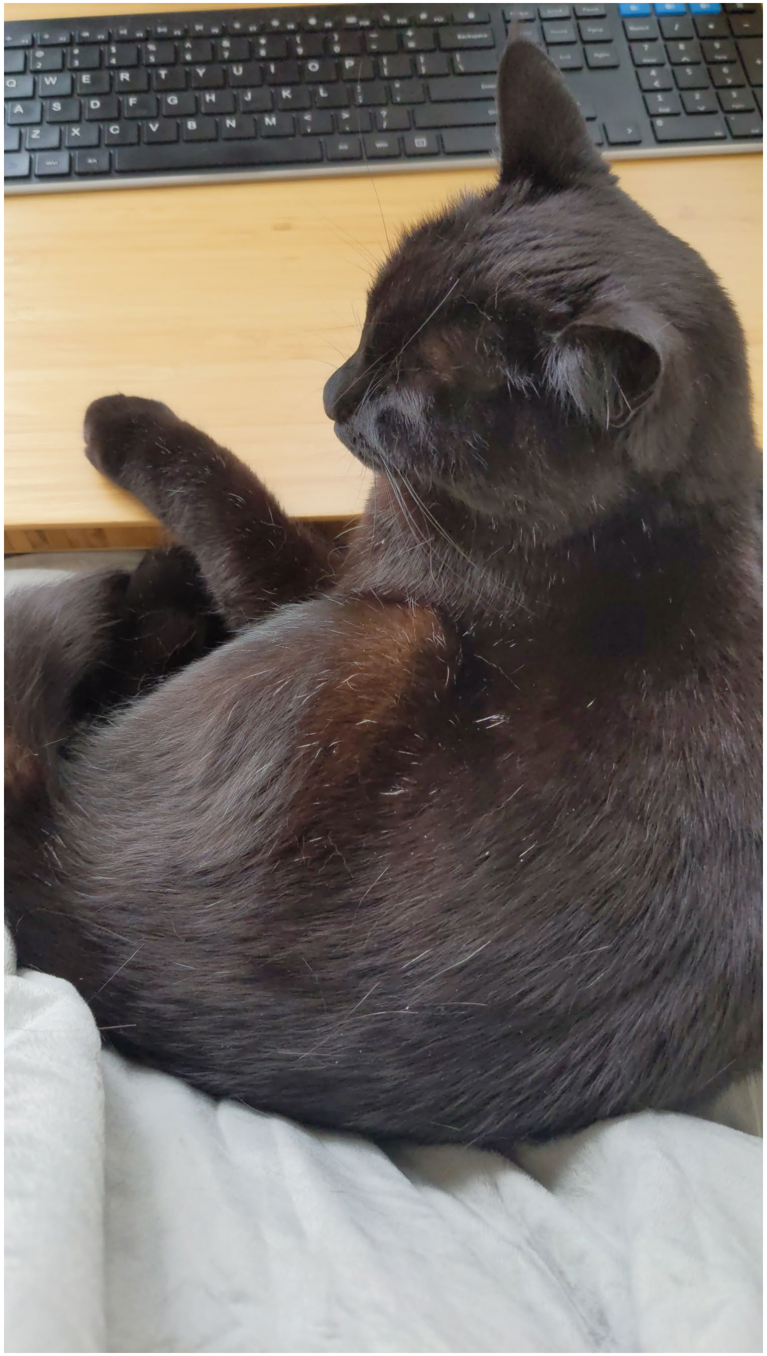


Storage

ERIN KEITH

Goals

1. Review General Storage Methods
2. Introduce Database Storage Methods

Memory

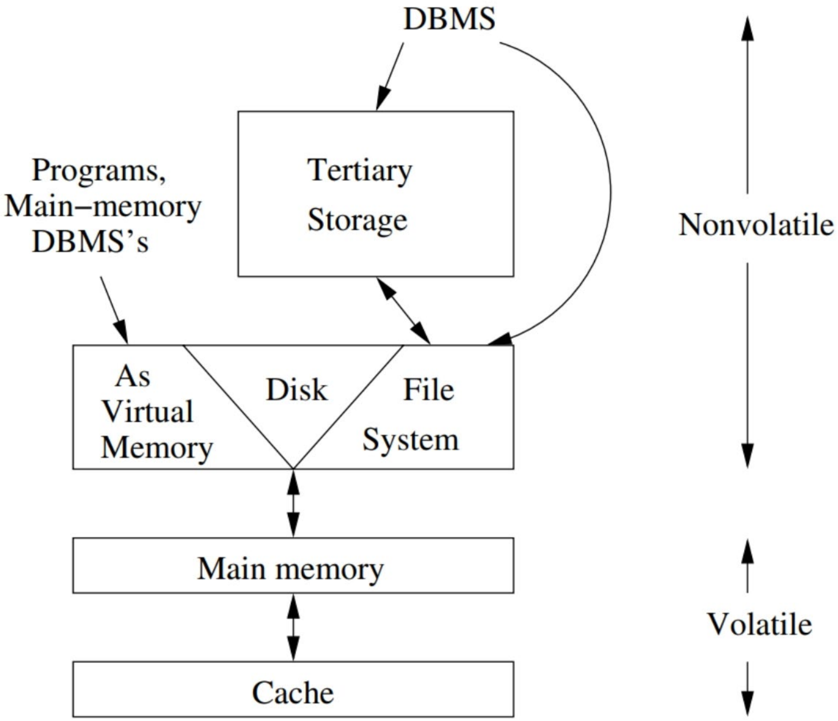


Figure 13.1: The memory hierarchy

Database systems always involve secondary storage

- Considerations
  - retrieval times
  - volatility

The Database Environment

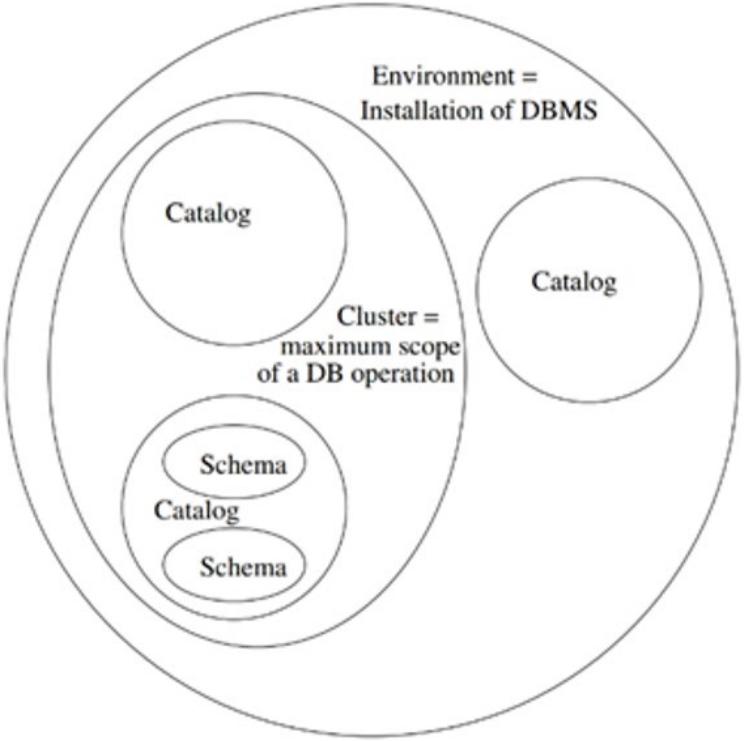


Figure 9.2: Organization of database elements within the environment

Vocabulary Review

record

- represents a tuple or object

tuple

- a row with linked data about a certain entity (it can be any object)

row

- an entry in a table

Database Storage

record

- storage element for an entry in a table
- consecutive bytes in some disk block

Database Storage

record

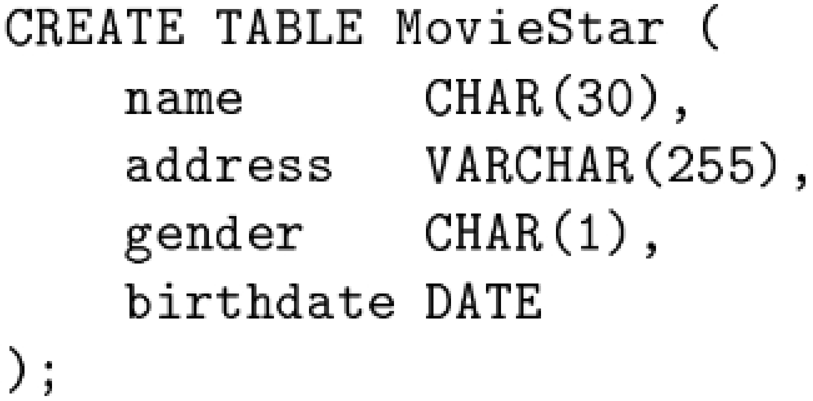
- The simplest sort of record consists of fixed-length fields, one for each attribute of the represented tuple.
  - ***field***
    - column, property, attribute
    - size becomes important when we're creating tables

Database Storage

header

- Often, the record begins with a header, a fixed-length region where information about the record itself is kept.
  - a pointer to the schema
  - length of the record (for skipping over)
  - timestamps (for last modified/read)
  - pointers to the fields (instead of the schema)

Example

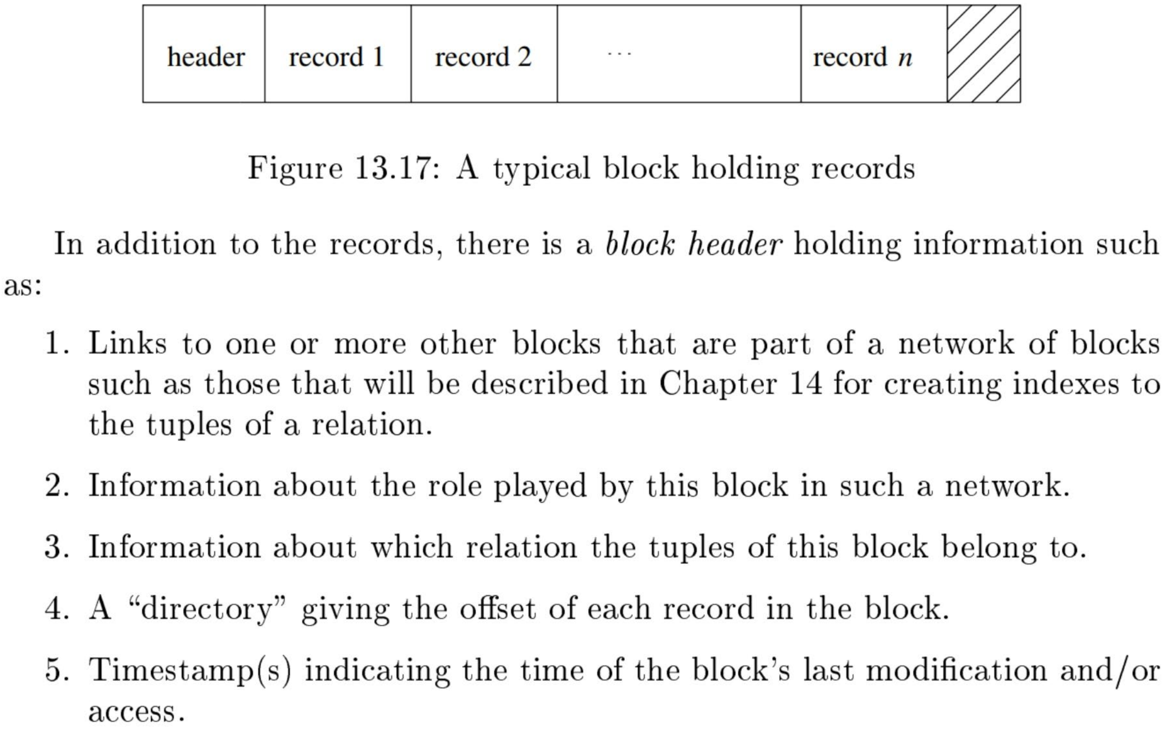


```
CREATE TABLE MovieStar (
    name      CHAR(30),
    address   VARCHAR(255),
    gender    CHAR(1),
    birthdate DATE
);
```

Database Storage

blocks

- Records representing tuples of a relation are stored in blocks of the disk and moved into main memory (along with their entire block) when we need to access or update them.



header	record 1	record 2	···	record *n*	▨

Figure 13.17: A typical block holding records

In addition to the records, there is a *block header* holding information such as:

1. Links to one or more other blocks that are part of a network of blocks such as those that will be described in Chapter 14 for creating indexes to the tuples of a relation.

2. Information about the role played by this block in such a network.

3. Information about which relation the tuples of this block belong to.

4. A "directory" giving the offset of each record in the block.

5. Timestamp(s) indicating the time of the block's last modification and/or access.

Client-Server Systems

Commonly, a database system consists of a server process that provides data from secondary storage to one or more client processes that are applications using the data.

- The server and client processes may be on one machine, or the server and the various clients can be distributed over many machines.
- There is a client virtual memory to server physical memory mapping, but it's not too important.

BLOBs

Binary Large OBjects

- GIF
- JPEG
- MPEG
- stored on a sequence of blocks (hopefully sequential)
- client may receive pieces

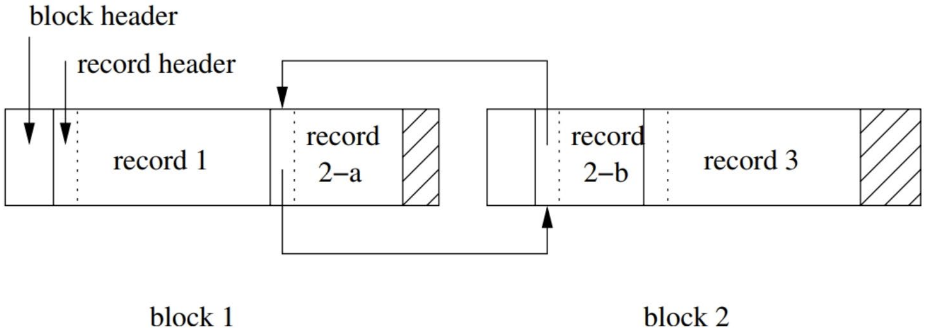


Figure 13.27: Storing spanned records across blocks

Operations

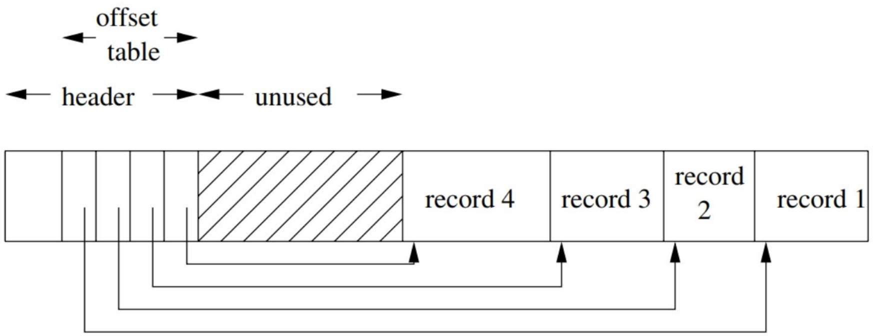


Figure 13.28: An offset table lets us slide records within a block to make room for new records

Insertion

- adding one or more records to a table
- keeping records ordered can cause complications

Operations

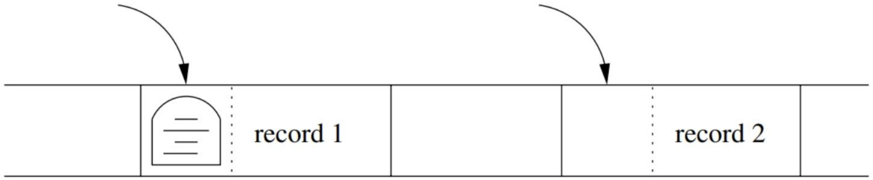


Figure 13.30: Record 1 can be replaced, but the tombstone remains; record 2 has no tombstone and can be seen when we follow a pointer to it

Deletion

- removing one or more records from a table
- can reclaim or “slide” around
- can replace with a “tombstone”

Operations

Update

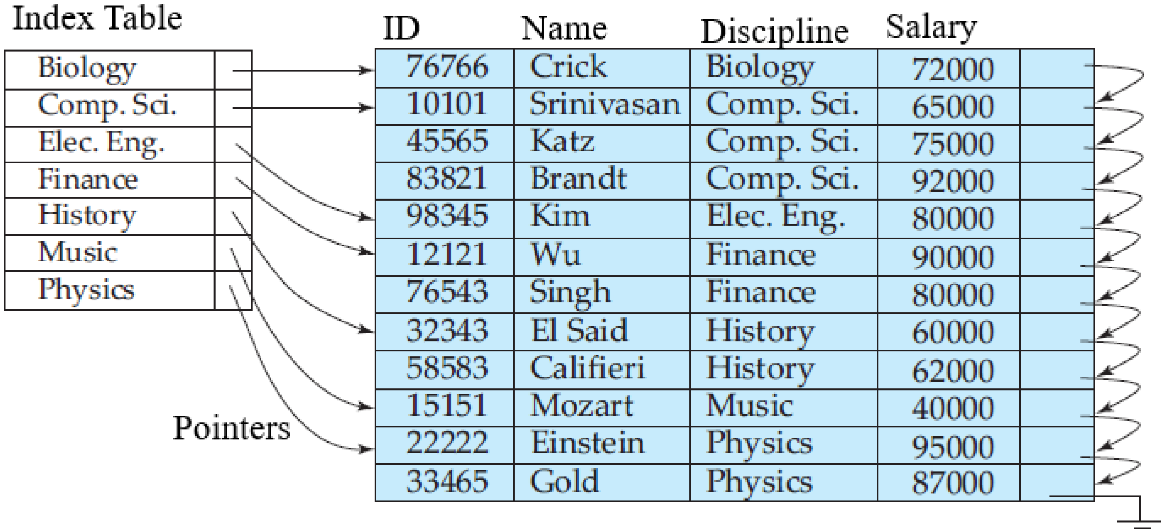
- changing one or more records in a table
- same problems if it is a variable length record

Retrieval

- finding one or more records in a table

Indexes

- used behind the scenes to connect tables
- optimizes data retrieval
- decreases performance of database modifications



Index Table

Biology
Comp. Sci.
Elec. Eng.
Finance
History
Music
Physics

Pointers

ID	Name	Discipline	Salary
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
33465	Gold	Physics	87000

Indexes

When relations are very large, it becomes expensive to scan all the tuples of a relation to find those (perhaps very few) tuples that match a given condition.

- An index on an attribute **`A`** of a relation is a data structure that makes it efficient to find those tuples that have a fixed value for attribute **`A`**.

Next Class

Module:

Week 2: Background, Ch 2

Topic:

The Relational Model