**IST370 Practice Work Sheet**

**Theme: Outer Joins**

**First things first: Copy the following SQL script and paste it onto the "Run SQL Command Line" program:**

DROP TABLE rooms CASCADE CONSTRAINTS;

DROP TABLE students CASCADE CONSTRAINTS;

CREATE TABLE rooms

(room\_code NUMBER(4) PRIMARY KEY,

bldg\_code VARCHAR2(10));

INSERT INTO rooms

VALUES (1001, 'ALPHA');

INSERT INTO rooms

VALUES (1002, 'ALPHA');

INSERT INTO rooms

VALUES (1003, 'DELTA');

INSERT INTO rooms

VALUES (1004, 'DELTA');

CREATE TABLE students

(stu\_num VARCHAR2(6) PRIMARY KEY,

lastname VARCHAR2(10),

firstname VARCHAR2(10),

ini CHAR(1),

room\_code NUMBER(4),

CONSTRAINT students\_roomscode\_fk FOREIGN KEY (room\_code)

REFERENCES rooms(room\_code));

INSERT INTO students

VALUES ('SIU001', 'SMITH', 'PETER', 'C', 1001);

INSERT INTO students

VALUES ('SIU002', 'JAMES', 'BOB', 'S', 1001);

INSERT INTO students

VALUES ('SIU003', 'DAVIS', 'JOHN', 'C', NULL);

INSERT INTO students

VALUES ('SIU004', 'LEE', 'ASHLEY', 'A', 1003);

INSERT INTO students

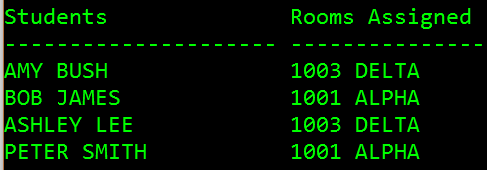
VALUES ('SIU005', 'BUSH', 'AMY', NULL, 1003);

INSERT INTO students

VALUES ('SIU006', 'OMABA', 'MIKE', 'D', NULL);

**PROBLEMS**

**-- CASE 1. List only those students who live in a dorm and only those rooms that are currently occupied (INNER JOIN)**



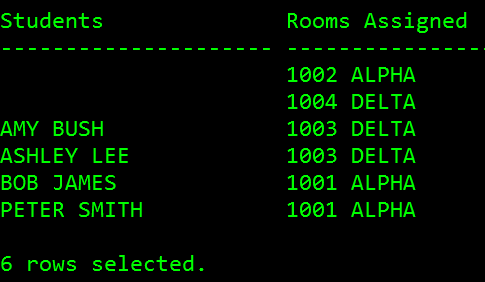
SELECT s.firstname || ' ' || s.lastname "Students", r.room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM students s, rooms r

WHERE s.room\_code = r.room\_code

ORDER BY 1;

**-- CASE 2. List all rooms (occupied + vacant)**



**-- Option 1: Implicit Joins**

SELECT s.firstname || ' ' || s.lastname "Students", r.room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM students s, rooms r

WHERE r.room\_code = s.room\_code (+)

ORDER BY 1;

**-- Option 2: Explicit Joins – LEFT OUTER JOIN … USING**

SELECT s.firstname || ' ' || s.lastname "Students", **room\_code** || ' ' || r.bldg\_code "Rooms Assigned"

FROM rooms r LEFT OUTER JOIN students s USING(room\_code)

ORDER BY 1;

**QUESTION: Can room\_code in the first line be qualified?**

**-- Option 3: Explicit Joins – RIGHT OUTER JOIN … USING**

SELECT s.firstname || ' ' || s.lastname "Students", room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM students s RIGHT OUTER JOIN rooms r USING(room\_code)

ORDER BY 1;

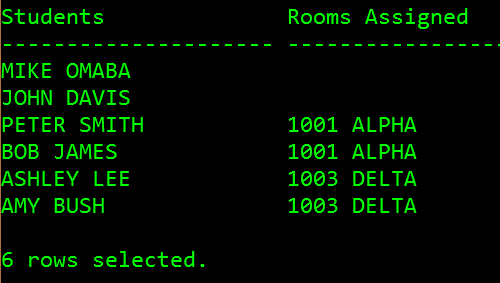
**-- Option 4: Explicit Joins – LEFT OUTER JOIN … ON**

SELECT s.firstname || ' ' || s.lastname "Students", r.room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM rooms r LEFT OUTER JOIN students s ON r.room\_code = s.room\_code (+)

ORDER BY 1;

**--- CASE 3. List all students (live on-campus + off-campus)**



**-- Option 1: Implicit Joins**

SELECT s.firstname || ' ' || s.lastname "Students", r.room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM students s, rooms r

WHERE s.room\_code =r.room\_code (+)

ORDER BY 1;

**-- Option 2: Explicit Joins – RIGHT OUTER JOIN … USING**

SELECT s.firstname || ' ' || s.lastname "Students", **room\_code** || ' ' || r.bldg\_code "Rooms Assigned"

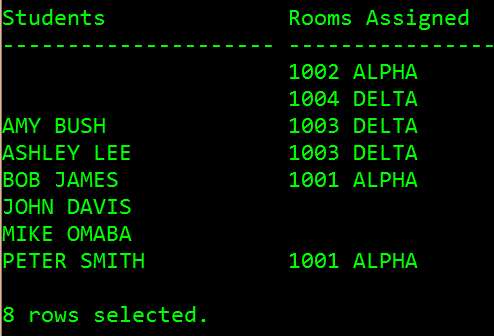
FROM rooms r RIGHT OUTER JOIN students s USING(room\_code)

ORDER BY 2;

NOTE: You can also use 'RIGHT OUTER JOIN … ON', 'LEFT OUTER JOIN …USING', 'LEFT OUTER JOIN …ON' to deliver the same result.

**--- CASE 4. List all students (living on-campus & off-campus) and all rooms (occupied & vacant)**

**QUESTION: Can we use the implicit join method to solve this problem? \_\_\_\_\_**



**-- Option 1: Explicit Joins – FULL OUTER JOIN … USING**

SELECT s.firstname || ' ' || s.lastname "Students", room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM rooms r FULL OUTER JOIN students s USING(room\_code)

ORDER BY 1;

**-- Option 2: Explicit Joins – FULL OUTER JOIN … ON**

SELECT s.firstname || ' ' || s.lastname "Students", r.room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM rooms r FULL OUTER JOIN students s On s.room\_code =r.room\_code

ORDER BY 1;

**-- Option 3: Explicit Joins – NATURAL FULL OUTER JOIN**

SELECT s.firstname || ' ' || s.lastname "Students", room\_code || ' ' || r.bldg\_code "Rooms Assigned"

FROM rooms r NATURAL FULL OUTER JOIN students s

ORDER BY 1;