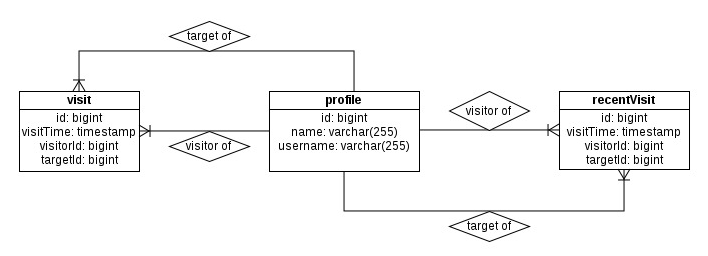
**Profile Database Design**

In this document, we will refer to the 2 APIs as below:

* view profile API: API to store the views of user profiles
* profile view history API: API to view the users who viewed this user’s profile in the past

This is the ER diagram of the profile application database:

There are 3 tables:

1) **Profile**: This table keeps user profile data with profile id, name and username.

2) **Visit:** This table keeps the profile view history. Each row of this table will correspond to one profile view made by another user. The fields in this table are:

* visitTime: The time when the profile is viewed.
* visitorId: Profile id of the user who view.
* targetId: Profile id of the user who is the target of the view.

3) **RecentVisit:** This table has the same structure as **Visit** table**.** The difference is that it only keeps the recent profile views to meet the application requirements, namely:

a. should include the viewer’s user id and also the date/time of the view

b. should NOT include more than 10 items

c. should NOT include views older than 10 days

The purpose of **RecentVisit** table is to allow faster retrieval of recent profile view history. When there is a profile view, a record will be created in both **Visit** and **RecentVisit** tables. However, while the data in **Visit** table will not be removed, records will be removed from **RecentVisit** table when:

* A new row is added to the table and the number of rows for the target profile exceeds 10, then the oldest record will be removed.
* A daily batch job kicks in and remove all the records which are more than 10 days old

The removal of recent visit records is to minimize the number of relevant data which will be retrieved by the profile view history API. For each profile, there won't be more than 10 entries in the **RecentVisit** table at any point of time. The daily batch job aims to shrink the **RecentVisit** table data even further. If there are millions of views per day, the daily batch job will potentially reduce a significant number of relevant data everyday. An index on the targetId field of **RecentVisit** table is also added to speed up this API query lookup since targetId is used for comparison.

The tradeoff to have this **RecentVisit** table are**:**

* There will be additional latency to the view profile API because of the time taken to add row and maintain data for the table.
* Additional space required to keep the recent visit data.

However, given the relatively small size of the **RecentVisit** data as compared to **Visit** table data (The more the number of views is, the bigger the difference is), the benefits should outweigh the drawbacks.