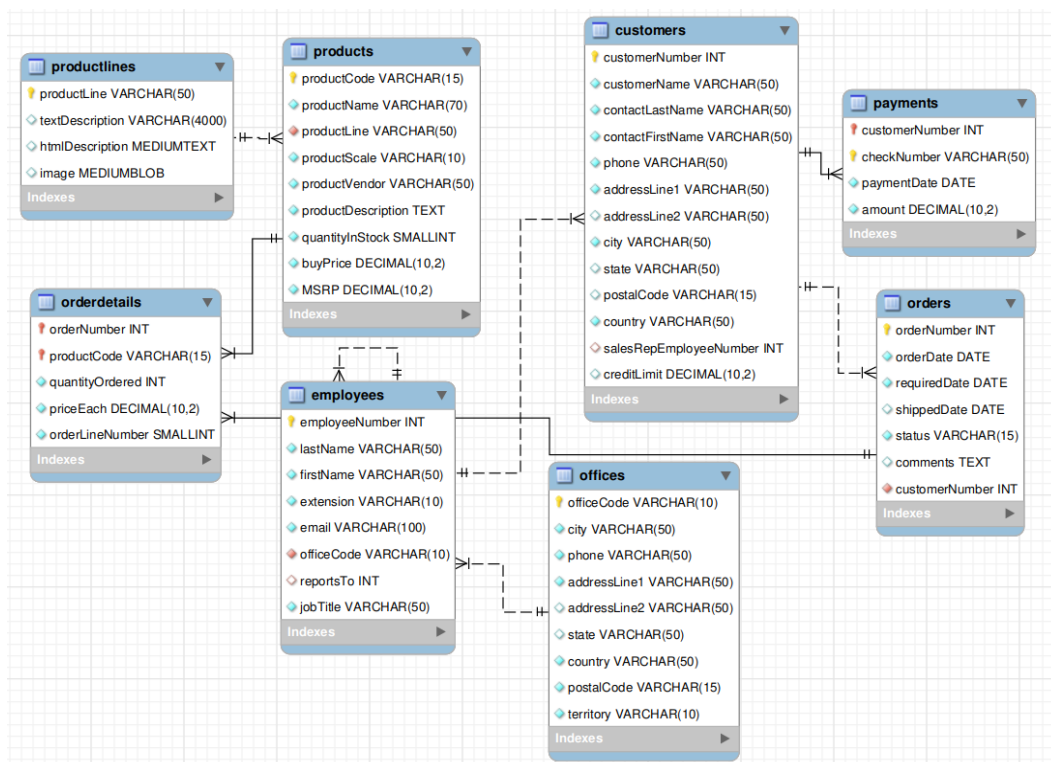


## نمونه سوال تمرینی SQL

دیتابیس را از این لینک دانلود کنید و در محیط MySQL Workbench اجرا کنید.

ساختار EER برای این پایگاه داده به صورت زیر است:

برای هر یک از سوالات زیر query مناسب بنویسید.



- (1) جدولی شامل تمام مشتریان
- (2) جدولی شامل ستون های contactLastName و contactFirstName و customerNumber تمام مشتریان
- (3) جدولی شامل ستون های contactLastName و contactFirstName و customerNumber تمام مشتریان اما عنوان این ستون ها به ترتیب First Name و Last Name و Customer ID باشد.
- (4) جدولی شامل ستون های customerNumber و creditScore که creditScore از حاصل تقسیم creditLimit به 1000 حاصل می شود.
- (5) جدولی شامل یک ستون state از ایالت های متفاوت که مشتریان در آن ها زندگی می کنند. (دقت شود ایالت های تکراری پذیرفته نیست).
- (6) تمام مشتریانی که در کشور USA زندگی می کنند.
- (7) تمام مشتریانی که در کشور ایالت CA یا NY زندگی می کنند و creditLimit بیشتر از 5000 دارند.
- (8) جدولی شامل تمام سفارشات (orders) که بعد از تاریخ 2004-01-01 ثبت شده اند.
- (9) جدولی شامل تمام سفارشات که وضعیت (status) آنها Shipped نیست.
- (10) جدولی شامل تمام مشتریانی که نصف creditScore آنها از 8 بیشتر است (توجه شود که creditScore حاصل تقسیم creditLimit بر 10000 است).
- (11) جدولی شامل تمام مشتریانی که در ایالت های CA یا NY یا Tokyo زندگی می کنند. (از عملگر IN استفاده کنید)
- (12) جدولی شامل تمام مشتریانی که در ایالت های مذکور در مورد 11 زندگی نمی کنند.
- (13) جدولی شامل تمام مشتریانی که creditLimit بین 80000 و 120000 دارند. (از عملگر BETWEEN استفاده کنید)
- (14) جدولی شامل تمام مشتریانی که creditLimit بین 80000 و 120000 ندارند. (از عملگر BETWEEN استفاده کنید)

- (15) جدولی شامل تمام مشتریانی که نام کوچکشان با حرف B شروع می شود.
- (16) جدولی شامل تمام مشتریانی که نام کوچکشان با حرف B شروع می شود و 3 کاراکتر دارد.
- (17) جدولی شامل تمام مشتریانی که نام و نام خانوادگی شان حرف c ندارد.
- (18) جدولی شامل تمام مشتریانی که نام خانوادگی شان با fer یا fre شروع شود. (از REGEXP استفاده کنید)
- (19) جدولی شامل تمام مشتریانی که کد پستی شان null است.
- (20) جدولی شامل تمام مشتریانی که آدرس دوم شان موجود است. (آدرس دوم در ستون addressLine2 می باشد)
- (21) جدولی شامل تمام مشتریان که creditLimit بیش از 40000 دارند و بر اساس نام خانوادگی مرتب شده اند.
- (22) جدولی شامل ستون های نام و نام خانوادگی و customerNumber که ابتدا بر اساس نام خانوادگی مرتب شده است و سپس در صورتی که نام خانوادگی یکسانی داشته باشند بر اساس نام شان مرتب شده باشد.
- (23) جدولی شامل تمام مشتریانی که بر اساس creditLimit شان که به صورت کاهشی مرتب شده اند.
- (24) جدولی شامل 10 مشتری که بیشترین creditLimit را دارند.
- (25) جدولی شامل 10 سوم که بیشترین creditLimit را دارند.
- (26) تعداد تمام مشتریان
- (27) مقدار بیشترین creditLimit
- (28) مقدار کمترین creditLimit
- (29) مقدار میانگین creditLimit
- (30) مقدار مجموع creditLimit تمام مشتریان
- (31) جدولی شامل (به ترتیب) ستون های orderNumber و status از جدول سفارشات که همراه با ستون های نام و نام خانوادگی و customerNumber از جدول مشتریان که هر سفارش را به مشتری مدنظر منتسب می کند. این جدول را بر اساس customerNumber مرتب مرتب کنید.
- (32) جدولی شامل ستون های orderNumber و status و customerNumber و نام و نام خانوادگی مشتریانی که حداقل یک سفارش دارند که وضعیت (status) آن shipped نیست. (مرتب شده بر اساس customerNumber)
- (33) می خواهیم رییس هر کارمند را مشخص کنیم. جدولی شامل ستون های employeeNumber و employeeName و employeeJob که به ترتیب ID کارمند و نام کامل کارمند (با یک فاصله بین نام و نام خانوادگی) و jobTitle کارمند. این جدول باید شامل همین ستون ها اما برای رییس آن کارمند باشد (به جای employee از manager استفاده کنید).
- (34) می خواهیم بدانیم از تمامی محصولات که در ماه اول سال ۲۰۰۴ خریداری شده اند چه تعداد در انبار موجود است. (شامل جدول های productCode و quantityInStock و buyPrice از جدول product)
- (35) دوباره می خواهیم رییس هر کارمند را مشخص کنیم. اما این بار هر دو فرد رییس و کارمند باید در یک office کار کنند. (دقیقا همان ستون های خواسته شده در مورد 33 علاوه بر officeCode شان)
- (36) Query سوال 31 را بدون این که از INNER JOIN استفاده کنید و با استفاده از WHERE بازنویسی کنید. (راهنمایی: می توانید چندین جدول را بعد از FROM انتخاب کنید)
- (37) لیست مشتریانی که تا به حال هیچ سفارشی ثبت نکرده اند. این جدول باید شامل customerNumber و نام و نام خانوادگی مشتری باشد.
- (38) می خواهیم بدانیم کدام محصولات تا به حال خریداری نشده اند. (ستون های productCode و productName این محصولات را در این جدول نمایش دهید)
- (39) تمام محصولات که در نیمه دوم سال ۲۰۰۳ خریداری شده اند. (ستون های productCode و productName این محصولات را در این جدول نمایش دهید)
- (40) تمام کارمندانی که مدیر ندارند (خروجی به صورت یک جدول شامل نام و نام خانوادگی کارمند و employeeNumber و jobTitle کارمند)
- (41) کارمندانی که در office شهر Paris کار میکنند. (شامل ستون های employeeNumber و نام و نام خانوادگی)
- (42) اگر شرط join را مشخص نکنید عملا cross join انجام داده اید. جدول های customers و orders را با یکدیگر cross join کنید.
- (43) جدولی شامل تمام سفارشات که شامل یک ستون اضافی است. این ستون status نام دارد و برای سفارشات که orderDate قبل از سال ۲۰۰۵ دارند archive است در غیر این صورت active است.
- (44) یک row به جدول customers با اطلاعات خودتان وارد کنید.
- (45) یک سفارش به جدول orders اضافه کنید که orderNumber آن ۱۲ orderDate آن تاریخ امروز requiredDate آن ۵ روز بعد status آن Shipped و customerNumber آن ۱ باشد.
- حال یک orderDetail برای این order اضافه کنید که productCode آن S10-1678 است و quantityOrdered آن ۲ است و eachPrice آن ۲۰۰ دلار است و orderLineNumber آن ۳ است.

- (46) یک جدول به نام `ordres_archived` بسازید که شامل محتوای جدول `ordres` باشد.
- (47) حال تمام row های جدول `orders_archived` را حذف کنید و فقط سفارشات که از تاریخ 2004-01-01 انجام شده است را به این جدول اضافه کنید.
- (48) جدولی به نام `customer_payment` بسازید که شامل ستون `checkNumber` از جدول `payment` و `customerNumber` و نام و نام خانوادگی از جدول `customer` باشد. تمام `payment` های در نظر گرفته شده در این جدول باید قبل از تاریخ 2004-01-01 انجام شده باشند.
- (49) مقدار `amount` پرداختی که `checkNumber` برابر با HQ336336 دارد را ۱۰ درصد افزایش دهید.
- (50) کاری که در مورد ۴۹ انجام دادید را برای `checkNumber` های JM555205 و GG31455 و FD317790 و NT141748 و ID10962 و HR182688 انجام دهید.
- (51) مقدار `requireDate` سفارشات که کنسل شده اند را یک روز افزایش دهید.
- (52) مقدار `creditLimit` مشتری که اولین خرید را انجام داده است ۲۰ درصد افزایش دهید.
- (53) مقدار `creditLimit` مشتریانی که حداقل یک خرید را `cancel` کرده اند را ۱۵ درصد کاهش دهید.
- (54) جدولی شامل `orderNumber` و `customerNumber` و مبلغ سفارش مربوطه با عنوان `totalPrice` برای همه سفارشات: `Note #`
- (55) جدولی شامل تمامی مشتریانی که تا به حال بیش از ۱۰۰۰۰۰ دلار پرداخت انجام داده اند و در کشور امریکا ساکن هستند. این جدول شامل ستون های `customerNumber` و نام کامل مشتری و تمام مبلغ پرداختی باشد.
- (56) جدولی شامل مقدار فروش از هر کشور و هر شهر (از `USE ROLLUP` استفاده کنید)
- (57) جدولی شامل مشتریانی که میزان پرداختی شان از میزان پرداختی میانگین مشتریان بیشتر بوده است.
- (58) محصولاتی که تا به حال خریداری نشده اند. (از `subquery` استفاده کنید)
- (59) محصولات به ترتیب پرفروش ترین
- (60) محصولات به ترتیب بیشترین سود (محصولی که حاصل ضرب قیمت فروخته شده و تعداد فروخته شده بیشتری دارد ابتدا می آید)
- (61) تمام مشتریانی که تا به حال محصول 'S12\_4473' را خریداری کرده اند. (query را به هر دو فرم `subquery` و `join` بنویسید.)
- (62) مشتریانی که از تمام مشتریان امریکایی `creditLimit` بالاتری دارند. (از `ALL` استفاده کنید)
- (63) مشتریانی که از حداقل یک مشتری امریکایی `creditLimit` بیشتری دارند (از `ANY` استفاده کنید)
- (64) مشتریانی که تا به حال بیش از ۳ پرداخت انجام داده اند.
- (65) تمام محصولات که بیش از 27 بار فروخته شده اند. (دقت کنید ۲۷ بار فروخته شده اند و نه ۲۷ عدد)
- (66) اطلاعات سفارشات تمام مشتریانی امریکایی
- (67) جدولی شامل مشتریان با یک ستون اضافه به نام `differenceCreditLimit` که مقدار آن مقدار `creditLimit` منهای میانگین `creditLimit` مشتریان امریکایی باشد.
- (68) مشتریانی که مجموع پرداخت هایشان از میانگین مجموع پرداخت تمام مشتریان کمتر است.
- (69) جدولی شامل `customerNumber` و `creditLimitNorm` که همان `creditLimit` است که تقسیم بر بیشترین منهای کمترین مقدار `creditLimit` شده است (دقت کنید که مقدار `creditLimitNorm` باید تا دو رقم بعد از اعشار رند شده باشد).
- (70) همان جدول بالا را این بار فقط تا دو رقم بعد از اعشار نمایش داده شود (رند نشود)
- (71) جدولی شامل `customerNumber` و طول آدرس `addressLine1` (طول رشته آدرس)
- (72) جدولی شامل `customerNumber` و `contactFirstName` با تمام حروف کوچک انگلیسی و `contactLastName` با تمام حروف بزرگ انگلیسی
- (73) اطلاعات سفارشات را به همراه یک ستون اضافه به نام `passedDays` که نشان دهنده تعداد روز های گذشته از ثبت سفارش است را نمایش دهید.
- (74) زمان جاری را نمایش دهید.
- (75) از جدول مشتریان `customerNumber` و `addressLine1` و `addressLine2` را نمایش دهید اما اگر `addressLine2` مقداری نداشت مقدار آن `NotAssigned` شود.
- (76)
- (77) تمام سفارشات را نمایش دهید. این جدول یک ستون اضافه به عنوان `dateStatus` دارد که اگر سفارش قبل از تاریخ 2004-01-01 انجام شده باشد مقدار `Archived` دارد و اگر بعد از این تاریخ باشد مقدار `Active` دارد.
- (78) مشتریان را به دو دسته تقسیم می کنیم. آنهایی که بیش از ۴ سفارش تا به حال انجام داده اند وفادار هستند و آنهایی که کمتر از این مقدار انجام داده باشند معمولی هستند. یک ستون اضافه به نام `vafadari` کنید که اگر مشتری وفادار بود مقدار `Vafadar` و در غیر این صورت `Mamoli` باشد.
- (79) می خواهیم یک دسته دیگر به این مشتریان اضافه کنیم. اگر فردی بیش از ۱۰ سفارش خیلی وفادار است و مقدار `vafadari` آن `very_vafadar` است.
- (80) یک `view` بسازید که شامل تمام مشتریان امریکایی باشد. نام این `view` را `american_costomers` بگذارید.

- (81) View ساخته شده در مثال قبل را حذف کنید.
- (82) یک view بسازید که شامل تمام مشتریان که محصول S10\_4962 را به تعداد بیش از ۳۰ خریداری کرده اند. نام این view را special\_customers بگذارید.
- (83) View قبلی را به صورتی آپدیت کنید که شامل تمام مشتریانی که محصول S12\_3380 را به تعداد بیشتر از ۲۰ عدد خریداری کرده اند شود.
- (84) یک view چه شرایطی باید داشته باشد که updateable باشد (updateable views). یعنی از آن view بتوانیم در statement های UPDATE و INSET و DELETE استفاده کنیم.
- (85) یک Stored Procedure به نام get\_customers بسازید که تمام مشتریان را برگرداند.
- (86) Stored Procedure ساخته شده در مثال قبل را حذف کنید.
- (87) یک Stored Procedure بسازید که مشتریان را با توجه به کشور آن مشتری برگرداند. در واقع کشور یک ورودی برای stored procedure است.
- (88) Stored Procedure قبلی را به صورتی آپدیت کنید که اگر بدون ورودی فراخوانی شد تمام مشتریان و اگر با ورودی فراخوانی شد شرط کشور را اعمال کند.
- (89) Stored Procedure قبلی را به صورتی آپدیت کنید که اگر بدون ورودی فراخوانی شود مشتریان امریکایی و اگر با ورودی فراخوانی شود مشتریان آن کشور برگردانده می شوند.
- (90) یک Stored Procedure بسازید که سفارشات را بر اساس customerNumber و status برگرداند. برای هر یک از customerNumber و status نیز مقدار پیش فرض را به صورتی در نظر بگیرید که داده نشده بود تمام customerNumber ها یا تمام status ها را برگرداند. در نظر داشته باشید که امکان دارد یکی از مقادیر داده شود و دیگری داده نشود (مقدار NULL داشته باشد).
- (91) یک Stored Procedure بسازید که creditLimit مشتری که customerNumber آن را دریافت می کند را آپدیت کند. نام این Stored Procedure را update\_creditLimit بگذارید. در صورتی که creditLimit ورودی مقدار منفی داشت یک خطا برگردانید. اگر customerNumber ورودی نیز نامعتبر بود یک خطا برگرداند.
- (92) یک Stored Procedure بسازید که تعداد پرداخت ها و میزان کل پرداختی مشتری که customerNumber او را دریافت می کند را برگرداند. در این stored procedure از output parameters استفاده کنید.
- (93) یک Stored Procedure بنویسید که تمام مشتریان را برگرداند و یک ستون اضافه داشته باشد که creditLimit هر مشتری را به عددی بین ۰ و ۱ نگاشت دهد. یعنی مقدار این ستون برای مشتری که بیشترین creditLimit را دارد مقدار ۱ و مشتری که کمترین creditLimit را دارد ۰ باشد. (از local variables استفاده کنید). نام این ستون را score بگذارید.
- (94) یک stored procedure بنویسید که customerNumber یک مشتری را دریافت می کند و بر اساس آن میانگین میزان پرداخت هایش را برگرداند.
- (95) یک function بنویسید که customerNumber یک مشتری را دریافت می کند و بر اساس آن میانگین میزان پرداخت هایش را برگرداند.
- (96) یک trigger بنویسید که پس از این که مشتری یک تراکنش انجام می دهد به میزان ۱۰ درصد مقدار آن تراکنش به creditLimit آن مشتری افزوده شود.
- (97) تمام trigger های تعریف شده را نمایش دهید.
- (98) یک جدول برای حسابرسی پرداخت ها ایجاد کنید که شامل customerNumber و date از نوع تاریخ و amount و action\_type و action\_date باشد. Trigger سوال قبل را به گونه ای تغییر دهید که با ایجاد هر payment یک log در جدول حسابرسی ایجاد شود. نام این جدول را payment\_audit بگذارید.
- (99) یک trigger بنویسید که اگر هر payment حذف شد به میزان ۱۰ درصد آن تراکنش از creditLimit آن مشتری کاسته شود و یک log در جدول حسابرسی ایجاد شود.
- (100) -
- (101) یک event بسازید که هر سال از زمان جاری تا ۱۰ سال آینده row هایی از جدول payment\_audit که از تاریخ action\_date آن یک سال گذشته است را حذف کند. نام این event را yearly\_delete\_audit\_rows بگذارید.
- (102) تمام EVENT ها را نمایش دهید.
- (103) yearly\_delete\_audit\_rows را به گونه ای تغییر دهید که این کار را در بازه های ۶ ماه انجام دهد.
- (104) yearly\_delete\_audit\_rows را موقتاً غیرفعال کنید.
- (105) yearly\_delete\_audit\_rows را حذف کنید.
- (106) Transaction را توضیح دهید.
- (107) یک transaction در بسازید که یک سفارش جدید که orderNumber آن ۱۰۵۰۰ و orderDate آن زمان جاری و requireDate آن ۳ روز بعد shippedDate آن NULL و status آن On Hold است برای مشتری ۱۷۷ ایجاد کنید. این سفارش دارای یک orderdetail که productCode آن S10\_4757 به تعداد ۱۰۰ با قیمت ۲۰۰ دلار و orderLineNumber برابر با ۳ باشد.
- (108)