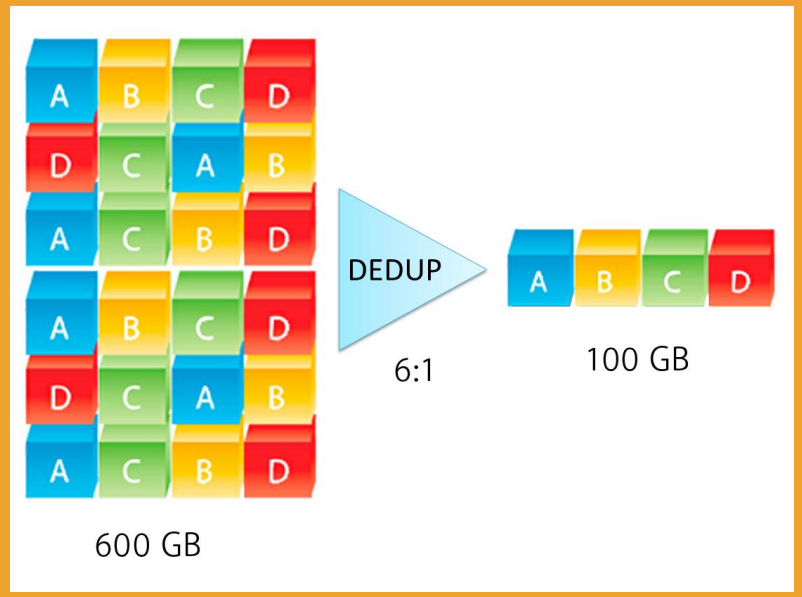


Data Manipulation

Finding and Removing Duplicate Records

Data Redundancy & Duplicacy

- Storage administrators are struggling to handle spiraling volumes of documents, audio, video, images and large email attachments.
- Adding storage is not always the best solution
- Many companies are turning to data reduction technologies such as data deduplication

Data Duplicacy

- Entries that have been added by a system user multiple times
- for example, re-registering because you have forgotten your details.
- It is one of the problem which causes inconsistency in databases.

Data Redundancy

- Same of data is stored at multiple locations or tables.
- Data redundancy is costly to address as it requires
  - additional storage,
  - synchronization between databases
  - design work to align the information represented by different presentation of the same data.

Data Redundancy & Duplication

- Storing the information several times
- It leads to waste of storage space

Problems with Data Redundancy & Duplication

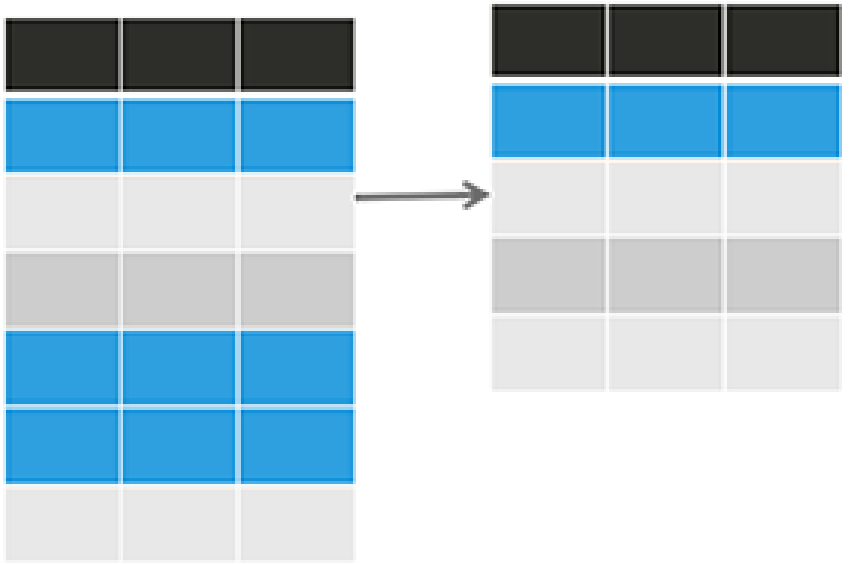
1. Wasted Storage Space.
2. More difficult Database Updates.
3. A Possibility of Inconsistent data.

Data Deduplication

- Data deduplication, also called intelligent compression or single-instance storage

Removing Duplicate Records

Removing Duplicate Data in R



- **duplicated()**: Find duplicate elements (R base)
- **unique()**: Keep only unique elements (R base)
- **dplyr::distinct()**: Keep only unique elements (more efficient than unique())

R base and dplyr

Removing Duplicate Records

- **unique()** - for extracting unique elements

- **duplicated()** - for identifying duplicated elements
  - returns a logical vector where TRUE specifies which elements of a vector or data frame are duplicates.

Find and drop duplicate elements: duplicated()

Example:

```
x <- c(1, 1, 4, 5, 4, 6)
```

Find the position of duplicate elements in x,

```
> duplicated(x)
```

O/P:

[1] FALSE TRUE FALSE FALSE TRUE FALSE

Find and drop duplicate elements: duplicated()

Example:

```
x <- c(1, 1, 4, 5, 4, 6)
```

Write a command to extract duplicate elements

Write a command to remove duplicate elements

Find and drop duplicate elements: duplicated()

Data Frame Example:

How can I remove duplicate rows from this example data frame?

```
A   1
A   1
A   2
B   4
B   1
B   1
C   2
C   2
```

I would like to remove the duplicates based on both the columns:

```
A   1
A   2
B   4
B   1
C   2
```

Order is not important.

Find and drop duplicate elements: duplicated()

Data Frame Example:

```
a <- c(rep("A", 3), rep("B", 3), rep("C",2))

b <- c(1,1,2,4,1,1,2,2)

df <-data.frame(a,b)
```

```
> df[duplicated(df), ]

> df[!duplicated(df), ]
```

Find and drop duplicate elements: duplicated()

Data Frame Example2:

Display the duplicate records based on the column "col-x" in the dataframe mydata

```
mydata[duplicated(mydata$col-x), ]
```

Find and drop duplicate elements: duplicated()

Data Frame Example2:

Display the duplicate records based on the column "mpg" in the dataframe mtcars

Remove the duplicate records based on the column "cyl" in the dataframe mtcars

Find and drop duplicate elements: unique()

```
x <- c(1, 1, 4, 5, 4, 6)

unique(x)
```

O/P:

```
[1] 1 4 5 6
```

Find and drop duplicate elements: unique()

```
a <- c(rep("A", 3), rep("B", 3), rep("C",2))
b <- c(1,1,2,4,1,1,2,2)
df <-data.frame(a,b)
unique(df)

> unique(df)
  a b
1 A 1
3 A 2
4 B 4
5 B 1
7 C 2
```