David Willis

Lingma Acheson

CIT 304

3.23.18

Course Project 5

Query 1 – Create a table that will store a customized report showing how many tables, views, indexes, and users are in the database at a particular time (also retrieving a count of the objects and insert the count into a new table that will be created before this runs and the current time the rows are created)

set serveroutput on;

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MON-YYYY HH24:MI:SS';

CREATE TABLE TABLE\_SUMMARY(

TABLE\_SUM NUMBER,

VIEW\_SUM NUMBER,

INDEX\_SUM NUMBER,

USER\_SUM NUMBER,

DATE\_ROW\_CREATED DATE

);

DECLARE

TABLE\_SUM NUMBER;

NUM\_TABLES NUMBER;

VIEW\_SUM NUMBER;

NUM\_VIEWS NUMBER;

INDEX\_SUM NUMBER;

NUM\_INDEXES NUMBER;

USER\_SUM NUMBER;

NUM\_USERS NUMBER;

BEGIN

SELECT COUNT(\*) INTO TABLE\_SUM

FROM USER\_OBJECTS

WHERE OBJECT\_TYPE = 'TABLE';

NUM\_TABLES := TABLE\_SUM;

DBMS\_Output.Put\_Line('There are a total of ' || NUM\_TABLES

|| ' tables in the database.');

SELECT COUNT(\*) INTO VIEW\_SUM

FROM USER\_OBJECTS

WHERE OBJECT\_TYPE = 'VIEW';

NUM\_VIEWS := VIEW\_SUM;

DBMS\_Output.Put\_Line('There are a total of ' || NUM\_VIEWS

|| ' views in the database.');

SELECT COUNT(\*) INTO INDEX\_SUM

FROM USER\_OBJECTS

WHERE OBJECT\_TYPE = 'INDEX';

NUM\_INDEXES := INDEX\_SUM;

DBMS\_Output.Put\_Line('There are a total of ' || NUM\_INDEXES

|| ' indexes in the database.');

SELECT COUNT(\*) INTO USER\_SUM

FROM ALL\_USERS;

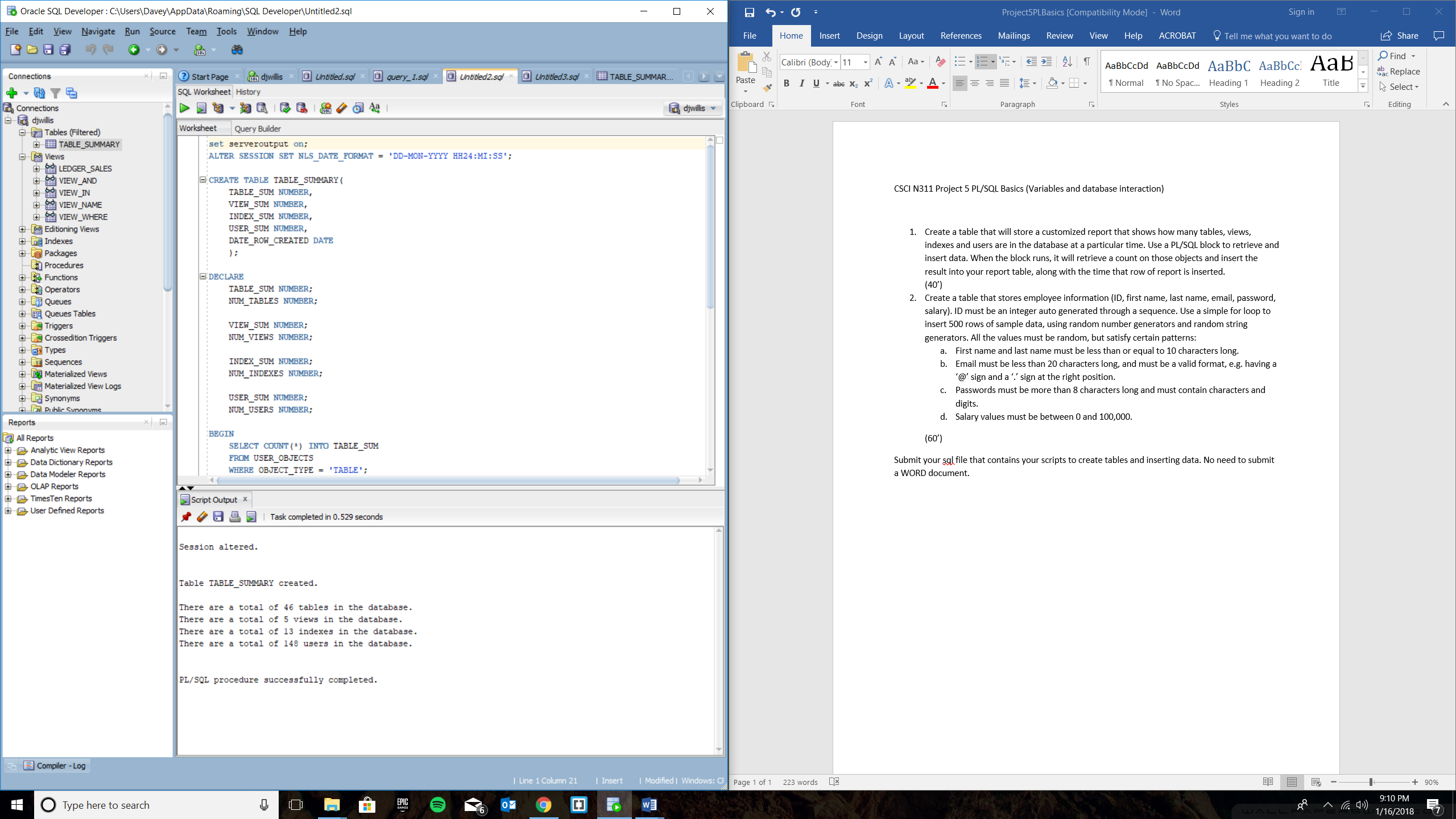
NUM\_USERS := USER\_SUM;

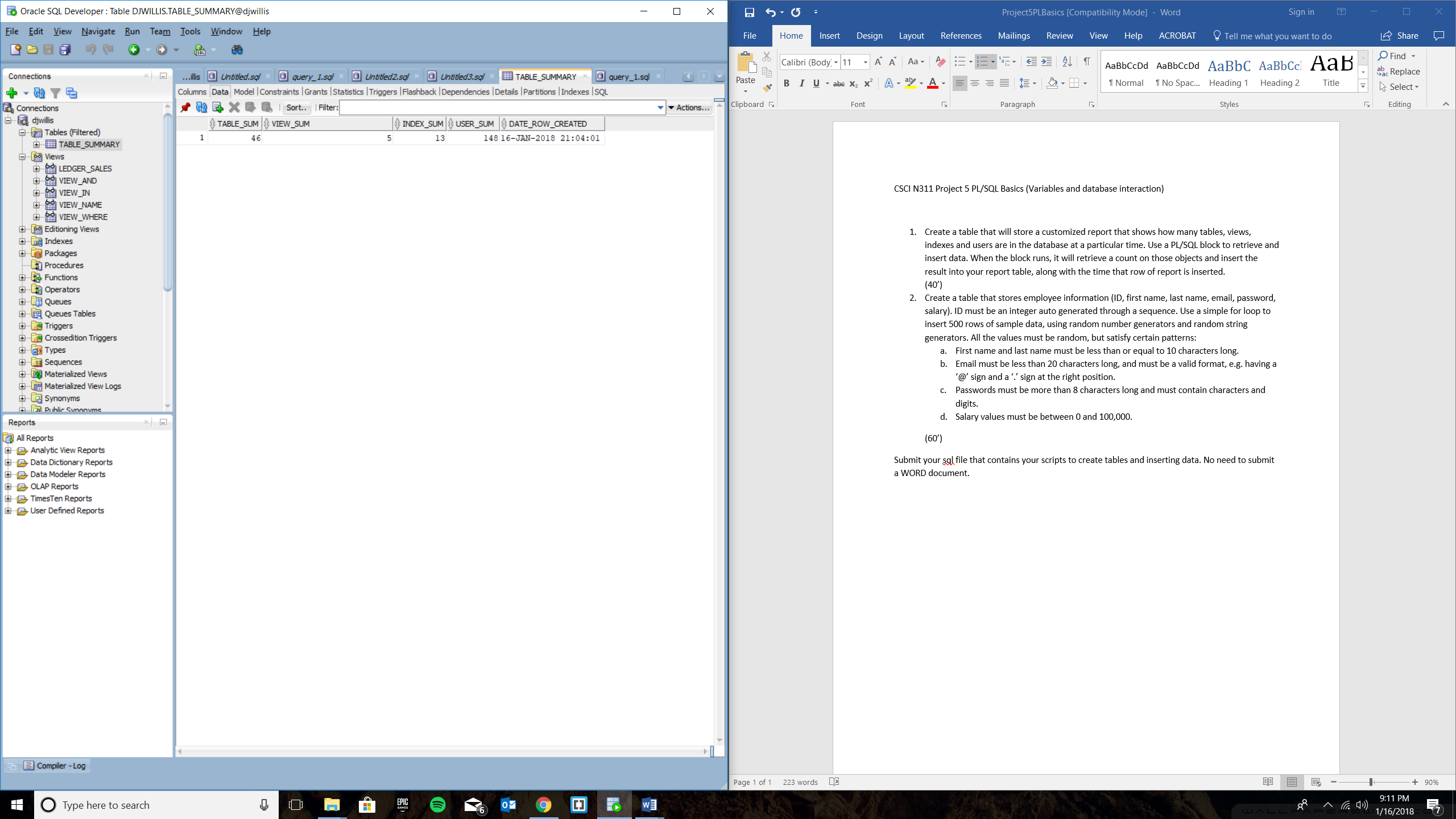
DBMS\_Output.Put\_Line('There are a total of ' || NUM\_USERS

|| ' users in the database.');

INSERT INTO TABLE\_SUMMARY VALUES(NUM\_TABLES, NUM\_VIEWS, NUM\_INDEXES, NUM\_USERS, CURRENT\_TIMESTAMP);

END;





Query 2 – Create a table that stores employee information (ID, first name, last name, email, password, and salary), but the id will be a auto generated via a created sequence (beginning with 1), and a for loop will be created to store random user data (looping 500 times to create 500 rows of random data for the table created before this runs); however, the first name must be 10 characters long, the email must be less than 20 characters long with a valid format (with an @ and .com), the passwords must be more than eight characters long and contain both characters and digits, and lastly, the salary must be between 0 and 100,000

CREATE TABLE EMPLOYEE\_INFO (

employee\_id int NOT NULL,

employee\_first\_name VARCHAR(10) NOT NULL,

employee\_last\_name VARCHAR(10) NOT NULL,

employee\_email VARCHAR(19),

employee\_password VARCHAR(255),

employee\_salary NUMBER,

CHECK (REGEXP\_LIKE (employee\_email, '^(\S+)\@(\S+)\.(\S+)$')),

CHECK (REGEXP\_LIKE (employee\_password, '[^0-9]+') AND

LENGTH(employee\_password) > 8),

CHECK (employee\_salary BETWEEN 0 AND 100000)

);

CREATE SEQUENCE INCREMENT\_BY\_ONE

MINVALUE 1

START WITH 1

INCREMENT BY 1

NOCACHE;

DECLARE

i int := 1;

RAND\_NAME VARCHAR(10);

RAND\_NAME\_INSERT VARCHAR(10);

RAND\_EMAIL VARCHAR(19);

RAND\_EMAIL\_INSERT VARCHAR(19);

RAND\_PASSWORD VARCHAR (255);

RAND\_PASSWORD\_INSERT VARCHAR(255);

RAND\_NUM NUMBER;

RAND\_NUM\_INSERT NUMBER;

BEGIN

FOR i IN 1 .. 500 LOOP

SELECT dbms\_random.string('U', 10) INTO RAND\_NAME from dual;

RAND\_NAME\_INSERT := RAND\_NAME;

SELECT ROUND(dbms\_random.value(1,100000)) INTO RAND\_NUM from dual;

RAND\_NUM\_INSERT := RAND\_NUM;

SELECT dbms\_random.string('X', 10) INTO RAND\_PASSWORD from dual;

RAND\_PASSWORD\_INSERT := RAND\_PASSWORD;

SELECT dbms\_random.string('l', 5) || '@' || dbms\_random.string('l', 5) || '.com' INTO RAND\_EMAIL from dual;

RAND\_EMAIL\_INSERT := RAND\_EMAIL;

INSERT INTO EMPLOYEE\_INFO VALUES(INCREMENT\_BY\_ONE.nextval, RAND\_NAME\_INSERT, RAND\_NAME\_INSERT, RAND\_EMAIL\_INSERT, RAND\_PASSWORD\_INSERT, RAND\_NUM\_INSERT);

END LOOP;

END;

