The table is stored in MySQL with the InnoDB engine, the Dynamic row format, and the utf8mb4 character set. With this setting, MySQL has an index key prefix length limit of 3072 bytes.

We have 3 additional requirements:

1/ The table can grow to millions of URLs. However, we can't have the user wait more than a couple of seconds for inserting CSV files with tens of thousands of URLs. What are the usual approaches?

Batch Processing: Import the data in small batches.This was implemented on the current example app. Simply flushing and clearing the entity manager.

Async ("background") Processing: Import the data in the background with a message queue (Kafka, RabbitMQ) or Symfony's Messenger component.

2/ The URLs can be up to 2048 characters. What problem are we facing? How to solve that problem?

MySQL's InnoDB with utf8mb4 character set and Dynamic row format limits index key prefix length to 3072 bytes. A single utf8mb4 character can take up to 4 bytes, so a 2048 character URL can exceed the maximum index length.

To always fit within the MySQL index length limit and allow for quick lookups of the URLs a urlHash column was created.

PS: for the actual test app purposes I've just used a SQLite database

3/ We want all versions of the same URL to match. For instance, if 2 URLs differ only by the scheme, they are considered the same URL. If one URL has the default port 80 and another has no port, they are considered the same URL. If the URLs have the same query parameters and values but in different orders, they are considered the same URL.
Note: you don't have to implement all conditions of this constraint. Just show that you understand what is required.

A "normalizeURL" method was created to remove the schema, doors and the order of query parameters.