7.3 Interne Ebene:Indexsequentielle Dateiorganisation

- allgemein: mind. zweistufiger Baum, wobei Blattebene Hauptdatei mit eigentl. Datensätzen enthält, jede andere Baumstufe Indexdateien

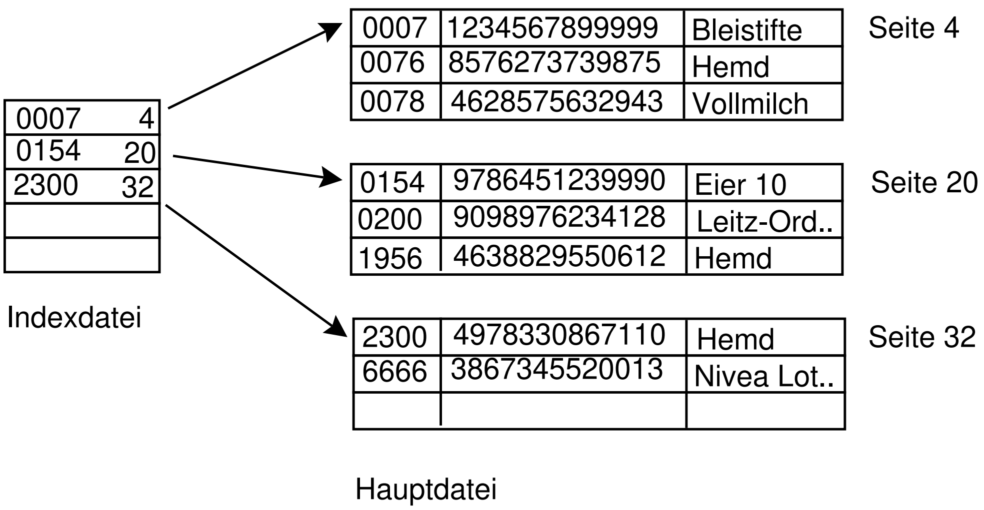
- einstufiger Index: 1 Hauptdatei + 1 Indexdatei

- n-stufiger Index: 1 Hauptdatei + n Indexdateien

- Aufbau Hauptdatei: Datensätze sequentiell nach Primärschlüssel sortiert

- Aufbau Indexdatei: Datensätze der Form (Primärschlüssel, Seitennummer),
  zu jeder Seite in Hauptdatei ex. nur genau ein Datensatz in Indexdatei (enthält Primärschl.wert des 1. Satzes der Seite) → Name *sparse index* (dünner Index),
  Datensätze in Indexdatei bzgl. Primärschl.wert sortiert

- Beispiel: Artikel-Relation, Artnr als Primärschl.


0007 4
0154 20
2300 32
Indexdatei
0007 1234567899999 Bleistifte
0076 8576273739875 Hemd
0078 4628575632943 Vollmilch
Seite 4
0154 9786451239990 Eier 10
0200 9098976234128 Leitz-Ord..
1956 4638829550612 Hemd
Seite 20
2300 4978330867110 Hemd
6666 3867345520013 Nivea Lot..
Seite 32
Hauptdatei

Indexsequentielle Dateiorganisation

- Lookup: gesucht Datensatz zum Primärschlüsselwert w
  Indexdatei sequentiell durchlaufen bis Satz (v1, s) gefunden mit v1 $\leq$ w und
  (v1, s) ist letzter Satz der Indexdatei, oder
  der nächste Satz (v2, s') im Index hat v2 $>$ w
  dann sequentielle Suche in Seite s der Hauptdatei

- Einfügen: Lookup anwenden,
  falls möglich, Satz sortiert in gefundener Seite speichern
  sonst: neue Seite holen (von Freiseitenverwaltung), Sätze der 'zu vollen Seite' gleichmäßig auf alte und neue Seite verteilen, Index anpassen

- Löschen: Lookup anwenden,
  Satz in gefundener Seite löschen; falls erster Satz Index anpassen, nachfolgende Sätze nach vorne schieben (Vorsicht bei pinned records)
  falls Seite nach Löschen leer: Index anpassen, Seite an Freiseitenverwaltung

- Suchverfahren:
  - lineares Suchen in Indexdatei, lineares Suchen in Seite der Hauptdatei
  - binäres Suchen der richtigen Indexseite, lineares Suchen in gefundener Indexseite, lineares Suchen in Seite der Hauptdatei