-- QUERY 1

CREATE DATABASE UNIVERSITY\_DATABASE1

-- QUERY 2

USE UNIVERSITY\_DATABASE1

-- QUERY 3

CREATE TABLE TBLCOURSES

(

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(30) NOT NULL,

COURSE\_DUR INT CHECK (COURSE\_DUR = 120 OR COURSE\_DUR = 180)

)

-- QUERY 4

INSERT INTO TBLCOURSES VALUES (101, 'COMPUTERS', 120), (102, 'ROBOTICS', 180)

INSERT INTO TBLCOURSES VALUES (103,'CIVIL', 180)

-- QUERY 5

SELECT \*FROM TBLCOURSES

-- QUERY 6:

CREATE TABLE TBLSTUDENTS

(

STD\_ID INT UNIQUE,

STD\_NAME VARCHAR(30),

STD\_AGE TINYINT CHECK (STD\_AGE >= 18),

STD\_COURSE\_ID INT REFERENCES tblcourses(course\_id)

)

-- QUERY 7:

INSERT INTO TBLSTUDENTS VALUES (1001, 'SAI', 34, 101), (1002, 'JON', 34, 101)

INSERT INTO TBLSTUDENTS VALUES (1003, 'JOHNE', 34, 102), (1004, 'JOHNY', 34, 102)

INSERT INTO TBLSTUDENTS VALUES (1005, 'AMIN', 35, 101), (1006, 'AMINI', 36, 102)

-- QUERY 8:

SELECT \* FROM TBLSTUDENTS

SELECT \* FROM TBLCOURSES

SELECT \* FROM TBLSTUDENTS

-- HOW TO COMPARE COURSES AND STUDENTS TABLES.

-- REQ 1: HOW TO FIND THE LIST OF ALL COURSES AND RESPECTIVE STUDENTS?

-- THIS MEANS, WE NEED TO COMPARE ABOVE TWO TABLES AND FIND THE MATCHING DATA

-- FOR DATA COMPARISONS AND REPORTING, WE CAN MAKE USE OF "JOINS"

-- IN THIS CASE, WE NEED TO COMPARE TWO TABLES AND FIND THE "MATCHING" DATA.

-- HENCE WE NEED TO USE "INNER JOIN"

SELECT \* FROM TBLCOURSES

INNER JOIN

TBLSTUDENTS

ON

TBLCOURSES.COURSE\_ID = TBLSTUDENTS.STD\_COURSE\_ID

-- REQ 2: HOW TO FIND THE LIST OF ALL COURSES WITH AND WITHOUT STUDENTS?

-- THIS MEANS, WE NEED TO COMPARE ABOVE TWO TABLES AND FIND THE MATCHING & MISSING DATA

-- HENCE WE NEED TO USE "OUTER JOIN". EITHER LEFT OUTER JOIN OR RIGHT OUTER JOIN OR FULL OUTER JOIN.

SELECT \* FROM TBLCOURSES -- THIS IS CALLED "LEFT" TABLE

LEFT OUTER JOIN -- ALL LEFT TABLE DATA + MATCHING RIGHT TABLE DATA. NON MATCH RIGHT : NULL

TBLSTUDENTS -- THIS IS CALLED "RIGHT" TABLE

ON

TBLCOURSES.COURSE\_ID = TBLSTUDENTS.STD\_COURSE\_ID

-- REQ 3: HOW TO FIND THE LIST OF ALL COURSES WITH AND WITHOUT STUDENTS?

-- THIS MEANS, WE NEED TO COMPARE ABOVE TWO TABLES AND FIND THE MATCHING & MISSING DATA

-- HENCE WE NEED TO USE "OUTER JOIN". EITHER LEFT OUTER JOIN OR RIGHT OUTER JOIN OR FULL OUTER JOIN.

SELECT \* FROM TBLSTUDENTS -- THIS IS CALLED "LEFT" TABLE

RIGHT OUTER JOIN -- ALL RIGHT TABLE DATA + MATCHING LEFT TABLE DATA. NON MATCH LEFT : NULL

TBLCOURSES -- THIS IS CALLED "RIGHT" TABLE

ON

TBLCOURSES.COURSE\_ID = TBLSTUDENTS.STD\_COURSE\_ID

-- REQ 4: HOW TO FIND THE LIST OF ALL COURSES WITHOUT STUDENTS?

-- THIS MEANS, WE NEED TO COMPARE ABOVE TWO TABLES AND FIND THE MISSING DATA

-- HENCE WE NEED TO USE "OUTER JOIN". EITHER LEFT OUTER JOIN OR RIGHT OUTER JOIN OR FULL OUTER JOIN.

SELECT \* FROM TBLCOURSES -- THIS IS CALLED "LEFT" TABLE

LEFT OUTER JOIN -- ALL RIGHT TABLE DATA + MATCHING LEFT TABLE DATA. NON MATCH LEFT : NULL

TBLSTUDENTS -- THIS IS CALLED "RIGHT" TABLE

ON

TBLCOURSES.COURSE\_ID = TBLSTUDENTS.STD\_COURSE\_ID

WHERE

TBLSTUDENTS.STD\_COURSE\_ID IS NULL

-- REQ 5: HOW TO FIND THE LIST OF ALL COURSES WITH STUDENTS?

-- THIS MEANS, WE NEED TO COMPARE ABOVE TWO TABLES AND FIND THE MISSING DATA

-- HENCE WE NEED TO USE "OUTER JOIN". EITHER LEFT OUTER JOIN OR RIGHT OUTER JOIN OR FULL OUTER JOIN.

SELECT \* FROM TBLCOURSES -- THIS IS CALLED "LEFT" TABLE

LEFT OUTER JOIN -- ALL RIGHT TABLE DATA + MATCHING LEFT TABLE DATA. NON MATCH LEFT : NULL

TBLSTUDENTS -- THIS IS CALLED "RIGHT" TABLE

ON

TBLCOURSES.COURSE\_ID = TBLSTUDENTS.STD\_COURSE\_ID

WHERE

TBLSTUDENTS.STD\_COURSE\_ID IS NOT NULL

-- SCENARIO FOR FULL OUTER JOINS

CREATE TABLE FLIGHT

(

FLIGHT\_ID INT,

FLIGHT\_CODE VARCHAR(30),

SOURCE\_CITY VARCHAR(30),

DESTINATION\_CITY VARCHAR(30)

)

insert FLIGHT values (10001, 1, 'city1', 'city2'),(10002, 2, 'city2', 'city3'),(10003, 3, 'city3', 'city1')

CREATE TABLE RESERVATIONS

(

RESV\_ID INT,

FLIGHT\_ID INT,

SEATS INT,

CLASS VARCHAR(30)

)

insert RESERVATIONS values (1001, 10001, 9, 'eco'),(1002, 10002, 8, 'eco')

insert RESERVATIONS values (1003, 10004, 9, 'eco')

SELECT \* FROM FLIGHT

SELECT \* FROM RESERVATIONS

-- REQ 6: HOW TO REPORT LIST OF ALL FLIGHTS WITH AND WITHOUT RESERVATIONS?

SELECT \* FROM FLIGHT

LEFT OUTER JOIN

RESERVATIONS

ON

FLIGHT.FLIGHT\_ID = RESERVATIONS.FLIGHT\_ID

-- REQ 7: HOW TO REPORT LIST OF ALL RESERVATIONS WITH AND WITHOUT FLIGHTS?

SELECT \* FROM FLIGHT

RIGHT OUTER JOIN

RESERVATIONS

ON

FLIGHT.FLIGHT\_ID = RESERVATIONS.FLIGHT\_ID

-- REQ 8: HOW TO REPORT LIST OF ALL FLIGHTS WITH AND WITHOUT RESERVATIONS AND ALL RESERVATIONS WITH AND WITHOUT FLIGHTS?

SELECT \* FROM FLIGHT

FULL OUTER JOIN

RESERVATIONS

ON

FLIGHT.FLIGHT\_ID = RESERVATIONS.FLIGHT\_ID

-- REQ 9: HOW TO REPORT ALL POSSIBLE COMBINATIONS OF FLIGHTS & RESERVATIONS?

SELECT \* FROM FLIGHT

CROSS JOIN

RESERVATIONS

SELECT \* FROM FLIGHT

CROSS APPLY

RESERVATIONS

SELECT \* FROM FLIGHT

,

RESERVATIONS

-- REQ 6: HOW TO FIND THE LIST OF ALL COURSES WITH & WITHOUT STUDENTS. ALL STUDENTS WITH & WITHOUT COURSES?

SELECT \* FROM

TBLSTUDENTS -- LEFT TABLE

FULL OUTER JOIN

tblcourses -- RIGHT TABLE

ON

TBLSTUDENTS.STD\_COURSE\_ID = tblcourses.course\_id

-- EXAMPLE FOR CROSS JOIN:

SELECT \* FROM

TBLSTUDENTS -- LEFT TABLE

CROSS JOIN

tblcourses -- RIGHT TABLE

-- EXAMPLES FOR JOIN OPTIONS:

-- When working with bigger tables (row count above 1000k rows) then we need to use MERGE JOIN option

SELECT \* FROM TBLCOURSES

LEFT OUTER MERGE JOIN -- AUTO SORTED AND FASTER FOR BIGGER TABLES

tblStudents

ON tblStudents.Std\_Course\_ID = TBLCOURSES.COURSE\_ID

WHERE tblStudents.Std\_Course\_ID is not null

SELECT COUNT(\*) FROM TBLCOURSES -- REPORTS TOTAL NUMBER OF ROWS

SELECT COUNT(DISTINCT COURSE\_ID) FROM TBLCOURSES -- REPORTS NUMBER OF ROWS WITH UNIQUE COURSE ID

SELECT \* FROM TBLCOURSES

RIGHT OUTER LOOP JOIN -- FOR SMALLER TABLES

tblStudents

ON tblStudents.Std\_Course\_ID = TBLCOURSES.COURSE\_ID

WHERE tblStudents.Std\_Course\_ID is not null

SELECT \* FROM TBLCOURSES

INNER HASH JOIN -- FOR DYNAMIC, HEAP [TABLES WITHOUT PK &/ INDEX] TABLES

tblStudents

ON tblStudents.Std\_Course\_ID = TBLCOURSES.COURSE\_ID

WHERE tblStudents.Std\_Course\_ID is not null

-- REQ 5 : HOW TO REPORT LIST OF ALL COURSES AND STUDENTS?

SELECT \* FROM TBLCOURSES CROSS JOIN tblStudents

SELECT \* FROM TBLCOURSES CROSS APPLY tblStudents

-- FOR CROSS JOIN, WE CANNOT USE "MERGE" OR "HASH" OR "LOOP".

CREATE TABLE RESERVATIONS

(

CRAFT\_CODE VARCHAR(30),

NO\_OF\_SEATS INT,

CLASS VARCHAR(30)

)

CREATE TABLE FLIGHT

(

CRAFT\_CODE VARCHAR(30),

SOURCE VARCHAR(30),

DESTINATION VARCHAR(30)

)

INSERT INTO RESERVATIONS VALUES ('AI01', 11, 'ECO'), ('AI02', 22, 'ECO'), ('AI03', 33, 'ECO'), ('AI04', 44, 'ECO')

INSERT INTO FLIGHT VALUES ('AI01', 'CITY1', 'CITY2'), ('AI02', 'CITY1', 'CITY3'), ('AI03', 'CITY3', 'CITY1')

-- HOW TO REPORT LIST OF ALL FLIGHTS AND MATCHING RESERVATIONS?

SELECT \* FROM

FLIGHT -- LEFT TABLE

INNER JOIN

RESERVATIONS -- RIGHT TABLE

ON FLIGHT.CRAFT\_CODE = RESERVATIONS.CRAFT\_CODE

-- HOW TO REPORT LIST OF ALL FLIGHTS WITHOUT ANY RESERVATIONS?

SELECT \* FROM

FLIGHT -- LEFT TABLE

LEFT OUTER JOIN

RESERVATIONS -- RIGHT TABLE

ON FLIGHT.CRAFT\_CODE = RESERVATIONS.CRAFT\_CODE

WHERE

RESERVATIONS.CRAFT\_CODE IS NULL

-- HOW TO REPORT LIST OF ALL RESERVATIONS WITHOUT ANY FLIGHTS?

SELECT \* FROM

RESERVATIONS -- LEFT TABLE

LEFT OUTER JOIN

FLIGHT -- RIGHT TABLE

ON FLIGHT.CRAFT\_CODE = RESERVATIONS.CRAFT\_CODE

WHERE

FLIGHT.CRAFT\_CODE IS NULL

-- HOW TO REPORT LIST OF ALL FLIGHTS WITHOUT ANY RESERVATIONS AND ALL RESERVATIONS WITHOUT ANY FLIGHTS ?

SELECT \* FROM

RESERVATIONS -- LEFT TABLE

FULL OUTER JOIN

FLIGHT -- RIGHT TABLE

ON FLIGHT.CRAFT\_CODE = RESERVATIONS.CRAFT\_CODE

-- HOW TO REPORT ALL POSSIBLE COMBINATIONS OF FLIGHTS AND RESERVATIONS?

SELECT \* FROM RESERVATIONS CROSS JOIN FLIGHT

-- FOR LEFT OUTER JOIN : ALL RESERVATIONS AND MATCHING FLIGHTS. NON MATCH FLIGHT IS NULL

-- FOR RIGHT OUTER JOIN : ALL FLIGHTS AND MATCHING RESRVATIONS. NON MATCH RESERVATIONS IS NULL

-- THIS IS YOUR TAKS ITEM : INSERT SOME SAMPLE DATA INTO ABOVE TABLES. TEST INNER, ALL OUTER JOIN QUERIES.

-- REQ 6 : HOW TO TUNE QUERIES WITH JOINS [FOR BIG TABLES] ?

SELECT \* FROM COURSES

INNER MERGE JOIN

tblStudents

ON COURSES.COURSE\_ID = tblStudents.StdCourse\_ID

-- REQ 7 : HOW TO TUNE QUERIES WITH JOINS [FOR SMALL TABLES] ?

SELECT \* FROM COURSES

LEFT OUTER LOOP JOIN

tblStudents

ON COURSES.COURSE\_ID = tblStudents.StdCourse\_ID

-- REQ 8 : HOW TO TUNE QUERIES WITH JOINS [FOR HEAP TABLES] ?

SELECT \* FROM COURSES

FULL OUTER LOOP JOIN

tblStudents

ON

COURSES.COURSE\_ID = tblStudents.StdCourse\_ID