```
/* TOPIC 1:  Unit testing with NULL

-- a quick reminder when looking at NULL
-- NULL has no value, so what do you expect from these tests?
Select LENGTH(NULL);
Select CONCAT(NULL,'B');

Select NULL=1;
Select NULL + 1;
Select NULL + 'B';

Select IFNULL(NULL,0)=0;

*/




/* TOPIC 2:   [INNER] JOIN

*/
-- Question:  Which students are signed up for classes?
-- Show the students and include information about the signup date


SELECT *
FROM  Student  -- this is now the "left-most" (or left) table
JOIN ClassStudent --  this is the right table
      ON Student.ID = ClassStudent.StudentID
ORDER BY Student.FirstName;


-- INNER  is an optional keyword. Inner is the default
-- This query is equivalent to the previous query
SELECT *
FROM  Student
INNER JOIN ClassStudent
      ON Student.ID = ClassStudent.StudentID
ORDER BY Student.FirstName;



/* TOPIC 3:  LEFT JOIN
*/
-- a LEFT JOIN statement:

--Request:  Show ALL students, and if the information is available,
--          also show details about when they signed up for classes.
SELECT *
FROM  Student
LEFT JOIN ClassStudent
      ON Student.ID = ClassStudent.StudentID
```

```
ORDER BY Student.FirstName;


/* TOPIC 4:   LEFT JOIN
              - check for only data without a match in the (right table)
*/

-- Question:  Which classes have no students signed up?
-- Start by finding ALL classes, and show information about possible signups
SELECT *
FROM  Class c
LEFT JOIN ClassStudent cs
      ON c.ID = cs.ClassID ;

-- Now modify the previous query to focus
-- only on those that don't have a match in ClassStudent

SELECT *
FROM  Class c
LEFT JOIN ClassStudent cs
      ON c.ID = cs.ClassID
WHERE cs.ID IS NULL;
-- This query finds all classes that do NOT have any students.




-- NOTE: this is a
-- LEFT join that includes a predicate (WHERE clause conditional)
-- that limits the results,
-- to only return data with NULL column values from the right table



/* TOPIC 5:  View many different IDs, to reinforce
      understanding of Foreign keys and Primary Key Join columns
*/
SELECT Student.ID as SID, ClassStudent.ID as CSID,
      ClassStudent.StudentID as StudentID, FirstName
FROM  Student
LEFT JOIN ClassStudent
      ON Student.ID = ClassStudent.StudentID
ORDER BY Student.FirstName;



-- !!!!
-- PERSONAL CHALLENGE (or think pair share if there is time)
-- REQUEST:  Find the Students that are not signed up for a class
```

```
/* TOPIC 6:  3 Table JOIN
*/

--
-- NOTE:
-- Each time you choose another table, you will want to
-- find the appropriate table and column for its
-- foreign key to match up correctly to the
-- relevant primary key values and return the right records

SELECT *
FROM  Student
LEFT JOIN ClassStudent
      ON Student.ID = ClassStudent.StudentID
LEFT JOIN Class
      ON Class.ID = ClassStudent.ClassID
ORDER BY Student.FirstName;




-- Personal Challenge (or think-pair-share if there is time)
-- what happens if we reverse the tables in the join?
-- What Question might this query answer?

SELECT *
FROM  Class
LEFT JOIN ClassStudent
      ON Class.ID = ClassStudent.ClassID
LEFT JOIN Student
      ON Student.ID = ClassStudent.StudentID
ORDER BY Student.FirstName;




-- do you see how I have to reverse the ON statements too? Why?
-- What happens if there are no student first names to order by?
-- What does * in the select do now?



-- Now I'm going to use table aliases
-- so I'm not required to keep typing all the table names
SELECT *
FROM  Class c
LEFT JOIN ClassStudent cs
      ON c.ID = cs.ClassID
LEFT JOIN Student s
      ON s.ID = cs.StudentID
ORDER BY s.FirstName;
```

```
-- now that I’ve aliased the tables, I must use the alias and not the table
name
```