SQL

Database – Lecture 12

Recap

- Limit
- Like
- Wildcards
- Aliases

Contents

- IN Operator
- BETWEEN Operator
- UNION Operator
- GROUP BY Clause
- HAVING Clause

courses

id	title	course_code	level	credits	instructor_id
1	Python Programming	P101	Basic	3	1
2	Database	DB202	Intermediate	4	2
3	Web Development	WD201	Intermediate	4	3
4	Web Design	WD102	Basic	3	1

instructors

id	name	email	designation	phone	salary
1	Alice B	ab@edu.com	Trainer	9876	60000
2	Bob Dan	bd@edu.com	Senior Trainer	456789	80000
3	Charlie Choe	ab@edu.com	Junior Trainer	98765	40000

learners

id	name	roll	email	date_of_birth	city
1	Alice Johnson	123	alice.j@students.com	10/10/99	Dhaka
2	Bob Williams	223	bob.w@students.com	22/6/99	Barishal
3	Carol Stevens	124	carol.s@students.com	12/11/01	Dhaka

enrollments

id	student_id	course_id	enrollment_date	status
1	1	1	1/9/23	1
2	2	2	15/8/23	2
3	3	3	1/9/23	1

assessments

id	course_id	assessment_title	total_marks
1	1	Python Midterm	20
2	2	Quiz 1	10
3	3	Project	20

results

id	student_id	assessment_id	marks_obtained
1	1	1	85
2	2	2	45
3	3	3	80

SQL IN Operator

- The **IN** operator allows you to ***specify multiple values*** in a **WHERE** clause.

- The **IN** operator is a shorthand for multiple **OR** conditions.

- ***Syntax:***

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

- ***Example:***

```
SELECT * FROM courses
WHERE level  IN (‘Basic’, ‘Intermediate’);
```

IN Example

- The following SQL statement selects all learners who are from Dhaka and Barishal that are located in "Germany", "France" or "UK":

  -
    ```
    SELECT * FROM learners
    WHERE city IN (‘Dhaka’, ’Barishal’);
    ```
- Who are NOT from Barishal:

  -
    ```
    SELECT * FROM learnres
    WHERE city NOT IN (‘Barishal’);
    ```
- Selects all learners who are from the same city Alice Johnson:

  -
    ```
    SELECT * FROM learners
    WHERE city IN (SELECT city FROM learners
    WHERE name=‘Alice Johnson’);
    ```

SQL BETWEEN Operator

- The BETWEEN operator ***selects values within a given range***. The values can be numbers, text, or dates.

- The BETWEEN operator is ***inclusive***: ***begin and end values are included***.

- ***Syntax:***

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN  value1 AND value2;
```

- ***Example:***

```
SELECT * FROM instructors
WHERE Salary BETWEEN 40000  AND 80000;
```

SQL UNION Operator

- The UNION operator is used to combine the result-set of two or more SELECT statements.

    1. Every SELECT statement within UNION must have the same number of columns

    2. The columns must also have similar data types

    3. The columns in every SELECT statement must also be in the same order

- ***Syntax:***

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

- ***Example:***

```
SELECT email FROM instructors
UNION
SELECT email FROM learners;
```

SQL GROUP BY

- The **GROUP BY** statement groups rows that have the same values into summary rows
- The **GROUP BY** statement is often used with aggregate functions (**COUNT()**, **MAX()**, **MIN()**, **SUM()**, **AVG()**) to group the result-set by one or more columns.

- ***Syntax:***

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

- ***Example:***

```
SELECT designation, COUNT(designation)
FROM instructors
GROUP BY designation;
```

Example

-
  ```
  SELECT city, COUNT(id)
  FROM learners
  GROUP BY city;
  ```
-

SQL HAVING Clause

- Filters the grouped results based on the aggregate function like COUNT(), MAX(), MIN(), SUM(), AVG()

- ***Syntax:***

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

- ***Example:***

```
SELECT city, COUNT(id)
FROM learners
GROUP BY city
HAVING COUNT(id) > 2;
```

Example

- **Retrieve details of courses where the course_code is either "P101" or "DB202" or “WD201”.**

```
SELECT *
FROM courses
WHERE course_code IN ('P101', 'DB202', ‘WD201’);
```

- **Find the total number of learners from each city.**

```
   SELECT city, COUNT(*) AS total_learners
FROM learners
GROUP BY city;
```

SOLUTION

- Get the instructors whose salary is between 50,000 and 100,000.

```
        SELECT name, salary
FROM instructors
WHERE salary BETWEEN 50000 AND 100000;
```

- Fetch the names and email addresses of both instructors and learners.

```
SELECT name, email
FROM instructors
UNION
SELECT name, email
FROM learners;
```

- Retrieve cities where more than 5 learners are enrolled.

```
SELECT city, COUNT(*) AS total_learners
FROM learners
GROUP BY city
HAVING COUNT(*) > 5;
```

Exercise

- *Get the instructors whose salary is between 50,000 and 100,000.*
- *Fetch the names and email addresses of both instructors and learners.*
- *Retrieve cities where more than 5 learners are enrolled.*
- Find the total number of students in enrolled in each course.
- Categorize each course based on the course level.
- How many instructors are there for each designation?
- Find the name and number of learners from cities where the number of learners is greater than two.
- Find the list of instructors whose id between 1 to 10.
- Find the list of instructors those who get same salary of Alice B.

Thank You