

QUERY PRACTICE TASK 2

-- 1. Join (with inner join) together person, personphone, businessentity and phonenumbertype in the persons schema. Return first name, middle name, last name, phone number and the name of the phone number type (home, office, etc.) Order by business entity id descending.

```
select b.businessentityid , p.firstname , p.middlename , p.lastname , p2.phonenumber ,  
p3."name"
```

```
from person p
```

```
inner join personphone p2 on p.businessentityid = p2.businessentityid
```

```
inner join businessentity b on p.businessentityid = b.businessentityid
```

```
inner join phonumbertype p3 on p2.phonumbertypeid = p3.phonumbertypeid
```

```
order by b.businessentityid desc ;
```

businessentityid	firstname	middlename	lastname	phonenumber	name
20777	Crystal	[NULL]	Hu	1 (11) 500 555-0126	Home
20776	Crystal	[NULL]	Zheng	1 (11) 500 555-0171	Home
20775	Crystal	S	He	813-555-0148	Home
20774	Isabella	F	Richardson	910-555-0166	Cell
20773	Crystal	[NULL]	Guo	1 (11) 500 555-0171	Cell
20772	Crystal	J	Liang	1 (11) 500 555-0120	Home
20771	Isabella	[NULL]	Bailey	808-555-0174	Cell
20770	Crystal	J	Gao	1 (11) 500 555-0136	Cell
20769	Crystal	C	Zhu	1 (11) 500 555-0180	Home
20768	Isabella	[NULL]	Murphy	1 (11) 500 555-0117	Home
20767	Crystal	E	Sun	1 (11) 500 555-0119	Cell
20766	Crystal	E	Xu	1 (11) 500 555-0127	Cell
20765	Francis	D	Ruiz	1 (11) 500 555-0190	Cell
20764	Crystal	A	Lu	1 (11) 500 555-0199	Cell
20763	Isabella	[NULL]	Bell	1 (11) 500 555-0157	Home
20762	Crystal	[NULL]	Zhao	1 (11) 500 555-0161	Home
20761	Isabella	L	Morgan	118-555-0127	Home
20760	Isabella	D	Cook	248-555-0151	Home
20759	Isabella	[NULL]	Reed	754-555-0118	Cell
20758	Isabella	[NULL]	Rogers	199-555-0140	Cell
20757	Carol	L	Zhou	1 (11) 500 555-0141	Cell
20756	Crystal	[NULL]	Lin	1 (11) 500 555-0113	Cell
20755	Crystal	W	Wu	1 (11) 500 555-0151	Home

from the results of the query that has been run, we can find out the name of the person and the type of phone number.

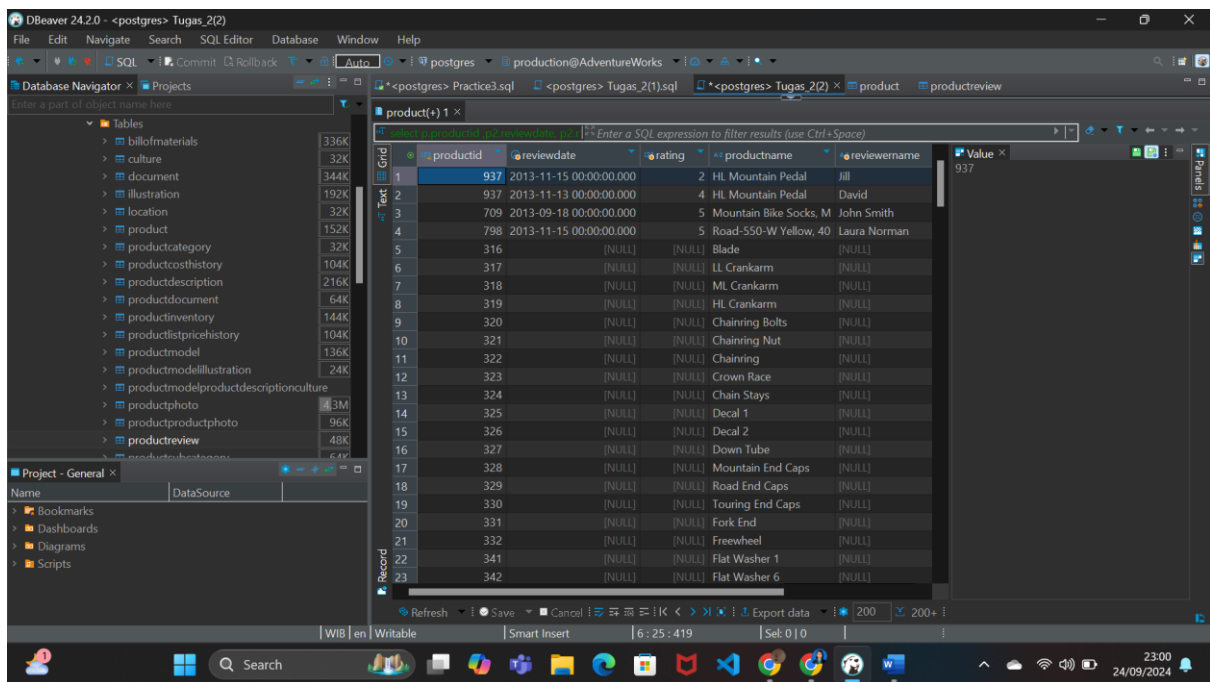
-- 2. Join product and productreview in the schema table. Include every record from product and any reviews they have. Return the product name, review rating and comments. Order by rating in ascending order.

```
select p.productid ,p2.reviewdate, p2.rating, p."name" as productname , p2.reviewname ,  
p2."comments"
```

```
from product p
```

```
left join productreview p2 on p.productid = p2.productid
```

```
order by p2.rating asc ;
```



The screenshot shows the DBeaver 24.2.0 interface with a SQL query executed against a PostgreSQL database. The query is a left join between the 'product' and 'productreview' tables. The results are displayed in a grid with columns: productid, reviewdate, rating, productname, and reviewname. The data shows 23 rows, where the first two rows have reviews and the remaining 21 rows do not (indicated by NULL in the rating and reviewname columns).

productid	reviewdate	rating	productname	reviewname
937	2013-11-15 00:00:00.000	2	HL Mountain Pedal	Jill
937	2013-11-13 00:00:00.000	4	HL Mountain Pedal	David
709	2013-09-18 00:00:00.000	5	Mountain Bike Socks, M	John Smith
798	2013-11-15 00:00:00.000	5	Road-550-W Yellow, 40	Laura Norman
316	[NULL]	[NULL]	Blade	[NULL]
317	[NULL]	[NULL]	LL Crankarm	[NULL]
318	[NULL]	[NULL]	ML Crankarm	[NULL]
319	[NULL]	[NULL]	HL Crankarm	[NULL]
320	[NULL]	[NULL]	Chainring Bolts	[NULL]
321	[NULL]	[NULL]	Chainring Nut	[NULL]
322	[NULL]	[NULL]	Chainring	[NULL]
323	[NULL]	[NULL]	Crown Race	[NULL]
324	[NULL]	[NULL]	Chain Stays	[NULL]
325	[NULL]	[NULL]	Decal 1	[NULL]
326	[NULL]	[NULL]	Decal 2	[NULL]
327	[NULL]	[NULL]	Down Tube	[NULL]
328	[NULL]	[NULL]	Mountain End Caps	[NULL]
329	[NULL]	[NULL]	Road End Caps	[NULL]
330	[NULL]	[NULL]	Touring End Caps	[NULL]
331	[NULL]	[NULL]	Fork End	[NULL]
332	[NULL]	[NULL]	Freewheel	[NULL]
341	[NULL]	[NULL]	Flat Washer 1	[NULL]
342	[NULL]	[NULL]	Flat Washer 6	[NULL]

The results of the query that has been run will show all products, whether there are reviews or not. The results of the query show data on how the customer reviews the product that they have bought, through ratings and comments.

-- 3. Use a right join to combine workorder and product from production schema to bring back all products and any work orders they have. Include the product name and workorder orderqty and scrappedqty fields. Order by productid ascending.

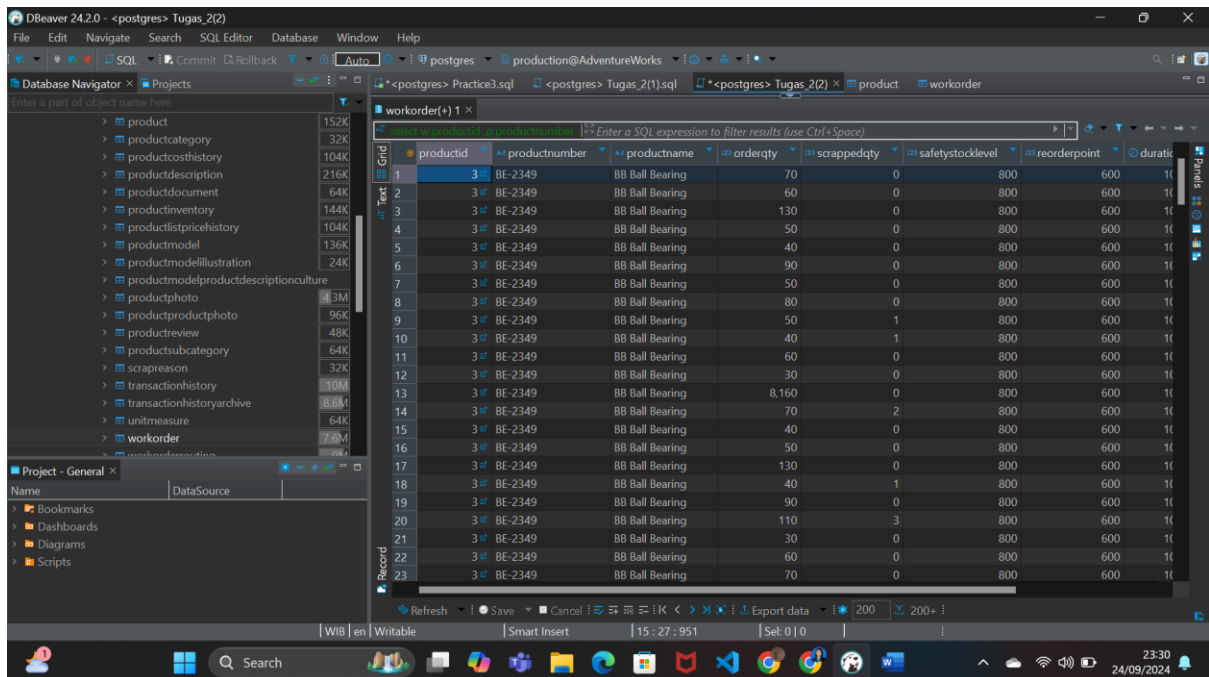
```
select w.productid ,p.productnumber , p."name" as productname , w.orderqty ,  
w.scrappedqty ,
```

```
p.safetystocklevel , p.reorderpoint , w.enddate - w.startdate as duration , w.duedate
```

```
from workorder w
```

```
right join product p on w.productid = p.productid
```

```
order by w.productid asc ;
```



The screenshot shows the DBeaver 24.2.0 interface with a SQL query executed against a PostgreSQL database. The query is a right join between the 'workorder' and 'product' tables. The results are displayed in a grid view, showing 23 records. The columns include productid, productnumber, productname, orderqty, scrappedqty, safetystocklevel, reorderpoint, and duration. The product names are all 'BB Ball Bearing'.

productid	productnumber	productname	orderqty	scrappedqty	safetystocklevel	reorderpoint	duration
1	BE-2349	BB Ball Bearing	70	0	800	600	1
2	BE-2349	BB Ball Bearing	60	0	800	600	1
3	BE-2349	BB Ball Bearing	130	0	800	600	1
4	BE-2349	BB Ball Bearing	50	0	800	600	1
5	BE-2349	BB Ball Bearing	40	0	800	600	1
6	BE-2349	BB Ball Bearing	90	0	800	600	1
7	BE-2349	BB Ball Bearing	50	0	800	600	1
8	BE-2349	BB Ball Bearing	80	0	800	600	1
9	BE-2349	BB Ball Bearing	50	1	800	600	1
10	BE-2349	BB Ball Bearing	40	1	800	600	1
11	BE-2349	BB Ball Bearing	60	0	800	600	1
12	BE-2349	BB Ball Bearing	30	0	800	600	1
13	BE-2349	BB Ball Bearing	8,160	0	800	600	1
14	BE-2349	BB Ball Bearing	70	2	800	600	1
15	BE-2349	BB Ball Bearing	40	0	800	600	1
16	BE-2349	BB Ball Bearing	50	0	800	600	1
17	BE-2349	BB Ball Bearing	130	0	800	600	1
18	BE-2349	BB Ball Bearing	40	1	800	600	1
19	BE-2349	BB Ball Bearing	90	0	800	600	1
20	BE-2349	BB Ball Bearing	110	3	800	600	1
21	BE-2349	BB Ball Bearing	30	0	800	600	1
22	BE-2349	BB Ball Bearing	60	0	800	600	1
23	BE-2349	BB Ball Bearing	70	0	800	600	1