SQL-to-R Code

Prithviraj Lakkakula

Contents

In this post, I will query the data from different data tables in both SQL and R's dplyr with a goal of obtaining the same output. SQL code will be on the your left and R code will be on your right.

This is a work in progress as I continue to add the queries in the coming weeks starting with simple queries to more complicated queries.

Note: In the html file, two columns appear side by side but in pdf version you can view the code only one after the other (not side by side).

Query 1: SELECTing single columns

SQL code

```sql
SELECT name
FROM people
LIMIT 10;
```

Table 1: Displaying records 1 - 10

name
50 Cent
A. Michael Baldwin
A. Raven Cruz
A.J. Buckley
A.J. DeLucia
A.J. Langer

name
Aaliyah
Aaron Ashmore
Aaron Hann
Aaron Hill

R code

```
people %>%
  select(name) %>%
  head(n = 10)
```

```
##                   name
## 1              50 Cent
## 2  A. Michael Baldwin
## 3       A. Raven Cruz
## 4        A.J. Buckley
## 5        A.J. DeLucia
## 6         A.J. Langer
## 7             Aaliyah
## 8       Aaron Ashmore
## 9          Aaron Hann
## 10         Aaron Hill
```

Query 2: SELECTing multiple columns

SQL code

```
SELECT name, birthdate
FROM people
LIMIT 10;
```

Table 2: Displaying records 1 - 10

name	birthdate
50 Cent	7/6/75
A. Michael Baldwin	4/4/63
A. Raven Cruz	
A.J. Buckley	2/9/78
A.J. DeLucia	
A.J. Langer	5/22/74
Aaliyah	1/16/79
Aaron Ashmore	10/7/79
Aaron Hann	
Aaron Hill	4/23/83

R code

```
people %>%
  select(name, birthdate) %>%
  head(n = 10)
```

```
##                   name birthdate
## 1              50 Cent    7/6/75
## 2  A. Michael Baldwin    4/4/63
## 3       A. Raven Cruz
## 4        A.J. Buckley    2/9/78
## 5        A.J. DeLucia
## 6         A.J. Langer   5/22/74
## 7             Aaliyah   1/16/79
## 8       Aaron Ashmore   10/7/79
## 9          Aaron Hann
## 10         Aaron Hill   4/23/83
```

Query 3: SELECTing all columns

SQL code

```
SELECT *
FROM people
LIMIT 10;
```

Table 3: Displaying records 1 - 10

id	name	birthdate	deathdate
1	50 Cent	7/6/75	
2	A. Michael Baldwin	4/4/63	
3	A. Raven Cruz		
4	A.J. Buckley	2/9/78	
5	A.J. DeLucia		
6	A.J. Langer	5/22/74	
7	Aaliyah	1/16/79	8/25/01
8	Aaron Ashmore	10/7/79	
9	Aaron Hann		
10	Aaron Hill	4/23/83	

R code

```
people %>%
  head(n = 10)
```

```
##    id               name birthdate deathdate
## 1   1            50 Cent    7/6/75
## 2   2 A. Michael Baldwin    4/4/63
## 3   3      A. Raven Cruz
## 4   4       A.J. Buckley    2/9/78
## 5   5       A.J. DeLucia
## 6   6        A.J. Langer   5/22/74
## 7   7            Aaliyah   1/16/79   8/25/01
## 8   8      Aaron Ashmore   10/7/79
## 9   9         Aaron Hann
## 10 10         Aaron Hill   4/23/83
```

Query 4: Excluding specific columns

SQL code

```
SELECT id, name, birthdate
FROM people
LIMIT 10;
```

Table 4: Displaying records 1 - 10

id	name	birthdate
1	50 Cent	7/6/75
2	A. Michael Baldwin	4/4/63
3	A. Raven Cruz	
4	A.J. Buckley	2/9/78
5	A.J. DeLucia	
6	A.J. Langer	5/22/74
7	Aaliyah	1/16/79
8	Aaron Ashmore	10/7/79
9	Aaron Hann	
10	Aaron Hill	4/23/83

R code

```
people %>%
  select(-deathdate) %>%
  head(n = 10)
```

```
##    id               name birthdate
## 1   1            50 Cent    7/6/75
## 2   2 A. Michael Baldwin    4/4/63
## 3   3      A. Raven Cruz
## 4   4       A.J. Buckley    2/9/78
## 5   5       A.J. DeLucia
## 6   6        A.J. Langer   5/22/74
## 7   7            Aaliyah   1/16/79
## 8   8      Aaron Ashmore   10/7/79
## 9   9         Aaron Hann
## 10 10         Aaron Hill   4/23/83
```

Query 5: SELECTing DISTINCT columns

SQL code

```
SELECT DISTINCT language
FROM films
LIMIT 10;
```

Table 5: Displaying records 1 - 10

language
NA
German
English
Japanese
Danish
Italian
French
Swedish
Russian
None

R code

```
films %>%
  distinct(language) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 1
##    language
##    <chr>
##  1 <NA>
##  2 German
##  3 English
##  4 Japanese
##  5 Danish
##  6 Italian
##  7 French
##  8 Swedish
##  9 Russian
## 10 None
```

SQL code

```
SELECT DISTINCT country
FROM films
LIMIT 10;
```

Table 6: Displaying records 1 - 10

country
USA
Germany

country
Japan
Denmark
UK
Italy
France
West Germany
Sweden
Soviet Union

R code

```
films %>%
  distinct(country) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 1
##    country
##    <chr>
##  1 USA
##  2 Germany
##  3 Japan
##  4 Denmark
##  5 UK
##  6 Italy
##  7 France
##  8 West Germany
##  9 Sweden
## 10 Soviet Union
```

SQL code

```
SELECT DISTINCT certification
FROM films
LIMIT 10;
```

Table 7: Displaying records 1 - 10

certification
Not Rated
NA
Passed
Unrated
Approved
G
PG
R
PG-13
M

R code

```
films %>%
  distinct(certification) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 1
##    certification
##    <chr>
##  1 Not Rated
##  2 <NA>
##  3 Passed
##  4 Unrated
##  5 Approved
##  6 G
##  7 PG
##  8 R
##  9 PG-13
## 10 M
```

SQL code

```
SELECT DISTINCT role
FROM roles
LIMIT 10;
```

Table 8: 2 records

role
director
actor

R code

```
roles %>%
  distinct(role) %>%
  head(n = 10)
```

```
## # A tibble: 2 x 1
##   role
##   <chr>
## 1 director
## 2 actor
```

Query 6: Learning to COUNT

SQL code: Count the number of rows in `people` table

```
SELECT COUNT(*)
FROM people;
```

Table 9: 1 records

COUNT(*)
8397

R code: Count the number of rows in `people` table

```
people %>%
  count()
```

```
##      n
## 1 8397
```

SQL code: Count the number of birth dates in the `people` table

```
SELECT COUNT(birthdate)
FROM people;
```

Table 10: 1 records

COUNT(birthdate)
8397

R code: Count the number of birth dates in the `people` table

```
people %>% select(birthdate) %>%
  count()
```

```
##      n
## 1 8397
```

SQL code: Count the number of DISTINCT birth dates in the `people` table

```
SELECT COUNT(DISTINCT birthdate)
FROM people;
```

Table 11: 1 records

COUNT(DISTINCT birthdate)
5399

R code: Count the number of DISTINCT birth dates in the `people` table

```
people %>% select(birthdate) %>%
  n_distinct()
```

```
## [1] 5399
```

SQL code: Count the number of DISTINCT languages in the `films` table

```sql
SELECT COUNT(DISTINCT language)
FROM films;
```

Table 12: 1 records

COUNT(DISTINCT language)
47

R code: Count the number of DISTINCT languages in the `films` table

```r
films %>% select(language) %>%
  n_distinct()
```

```
## [1] 48
```

::::

SQL code: Count the number of DISTINCT languages in the `films` table

```sql
SELECT COUNT(DISTINCT country)
FROM films;
```

Table 13: 1 records

COUNT(DISTINCT country)
64

R code: Count the number of DISTINCT languages in the `films` table

```r
films %>% select(country) %>%
  n_distinct()
```

```
## [1] 65
```

Query 7: Filtering of numeric values

SQL code: selects all details for films with a budget over ten thousand dollars

```sql
SELECT *
FROM films
WHERE budget > 10000
LIMIT 5;
```

Table 14: 5 records

id	title	release_year	country	duration	language	certification	gross	budget
1	Intolerance: Love's Struggle Throughout the Ages	1916	USA	123	NA	Not Rated	NA	385907
2	Over the Hill to the Poorhouse	1920	USA	110	NA	NA	3000000	100000
3	The Big Parade	1925	USA	151	NA	Not Rated	NA	245000
4	Metropolis	1927	Germany	145	German	Not Rated	26435	6000000
6	The Broadway Melody	1929	USA	100	English	Passed	2808000	379000

R code: selects all details for films with a budget over ten thousand dollars

```
films %>%
  filter(budget > 10000) %>%
  head(n = 5)
```

```
## # A tibble: 5 x 9
##      id title       release_year country duration language certification   gross
##   <dbl> <chr>              <dbl> <chr>      <dbl> <chr>    <chr>           <dbl>
## 1     1 Intoleranc~         1916 USA          123 <NA>     Not Rated          NA
## 2     2 Over the H~         1920 USA          110 <NA>     <NA>          3000000
## 3     3 The Big Pa~         1925 USA          151 <NA>     Not Rated          NA
## 4     4 Metropolis          1927 Germany      145 German   Not Rated       26435
## 5     6 The Broadw~         1929 USA          100 English  Passed        2808000
## # ... with 1 more variable: budget <dbl>
```

SQL code: selects all details for all films released in 2016

```
SELECT *
FROM films
WHERE release_year = 2016
LIMIT 5;
```

Table 15: 5 records

id	title	release_year	country	duration	language	certification	gross	budg
4821	10 Cloverfield Lane	2016	USA	104	English	PG-13	71897215	1.5e+
4822	13 Hours	2016	USA	144	English	R	52822418	5.0e+
4823	A Beginner's Guide to Snuff	2016	USA	87	English	NA	NA	N
4824	Airlift	2016	India	130	Hindi	NA	NA	4.4e+
4825	Alice Through the Looking Glass	2016	USA	113	English	PG	76846624	1.7e+

R code: selects all details for all films released in 2016

```
films %>%
  filter(release_year == 2016) %>%
  head(n = 5)
```

```
## # A tibble: 5 x 9
##      id title       release_year country duration language certification   gross
##   <dbl> <chr>              <dbl> <chr>      <dbl> <chr>    <chr>           <dbl>
## 1  4821 10 Cloverf~         2016 USA          104 English  PG-13          7.19e7
## 2  4822 13 Hours            2016 USA          144 English  R              5.28e7
## 3  4823 A Beginner~         2016 USA           87 English  <NA>          NA
## 4  4824 Airlift             2016 India        130 Hindi    <NA>          NA
## 5  4825 Alice Thro~         2016 USA          113 English  PG             7.68e7
## # ... with 1 more variable: budget <dbl>
```

SQL code: selects number of films released before 2000

```
SELECT COUNT(release_year)
FROM films
WHERE release_year <2000;
```

Table 16: 1 records

COUNT(release_year)
1337

R code: selects number of films released before 2000

```
films %>%
  count(release_year < 2000)
```

```
## # A tibble: 3 x 2
##   ‘release_year < 2000‘     n
##   <lgl>                 <int>
## 1 FALSE                  3589
## 2 TRUE                   1337
## 3 NA                       42
```

Query 8: Filtering text

SQL code: gets the titles of all films which were filmed in China

```
SELECT title
FROM films
WHERE country = 'China' -- in PostgreSQL you must use single quotes
LIMIT 5;
```

Table 17: 5 records

title
The Last Emperor
Hero
Hero
House of Flying Daggers

title
The Promise

R code: gets the titles of all films which were filmed in China

```
films %>%
  filter(country == "China") %>% # here you must use double quotes around text
  select(title) %>%
  head(n = 5)
```

```
## # A tibble: 5 x 1
##   title
##   <chr>
## 1 The Last Emperor
## 2 Hero
## 3 Hero
## 4 House of Flying Daggers
## 5 The Promise
```

SQL code: gets all the details for all French language films

```
SELECT *
FROM films
WHERE language = 'French' -- in PostgreSQL you must use single quotes
LIMIT 5;
```

Table 18: 5 records

id	title	release_year	country	duration	language	certification	gross	budget
108	Une Femme Mariée	1964	France	94	French	NA	NA	1.2e+05
111	Pierrot le Fou	1965	France	110	French	Not Rated	NA	3.0e+05
140	Mississippi Mermaid	1969	France	123	French	R	26893	1.6e+06
423	Subway	1985	France	98	French	R	NA	1.7e+07
662	Les visiteurs	1993	France	107	French	R	700000	5.0e+07

R code: gets all the details for all French language films

```
films %>%
  filter(language == "French") %>% # here you must use double quotes around text
  head(n = 5)
```

```
## # A tibble: 5 x 9
##      id title release_year country duration language certification  gross budget
##   <dbl> <chr>        <dbl> <chr>      <dbl> <chr>    <chr>          <dbl>  <dbl>
## 1   108 Une ~         1964 France        94 French   <NA>              NA  1.2e5
## 2   111 Pier~         1965 France       110 French   Not Rated         NA  3   e5
## 3   140 Miss~         1969 France       123 French   R              26893  1.6e6
## 4   423 Subw~         1985 France        98 French   R                 NA  1.7e7
## 5   662 Les ~         1993 France       107 French   R             700000  5   e7
```

SQL code: Get the name and birth date of the person born on November 11th, 1974.

```sql
SELECT name birthdate
FROM people
WHERE birthdate = '1974-11-11' -- in PostgreSQL you must use single quotes
LIMIT 5;
```

Table 19: 0 records

birthdate

R code: Get the name and birth date of the person born on November 11th, 1974.

```r
people %>%
  select(name, birthdate) %>%
  filter(birthdate == "1974-11-11") %>% # here you must use double quotes around text
  head(n = 5)
```

```
## [1] name      birthdate
## <0 rows> (or 0-length row.names)
```

SQL code: Get the number of Hindi language films

```sql
SELECT COUNT(language)
FROM films
WHERE language = 'Hindi'; -- in PostgreSQL you must use single quotes
```

Table 20: 1 records

COUNT(language)
28

R code: Get the number of Hindi language films

```r
films %>%
  filter(language == "Hindi") %>% # here you must use double quotes around text
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    28
```

Query 9: Use WHERE and AND for multiple conditions