SQL

Database – Lecture 17

Contents

- Joins
- Inner Join
- Left Join
- Right Join
- Full Join
- Self Join
- Union

departments

id	name	phone
1	CSE	899876
2	Bangla	775465
3	English	674565
4	AIS	9787686

students

id	name	roll	dept_id
1	Alice	121	1
2	Brown	122	1
3	Charlie	123	2
4	Davis	124	3
5	Elis	125	2
6	Frank	126	2

Student_info

student_name	student_roll	dept_name

courses

id	title	course_code	level	credits	instructor_id
1	Introduction to Computer Sci	CSE101	Undergraduate	3.o	1
2	Data Structure	CSE201	Undergraduate	3.5	2
3	Database Systems	CSE301	Undergraduate	3.0	3

instructors

id	name	email	designation	phone	salary
1	Md. Rahim	rahim@univ.edu.bf	Professor	01710000001	90000.00
2	Ayesha Begum	ayesha@univ.edu.bd	Asst. Professor	01710000002	70000.00
3	Tanvir Hossain	tanvir@univ.edu.bd	Lecturer	01710000003	50000.00

learners

id	name	roll	email	date_of_birth	city
1	Hasibul Alam	20181001	arif@univ.edu.bd	1998-07-15	Dhaka
2	Shirin Akter	20181002	shirin@univ.edu.bd	1999-04-12	Chittagong
3	Arif Hossain	20181003	arif@univ.edu.bd	2000-01-25	Rajshahi

enrollments

id	student_id	course_id	enrollment_date	status
1	1	1	2023-01-15	Enrolled
2	2	2	2023-01-16	Enrolled
3	3	3	2023-01-17	Enrolled

assessments

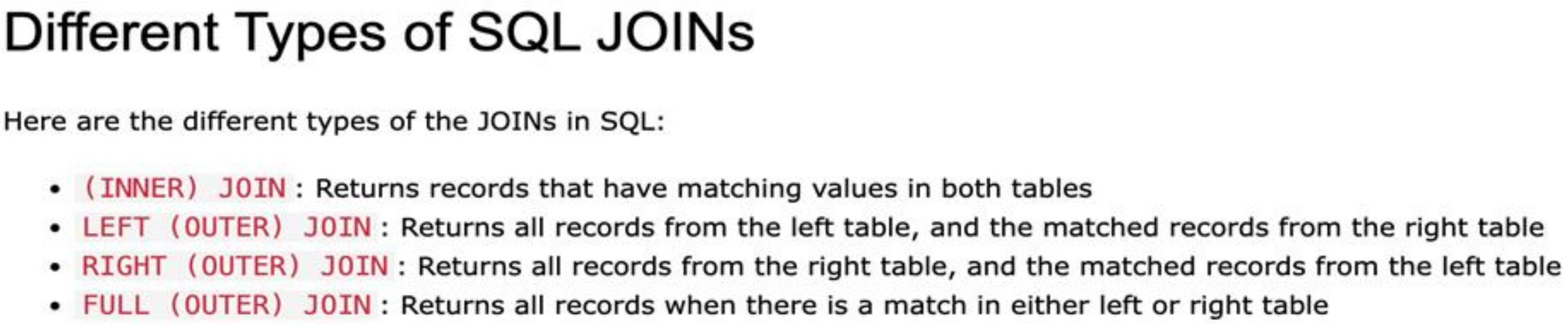
id	course_id	assessment_title	total_marks
1	1	Midterm Exam	100
2	2	Final Exam	100
3	3	Project Evaluation	50

results

id	student_id	assessment_id	marks_obtained
1	1	1	85
2	2	2	90
3	3	3	40

Joins

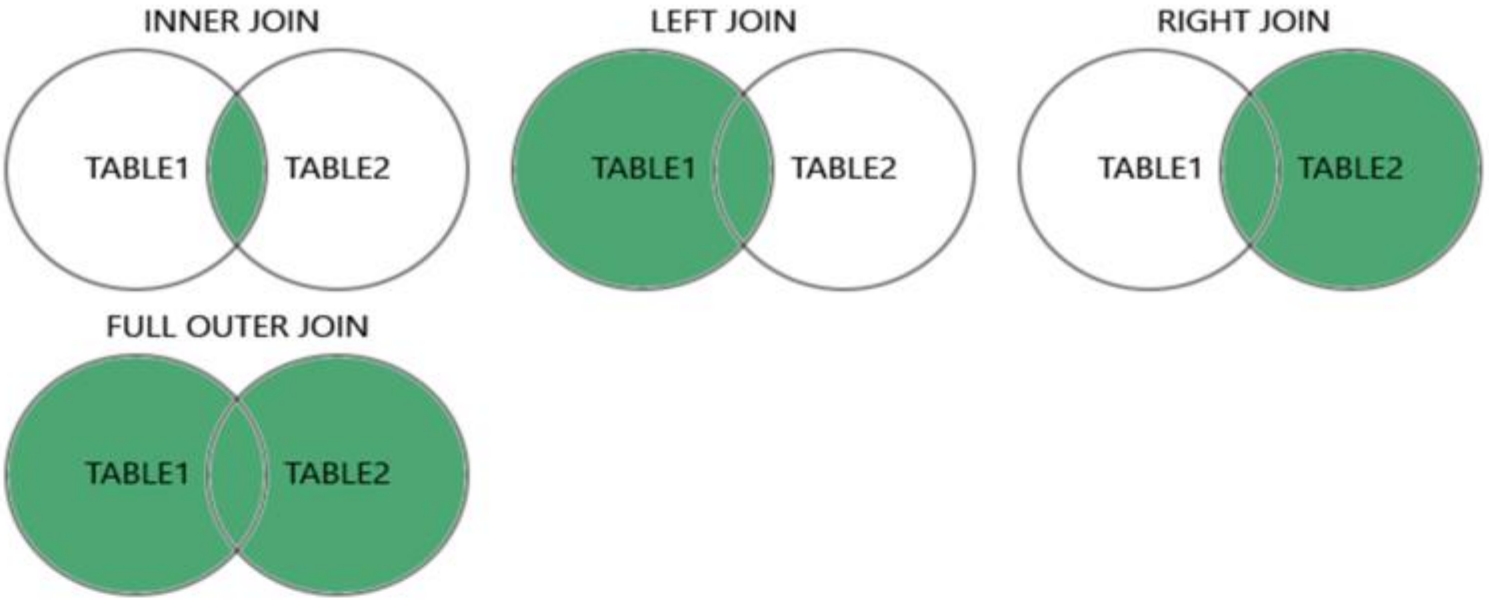
- A **JOIN** clause is used to combine rows from two or more tables, based on a related column between them.



Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table



Inner Join

- The `INNER JOIN` keyword selects records that have matching values in both tables.

- ***Syntax:***

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

- ***Example: Between Learners and Enrollments table***

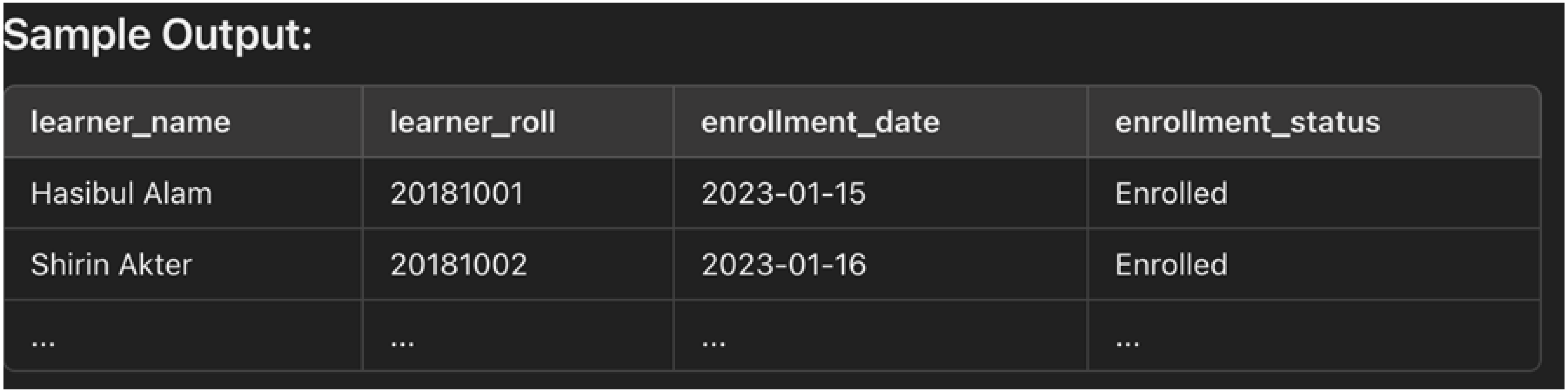
```
SELECT courses.title, courses.course_code,instructors.name,
FROM courses INNER JOIN instructors
ON instrctors.id = courses.instructors_id;
```

Inner Join

- ***Example:* Between Learners and Enrollments table**

```
SELECT
learners.name AS learner_name,
learners.roll AS learner_roll,
enrollments.enrollment_date AS enrollment_date,
enrollments.status AS enrollment_status
FROM
learners
INNER JOIN enrollments ON learners.id = enrollments.student_id
ORDER BY learners.name;
```

- ***Sample Output:***



Sample Output:

learner_name	learner_roll	enrollment_date	enrollment_status
Hasibul Alam	20181001	2023-01-15	Enrolled
Shirin Akter	20181002	2023-01-16	Enrolled
...	...	...	...

Left Join

- The `LEFT JOIN` keyword returns all records from the left table (table1), and the matching records from the right table (table2).

- The result is 0 records from the right side, if there is no match.

- ***Syntax:***

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

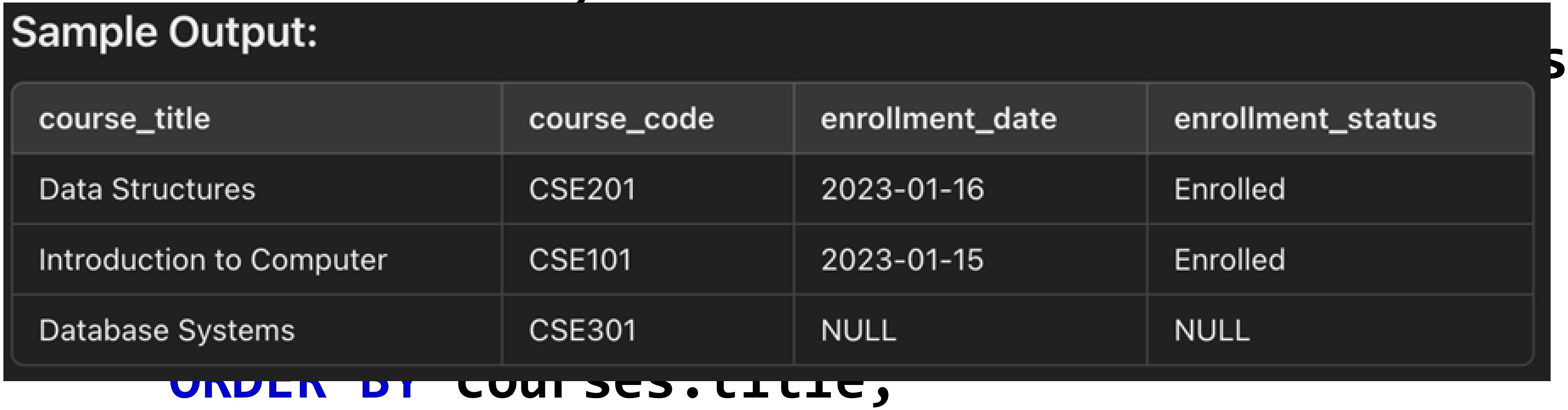
- ***Example:***

```
SELECT
courses.title AS course_title,
courses.course_code AS course_code,
enrollments.enrollment_date AS enrollment_date,
enrollments.status AS enrollment_status
FROM Courses
LEFT JOIN enrollments ON courses.id = enrollments.course_id
ORDER BY courses.title;
```

Left Join

- ***Example: Between Courses and Enrollments***

```
    SELECT
            courses.title AS course_title,
            courses.course_code AS course_code,
            enrollments.enrollment_date AS
  enrollment_date,
```



Sample Output:

course_title	course_code	enrollment_date	enrollment_status
Data Structures	CSE201	2023-01-16	Enrolled
Introduction to Computer	CSE101	2023-01-15	Enrolled
Database Systems	CSE301	NULL	NULL

```
    ORDER BY courses.title;
```

Right Join

- The `LEFT JOIN` keyword returns all records from the right table (table2), and the matching records from the left table (table2).

- The result is 0 records from the right side, if there is no match.

- ***Syntax:***

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

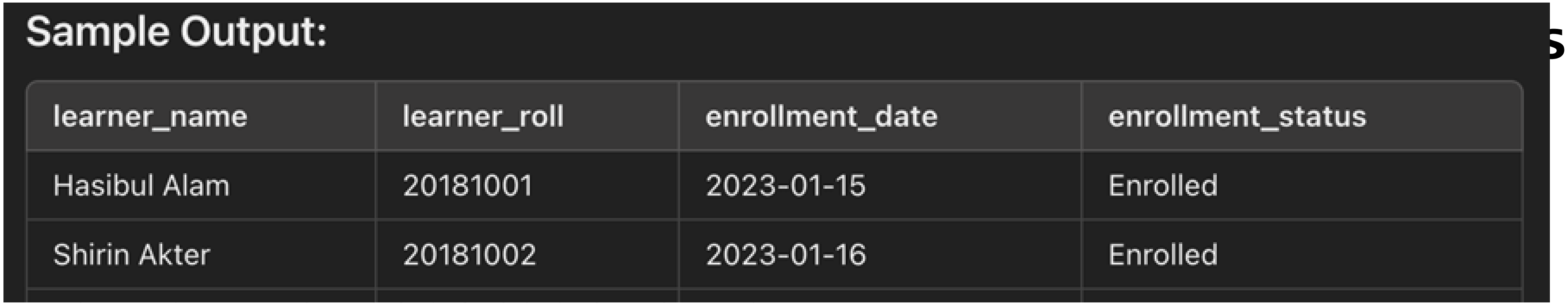
- ***Example: Between Learners and Enrollments table***

```
SELECT
 learners.name AS learner_name,
 learners.roll AS learner_roll,
 enrollments.enrollment_date AS enrollment_date,
 enrollments.status AS enrollment_status
FROM learners
RIGHT JOIN enrollments ON learners.id = enrollments.student_id
ORDER BY enrollments.enrollment_date;
```

Right Join

- ***Example: Between Learners and Enrollments table***

```
    SELECT
            learners.name AS learner_name,
            learners.roll AS learner_roll,
            enrollments.enrollment_date AS
 enrollment_date,
```



Sample Output:

learner_name	learner_roll	enrollment_date	enrollment_status
Hasibul Alam	20181001	2023-01-15	Enrolled
Shirin Akter	20181002	2023-01-16	Enrolled

```
    ORDER BY enrollments.enrollment_date;
```

Exercise

- Retrieve a list of all learners along with the courses they are enrolled in. Include the learner's name, course title, enrollment date, and enrollment status.

- List all instructors along with the courses they teach. Include the instructor's name, course title, and course code. Ensure that instructors who are not teaching any courses are also included in the result.

- List all courses along with their assigned instructors. Include the course title, course code, and instructor's name. Ensure that courses without an assigned instructor are also included in the result.

SOLUTION

```sql
1.  SELECT learners.name AS Learner_Name,
    courses.title AS Course_Title,
    enrollments.enrollment_date,
    enrollments.status
    FROM enrollments
    INNER JOIN
    learners ON enrollments.student_id = learners.id
    INNER JOIN
    courses ON enrollments.course_id = courses.id;
```

```sql
2. SELECT instructors.name AS Instructor_Name,
   courses.title AS Course_Title,
   courses.course_code
   FROM instructors
   LEFT JOIN
   courses ON instructors.id = courses.instructor_id
   ORDER BY instructors.name;
```

SOLUTION

```
3. SELECT courses.title AS Course_Title,
   courses.course_code,
   instructors.name AS Instructor_Name
   FROM courses
   RIGHT JOIN
   instructors ON courses.instructor_id =     instructors.id
  ORDER BY courses.title;
```

Exercises

- Retrieve the list of learners along with the courses they are enrolled in.
- List all learners, even if they haven’t enrolled in any course. If they haven’t enrolled, show NULL for the course title. (Left JOIN)
- Find the total marks obtained by each learner. (Join, Group by)
- List the learners enrolled in courses that are taught by instructors with a salary higher than 50,000 (join, where).
- Find the names of learners who are enrolled in the course "Database Systems". (join, where)
- Write a query to list the names of instructors and the titles of the courses they teach. (join)
- Write a query to find the names of learners who obtained more than 60 marks in any course.