**PART 1**

**Task 1**

-- PART 1, task 1

-- First variant

**select** **distinct** *ff*.title,name **from** film *ff*

**left** **join** (**select** film\_id,

category\_id

**from** public.film\_category) *fc* **on** *ff*.film\_id = *fc*.film\_id

**left** **join** (**select** category\_id,

name

**from** public.category) *ct* **on** *ct*.category\_id = *fc*.category\_id

**where** *ff*.release\_year **between** 2017 **and** 2019 **and** rental\_rate > 1 **and** name = 'Animation'

**order** **by** title;

-- Second variant

**with** film\_category **as**( **select** *fc*.film\_id,

*fc*.category\_id,

*ct*.name

**from** public.film\_category *fc*

**left** **join** (**select** category\_id,

name

**from** public.category) *ct*

**on** *ct*.category\_id = *fc*.category\_id )

**select** **distinct** *ff*.title,

*ct*.***name*** **from** film *ff*

**left** **join** film\_category *ct*

**on** *ff*.film\_id = *ct*.film\_id

**where** *ff*.release\_year **between** 2017 **and** 2019 **and** rental\_rate > 1 **and** ***name*** = 'Animation'

**order** **by** title;

-- Third variant

**select** **distinct** *ff*.title,name **from** film *ff*

**left** **join** (**select** film\_id,

category\_id

**from** public.film\_category) *fc* **on** *ff*.film\_id = *fc*.film\_id

**inner** **join** (**select** category\_id,

name

**from** public.category **where** name = 'Animation' ) *ct* **on** *ct*.category\_id = *fc*.category\_id

**where** *ff*.release\_year **between** 2017 **and** 2019 **and** rental\_rate > 1

**order** **by** title;

I would like to use third variant, because this variant is faster then other. It is faster because here we write condition “name = 'Animation'” in subquery which joined by “inner join” and it let us to join only needed rows.

**Task 2**

-- PART 1, task 2

-- First variant

**select** **concat**(*ad*.address,' ',*ad*.address2) **as** *address*,

**sum**(amount) **as** *revenue* **from** store *s*

**left** **join** (**select** store\_id,

inventory\_id **from** inventory )*i* **on** *i*.store\_id = *s*.store\_id

**left** **join** (**select** inventory\_id,

rental\_id **from** rental) *r* **on** *r*.inventory\_id = *i*.inventory\_id

**left** **join** (**select** rental\_id,

payment\_date,

amount **from** payment ) *p* **on** *p*.rental\_id = *r*.rental\_id

**left** **join** (**select** address\_id,

address,

address2 **from** address) *ad* **on** *ad*.address\_id = *s*.address\_id

**where** payment\_date >= '01.03.2017'

**group** **by** **concat**(*ad*.address,' ',*ad*.address2),

*s*.store\_id;

-- Second variant

**with** store\_with\_address **as** (**select** *s*.store\_id,

*s*.address\_id,

**concat**(*ad*.address,' ',*ad*.address2) **as** *address* **from** store *s*

**left** **join** (**select** address\_id,

address,

address2 **from** address) *ad* **on** *ad*.address\_id = *s*.address\_id)

**select** *s*.address,

**sum**(amount) **from** ***store\_with\_address*** *s*

**left** **join** (**select** store\_id,

inventory\_id **from** inventory )*i* **on** *i*.store\_id = *s*.store\_id

**left** **join** (**select** inventory\_id,

rental\_id **from** rental) *r* **on** *r*.inventory\_id = *i*.inventory\_id

**inner** **join** (**select** rental\_id,

payment\_date,

amount **from** payment **where** payment\_date >= '01.03.2017' ) *p*

**on** *p*.rental\_id = *r*.rental\_id

**group** **by** address; -- I will use this variant because this variant is more optimized

--and execution time is smaller then in other scripts

I would like to use second variant, because this variant is faster because I optimized this script by creating a tempo table-with. Also used ‘Inner join’ instead of ‘left join’ where we join table ‘payment’.

**Task 3**

-- Part 1, task 3

-- First variant

**select** first\_name,

last\_name,

**count**(*fl*.film\_id) **as** *number\_of\_movies*

**from** actor *ac*

**left** **join** public.film\_actor *fc*

**on** *ac*.actor\_id = *fc*.actor\_id

**left** **join** public.film *fl*

**on** *fl*.film\_id = *fc*.film\_id

**where** *fl*.release\_year >= 2015

**group** **by** first\_name, last\_name

**order** **by** **count**(*fl*.film\_id) **desc**

**limit** 5 ;

-- Second variant

**select** first\_name,

last\_name,

**count**(*fl*.film\_id) **as** *number\_of\_movies*

**from** actor *ac*

**left** **join** (**select** actor\_id,

film\_id

**from** public.film\_actor) *fc*

**on** *ac*.actor\_id = *fc*.actor\_id

**inner** **join** (**select** film\_id

**from** public.film *fl* **where** *fl*.release\_year >= 2015) *fl*

**on** *fl*.film\_id = *fc*.film\_id

--where fl.release\_year >= 2015

**group** **by** first\_name, last\_name

**order** **by** **count**(*fl*.film\_id) **desc**

**limit** 5 ; -- I will use this variant because this variant is more optimized

--and execution time is smaller then in other scripts

-- Third variant

**with** number\_of\_movies **as** ( **select** actor\_id,

**count**(*fl*.film\_id) *number\_of\_movies*

**from** film\_actor *fc*

**left** **join** (**select** film\_id **from** film *fl*

**where** *fl*.release\_year >= 2015 ) *fl*

**on** *fl*.film\_id = *fc*.film\_id

**group** **by** actor\_id)

**select** first\_name,

last\_name,

*nm*.number\_of\_movies **from** actor *ac*

**left** **join** ***number\_of\_movies*** *nm*

**on** *ac*.actor\_id = *nm*.actor\_id

**order** **by** *nm*.number\_of\_movies **desc**

**limit** 5 ;

I would like to use second variant, because this variant is faster because we use ‘Inner join’ instead of ‘left join’ where we join subquery with table ‘film’ and also we wrote condition “*fl*.release\_year >= 2015” in this subquery and it let us to join only needed rows

**Task 4**

-- PART 1, Tasc 4

-- First variant

**select** **distinct** f.release\_year,

--fc.name,

**count**(**case** **when** **name** = 'Drama' **then** 1 **end**) **as** number\_of\_drama\_movies,

**count**(**case** **when** **name** = 'Travel' **then** 1 **end** ) **as** number\_of\_travel\_movies,

**count**(**case** **when** **name** = 'Documentary' **then** 1 **end** ) **as** number\_of\_documentary\_movies

**from** film f

**left** **join** film\_category fc **on** f.film\_id = fc.film\_id

**left** **join** (**select** category\_id,

**name**

**from** public.category) ct

**on** ct.category\_id = fc.category\_id

**group** **by** f.release\_year

**order** **by** release\_year **desc**;

-- Second variant

**with** film\_category\_new **as**( **select** fc.film\_id,

fc.category\_id,

ct.**name**

**from** public.film\_category fc

**left** **join** (**select** category\_id,

**name**

**from** public.category) ct

**on** ct.category\_id = fc.category\_id )

**select** **distinct** *f*.release\_year,

--fc.name,

**count**(**case** **when** **name** = 'Drama' **then** 1 **end**) **as** *number\_of\_drama\_movies*,

**count**(**case** **when** **name** = 'Travel' **then** 1 **end** ) **as** *number\_of\_travel\_movies*,

**count**(**case** **when** **name** = 'Documentary' **then** 1 **end** ) **as** *number\_of\_documentary\_movies*

**from** film *f*

**left** **join** film\_category\_new fc **on** f.film\_id = fc.film\_id

**where** **name** **in** ('Drama','Travel','Documentary') **and** *f*.release\_year **is** **not** **null**

**group** **by** *f*.release\_year

**order** **by** release\_year **desc**;

I would like to use second variant, because this variant is faster because I optimized this script by creating a tempo table-with.

**Task 5**

--PART 1, task 5

-- First variant

**select** *ct*.customer\_id,

**string\_agg**(title,',') **as** *list\_of\_horrors*,

**sum**(amount) **from** public.customer *ct*

**left** **join** public.payment *pt*

**on** *pt*.customer\_id = *ct*.customer\_id

**left** **join** public.rental *r*

**on** *pt*.rental\_id = *r*.rental\_id

**left** **join** public.inventory *inv*

**on** *inv*.inventory\_id = *r*.inventory\_id

**left** **join** public.film *f*

**on** *f*.film\_id = *inv*.film\_id

**left** **join** public.film\_category *fc*

**on** *fc*.film\_id = *f*.film\_id

**left** **join** public.category *cg*

**on** *cg*.category\_id = *fc*.category\_id

**where** *cg*.name = 'Horror'

**group** **by** *ct*.customer\_id;

-- Second variant

**select** *ct*.customer\_id,

**string\_agg**(title,',') **as** *list\_of\_horrors*,

**sum**(amount) **from** public.customer *ct*

**left** **join** (**select** customer\_id,

amount,

rental\_id

**from** public.payment ) *pt*

**on** *pt*.customer\_id = *ct*.customer\_id

**left** **join** (**select** rental\_id,

inventory\_id

**from** public.rental) *r*

**on** *pt*.rental\_id = *r*.rental\_id

**left** **join** (**select** inventory\_id,

film\_id

**from** public.inventory) *inv*

**on** *inv*.inventory\_id = *r*.inventory\_id

**left** **join** (**select** film\_id,

title

**from** public.film) *f*

**on** *f*.film\_id = *inv*.film\_id

**left** **join** (**select** film\_id,

category\_id

**from** public.film\_category) *fc*

**on** *fc*.film\_id = *f*.film\_id

**inner** **join** (**select** category\_id

**from** public.category

**where** name = 'Horror') *cg*

**on** *cg*.category\_id = *fc*.category\_id

**group** **by** *ct*.customer\_id;

I prefer to use second variant because we make subqueries with tables that we join and select only the required fields

**PART 2**

**Task 1**

**select** ll.staff\_id,

store\_id,

amount

**from** (**select** \* **from** (

**select** r.staff\_id,store\_id,payment\_date,

**row\_number**() **over** ( **partition** **by** r.staff\_id **order** **by** pm.payment\_date **desc**) rn **from** rental r

**left** **join** payment pm

**on** pm.rental\_id = r.rental\_id

**left** **join** public.inventory inv

**on** inv.inventory\_id = r.inventory\_id ) l **where** rn = 1 ) ll

-- this part of the script give us information about

-- in which store manager worked last time

**left** **join** (

**select** r.staff\_id, **sum**(amount) **as** amount

**from** rental r

**left** **join** payment pm

**on** pm.rental\_id = r.rental\_id

**left** **join** public.inventory inv

**on** inv.inventory\_id = r.inventory\_id

**where** payment\_date **between** '01.01.2017' **and** '31.12.2017'

**group** **by** r.staff\_id ) r

-- this part of the script give us information about how much revenue

-- a manager generated in the stores

**on** r.staff\_id = ll.staff\_id

**order** **by** amount **desc**

**limit** 3

**Comments in script**

**Task 2**

--Part 2, task 2

**select** title,

**count**(*r*.rental\_id) **as** *num\_of\_rent*,

**case** **when** *f*.rating = 'G' **then** '3+'

**when** *f*.rating = 'PG' **then** '7+'

**when** *f*.rating = 'PG-13' **then** '13+'

**when** *f*.rating = 'R' **then** '17+'

**when** *f*.rating = 'NC-17' **then** '18+'

**end** *expected\_age*

**from** film *f*

**left** **join** public.inventory *inv*

**on** *f*.film\_id = *inv*.film\_id

**left** **join** public.rental *r*

**on** *inv*.inventory\_id = *r*.inventory\_id

**group** **by** title,*f*.rating

**order** **by** **count**(rental\_id) **desc**

**limit** 5

**PART 3**

**V1**

**select** actor\_id,

*name*,

**EXTRACT**(**YEAR** **FROM** current\_date) - *last\_year* **as** *inactive\_period*

**from** (

**select** *act*.actor\_id,

**concat**( first\_name,' ', last\_name) **as** *name*,

**max**(*f*.release\_year) **as** *last\_year*

**from** public.actor *act*

**left** **join** public.film\_actor *fl*

**on** *fl*.actor\_id = *act*.actor\_id

**left** **join** film *f*

**on** *f*.film\_id = *fl*.film\_id

**group** **by** *act*.actor\_id,

**concat**( first\_name,' ', last\_name) ) *k*

**order** **by** **EXTRACT**(**YEAR** **FROM** current\_date) - last\_year **desc**