1)

select category, count(1) as number\_of\_bikes

from bike

group by category

having count(1) > 2;

2)

select c.name, count(m.id) as membership\_count

from membership m

right join customer c on c.id=m.customer\_id

group by c.name

order by count(1) desc;

3)

select id, category

, price\_per\_hour as old\_price\_per\_hour

, case when category = 'electric' then round(price\_per\_hour - (price\_per\_hour\*0.1) ,2)

when category = 'mountain bike' then round(price\_per\_hour - (price\_per\_hour\*0.2) ,2)

else round(price\_per\_hour - (price\_per\_hour\*0.5) ,2)

end as new\_price\_per\_hour

, price\_per\_day as old\_price\_per\_day

, case when category = 'electric' then round(price\_per\_day - (price\_per\_day\*0.2) ,2)

when category = 'mountain bike' then round(price\_per\_day - (price\_per\_day\*0.5) ,2)

else round(price\_per\_day - (price\_per\_day\*0.5) ,2)

end as new\_price\_per\_day

from bike;

4)

select category

, count(case when status ='available' then 1 end) as available\_bikes\_count

, count(case when status ='rented' then 1 end) as rented\_bikes\_count

from bike

group by category;

5)

SOL1 using Group by and Union all:

select extract(year from start\_timestamp) as year

, extract(month from start\_timestamp) as month

, sum(total\_paid) as revenue

from rental

group by extract(year from start\_timestamp), extract(month from start\_timestamp)

union all

select extract(year from start\_timestamp) as year

, null as month, sum(total\_paid) as revenue

from rental

group by extract(year from start\_timestamp)

union all

select null as year, null as month, sum(total\_paid) as revenue

from rental

order by year, month;

SOL2 using Grouping Sets:

select extract(year from start\_timestamp) as year

, extract(month from start\_timestamp) as month

, sum(total\_paid) as revenue

from rental

group by grouping sets ( (year, month), (year), () )

order by year, month;

SOL3 using ROLLUP:

select extract(year from start\_timestamp) as year

, extract(month from start\_timestamp) as month

, sum(total\_paid) as revenue

from rental

group by rollup(year, month)

order by year, month;

6)

select extract(year from start\_date) as year

, extract(month from start\_date) as month

, mt.name as membership\_type\_name

, sum(total\_paid) as total\_revenue

from membership m

join membership\_type mt on m.membership\_type\_id = mt.id

group by year, month, mt.name

order by year, month, mt.name

7)

select mt.name as membership\_type\_name

, extract(month from start\_date) as month

, sum(total\_paid) as total\_revenue

from membership m

join membership\_type mt on m.membership\_type\_id = mt.id

where extract(year from start\_date) = 2023

group by CUBE(membership\_type\_name, month)

order by membership\_type\_name, month;

8)

with cte as

(select customer\_id, count(1)

, case when count(1) > 10 then 'more than 10'

when count(1) between 5 and 10 then 'between 5 and 10'

else 'fewer than 5'

end as category

from rental

GROUP by customer\_id)

select category as rental\_count\_category

, count(\*) as customer\_count

from cte

group by category

order by customer\_count;