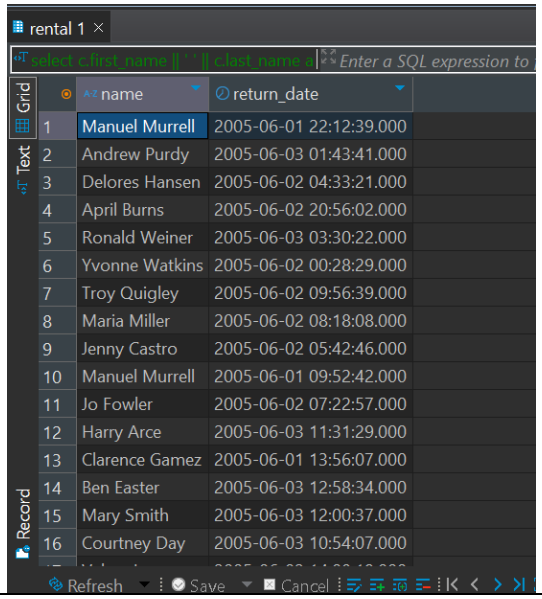
Assignment Day 13

1. Dapatkan **nama lengkap** customers dan waktu pengembalian untuk semua transaksi yang melebihi batas waktu pengembalian (7 hari)

```sql
select
	c.first_name || ' ' ||	c.last_name as name,
	r.return_date
from
customer c
join
rental r on c.customer_id = r.customer_id
where
r.return_date - r.rental_date > INTERVAL '7 days';
```



rental 1

	name	return_date
1	Manuel Murrell	2005-06-01 22:12:39.000
2	Andrew Purdy	2005-06-03 01:43:41.000
3	Delores Hansen	2005-06-02 04:33:21.000
4	April Burns	2005-06-02 20:56:02.000
5	Ronald Weiner	2005-06-03 03:30:22.000
6	Yvonne Watkins	2005-06-02 00:28:29.000
7	Troy Quigley	2005-06-02 09:56:39.000
8	Maria Miller	2005-06-02 08:18:08.000
9	Jenny Castro	2005-06-02 05:42:46.000
10	Manuel Murrell	2005-06-01 09:52:42.000
11	Jo Fowler	2005-06-02 07:22:57.000
12	Harry Arce	2005-06-03 11:31:29.000
13	Clarence Gamez	2005-06-01 13:56:07.000
14	Ben Easter	2005-06-03 12:58:34.000
15	Mary Smith	2005-06-03 12:00:37.000
16	Courtney Day	2005-06-03 10:54:07.000

Grid Text Record Refresh Save Cancel

2. Tampilkan nama pelanggan yang melakukan transaksi peminjaman lebih dari sekali pada hari Senin! **Gunakan CTE**!

```sql
with transactions_on_monday as(
select
	c.first_name || ' ' ||	c.last_name as name,
	c.customer_id,
	r.rental_date ,
	to_char(r.rental_date::date, 'Day') as rental_day
from
	customer c
join
	rental r on c.customer_id = r.customer_id
where
	extract (DOW from r.rental_date::date)=1
)
select
	name,
	count(rental_date) as total_transactions
from
	transactions_on_monday
group by
```

```
        name
having
        count(rental_date) > 1
```

Results 1

with transactions_on_monday as(sele | Enter a SQL expression to filter results (us

Grid | Text | Record

	name	total_transactions
1	Aaron Selby	6
2	Adam Gooch	4
3	Adrian Clary	4
4	Agnes Bishop	4
5	Alan Kahn	2
6	Albert Crouse	2
7	Alberto Henning	4
8	Alex Gresham	5
9	Alexander Fennell	4
10	Alfred Casillas	4
11	Alfredo Mcadams	5
12	Alice Stewart	3
13	Alicia Mills	3
14	Allan Cornish	3
15	Allen Butterfield	3

Refresh | Save | Cancel | Export

200 row(s) fetched - 0.011s, on 2024-09-28 at 08:00:53

3. Temukan nama aktor dan jumlah film yang dimainkan, serta peringkat aktor berdasarkan jumlah film. Urutkan berdasarkan peringkat secara **ascending**. Gunakan **RANK**!

```
with ActorFilmCount as (
    select
        a.actor_id,
        a.first_name || ' ' || a.last_name as actor_name,
        count(fa.film_id) as film_count
    from
        actor a
    join
        film_actor fa on a.actor_id = fa.actor_id
    join
        film f on fa.film_id = f.film_id
    group by
        a.actor_id, a.first_name, a.last_name
)
select
    actor_name,
    film_count,
    dense_rank() over (order by film_count asc) as actor_rank
from
    ActorFilmCount
order by
    actor_rank asc;
```

Results 1

Enter a SQL expression to filter results (use Ctrl+Space)

	actor_name	film_count	actor_rank
1	Emily Dee	14	1
2	Julia Fawcett	15	2
3	Judy Dean	15	2
4	Julia Zellweger	16	3
5	Adam Grant	18	4
6	Sissy Sobieski	18	4
7	Russell Close	19	5
8	Sandra Peck	19	5
9	Penelope Guiness	19	5
10	Cameron Wray	19	5
11	Christopher Berry	20	6
12	Minnie Kilmer	20	6
13	Chris Depp	20	6
14	Matthew Johansson	20	6
15	Kenneth Pesci	20	6

Refresh Save Cancel Export data 200 200+

200 row(s) fetched - 0.006s (0.001s fetch), on 2024-09-28 at 21:25:02

Smart Insert 63 : 16 : 1293 Sel: 0 | 0

4. Tampilkan (semua kolom) dengan job_title yang memiliki salary_in_usd lebih besar dari rata-rata salary dari seluruh job_title. Namun, tampilkan hanya company_size = S.

```sql
select
	*
from
	ds_salaries ds
where
	salary_in_usd >
	(select avg(salary_in_usd) from ds_salaries ds2)
and
	company_size = 'S';
```

ds_salaries 1

Enter a SQL expression to filter results (use Ctrl+Space)

	id	work_year	expr	em	job_title	salary	sa	salary_in_usd	employee_residen
1	1	2,020	SE	FT	Machine Learning Scientist	260,000	USD	260,000	JP
2	6	2,020	SE	FT	Lead Data Scientist	190,000	USD	190,000	US
3	9	2,020	SE	FT	Lead Data Engineer	125,000	USD	125,000	NZ
4	17	2,020	SE	FT	Big Data Engineer	100,000	EUR	114,047	PL
5	39	2,020	EN	FT	Machine Learning Engineer	138,000	USD	138,000	US
6	107	2,021	SE	FT	Data Engineer	115,000	USD	115,000	US
7	126	2,021	SE	FT	Machine Learning Scientist	120,000	USD	120,000	US
8	149	2,021	SE	FT	Cloud Data Engineer	160,000	USD	160,000	BR
9	150	2,021	SE	FT	Director of Data Science	168,000	USD	168,000	JP
10	159	2,021	EN	FT	Machine Learning Engineer	125,000	USD	125,000	US
11	225	2,021	EX	CT	Principal Data Scientist	416,000	USD	416,000	US
12	231	2,021	SE	FT	ML Engineer	256,000	USD	256,000	US
13	265	2,021	SE	FT	Lead Data Engineer	160,000	USD	160,000	PR
14	375	2,022	EX	FT	Lead Data Engineer	150,000	CAD	118,187	CA

Refresh Save Cancel Export data 200 17

17 row(s) fetched - 0.015s, on 2024-09-28 at 09:21:30

Lampiran :

```sql
SELECT count(*)
FROM rental r
WHERE return_date - rental_date > INTERVAL '7 days';

-- No 1
select
	c.first_name || ' ' ||	c.last_name as name,
	r.return_date
from
customer c
join
rental r on c.customer_id = r.customer_id
where
r.return_date - r.rental_date > INTERVAL '7 days';

select * from rental r limit 5

--NO 2
with transactions_on_monday as(
select
	c.first_name || ' ' ||	c.last_name as name,
	c.customer_id,
	r.rental_date ,
	to_char(r.rental_date::date, 'Day') as rental_day
from
	customer c
join
	rental r on c.customer_id = r.customer_id
where
	extract (DOW from r.rental_date::date)=1
)
select
	name,
	count(rental_date) as total_transactions
from
	transactions_on_monday
group by
	name
having
	count(rental_date) > 1

select * from actor a limit 2;

select * from film_actor fa limit 2;

--No 3
with ActorFilmCount as (
    select
        a.actor_id,
        a.first_name || ' ' || a.last_name as actor_name,
        count(fa.film_id) as film_count
    from
        actor a
    join
        film_actor fa on a.actor_id = fa.actor_id
    join
        film f on fa.film_id = f.film_id
```

```sql
    group by
        a.actor_id, a.first_name, a.last_name
)
select
    actor_name,
    film_count,
    dense_rank() over (order by film_count asc) as actor_rank
from
    ActorFilmCount
order by
    actor_rank asc;

select * from ds_salaries ds where company_size ='S'

select avg(salary_in_usd) from ds_salaries ds

--No 4
select
      *
from
      ds_salaries ds
where
      salary_in_usd >
      (select avg(salary_in_usd) from ds_salaries ds2)
and
      company_size = 'S'
```