Creating Transactional Query

1. Mencari mobil keluaran 2015 ke atas

```
1  -- nomer 1
2  select * from product
3  where tahun > 2015
4  limit 10
```

Data output

	product_id [PK] integer	brand character varying (50)	model character varying (100)	body_type character varying (50)	tahun integer	harga integer
1	1	Toyota	Toyota Yaris	Hatchback	2016	175000000
2	2	Toyota	Toyota Yaris	Hatchback	2018	215000000
3	4	Toyota	Toyota Yaris	Hatchback	2020	220000000
4	6	Toyota	Toyota Agya	Hatchback	2019	114000000
5	8	Toyota	Toyota Agya	Hatchback	2016	110000000
6	9	Toyota	Toyota Agya	Hatchback	2022	155500000
7	10	Toyota	Toyota Agya	Hatchback	2017	115500000
8	11	Toyota	Toyota Calya	Wagon	2019	130000000
9	12	Toyota	Toyota Calya	Wagon	2019	137000000
10	13	Toyota	Toyota Calya	Wagon	2017	115500000

2. Menambahkan satu data bid produk baru
   Sebelum insert

Data Output | Messages | Notifications

	bid_id [PK] integer	product_id integer	user_id integer	iklan_id integer	harga_bit integer	tanggal_bit timestamp without time zone	bit_status character varying (50)
30	63	22	5	67	181614169	2023-04-04 17:51:31.63317	Sent
31	64	33	48	85	631962031	2023-03-04 20:47:55.346885	Sent
32	65	32	47	4	118562855	2023-04-13 23:12:36.250155	Sent
33	66	45	49	66	417810129	2023-08-27 15:47:17.368715	Sent
34	68	17	25	95	690175931	2023-11-04 09:21:25.22614	Sent
35	69	45	10	34	347175550	2023-11-19 21:45:57.035403	Sent
36	71	18	27	41	247712235	2023-06-28 23:40:30.637485	Sent
37	74	12	9	94	686071889	2023-08-06 12:25:22.647637	Sent
38	75	45	49	73	142627612	2023-05-09 18:23:29.920264	Sent
39	76	37	11	51	476607713	2023-05-27 01:38:39.786163	Sent
40	77	22	12	68	502344381	2023-09-06 09:36:31.478086	Sent
41	78	50	40	38	121006362	2023-03-30 12:02:52.790589	Sent
42	79	15	50	82	473536765	2023-02-28 02:13:59.140626	Sent
43	81	29	19	29	380434064	2023-11-23 02:07:14.728824	Sent
44	83	42	34	55	471290636	2023-08-17 10:29:56.904526	Sent
45	85	17	45	57	460033004	2023-10-13 09:34:07.275593	Sent
46	87	11	10	58	489521028	2023-03-06 11:54:46.416745	Sent
47	91	21	47	41	443294635	2023-05-04 09:50:50.935024	Sent
48	92	40	45	35	242788541	2023-10-14 12:02:57.497866	Sent
49	95	23	29	79	129357596	2023-08-15 00:28:15.915886	Sent
50	97	4	10	76	489975150	2023-05-01 14:25:01.162891	Sent

Total rows: 50 of 50 | Query complete 00:00:00.091

```sql
-- nomor 2 insert
insert into bid (bid_id, product_id, user_id, iklan_id, harga_bit, tanggal_bit, bit_status)
values('101','2','100','30','150000000', '2023-01-01 14:05:27.12345','Sent')
```

Setelah di insert datanya nambah

```sql
select * from bid
where bit_status = 'Sent'
```

Total row nya menjadi 51

Data Output | Messages | Notifications

	bid_id [PK] integer	product_id integer	user_id integer	iklan_id integer	harga_bit integer	tanggal_bit timestamp without time zone	bit_status character varying (50)
29	61	3	50	100	117432627	2023-05-22 02:27:47.510318	Sent
30	63	22	5	67	181614169	2023-04-04 17:51:31.63317	Sent
31	64	33	48	85	631962031	2023-03-04 20:47:55.346885	Sent
32	65	32	47	4	118562855	2023-04-13 23:12:36.250155	Sent
33	66	45	49	66	417810129	2023-08-27 15:47:17.368715	Sent
34	68	17	25	95	690175931	2023-11-04 09:21:25.22614	Sent
35	69	45	10	34	347175550	2023-11-19 21:45:57.035403	Sent
36	71	18	27	41	247712235	2023-06-28 23:40:30.637485	Sent
37	74	12	9	94	686071889	2023-08-06 12:25:22.647637	Sent
38	75	45	49	73	142627612	2023-05-09 18:23:29.920264	Sent
39	76	37	11	51	476607713	2023-05-27 01:38:39.786163	Sent
40	77	22	12	68	502344381	2023-09-06 09:36:31.478086	Sent
41	78	50	40	38	121006362	2023-03-30 12:02:52.790589	Sent
42	79	15	50	82	473536765	2023-02-28 02:13:59.140626	Sent
43	81	29	19	29	380434064	2023-11-23 02:07:14.728824	Sent
44	83	42	34	55	471290636	2023-08-17 10:29:56.904526	Sent
45	85	17	45	57	460033004	2023-10-13 09:34:07.275593	Sent
46	87	11	10	58	489521028	2023-03-06 11:54:46.416745	Sent
47	91	21	47	41	443294635	2023-05-04 09:50:50.935024	Sent
48	92	40	45	35	242788541	2023-10-14 12:02:57.497866	Sent
49	95	23	29	79	129357596	2023-08-15 00:28:15.915886	Sent
50	97	4	10	76	489975150	2023-05-01 14:25:01.162891	Sent
51	101	2	100	30	150000000	2023-01-01 14:05:27.12345	Sent

Total rows: 51 of 51 | Query complete 00:00:00.098

3. Melihat semua mobil yg dijual 1 akun dari yg paling baru

```
16  -- nomer 3
17  select product_id, brand, model, tahun, harga, tanggal_posting from iklan
18  left join user_info using(user_id)
19  left join product using(product_id)
20  where nama_user = 'Daniswara Astuti, S.H.'
21  order by tanggal_posting desc
```

Data output

	product_id integer	brand character varying (50)	model character varying (100)	tahun integer	harga integer	tanggal_posting timestamp without time zone
1	40	Honda	Honda CR-V	2018	398500000	2023-08-22 00:00:00
2	38	Honda	Honda CR-V	2009	116000000	2023-05-07 00:00:00
3	6	Toyota	Toyota Agya	2019	114000000	2023-01-16 00:00:00

4. Mencari mobil bekas yang termurah berdasarkan keyword

```
23  -- nomer 4
24  select product_id, brand, model, tahun, harga
25  from product
26  where lower(model) like '%yaris%'
27  order by harga
```

Data output

Data Output | Messages | Notifications

	product_id [PK] integer	brand character varying (50)	model character varying (100)	tahun integer	harga integer
1	5	Toyota	Toyota Yaris	2012	124000000
2	3	Toyota	Toyota Yaris	2014	162000000
3	1	Toyota	Toyota Yaris	2016	175000000
4	2	Toyota	Toyota Yaris	2018	215000000
5	4	Toyota	Toyota Yaris	2020	220000000

5. Mencari mobil bekas yang terdekat berdasarkan sebuah id kota, jarak terdekat dihitung berdasarkan latitude longitude. Perhitungan jarak dapat dihitung menggunakan rumus jarak euclidean berdasarkan latitude dan longitude

```
30  SELECT
31      p.product_id,
32      p.brand,
33      p.model,
34      k.kota_id,
35      SQRT(POW(k.lokasi[0] - k.lokasi[0], 2) + POW(k.lokasi[1] - k.lokasi[1], 2)) AS distance
36  FROM
37      product p
38  JOIN
39      iklan i ON p.product_id = i.product_id
40  JOIN
41      user_info ui on ui.user_id = i.user_id
42  JOIN
43      kota k ON ui.kota_id = k.kota_id
44  WHERE
45      k.kota_id = 3
46  ORDER BY
47      distance
```

Data output

Data Output | Messages | Notifications

	product_id integer	brand character varying (50)	model character varying (100)	kota_id integer	distance double precision
1	20	Daihatsu	Daihatsu Ayla	3	0
2	43	Honda	Honda Civic	3	0
3	40	Honda	Honda CR-V	3	0
4	11	Toyota	Toyota Calya	3	0
5	7	Toyota	Toyota Agya	3	0
6	12	Toyota	Toyota Calya	3	0
7	45	Honda	Honda Civic	3	0
8	4	Toyota	Toyota Yaris	3	0
9	50	Suzuki	Suzuki Ertiga	3	0

Creating Analytical Query

1. Ranking popularitas model mobil berdasarkan jumlah bid

```sql
-- nomer 1
select
    model,
    count(model) as count_product,
    count(*) as count_bid
from bid
left join product using(product_id)
group by 1
order by count_bid desc
limit 5
```

Data output

Data Output | Messages | Notifications

	model character varying (100)	count_product bigint	count_bid bigint
1	Honda Civic	13	13
2	Honda CR-V	12	12
3	Daihatsu Ayla	12	12
4	Honda Jazz	12	12
5	Daihatsu Xenia	11	11

2. Membandingkan harga mobil berdasarkan harga rata-rata per kota

```sql
SELECT
    k.nama_kota,
    p.brand,
    p.model,
    p.tahun,
    p.harga,
    ROUND(AVG(p.harga) OVER (PARTITION BY k.nama_kota)) AS avg_car_city
FROM
    product p
JOIN
    iklan i ON p.product_id = i.product_id
JOIN
    user_info ui ON i.user_id = ui.user_id
JOIN
    kota k on ui.kota_id = k.kota_id
limit 10
```

Data output

Data Output | Messages | Notifications

	nama_kota character varying (50)	brand character varying (50)	model character varying (100)	tahun integer	harga integer	avg_car_city numeric
1	Kota Balikpapan	Honda	Honda Jazz	2019	250000000	210700000
2	Kota Balikpapan	Daihatsu	Daihatsu Terios	2018	190000000	210700000
3	Kota Balikpapan	Daihatsu	Daihatsu Ayla	2016	83000000	210700000
4	Kota Balikpapan	Honda	Honda CR-V	2018	415000000	210700000
5	Kota Balikpapan	Toyota	Toyota Calya	2017	115500000	210700000
6	Kota Bandung	Honda	Honda CR-V	2018	415000000	236666667
7	Kota Bandung	Daihatsu	Daihatsu Ayla	2017	113000000	236666667
8	Kota Bandung	Honda	Honda CR-V	2016	269000000	236666667
9	Kota Bandung	Toyota	Toyota Calya	2019	137000000	236666667
10	Kota Bandung	Honda	Honda Jazz	2019	250000000	236666667

3. Dari penawaran suatu model mobil, cari perbandingan tanggal user melakukan bid dengan bid selanjutnya beserta harga tawar yang diberikan

```sql
-- nomer 3
WITH BidComparison AS (
    SELECT
        model,
        user_id,
        tanggal_bid AS first_bid_date,
        LEAD(tanggal_bid) OVER (PARTITION BY model ORDER BY tanggal_bid) AS next_bid_date,
        harga_bid AS first_bid_price,
        LEAD(harga_bid) OVER (PARTITION BY model ORDER BY tanggal_bid) AS next_bid_price
    FROM
        bid join product using(product_id)
    WHERE
        model = 'Toyota Yaris'
)
SELECT
    model,
    user_id,
    first_bid_date,
    next_bid_date,
    first_bid_price,
    next_bid_price
FROM
    BidComparison
ORDER BY
    user_id;
```

Data output

Data Output | Messages | Notifications

	model character varying (100)	user_id integer	first_bid_date timestamp without time zone	next_bid_date timestamp without time zone	first_bid_price integer	next_bid_price integer
1	Toyota Yaris	5	2023-12-04 12:45:41.855237	[null]	435253248	[null]
2	Toyota Yaris	10	2023-05-01 14:25:01.162891	2023-05-09 21:41:19.408239	489975150	616690543
3	Toyota Yaris	23	2023-04-30 22:05:50.323165	2023-05-01 14:25:01.162891	145631141	489975150
4	Toyota Yaris	24	2023-10-18 09:55:44.944169	2023-11-06 02:33:12.287781	109669259	302569259
5	Toyota Yaris	31	2023-05-09 21:41:19.408239	2023-05-22 02:27:47.510318	616690543	117432627
6	Toyota Yaris	31	2023-09-15 18:03:53.463784	2023-10-18 09:55:44.944169	356885915	109669259
7	Toyota Yaris	41	2023-01-15 03:39:48.474727	2023-04-30 22:05:50.323165	670041833	145631141
8	Toyota Yaris	50	2023-05-22 02:27:47.510318	2023-09-15 18:03:53.463784	117432627	356885915
9	Toyota Yaris	50	2023-11-06 02:33:12.287781	2023-12-04 12:45:41.855237	302569259	435253248
10	Toyota Yaris	100	2023-01-01 14:05:27.12345	2023-01-15 03:39:48.474727	150000000	670041833

4. Membandingkan persentase perbedaan rata-rata harga mobil berdasarkan modelnya dan rata-rata harga bid yang ditawarkan oleh customer pada 6 bulan terakhir

- Difference adalah selisih antara rata-rata harga model mobil(avg_price) dengan rata-rata harga bid yang ditawarkan terhadap model tersebut(avg_bid_6month)
- Difference dapat bernilai negatif atau positif
- Difference_percent adalah persentase dari selisih yang telah dihitung, yaitu dengan cara difference dibagi rata-rata harga model mobil(avg_price) dikali 100%
- Difference_percent dapat bernilai negatif atau positi

```sql
WITH AvgPrices AS (
    SELECT
        model,
        AVG(harga) AS avg_price
    FROM
        product
    GROUP BY
        model
),
AvgBids AS (
    SELECT
        p.model,
        AVG(b.harga_bid) AS avg_bid_6month
    FROM
        bid b
    JOIN
        iklan i ON b.iklan_id = i.iklan_id
    JOIN
        product p ON i.product_id = p.product_id
    WHERE
        i.tanggal_posting >= CURRENT_DATE - INTERVAL '6 months'
    GROUP BY
        p.model
)
SELECT
    a.model,
    round(a.avg_price) as avg_price,
    round(b.avg_bid_6month) as avg_bid_6month,
    abs(round(a.avg_price - b.avg_bid_6month)) AS difference,
    abs(round(((a.avg_price - b.avg_bid_6month) / a.avg_price) * 100,2)) AS difference_percent
FROM
    AvgPrices a
JOIN
    AvgBids b ON a.model = b.model;
```

Data output

Data Output | Messages | Notifications

	model character varying (100)	avg_price numeric	avg_bid_6month numeric	difference numeric	difference_percent numeric
1	Toyota Yaris	179200000	272872734	93672734	52.27
2	Daihatsu Ayla	105400000	350507593	245107593	232.55
3	Suzuki Ertiga	147600000	426208813	278608813	188.76
4	Daihatsu Xenia	156900000	424736478	267836478	170.71
5	Toyota Agya	118400000	485595728	367195728	310.13
6	Toyota Calya	118700000	444881649	326181649	274.79
7	Honda Civic	255500000	273998365	18498365	7.24
8	Honda Jazz	214400000	457239749	242839749	113.26
9	Honda CR-V	308700000	364534370	55834370	18.09
10	Daihatsu Terios	193980000	267996211	74016211	38.16

5. Membuat window function rata-rata harga bid sebuah merk dan model mobil selama 6 bulan terakhir

```sql
SELECT
    brand,
    model,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE) THEN harga_bid END)) AS m_min_1,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '1 month') THEN harga_bid END)) AS m_min_2,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '2 months') THEN harga_bid END)) AS m_min_3,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '3 months') THEN harga_bid END)) AS m_min_4,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '4 months') THEN harga_bid END)) AS m_min_5,
    round(AVG(CASE WHEN EXTRACT(MONTH FROM tanggal_bid) = EXTRACT(MONTH FROM CURRENT_DATE - INTERVAL '5 months') THEN harga_bid END)) AS m_min_6
FROM
    bid b
JOIN
    product p ON b.product_id = p.product_id
WHERE
    p.model = 'Toyota Yaris'
    AND b.tanggal_bid >= CURRENT_DATE - INTERVAL '6 months'
GROUP BY
    brand, model;
```

Data output

Data Output | Messages | Notifications

	brand character varying (50)	model character varying (100)	m_min_1 numeric	m_min_2 numeric	m_min_3 numeric	m_min_4 numeric	m_min_5 numeric	m_min_6 numeric
1	Toyota	Toyota Yaris	[null]	435253248	302569259	109669259	356885915	[null]