



كلاس حل تمرين

٩٩/٠٨/١٩

آشنایی با View

۱- یک View تعریف کنید که تاریخ معاملات مربوط به دوچرخه را در خروجی نمایش دهد. پیوند جداول را طوری انجام دهید که فیلدهای زیر را در View داشته باشیم:

ProductName,

ProductNumber,

ProductCategory,

ProductSubCategory,

TransactionID,

ReferenceOrderID,

ReferenceOrderLineID,

TransactionDate,

TransactionType,

Quantity,

ActualCost

پاسخ

```
CREATE VIEW v_Product_TransactionHistory AS
SELECT p.Name AS ProductName,
       p.ProductNumber,
       pc.Name AS ProductCategory,
       ps.Name AS ProductSubCategory,
       th.TransactionID,
       th.ReferenceOrderID,
       th.ReferenceOrderLineID,
       th.TransactionDate,
       th.TransactionType,
       th.Quantity,
       th.ActualCost
FROM Production.TransactionHistory th
     INNER JOIN Production.Product p
       ON th.ProductID = p.ProductID
     INNER JOIN Production.ProductSubcategory ps
       ON ps.ProductSubcategoryID = p.ProductSubcategoryID
     INNER JOIN Production.ProductCategory pc
       ON pc.ProductCategoryID = ps.ProductCategoryID
WHERE pc.Name = 'Bikes';
```

نمونه خروجی

Data Output											Explain	Messages	Notifications
	productname character varying (50)	productnumber character varying (25)	productcategory character varying (50)	productsubcategory character varying (50)	transactionid integer	referenceorderid integer	referenceorderid integer	transactiondate timestamp without time zone	transactiontype character (1)	quantity integer			
1	Mountain-200 Black, 46	BK-M68B-46	Bikes	Mountain Bikes	100000	41590	0	2013-07-31 00:00:00	W	2			
2	Road-250 Black, 48	BK-R89B-48	Bikes	Road Bikes	100001	41591	0	2013-07-31 00:00:00	W	1			
3	Road-550-W Yellow, 38	BK-R64Y-38	Bikes	Road Bikes	100002	41592	0	2013-07-31 00:00:00	W	1			
4	Road-550-W Yellow, 40	BK-R64Y-40	Bikes	Road Bikes	100003	41593	0	2013-07-31 00:00:00	W	1			
5	Road-550-W Yellow, 42	BK-R64Y-42	Bikes	Road Bikes	100004	41594	0	2013-07-31 00:00:00	W	1			
6	Road-550-W Yellow, 44	BK-R64Y-44	Bikes	Road Bikes	100005	41595	0	2013-07-31 00:00:00	W	1			
7	Road-550-W Yellow, 48	BK-R64Y-48	Bikes	Road Bikes	100006	41596	0	2013-07-31 00:00:00	W	1			
8	Touring-1000 Yellow, 46	BK-T79Y-46	Bikes	Touring Bikes	100007	41597	0	2013-07-31 00:00:00	W	1			
9	Touring-1000 Yellow, 50	BK-T79Y-50	Bikes	Touring Bikes	100008	41598	0	2013-07-31 00:00:00	W	1			
10	Touring-1000 Blue, 46	BK-T79U-46	Bikes	Touring Bikes	100009	41599	0	2013-07-31 00:00:00	W	1			
11	Touring-1000 Blue, 54	BK-T79U-54	Bikes	Touring Bikes	100010	41600	0	2013-07-31 00:00:00	W	1			
12	Touring-2000 Blue, 54	BK-T44U-54	Bikes	Touring Bikes	100011	41601	0	2013-07-31 00:00:00	W	1			
13	Road-350-W Yellow, 40	BK-R79Y-40	Bikes	Road Bikes	100012	41602	0	2013-07-31 00:00:00	W	1			

آشنایی با View

2- (سوال تشویقی اختیاری) یک View طراحی کنید که آمار خرید مشتری را به تفکیک سال و ماه نشان دهد، که شامل تعداد کالا و جمع ارزش آنها است. همچنین در هر سطر نشان دهد این مشتری در آن سال و ماه از کدام category و از کدام subcategory تعداد کالای بیشتری خریده است، همچنین در هر سطر نشان دهد برای این مشتری در آن سال و ماه از کدام category و از کدام subcategory مجموعاً ارزش کالاهای خریداری شده بیشتر بوده است. یک کد مشتری دلخواه را در متن دستور خود بگذارید طوری که ستونهای View مقدار داشته باشند. خروجی شامل فیلدهای CustomerID و فیلدهایی با نام MaxProdCount و MaxProdValue است که مقادیر تجمیعی فوق را نشان دهند. نام فیلدهای بعدی را نیز MaxCatByCount, MaxSubcatByValue, MaxCatByValue, MaxSubcatByCount بگذارید که داخل هر کدام نام Category یا Subcategory که حائز بیشترین تعداد یا ارزش برای آن ماه و آن مشتری شده ثبت می شود.

پاسخ

```
CREATE VIEW Q2 AS(  
  with all_details(customerID, productID, quantity, value, prodCatId, prodSubCatID, year, month) as  
  (  
    select customer.customerid,  
           details.productid,  
           details.orderqty,  
           (details.unitprice * details.orderqty) as value,  
           productCat.productcategoryid,  
           productSubCat.productssubcategoryid,  
           extract(year from header.orderdate),  
           extract(month from header.orderdate)  
    from sales.salesorderheader as header  
         inner join sales.customer customer using (customerid)  
         inner join sales.salesorderdetail details using (salesorderid)  
         inner join production.product product using (productid)  
         inner join production.productssubcategory productSubCat using (productssubcategoryid)  
         inner join production.productcategory productCat using (productcategoryid)  
  ),  
  customer_category_details(customerID, categoryID, sumQty, sumValue, year, month) as  
  (  
    select customerID, prodCatId, sum(quantity), sum(value), year, month  
    from all_details  
    group by customerID, prodCatId, year, month  
  ),  
)
```

پاسخ

```
customer_category_max_count(customerID, maxCount, year, month) as
(
    select customerID,
           max(sumQty),
           year,
           month
    from customer_category_details
    group by customerID, year, month
),
customer_category_max_count_result(customerID, categoryID, year, month) as
(
    select customerID,
           min(categoryID),
           year,
           month
    from customer_category_details
         inner join customer_category_max_count using (customerID, year, month)
    where customer_category_details.sumQty = customer_category_max_count.maxCount
    group by customerID, year, month
),
customer_category_max_value(customerID, max_value, year, month) as
(
    select customerID,
           max(sumValue),
           year,
           month
    from customer_category_details
    group by customerID, year, month
),
```

پاسخ

```
customer_category_max_value_result(customerID, categoryID, year, month) as
(
    select customerID,
           min(categoryID),
           year,
           month
    from customer_category_details
         inner join customer_category_max_value using (customerID, year, month)
    where customer_category_details.sumValue = customer_category_max_value.maxValue
    group by customerID, year, month
),
customer_subcat_details(customerID, subCatID, sumQty, sumValue, year, month) as
(
    select customerID, prodSubCatID, sum(quantity), sum(value), year, month
    from all_details
    group by customerID, prodSubCatID, year, month
),
customer_subcategory_max_count(customerID, maxCount, year, month) as
(
    select customerID,
           max(sumQty),
           year,
           month
    from customer_subcat_details
    group by customerID, year, month
),
```


پاسخ

```
customer_subcategory_max_value(customerID, maxValue, year, month) as
(
    select customerID,
           max(sumValue),
           year,
           month
    from customer_subcat_details
    group by customerID, year, month
),
customer_subcategory_max_value_result(customerID, subCategoryID, year, month) as
(
    select customerID,
           min(subCatID),
           year,
           month
    from customer_subcat_details
         inner join customer_subcategory_max_value using (customerID, year, month)
    where customer_subcat_details.sumValue = customer_subcategory_max_value.maxValue
    group by customerID, year, month
),
customer_product_details(customerID, sumProductQty, sumProductValues, year, month) as (
    select customerID,
           sum(quantity),
           sum(value),
           year,
           month
    from all_details
    group by customerID, year, month
),
```

پاسخ

```
report(customerID, MaxProdCount, maxProdValue, year, month, MaxCatCountID, MaxCatValueID, MaxSubCatCountID,
        MaxSubCatValueID) as
(
    select customer_product_details.customerID                as customerID,
           customer_product_details.sumProductQty            as MaxProcCount,
           customer_product_details.sumProductValues          as maxProdValue,
           customer_product_details.year                      as year,
           customer_product_details.month                     as month,
           customer_category_max_count_result.categoryID       as MaxCatCountID,
           customer_category_max_value_result.categoryID       as MaxCatValueID,
           customer_subcategory_max_count_result.subCategoryID as MaxSubCatCountID,
           customer_subcategory_max_value_result.subCategoryID as MaxSubCatValueID
    from customer_product_details
         inner join customer_category_max_count_result using (customerID, year, month)
         inner join customer_category_max_value_result using (customerID, year, month)
         inner join customer_subcategory_max_count_result using (customerID, year, month)
         inner join customer_subcategory_max_value_result using (customerID, year, month)
)
select customerID,
       year,
       month,
       MaxProdCount,
       maxProdValue,
       cat1.name      as MaxCatByCount,
       cat2.name      as MaxCatByValue,
       subcat1.name   as MaxSubCatByCount,
       subcat2.name   as MaxSubCatByValue
from report
     inner join production.productcategory cat1 on cat1.productcategoryid = MaxCatCountID
     inner join production.productcategory cat2 on cat2.productcategoryid = MaxCatValueID
     inner join production.productssubcategory subcat1 on subcat1.productssubcategoryid = MaxSubCatCountID
     inner join production.productssubcategory subcat2 on subcat2.productssubcategoryid = MaxSubCatValueID
where customerID = 29825
order by year, month
);
```





نمونه خروجی

Data Output	Explain	Messages	Notifications									
customerid integer	year double precision	month double precision	maxprodcount bigint	maxprodvalue numeric	maxcatbycount character varying (50)	maxcatbyvalue character varying (50)	maxsubcatbycount character varying (50)	maxsubcatbyvalue character varying (50)				
1	29825	2011	5	26	20565.6206	Clothing	Bikes	Mountain Bikes				Mountain Bikes
2	29825	2011	8	39	18881.9746	Clothing	Bikes	Socks				Mountain Bikes
3	29825	2011	12	44	73261.1513	Bikes	Bikes	Mountain Bikes				Mountain Bikes
4	29825	2012	2	27	18031.6005	Clothing	Bikes	Mountain Bikes				Mountain Bikes
5	29825	2012	5	149	37182.6841	Clothing	Bikes	Gloves				Mountain Bikes
6	29825	2012	8	108	33134.0092	Clothing	Bikes	Mountain Bikes				Mountain Bikes
7	29825	2012	11	79	35989.7381	Clothing	Bikes	Mountain Bikes				Mountain Bikes
8	29825	2013	2	119	32230.2682	Clothing	Bikes	Mountain Bikes				Mountain Bikes
9	29825	2013	5	202	36828.4322	Clothing	Bikes	Jerseys				Mountain Bikes
10	29825	2013	8	155	48197.584	Clothing	Bikes	Mountain Bikes				Mountain Bikes
11	29825	2013	11	93	29140.800	Bikes	Bikes	Mountain Bikes				Mountain Bikes
12	29825	2014	3	166	39459.624	Clothing	Bikes	Mountain Bikes				Mountain Bikes

آشنایی با join

۳- دستوری بنویسید که تمام محصولات را به همراه SalesOrderID نمایش دهد حتی اگر هرگز سفارشی برای آن محصول ثبت نشده باشد.

```
SELECT SalesOrderID, P.ProductID, P.Name
FROM Production.Product AS P
LEFT OUTER JOIN Sales.SalesOrderDetail AS SOD
ON P.ProductID = SOD.ProductID
```

	Data Output	Explain	Messages	Notifications
	 salesorderid integer	 productid integer	 name character varying (50)	
1	43659	776	Mountain-100 Black, 42	
2	43659	777	Mountain-100 Black, 44	
3	43659	778	Mountain-100 Black, 48	
4	43659	771	Mountain-100 Silver, 38	
5	43659	772	Mountain-100 Silver, 42	
6	43659	773	Mountain-100 Silver, 44	
7	43659	774	Mountain-100 Silver, 48	
8	43659	714	Long-Sleeve Logo Jersey, M	
9	43659	716	Long-Sleeve Logo Jersey, XL	
10	43659	709	Mountain Bike Socks, M	
11	43659	712	AWC Logo Cap	
12	43659	711	Sport-100 Helmet, Blue	
13	43660	762	Road-650 Red, 44	
14	43660	758	Road-450 Red, 52	

✓ Successfully run. Total query runtime: 219 msec. 121555 rows affected.

آشنایی با join

۴- دستور نوشته شده در قسمت اول را تغییر دهید تا فقط کالاهایی که سفارش داده نشده اند ، نمایش داده شوند.

```
SELECT SalesOrderID, P.ProductID, P.Name
FROM Production.Product AS P
LEFT OUTER JOIN Sales.SalesOrderDetail AS SOD
ON P.ProductID = SOD.ProductID
WHERE SalesOrderID IS NULL
```





	Data Output	Explain	Messages	Notifications
	salesorderid integer		productid integer	name character varying (50)
1	[null]		497	Pinch Bolt
2	[null]		452	Lock Nut 2
3	[null]		496	Paint - Yellow
4	[null]		455	Lock Nut 3
5	[null]		384	Hex Nut 12
6	[null]		399	Head Tube
7	[null]		437	Thin-Jam Lock Nut 11
8	[null]		518	ML Road Seat Assembly
9	[null]		397	Hex Nut 19
10	[null]		486	Metal Sheet 5
11	[null]		330	Touring End Caps
12	[null]		847	Headlights - Dual-Beam
13	[null]		359	Thin-Jam Hex Nut 9
14	[null]		405	External Lock Washer 9

✓ Successfully run. Total query runtime: 94 msec. 238 rows affected.

آشنایی با join

۵- دستوری بنویسید که تمام ردیف های جدول Sales.SalesPerson به جدول Sales.SalesOrderHeader راجع بر اساس SalesOrderID پیوند دهد، حتی اگر سفارشی مطابقت نداشته باشد. ستون های SalesYTD و SalesPersonID را در نتایج بیاورید.

```
SELECT SalesOrderID, SalesPersonID, SalesYTD
FROM Sales.SalesPerson AS SP
LEFT OUTER JOIN Sales.SalesOrderHeader AS SOH
ON SP.BusinessEntityID = SOH.SalesPersonID
```







	Data Output	Explain	Messages	Notifications
	 salesorderid integer	 salespersonid integer	 salesytd numeric	
1	43659	279	2315185.611	
2	43660	279	2315185.611	
3	43661	282	2604540.7172	
4	43662	282	2604540.7172	
5	43663	276	4251368.5497	
6	43664	280	1352577.1325	
7	43665	283	1573012.9383	
8	43666	276	4251368.5497	
9	43667	277	3189418.3662	
10	43668	282	2604540.7172	
11	43669	283	1573012.9383	
12	43670	275	3763178.1787	
13	43671	283	1573012.9383	
14	43672	282	2604540.7172	

✓ Successfully run. Total query runtime: 1 secs 344 msec. 3806 rows affected.

آشنایی با join

۶- دستور نوشته شده قسمت ۵ را تغییر دهید تا نام فروشنده از جدول Person.Person نیز نشان داده شود.

```
SELECT SalesOrderID,  
       SalesPersonID,  
       SalesYTD,  
       FirstName,  
       MiddleName,  
       LastName  
FROM Sales.SalesPerson AS SP  
LEFT OUTER JOIN Sales.SalesOrderHeader AS SOH  
ON SP.BusinessEntityID = SOH.SalesPersonID  
LEFT OUTER JOIN Person.Person AS P  
ON P.BusinessEntityID = SP.BusinessEntityID;
```

Data Output		Explain	Messages	Notifications		
	 salesorderid integer	 salespersonid integer	 salesytd numeric	 firstname character varying (50)	 middlename character varying (50)	 lastname character varying (50)
1	43659	279	2315185.611	[null]	[null]	[null]
2	43660	279	2315185.611	[null]	[null]	[null]
3	43661	282	2604540.7172	[null]	[null]	[null]
4	43662	282	2604540.7172	[null]	[null]	[null]
5	43663	276	4251368.5497	[null]	[null]	[null]
6	43664	280	1352577.1325	[null]	[null]	[null]
7	43665	283	1573012.9383	[null]	[null]	[null]
8	43666	276	4251368.5497	[null]	[null]	[null]
9	43667	277	3189418.3662	[null]	[null]	[null]
10	43668	282	2604540.7172	[null]	[null]	[null]
11	43669	283	1573012.9383	[null]	[null]	[null]
12	43670	275	3763178.1787	[null]	[null]	[null]
13	43671	283	1573012.9383	[null]	[null]	[null]
14	43672	282	2604540.7172	[null]	[null]	[null]

آشنایی با join

۷- جدول Sales.SalesOrderHeader شامل کلیدهای خارجی به جداول Sales.CurrencyRate و Purchasing.ShipMethod است. برای پیوند سه جدول، یک دستور بنویسید و مطمئن شوید که شامل همه ردیف های Sales.SalesOrderHeader باشد حتی اگر منطق نشده باشند. (ستون های SalesOrderID ، AverageRate ، CurrencyRateID و ShipBase قسمتی از خروجی است.)

```
SELECT CR.CurrencyRateID,  
       CR.AverageRate,  
       SM.ShipBase,  
       SalesOrderID  
FROM Sales.SalesOrderHeader AS SOH  
LEFT OUTER JOIN Sales.CurrencyRate AS CR  
ON SOH.CurrencyRateID = CR.CurrencyRateID  
LEFT OUTER JOIN Purchasing.ShipMethod AS SM  
ON SOH.ShipMethodID = SM.ShipMethodID
```

	Data Output	Explain	Messages	Notifications
	currencyrateid integer	averagerate numeric	shipbase numeric	salesorderid integer
1	[null]	[null]	8.99	43659
2	[null]	[null]	8.99	43660
3	4	1.4641	8.99	43661
4	4	1.4641	8.99	43662
5	[null]	[null]	8.99	43663
6	[null]	[null]	8.99	43664
7	[null]	[null]	8.99	43665
8	[null]	[null]	8.99	43666
9	[null]	[null]	8.99	43667
10	4	1.4641	8.99	43668
11	[null]	[null]	8.99	43669
12	[null]	[null]	8.99	43670
13	[null]	[null]	8.99	43671
14	4	1.4641	8.99	43672

✓ Successfully run. Total query runtime: 730 msec. 31465 rows affected.

آشنایی با join

۸- دستوری بنویسید که ستون BusinessEntityID را از جدول Sales.SalesPerson به همراه هر ProductID از جدول Production.Product برگرداند.

```
SELECT SP.BusinessEntityID, P.ProductID
FROM Sales.SalesPerson AS SP
      CROSS JOIN Production.Product AS P;
```

OR

```
SELECT SP.BusinessEntityID, P.ProductID
FROM Sales.SalesPerson , Production.Product;
```

Data Output	Explain	Messages	Notifications
	businessentityid integer	productid integer	
1	274	1	
2	275	1	
3	276	1	
4	277	1	
5	278	1	
6	279	1	
7	280	1	
8	281	1	
9	282	1	
10	283	1	
11	284	1	
12	285	1	
13	286	1	
14	287	1	

✓ Successfully run. Total query runtime: 91 msec. 8568 rows affected.

آشنایی با join

۹- دستوری بنویسید که نام مشتریان به همراه نام کالاهای خریداری شده توسط آنها را نمایش دهد. (نکته: برای نوشتن این پرسش چهار جدول لازم است!)

```
SELECT soh.CustomerID, Prod.Name
FROM Sales.Customer AS C
INNER JOIN Sales.SalesOrderHeader AS SOH
ON C.CustomerID = SOH.CustomerID
INNER JOIN Sales.SalesOrderDetail AS SOD
ON SOH.SalesOrderID = SOD.SalesOrderID
INNER JOIN Production.Product AS Prod
ON SOD.ProductID = Prod.ProductID
```




	Data Output	Explain	Messages	Notifications
	customerid integer		name character varying (50)	
1	29825		Mountain-100 Black, 42	
2	29825		Mountain-100 Black, 44	
3	29825		Mountain-100 Black, 48	
4	29825		Mountain-100 Silver, 38	
5	29825		Mountain-100 Silver, 42	
6	29825		Mountain-100 Silver, 44	
7	29825		Mountain-100 Silver, 48	
8	29825		Long-Sleeve Logo Jersey, M	
9	29825		Long-Sleeve Logo Jersey, XL	
10	29825		Mountain Bike Socks, M	
11	29825		AWC Logo Cap	
12	29825		Sport-100 Helmet, Blue	
13	29672		Road-650 Red, 44	
14	29672		Road-450 Red, 52	

✓ Successfully run. Total query runtime: 241 msec. 121317 rows affected.

آشنایی با index

۱۰- با جستجوی اینترنتی، دستوری بنویسید که آمار تعداد رکورد کلیه جداول AW را نشان دهد.

```
SELECT schemaname,relname,n_live_tup
FROM pg_stat_user_tables
ORDER BY n_live_tup DESC;
```

	 schemaname name	 relname name	 n_live_tup bigint
1	sales	salesorderdetail	121317
2	production	transactionhistory	113443
3	production	transactionhistoryarchive	89253
4	production	workorder	72591
5	production	workorderrouting	67131
6	sales	salesorderheader	31465
7	sales	salesorderheadersalesreason	27647
8	sales	customer	19820
9	person	address	19614
10	sales	creditcard	19118
11	sales	personcreditcard	19118
12	sales	currencyrate	13532
13	purchasing	purchaseorderdetail	8845
14	purchasing	purchaseorderheader	4012
15	production	billofmaterials	2679

✓ Successfully run. Total query runtime: 123 msec. 70 rows affected.

آشنایی با index

* برای مراحل بعد، بزرگترین جدول از نظر تعداد رکورد را انتخاب کنید.

۱۱- براساس روش مطرح شده در درس، دستوری بنویسید که مشخص کند کدام فیلدهای این جدول ایندکس دارند و چه دستوری برای ساخت آن ایندکس استفاده شده است.

```
SELECT
    indexname,
    indexdef
FROM
    pg_indexes
WHERE
    tablename = 'salesorderdetail';
```

Data Output Explain Messages Notifications

	Indexname name	Indexdef text
1	PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID	CREATE UNIQUE INDEX "PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID" ON sales.salesorderdetail USING btree (salesorderid, salesorderdetailid)

آشنایی با index

از بین فیلدهایی که روی آنها ایندکسی وجود ندارد یکی را با توجه به سوالات زیر انتخاب کنید (مثلا field1). فعلا ایندکسی نسازید (یا اگر بطور تستی ساخته‌اید drop کنید)، و دستوراتی بنویسید که:

۱۲ - رکورد(های) با شرط خاصی از نوع تساوی یعنی field1=myvalue1 را برگرداند.

```
EXPLAIN
  SELECT *
  FROM sales.salesorderdetail
 WHERE orderqty = 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN		
	text		
1	Seq Scan on salesorderdetail (cost=0.00..2955.46 rows=14364 width=64)		
2	Filter: (orderqty = 2)		

آشنایی با index

از بین فیلدهایی که روی آنها ایندکسی وجود ندارد یکی را با توجه به سوالات زیر انتخاب کنید (مثلا field1). فعلا ایندکسی نسازید (یا اگر بطور تستی ساخته‌اید drop کنید)، و دستوراتی بنویسید که:

۱۳- رکورد(های) با شرط خاصی از نوع کوچکتر یعنی `field1 < myvalue2` را برگرداند.

```
EXPLAIN
SELECT *
FROM sales.salesorderdetail
WHERE orderqty < 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Seq Scan on salesorderdetail (cost=0.00..2955.46 rows=74311 width=64)		
2	Filter: (orderqty < 2)		

آشنایی با index

از بین فیلدهایی که روی آنها ایندکسی وجود ندارد یکی را با توجه به سوالات زیر انتخاب کنید (مثلا field1). فعلا ایندکسی نسازید (یا اگر بطور تستی ساخته‌اید drop کنید)، و دستوراتی بنویسید که:

۱۴- رکورد(های) با شرط خاصی از نوع بزرگتر یعنی field1>myvalue2 را برگرداند.

```
EXPLAIN
SELECT *
FROM sales.salesorderdetail
WHERE orderqty > 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN		
	text		
1	Seq Scan on salesorderdetail (cost=0.00..2955.46 rows=32642 width=64)		
2	Filter: (orderqty > 2)		

آشنایی با index

اکنون ایندکسی روی فیلد field1 بسازید و دستورات زیر را بنویسید:

۱۵- دستور سوال (۱۲) را دوباره با استفاده از ایندکس اجرا کنید.

(۱۲ - رکورد(های) با شرط خاصی از نوع تساوی یعنی field1=myvalue1 را برگرداند.)

```
CREATE INDEX ORDER_QTY_INDEX on sales.salesorderdetail (orderqty);
```

```
EXPLAIN
```

```
SELECT *  
FROM sales.salesorderdetail  
WHERE orderqty = 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Bitmap Heap Scan on salesorderdetail (cost=271.74..1890.29 rows=14364 width=64)		
2	Recheck Cond: (orderqty = 2)		
3	-> Bitmap Index Scan on order_qty_index (cost=0.00..268.15 rows=14364 width=0)		
4	Index Cond: (orderqty = 2)		

آشنایی با index

اکنون ایندکسی روی فیلد field1 بسازید و دستورات زیر را بنویسید:

۱۶- دستور سوال (۱۳) را دوباره با استفاده از ایندکس اجرا کنید.

(۱۳- رکورد(های) با شرط خاصی از نوع کوچکتر یعنی field1<myvalue2 را برگرداند.)

```
EXPLAIN
  SELECT *
  FROM sales.salesorderdetail
 WHERE orderqty < 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Seq Scan on salesorderdetail (cost=0.00..2955.46 rows=74311 width=64)		
2	Filter: (orderqty < 2)		

آشنایی با index

اکنون ایندکسی روی فیلد field1 بسازید و دستورات زیر را بنویسید:

۱۶- دستور سوال (۱۴) را دوباره با استفاده از ایندکس اجرا کنید.

(۱۴- رکورد(های) با شرط خاصی از نوع بزرگتر یعنی field1>myvalue2 را برگرداند.)

```
EXPLAIN
SELECT *
FROM sales.salesorderdetail
WHERE orderqty > 2;
```

Data Output	Explain	Messages	Notifications
	QUERY PLAN text		
1	Bitmap Heap Scan on salesorderdetail (cost=617.32..2464.22 rows=32632 width=64)		
2	Recheck Cond: (orderqty > 2)		
3	-> Bitmap Index Scan on order_qty_index (cost=0.00..609.16 rows=32632 width=0)		
4	Index Cond: (orderqty > 2)		