به نام خدا



فاز سوم

نيمسال اول سال تحصيلي ۱۴۰۲–۱۴۰۱

اعضای گروه: هیربد بهنام، عرفان مجیبی، علی نظری

شمارههای دانشجویی:

هیربد بهنام ۹۹۱۷۱۳۳۳

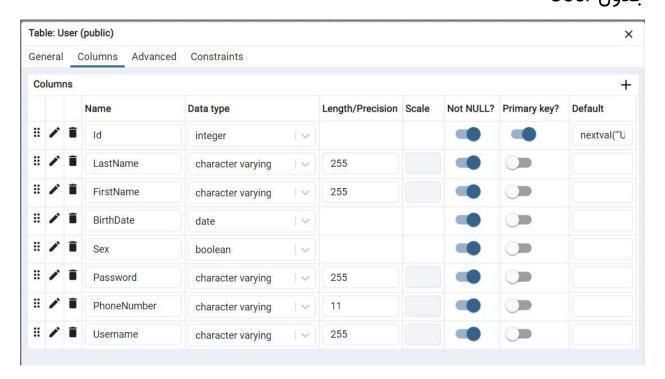
عرفان مجیبی ۹۹۱۰۵۷۰۷

علی نظری ۹۹۱۰۲۴۰۱

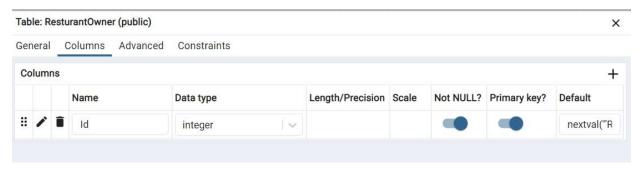
ایمیلها:

میربد بهنام <u>mojibierfan@gmail.com</u> عرفان مجیبی عان مجیبی <u>ali.nazari.8102@gmail.com</u> علی نظری

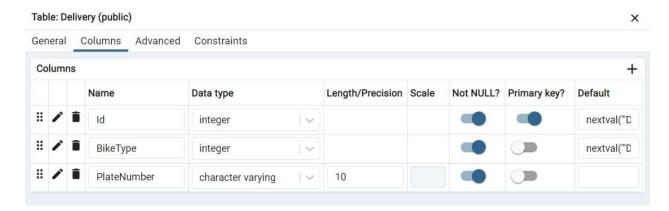
جداول طراحی شده: جدول User



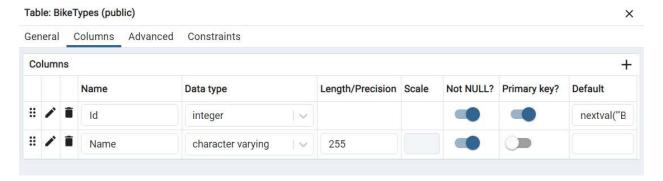
جدول Customer



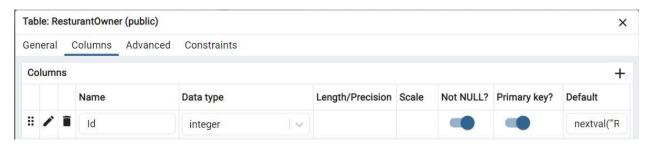
جدول Delivery



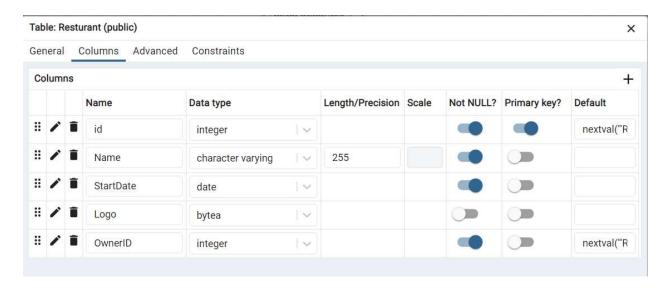
جدول BikeTypes



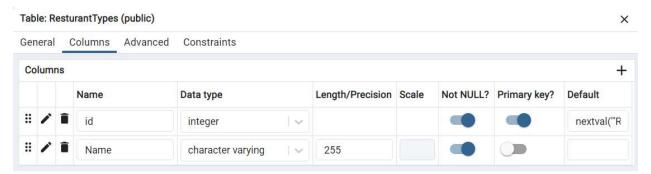
جدول RestaurantOwner



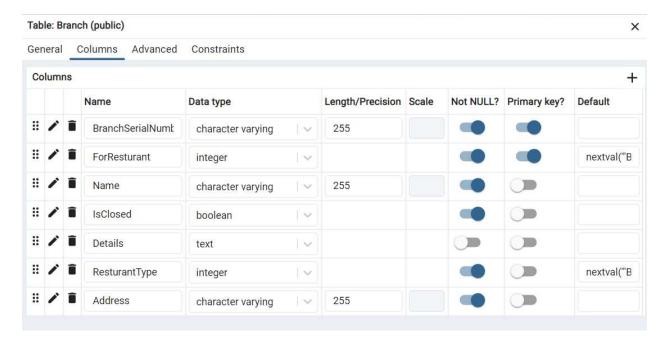
جدول Restaurant



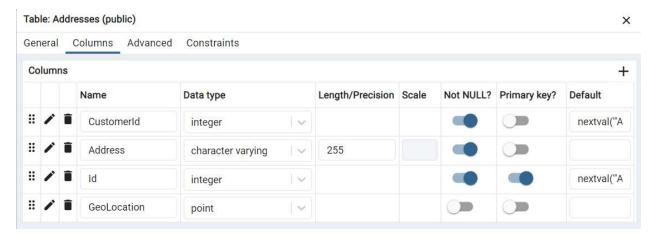
جدول RestaurantTypes



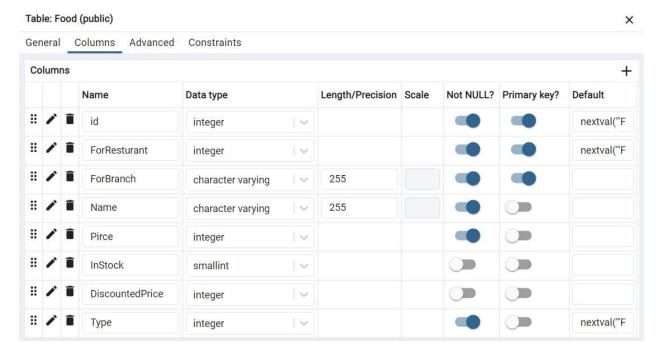
جدول Branch



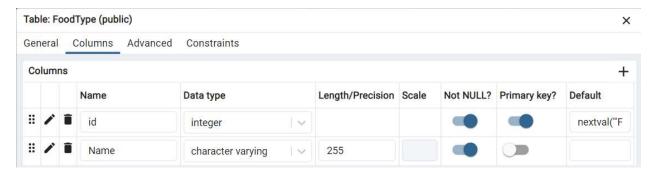
جدول Addresses



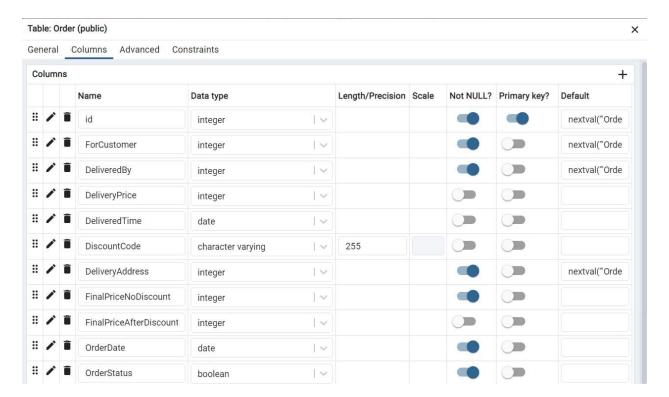
جدول Food



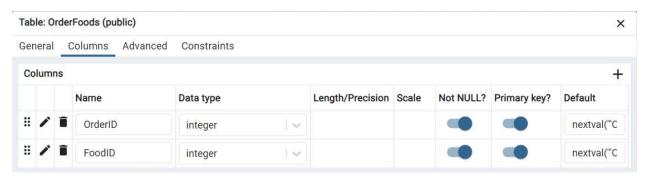
جدول FoodType



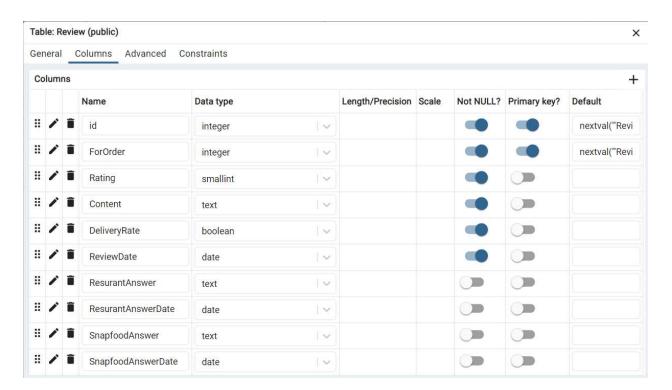
جدول Order



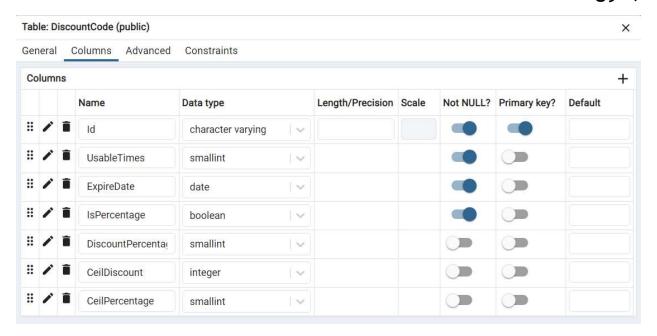
جدول OrderFoods



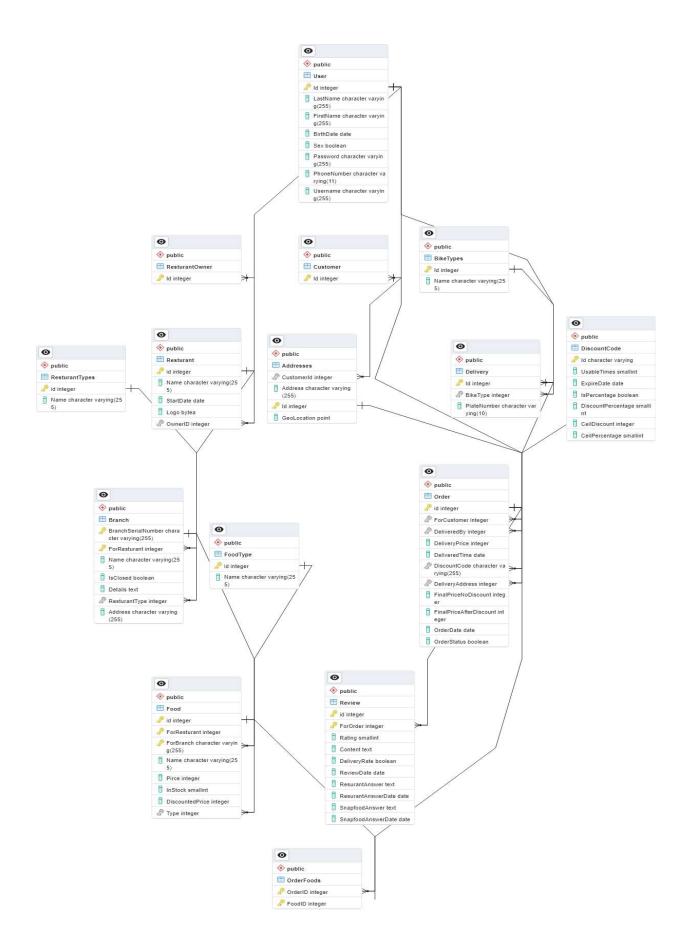
جدول Review



جدول DiscountCode



همه جداول در یک نما در صفحه بعد قابل مشاهده است.



دامنهها:

به طور کلی دامنه ستونهای مختلف بر اساس نوع و محدودیتهایی که بر آنها گذاشته شده در جداول در تصویر مشخص است. مثلا دادههای integer در محدوده مورد پشتیبانی postgres هستند. Character varying ها رشتهها هستند که طول آنها در تصویر در جلوی آنها نوشته شده است. Boolean ها نیز مقدار true-false میتوانند به خود بگیرند. محدودیتهای null شدن (هیچمقدار پذیری) در تصویر قابل مشاهده است.

مواردی که شاید ابهام داشته باشد یا محدودیتی دستی بر آنها اعمال شده که در تصویر قابل مشاهده نیست(اما در کد SQL هست) شامل اینهاست:

- ستون GeoPoint در جدول Restaurant برای نمایش طول و عرض جغرافیایی است.
 - DiscountCode در جدول DiscountPercentage که درصد کد تخفیف است و باید بین صفر تا صد باشد.
- CeilPercentage در جدول DiscountCode که سقف درصد است و باید بین صفر تا صد باشد.
 - Rating در جدول Review که باید بین 1 تا 5 باشد.

اعمال جامعیت ارجاعی هنگام حذف:

- با حذف یک رستوران، شعبه و ارجاعات به آن نیز حذف میشود (On) Delete Cascade).
- با حذف یک Cutomer همه آدرسهای او در جدول Addresses و جاهایی که استفاده شدهاند حذف میشود(On Delete Cascade).

- با حذف یک شعبه غذای آن از جدول Food و استفادههایش حذف میشود (On Delete Cascade).
- با حذف یک سفارش (Order) نظرات مربوط به آن هم حذف میشود(On Delete Cascade).
 - در سایر موارد حذف باید ارور داده شود.

نحوه به دست آوردن جدولها:

در مرحله اول، کار نگاشت موجودیتهای قوی را انجام دادیم. موجودیتهای قوی در طراحی ما، شامل سفارش، مشتری، فرد، پیک، صاحب رستوران، رستوران و کد تخفیف است. سپس ویژگیهای ساده این موجودیتها را به جدول آنها اضافه میکنیم. کلید اصلی را هم از بین این موارد مشخص میکنیم. صفتهای ترکیبی را هم به صفتهای ساده میشکانیم و به عنوان field به جدول اضافه میکنیم. دقت داریم که در این مرحله برای صفتهای چند مقداری، کاری انجام نمیدهیم. در مورد صفتهای مشتق هم توجه داریم که به هیچ عنوان فیلدی در نظر نمیگیریم.

در مرحله دوم، به سراغ نگاشت موجودیتهای ضعیف میرویم که شامل نظر، غذا و شعبه است. نظر موجودیت ضعیف وابسته به سفارش است. غذا موجودیت ضعیف وابسته به سفارش و رستوران است. شعبه هم موجودیت ضعیف وابسته به رستوران است. برای هر موجودیت ضعیف یک جدول ایجاد میکنیم. ویژگیهای موجودیت به جدول آن اضافه میشود. کلید اصلی موجودیتهای قوی که هر موجودیت ضعیف به آن وابسته است را به عنوان کلید خارجی به موجودیت ضعیف اضافه میکنیم. کلید جزئیای هم برای هر موجودیت ضعیف در نظر میگیریم.

در مرحله سوم به سراغ نگاشت رابطه N:1 میرسیم. در این حالت کلید اصلی طرف 1 به عنوان کلید خارجی به طرف N اضافه میشود که برای رابطهها N:1 موجود، اینکار انجام شده است. صفتهای رابطه را هم به طرف N اضافه میکنیم. رابطههایی که این موارد برای آنها انجام شده است، شامل سفارش

و مشتری، سفارش و پیک، رستوران و صاحب رستوران، رستوران و شعبه، شعبه و غذا، کد تخفیف و سفارش میشود.

در مرحله چهارم به سراغ نگاشت رابطههای N:M میرویم. برای این حالت، یک جدول جدید به نام pivot table ایجاد میکنیم. کلیدهای اصلی از دو طرف رابطه به عنوان field به این table اضافه میشود. صفتهای رابطه هم به عنوان field اضافه میشود. در طراحی ما، رابطه سفارش و غذا فقط رابطه M:N است.

در مرحله پنجم به سراغ نگاشت رابطههای 1:1 میرویم. در طراحی ما رابطه سفارش و نظر به این شکل است. در این مرحله سه رویکرد وجود دارد. یا کلید یکی را به طرف دیگر اضافه کنیم. یا هر دو موجودیت را ترکیب کنیم. یا اینکه pivot table تشکیل دهیم. در طراحی ما، از آن جا نظر موجودیت ضعیف وابسته به سفارش به حساب میآید، در مرحله دوم کلید اصلی سفارش را به جدول نظر اضافه کردیم و در نتیجه در این مرحله لازم نبود کار خاصی انجام دهیم. در نتیجه صفتهای رابطه هم به سمت نظر برده شدند.با این عمل مقدار null ها و duplicate ها کمتر میشود.

در مرحله ششم به سراغ نگاشت صفتهای چند مقداره میرویم. به عنوان مثال آدرس مشتری نمونهای از صفت چند مقداری است. برای این حالت یک جدول جدید میسازیم که کلید اصلی موجودیت اصلی را دارد و ستون شامل صفت هم دارد. سپس در هر row این table جدید، تنها یکی از مقدارها میآید و ممکن است ستون آیدی یکسان باشد بین چند row و در هر row یکی از مقدارهای موجود برای یک صفت بیاید.

در مرحله هفتم به نگاشت روابط چندگانه میرسیم. برای این حالت، یک جدول جدید ایجاد کردیم که کلید اصلی همه طرفهای رابطه به عنوان کلید خارجی در نظر گرفته میشوند و در جدول جدید قرار میگیرند. صفتهای رابطه هم به این جدول اضافه میشوند.

در مرحله آخر به نگاشت روابط is a میرسیم. ما دو تا از این رابطهها داشتیم. یکی رابطه فرد و پیک و مشتری و صاحب رستوران و دیگری هم رابطه بین کد تخفیف و کد تخفیف عددی.

برای اولین رابطه is a که بین فرد و مشتری و پیک و صاحب رستوران است،از این روش استفاده کردیم که جدول پدر را نگه داشتیم و به ازای همه فرزندان یک جدول ایجاد کردیم که کلید اصلی پدر خود را دارند و همچنین صفتهای ویژه خودشان را در جدول خودشان دارند و همه صفتهای مشترک در جدول پدر آمده است.

برای دومین رابطه is a که داریم که بین کد تخفیف و کد تخفیف درصدی و کد تخفیف عددی است، از روش دیگری استفاده کردیم. از این روش استفاده کردیم که در کل یک table در نظر گرفتیم و در آن کلید اصلی و همه ویژگیهای مشترک و غیر مشترک بین فرزندان را آوردیم. یک صفت هم در آن داریم که هر ندای که هر داریم که علی با این صفت مشخص میکنیم که هر row متعلق به کدام نوع است.

به این ترتیب همه مرحلهها طی میشود و طراحی ما کامل میشود.