QUERIES-OUTPUTS

QUESTION 1- USE SELECT

```
1  create database Ecommerce;
2  use Ecommerce;
3
4  select * from orders
5
6
7
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_deliv
136cce7faa42fdb2cefd53fdc79a6098	ed0271e0b7da060a393796590e7b737a	invoiced	11-04-2017 12:22	13-04-2017 13:25	
6514b8ad8028c9f2cc2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	16-05-2017 13:10	16-05-2017 13:22	22-05-2017
76c6e866289321a7c93b82b54852dc33	f54a9f0e6b351c431402b8461ea51999	delivered	23-01-2017 18:29	25-01-2017 02:50	26-01-2017
e69bfb5eb88e0ed6a785585b27e16dbf	31ad1d1b63eb9962463f764d4e6e0c9d	delivered	29-07-2017 11:55	29-07-2017 12:05	10-08-2017
e6ce16cb79ec1d90b1da9085a6118aeb	494dded5b201313c64ed7f100595b95c	delivered	16-05-2017 19:41	16-05-2017 19:50	18-05-2017
34513ce0c4fab462a55830c0989c7edb	7711cf624183d843aafe81855097bc37	delivered	13-07-2017 19:58	13-07-2017 20:10	14-07-2017
82566a660a982b15fb86e904c8d32918	d3e3b74c766bc6214e0c830b17ee2341	delivered	07-06-2018 10:06	09-06-2018 03:13	11-06-2018
5ff96c15d0b717ac6ad1f3d77225a350	19402a48fe860416adf93348aba37740	delivered	25-07-2018 17:44	25-07-2018 17:55	26-07-2018
432aaf21d85167c2c86ec9448c4e42cc	3df704f53d3f1d4818840b34ec672a9f	delivered	01-03-2018 14:14	01-03-2018 15:10	02-03-2018
dcb36b511fcac050b97cd5c05de84dc3	3b6828a50ffe546942b7a473d70ac0fc	delivered	07-06-2018 19:03	12-06-2018 23:31	11-06-2018
403b97836b0c04a622354cf531062e5f	738b086814c6fcc74b8cc583f8516ee3	delivered	02-01-2018 19:00	02-01-2018 19:09	03-01-2018
116f0b09343b49556bbad5f35bee0cdf	3187789bec990987628d7a9beb4dd6ac	delivered	26-12-2017 23:41	26-12-2017 23:50	28-12-2017
85ce859fd6dc634de8d2f1e290444043	059f7fc5719c7da6cbafe370971a8d70	delivered	21-11-2017 00:03	21-11-2017 00:14	23-11-2017
83018ec114eee8641c97e08f7b4e926f	7f8c8b9c2ae27bf3300f670c3d478be8	delivered	26-10-2017 15:54	26-10-2017 16:08	26-10-2017

QUESTION 1- USE WHERE

```
1  create database Ecommerce;
2  use Ecommerce;
3
4  select order_id,order_status,order_estimated_delivery_date from orders
5  where order_estimated_delivery_date between'01-01-2017 ' and  '30-12-2017'
6  order by 1 asc;
7
8
9
10
11
12
13
14
15
16
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	order_status	order_estimated_delivery_date
00571ded73b3c061925584feab0db425	delivered	12-06-2017 00:00
01a75ac3dd000b7628a684be97c29234	delivered	21-02-2018 00:00
02aec12849744e266394182861265286	delivered	27-09-2017 00:00
0341b60065c49221a27293793f11b117	delivered	24-05-2018 00:00
04040ee654b248cdc512a68ecc83e4cc	delivered	09-04-2018 00:00
0438fc3e115633e11a93c878a1591016	delivered	28-08-2018 00:00
04a59fe46fc5aa341da5378c195ed53c	delivered	19-12-2017 00:00
05387d537bd2e5027aadbfb6a5123e4b	delivered	29-11-2017 00:00
06a6627d9cc91a04e9d146bf65fee0a2	delivered	04-08-2017 00:00

QUESTION 1- USE WHERE ,GROUP BY AND ORDER BY

```sql
create database Ecommerce;
use Ecommerce;

select order_status,count(*),order_estimated_delivery_date from orders
where order_estimated_delivery_date between'01-01-2017 ' and  '30-12-2017'
group by 1,3
order by 1 asc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_status	count(*)	order_estimated_delivery_date
delivered	1	01-06-2017 00:00
delivered	1	01-08-2017 00:00
delivered	4	01-09-2017 00:00
delivered	1	02-01-2018 00:00
delivered	1	02-02-2018 00:00
delivered	1	02-03-2017 00:00
delivered	3	02-04-2018 00:00
delivered	1	02-05-2018 00:00
delivered	3	02-08-2018 00:00
delivered	1	02-10-2017 00:00
delivered	3	03-01-2018 00:00
delivered	1	03-04-2017 00:00
delivered	2	03-04-2018 00:00
delivered	2	03-05-2018 00:00
delivered	1	03-07-2018 00:00
delivered	1	03-08-2017 00:00
delivered	1	03-11-2017 00:00
delivered	4	04-01-2018 00:00

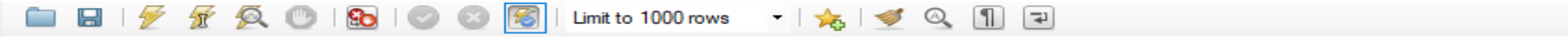
QUESTION 2- USE LEFT JOIN

```
2  use Ecommerce;
3
4  select o.order_id,o.customer_id,orv.review_score from orders o
5  left join order_reviews orv
6  on orv.order_id=o.order_id
7  where review_score is not null
8  order by 1 desc
9
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	customer_id	review_score
c186ebe3937470a2f562a2dc0bc74dd7	20b5aae6a3e31111009f9a7ecc31a232	5
4709741e829775567b92abc42437b461	bd13582471af02ce718568d9cf8bbd93	5

QUESTION 2- USE RIGHT JOIN




```
1  create database Ecommerce;
2  use Ecommerce;
3
4  select o.order_id,o.customer_id,orv.review_score from orders o
5  right join order_reviews orv
6  on orv.order_id=o.order_id
7  order by 1 desc
8
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	customer_id	review_score
c186ebe3937470a2f562a2dc0bc74dd7	20b5aae6a3e31111009f9a7ecc31a232	5
4709741e829775567b92abc42437b461	bd13582471af02ce718568d9cf8bbd93	5
NULL	NULL	5
NULL	NULL	5
NULL	NULL	5
NULL	NULL	5
NULL	NULL	1
NULL	NULL	5
NULL	NULL	5
NULL	NULL	5
NULL	NULL	4
NULL	NULL	5
NULL	NULL	5
NULL	NULL	4
NULL	NULL	4
NULL	NULL	3
NULL	NULL	5
NULL	NULL	2

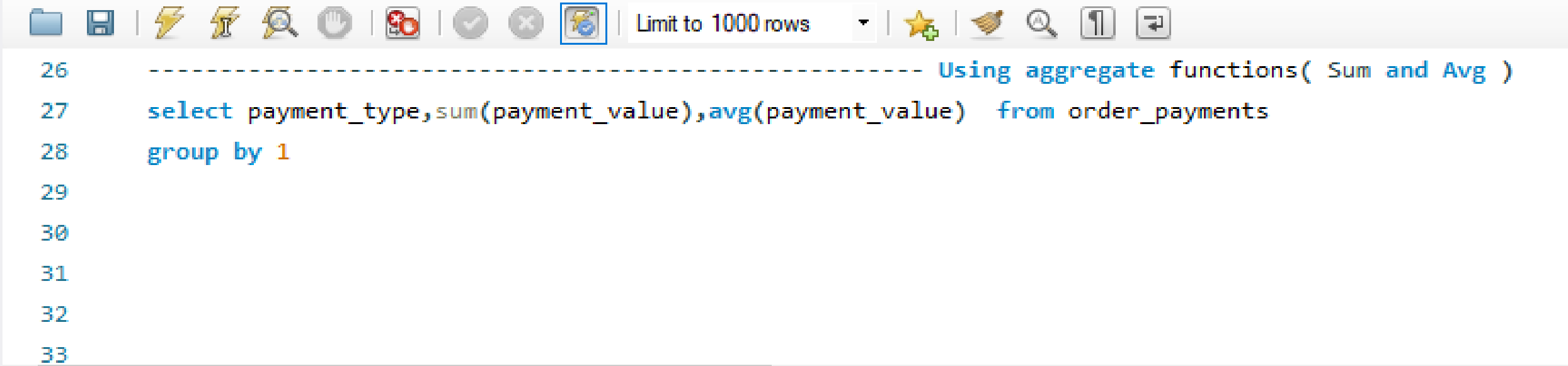
QUESTION 3- USE SUBQURIES

```
29
30      ----------------------------------------------------------Subqueries
31      select payment_type  from order_payments
32      WHERE payment_value=(select max(payment_value) from order_payments)
33
34
35
36
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

payment_type
credit_card

QUESTION 4- USE AGGREGATE FUNCTION SUM AND AVY



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

payment_type	sum(payment_value)	avg(payment_value)
credit_card	74198.07000000002	163.79264900662255
boleto	11659.589999999997	115.44148514851481
voucher	2656.88	83.0275
debit_card	1631.6999999999998	125.5153846153846