

Requirements for creating interfaces using Laravel:

- For each table, user should be able to **view**, **insert**, **edit** and **delete** data.
- There should be a **Search field** on each view of the table
- **Pagination** of the records should be implemented (each page should have a max of 10 records and navigation buttons eg next, previous)
- There shouldn't be total deletion of a record - this can pose a legal or operational threat when certain records are deleted from the database through the user interface e.g. an employee record being deleted and there being no proof that he/she worked there, or deletion by accident.

A solution of this is a db design implementation whereby the delete button on the interface doesn't actually *delete* but rather *updates* a specific column of that record, this column is called '**status**'. Strong entities of the database are commonly given this column, its purpose is:

- To store the values **active** or **inactive** for every record within a specific table
- To update "deleted" records ; when a record gets "deleted" by the user, an update query will be issued that will set the status field of that record to 'inactive'
- To fetch "all" data from a table; it is used within a select query as a condition to distinguish the "deleted" records from the undeleted ones to appear as if it is fetching all records within a table e.g. instead of using `select * from table`; we use `select * from table where status = 'active'`; for the user this will appear as if the records have been deleted