1. **Introduce Realm**

Realm Database stores data in **realms**: collections of heterogeneous realm objects. You can think of **each realm as a database. Each object in a realm is equivalent to a row in a SQL database table** or a MongoDB document. Unlike SQL, **realms do not separate different object types into individual tables.**

Realm Database stores **objects** **as groups of property values**. We call this column-**based storage**. This means that queries or writes for individual objects can be **slower than row-based storage equivalents when unindexed**, but querying a single field across multiple objects or fetching multiple objects can be much faster due to spatial locality and in-CPU vector operations.

**Realm Files**: Realm Database persists data in files saved on device storage. The database uses several kinds of file:

* **realm files,** suffixed with "realm", e.g. default.realm: contain **object data**.
* **lock files,** suffixed with "lock", e.g. default.realm.lock: keep track of which versions of data in a realm are actively in use. This prevents realm from reclaiming storage space that is still used by a client application.
* **note files,** suffixed with "note", e.g. default.realm.note: enable inter-thread and inter-process notifications.
* **management files,** suffixed with "management", e.g. default.realm.management: internal state management.

**Realm files contain object data with the following data structures**: Groups, Tables, Cluster Trees, and Clusters. Realm Database organizes these data structures into a tree structure with the following form:

* The top level, known as a Group, stores object metadata, a transaction log, and a collection of Tables.
* **Each class in the realm schema corresponds to a Table within the top-level Group.**
* Each Table contains a Cluster Tree, an implementation of a B+ tree.
* **Leaves on the Cluster Tree are called Clusters. Each contains a range of objects sorted by key value.**
* Clusters **store objects as collections of columns**.
* Each column contains data for a single property for multiple instances of a given object. Columns are arrays of data with uniformly sized values.
* Columns store data in one of the following sizes: 1, 2, 4, 8, 16, 32, or 64 bits. Each column uses one value size, determined by the largest value.

**Since pointers refer to memory addresses, objects written to persistent files cannot store references as pointers**. Instead, realm files refer to **data using the offset from the beginning of the file**. We call this a **ref**. **As Realm Database uses memory mapping to read and write data**, database operations translate these refs from offsets to memory pointers when navigating database structures.

Kĩ thuật **copy-on-write:**

Realm Database uses a technique called copy-on-write, which copies data to a new location on disk for every write operation instead of overwriting older data on disk. Once the new copy of data is fully written, the database updates existing references to that data. Older data is only garbage collected when **it is no longer referenced or actively in use by a client application.**

Because of copy-on-write, **older copies of data remain valid, since all of the references in those copies still point to other valid data. Realm Database leverages(đòn bẩy) this fact to offer multiple versions of data simultaneously to different threads in client applications**.