Practical No. 01

AIM - Practical of Data Collection, Data curation and management for Unstructured data (NoSQL, CouchDB).

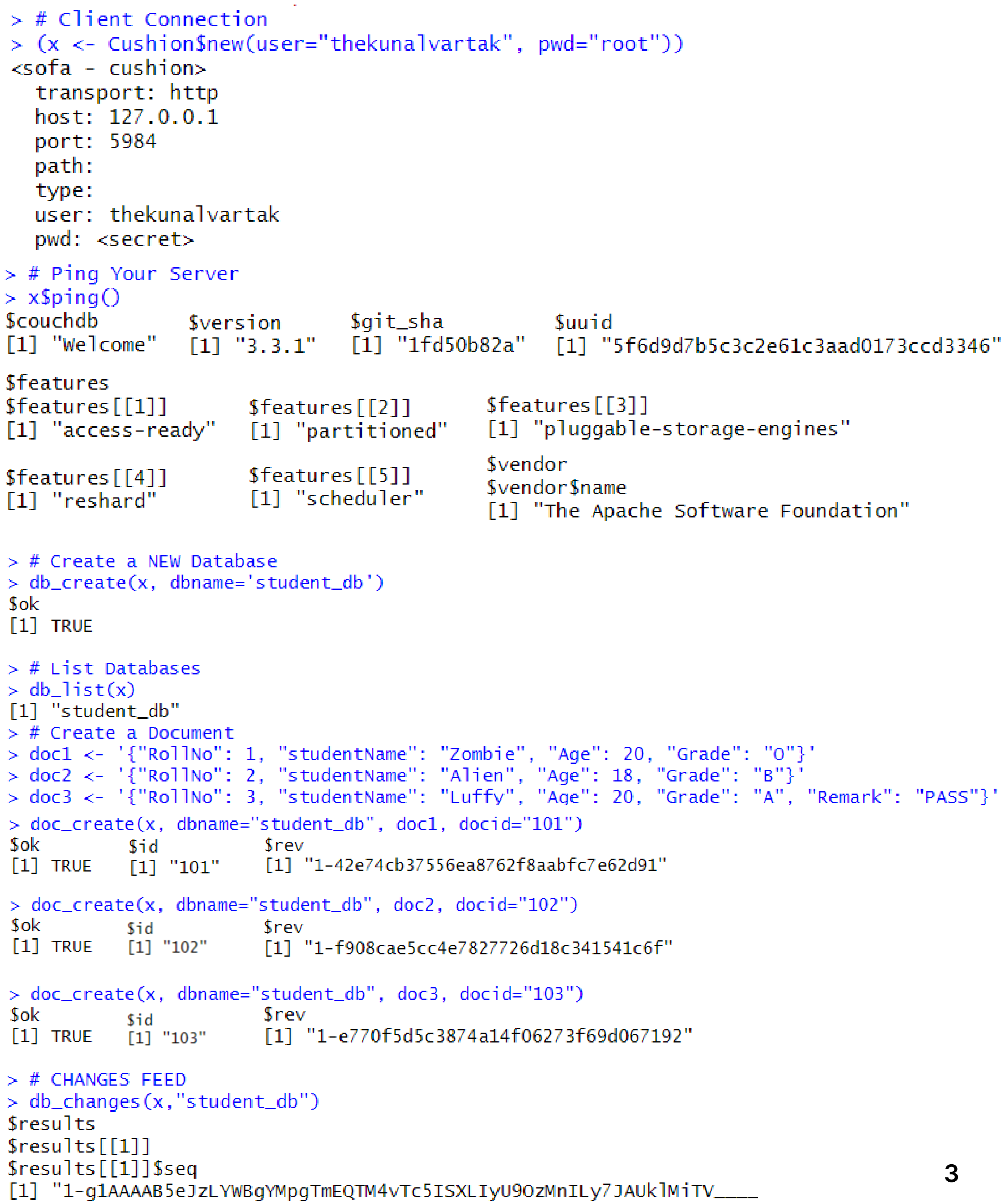
Source Code -

```
install.packages('sofa')
library('sofa')
#------------------------------

# Client Connection
(x <- Cushion$new(user="thekunalvartak", pwd="root"))

# Ping Your Server
x$ping()

# Create a NEW Database
db_create(x, dbname='student_db')

# List Databases
db_list(x)

# Create a Document
doc1 <- '{"RollNo": 1, "studentName": "Zombie", "Age": 20, "Grade": "O"}'
doc2 <- '{"RollNo": 2, "studentName": "Alien", "Age": 18, "Grade": "B"}'
doc3 <- '{"RollNo": 3, "studentName": "Luffy", "Age": 20, "Grade": "A", "Remark":
"PASS"}'

doc_create(x, dbname="student_db", doc1, docid="101")
doc_create(x, dbname="student_db", doc2, docid="102")
doc_create(x, dbname="student_db", doc3, docid="103")

# CHANGES FEED
db_changes(x,"student_db")

# Search for id -> null so all docs will display
db_query(x, dbname="student_db", selector=list('_id'=list('$gt'=NULL)))$docs

# Search for Students with Grade A
db_query(x, dbname="student_db", selector=list(Grade="A"))$docs

# Search for Students with Remark PASS
db_query(x, dbname="student_db", selector=list(Remark="PASS"))$docs
```

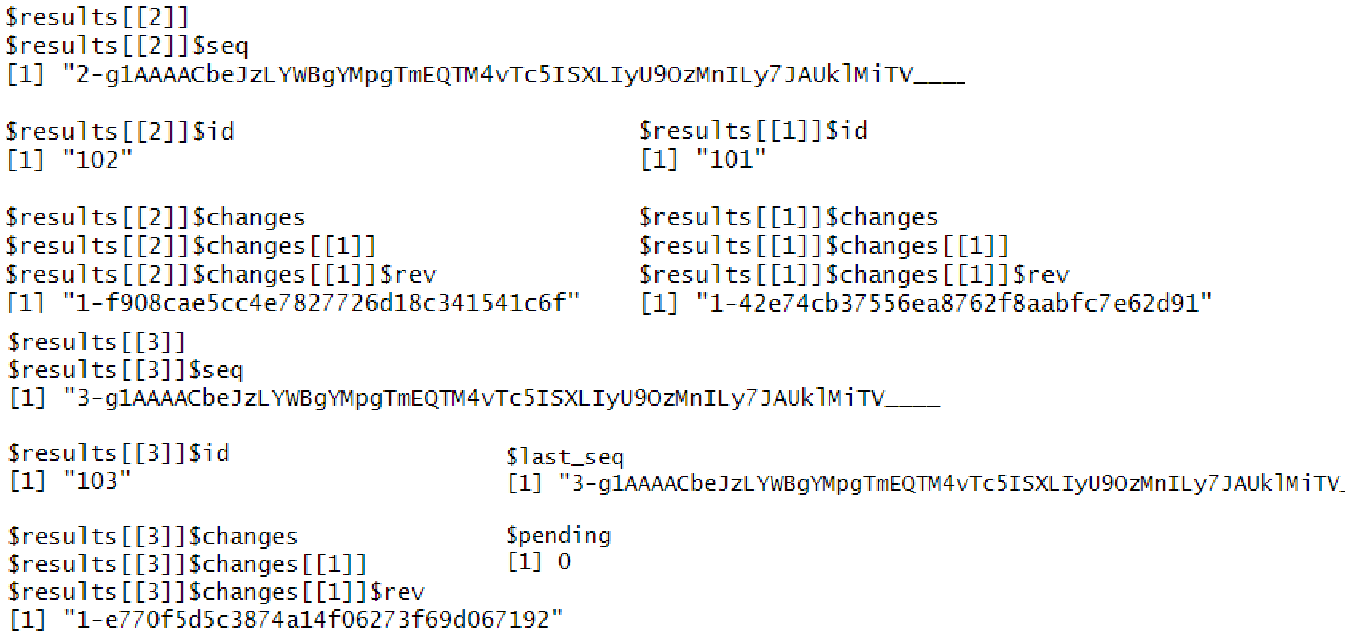
```
# Search for only certain fields where RollNo -> 2
db_query(x, dbname="student_db", selector=list(RollNo=list('$gt'='2')),
fields=c("studentName","Grade"))$docs

# Search for Students with Age 20
db_query(x, dbname="student_db", selector=list(Age=20))$docs


# Convert Result of a Query into a dataframe using JSONLITE
library("jsonlite")
res <- db_query(x, dbname="student_db", selector=list('_id'=list('$gt'=NULL)),
fields=c("studentName","RollNo","Grade","Remark"), as="json")

# Display json DOC
fromJSON(res)$docs


# Update DOC
doc4 <- '{"RollNo": 3, "studentName": "Luffy", "Age": 20, "Grade": "F", "Remark":
"FAIL"}'
doc_update(x, dbname="student_db", doc=doc4, docid="103",
rev="1-e770f5d5c3874a14f06273f69d067192")


# Delete DOC
doc_delete(x, dbname="student_db", docid="103")
doc_get(x, dbname="student_db", docid="103")

# Drop Database
db_delete(x, dbname="student_db")
```

OUTPUT -

```
> install.packages('sofa')
WARNING: Rtools is required to build R packages but is not currently installed.

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Kunal/AppData/Local/R/win-library/4.2’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/sofa_0.4.0.zip'
Content type 'application/zip' length 964620 bytes (942 KB)
downloaded 942 KB

package ‘sofa’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Kunal\AppData\Local\Temp\RtmpuMXcwg\downloaded_packages
> library('sofa')
```



```
> # Client Connection
> (x <- Cushion$new(user="thekunalvartak", pwd="root"))
<sofa - cushion>
  transport: http
  host: 127.0.0.1
  port: 5984
  path:
  type:
  user: thekunalvartak
  pwd: <secret>

> # Ping Your Server
> x$ping()
$couchdb        $version     $git_sha        $uuid
[1] "Welcome"   [1] "3.3.1"  [1] "1fd50b82a"  [1] "5f6d9d7b5c3c2e61c3aad0173ccd3346"

$features
$features[[1]]       $features[[2]]      $features[[3]]
[1] "access-ready"   [1] "partitioned"   [1] "pluggable-storage-engines"

$features[[4]]       $features[[5]]      $vendor
[1] "reshard"        [1] "scheduler"     $vendor$name
                                         [1] "The Apache Software Foundation"

> # Create a NEW Database
> db_create(x, dbname='student_db')
$ok
[1] TRUE

> # List Databases
> db_list(x)
[1] "student_db"
> # Create a Document
> doc1 <- '{"RollNo": 1, "studentName": "Zombie", "Age": 20, "Grade": "O"}'
> doc2 <- '{"RollNo": 2, "studentName": "Alien", "Age": 18, "Grade": "B"}'
> doc3 <- '{"RollNo": 3, "studentName": "Luffy", "Age": 20, "Grade": "A", "Remark": "PASS"}'
> doc_create(x, dbname="student_db", doc1, docid="101")
$ok         $id         $rev
[1] TRUE    [1] "101"   [1] "1-42e74cb37556ea8762f8aabfc7e62d91"

> doc_create(x, dbname="student_db", doc2, docid="102")
$ok         $id         $rev
[1] TRUE    [1] "102"   [1] "1-f908cae5cc4e7827726d18c341541c6f"

> doc_create(x, dbname="student_db", doc3, docid="103")
$ok         $id         $rev
[1] TRUE    [1] "103"   [1] "1-e770f5d5c3874a14f06273f69d067192"

> # CHANGES FEED
> db_changes(x,"student_db")
$results
$results[[1]]
$results[[1]]$seq
[1] "1-g1AAAAB5eJzLYWBgYMpgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUklMiTV____
```

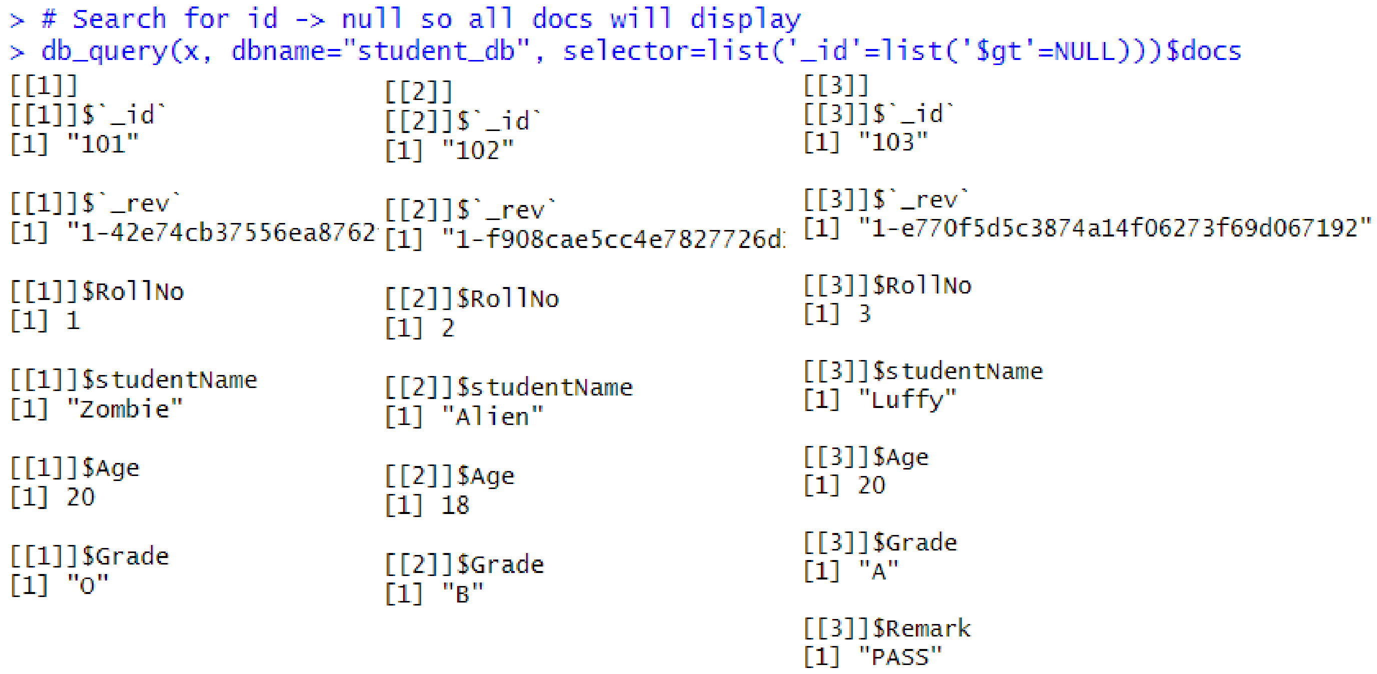


```
$results[[2]]
$results[[2]]$seq
[1] "2-g1AAAACbeJzLYWBgYMpgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUklMiTV

$results[[2]]$id
[1] "102"

$results[[2]]$changes
$results[[2]]$changes[[1]]
$results[[2]]$changes[[1]]$rev
[1] "1-f908cae5cc4e7827726d18c341541c6f"
```

```
$results[[1]]$id
[1] "101"

$results[[1]]$changes
$results[[1]]$changes[[1]]
$results[[1]]$changes[[1]]$rev
[1] "1-42e74cb37556ea8762f8aabfc7e62d91"
```

```
$results[[3]]
$results[[3]]$seq
[1] "3-g1AAAACbeJzLYWBgYMpgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUklMiTV

$results[[3]]$id
[1] "103"

$results[[3]]$changes
$results[[3]]$changes[[1]]
$results[[3]]$changes[[1]]$rev
[1] "1-e770f5d5c3874a14f06273f69d067192"
```

```
$last_seq
[1] "3-g1AAAACbeJzLYWBgYMpgTmEQTM4vTc5ISXLIyU9OzMnILy7JAUklMiTV

$pending
[1] 0
```

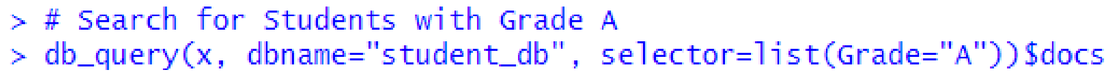


```
> # Search for id -> null so all docs will display
> db_query(x, dbname="student_db", selector=list('_id'=list('$gt'=NULL)))$docs
```

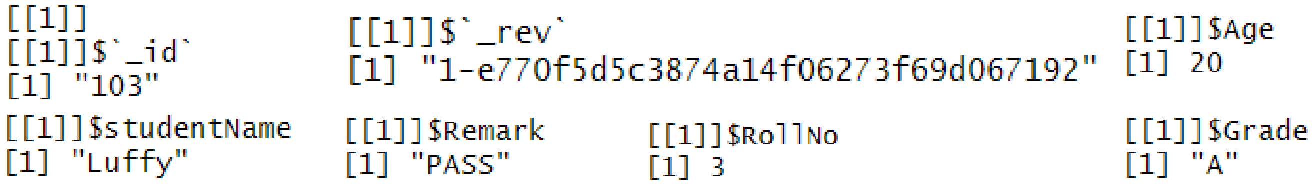
```
[[1]]
[[1]]$`_id`
[1] "101"

[[1]]$`_rev`
[1] "1-42e74cb37556ea8762

[[1]]$RollNo
[1] 1

[[1]]$studentName
[1] "Zombie"

[[1]]$Age
[1] 20

[[1]]$Grade
[1] "O"
```

```
[[2]]
[[2]]$`_id`
[1] "102"

[[2]]$`_rev`
[1] "1-f908cae5cc4e7827726d

[[2]]$RollNo
[1] 2

[[2]]$studentName
[1] "Alien"

[[2]]$Age
[1] 18

[[2]]$Grade
[1] "B"
```

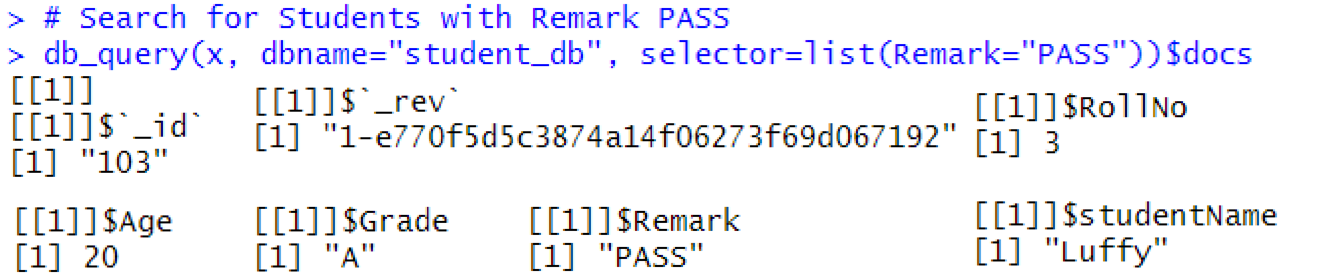
```
[[3]]
[[3]]$`_id`
[1] "103"

[[3]]$`_rev`
[1] "1-e770f5d5c3874a14f06273f69d067192"

[[3]]$RollNo
[1] 3

[[3]]$studentName
[1] "Luffy"

[[3]]$Age
[1] 20

[[3]]$Grade
[1] "A"

[[3]]$Remark
[1] "PASS"
```



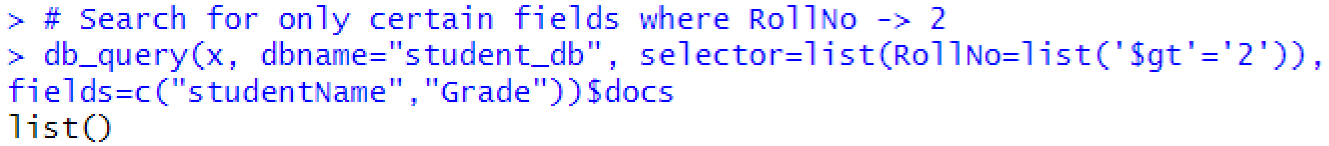
```
> # Search for Students with Grade A
> db_query(x, dbname="student_db", selector=list(Grade="A"))$docs
```



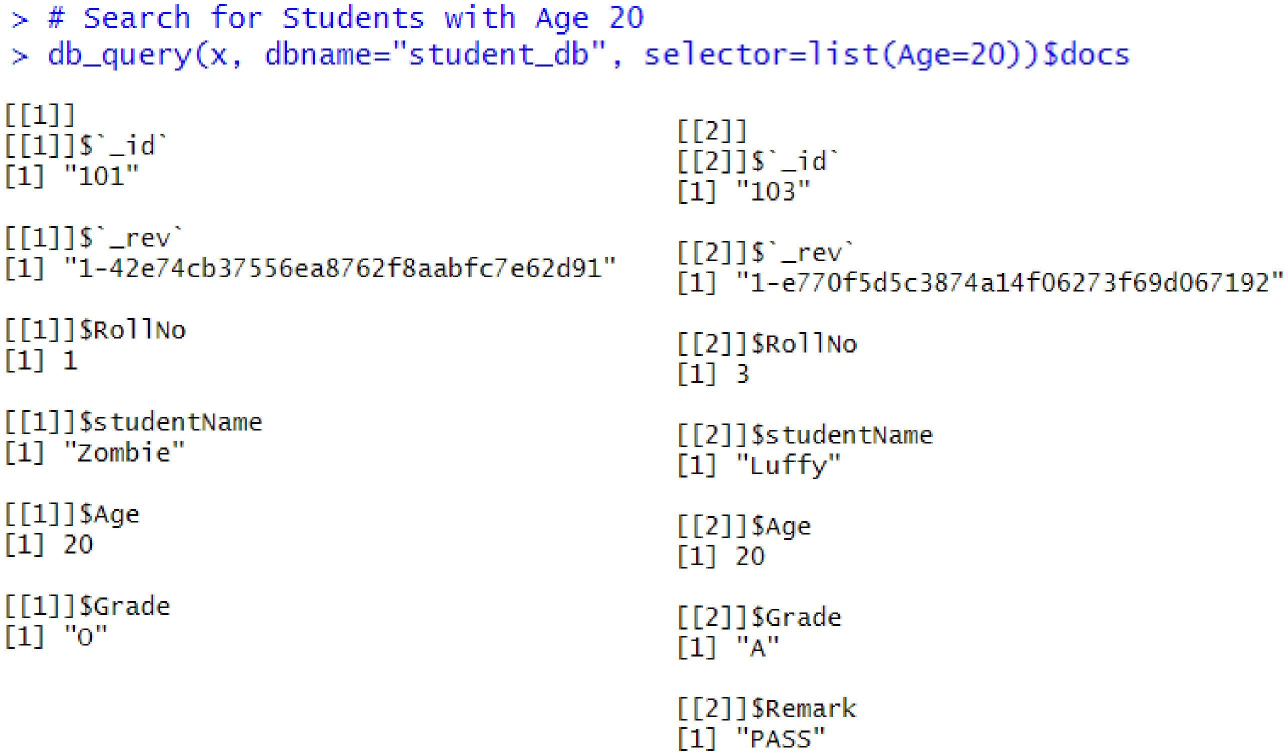
```
[[1]]
[[1]]$`_id`
[1] "103"

[[1]]$`_rev`
[1] "1-e770f5d5c3874a14f06273f69d067192"

[[1]]$Age
[1] 20

[[1]]$studentName
[1] "Luffy"

[[1]]$Remark
[1] "PASS"

[[1]]$RollNo
[1] 3

[[1]]$Grade
[1] "A"
```



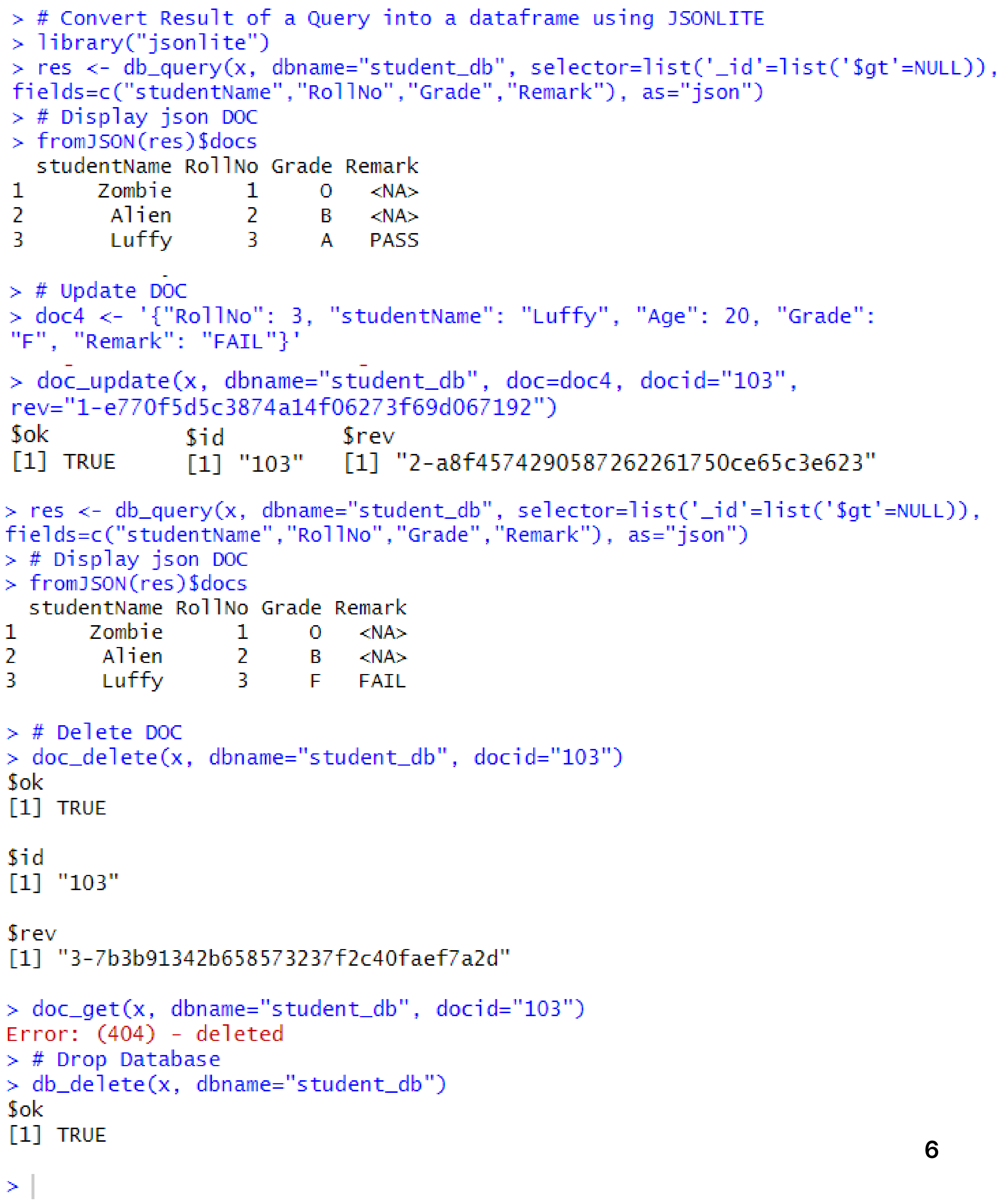
```
> # Search for Students with Remark PASS
> db_query(x, dbname="student_db", selector=list(Remark="PASS"))$docs
[[1]]
[[1]]$`_id`
[1] "103"

[[1]]$`_rev`
[1] "1-e770f5d5c3874a14f06273f69d067192"

[[1]]$RollNo
[1] 3

[[1]]$Age
[1] 20

[[1]]$Grade
[1] "A"

[[1]]$Remark
[1] "PASS"

[[1]]$studentName
[1] "Luffy"
```



```
> # Search for only certain fields where RollNo -> 2
> db_query(x, dbname="student_db", selector=list(RollNo=list('$gt'='2')),
fields=c("studentName","Grade"))$docs
list()
```



```
> # Search for Students with Age 20
> db_query(x, dbname="student_db", selector=list(Age=20))$docs

[[1]]
[[1]]$`_id`
[1] "101"

[[1]]$`_rev`
[1] "1-42e74cb37556ea8762f8aabfc7e62d91"

[[1]]$RollNo
[1] 1

[[1]]$studentName
[1] "Zombie"

[[1]]$Age
[1] 20

[[1]]$Grade
[1] "O"

[[2]]
[[2]]$`_id`
[1] "103"

[[2]]$`_rev`
[1] "1-e770f5d5c3874a14f06273f69d067192"

[[2]]$RollNo
[1] 3

[[2]]$studentName
[1] "Luffy"

[[2]]$Age
[1] 20

[[2]]$Grade
[1] "A"

[[2]]$Remark
[1] "PASS"
```



```
> # Convert Result of a Query into a dataframe using JSONLITE
> library("jsonlite")
> res <- db_query(x, dbname="student_db", selector=list('_id'=list('$gt'=NULL)),
fields=c("studentName","RollNo","Grade","Remark"), as="json")
> # Display json DOC
> fromJSON(res)$docs
  studentName RollNo Grade Remark
1      Zombie      1     O   <NA>
2       Alien      2     B   <NA>
3       Luffy      3     A   PASS

> # Update DOC
> doc4 <- '{"RollNo": 3, "studentName": "Luffy", "Age": 20, "Grade":
"F", "Remark": "FAIL"}'

> doc_update(x, dbname="student_db", doc=doc4, docid="103",
rev="1-e770f5d5c3874a14f06273f69d067192")
$ok            $id            $rev
[1] TRUE       [1] "103"      [1] "2-a8f4574290587262261750ce65c3e623"

> res <- db_query(x, dbname="student_db", selector=list('_id'=list('$gt'=NULL)),
fields=c("studentName","RollNo","Grade","Remark"), as="json")
> # Display json DOC
> fromJSON(res)$docs
  studentName RollNo Grade Remark
1      Zombie      1     O   <NA>
2       Alien      2     B   <NA>
3       Luffy      3     F   FAIL

> # Delete DOC
> doc_delete(x, dbname="student_db", docid="103")
$ok
[1] TRUE

$id
[1] "103"

$rev
[1] "3-7b3b91342b658573237f2c40faef7a2d"

> doc_get(x, dbname="student_db", docid="103")
Error: (404) - deleted
> # Drop Database
> db_delete(x, dbname="student_db")
$ok
[1] TRUE

> 
```