CLAIMS:

1. A method comprising:

copying multiple rows from an origin database to a target database;

iteratively performing the following steps for each of the multiple rows in the target database that were copied from the origin database to calculate a target hash array:

calculating for that row in the target database a first hash value;

mapping the first hash value to a position in the target hash array, the target hash array being longer than the first hash value; and

incorporating the first hash value into the target hash array at the position;

iteratively performing the following steps for each of the multiple rows in the origin database that were copied to the target database to calculate an origin hash array:

calculating for that row in the origin database a second hash value;

mapping the second hash value to a position in the origin hash array, the origin hash array being longer than the second hash value; and

incorporating the second hash value into the origin hash array at the position; and

comparing the origin hash array to the target hash array to verify the multiple rows in the target database against the multiple rows in the origin database.

2. The method of claim 1, wherein mapping the first hash value to the position in the target hash array is based on the first hash value.

3. The method of claim 2, wherein mapping the first hash value to the position in the target hash array is based on one or more bytes of the first hash value.

4. The method of claim 3, wherein the one or more bytes are leading bytes.

5. The method of claim 3 or 4, wherein mapping the first hash value to the position in the target hash array comprises multiplying the one or more bytes of the first hash value by a mapping factor and using the result of the multiplying as an index in the target hash array.

6. The method of claim 5, wherein multiplying the mapping factor with a maximum value of the one or more bytes results in a length of the target hash array.

7. The method of any one of the preceding claims, wherein the target hash array wraps around in case the first hash value overflows the target hash array.

8. The method of any one of the preceding claims, wherein incorporating the first hash value into the target hash array comprises adding the first hash value to values in the target hash array at the position.

9. The method of any one of the preceding claims, wherein the first hash value has 32 bytes and the target hash array has 256 bytes.

10. The method of any one of the preceding claims, wherein the steps of calculating, mapping and incorporating are identical for creating the target hash array and for creating the origin hash array.

11. A method comprising:

creating an origin database by repeatedly adding rows to the origin database, each row comprising an attribute indicative of a physical location at which that row is stored on a non-volatile storage medium;

reading the rows from the origin database in an order of the attribute indicative of the physical location; and

writing the rows from the origin database to a target database to migrate the origin database to the target database.

12. The method of claim 11, wherein the attribute is a number that is incremented each time a row is added to the origin database.

13. The method of claim 11 or 12, wherein each row added to the origin database comprises a primary key, different from the attribute indicative of the physical location, the primary key being unique to the origin database and the target database.

14. A method comprising:

analysing a schema of an origin database to determine, for multiple tables in the origin database, an order of dependency in which to migrate the multiple tables to a target database;

generating references in the origin database to cross-reference rows in different tables independent from a primary key of the different tables;

copying the multiple tables in the order of dependency and including the references in the target database; and

using the references in the target database to replace primary key values in the target database and assigned by the origin database with primary key values assigned by the target database.

15. The method of claim 14, wherein copying the multiple tables comprises copying rows of each of the multiple tables according to an order as the rows are stored on a non-volatile storage medium.

16. The method of claim 14 or 15, wherein analysing the schema comprises identifying dependencies and wherein the order comprises positioning a first table of the multiple tables before a second table of the multiple tables if the second table depends on the first table.

17. A method comprising:

monitoring access to the origin database over a period of time to capture usage data;

determining, based on the usage data, one or more time slots where usage of the database is at a minimum;

determining a data migration period based on the one or more time slots;

inhibiting updates to the origin database during the data migration period to prevent changes to the origin database during migration;

copying contents of the origin database to the target database;

verifying contents on the target database copied from the origin database against the origin database; and

upon successfully verifying the contents, permitting updates to the origin database.

18. The method of claim 17, wherein monitoring the access comprises capturing time stamps of operations performed on the origin database, and wherein the method further comprises aggregating the usage data for multiple pre-defined time slots.

19. A system comprising:

one or more processors; and

memory comprising computer executable instructions, which when executed by the one or more processors, cause the system to perform the method of any one of claims 1 to 18.

20. A non-transitory machine-readable storage medium including instructions that, when executed by a machine, cause the machine to perform the method of any one of claims 1 to 18.