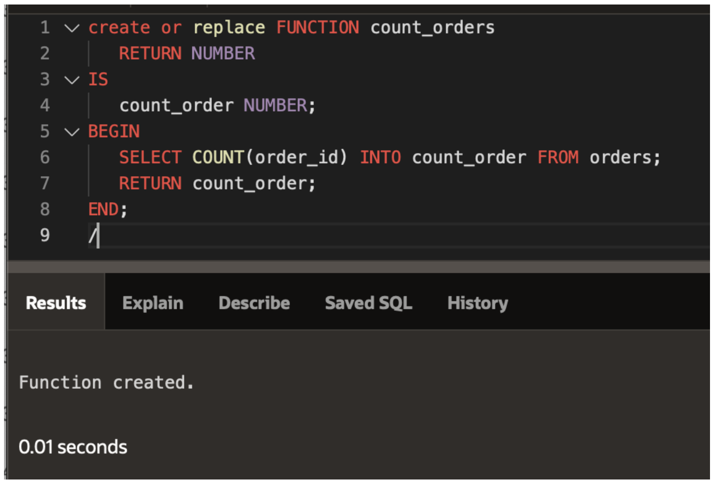
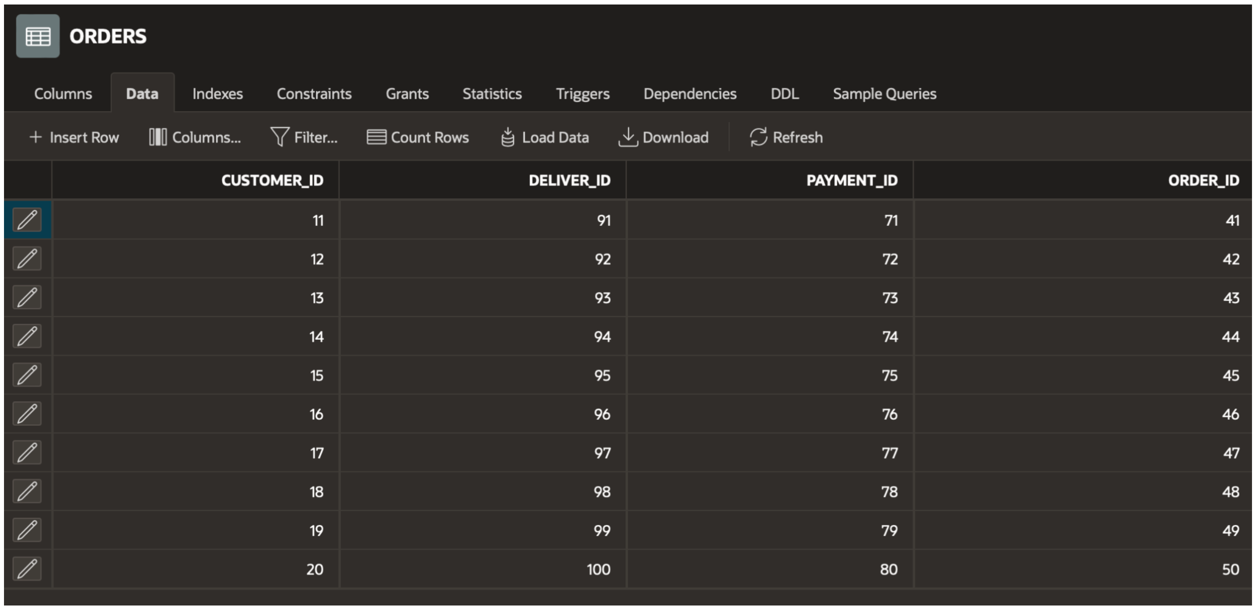
1) Hello everyone. My part in this project is the functional department that counts the number of customer orders for us.We need this in order to find out if we have achieved our goal, which we had to buy. For example, when we want to buy sports equipment for football (let's say we buy all the equipment on the website “sportmaster.com ” and we make a list of what we need to buy) we calculate how many products we have to buy, and after buying things, the function automatically calculates the cost of the number of products we bought. I think this is a very useful feature for convenience.



```
create or replace FUNCTION count_orders
    RETURN NUMBER
IS
    count_order NUMBER;
BEGIN
    SELECT COUNT(order_id) INTO count_order FROM orders;
    RETURN count_order;
END;
/
```

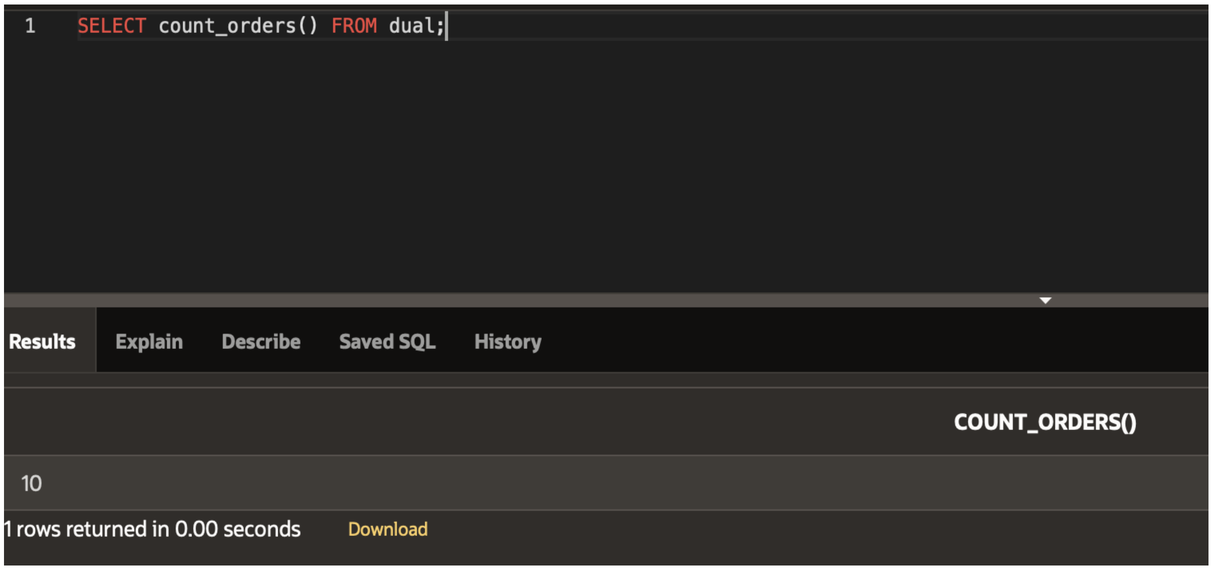
```
Function created.

0.01 seconds
```

Here we have 10 orders (41-50)



CUSTOMER_ID	DELIVER_ID	PAYMENT_ID	ORDER_ID
11	91	71	41
12	92	72	42
13	93	73	43
14	94	74	44
15	95	75	45
16	96	76	46
17	97	77	47
18	98	78	48
19	99	79	49
20	100	80	50

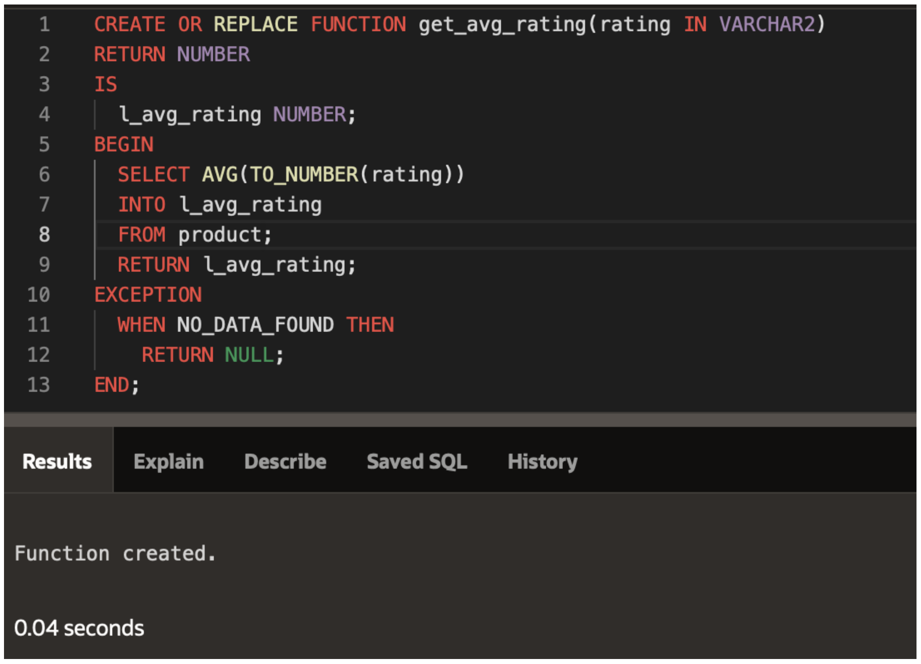
And we can check this by writing:



```
SELECT count_orders() FROM dual;
```

```
COUNT_ORDERS()
10
1 rows returned in 0.00 seconds
```

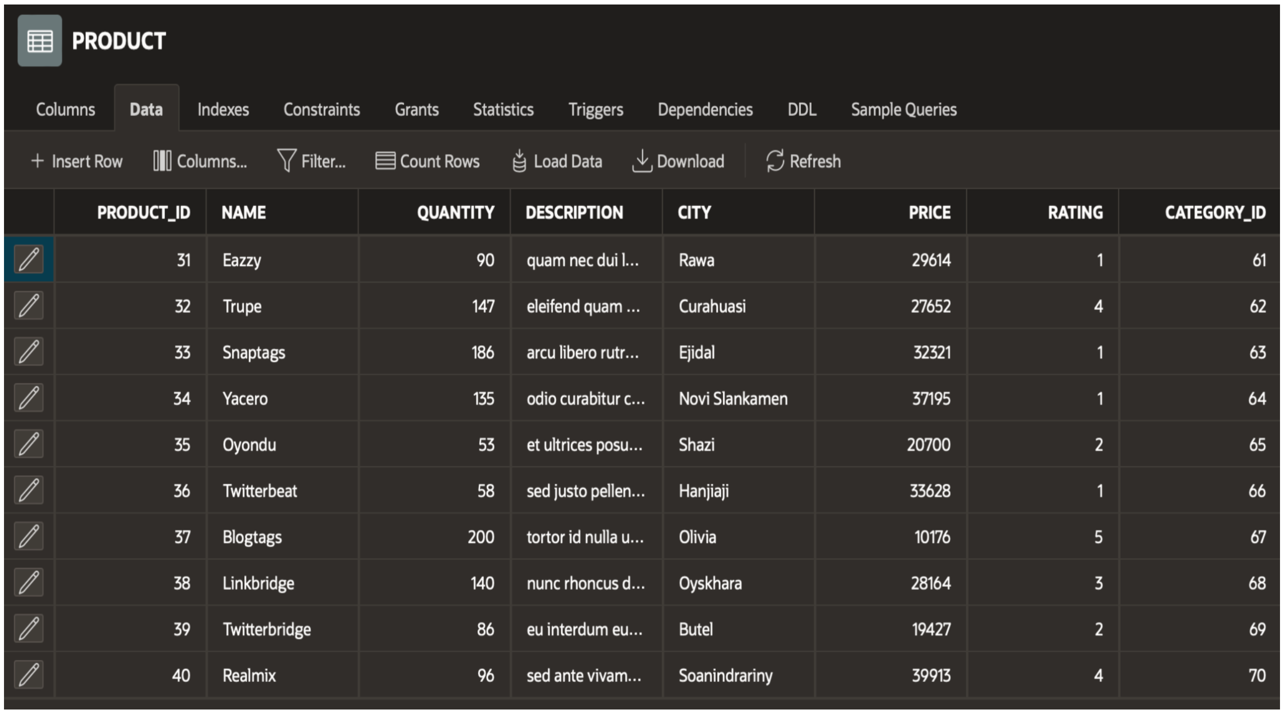
2) In the second function, we provide the average of the product ratings to find out how good our product is.



```
CREATE OR REPLACE FUNCTION get_avg_rating(rating IN VARCHAR2)
RETURN NUMBER
IS
  l_avg_rating NUMBER;
BEGIN
  SELECT AVG(TO_NUMBER(rating))
  INTO l_avg_rating
  FROM product;
  RETURN l_avg_rating;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;
END;
```

```
Function created.

0.04 seconds
```

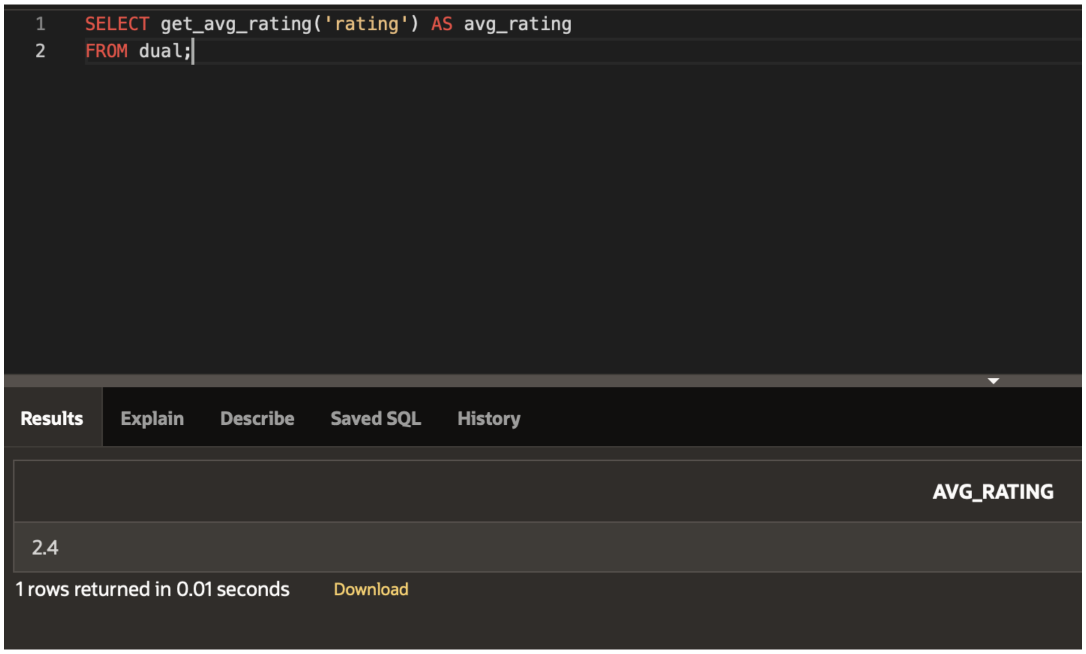
Let's take our “PRODUCT” table



PRODUCT_ID	NAME	QUANTITY	DESCRIPTION	CITY	PRICE	RATING	CATEGORY_ID
31	Eazzy	90	quam nec dui l...	Rawa	29614	1	61
32	Trupe	147	eleifend quam ...	Curahuasi	27652	4	62
33	Snaptags	186	arcu libero rutr...	Ejidal	32321	1	63
34	Yacero	135	odio curabitur c...	Novi Slankamen	37195	1	64
35	Oyondu	53	et ultrices posu...	Shazi	20700	2	65
36	Twitterbeat	58	sed justo pellen...	Hanjiaji	33628	1	66
37	Blogtags	200	tortor id nulla u...	Olivia	10176	5	67
38	Linkbridge	140	nunc rhoncus d...	Oyskhara	28164	3	68
39	Twitterbridge	86	eu interdum eu...	Butel	19427	2	69
40	Realmix	96	sed ante vivam...	Soanindrariny	39913	4	70

(1+4+1+1+2+1+5+3+2+4)/10=2.4

Here we can check the product rating by writing:



```
SELECT get_avg_rating('rating') AS avg_rating
FROM dual;
```

```
AVG_RATING
2.4
1 rows returned in 0.01 seconds     Download
```