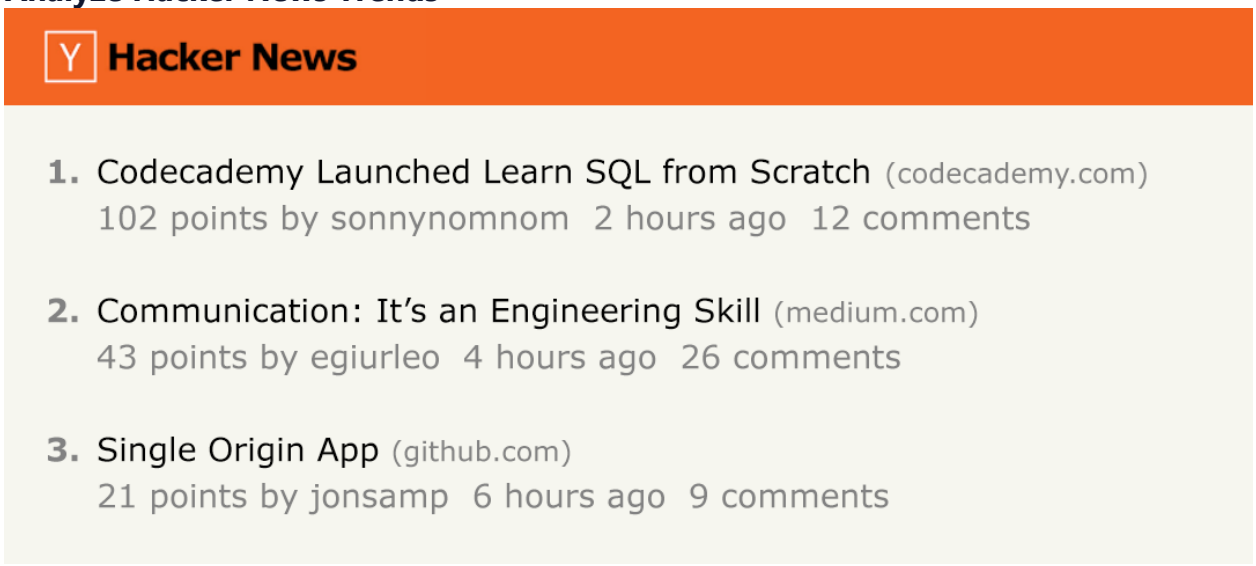
PROJECT

Analyze Hacker News Trends



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!

Tasks

10/11 Complete
Mark the tasks as complete by checking them off

Understanding the dataset

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?
Hint
Using `LIMIT` caps the number of rows in the result.

It is a simple way to keep queries from taking too long to run if you are dealing with a big dataset.

`ORDER BY` simply sorts the `score` column.

The most popular stories are:

1. 'Penny Arcade - Surface Pro 3 update'
2. 'Hacking The Status Game'
3. 'Postgres CLI with autocompletion and syntax highlighting'
4. 'Stephen Fry hits out at 'infantile' culture of trigger words and safe spaces'
5. 'Reversal: Australian Govt picks ODF doc standard over Microsoft'

Hacker News Moderating

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.
Hint

```
SELECT SUM(score)
FROM hacker_news;
```

The total `score` of this table is 6366.
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!

Hint

```
SELECT user, SUM(score)
FROM hacker_news
GROUP BY user
HAVING SUM(score) > 200
ORDER BY 2 DESC;
```

`HAVING` does not support aliases in the same way that `ORDER BY` does, so use the full column name.

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?

Hint

The query should look like:

```
SELECT (517 + 309 + 304 + 282) / 6366.0;
```

That is ≈ 22%.

These 4 users have a combined 22% of the total scores in the table. Jeez!

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?

Hint

You can `GROUP BY` the users and use `WHERE` to restrict `url`:

```
SELECT user,
   COUNT(*)
FROM hacker_news
WHERE url LIKE '%watch?v=dQw4w9WgXcQ%'
GROUP BY user
ORDER BY COUNT(*) DESC;
```

Rewrite this using column reference numbers instead of column names:

```
-- Hacker News Moderating

SELECT user,
   COUNT(*)
FROM hacker_news
WHERE url LIKE '%watch?v=dQw4w9WgXcQ%'
GROUP BY 1
ORDER BY 2 DESC;
```

Dear @sonnynomnom, you're banned.
Dear @scorpiosister, warning!

Which sites feed Hacker News?

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.
Hint
Your query should look like:

```
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 'Other'
  END AS 'Source'
FROM hacker_news;
```

`--` starts a single line comment. The text after `--` will be ignored (not executed).

Note: If we want to be more accurate, we should use `url LIKE '%github%'` because some GitHub pages end with **.io** instead of **.com**.

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.
Hint

```
-- Which sites feed Hacker News?

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 'Other'
  END AS 'Source',
  COUNT(*)
FROM hacker_news
GROUP BY 1;
```

The number of times stories are linked to:

- **GitHub** - 23
- **Medium** - 12
- **New York Times** - 13

What's the best time to post a story?

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`

Hint
The `T` is just the separator between the date and time. You can read it as an abbreviation for 'Time'.

The `Z` stands for the Zero timezone, as it is offset by 0 from the Coordinated Universal Time (UTC).

If you don't look at the `T` and `Z`, it is easier to see the pattern in the `timestamp` column.

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.

Hint

This returns the hour, `HH`, of the `timestamp` column!

For `strftime(__, timestamp)`:

- `%Y` returns the year (YYYY)
- `%m` returns the month (01-12)
- `%d` returns the day of the month (1-31)
- `%H` returns 24-hour clock (00-23)
- `%M` returns the minute (00-59)
- `%S` returns the seconds (00-59)

if `timestamp` format is `YYYY-MM-DD HH:MM:SS`.

Read more on the SQLite documentation.

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

Hint

```
SELECT strftime('%H', timestamp),
   AVG(score),
   COUNT(*)
FROM hacker_news
GROUP BY 1
ORDER BY 1;
```

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?

Hint

The `ROUND()` function returns a number or column rounded to a certain number of decimal places.

For example, `ROUND(temp, 2)` rounds the `temp` values to `2` decimal places.

The final query should look something like:

```
-- What's the best time to post a story?

SELECT strftime('%H', timestamp) AS 'Hour',
   ROUND(AVG(score), 1) AS 'Average Score',
   COUNT(*) AS 'Number of Stories'
FROM hacker_news
WHERE timestamp IS NOT NULL
GROUP BY 1
ORDER BY 1;
```

The best hours are in the morning around 7 am and afternoon around 6 - 8 pm! Monster difference!