

بسم الله الرحمن الرحيم

درس: پایگاه داده‌ها

استاد: دکتر مجتبی رفیعی

دانشجویان:

سنا نادعلی – امیرحسین رستمی

مستند فاز ۲ و ۳ پروژه درس

تابستان ۱۴۰۰

فهرست مطالب مستند:

بخش اول:

بیان جدول‌های طراحی شده و جزییات ستون‌های هرکدام

بخش دوم:

بیان نحوه مدل‌سازی query ها جهت تحقق نیازمندی‌ها

بخش سوم:

بیان دیدهایی که هرکاربر ممکن است نیاز داشته باشد

بخش چهارم:

بیان ابزار مورد استفاده در مدیریت پایگاه داده

بخش پنجم:

بیان Assertion ها – دامنه‌های صفات – محدودیت‌های جامعیتی

بخش ششم:

بیان رهانا‌های (و Job های) موردنیاز

بخش هفتم:

بیان رابط‌های کاربری پیاده‌سازی شده (جهت اتصال به پایگاه‌داده و انجام پرس‌وجوهای لازم)

بخش هشتم:

بیان میزان مشارکت اعضای گروه در پیاده‌سازی بخش‌های مختلف

بخش اول (بیان جدول‌های طراحی شده و جزئیات ستون‌های هرکدام)

فروشنده (seller):

کد فروشندگی (C.K) (int) – نام برند (varchar) – برآیند امتیاز (float) – وضعیت ارائه خدمات (float) – محدودیت فروش در روز (int) – پروانه کسب/آدرس پروانه در سرور* (varchar) – اسکن شناسنامه/آدرس فایل در سرور* (varchar) – اسکن کارت ملی/آدرس پروانه در سرور* (varchar) – وضعیت تایید (varchar) – تغییر بعد از آخرین بررسی (bit)

فقط «محدودیت فروش» هیچ مقدارپذیر است و هیچ مقدار بودن آن به این معنی ست که فروشنده می‌تواند فروش نامحدود داشته‌باشد.

کلید کاندید: (کد فروشندگی)

توجه: نیاز به یک رهانا داریم که هر زمان فروشنده، اطلاعاتی از این جدول را تغییر داد، مقدار «تغییر بعد از آخرین بررسی» به ۱ تغییر کند.

جدول نشانی فروشگاه (shopAddress):

کد پستی (C.K) (int) – آدرس (varchar) – موقعیت جغرافیایی (varchar) – کد ملی فروشنده (foreign key) (int)

همگی هیچ مقدارناپذیر

کلید کاندید: (کد پستی)

جدول پروفایل فروشنده (sellerProfile):

نام (varchar) – نام خانوادگی (varchar) – عکس/آدرس عکس در سرور* (varchar) – تاریخ تولد (date) – شماره تلفن همراه (varchar) – ایمیل (varchar) – رمز عبور (varchar) – نام کاربری (varchar) – شماره حساب (varchar)

کد ملی (varchar) – کد فروشنده (foreign key) (int)

فقط «عکس» هیچ مقدارپذیر است.

کلید کاندید: (کد فروشنده)

توجه: اگرچه کد ملی یک identifier برای انسان است، اما پروفایل با یک کد ملی خالی، لزوماً پروفایل یک فروشنده نیست.

کالا (product):

کد کالا (C.K) (int) – وضعیت تایید (varchar) – برند (varchar) – بارکد (varchar) – نام تولیدکننده (varchar) – قیمت (int) – موجودی (float) – تعداد مجاز در یک خرید (int) – نام (varchar) – تغییر بعد از بررسی (Boolean) – امتیاز خرید (int) – تخفیف (int) – گارانتی (varchar) – میانگین امتیاز کاربران (float) – میزان فروش (int) – مشخصات (varchar)

تاریخ بارگذاری (date) – کد فروشنده بارگذارنده (foreign key) (int)

صفات «تعداد مجاز در هر خرید»، «تخفیف» و «گارانتی» هیچ مقدارپذیرند.

کلید کاندید: (کد کالا)

دسته بندی کالا (category):

کد دسته (C.K) (int) - نام دسته (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد دسته)

جدول دسته - برچسب (categoryLabel):

کد دسته (foreign key) (int) - برچسب (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد دسته، برچسب)

جدول کالا - تصویر (productPic):

کد کالا (foreign key) (int) - آدرس ذخیره عکس (با نام عکس) (varchar) - تاریخ بارگذاری (timestamp)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کالا، آدرس ذخیره عکس)

توجه: اغلب برای یک کالا عکس های متعددی در نظر گرفته می شود و لذا نگهداری تصاویر که یک صفت چند مقداری کالا است را به صورت یک جدول مجزا در نظر گرفتیم.

جدول کالا - برچسب (productLabel):

کد کالا (foreign key) (int) - برچسب (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کالا، برچسب)

توجه: اغلب بر یک کالا بر حسب دسته بندی های مختلف، برچسب های متعددی زده می شود و لذا نگهداری برچسب ها که یک صفت چند مقداری کالا است را به صورت یک جدول مجزا در نظر گرفتیم.

نوع موجودیت نشانی (location):

کد پستی (C.K) (int) - آدرس (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد پستی)

کارمند(employee):

کد کارمندی (C.K) (int) – قرارداد (varchar) – حقوق (int) – گواهی عدم سو سابقه / * آدرس در سرور (varchar) / * – اسکن کارت ملی / * آدرس در سرور (varchar) / * – رزومه / * آدرس در سرور (varchar) / * – اسکن شناسنامه / * آدرس در سرور (varchar) / * – سال ورود (timestamp) – سمت در شرکت (مدیر (manager) – بازرس (controllor) – مسئول حمل و نقل (transporter) – پشتیبانی (support) – معزول (dismissed) (varchar) – اسکن گواهی نامه / * آدرس در سرور (varchar) / * – کد کارمند استخدام کننده (int) – تاریخ استخدام (date)

صفت «اسکن گواهی نامه» هیچ مقدارپذیر است. اما همه مسئولان حمل و نقل باید گواهی نامه داشته باشند.

کلید کاندید: کد کارمندی

به علت زیاد بودن صفات زیرنوع و بسیار اندک بودن صفات زیرنوع و تخصیص کامل و مجزا بودن شجره وراثت کارمند، از تکنیک 6 جهت مدل سازی استفاده کرده ام و با قراردادن «سمت» در جدول کارمند مشخص می کنیم از کدام زیرنوع است.

پروفایل کارمند (employeeProfile):

نام (varchar) – نام خانوادگی (varchar) – عکس (varchar) – کد ملی (varchar) – کد کارمند (foreign key) (int) – تاریخ تولد (date) – شماره تلفن همراه (varchar) – شماره حساب (varchar) – نام کاربری (varchar) – رمز عبور (varchar) – ایمیل (varchar) – نشانی ۱ (foreign key) (varchar) – نشانی ۲ (foreign key) (varchar)

صفات «عکس» و «نشانی ۲» هیچ مقدارپذیرند.

کلید کاندید: (کد کارمند)

توجه ۱: اگرچه کد ملی یک identifier برای انسان است، اما پروفایل با یک کد ملی خالی، لزوماً پروفایل یک کارمند نیست.

توجه ۲: برای صفت چندمقداری «نشانی»، حداقل یک و حداکثر دو مقدار نگه می داریم.

جدول عزل (dismiss):

کد کارمند (مدیر) < کارمندی که سمت "مدیر" دارد (foreign key) (int) – کد کارمند (foreign key) (int) – تاریخ (Date) – علت (رشته)

همگی هیچ مقدار ناپذیر اند.

کلید کاندید: (کد کارمند مدیر، کد کارمند عزل شده)

توجه: از آنجا که نرخ عزل در شرکت اندک (زیر ۳۰ درصد) است لذا برای پرهیز از هیچ مقداری، نوع ارتباط عزل را به صورت جدولی مجزا مدل سازی کردیم.

جدول تراکنش (transaction):

شماره تراکنش (C.K) (int) – مبلغ (float) – تاریخ (timestamp)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: شماره تراکنش

جدول تسویه حساب فروشندگان(sellerPayment):

کد فروشنده (foreign key)(int) – کد کارمند مدیر (foreign key)(int) – شماره تراکنش (foreign key)(int) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد فروشنده، کد کارمند مدیر، شماره تراکنش)

توجه: نیاز به Assertion اینکه کارمند پرداخت کننده "مدیر" باشد داریم.

جدول پرداخت حقوق(employeePayment):

کد کارمند (foreign key)(int) – کد کارمند مدیر (foreign key)(int) – شماره تراکنش (foreign key)(int) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کارمند، کد کارمند مدیر، شماره تراکنش)

توجه: نیاز به Assertion اینکه کارمند پرداخت کننده "مدیر" باشد داریم.

جدول بررسی فروشنده (monitoringSeller):

کد کارمند بازرس (foreign key)(int) – کد فروشنده (foreign key)(int) – تاریخ بررسی (timestamp) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کارمند بازرس، کد فروشنده، تاریخ)

توجه ۱: فرض می‌کنیم هر فروشنده روزانه توسط هر کارمند حداکثر یک بار بررسی می‌شود.

توجه ۲: نیاز به Assertion اینکه کارمند بررسی کننده "بازرس" باشد داریم.

جدول بررسی کالا (monitoringProduct):

کد کارمند بازرس (foreign key)(int) – کد کالا (foreign key)(int) – تاریخ بررسی (timestamp) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کارمند بازرس، کد کالا، تاریخ)

توجه ۱: فرض می‌کنیم هر کالا روزانه توسط هر کارمند حداکثر یک بار بررسی می‌شود.

توجه ۲: نیاز به Assertion اینکه کارمند بررسی کننده "بازرس" باشد داریم.

جدول دریافت کالا (productDelivery):

کد کارمند حامل (foreign key) (int) – کد کالا (foreign key) (int) – تعداد (int) – تاریخ (timestamp) – همگی هیچ مقدار ناپذیراند.

کلید کاندید: (کد کالا، تاریخ)

توجه ۱: فرض می‌کنیم هر کالا روزانه توسط هر کارمند حداکثر یک بار بررسی می‌شود.

توجه ۲: نیاز به Assertion اینکه کارمند دریافت کننده "مسئول حمل و نقل" باشد داریم.

جدول خریدار (buyer):

کد درون سامانه خریداری (C.K) (int) – موجودی حساب فروشگاه (float) – برآیند امتیاز (int) – همگی هیچ مقدار ناپذیراند.

کلید کاندید: (کد درون سامانه خریداری)

جدول نشانی کاربر (buyerLocation):

کد ملی کاربر (foreign key) (int) – کد پستی (C.K) (int) – موقعیت جغرافیای (varchar) – آدرس (varchar) – همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد پستی)

جدول حساب کاربری (buyerProfile):

کد ملی (int) – نام (varchar) – نام خانوادگی (varchar) – تاریخ تولد (varchar) – شماره تلفن همراه (varchar) – ایمیل (varchar) – رمز عبور (varchar) – نام کاربری (varchar) – شماره حساب (varchar) – کد درون سامانه‌ای خریدار (foreign key) (int) – همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد درون سامانه‌ای خریدار)

توجه: اگرچه کد ملی یک identifier برای انسان است، اما حساب با یک کد ملی خالی، لزوماً حساب کاربری یک خریدار نیست.

جدول بازدیدهای اخیر کاربران (recent):

کد کالای مورد بازدید (foreign key) (int) – کد درون سامانه‌ای خریدار (foreign key) (int) – همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد کالای مورد بازدید، کد درون سامانه‌ای خریدار)

جدول نظر(review):

شماره نظر (C.K)(int) – دیدگاه کلی (int) – کارایی (int) – متن نظر (varchar) – کیفیت محصول (int) – ارزش در برابر کیفیت (int) – ارسال به موقع (bit)

تنها «متن نظر» هیچ مقدارپذیر است.

کلید کاندید: شماره نظر

جدول نظرات کاربران(buyerReview):

شماره نظر (foreign key)(int) – کد کالا (foreign key)(int) – کد درون سامانه‌ای خریدار (foreign key)(int)

همگی هیچ مقدارناپذیرند.

کلید کاندید: (شماره نظر، کد کالا، کد درون سامانه‌ای خریدار)

جدول کارت هدیه(giftCard):

شماره کارت هدیه (C.K)(int) – وضعیت (bit) – مبلغ (varchar) – مناسبت (varchar) – مدت اعتبار (varchar)

فقط «مدت اعتبار» هیچ مقدارپذیر است و هیچ مقداری آن به این معناست که اعتبار آن تمام نمی‌شود.

کلید کاندید: (شماره کارت هدیه)

جدول تخصیص کارت هدیه(giftCardBuyer):

کد کارمند مدیر (foreign key)(int) – کد خریدار (foreign key)(int) – کد کارت هدیه (foreign key)(int)

همگی هیچ مقدارناپذیرند.

کلید کاندید: (کد مدیر، کد خریدار، کد کارت هدیه)

جدول علاقه‌مندی‌های خریدار(watchList):

کد کالا (foreign key)(int) – کد خریدار (foreign key)(int)

همگی هیچ مقدارناپذیر اند.

کلید کاندید: (کد کالا، کد خریدار)

جدول درخواست مرجوعی(returningRequest):

شماره درخواست مرجوعی (C.K)(int) – تاریخ ثبت (timestamp) – نتیجه بررسی (varchar) – تاریخ بررسی (date) – متن درخواست (varchar)

«نتیجه بررسی» هیچ مقدارپذیر است و هیچ مقدار بودن آن به این معنی است که هنوز مورد بررسی قرار نگرفته است.

کلید کاندید: (شماره درخواست مرجوعی)

جدول درخواست - کالا مرجوعی (returningRequestProduct):

شماره درخواست مرجوعی (foreign key) (int) - کد کالا (int) - تعداد مرجوعی (int) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (شماره درخواست، کد کالا)

جدول سفارش (orders):

شماره سفارش (C.K) (int) - شناسه تحویل (varchar) - زمان انتظار (int) - تاریخ ثبت (date) - تاریخ تحویل نهایی (date) - نحوه ارسال (varchar) -

نحوه پرداخت (varchar) - وضعیت (varchar) - کد خریدار ثبت کننده سفارش (foreign key) (int) - شماره تراکنش (foreign key) (int) - شماره

سبد (foreign key) (int)

«تاریخ تحویل نهایی» هیچ مقدار پذیر است.

کلید کاندید: شماره سفارش

جدول سفارشات لغو شده (canceledOrder):

کد خریدار (foreign key) (int) - شماره سفارش (foreign key) (int) - تاریخ لغو (timestamp) همگی هیچ مقدار ناپذیراند.

کلید کاندید: (کد خریدار، شماره سفارش)

جدول درخواست های مرجوعی خریداران (returningRequestBuyer):

شماره درخواست مرجوعی (foreign key) (int) - شماره سفارش (foreign key) (int) - کد درون سامانه خریدار (foreign key) (int) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (شماره درخواست مرجوعی، شماره سفارش، کد درون سامانه خریدار)

جدول سبد خرید (cart):

شماره سبد (C.K) (int) - مبلغ کل (float) - شماره خریدار (foreign key) (int) همگی هیچ مقدار ناپذیرند.

کلید کاندید: (شماره سبد)

جدول کالاهای سبدهای خرید (cartProduct):

شماره سبد (foreign key) (int) – کد کالا (foreign key) (int) – تعداد (int)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (شماره سبد، کد کالا)

جدول پرسش (question):

شماره پرسش (C.K) (int) – متن پرسش (varchar) – تاریخ ثبت (timestamp)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (شماره پرسش)

جدول پاسخ گویی خریداران به پرسش ها (buyerAnswer):

کد خریدار (foreign key) (int) – شماره پرسش (foreign key) (int) – تاریخ ثبت پاسخ (timestamp) – متن پاسخ (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد خریدار، شماره پرسش)

جدول پاسخ گویی واحد پشتیبانی به پرسش ها (supportAnswer):

کد پشتیبان (کد کارمندی کارمند پشتیبانی ای که پاسخ این سوال را می دهد) (foreign key) (int) – شماره پرسش (foreign key) (int) – تاریخ ثبت (timestamp) – متن پاسخ (varchar)

همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد پشتیبان، شماره پرسش)

جدول پرسش های خریداران (buyerQuestion):

کد خریدار (foreign key) (int) – شماره پرسش (foreign key) (int) – کد کالا (foreign key) (int) – تاریخ ثبت پرسش (timestamp)

همگی هیچ مقدار ناپذیرند.

کلید کاندید (کد خریدار، شماره پرسش، کد کالا)

بررسی درخواست مرجوعی (returningHandeling):

کد پشتیبان (کد کارمندی کارمند پشتیبانی ای که این درخواست مرجوعی را بررسی می کند) (int)(foreign key)
شماره تراکنش (int)(foreign key) – شماره درخواست مرجوعی (int)(foreign key) – توضیحات (varchar)
همگی هیچ مقدار ناپذیرند.

کلید کاندید: (کد پشتیبان، شماره درخواست مرجوعی)

توجه: پس از بررسی درخواست مرجوعی ، در صورت وارد بودن درخواست ارجاع، تراکنشی مبنی بر عودت وجه به خریدار توسط واحد پشتیبانی تنظیم می گردد و در صورت وارد نبودن، این شماره تراکنش null خواهد بود و علت رد مرجوعی، در قسمت توضیحات آورده می شود.

بخش دوم (بیان نحوه مدل سازی query ها جهت تحقق نیازمندی ها)

Procedure ها:

در فاز صفر برای واحدهای مختلف پروژه نیازمندی های مختلفی را نوشتیم. حال برای محقق شدن این نیازمندی ها لازم است query های مورد نیاز را بنویسیم، ابتدا تعدادی از این نیازمندی ها را بررسی میکنیم:

نیازمندی خریداران: جستجوی کالا (براساس نام کالا، برند)

نیازمندی خریداران: مرتب سازی کالا های جست و جوشده (براساس میزان تخفیف، موجودی، میزان فروش، قیمت، امتیاز)

همانطور که مشاهده می کنید ساختار کوئری مورد نیاز برای تحقق نیازمندی های یک دسته با یکدیگر شباهت زیادی دارند و اغلب در یک ستون با همدیگر تفاوت دارند (مثلا ستون مربوط به مرتب سازی یا ستون مربوط به معیار جست و جوی کالا) لذا به جای نوشتن کوئری های ساده و مستقل جهت تحقق هر کدام، از procedure ها استفاده کردیم که همانند function ها ورودی خروجی دارند و بسته به ورودی به و به کمک عبارات شرطی می توان حالت های مختلف نیازمندی ها را محقق کرد.

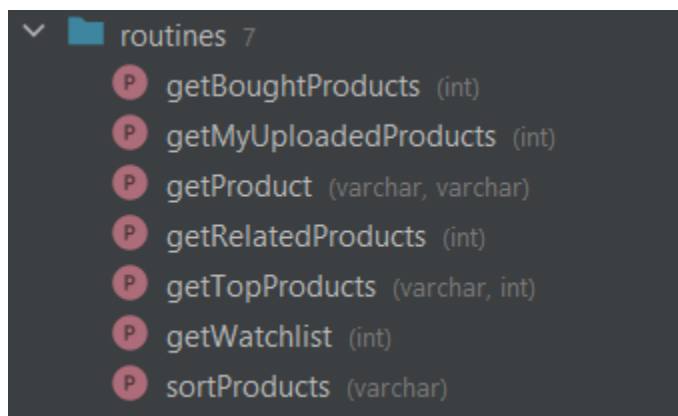
مثال:

تحقق مرتب سازی بر حسب ستون های مختلف را به صورت زیر به کمک sortProducts درآوردیم:

```
create procedure sortProducts(  
    in theSortColumn varchar(10)  
)  
begin  
    if (isnull(theSortColumn)) then  
        select * from product;  
    else  
        select *  
        from product  
        order by CASE  
            WHEN theSortColumn = 'price' THEN price  
            WHEN theSortColumn = 'off' THEN off  
            WHEN theSortColumn = 'available' THEN available  
            WHEN theSortColumn = 'sold' THEN sold  
            WHEN theSortColumn = 'privilege' THEN privilege  
            WHEN theSortColumn = 'score' THEN score  
            END DESC;  
    end if;  
end;
```

همانطور که ملاحظه می کنید بسته به رشته ورودی که تعیین کننده ی ستونی است که بر حسب آن مرتب سازی شود، عملیات مرتب سازی در یک کوئری انجام می شود و ۶ کوئری مختلف همگی در یک procedure بسته بندی شده اند و نیز گسترش دادن و اضافه کردن ستون های جدید برای مرتب سازی در این حالت نیز بسیار ساده تر بوده و با اضافه کردن تنها یک خط کد محقق می شود و نیاز به نوشتن کوئری جدید نیست.

بدین ترتیب با نوشتن ۷ عدد procedure یا همان routine به پیاده‌سازی کویری‌های موردنیاز جهت تحقق نیازمندی‌ها پرداختیم.



Procedure شماره اول جهت برگرداندن کالاهای خریده شده از یک فروشنده است (ورودی کد فروشنده فروشنده است).

Procedure شماره دوم جهت برگرداندن کالاهای بارگذاری شده یک فروشنده است (ورودی کد فروشنده فروشنده است).

Procedure شماره سوم جهت برگرداندن کالاهایی با یک "برند" یا "نام" خاص است. (دو ورودی دارد و یکی "برند" است و یکی "نام" است و اگر :

هر دو null باشد در خروجی همه کالاها برگردانده می شود

و در صورت null نبودن برحسب آن جست‌وجو شده و بازگردانی می‌شود.

Procedure شماره چهارم جهت برگرداندن کالاهای مرتبط با یک کالا خاص استفاده می‌شود (در ورودی شماره کالا را می‌گیرد).

Procedure شماره پنجم کالاهای برتر را به این گونه برمیگرداند که یک ورودی می‌گیرد که ستون مورد معیار است و دیگری یک ترشهولد است که آن کالاهایی که ستون مورد معیار، مقدارش از ترشهولد مدنظر بیشتر است را برمی‌گرداند.

Procedure شماره ششم جهت برگرداندن لیست کالاهایی است که خریدار مشاهده می‌کند و در ورودی کد خریدار را می‌گیرد.

Procedure شماره هفتم جهت مرتب‌سازی کالاها بر اساس یک ستون خاص است و کالاهای مرتب شده را برمی‌گرداند.

بخش سوم (بیان دیدهایی که هرکاربر ممکن است نیاز داشته باشد)

بررسی دیدهای مفید کاربران مختلف

توجه: برخی واحدها اعم از واحد پشتیبانی، به تمام آنچه مرتبط با بخششان است (مثلا برای واحد پشتیبانی همه پرسشها نیاز است ارایه شود) نیاز دارند لذا اگر دید خارجی ای تعریف شود، این دید تعریف شده روی لایه ادراک، صرفا یک indirection ایجاد کرده و به نوعی دگرنام است (این لایه indirection سبب کمک به حفظ بیشتر امنیت دیتابیس می کند -> از طریق کمینه کردن اطلاع از ساختارهای واقعی و نام های واقعی جداول)

بنده برای پرهیز از پیچیدگی بیشتر، از بیان دیدهایی که صرفا به منظور ایجاد indirection می شوند (یعنی تا حد خوبی مشابه خود جداول سطح ادراک اند) در اینجا پرهیز کردم و تنها مواردی که به نظم تعریف کردنشان مازاد بر indirection مفید است را بیان کردم.

واحد مدیریت:

دید شماره ۱: لیست کارمندانی که حقوقشان پرداخت نشده است.

این view در پایگاه داده نام unpaidclerks را دارد.

واحد بازرسی:

دید شماره ۱: لیست فروشندگانی که تایید نشده اند.

این view در پایگاه داده نام deniedSellers را دارد.

واحد حمل و نقل:

دید شماره ۱: لیست کالاهایی که امروز باید از فروشنده تحویل بگیرند.

نام این view در پایگاه داده getFromSellerOrders است.

دید شماره ۲: لیست سفارش های در حال ارسال

نام این view در پایگاه داده sendingOrders است.

فروشنده:

دید شماره ۱: لیست کالاهایی که بارگذاری کرده است.

دید شماره ۲: لیست کالاهای خریداری شده از این فروشنده

اگر بخواهیم چنین دیدهایی را تعریف کنیم، لازم است که آن هارا برای هر فروشنده تعریف بکنیم و خب این یعنی تعریف تعداد زیادی دید! و مقرون به صرفه نیست (استفاده زیاد از حافظه) و همچنین از آنجا که این دیدها مدام در حال تغییر اند و فروشنده ممکن است گاه به گاه به آن نیاز داشته باشد تصمیم گرفتیم که برای تحقق این نیازمندی ها از procedure استفاده کردم که در صورت نیاز call شود و مطالبات مدنظر استخراج شده و بازگردانی شود.

برای این دو نیازمندی دو procedure به نام های getMyUploadedProducts(int sellerId) و

getBoughtProducts(int sellerId) را تعریف کردیم.

بخش چهارم (بیان ابزار مورد استفاده در مدیریت پایگاه داده)

برای مدیریت پایگاه داده ما از ابزار MySQL ورژن 5.7.34 استفاده کردیم، علت انتخاب این ابزار:

- 1 - آشنایی بودن هر دو عضو گروه با این ابزار
- 2 - توانایی این ابزار مدیریت پایگاه داده جهت تحقق کامل نیازمندی‌های پروژه
- 3 - Open source بودن و Free بودن برای کاربری پژوهشی
- 4 - High Performance
- 5 - Data Security و ...

بخش پنجم (بیان Assertion ها - دامنه‌های صفات - محدودیت‌های جامعیتی)

Assertion ها:

توجه در MySQL ابزاری به نام Assertion به آن صورت که در درس تعریف شد نداریم و در عوض برای تحقق Assertion ها مستندات مختلف به استفاده از Check(condition) در تعریف جداول، ارجاع داده اند و ما نیز چنین کاری کرده ایم.

- 1 - نباید در اثر خرید کالاها، موجودی کالاها منفی شود.
- 2 - در هنگام تسویه حساب، کارمند تسویه حساب کننده باید مدیر باشد
- 3 - در هنگام پرداخت پرداخت حقوق، کارمند پرداخت کننده باید مدیر باشد
- 4 - در هنگام بررسی فروشنده، کارمند بررسی کننده باید بازرس باشد.
- 5 - در هنگام بررسی کالا، کارمند بررسی کننده باید، بازرس باشد.
- 6 - کارمند حامل باید سمت حمل و نقل داشته باشد.

بخش ششم (بیان رهاناها و Job های مورد نیاز)

Trigger ها:

- (buyingScore): در اثر تراکنش خرید موفق امتیاز خریدار باید افزایش پیدا کند
- (productSoledAvailableUpdate) در اثر تراکنش خرید موفق میزان فروش کالا افزایش پیدا می کند و موجودی کالای فروشنده کاهش پیدا می کند.
- (reviewScore) اگر فروشنده اطلاعات یک کالا را تغییر داد، مقدار صفت «تغییر بعد از آخرین بررسی» به ۱ تغییر می کند.
- (productUpdate) اگر خریداری نظر یا پاسخی ثبت کرد امتیازش افزایش یابد.

بقیه رهاناها که مشابه موارد قبلی بودند را پیاده سازی نکردیم:

- اگر فروشنده ای اطلاعات اساسی اش (مثل اسکن شناسنامه، پروانه کسب و...) را تغییر داد مقدار صفت «تغییر بعد از آخرین بررسی» به ۱ تغییر کند.
- اگر «وضعیت تایید» یک فروشنده 'confirmed' باشد، دیگر فروشنده امکان تغییر اطلاعات اساسی اش را نخواهد داشت. این کار را می توان بدون رهانا انجام داد. یعنی هر گاه «وضعیت تایید» یک فروشنده 'confirmed' شد، گزینه به روزرسانی اطلاعات برایش غیرفعال شود.
- اگر «وضعیت تایید» کالایی 'confirmed' باشد، فروشنده می تواند تنها موجودی، تخفیف و قیمت کالا را تغییر دهد. این کار را نیز می توان بدون رهانا انجام داد. یعنی هر گاه «وضعیت تایید» یک کالا 'confirmed' شد، گزینه به روزرسانی اطلاعات برایش غیرفعال شود.
- اگر کارمندی عزل شد، سمتش به 'dismiss' تغییر پیدا کند.
- در اثر لغو سفارش، امتیاز خریدار باید کاهش پیدا کند.

موارد زیر برخی از مواردی هستند که می توان به صورت رهانا آنها را پیاده سازی کرد. اما به دو دلیل چنین کاری به صرفه نیست. زیرا اولاً با به روزرسانی نکردن سریع اطلاعات، ضربه ای به پایگاه داده ها وارد نمی شود و ثانیاً هزینه به روزرسانی بالاست. به همین دلیل ما این به روزرسانی ها را با رهانا پیاده سازی نکرده ایم. برای اعمال این تغییرات job هایی تعریف می شوند که به صورت متناوب اجرا می شوند و تغییرات دلخواه را اعمال می کنند. نمونه هایی از این موارد در زیر آمده است:

- در پایان هر هفته، با بررسی جدول «دریافت کالا»، مشخص می شود که هر فروشنده چه تعداد از چه کالاهایی تحویل داده است. می توان محاسبه کرد که قیمت کل این کالاها چقدر است. شرکت درصدی از این پول را برای خودش برمی دارد و ما بقیه را به حساب فروشنده واریز می کند.
- می خواهیم اگر درخواست مرجوعی زده شد و درخواست وارد بود، امتیاز فروشنده کاهش پیدا می کند. این کار را به صورت روزانه انجام می دهیم. مثلاً در پایان هر روز، با بررسی جداول «درخواست های مرجوعی»، «درخواست - کالای مرجوعی» و «کالا» مشخص می شود که هر فروشنده چه تعدادی از کالاهایش برگردانده شده اند. از این طریق محاسبه می کنیم که امتیاز وی چقدر باید تغییر کند.
- در پایان هر هفته، امتیازهای کالاها را به روزرسانی می کنیم. به این صورت که جدول نظرات را بررسی می کنیم و از روی آن میانگین امتیازی که کاربران به هر کالا داده اند را محاسبه کرده و مقدار «میانگین امتیاز کاربران» در جدول کالا را به روزرسانی می کنیم.

- در پایان هر هفته امتیاز فروشنده را با بررسی جدول سفارشات و سفارشات مرجوعی و... به روزرسانی می‌کنیم. هر چه فروش بیشتر و مرجوعی کمتر باشد، امتیاز فروشنده بیشتر خواهد بود. همچنین صفت وضعیت ارائه خدمات، که نشان می‌دهد این فروشنده چند درصد از سفارشاتش را به موقع آماده کرده، نیز هفتگی به روزرسانی می‌شود.

بخش هفتم (بیان میزان مشارکت اعضای گروه در پیاده‌سازی بخش‌های مختلف)

- ✓ فاز صفر (پروپوزال)
در این فاز هر دو عضو گروه با همکاری کامل یکدیگر پروپوزال پروژه را آماده کردند.
 - ✓ فاز اول (نمودار EER)
در این فاز مشابه فاز قبلی دو عضو گروه با گفت‌وگو و مشارکت کامل به مدل‌سازی نمودار EER پرداختند اما رسم نمودارها را خانم سنا نادعلی انجام دادند.
 - ✓ فاز دوم (طراحی منطقی جداول - دیدهای لازم و انتخاب ابزار مناسب)
این فاز هم مشابه فاز قبلی با گفت‌وگو و مشارکت کامل اعضا جلو رفت اما نوشتن مستندات را آقای امیرحسین رستمی انجام دادند.
 - ✓ فاز سوم (پیاده‌سازی + محدودیت صفات - رهاناها - محدودیت‌های جامعیتی و...)
این فاز عمیقاً توسط مشارکت توأم اعضای گروه انجام شد و وظایف ریزدانه به شرح زیر است:
 - 1 - استخراج موارد لازم: توسط گفت‌وگو و جلسات مکرر -> سنا نادعلی و امیرحسین رستمی
 - 2 - پیاده‌سازی تریگرها - نوشتن دستورات SQL ساخت جداول - پرکردن جداول با اطلاعات تقریباً با معنی -> سنا نادعلی
 - 3 - پیاده‌سازی دیدها - procedure ها و Assertion ها (که به صورت check در آمدند) -> امیرحسین رستمی
 - 4 - بررسی بخش‌های پیاده‌سازی شده هم‌دیگر و دادن فیدبک -> سنا نادعلی و امیرحسین رستمی
 - 5 - تکمیل مستندات فاز دوم و سوم -> سنا نادعلی و امیرحسین رستمی
 - 6 - پیاده‌سازی ربات تلگرام + اتصال داشبورد Grafana به پایگاه داده -> امیرحسین رستمی
- نکته: هر دو عضو گروه، در طی پروژه نظارت کامل بر بخش‌های هم‌دیگر داشتند.

بخش هشتم) بیان رابط‌های کاربری پیاده‌سازی شده جهت اتصال به پایگاه داده و انجام پرس‌وجوهای لازم)

UI و تکنولوژی‌های مورد استفاده در واسط کاربری:

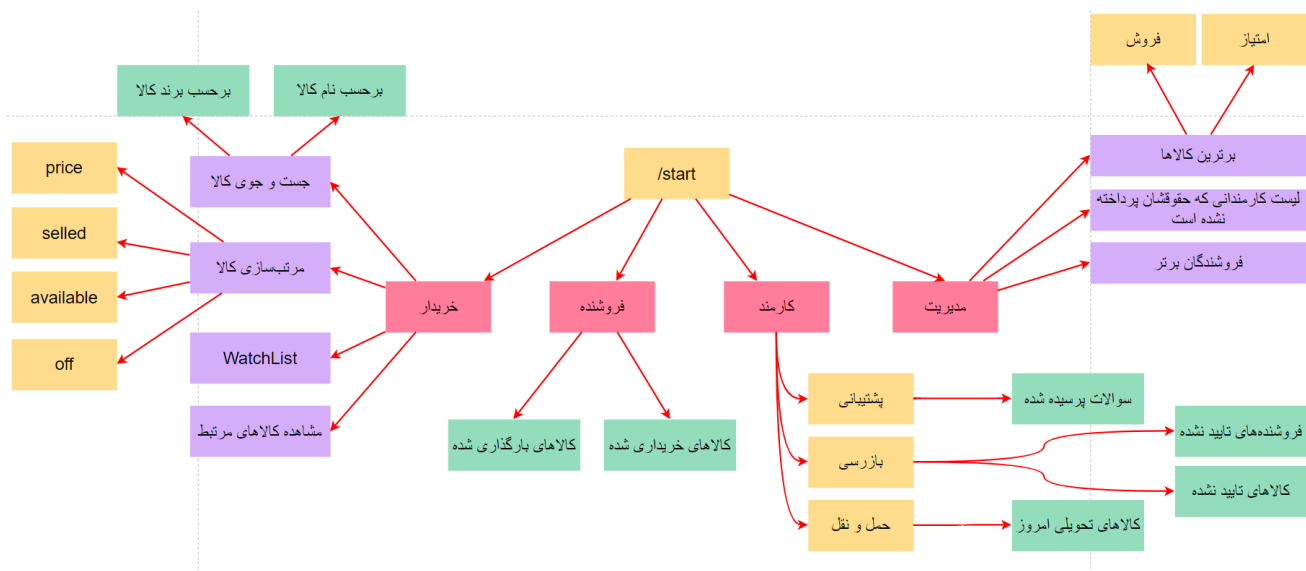
1 - Grafana

این ابزار یک پروژه open source است که داشبوردها و پنل‌های مختلف جهت ویژوالایز کردن نتایج و پژوهش‌ها روی دیتابیس را به ما می‌دهد. به کمک این ابزار میتواند نتایج درخواست‌های مختلف را کنار همدیگر گذاشت و برای ویژوالایز کردن نتایج time-series بسیار مناسب است. بنده به منظور داشتن یک پنل مدیریتی مبتی بر وب این داشبورد را بالا آوردم.

2 - Telegram bot

برای محقق کردن نیازمندی‌های نوشته شده ما یک ربات تلگرام نوشتیم (from scratch) که ساختار شجره‌ای منوهایش به شکل زیر است:

این ربات deploy نشده است و در هنگام تحویل جهت ارایه روشن می‌شود.



نکات:

- جداول حاصل از درخواست‌ها در قالب file (تصویر با PNG) در پاسخ برگردانده می‌شود.
- ربات به زبان JAVA زده شده است.
- کویری ها

1 - ابتدا به کمک درایور JDBC به پایگاه داده زده می‌شوند

2 - سپس نتیجه حاصل ابتدا توسط parser ای که نوشتیم تبدیل به html می‌شود

3 - سپس فایل html توسط آداپتور HtmlImageGenerator به تصویر با فرمت png تبدیل می‌شود

4 - و در نهایت به متقاضی در قالب file (جهت حفظ کیفیت و پرهیز از اعمال compression تلگرام) ارسال می‌شود.

آیدی ربات تلگرام @db_projectbot است.