Name : Avdhut Pailwan **Roll No. :** BTB22
PRN : 2122000380 **Subject :** Advanced Database Systems Lab

Experiment No. : 6

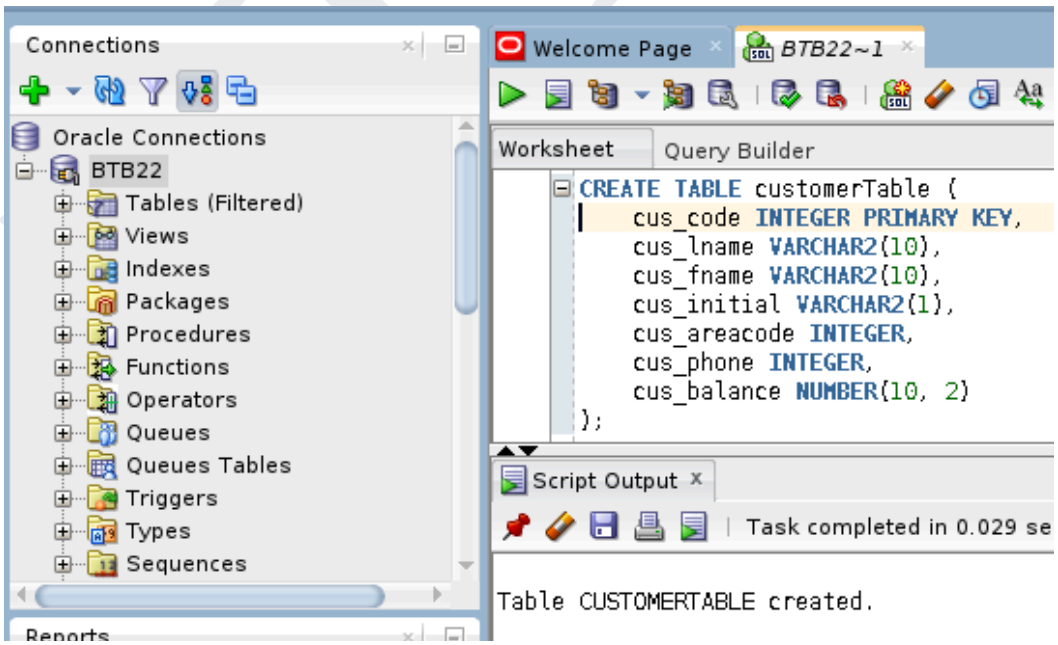
Advanced SQL

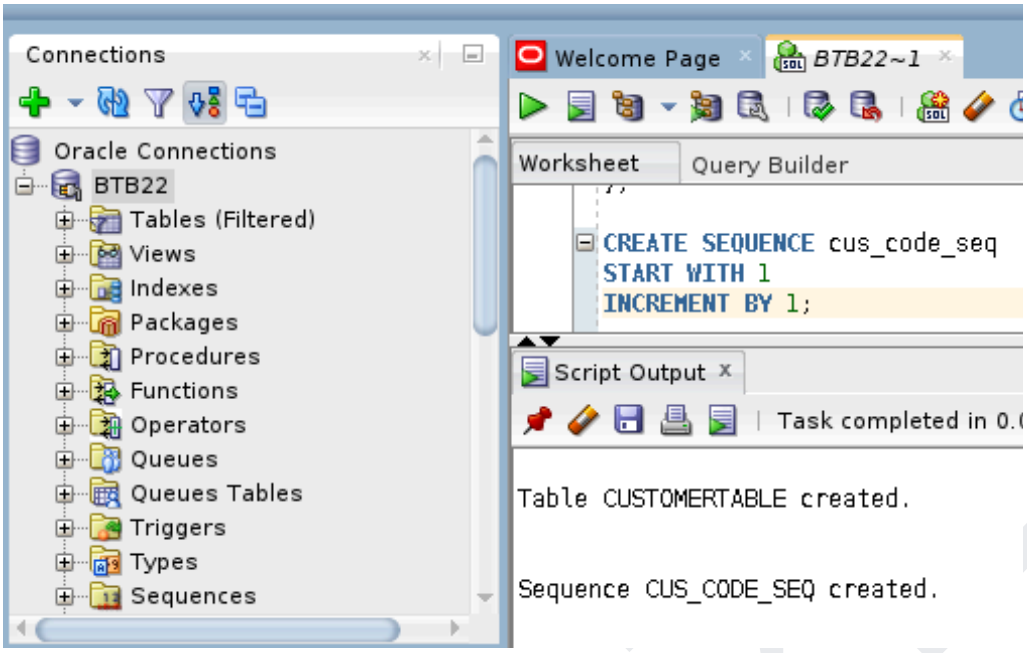
1. Oracle Sequences:

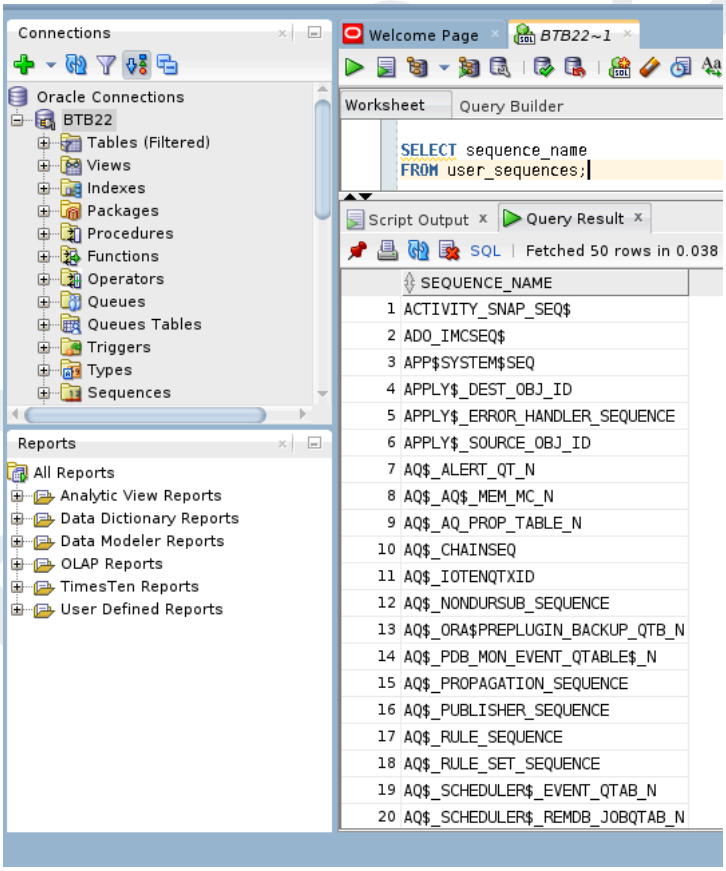
Consider table customer with primary key(cus_code)

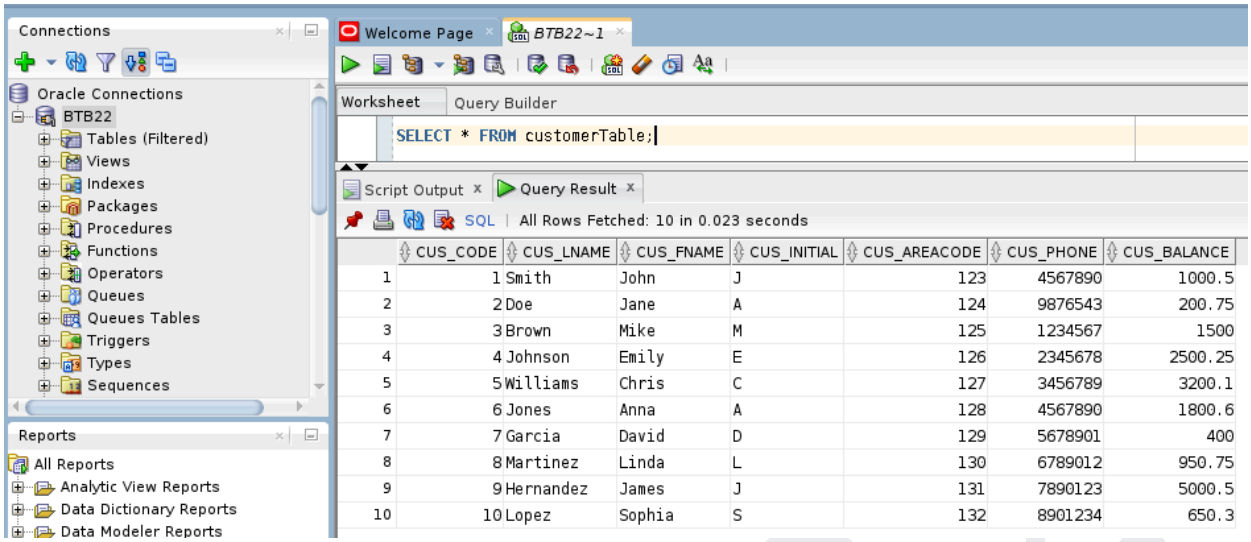
Field Type	Data Type
cus_code	Integer
cus_lname	varchar2(10)
cus_fname	varchar2(10)
cus_initial	varchar2(1)
cus_areacode	INTEGER
cus_phone	INTEGER
cus_balance)	number(10,2

1. Create sequence on cus_code
2. Display user sequences
3. Insert values into customer using created sequence
4. Display customer records




Connections
Oracle Connections
BTB22
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Welcome Page
BTB22~1
Worksheet
Query Builder
CREATE SEQUENCE cus_code_seq
START WITH 1
INCREMENT BY 1;
Script Output
Task completed in 0.
Table CUSTOMERTABLE created.
Sequence CUS_CODE_SEQ created.


Connections
Oracle Connections
BTB22
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
BTB22~1
Worksheet
Query Builder
SELECT sequence_name
FROM user_sequences;
Script Output
Query Result
SQL | Fetched 50 rows in 0.038
SEQUENCE_NAME
1 ACTIVITY_SNAP_SEQ$
2 ADO_IMCSEQ$
3 APP$SYSTEM$SEQ
4 APPLY$_DEST_OBJ_ID
5 APPLY$_ERROR_HANDLER_SEQUENCE
6 APPLY$_SOURCE_OBJ_ID
7 AQ$_ALERT_QT_N
8 AQ$_AQ$_MEM_MC_N
9 AQ$_AQ_PROP_TABLE_N
10 AQ$_CHAINSEQ
11 AQ$_IOTENQTXID
12 AQ$_NONDURSUB_SEQUENCE
13 AQ$_ORA$PREPLUGIN_BACKUP_QTB_N
14 AQ$_PDB_MON_EVENT_QTABLE$_N
15 AQ$_PROPAGATION_SEQUENCE
16 AQ$_PUBLISHER_SEQUENCE
17 AQ$_RULE_SEQUENCE
18 AQ$_RULE_SET_SEQUENCE
19 AQ$_SCHEDULER$_EVENT_QTAB_N
20 AQ$_SCHEDULER$_REMDB_JOBQTAB_N



Code :

```
CREATE TABLE customerTable (
    cus_code INTEGER PRIMARY KEY,
    cus_lname VARCHAR2(10),
    cus_fname VARCHAR2(10),
    cus_initial VARCHAR2(1),
    cus_areacode INTEGER,
    cus_phone INTEGER,
    cus_balance NUMBER(10, 2)
);

CREATE SEQUENCE cus_code_seq
START WITH 1
INCREMENT BY 1;

SELECT sequence_name
FROM user_sequences;

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Smith', 'John', 'J', 123, 4567890,
1000.50);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
```

```
VALUES (cus_code_seq.NEXTVAL, 'Doe', 'Jane', 'A', 124, 9876543,
200.75);


INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Brown', 'Mike', 'M', 125, 1234567,
1500.00);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Johnson', 'Emily', 'E', 126, 2345678,
2500.25);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Williams', 'Chris', 'C', 127, 3456789,
3200.10);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Jones', 'Anna', 'A', 128, 4567890,
1800.60);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Garcia', 'David', 'D', 129, 5678901,
400.00);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Martinez', 'Linda', 'L', 130, 6789012,
950.75);

INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Hernandez', 'James', 'J', 131,
7890123, 5000.50);
```

```
INSERT INTO customerTable (cus_code, cus_lname, cus_fname,
cus_initial, cus_areacode, cus_phone, cus_balance)
VALUES (cus_code_seq.NEXTVAL, 'Lopez', 'Sophia', 'S', 132, 8901234,
650.30);

SELECT * FROM customerTable;
```

2. **Trigger:**

Consider Student Report table, in which student marks assessment is recorded. In such schema, create a trigger so that the total and percentage of specified marks is automatically inserted whenever a record is inserting. Initial insert 0 for total and per attributes. Maximum marks should be 20 for each subject

Field | Type | Null | Key |

+------+-----+---------+----------------+

| tid | int(4) | NO | PRI |

| name | varchar(30) | YES |

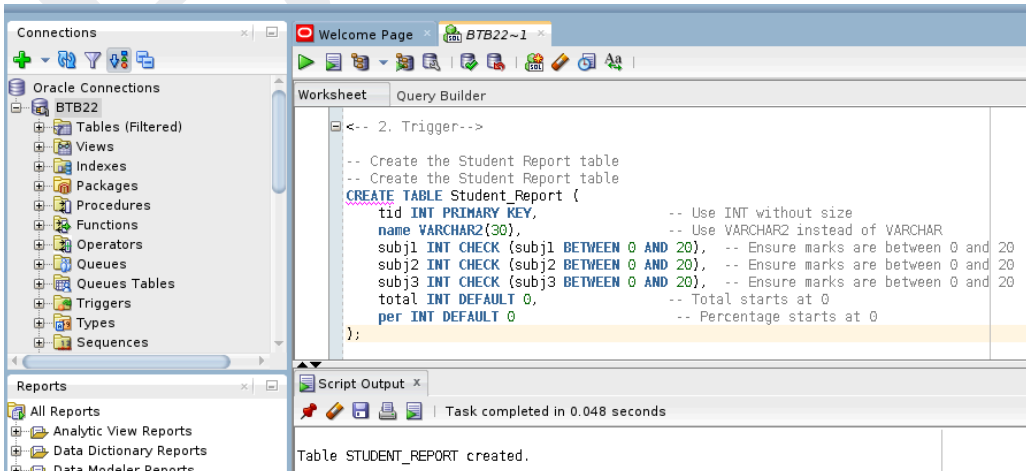
| subj1 | int(2) | YES |

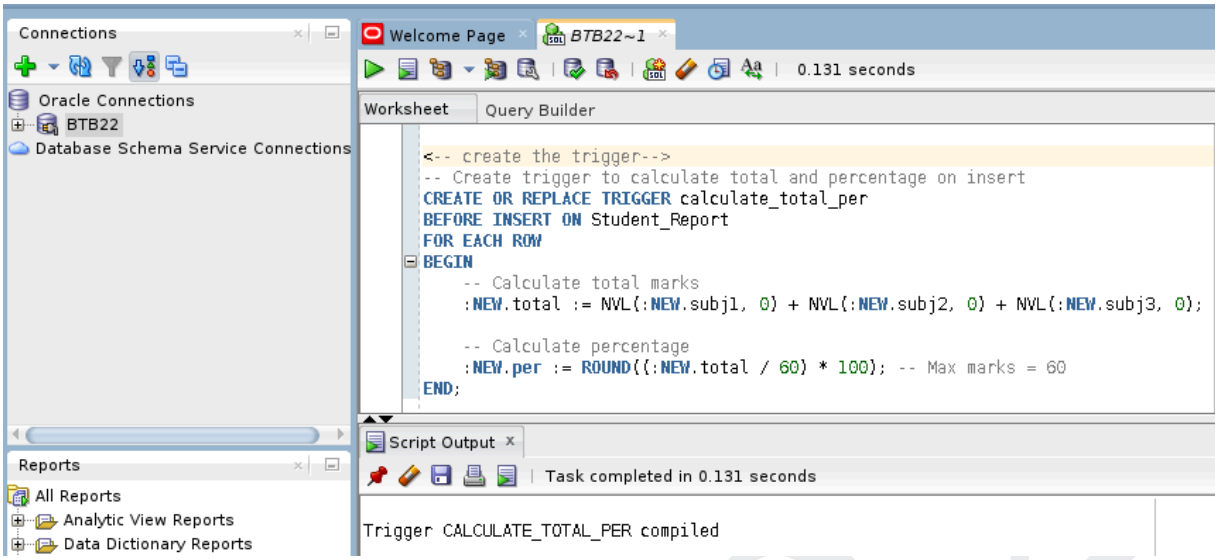
| subj2 | int(2) | YES |

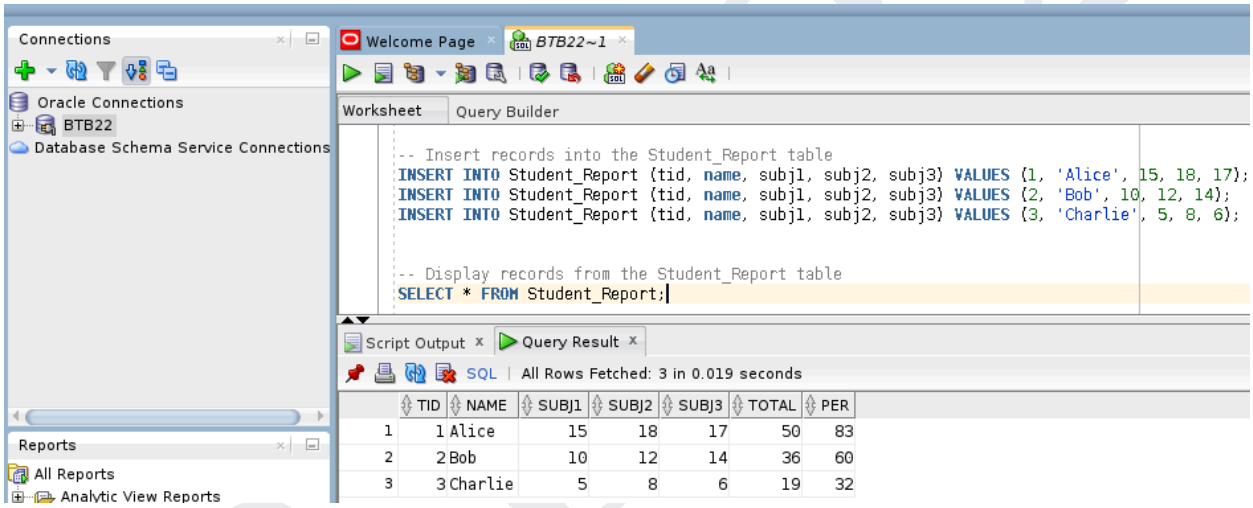
| subj3 | int(2) | YES |

| total | int(3) | YES |

| per | int(3) | YES |







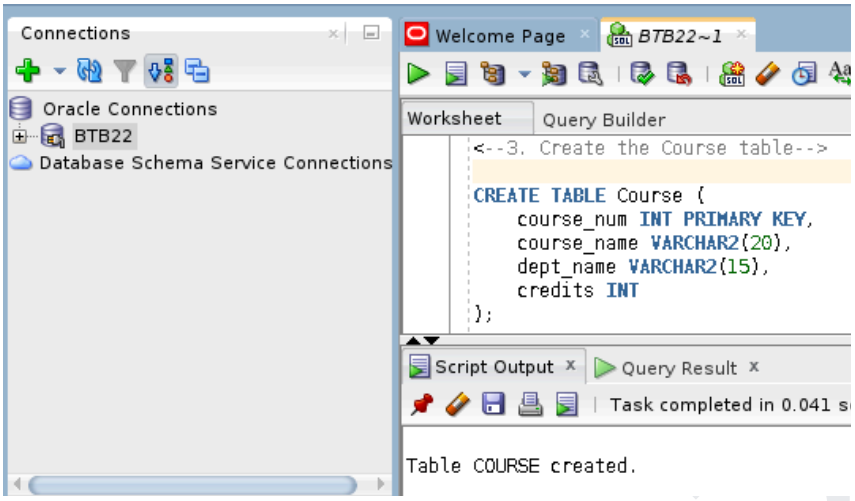
3. Procedure and Cursor:

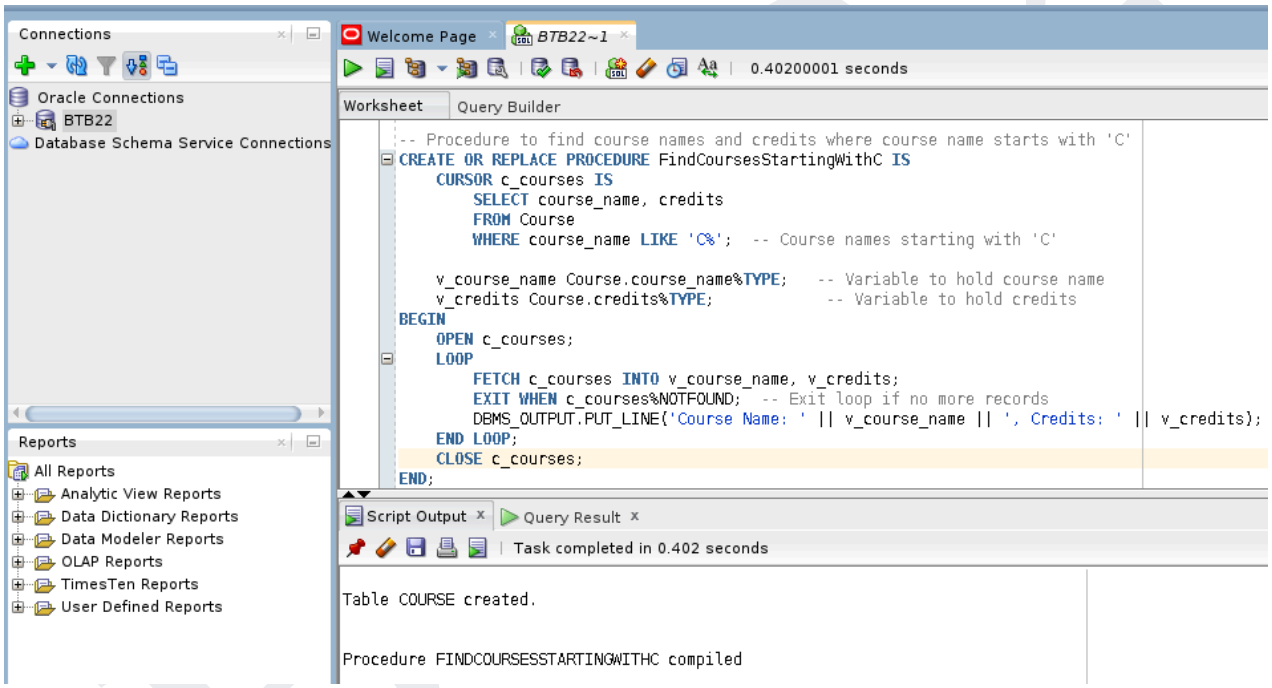
Consider Course Table with course_num as primary key.

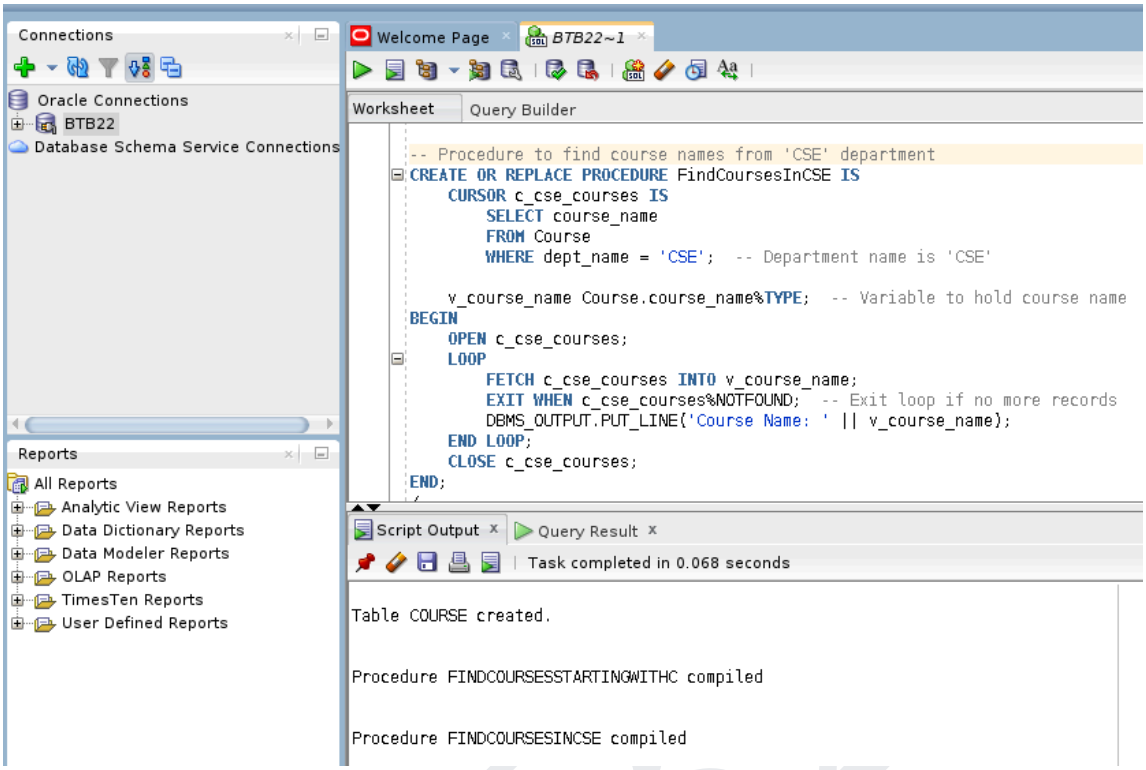
Field Type	Data Type
course_num	Integer
course_name	varchar2(20)
dept_name	varchar2(15)
credits	Integer

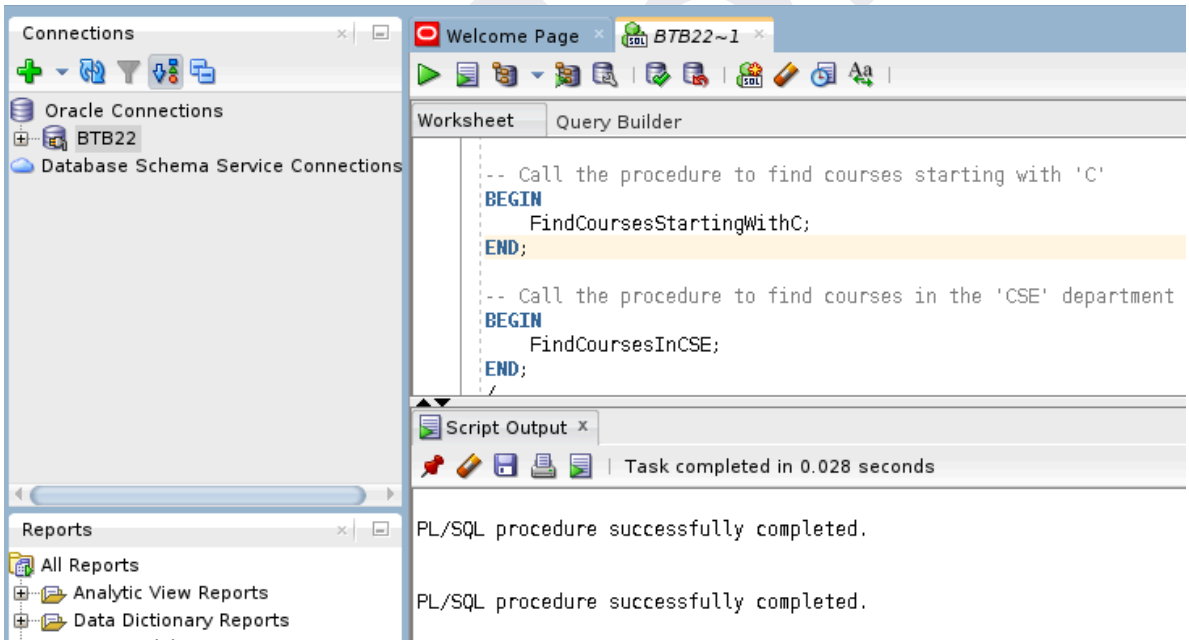
i) Write a procedure which includes cursors: Find course_name and credits where course name starts with ‘C’

ii) Write a procedure which includes cursors: Find course names from ‘CSE’ department


Connections
Oracle Connections
BTB22
Database Schema Service Connections
Welcome Page
BTB22~1
Worksheet
Query Builder
<--3. Create the Course table-->
CREATE TABLE Course (
course_num INT PRIMARY KEY,
course_name VARCHAR2(20),
dept_name VARCHAR2(15),
credits INT
);
Script Output
Query Result
Task completed in 0.041 s
Table COURSE created.


Connections
Oracle Connections
BTB22
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
BTB22~1
0.40200001 seconds
Worksheet
Query Builder
-- Procedure to find course names and credits where course name starts with 'C'
CREATE OR REPLACE PROCEDURE FindCoursesStartingWithC IS
CURSOR c_courses IS
SELECT course_name, credits
FROM Course
WHERE course_name LIKE 'C%'; -- Course names starting with 'C'
v_course_name Course.course_name%TYPE; -- Variable to hold course name
v_credits Course.credits%TYPE; -- Variable to hold credits
BEGIN
OPEN c_courses;
LOOP
FETCH c_courses INTO v_course_name, v_credits;
EXIT WHEN c_courses%NOTFOUND; -- Exit loop if no more records
DBMS_OUTPUT.PUT_LINE('Course Name: ' || v_course_name || ', Credits: ' || v_credits);
END LOOP;
CLOSE c_courses;
END;
Script Output
Query Result
Task completed in 0.402 seconds
Table COURSE created.
Procedure FINDCOURSESSTARTINGWITHC compiled


Connections
Oracle Connections
BTB22
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Welcome Page
BTB22~1
Worksheet
Query Builder
-- Procedure to find course names from 'CSE' department
CREATE OR REPLACE PROCEDURE FindCoursesInCSE IS
CURSOR c_cse_courses IS
SELECT course_name
FROM Course
WHERE dept_name = 'CSE'; -- Department name is 'CSE'
v_course_name Course.course_name%TYPE; -- Variable to hold course name
BEGIN
OPEN c_cse_courses;
LOOP
FETCH c_cse_courses INTO v_course_name;
EXIT WHEN c_cse_courses%NOTFOUND; -- Exit loop if no more records
DBMS_OUTPUT.PUT_LINE('Course Name: ' || v_course_name);
END LOOP;
CLOSE c_cse_courses;
END;
Script Output
Query Result
Task completed in 0.068 seconds
Table COURSE created.
Procedure FINDCOURSESSTARTINGWITHC compiled
Procedure FINDCOURSESINCSE compiled


Connections
Oracle Connections
BTB22
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Welcome Page
BTB22~1
Worksheet
Query Builder
-- Call the procedure to find courses starting with 'C'
BEGIN
FindCoursesStartingWithC;
END;
-- Call the procedure to find courses in the 'CSE' department
BEGIN
FindCoursesInCSE;
END;
Script Output
Task completed in 0.028 seconds
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.