

## SQL | DIVISION

Division in SQL is typically required when you want to find out entities that are interacting with all entities of a set of different types of entities. The division operator is used when we have to evaluate queries that contain the keyword 'all'.

SQL में विभाजन की आवश्यकता आमतौर पर तब होती है जब आप उन संस्थाओं का पता लगाना चाहते हैं जो विभिन्न प्रकार की संस्थाओं के समूह की सभी संस्थाओं के साथ बातचीत कर रही हैं। विभाजन ऑपरेटर का उपयोग तब किया जाता है जब हमें उन केरीज़ का मूल्यांकन करना होता है जिनमें कीवर्ड 'सभी' होता है।

### When to Use the Division Operator

You typically require the division operator in scenarios where you need to ascertain entities that meet all specified conditions across different datasets. Here are a few common examples:

आपको आमतौर पर उन परिदृश्यों में विभाजन ऑपरेटर की आवश्यकता होती है जहाँ आपको उन संस्थाओं का पता लगाने की आवश्यकता होती है जो विभिन्न डेटासेट में सभी निर्दिष्ट शर्तों को पूरा करती हैं। यहाँ कुछ सामान्य उदाहरण दिए गए हैं:

Identifying people who have accounts in every bank within a particular city.

किसी विशेष शहर के भीतर हर बैंक में खाते रखने वाले लोगों की पहचान करना।

Determining students who have enrolled in all necessary courses to qualify for graduation.

स्नातक की योग्यता प्राप्त करने के लिए सभी आवश्यक पाठ्यक्रमों में नामांकित छात्रों का निर्धारण करना।

These use cases involve checking against all elements of a subset, thereby necessitating a division-like operation.

इन उपयोग मामलों में एक उपसमूह के सभी तत्वों की जाँच करना शामिल है, जिससे विभाजन-जैसे ऑपरेशन की आवश्यकता होती है।

In all these queries, the description after the keyword 'all' defines a set which contains some elements and the final result contains those units who satisfy these requirements.

इन सभी क्वेरीज़ में, कीवर्ड 'सभी' के बाद का विवरण एक सेट को परिभाषित करता है जिसमें कुछ तत्व होते हैं और अंतिम परिणाम में वे इकाइयाँ होती हैं जो इन आवश्यकताओं को पूरा करती हैं।

### Step 1: Create Tables

```
CREATE TABLE Students (  
    StudentID INT,  
    CourseID INT  
);
```

Create tables courses-2

```
CREATE TABLE Courses (  
    CourseID INT  
);
```

**Step 2: Insert Data into Tables**

```
INSERT INTO Students (StudentID, CourseID) VALUES
```

```
(1, 101),
```

```
(1, 102),
```

```
(2, 101),
```

```
(2, 103),
```

```
(3, 101),
```

```
(3, 102),
```

```
(3, 103);
```

**Insert Data into Courses**

```
INSERT INTO Courses (CourseID) VALUES
```

```
(101),
```

```
(102),
```

```
(103);
```

**Step 3: Count the Total Number of Courses**

```
SELECT COUNT(*) AS TotalCourses
```

```
FROM Courses;
```

**Step 4: Count the Number of Courses Each Student is Enrolled In**

```
SELECT StudentID, COUNT(DISTINCT CourseID) AS EnrolledCourses
```

```
FROM Students
```

```
GROUP BY StudentID;
```

### **Step 5: Identify Students Enrolled in All Courses**

SELECT StudentID

FROM Students

GROUP BY StudentID

HAVING COUNT(DISTINCT CourseID) = (SELECT COUNT(\*) FROM  
Courses);

### **Step 6: Clean Up (Optional)**

DROP TABLE Students;

DROP TABLE Courses;