

Ödev Çözümlerim

Soru 1

Invoice tablosunda, tüm değerleri NULL olan kayıtların sayısını bulmanız isteniyor. Bu işlemi tek bir sorgu ile yapmalısınız. Sorguyu yazdıktan sonra, PostgreSQL'in sol alt kısmındaki Row sayısını, SQL sorgunuzda yorum satırında belirtmeniz gerekmektedir.

Cevap 1

“select * from invoice” ifadesi invoice tablosundaki tüm verileri getirir.

“count” fonksiyonu sayıcı görevi görür. “as” ifadesi aliastır, takma ad olarak kullanılır.

where ifadesi ile istediğim koşulları yazdım.

bu yazdığım koşulların “from” ifadesi ile invoice tablosundan gelmesini istedim.

“is null” ifadesi bir değerin null olup olmadığını sorgular.

bu sorgu sonucunda bana 0 satır döndü çünkü bu tabloda tüm verileri null olan bir kayıt yok.

The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 select * from invoice
2 --rows: 0
3 select count(*) as kolon_sayisi from invoice
4 where invoice_id is null
5 and customer_id is null
6 and invoice_date is null
7 and billing_address is null
8 and billing_city is null
9 and billing_city is null
10 and billing_state is null
11 and billing_country is null
12 and billingpostal_code is null
13 and total is null ;
```

The query results are displayed in a table with the following structure:

kolon_sayisi
0

The interface also shows a status bar at the bottom indicating "Total rows: 1" and "Query complete 00:00:00.105".

Soru 2

Koordinasyondaki kişiler, Total değerlerinde bir hata olduğunu belirtiyorlar. Bu değerlerin iki katını görmek ve eski versiyonlarıyla birlikte karşılaştırmak için bir sorgu yazmanız isteniyor. Ayrıca, verilerin daha rahat takip edilebilmesi için, tablonun yeni versiyonuna ait kolona göre küçükten büyüğe sıralama yapılması isteniyor.

Cevap 2

select ifadesi ile istediğim sütunları seçtim ve as ifadesi ile istediğim sütuna takma ad verdim ve from ile bu seçtiğim sütunların geleceği tabloyu belirttim.

where ile totalın null olmadığını kayıtların gelmesini istedim.

“order by” ile istediğim sütunu “asc” ile küçükten büyüğe sıraladım. “asc” ifadesini yazmasam da olurdu çünkü order by kullandığımda default değer asc’dir.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × postgres/postgres@ödev1* ×

postgres/postgres@ödev1

Query Query History Scratch Pad ×

```
1 select invoice_id, customer_id, total as eski_total, (total*2) as yeni_total
2 from invoice
3 where total is not null
4 order by yeni_total asc;
5
```

Data Output Messages Notifications

Showing rows: 1 to 412 Page No: 1 of 1

	invoice_id [PK] integer	customer_id integer	eski_total numeric (10,2)	yeni_total numeric
1	104	38	0.99	1.98
2	174	5	0.99	1.98
3	69	25	0.99	1.98
4	300	40	0.99	1.98
5	216	56	0.99	1.98
6	181	43	0.99	1.98
7	314	57	0.99	1.98
8	111	17	0.99	1.98
9	377	45	0.99	1.98

Total rows: 412 Query complete 00:00:00.120 CRLI

Soru 3

Adres kolonundaki verileri, soldan 3 karakter ve sağdan 4 karakter olarak birleştirmeniz ve "Açık Adres" olarak yazmanız isteniyor. Ayrıca, bu yeni açık adresi 2013 yılı ve 10. ay'a göre filtrelemeniz gerekiyor.

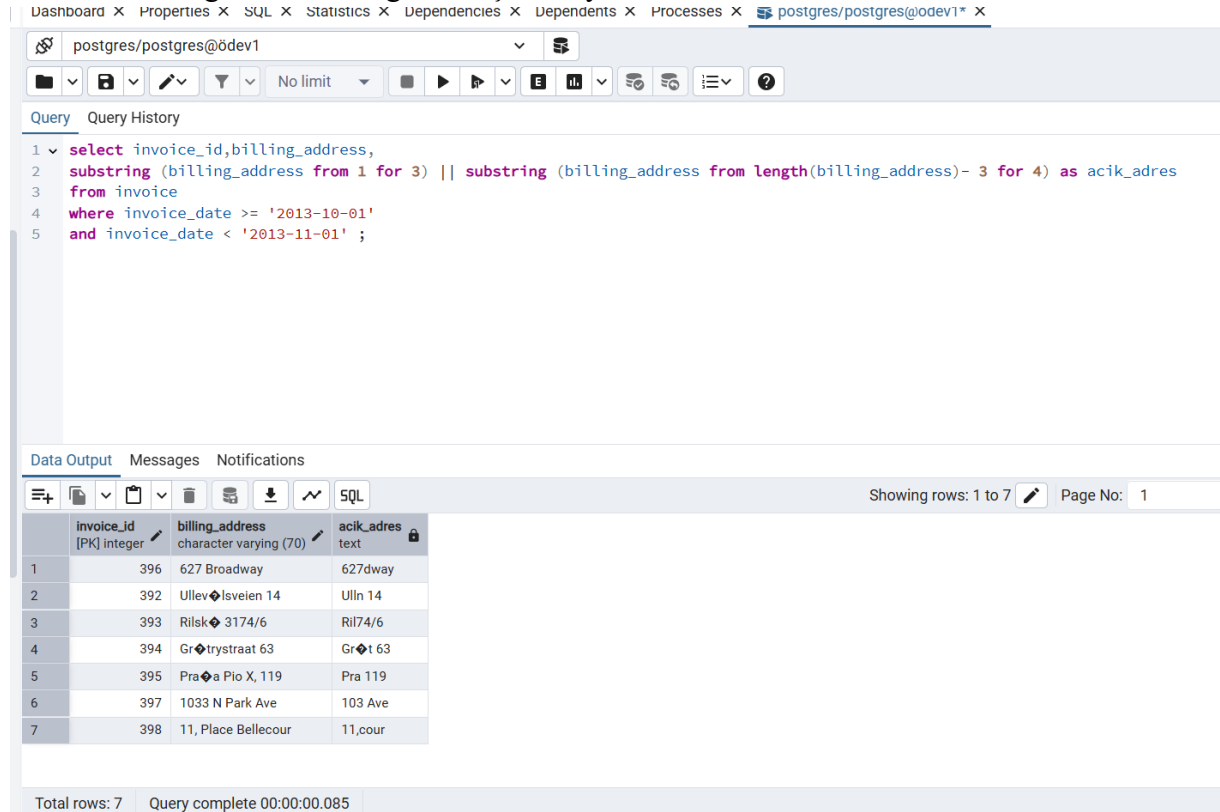
Cevap 3

select ifadesi ile istediğim sütunları seçtim.

“substring” ifadesi ile bu kayıta istediğim indeksten başlayıp istediğim indekse kadar olan değerleri döndürdüm. İlk substring ifadesi soldan 3 karakter almaya yarar. İkinci substring ifadesi length ile billing_addressin karakter uzunluğunu alır ve bundan 3 eksiltir ve bu sayede kaçınıcı karakter başlamam gerektiğini bana verir ve sonrasında sağdan 4 karakter aldım.

” || “ ile bu ikisini birleştirdim

where ile istediğim tarih aralığının koşulunu yazdım.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 select invoice_id, billing_address,
2 substring (billing_address from 1 for 3) || substring (billing_address from length(billing_address)- 3 for 4) as acik_adres
3 from invoice
4 where invoice_date >= '2013-10-01'
5 and invoice_date < '2013-11-01' ;
```

The results are displayed in a table with the following columns: invoice_id, billing_address, and acik_adres. The table contains 7 rows of data.

invoice_id	billing_address	acik_adres
396	627 Broadway	627dway
392	Ullevølsveien 14	Ulln 14
393	Rilsk 3174/6	Ril74/6
394	Grøtstraat 63	Grøt 63
395	Praa Pio X, 119	Pra 119
397	1033 N Park Ave	103 Ave
398	11, Place Bellecour	11,cour

Total rows: 7 Query complete 00:00:00.085

Bu ise farklı bir yoldan çözümü eğitime dahil olduğu için bunu da kullanmak istedim. yine select ile istediğim sütunları seçtim ve bu sefer “concat” kullandım. Bu da birleştirmeye yarıyor. İki ifade yazdım birisi soldan birisi sağdan karakterler alıyor ve concat bunları birleştiriyor. Üstteki çözümde yaptığım gibi yine where ile tarih sorgusu yaptım.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × postgres/postgres@ödev1 ×

postgres/postgres@ödev1

Query Query History

```
1 select invoice_id,billing_address,
2 concat(
3 left(billing_address,3),
4 right(billing_address,4)
5 ) as acik_adres
6 from invoice
7 where invoice_date >= '2013-10-01'
8 and invoice_date < '2013-11-01' ;
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1 of 1

	invoice_id [PK] integer	billing_address character varying (70)	acik_adres text
1	396	627 Broadway	627dway
2	392	Ullevøysveien 14	Ulln 14
3	393	Rilsk 3174/6	Ril74/6
4	394	Grøtstrøst 63	Grøt 63
5	395	Praa Plo X, 119	Pra 119
6	397	1033 N Park Ave	103 Ave
7	398	11, Place Bellecour	11,cour

Total rows: 7 Query complete 00:00:00.061 CRLF