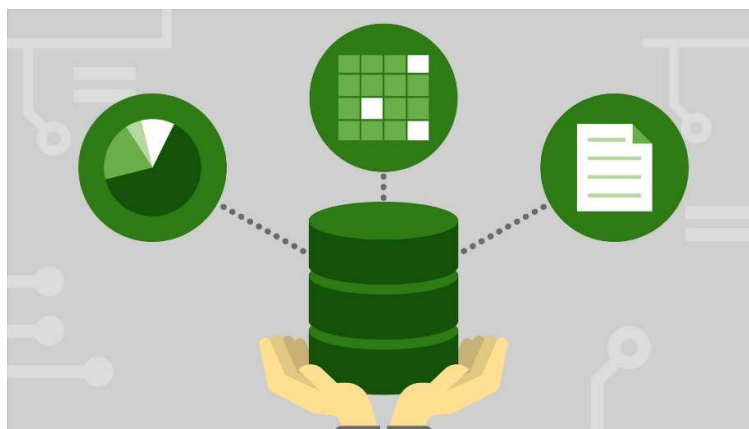


به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پایگاه داده

دستورکار شماره ۵

مهلت تحویل :

۱۴۰۰/۰۸/۳۰

مجتبی بنائی

آنچه خواهید آموخت

هدف اصلی از این تمرین، آشنایی عملی با دستورات پایه SQL بخصوص انواع اتصالات، گروه‌بندی و مرتب‌سازی خواهد بود. اگر قصد کار تخصصی در حوزه نرم افزار را دارید، آشنایی با SQL به صورت تخصصی جزء ملزومات اصلی این کار خواهد بود و در این تمرین هم برآنیم که تا حد امکان به این هدف برسیم.

دیتابسی که برای این تمرین در نظر گرفته شده است، دیتابیس معروف آموزشی میکروسافت با نام Northwind¹ است که هر چند کمی قدیمی است اما برای اهداف آموزشی ما، بسیار مناسب است و نسخه پستگرس آن را در این تمرین استفاده خواهیم کرد.

این دیتابیس که یک فروشگاه بزرگ را مدلسازی می‌کند، از جداول اصلی زیر تشکیل شده است :

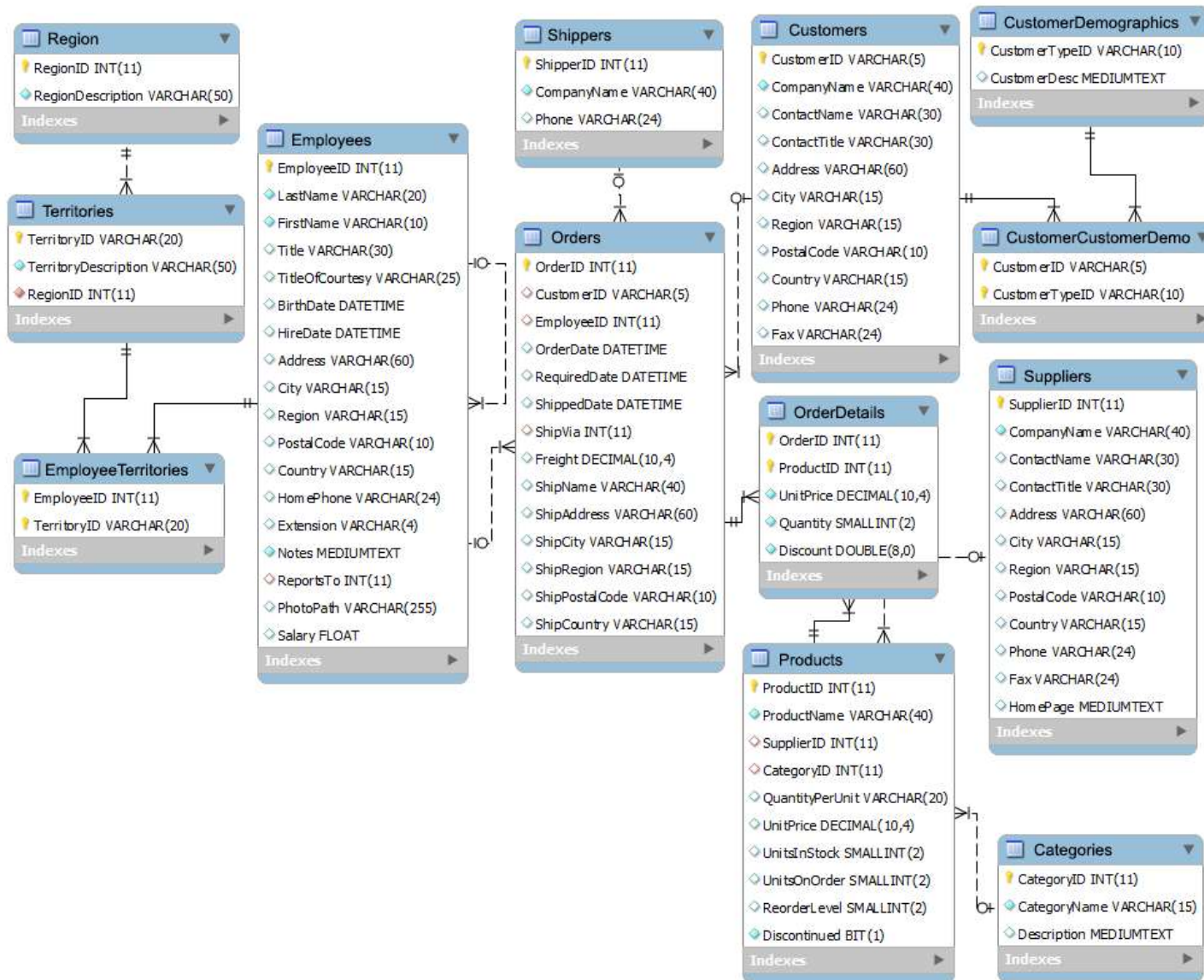
- Customer : مشتریان فروشگاه
- Order : سفارشات هر مشتری - اطلاعات کلی
- OrderDetails : جزئیات هر سفارش شامل آیتم‌های خریداری شده .
- Products : محصولات فروشگاه
- Suppliers: تامین کنندگان کالا که محصولات از آنها تهیه می‌شوند.
- Ctaegories : گروه بندی کالاها
- Shippers : شرکت های حمل و نقلی که وظیفه ارسال کالا به دست مشتری را بر عهده دارند.
- Employee: کارمندان فروشگاه که هر سفارش، توسط یک کارمند ثبت یا تایید می‌شود.
- Territory & Region: مناطق و نواحی ای که این فروشگاه در آنها شعبه دارد. هر منطقه شامل چندین ناحیه است و هر کارمند، می‌تواند به سفارشات چندین ناحیه رسیدگی کند.
- CustomerDemographics : برای رتبه بندی مشتریان استفاده می‌شود که در این تمرین با آن سروکار نخواهیم داشت.

در صفحه بعد، نمودار ER این دیتابیس را مشاهده می‌کنید.

نسخه آنلاین آنرا می‌توانید از این آدرس دانلود کنید :

<https://www.w3resource.com/mysql-exercises/northwind/products-table-exercises/mysql-northwind-database.php>

¹ <https://github.com/Microsoft/sql-server-samples/tree/master/samples/databases/northwind-pubs>

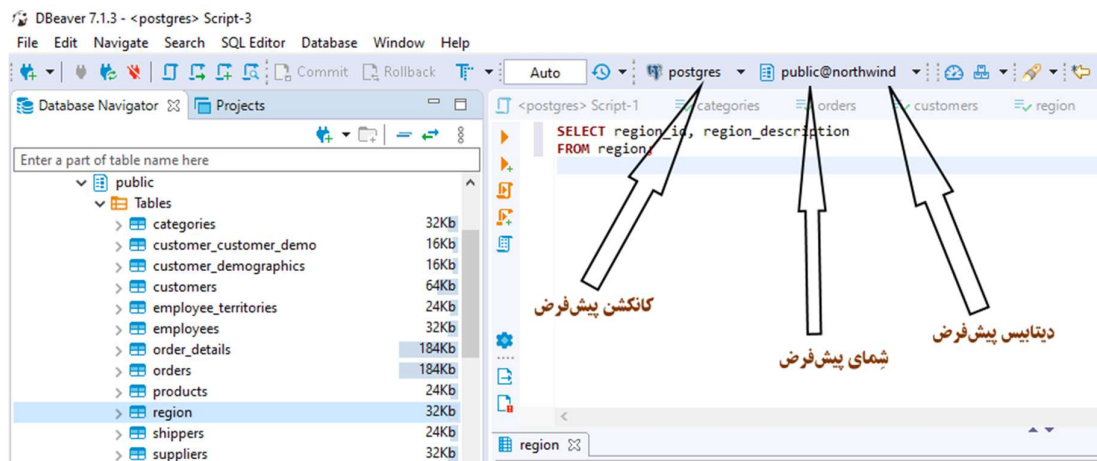


ایجاد و ایمپورت داده‌های دیتابیس Northwind

قبل از شروع به کار، باید خود دیتابیس Northwind را درون پستگرس بسازیم و سپس ساختار جداول و داده‌های آنرا با اجرای مجموعه دستورات SQL، ایجاد کنیم.

دیتابیسی با نام Northwind ایجاد کنید. (پستگرس به حروف بزرگ و کوچک حساس است بنابراین حواستان به این موضوع باشد). حال بر روی دیتابیس کلیک راست کرده، گزینه Set As Default را بزنید که دیتابیس پیش‌فرض شما در هنگام اجرای دستورات SQL، این دیتابیس باشد.

سپس گزینه F3 را بزنید و یا از منوی بالا، SQL Editor را انتخاب کنید. یک فایل اسکریپت جدید باز می‌شود که در مرحله بعد از منوی SQL Editor گزینه Import SQL Script را انتخاب کرده، فایلی که در این آدرس^۱ قرار گرفته است را دانلود کرده، آنرا ایمپورت نمایید. دقت کنید که دیتابیس پیش‌فرض شما حتما Northwind باشد (مطابق تصویر زیر):



حال با زدن دکمه Ctrl+A همه دستورات را انتخاب کرده و مثلث کوچک نارنجی رنگ صفحه ویرایشگر SQL را برای اجرای تمام دستورات انتخاب شده، بزنید.

مطابق شکل بالا، درون شمای پابلیک باید جداول شما ساخته شده باشد. (البته نیاز خواهید داشت روی دیتابیس Northwind کلیک راست کرده و گزینه Refresh را بزنید). حال اگر جداول اصلی را باز کنید، در قسمت Data، داده‌های آنها را می‌توانید مشاهده کنید.

اگر تا اینجا همه چیز بدون مشکل پیش رفته باشد، آماده انجام تمرینات این دستورکار شده اید.

^۱ https://github.com/pthom/northwind_psql/blob/master/northwind.sql

دستورات پایه

برای انجام این تمرین، یک فایل SQL جدید ایجاد کنید و این فایلها را هم در انتها، کنار گزارش خود ارسال نمایید. (با زدن دکمه F3 و انتخاب new script)

احیاناً اگر ابهامی در یک سوال مشاهده می کنید، با در نظر گرفتن فرضی که این ابهام را برطرف کند آنرا حل کنید.

1. می خواهیم ببینیم به ازای هر منطقه (region)، چه نواحی ای تعریف شده است. دستور SQL متناظر را با استفاده از inner join جدول region و territories بنویسید. فقط نام منطقه و نام ناحیه در خروجی باید ظاهر شود. دستور inner join مشابه این خواهد بود (r و t نام مستعار دو جدول هستند که برای ساده تر شدن نوشتن دستورات SQL معمولاً از این شیوه استفاده می کنیم):

```
from region r inner join territories t on r.region_id = t.region_id
```

2. حال می خواهیم تعداد کارمندان هر منطقه (region) را به دست آوریم. دستور SQL متناظر را بنویسید.

راهنمایی: رابطه مستقیمی بین ناحیه و کارمند وجود ندارد و باید از جدول واسط این دو یعنی employee_territories استفاده کنیم. بنابراین در قسمت from با inner join این چند جدول سروکار داریم. از طرفی برای شمردن تعداد کارمندان بر حسب هر منطقه، باید داده ها بر حسب منطقه گروه بندی شده و سپس کارمندان آنها شمارش شوند. در اینجا باید از دستور group by r.region_id بعد از from استفاده کنیم. در مرحله آخر هم دستور شمارش را در جلوی select بنویسیم.

بنابراین در قسمت select خواهیم داشت:

```
select r.region_description , count(e.employee_id )
```

3. با در نظر گرفتن میزان تخفیف و تعداد هر محصول در سفارش، قیمت کل پرداختی مشتری به ازای هر سفارش را محاسبه کنید. از جدول OrderDetails استفاده کنید. خروجی این دستور، تمام شماره سفارش ها و مبلغ پرداختی به ازای آن سفارش ها خواهد بود.

4. ده محصولی که بیشترین تعداد خرید را به خود اختصاص داده اند را بیابید. منظور از تعداد خرید، مجموع Quantity آن محصول در جدول OrderDetails است.

5. آیا محصولی وجود دارد که هنوز فروش نرفته باشد؟

6. می خواهیم گزارشی تهیه کنیم از محصولات و تعداد سفارش هایی که این محصول در آنها به کار رفته است (تعداد سفارش هایی که این محصول در آنها به کار رفته و نه تعداد کل فروش محصول). دستور sql مربوطه را بنویسید. دقت کنید که برای این سوال، نیاز دارید از جویین های خارجی (چپ یا راست) استفاده کنید.

7. می خواهیم به کارمندی که بیشترین فروش (از لحاظ مبلغ پرداختی مشتری که در گام سه محاسبه کرده اید) را در یک سال خاص داشته است (یک سال فرضی در نظر بگیرید)، جایزه بدهیم. این کارمند را چگونه به دست می آوریم؟ از توابع تاریخ در پستگرس برای استخراج سال از روی OrderDate و گروه بندی داده ها بر اساس آن، استفاده کنید. توصیه میکنم برای راحتی کار از CTE در SQL استفاده کنید. یعنی جداول مورد نیاز را به صورت موقت

در ابتدا بسازید (با **with**) - مثل جدولی که خروجی دستور سوم است و اینجا به آن نیاز دارید - سپس از آن جدول در دستورات پایین تر استفاده کنید. به مثال زیر دقت کنید :

```
WITH regional_sales AS (
    SELECT region, SUM(amount) AS total_sales
    FROM orders
    GROUP BY region
), top_regions AS (
    SELECT region
    FROM regional_sales
    WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
)
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

8. می خواهیم به سفارشات که همان روز سفارش ارسال شده اند (یعنی **OrderDate** و **ShippedDate** آنها برابر است)، برچسب **عالی**، آنهایی که تا سه روز بعد از سفارش ارسال شده اند، برچسب **خوب** و به بقیه سفارشات برچسب **نامناسب** بزنیم . در گزارش نهایی، کد سفارش و برچسب آن باید ظاهر شود. از دستور **when** و **case** استفاده کنید.

9. (سوال امتیازی) برای دستور **WITH RECURSIVE** در این دیتاست، یک مثال بنویسید که خروجی مناسب تولید کند.

10. کوئری زیر را در نظر بگیرید :

```
select distinct date(a.ShippedDate) as ShippedDate,
       a.OrderID,
       b.Subtotal,
       year(a.ShippedDate) as Year
from Orders a
inner join
(
    select distinct OrderID,
           sum(UnitPrice * Quantity) as Subtotal
    from order_details
    group by OrderID
) b on a.OrderID = b.OrderID
where a.ShippedDate is not null
      and a.ShippedDate between date('1996-12-24') and date('1997-09-30')
order by a.ShippedDate;
```

این کوئری چه کاری انجام می دهد؟ آنرا به گونه ای تغییر دهید که مجموع فروش هر سال از داده های دیتابیس را تولید کند. خروجی مورد نظر، **سال و میزان کل فروش آن** خواهد بود.

11. می خواهیم سفارش جدیدی برای محصولات که تعداد آنها (**UnitsInStock**) از حد مجاز (**ReorderLevel**) کمتر شده است، بدهیم. این محصولات را بیابید و آنها بر اساس میزان موجودی (به صورت صعودی) نمایش دهید. یک ویو برای ذخیره این دستور ایجاد کنید که همواره برای مشاهده محصولات که نیاز به خرید دارند، این ویو، استفاده شود.

12. از کدام دسته محصولات، به فرانسه چیزی ارسال نشده است؟

13. از کدام دسته محصولات، به فرانسه چیزی ارسال نشده است؟
14. مشتریانی را بیابید که شماره فکس آنها در سیستم موجود نیست. (مقایسه **fax** با **null**)
15. می خواهیم اطلاعات کارکنان را نمایش بدهیم و قصد داریم سن کارکنان را براساس تاریخ امروز محاسبه (توابع کار با تاریخ پستگرس) و به عنوان یک ستون جدید با نام **Age** در خروجی **Select** نمایش دهیم. خروجی این دستور را به عنوان یک **View** ذخیره کنید. سپس میانگین سن هر منطقه را با استفاده از این **View** محاسبه کنید. (می توانید از تابع **age** در پستگرس یا **current_date** استفاده کنید)