rstatsZH - Data Science mit R

SQL in R Markdown + Datenbank Abfragen

Lars Schöbitz

2021-11-23

Datenbanken - SQL queries in R

- `DBI` Package: https://dbi.r-dbi.org/
- `dbplyr` Package: https://dbplyr.tidyverse.org/index.html

DBI - Mit einer Datenbank verbinden

Argumente variieren je nach Datenbank, aber das erste Argument ist immer das Datenbank Backend.

```
library(DBI)

con <- dbConnect(

   # Hier wird das Backend definiert
   RMariaDB::MariaDB(),

   host = "relational.fit.cvut.cz",
   port = 3306,

   # Niemals Passwort in Skript speichern
   username = "guest",
   password = rstudioapi::askForPassword("Datenbank Passwort"),

   dbname = "sakila"
)
```

DBI - Tabellen in Datenbank anzeigen

```
dbListTables(con)
```

```
 [1] "actor"          "address"        "category"
 [4] "city"           "country"        "customer"
 [7] "film"           "film_actor"     "film_category"
[10] "film_text"      "inventory"      "language"
[13] "payment"        "rental"         "staff"
[16] "store"
```

DBI - Spaltennamen einer Tabelle anzeigen

```
dbListFields(con, "film")
```

```
 [1] "film_id"              "title"
 [3] "description"          "release_year"
 [5] "language_id"          "original_language_id"
 [7] "rental_duration"      "rental_rate"
 [9] "length"               "replacement_cost"
[11] "rating"               "special_features"
[13] "last_update"
```

SQL Queries in R Markdown Dateien

1. Datenbank Verbindung: Code-chunk mit `sql connection=con` starten
2. Daten Output: Resultierende Daten mit `output.var = "NAME"` als Objekt im Environment speichern
3. SQL Code schreiben

````
```{sql connection=con, output.var="film_length_big180"}
SELECT film_id, title, length
FROM film
WHERE length > 180
```
````

SQL Query - Mit Daten weiter arbeiten

1. Objekt mit Funktion `as_tibble()` in einen Tibble umwandeln

```
```{r, echo=TRUE}
film_length_big180 %>%
   as_tibble()
```
```

```
# A tibble: 39 × 3
  film_id title              length
    <int> <chr>               <int>
1      24 ANALYZE HOOSIERS      181
2      50 BAKED CLEOPATRA       182
3     128 CATCH AMISTAD         183
4     141 CHICAGO NORTH         185
5     180 CONSPIRACY SPIRIT     184
6     182 CONTROL ANTHEM        185
# … with 33 more rows
```

dbplyr - Mit Tabelle in Datenbank verbinden

```
film_tab <- tbl(con, "film")
film_tab
```

```
# Source:   table<film> [?? x 13]
# Database: mysql [guest@relational.fit.cvut.cz:NA/sakila]
  film_id title           description   release_year language_id
    <int> <chr>           <chr>                <int>       <int>
1       1 ACADEMY DINOSAUR A Epic Drama…         2006           1
2       2 ACE GOLDFINGER   A Astounding…         2006           1
3       3 ADAPTATION HOLES A Astounding…         2006           1
4       4 AFFAIR PREJUDICE A Fanciful D…         2006           1
5       5 AFRICAN EGG      A Fast-Paced…         2006           1
6       6 AGENT TRUMAN     A Intrepid P…         2006           1
# … with more rows, and 8 more variables:
#   original_language_id <int>, rental_duration <int>,
#   rental_rate <dbl>, length <int>, replacement_cost <dbl>,
#   rating <chr>, special_features <chr>, last_update <dttm>
```

dbplyr - Queries als dplyr code

```
film_tab %>%
   select(film_id, title, length) %>%
   filter(length > 180)
```

```
# Source:   lazy query [?? x 3]
# Database: mysql [guest@relational.fit.cvut.cz:NA/sakila]
  film_id title              length
    <int> <chr>               <int>
1      24 ANALYZE HOOSIERS      181
2      50 BAKED CLEOPATRA       182
3     128 CATCH AMISTAD         183
4     141 CHICAGO NORTH         185
5     180 CONSPIRACY SPIRIT     184
6     182 CONTROL ANTHEM        185
# … with more rows
```

dbplyr - Resultierende Daten aus Datenbank holen

```
film_tab %>%
   select(film_id, title, length) %>%
   filter(length > 180) %>%
   collect()
```

```
# A tibble: 39 × 3
  film_id title              length
    <int> <chr>               <int>
1      24 ANALYZE HOOSIERS      181
2      50 BAKED CLEOPATRA       182
3     128 CATCH AMISTAD         183
4     141 CHICAGO NORTH         185
5     180 CONSPIRACY SPIRIT     184
6     182 CONTROL ANTHEM        185
# … with 33 more rows
```

dbplyr - Queries als dplyr code

```
film_tab %>%
   summarise(min_rate = min(rental_rate),
             max_rate = max(rental_rate),
             mean_rate = mean(rental_rate)) %>%
   collect()
```

```
# A tibble: 1 × 3
  min_rate max_rate mean_rate
     <dbl>    <dbl>     <dbl>
1     0.99     4.99      2.98
```

R Packages für andere Datentypen

- **googlesheets4:** Google Sheets
- **haven**: SPSS, Stata, und SAS Dateien
- **jsonline**: JSON
- **xml2**: xml
- **rvest**: web scraping
- **httr**: web APIs
- **sparklyr**: data loaded into spark

🌻 Danke

Für die Aufmerksamkeit!

Für die R packages {xaringan} und {xaringanthemer} mit welchen die Folien geschrieben wurden.

Eine PDF Version der Folien kann hier heruntergeladen werden: https://github.com/rstatsZH/website/raw/master/slides/e1_d01-willkommen/e1_d01-willkommen.pdf

---