3.4 SQL queries and joins

Name: _____ Class: _____ Date: _____ **Total: 15 marks**

KEY WORDS SQL • SELECT • WHERE • JOIN

Objective

Use SQL to retrieve related data safely and accurately.

You must be able to:

- write a basic SELECT query with WHERE
- explain a JOIN condition
- identify the effect of a missing JOIN condition

1 Key knowledge and worked examples

■ Ask for exactly the rows needed

`SELECT name FROM courses WHERE level = 'beginner';` asks for names of beginner courses. SQL is **declarative**: it states what is wanted.

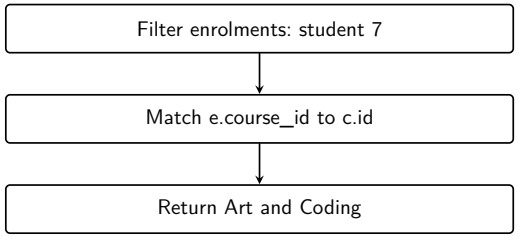
A **JOIN** combines related tables through matching keys: `enrolments e JOIN courses c ON c.id = e.course_id`. The `ON` condition prevents unrelated rows being paired.

■ Key knowledge: Trace a join result

A join pairs rows satisfying its ON condition. Without that condition, a Cartesian product can multiply unrelated rows; name the matching keys explicitly.

■ Worked example: Trace a join result

Courses have rows `(2, Art)` and `(5, Coding)`; enrolments are `(7,2)`, `(7,5)`, `(9,2)`. The query `SELECT e.student_id, c.name FROM enrolments e JOIN courses c ON c.id = e.course_id WHERE e.student_id = 7;` returns `(7, Art)` and `(7, Coding)`. The filter selects one student; the join condition attaches the matching course name, not every course name.



2 Practice

2.1 State which SQL word selects columns from a table. [1]

2.2 State the purpose of a WHERE condition. [1]

2.3 Explain why a JOIN needs an ON condition. [2]

3 Exam-style questions

3.1 Write a query to return `title` from `books` where `available` is true. [4]

3.2 Explain what can happen when two tables are joined without a condition linking their keys. [4]

3.3 Three enrolment rows are combined with two course rows using no matching condition. State how many pairings are produced and explain the condition that avoids unrelated pairings. [3]

Solutions

2.1

- SELECT.

2.2

- It filters rows to those matching a condition.

2.3

- It states which matching keys connect the two tables.
- It prevents unrelated rows being combined.

3.1

- `SELECT title FROM books WHERE available = true;`

3.2

- Every row may be paired with every row in the other table.
- The query returns many meaningless combinations.
- This can give the wrong result and use unnecessary resources.
- An ON condition matching the foreign and primary keys is required.

3.3

- a Cartesian product produces 3 x 2 = 6 pairings; join on `courses.id = enrolments.course_id` to pair only matching course records.