Multiple Joins

Name:	Class:	Date:

Key Theory

Multiple joins happen when we combine three or more tables.

We can combine INNER, LEFT, RIGHT, CROSS types of join to create views of multiple tables.

After each JOIN operation a view is created that becomes the next LEFT table.

The table after the next join becomes the new right table.

This way we can combine data from many tables.

As a result it is important understand how table relationships work so queries can be planned to obtain accurate results.

A side effect of multiple joins is that you may get the same view from different combinations.

Example

Two RIGHT JOINS

```
SELECT *
FROM enrolments
RIGHT JOIN students ON enrolments.student_id = students.student_number      -- View created from enrolments and students
RIGHT JOIN subjects ON subjects.subject_id = enrolments.subject_id;          -- The view then becomes the LEFT table
```

enrolment_id	subject_id	student_id	student_number	student_age	passport	first_name	last_name	subject_id	trainer_id	subject_name	start_date
64	4	1	1	19	10540523	Harry	Biker	4	4	Software Development	2023-07-20
66	4	2	2	51	552440523	Aura	Lavender	4	4	Software Development	2023-07-20
68	4	1	1	19	10540523	Harry	Biker	4	4	Software Development	2023-07-20
65	6	4	4	19	10540523	Chucky	De Santos	6	3	Database	2023-06-20
67	6	1	1	19	10540523	Harry	Biker	6	3	Database	2023-06-20
69	6	1	1	19	10540523	Harry	Biker	6	3	Database	2023-06-20
70	7	2	2	51	552440523	Aura	Lavender	7	4	Software Development	2023-07-20

Multiple Joins

Task 1

Task 1.1

Create a query that combines three tables, enrolments, students, and subjects.

This query should return all students enrolled on a course and the subject they take.

```
SELECT *
FROM enrolments
RIGHT JOIN students
ON enrolments.student_id = students.student_number
RIGHT JOIN subjects
ON subjects.subject_id = enrolments.subject_id;
```

Task 1.2

Now change the second RIGHT JOIN to as LEFT JOIN.

```
LEFT JOIN subjects ON subjects.subject_id = enrolments.subject_id;
```

If you have students not enrolled they will have NULL values in both the enrolment and subject fields.

Task 1.3

Change both JOINS to INNER JOINS.

Can you explain what the result of this process is?

You may find it helpful to visualise how the data is combined.

Multiple Joins

Task 2

Task 2.1

Next want to find out what course each student is on and who their teacher is.

To do this we need three INNER JOINS.

The order of these joins needs to be well considered as we need to know where relationships exist between tables.

```
SELECT subject_name, CONCAT(first_name , " ", last_name) AS student_name,
trainer_name
FROM teachers INNER JOIN subjects ON teachers.trainer_id = subjects.trainer_id
INNER JOIN enrolments ON subjects.subject_id = enrolments.subject_id
INNER JOIN students ON enrolments.student_id = students.student_number;
```

Task 2.2

Change the last INNER JOIN to a RIGHT JOIN, this will only work if you have a student not enrolled on a course.

What is the result of this change?

Challenges

Challenge 1

Richard wants to know the names of all students taking his Software Development course.

Expand the query from Task 2.1 to achieve this.

Multiple Joins

Challenge 2

Richard wants to find out if students have been enrolled twice.

To do this you can alter the query from challenge 1:

COUNT (subject_name)

GROUP BY student_number

Challenge 3

Due to duplicate enrolments being identified you have been asked to ensure no other duplicates exist.

Removed the WHERE so all teachers are shown.

Toy will have to expand the GROUP BY:

GROUPBY trainer_name, student_name;