r-project

Importujemy pakiet DBI by łączyć się z bazą danych SQLite.

```
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 4.1.2
```

Łączymy się z bazą danych "got-dataset.db"

```
con <- dbConnect(drv=RSQLite::SQLite(), dbname="data/got-dataset.db")
print(con)
```

```
## <SQLiteConnection>
##   Path: C:\Users\WLGS\Documents\GitHub\got-dialogues-data-stats\data\got-dataset.db
##   Extensions: TRUE
```

Sprawdzamy jakie tabele zawiera baza danych.

```
dbListTables(con)
```

```
## [1] "got-dialogues"
```

Przystępujemy do procesu czyszczenia danych. Interesują nas dialogi wypowiedziane tylko przez postacie w serialu (wpis Speaker nie może być pusty).

```
query <- "DELETE
FROM [got-dialogues]
WHERE SPEAKER = ''"
res <- dbSendStatement(con, query)
print("Amount of rows the query affected: ")
```

```
## [1] "Amount of rows the query affected: "
```

```
dbGetRowsAffected(res)
```

```
## [1] 0
```

```
dbClearResult(res)
```

Usuwamy dialogi postaci w tle, przedstawianych jako np. Woman #4

```
query <- "DELETE
FROM [got-dialogues]
WHERE Speaker LIKE '%#%'"
res <- dbSendStatement(con, query)
print("Amount of rows the query affected: ")
```

```
## [1] "Amount of rows the query affected: "
```

```
dbGetRowsAffected(res)
```

```
## [1] 0
```

```
dbClearResult(res)
```

Usuwamy wspólne dialogi, dotyczy pozycji typu ‘ALL TOGETHER’, ‘ALL THREE’, ‘ALL AT THE BACK’.

```
query <- "DELETE
FROM [got-dialogues]
WHERE Speaker LIKE '%ALL%'"
res <- dbSendStatement(con, query)
print("Amount of rows the query affected: ")
```

```
## [1] "Amount of rows the query affected: "
```

```
dbGetRowsAffected(res)
```

```
## [1] 0
```

```
dbClearResult(res)
```

Usuwamy pojedyncze, mało znaczące wypowiedzi.

```
query <- "DELETE
FROM [got-dialogues]
where Speaker in (SELECT Speaker
FROM [got-dialogues]
GROUP BY Speaker
HAVING COUNT(*)=1)"
res <- dbSendStatement(con, query)
print("Amount of rows the query affected: ")
```

```
## [1] "Amount of rows the query affected: "
```

```
dbGetRowsAffected(res)
```

```
## [1] 0
```

```
dbClearResult(res)
```

W bazie danych pojawiają się wpisy typu Speaker=‘Roose’ oraz Speaker=‘ROOSE’, dlatego trzymamy się jednej wersji (UPPER)

```
query <- "UPDATE [got-dialogues]
SET
    Speaker = UPPER(Speaker)"
res <- dbSendStatement(con, query)
print("Amount of rows the query affected: ")
```

```
## [1] "Amount of rows the query affected: "
```

```
dbGetRowsAffected(res)
```

```
## [1] 24096
```

```
dbClearResult(res)
```