**Calculating Free-to-Paid Conversion Rate with SQL Project**

**Create the Subquery**

This practice project allows you to apply your SQL knowledge to a real-world dataset. Once you complete all tasks, you will have found the answer to the following questions:

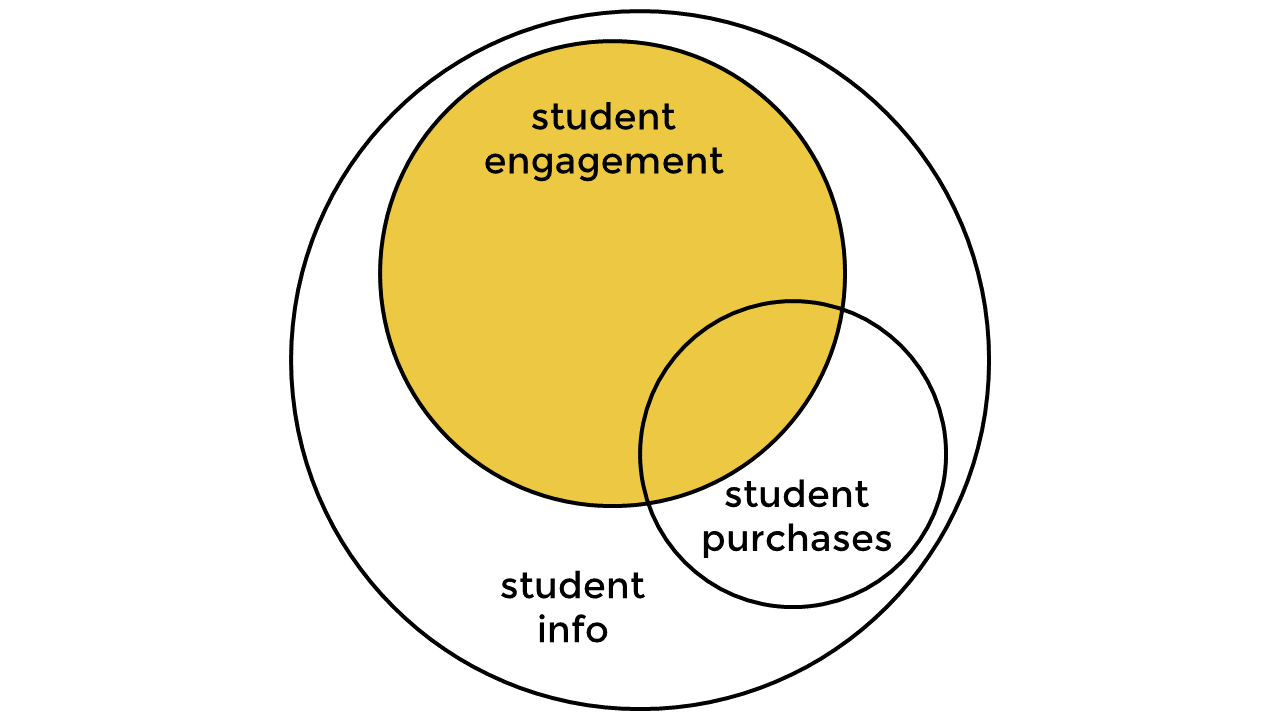
* What is the free-to-paid conversion rate of students who have watched a lecture on the 365 platform?
* What is the average duration between the registration date and when a student has watched a lecture for the first time (date of first-time engagement)?
* What is the average duration between the date of first-time engagement and when a student purchases a subscription for the first time (date of first-time purchase)?
* How can we interpret these results, and what are their implications?

Import the db\_course\_conversions database—stored in the db\_course\_conversions.sql file—into your schemas and study its content. Then, by appropriately joining and aggregating the tables, create a new result dataset comprising the following columns:

* student\_id – (int) the unique identification of a student
* date\_registered – (date) the date on which the student registered on the 365 platform
* first\_date\_watched – (date) the date of the first engagement
* first\_date\_purchased – (date) the date of first-time purchase (NULL if they have no purchases)
* date\_diff\_reg\_watch – (int) the difference in days between the registration date and the date of first-time engagement
* date\_diff\_watch\_purch – (int) the difference in days between the date of first-time engagement and the date of first-time purchase (NULL if they have no purchases)

**Hint:***Research the DATEDIFF function in MySQL.*

Note the Venn diagram below.



The resulting set you retrieve should include the student IDs of students entering the diagram’s shaded region. Additionally, your objective is to determine the conversion rate of students *who have already watched a lecture*. Therefore, filter your result dataset so that the date of first-time engagement comes before (or is equal to) the date of first-time purchase.

**Sanity check:***The number of records in the resulting set should be 20,255.*

To complete the task, follow the **instructions** below.

1. First, remember to import the db\_course\_conversions database and refresh the Schemas pane to see it appear. Apply theUSEkeyword to use the named database as the default (current) one.
2. Retrieve the columns one by one as listed in the task. Use theMINaggregate function to find the first-time engagement and purchase dates. Apply the DATEDIFF function to see the difference in the respective days.

SELECT

???,

???,

MIN(???) AS first\_date\_watched,

MIN(???) AS first\_date\_purchased,

DATEDIFF(???) AS days\_diff\_reg\_watch,

DATEDIFF(???) AS days\_diff\_watch\_purch

1. Next, consider how to join the three tables to retrieve the highlighted records in the Venn diagram.

FROM

student\_engagement e

???

student\_info i ON ???

???

student\_purchases p ON ???

1. Applying the MIN aggregate function in the previous step requires grouping the results appropriately.

SELECT

???,

???,

MIN(???) AS first\_date\_watched,

MIN(???) AS first\_date\_purchased,

DATEDIFF(???) AS days\_diff\_reg\_watch,

DATEDIFF(???) AS days\_diff\_watch\_purch

FROM

student\_engagement e

???

student\_info i ON ???

???

student\_purchases p ON ???

GROUP BY ???;

1. Filter the data to exclude the records where the date of first-time engagement comes later than the date of first-time purchase. Remember to keep the students who have never made a purchase.

SELECT

???,

???,

MIN(???) AS first\_date\_watched,

MIN(???) AS first\_date\_purchased,

DATEDIFF(???) AS days\_diff\_reg\_watch,

DATEDIFF(???) AS days\_diff\_watch\_purch

FROM

student\_engagement e

???

student\_info i ON ???

???

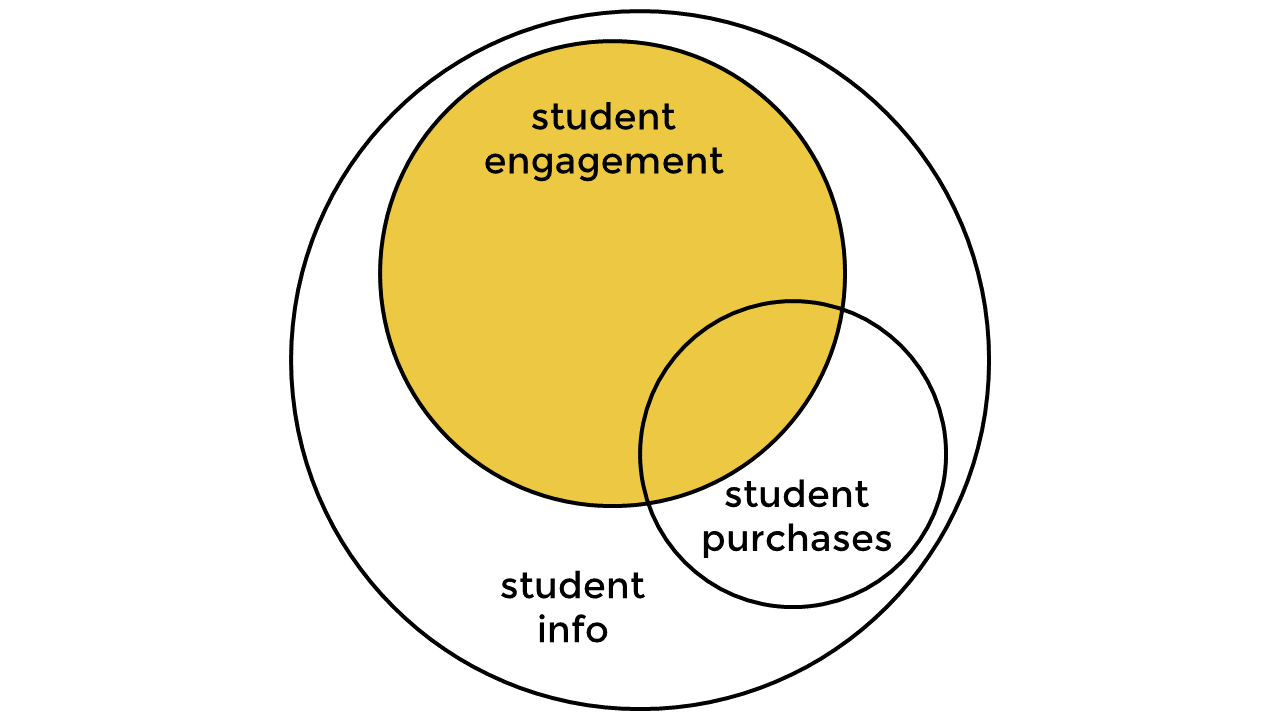
student\_purchases p ON ???

GROUP BY ???

HAVING ???;

**Create the Main Query**

Excellent work! You’ve created a temporary result set adhering to the following Venn diagram and columns:



* student\_id – (int) the unique identification of a student
* date\_registered – (date) the date on which the student registered on the 365 platform
* first\_date\_watched – (date) the date of first-time engagement
* first\_date\_purchased – (date) the date of first-time purchase (NULL if they have no purchases)
* date\_diff\_reg\_watch – (int) the difference in days between the registration date and the date of first-time engagement
* date\_diff\_watch\_purch – (int) the difference in days between the date of first-time engagement and the date of first-time purchase (NULL if they have no purchases)

In this task, you should use the subquery you’ve created and retrieve the following three metrics.

* **Free-to-Paid Conversion Rate:**

This metric measures the proportion of engaged students who choose to benefit from full course access on the 365 platform by purchasing a subscription after watching a lecture. It is calculated as the ratio between:

* + The number of students who watched a lecture and purchased a subscription on the same day or later.
  + The total number of students who have watched a lecture.

Convert the result to percentages and call the field conversion\_rate.

* **Average Duration Between Registration and First-Time Engagement:**

This metric measures the average duration between the date of registration and the date of first-time engagement. This will tell us how long it takes, on average, for a student to watch a lecture after registration. The metric is calculated by finding the ratio between:

* + The sum of all such durations.
  + The count of these durations, or alternatively, the number of students who have watched a lecture.

Call the field av\_reg\_watch.

* **Average Duration Between First-Time Engagement and First-Time Purchase:**

This metric measures the average time it takes individuals to subscribe to the platform after viewing a lecture. It is calculated by dividing:

* + The sum of all such durations.
  + The count of these durations, or alternatively, the number of students who have made a purchase.

Call the field av\_watch\_purch.

Use the following **instructions** to carry out the task.

1. Surround the subquery you created in the previous part (Create the Subquery) in parentheses and give it an alias, say a.
2. Consider the skeleton below. Fill in the appropriate columns to retrieve the three metrics described in this task. The results are rounded to two decimal places for clarity. Don’t forget to convert the conversion\_rate metric to percentages.

SELECT

ROUND(COUNT(???) / COUNT(???),

2) AS conversion\_rate,

ROUND(SUM(???) / COUNT(???),

2) AS av\_reg\_watch,

ROUND(SUM(???) / COUNT(???),

2) AS av\_watch\_purch

FROM

(

-- Subquery

) a;

Interpretation

Well done in reaching this final part of the project! What you should’ve retrieved by now are the free-to-paid conversion rate of students who’ve started a lecture, the average duration between the registration date and date of first-time engagement, and the average duration between the dates of first-time engagement and first-time purchase. Now, it’s time to interpret the numbers you’ve obtained.

First, consider the conversion rate and compare this metric to industry benchmarks or historical data. Second, examine the duration between the registration date and date of first-time engagement. A short duration—watching on the same or the next day—could indicate that the registration process and initial platform experience are user-friendly. At the same time, a longer duration may suggest that users are hesitant or facing challenges. Third, regarding the time it takes students to convert to paid subscriptions after their first lecture, a shorter span would suggest compelling content or effective up-sell strategies. A longer duration might indicate that students have been waiting for the product to be offered at an exclusive price.

Optional: Using a tool different from SQL (e.g., Python), calculate the median and mode values of the date difference between registering and watching a lecture. Do the same for the date difference between watching a lecture and purchasing a subscription. Compare the two metrics of each set to their respective mean values. To interpret the results even better, create a distribution graph and try to understand the relationship between these metrics (mean, median, and mode). Focus on the following key points.

Distribution Symmetry

The distribution is likely symmetrical when the mean, median, and mode are equal or very close, forming a bell curve. If they differ, the data might be skewed to the left—indicated by a long tail on the left side—or to the right with a long tail on the right side.

Outliers

If the mean is much higher or lower than the median, it suggests that there are outliers. For instance, if the average time to purchase a subscription is significantly higher than the median, it may imply that a few students took an exceptionally long time to decide.

Common Patterns

If a specific value or set of values has a high frequency—corresponding to the mode of the dataset—it can spotlight common behaviors. For instance, a mode of zero or one day between registration and lecture viewing suggests that most students begin watching on the registration day or the day after.

**Quiz**

**Question 1:**

When did a student with ID 268727 first watch a lecture?

2022Mar 27

**Question 2:**

Regarding the same student's first subscription purchase, which statement is accurate?

They have never made a subscription purchase.

**Question 3:**

What is the approximate free-to-paid conversion rate of students who have watched a lecture on the 365 platform?

11%

**Question 4:**

What is the approximate average duration between the registration date and the date of first-time engagement?

8days

**Question 5:**

What is the approximate average duration between the date of first-time engagement and the date of first-time purchase?

17 days