correction:

```
IF (r_gggs.process_type = k_new) THEN
  INSERT INTO gggs_vendor
  VALUES (gggs_vendor_seq.NEXTVAL, r_gggs.column1, r_gggs.column2, r_gggs.column3,
  r_gggs.column4, r_gggs.column6, k_active_status);
```

### Logical Error 3

Expected Result:

- Status is set to 'A' (active) where vendorid = 7

Actual Result:

- Status remains as default status ('S')

Code Section in Error:

```
IF (r_gggs.process_type = k_new) THEN
  INSERT INTO gggs_customer
  VALUES (gggs_customer_seq.NEXTVAL, r_gggs.column1, r_gggs.column2, r_gggs.column3,
  r_gggs.column4, r_gggs.column5, r_gggs.column6, k_active_status);
```

Reason for Error:

- The INSERT INTO ... VALUES clause inserts k\_status into gggs\_vendor instead of k\_active\_status.

Correction:

- Change the value inserted to k\_active\_status.

```
IF (r_gggs.process_type = k_new) THEN
  INSERT INTO gggs_customer
  VALUES (gggs_customer_seq.NEXTVAL, r_gggs.column1, r_gggs.column2, r_gggs.column3,
  r_gggs.column4, r_gggs.column5, r_gggs.column6, r_gggs.column7, k_active_status);
```

# Recommendations

The code lacks documentation, making it difficult to read. To improve readability, multi-line comments should be added before code blocks explaining the purpose of the block, how the code interacts with the GGGS database, and the parameters going into the block. Another way to improve the readability and efficiency of the code is to reduce the number of if statements and eliminate the amount of repetition in the code. Case statements could possibly be used to make the code cleaner. The code could also use better naming conventions. For instance, constants should be in all caps and prefixed with C\_ for clarity purposes.