Amr Elhelw's

TECH VAULT

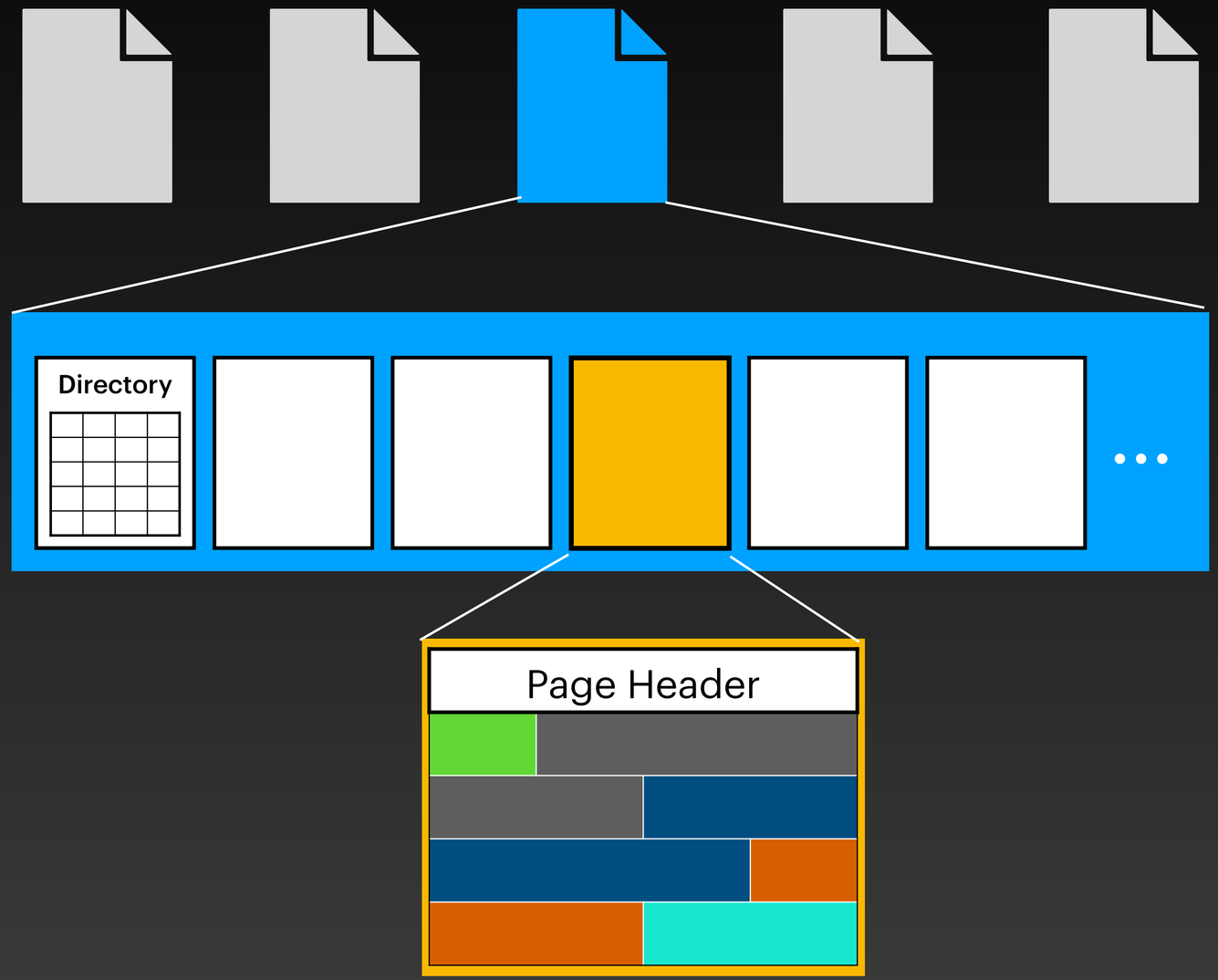
DATABASE STORAGE PART 2

Database Storage

Database Files

Pages

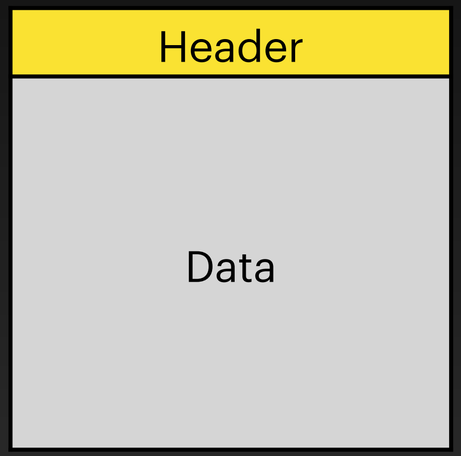
Tuples



Page Header

- Exists in every page

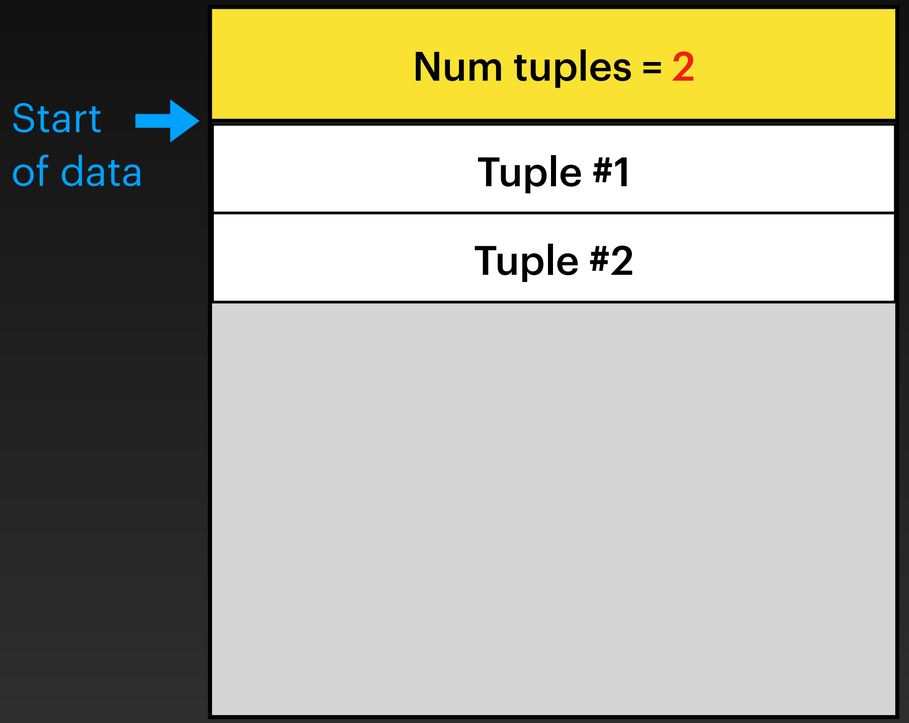
- Metadata
  - Page size
  - Compression/encoding info
  - DBMS version
  - ...



Naive Approach

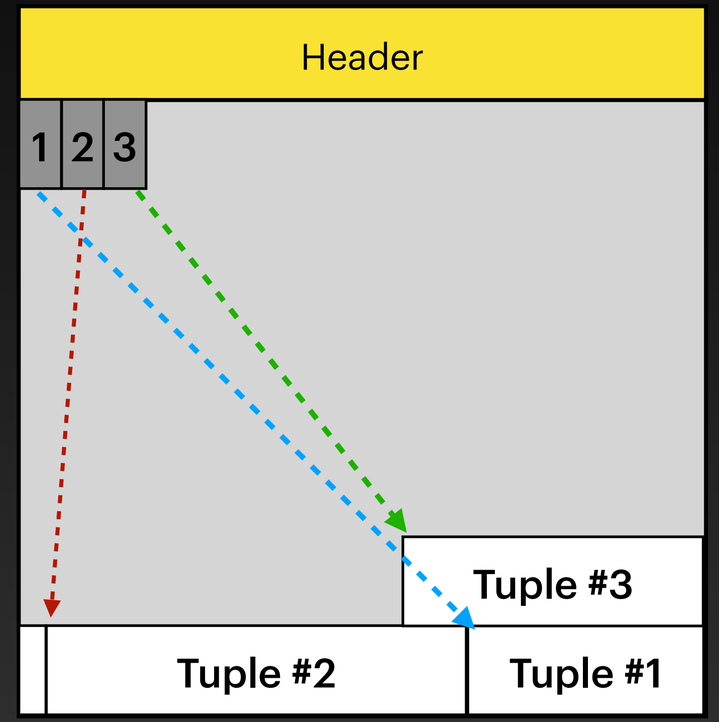
- Maintain # of tuples in page
- Append new tuples at the end
  - Use the # of tuples to determine “the end” (assuming equal-size tuples)

- **Challenges**
  - Deletion
  - Variable-sized tuples



Slotted Pages

- Header maintains:
  - # slots
  - Offset of last used slot
- Each slot points to corresponding tuple

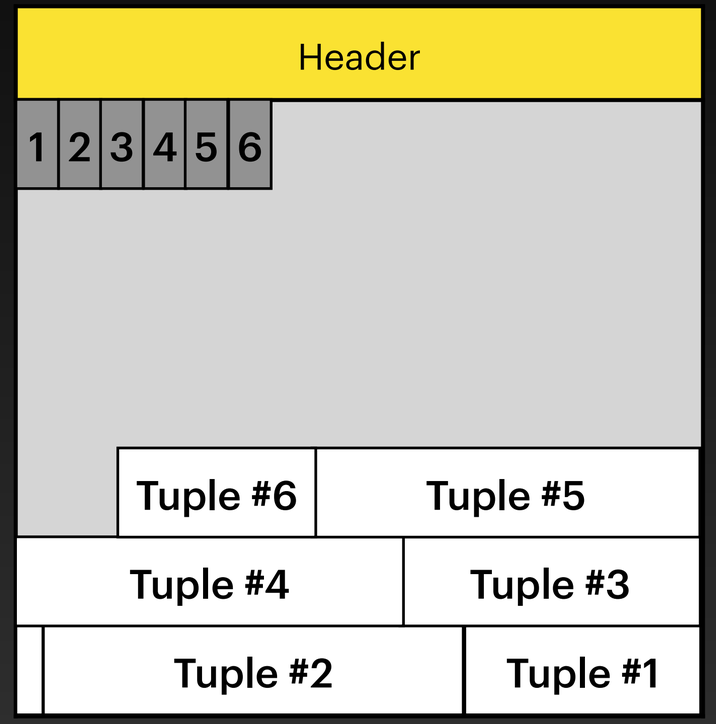


PostgreSQL

Microsoft® SQL Server®

Slotted Pages

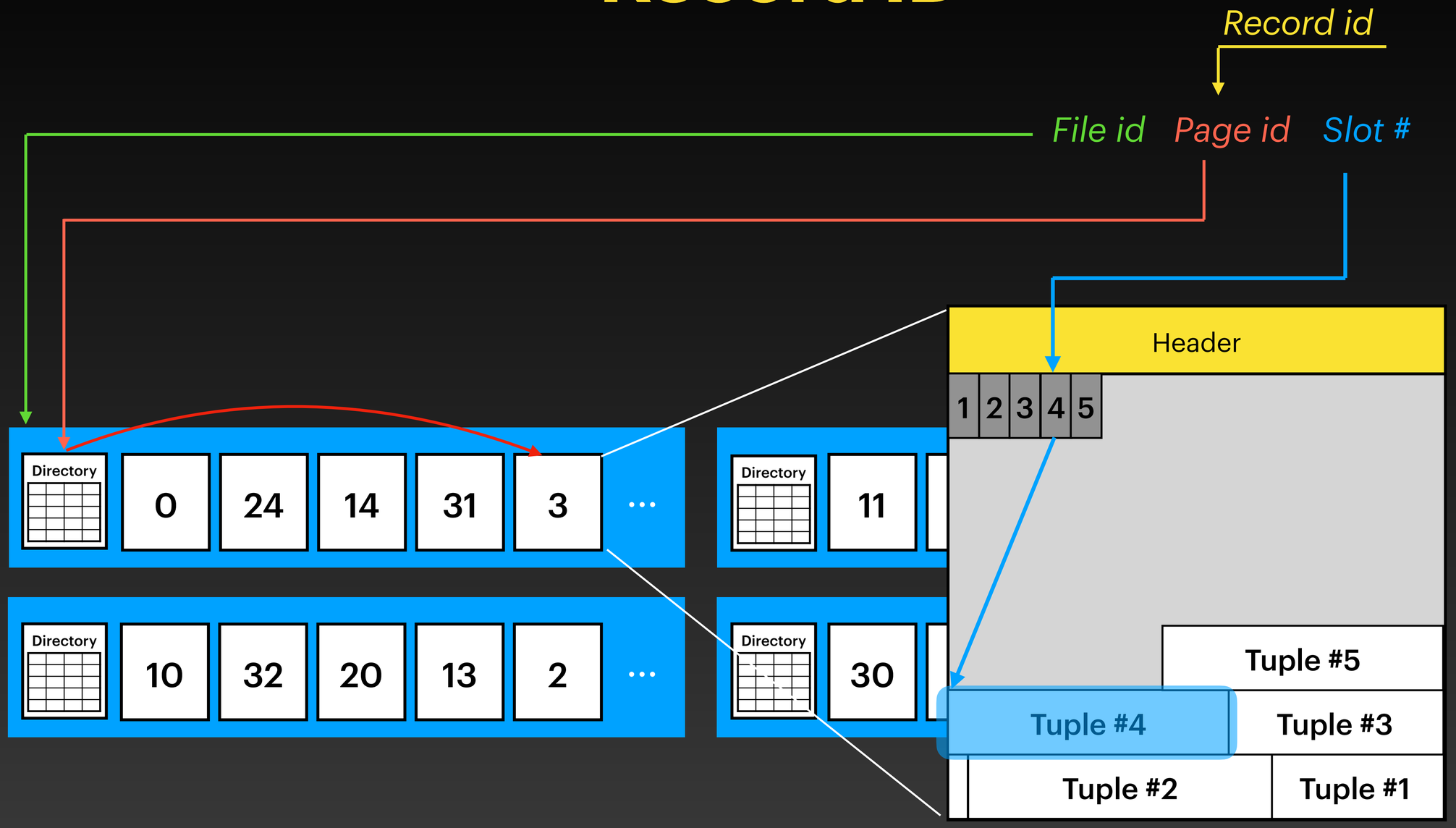
- Header maintains:
  - # slots
  - Offset of last used slot
- Each slot points to corresponding tuple
- Tuples are added starting from the end of the page



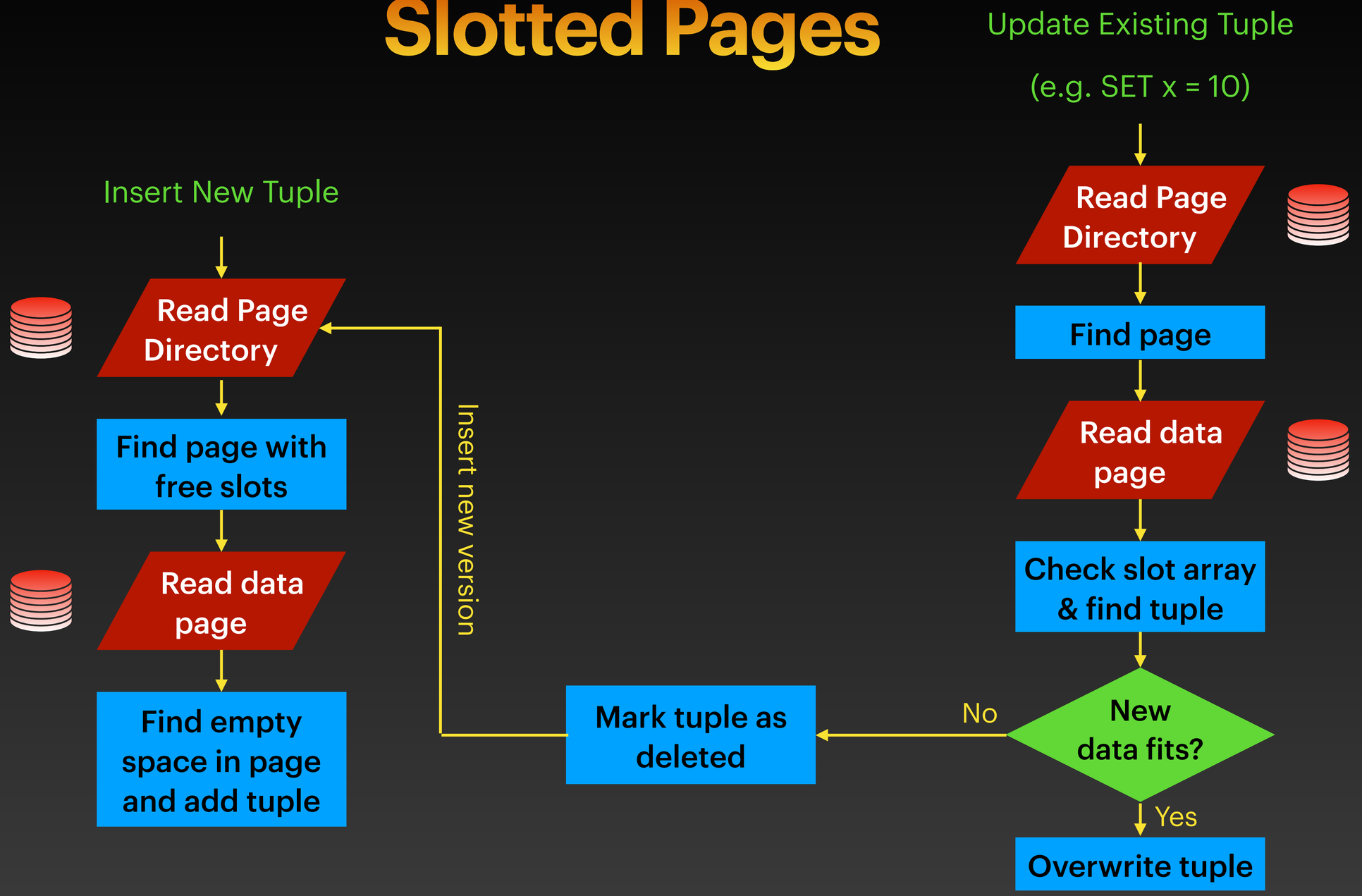
Record ID

- Unique for every tuple
- Represents physical location of tuple
  - Some encoding of (file id, page id, slot #)
  - Internal “index”
- Usually not stored with the tuple

Record ID



Slotted Pages



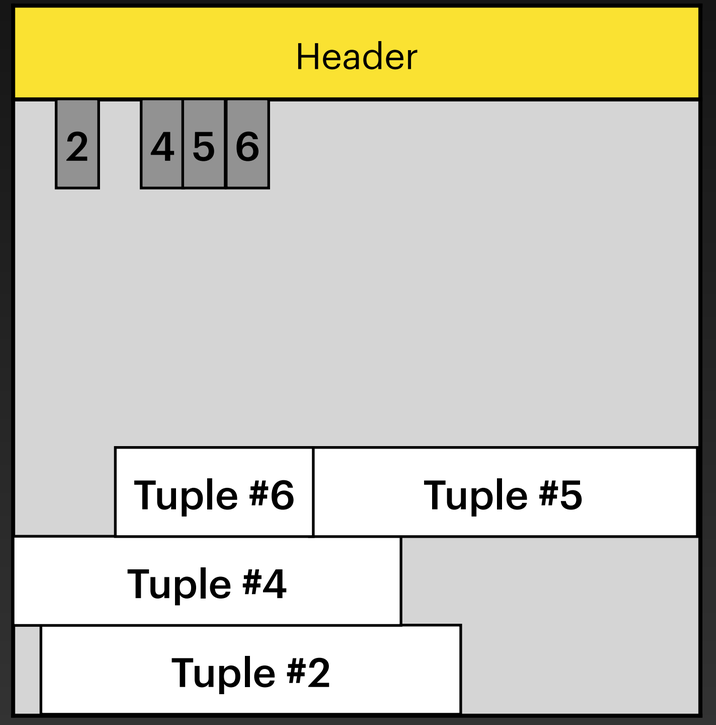
Amr Elhelw's TECH VAULT

Slotted Pages - Challenges

- **Challenge #1 - Too much Disk I/O**
  - 2 pages for an insert
  - 2-4 pages for an update
- **Challenge #2 - Random Disk I/O**
  - E.g. updating multiple tuples in separate pages

Slotted Pages - Challenges

- **Challenge #3 - Fragmentation**
  - Unused tuple and slot spaces



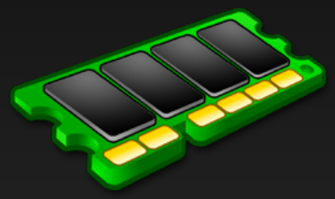
Log-Structured Storage

- Also known as Log-structure merge trees (LSM trees)
- Record “changes” rather than storing the tuples in pages
  - Each entry: SET or DELETE operation on a tuple
  - Must include the tuple id

Data Page (in memory)

```
SET #31 (val=10)
SET #32 (val=20)
DEL #30
SET #31 (val=15)
SET #33 (val=42)
SET #31 (val=25)
```

Log-Structured Storage

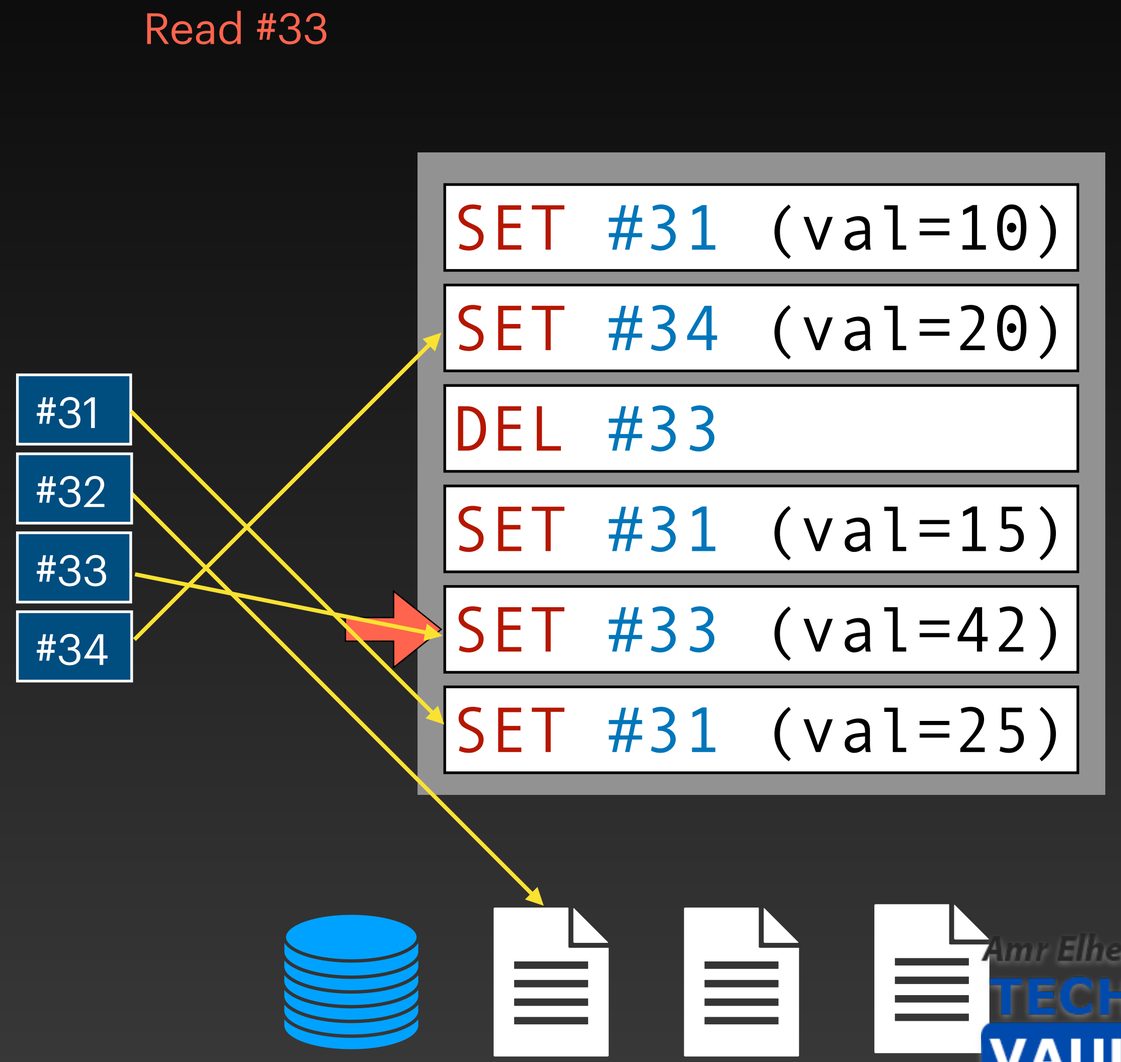


Memory

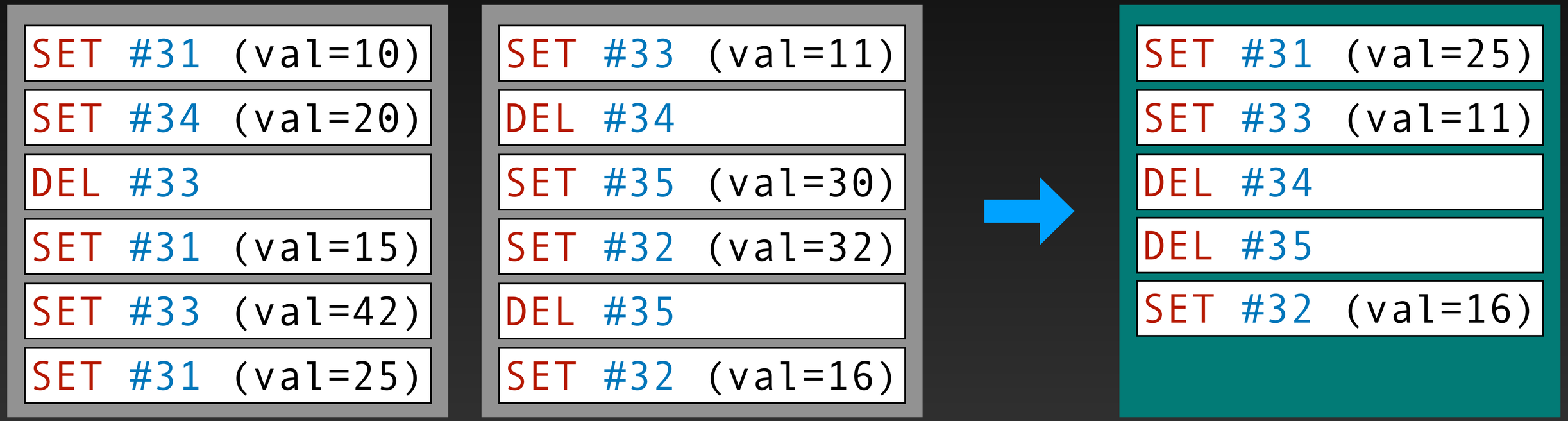
Disk

Log-Structured Storage

- Read tuple with given id
  - Find newest entry with that id
  - Scan log from end to beginning
- Scanning is inefficient
  - Use index



Compaction



Index-Organized Tables (IOT)

- Store the actual data inside an index (primary key)

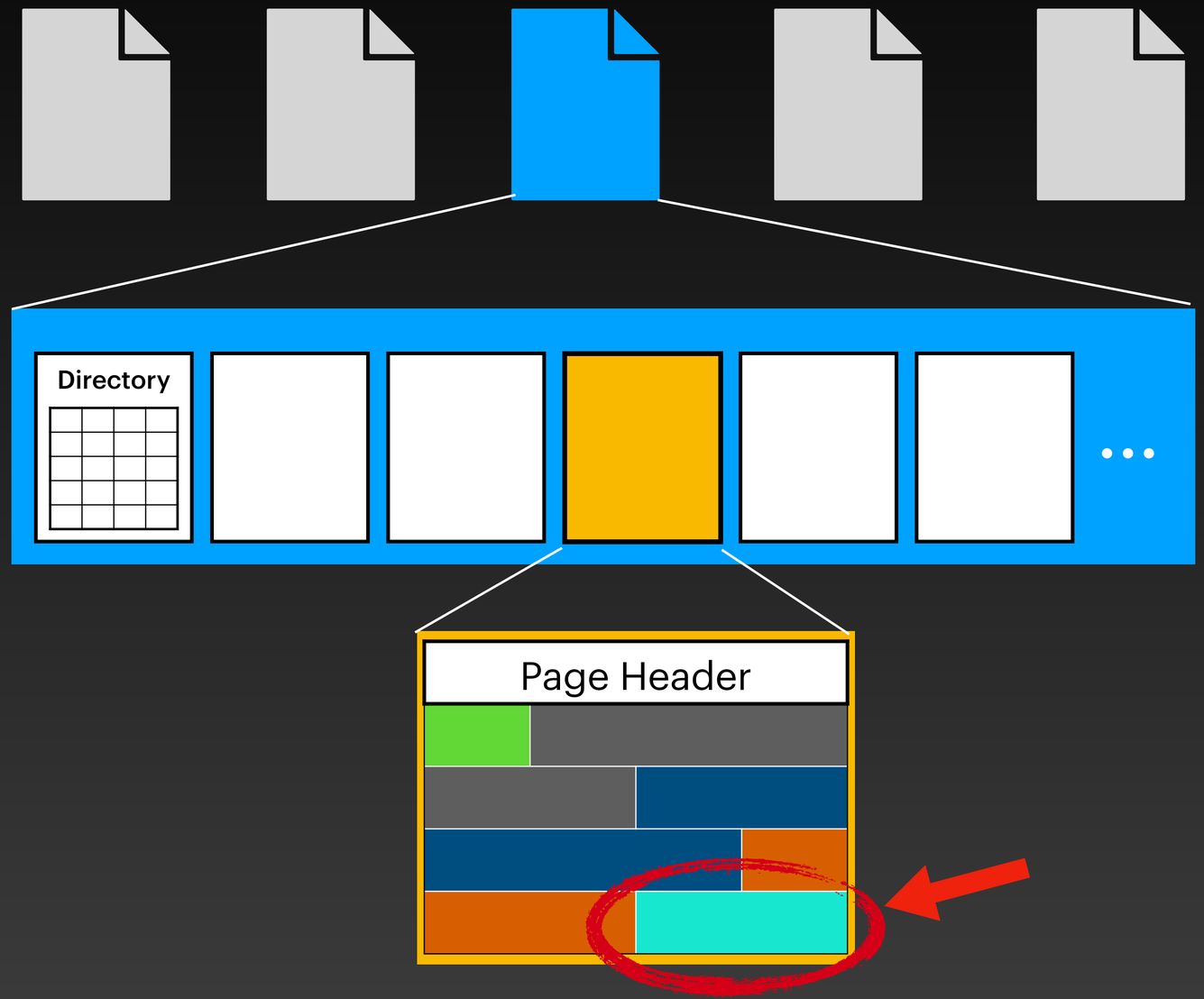
- Faster reads
- Slower writes
- Reduced storage

Database Storage

Database Files

Pages

Tuples

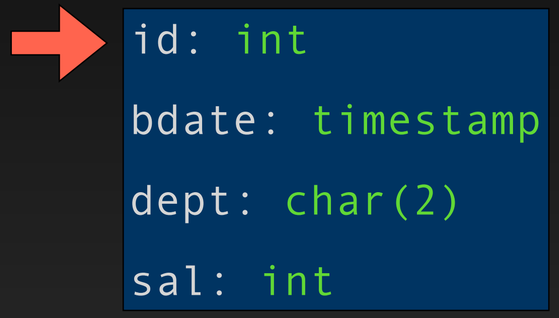


Tuple Storage

- A sequence of bytes
- Header + Data

- Interpreted by the DBMS using schema information

Tuple Storage



Tuple Storage

Word-alignment: Padding

`id: int`	*32 bits*
`bdate: timestamp`	*64 bits*
`dept: char(2)`	*16 bits*
`sal: int`	*32 bits*