1. CREATE TABLE department (

department\_id NUMBER,

department\_name VARCHAR(30),

department\_block\_number NUMBER,

CONSTRAINT department\_id PRIMARY KEY (department\_id)   
);

1. CREATE TABLE student (

student\_id NUMBER,

student\_name VARCHAR(30),

address VARCHAR(40),

city VARCHAR(30),

department\_id NUMBER,

       CONSTRAINT student\_pk PRIMARY KEY (student\_id),

CONSTRAINT fk\_student FOREIGN KEY (department\_id)  REFERENCES department(department\_id)

);

1. CREATE TABLE staff (

staff\_id NUMBER,

staff\_name VARCHAR(30),

department\_id NUMBER,

CONSTRAINT staff\_pk PRIMARY KEY (staff\_id),

CONSTRAINT fk\_staff FOREIGN KEY (department\_id) REFERENCES department(department\_id)

);

1. CREATE TABLE subject (

subject\_id NUMBER,

student\_name VARCHAR(30),

student\_code VARCHAR(10),

staff\_id NUMBER,

CONSTRAINT subject\_pk PRIMARY KEY (subject\_id),

CONSTRAINT fk\_subject FOREIGN KEY (staff\_id) REFERENCES staff(staff\_id)   
);

1. CREATE TABLE mark (

  value\_mark NUMBER,

subject\_id NUMBER,

student\_id NUMBER,

CONSTRAINT fk\_markTwo FOREIGN KEY (subject\_id) REFERENCES subject(subject\_id),

CONSTRAINT fk\_markThree FOREIGN KEY (student\_id) REFERENCES student(student\_id)

);

1. Add a constraint by writing a query to add a NOT NULL constraint to column

email\_id VARCHAR(50) NOT NULL in the staff table column staff-name.

1. Add a constraint NOT NULL to column email\_id to student table

ALTER table add email\_id VARCHAR(50) NOT NULL.

1. Modify the size of the type of field studentid on the student table by writing a query to change it to VARCHAR 50.
2. Remove the email\_id column on the Student table by writing a query.

SECTION 2

Importing into Tables

1. Download the following Excel sheets from LMS: Department, Staff, Mark, Subject, Student.
2. Import those excel sheets into corresponding tables: Department, Staff, Mark, Subject, Student.

SECTION 3

Updating Records

1. Update a record by writing a query to update the subject\_name in the Subject Table from Sales to Computer Science and subject\_ code from 1842 to 1919.

SECTION 4

Deleting Records

1. Delete the row from the subject table where subject name is Accounting by writing the appropriate query.

SECTION 5

Basic Selection of Records

1. Display the names of the department in the college by writing the appropriate query.  Please note that these must be displayed in ascending order.
2. Display the name of the department where the departments block number is between 3 and 10

by writing the appropriate query.

SELECT Department\_Block\_Number

FROM Department

WHERE Department\_Block\_Number BETWEEN 3 AND 10

Reported “No data found” Reason: I can’t load Excel Tables using SQL Live.

1. Display the names of all the students in the college by writing the appropriate query.  Please note these must be in ascending order.

SELECT Student\_name

FROM Student TABLE

ORDER BY last\_name

SECTION 6

Selecting Single Rows

1. Display the names of students who are from Chicago, Taylor and San Jose. Please note these must be displayed in ascending order of their respective id.

SELECT Student\_name (

FROM Student TABLE

ORDER BY City\_name (Chicago, Taylor, San Jose)

);

1. Writing the correct query display the address and city of the students table give the alias as Address\_Student
2. Display all the student’s names whose names are of six characters by writing the correct query.

SELECT Student\_name

FROM Student TABLE

WHERE Student\_name (LEN>6)

SQL Live NOT SAVING NEW STATEMENTS.  After three attempts have postponed work on assignment.

20.

Section 8-

SQL Joins

22- Display the names of the department and the student count in each department by writing the correct query. The student count in each department must be in ascending order based on the department name and an alias of student\_count for the student count.

23- Display the Student\_Name from STUDENT and the Subject\_name from SUBJECT where the Subject\_code from SUBJECT is greater than 1600.

24- Display the Stundent\_Name from STUDENTS and the Subject\_name from SUBJECT where the value on MARK table is less 3.

SELECT    DEPARTMENT\_NAME

   AS    DEPARTMENT\_NAME,

COUNT(\*)

   AS    STUDENT\_COUNT

FROM      DEPARTMENT

INNER

JOIN      STUDENT

ON        STUDENT.DEPARTMENT\_ID = DEPARTMENT.DEPARTMENT\_ID

GROUP BY  DEPARTMENT.DEPARTMENT\_ID, DEPARTMENT\_NAME

ORDER BY  DEPARTMENT.DEPARTMENT\_NAME;

SELECT  STUDENT\_NAME, SUBJECT.SUBJECT\_NAME

FROM

       (

           SELECT    STUDENT\_NAME

           FROM      STUDENT

           GROUP BY  STUDENT\_NAME

       )

JOIN    SUBJECT

ON      SUBJECT.SUBJECT\_CODE > 1600;

SELECT    STUDENT\_NAME,SUBJECT.SUBJECT\_NAME

FROM      STUDENT

INNER

JOIN      MARK

ON        MARK.STUDENT\_ID = STUDENT.STUDENT\_ID

INNER

JOIN      SUBJECT

ON        MARK.SUBJECT\_ID = SUBJECT.SUBJECT\_ID

WHERE     MARK.VALUE = 3;

\*\*

Section 9 –

Selecting Sub-Queries

25- Display the block number in which the maximum number of departments is located by writing the correct sub-query.

26- Display the names of the staff who are not handling any subjects by ascending order using the correct sub-query.

SELECT  DEPARTMENT\_NAME,

       DEPARTMENT\_BLOCK\_NUMBER

FROM    DEPARTMENT

WHERE   DEPARTMENT\_NAME = (

                             SELECT MAX(DEPARTMENT\_NAME)

                             FROM DEPARTMENT

                         );

SELECT  STAFF\_NAME,

       STAFF\_ID

FROM    STAFF

WHERE   STAFF\_ID NOT IN

                       (

                         SELECT STAFF\_ID

                         FROM   SUBJECT

                       );

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Section 10-

Functions

27- Write a function that takes department\_id as the input and returns the department\_name.

Use the function name below:

Function name: find\_dept\_name

CREATE OR REPLACE FUNCTION

                 FIND\_DEPT\_NAME(X NUMBER)

  RETURN VARCHAR2

  IS DEP VARCHAR2(30);

  BEGIN

         SELECT  DEPARTMENT\_NAME

         INTO    DEP

         FROM    DEPARTMENT

         WHERE   DEPARTMENT\_ID =X;

         RETURN  (DEP);

   END;

28- Write a function that takes department id as the input and returns the block number.

Use the function name below:

Function name:  find\_dept\_block

CREATE OR REPLACE FUNCTION

                     FIND\_DEPT\_BLOCK(X NUMBER)

  RETURN     NUMBER

  IS DEPNUM  NUMBER;

  BEGIN

             SELECT  DEPARTMENT\_BLOCK\_NUMBER

             INTO    DEPNUM

             FROM    DEPARTMENT

             WHERE   DEPARTMENT\_ID =X;

             RETURN  (DEPNUM);

   END;

29- Write a function that takes the staff id as the input and returns the staff name.

Use the function name below:

Function name:  find\_staff\_name

CREATE OR REPLACE FUNCTION

                 FIND\_DEPT\_NAME(X NUMBER)

  RETURN VARCHAR2

  IS DEP VARCHAR2(30);

  BEGIN

         SELECT  DEPARTMENT\_NAME

         INTO    DEP

         FROM    DEPARTMENT

         WHERE   DEPARTMENT\_ID =X;

         RETURN  (DEP);

 END;