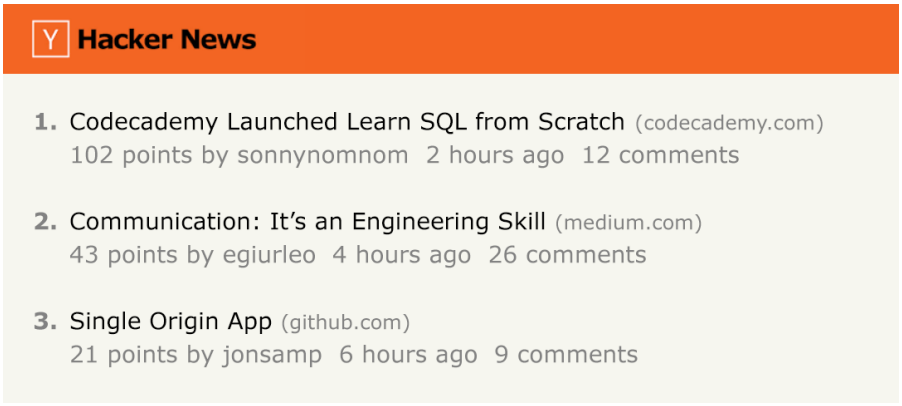
How to Hack Hacker News



Hacker News is a popular website run by Y Combinator. It's widely known by people in the tech industry as a community site for sharing news, showing off projects, asking questions, among other things.

In this project, you will be working with a table named `hacker_news` that contains stories from Hacker News since its launch in 2007. It has the following columns:

- `title`: the title of the story
- `user`: the user who submitted the story
- `score`: the score of the story
- `timestamp`: the time of the story
- `url`: the link of the story
-

This data was kindly made publicly available under the MIT license.
Let's get started!

If you get stuck during this project or would like to see an experienced developer work through it, click "**Get Help**" to see a **project walkthrough video**.

Pre-Gaming for Aggregates

1. Start by getting a feel for the `hacker_news` table!

Let's find the most popular Hacker News stories:

```
SELECT title, score
FROM hacker_news
ORDER BY score DESC
LIMIT 5;
```

What are the top five stories with the highest `score`s?

2. Recent studies have found that online forums tend to be dominated by a small percentage of their users (1-9-90 Rule). *Is this true of Hacker News? Is a small percentage of Hacker News submitters taking the majority of the points?* First, find the total `score` of all the stories.

```
select sum(score)
from hacker_news;
```

3. Next, we need to pinpoint the users who have accumulated a lot of points across their stories. Find the individual users who have gotten combined `score`s of more than 200, and their combined `score`s. `GROUP BY` and `HAVING` are needed!

```
select user, sum(score) as total_score
from hacker_news
group by 1
having total_score > 200
order by 2 desc;
```

4. Then, we want to add these users' `score`s together and divide by the total to get the percentage. Add their scores together and divide it by the total sum. Like so:

```
SELECT (1.0 + 2.0 + 3.0) / 6.0;
```

So, is Hacker News dominated by these users?

```
select 100 * (517 + 309 + 304 + 282) / 6366.0 as pct;
```

5. Oh no! While we are looking at the power users, some users are rickrolling — tricking readers into clicking on a link to a funny video and claiming that it links to information about coding. The `url` of the video is: https://www.youtube.com/watch?v=dQw4w9WgXcQ

   How many times has each offending user posted this link? Which sites feed Hacker News?

```
select user, count(*)
from hacker_news
where url = 'https://www.youtube.com/watch?v=dqw4w9wgxcq'
group by 1
order by 2 desc;
```

6. Hacker News stories are essentially links that take users to other websites. *Which of these sites feed Hacker News the most*: *GitHub*, *Medium*, *or New York Times*? First, we want to categorize each story based on their source. We can do this using a `CASE` statement:

```
SELECT CASE
   WHEN url LIKE '%github.com%' THEN 'GitHub'
   -- WHEN statement here
   -- WHEN statement here
```

```
   -- ELSE statement here
  END AS 'Source'
FROM hacker_news;
```

Fill in the other `WHEN` statements and the `ELSE` statement.

```
select case
   when url like '%github.com%' then 'github'
   when url like '%medium.com%' then 'medium'
   when url like '%nytimes.com%' then 'new york times'
   else 'na'
  end as 'source'
from hacker_news
limit 10;
```

7. Next, build on the previous query: Add a column for the number of stories from each URL using `COUNT()`. Also, `GROUP BY` the `CASE` statement. Remember that you can refer to a column in `GROUP BY` using a number.

```
select case
   when url like '%github.com%' then 'github'
   when url like '%medium.com%' then 'medium'
   when url like '%nytimes.com%' then 'new york times'
   else 'na'
  end as 'source', count(url)
from hacker_news
group by 1
limit 10;
```

8. Every submitter wants their story to get a high score so that the story makes it to the front page, but… *What's the best time of the day to post a story on Hacker News?* Before we get started, let's run this query and take a look at the `timestamp` column:

```
SELECT timestamp
FROM hacker_news
LIMIT 10;
```

Notice that the values are formatted like:
`2018-05-08T12:30:00Z`
If you ignore the `T` and `Z`, the format is:
`YYYY-MM-DD HH:MM:SS`

9. SQLite comes with a `strftime()` function - a very powerful function that allows you to return a formatted date. It takes two arguments: `strftime(format, column)` Let's test this function out:

```
SELECT timestamp,
   strftime('%H', timestamp)
```

```
FROM hacker_news
GROUP BY 1
LIMIT 20;
```

What do you think this does? Open the hint if you'd like to learn more.

10. Okay, now we understand how `strftime()` works. Let's write a query that returns three columns:

1. The hours of the `timestamp`
2. The *average* `score` for each hour
3. The *count* of stories for each hour

```
select strftime('%H', timestamp) as hour, avg(score), count(url)
from hacker_news
group by 1
order by 1
limit 10;
```

11. Let's edit a few things in the previous query:

- Round the average `score`s (`ROUND()`).
- Rename the columns to make it more readable (`AS`).
- Add a `WHERE` clause to filter out the `NULL` values in `timestamp`.

Take a look at the result again:
What are the best hours to post a story on Hacker News?

```
select strftime('%H', timestamp) as hour, round(avg(score)) as avg_score,
count(url) as stories
from hacker_news
where timestamp is not null
group by 1
order by 1
limit 10;
```