CSC 350

# STORAGE AND FILE STRUCTURE

# Approximate Storage Latency

Read 1MB From RAM: 0.25ms

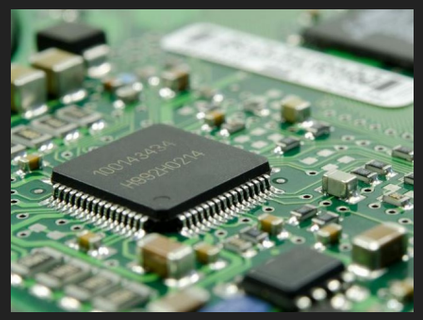
Read 1MB From Network - 10ms

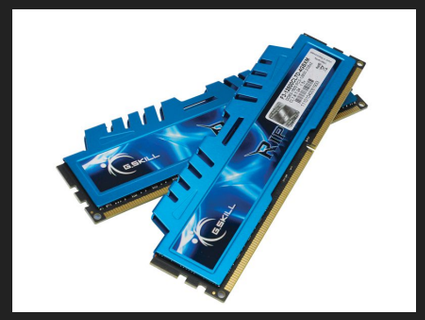
Read 1MB From Disk - 30ms

https://stackoverflow.com/questions/4087280/approximate-cost-to-access-various-caches-and-main-memory

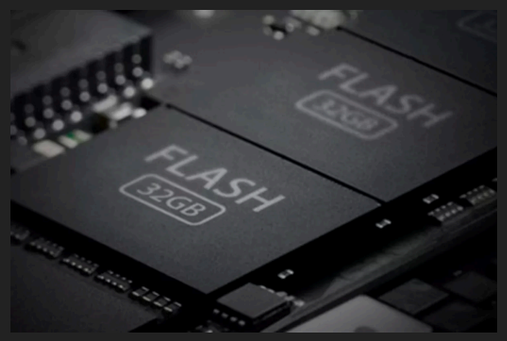
# STORAGE LEVELS

Primary Storage: Fast, but volatile (cache and RAM)

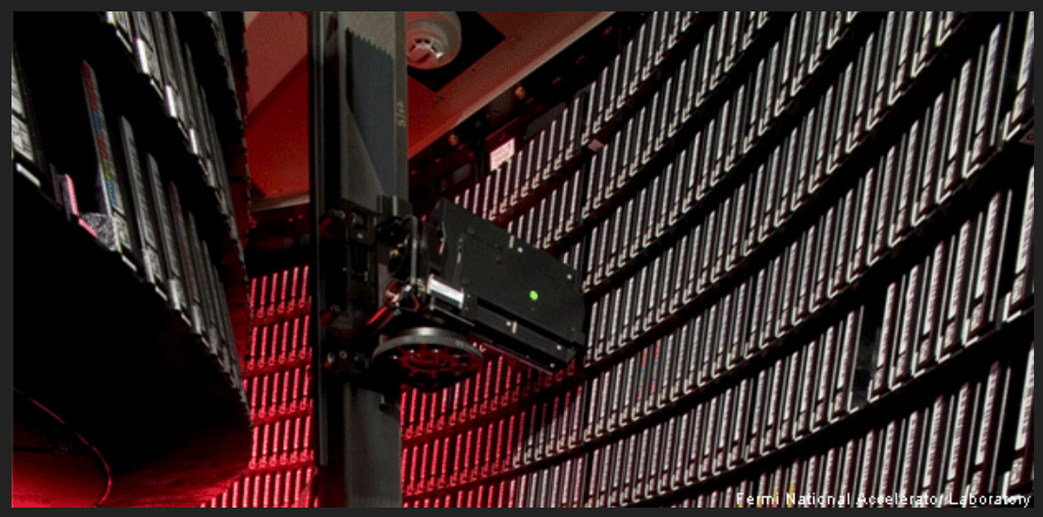




Secondary Storage: Moderately-Fast, non-volatile
(AKA “online storage”)




Tertiary Storage: Slow, non-volatile
(AKA “offline storage”)



## RECORD ORGANIZATION OPTIONS

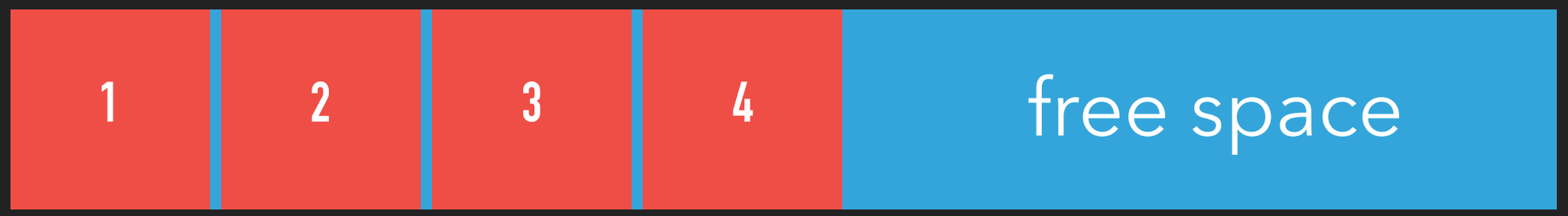
I. Sequential File, Fixed-Size Records

Assume records are fixed size.

Assume each file has the same type of record.

Different relation types are stored in different relations.

Don’t allow records to cross block boundaries.



# I. Sequential File, Fixed-Size Records

| | | | | |
|---|---|---|---|---|
| record 0 | 10101 | Srinivasan | Comp. Sci. | 65000 |
| record 1 | 12121 | Wu | Finance | 90000 |
| record 2 | 15151 | Mozart | Music | 40000 |
| record 3 | 22222 | Einstein | Physics | 95000 |
| record 4 | 32343 | El Said | History | 60000 |
| record 5 | 33456 | Gold | Physics | 87000 |
| record 6 | 45565 | Katz | Comp. Sci. | 75000 |
| record 7 | 58583 | Califieri | History | 62000 |
| record 8 | 76543 | Singh | Finance | 80000 |
| record 9 | 76766 | Crick | Biology | 72000 |
| record 10 | 83821 | Brandt | Comp. Sci. | 92000 |
| record 11 | 98345 | Kim | Elec. Eng. | 80000 |

How to handle record deletions?

# I. Sequential File, Fixed-Size Records

## Option 1: Delete and Compact

| | | | | |
|---|---|---|---|---|
| record 0 | 10101 | Srinivasan | Comp. Sci. | 65000 |
| record 1 | 12121 | Wu | Finance | 90000 |
| record 2 | 15151 | Mozart | Music | 40000 |
| record 4 | 32343 | El Said | History | 60000 |
| record 5 | 33456 | Gold | Physics | 87000 |
| record 6 | 45565 | Katz | Comp. Sci. | 75000 |
| record 7 | 58583 | Califieri | History | 62000 |
| record 8 | 76543 | Singh | Finance | 80000 |
| record 9 | 76766 | Crick | Biology | 72000 |
| record 10 | 83821 | Brandt | Comp. Sci. | 92000 |
| record 11 | 98345 | Kim | Elec. Eng. | 80000 |

# I. Sequential File, Fixed-Size Records

## Option 2: Delete and Move

| | | | | |
|---|---|---|---|---|
| record 0 | 10101 | Srinivasan | Comp. Sci. | 65000 |
| record 1 | 12121 | Wu | Finance | 90000 |
| record 2 | 15151 | Mozart | Music | 40000 |
| record 4 | 32343 | El Said | History | 60000 |
| record 5 | 33456 | Gold | Physics | 87000 |
| record 6 | 45565 | Katz | Comp. Sci. | 75000 |
| record 7 | 58583 | Califieri | History | 62000 |
| record 8 | 76543 | Singh | Finance | 80000 |
| record 9 | 76766 | Crick | Biology | 72000 |
| record 10 | 83821 | Brandt | Comp. Sci. | 92000 |
| record 11 | 98345 | Kim | Elec. Eng. | 80000 |

Which approach is best?

# I. Sequential File, Fixed-Size Records

## Option 3: Free Lists

store location of first deleted record in header

Each deleted record stores the location of the next deleted record

| | | | | |
|---|---|---|---|---|
| header | | | | |
| record 0 | 10101 | Srinivasan | Comp. Sci. | 65000 |
| record 1 | | | | |
| record 2 | 15151 | Mozart | Music | 40000 |
| record 3 | 22222 | Einstein | Physics | 95000 |
| record 4 | | | | |
| record 5 | 33456 | Gold | Physics | 87000 |
| record 6 | | | | |
| record 7 | 58583 | Califieri | History | 62000 |
| record 8 | 76543 | Singh | Finance | 80000 |
| record 9 | 76766 | Crick | Biology | 72000 |
| record 10 | 83821 | Brandt | Comp. Sci. | 92000 |
| record 11 | 98345 | Kim | Elec. Eng. | 80000 |

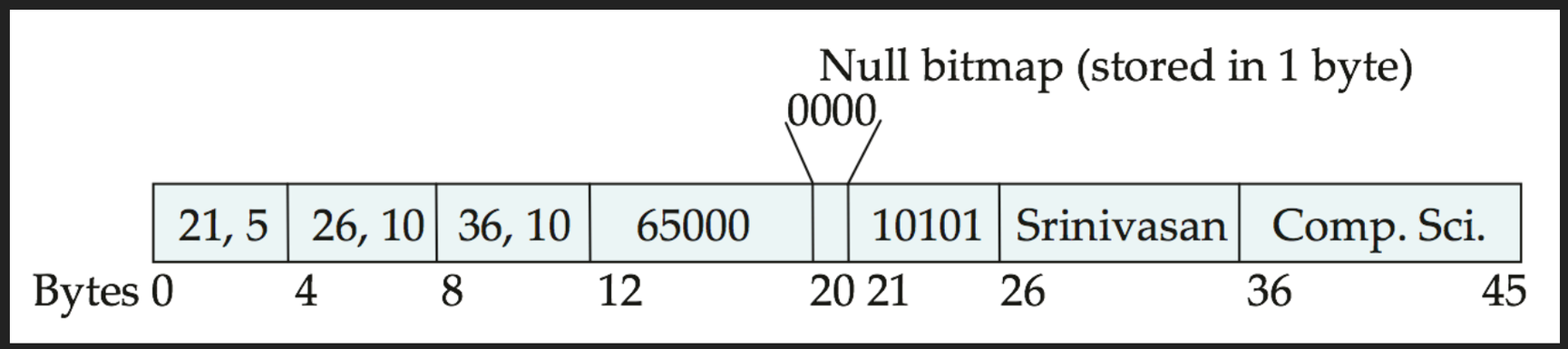
Subsequent inserts will fill these gaps.

## RECORD ORGANIZATION OPTIONS

II Sequential File, Variable-Size Records

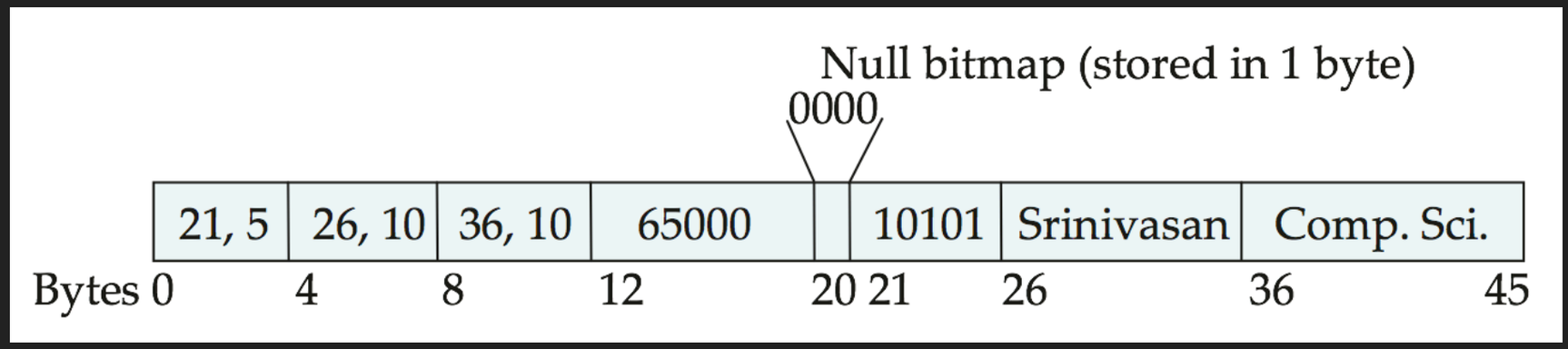
- Attributes stored in order.
- Variable length attributes are represented by an {offset, length} marker.
- Variable length attributes are represented by an {offset, length} marker.

Null bitmap field indicates which fields are empty, which saves unnecessary data lookups.

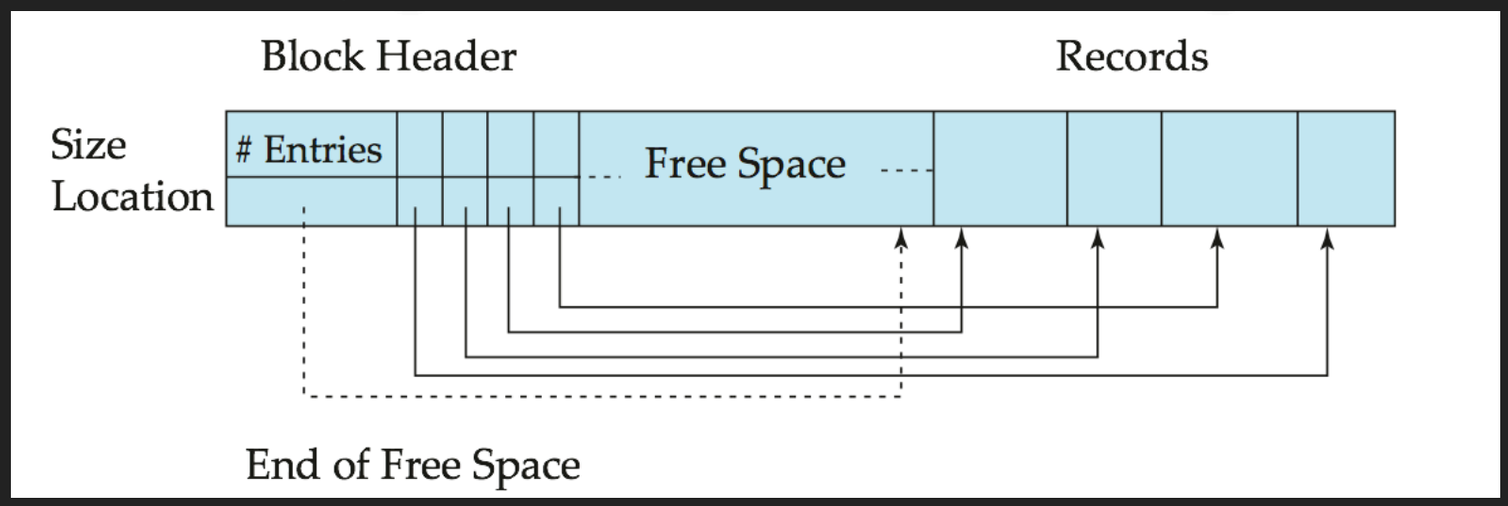


# II Sequential File, Variable-Size Records

Null bitmap field indicates which fields are empty, which saves unnecessary data lookups.




Records stored using "Slotted Page Structure"



## RECORD STORAGE OPTIONS

I. Heap: records are stored anywhere in the file where there is space.

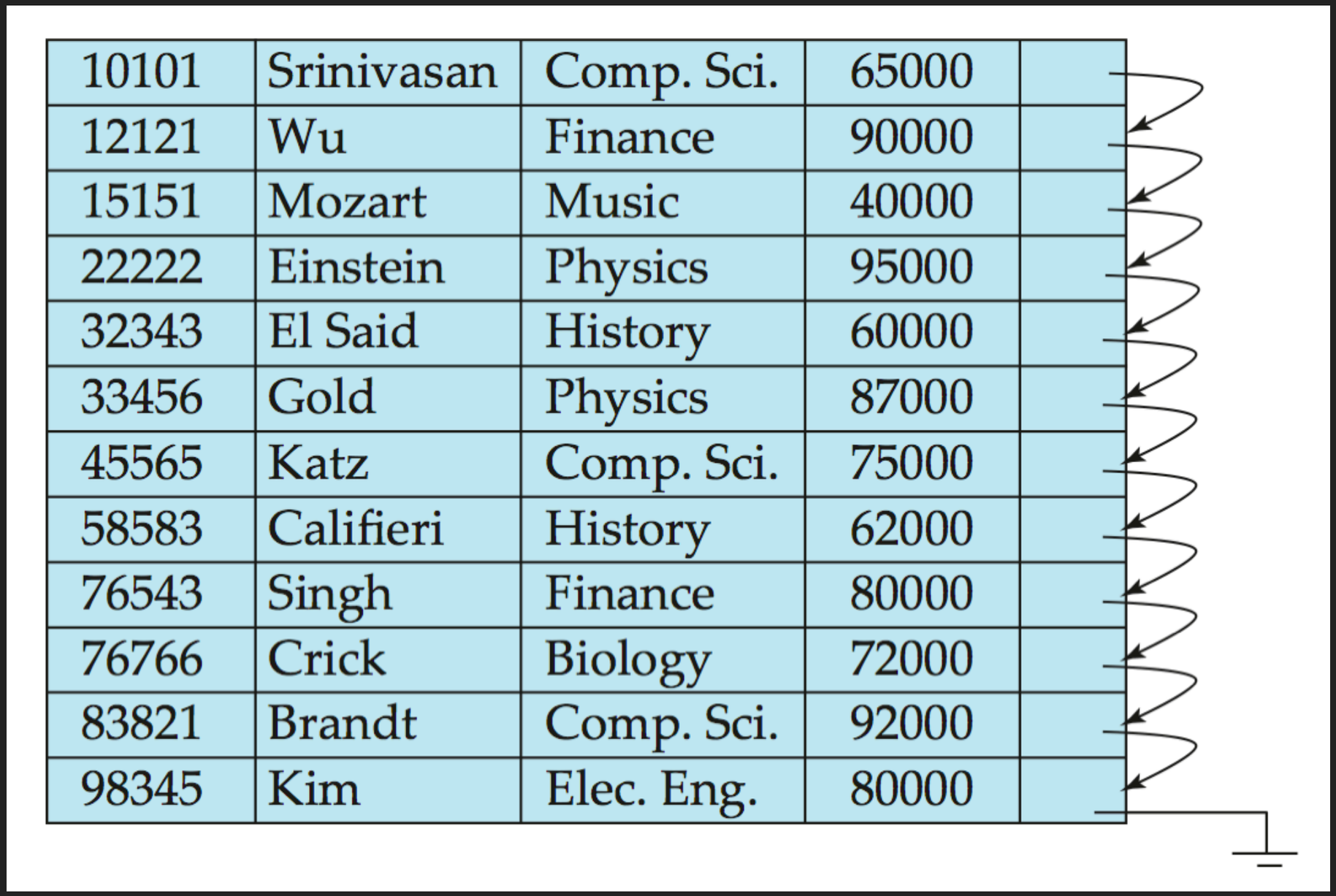
II. Sequential: store records in sequential order, based on a key

III. Hashing: Hash some attribute of each record and use that to determine the storage order.

IV. Multitable Clustering: Store related records in the same block, even if they belong to different relations.

## RECORD STORAGE OPTIONS

I. Heap: records are stored anywhere in the file where there is space.



| | | | | |
|---|---|---|---|---|
| 10101 | Srinivasan | Comp. Sci. | 65000 | |
| 12121 | Wu | Finance | 90000 | |
| 15151 | Mozart | Music | 40000 | |
| 22222 | Einstein | Physics | 95000 | |
| 32343 | El Said | History | 60000 | |
| 33456 | Gold | Physics | 87000 | |
| 45565 | Katz | Comp. Sci. | 75000 | |
| 58583 | Califieri | History | 62000 | |
| 76543 | Singh | Finance | 80000 | |
| 76766 | Crick | Biology | 72000 | |
| 83821 | Brandt | Comp. Sci. | 92000 | |
| 98345 | Kim | Elec. Eng. | 80000 | |

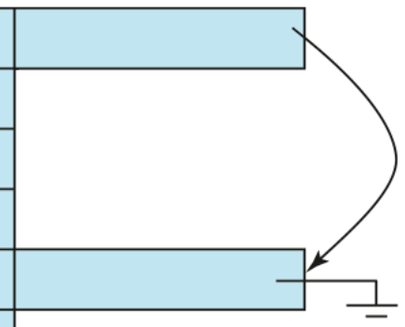
IV. Multitable Clustering: Store related records in the same block, even if they belong to different relations.

| *dept_name* | *building* | *budget* |
|---|---|---|
| Comp. Sci. | Taylor | 100000 |
| Physics | Watson | 70000 |

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 33456 | Gold | Physics | 87000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 83821 | Brandt | Comp. Sci. | 92000 |

| Comp. Sci. | Taylor | 100000 |
|---|---|---|
| 45564 | Katz | 75000 |
| 10101 | Srinivasan | 65000 |
| 83821 | Brandt | 92000 |
| Physics | Watson | 70000 |
| 33456 | Gold | 87000 |

| Comp. Sci. | Taylor | 100000 | |
|---|---|---|---|
| 45564 | Katz | 75000 | |
| 10101 | Srinivasan | 65000 | |
| 83821 | Brandt | 92000 | |
| Physics | Watson | 70000 | |
| 33456 | Gold | 87000 | |



## DATA DICTIONARY

The Data Dictionary (AKA The System Catalog) stores metadata (data about data).

- Relation data: names of relations; names, types, and lengths of attributes, constraints, etc...
- User and permission data
- Statistical data
- Physical file information: how and where relations are stored.
- Data about indices