

۱- سپس طرح رابطه های زیر را در نظر بگیرید

Part (part_id, name, cost)

Subpart (part_id, subpart_id, count)

این روابط بیان می کنند که اگر چندتایی (3 , p2 , p1) عضو رابطه subpart باشد، به این معنی است که قطعه p2 به طور مستقیم در ساخت قطعه p1 استفاده شده است و p1 تعداد ۳ قطعه p2 را در خود دارد. با توجه به اینکه خود p2 نیز می تواند از تعداد قطعات دیگری تشکیل شده باشد، پرس و جوی بازگشتی (با استفاده از اسلایدهای ۵۱ تا ۵۳ فصل پنجم) به زبان SQL در پاسخنامه بنویسید که تمامی قطعات به کار رفته در ساخت قطعه با آی دی p100 را به همراه تعداد استفاده کل آن ها در این قطعه نمایش دهد.

```
WITH RECURSIVE included_parts(subpart_id, part_id, quantity) AS
(
    SELECT
        subpart_id,
        part_id,
        count
    FROM Subpart
    UNION ALL
    SELECT
        p.subpart_id,
        p.part_id,
        p.count
    FROM included_parts pr, Subpart p
    WHERE p.part_id = pr.subpart_id
)
SELECT subpart_id, SUM(quantity) AS total_quantity
FROM included_parts
GROUP BY subpart_id
WHERE part_id=p100;
```

۲- قرار است برای یک شرکت بیمه خودرو یک پایگاه داده طراحی کنیم؛ هر مشتری (راننده) این شرکت بیمه حداقل یک خودرو دارد. در این پایگاه داده اطلاعات مشتریان، خودرو های بیمه شده آن ها و اطلاعات تصادفات رخ داده ذخیره

خواهد شد. در هر تصادف ممکن است یک یا چند خودرو درگیر باشند و برای هر تصادف باید گزارشی از خودرو های درگیر، و هزینه تعمیر برآورد شده برای هر مشتری جهت پرداخت از طرف شرکت بیمه قابل مشاهده باشد؛ با توجه به خواسته های بیان شده به سوالات زیر پاسخ دهید:

(الف) به نظر شما منظور از « قابل مشاهده بودن گزارش تصادفات » در متن فوق چیست؟

ب) جداول مورد نیاز پایگاه داده با فیلد هایی که فکر می کنید با توجه به خواسته های مطرح شده برای ذخیره داده ها ضروری هستند به فرمت زیر نمایش دهید؛ کلید یا مجموعه کلید اصلی (Primary key) را نیز با کشیدن یک خط زیر نام فیلد یا فیلد های مورد نظر در هر جدول مشخص کنید.

نام جدول (فیلد اول ، فیلد دوم ، فیلد سوم ، ...)

(الف)

یعنی هر مشتری باید اجازه ی **select** روی جدول یا **view** مربوط به گزارش تصادفات را داشته باشد .

(ب)

مشتریان (کدملی ، نام ، نام خانوادگی ، شماره تلفن ، آدرس)

خودرو (شماره شناسایی خودرو ، نوع وسیله نقلیه ، نام وسیله نقلیه ، مدل ، سال ساخت ، کاربری ، پلاک)

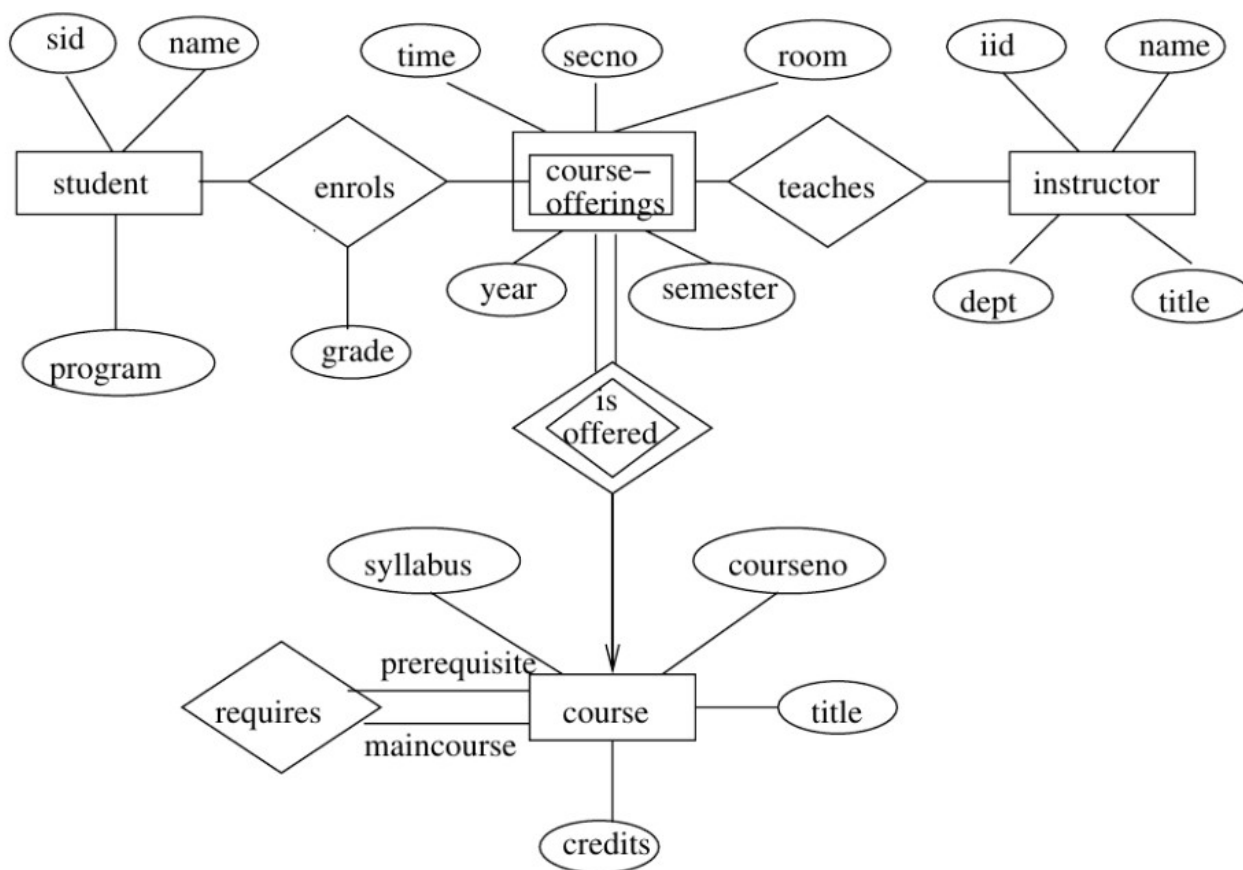
بیمه شدگان (کد بیمه نامه ، کدملی شخص بیمه کننده ، شماره شناسایی وسیله نقلیه بیمه شده)

تصادفات (کد تصادف ، شماره شناسایی خودرو ، هزینه برآورد شده)

توجه : منظور از شماره شناسایی خودرو (**Vehicle Identification Number** به اختصار: **VIN**) یک کد ۱۷ رقمی منحصر بفرد است، که شامل اعداد از ۱ تا ۹ و حروف الفبای انگلیسی به جز حروف **O** و **Q** که برای جلوگیری از تشابه حذف شده اند. این کد که توسط سازمان بین المللی استاندارد سازی تعریف می شود، جهت شناسایی موتور وسایل نقلیه سبک و سنگین، موتورسیکلت ها و موتورهای گازی، مورد استفاده قرار می گیرد.

این کد را معمولاً زیر لبه شیشه جلو سمت راننده، یا بر روی پلاک پیشرا نه خودرو، می توان مشاهده کرد.

۳- ابتدا نمودار ER زیر را بررسی نموده و ضمن بیان منظور طراحی و شرح مختصر نحوه کارکرد کلی سیستمی که از این پایگاه داده برای ذخیره داده استفاده میکند، برای آن یک پایگاه داده رابطه ای طراحی کنید. (نوشتن جداول و کلید های اصلی و خارجی نظیر سوال ۲-ب لازم و کافیست)



این طراحی به منظور پیاده سازی فرایند ثبت نام دانشجو در دروس ارائه شده در هر ترم انجام گرفته است. به این صورت که هر دانشجو با توجه به دروسی که در آن ترم ارائه شده و همچنین رعایت پیشنیاز های دروس میتواند نسبت به اخذ آن اقدام کند. به علاوه برای هر درس ارائه شده ، استاد ارائه دهنده ی آن نیز مشخص است .

Student(sid , name , program);

Enrols(sid(FK→Student) , {courseno , secno , semester , year} (FK→Course_Offerings) , grade);

Course_Offerings(courseno , secno , semester , year , room , time);

Teaches(iid(FK→Instructor) , {courseno , secno , semester , year} (FK→Course_Offerings));

Instructor(iid , name , dept , title);

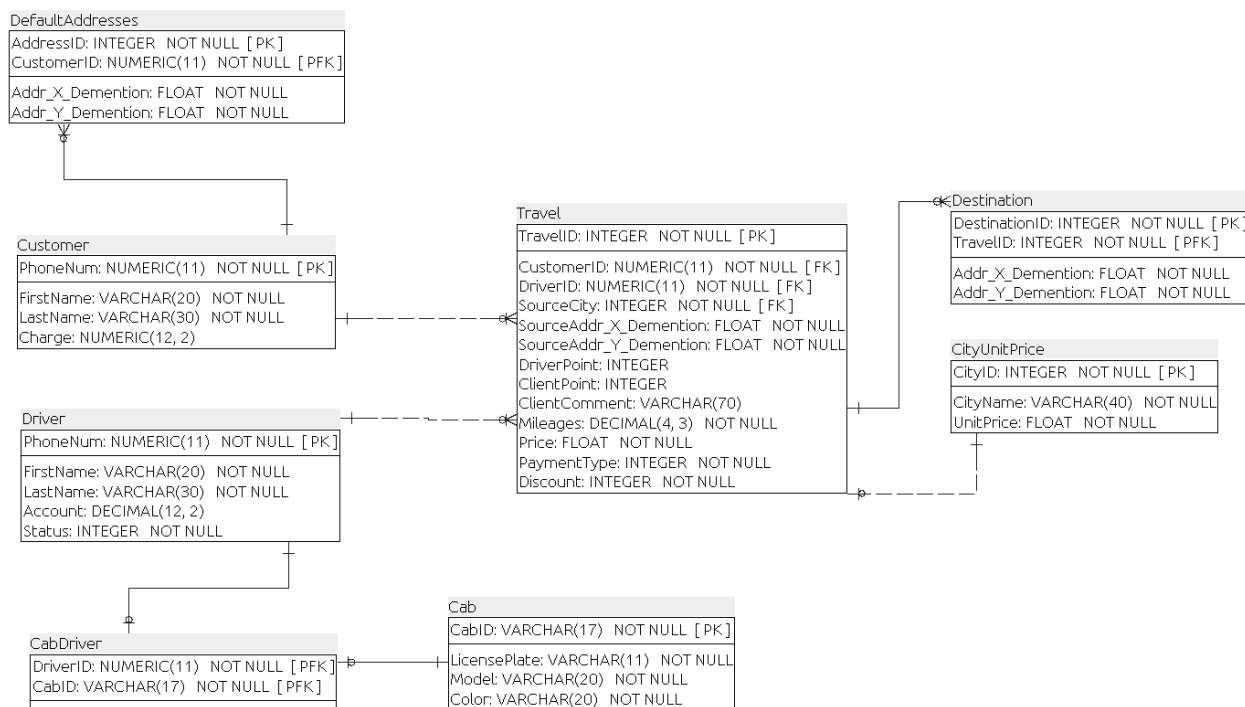
Course(courseno , title , syllabus , credits);

Prereq(courseno(FK→Course) , prereqno(FK→Course));

۴- فرض کنید می خواهیم یک سیستم تاکسی یاب آنلاین برای جابجایی مسافری طراحی کنیم که امکانات زیر را داشته

باشد:

- هر فرد ابتدا در این سامانه با ارائه اطلاعات شناسایی مانند نام و نام خانوادگی و شماره تلفن همراه ثبت نام می کند. هر شماره تلفن همراه بعنوان یک مشتری جدید در نظر گرفته می شود. هر مشتری با استفاده از این سامانه می تواند بارها درخواست سفر دهد و سفر انجام دهد. هر سفر می تواند از چند بخش تشکیل شود و بین هر بخش می تواند مدتی توقف وجود داشته باشد. مثلا مشتری از دانشگاه صنعتی اصفهان به دروازه شیراز رفته ۱۵ دقیقه آنجا توقف داشته سپس به سمت پل شهرستان حرکت می کند. در ساده ترین حالت سفر از یک بخش تشکیل شده و توقفی هم ندارد. هر مشتری می تواند چند نقطه (با مختصات مشخص) را بعنوان آدرس های پر مصرف برای خود ذخیره کند.
- قیمتها بر اساس فاصله جغرافیایی نقاط سفر محاسبه می شود و از شهری به شهری دیگر متفاوت خواهد بود. هزینه سفر می تواند بصورت نقدی و یا الکترونیکی پرداخت شود. همچنین مشتری می تواند مبلغی را به اکانت خود شارژ نماید تا در هر سفر مجبور به پرداخت الکترونیکی نشود.
- مشتری در انتهای سفر می تواند نظر خود را در مورد سفر در قالب امتیاز عددی، انتخاب متن از پیش تعیین شده و یا نوشتن نظر بصورت متنی بیان کند.
- رانندگان نیز در این سامانه ثبت نام کرده و به درخواستهای مشتریان پاسخ می دهند. رانندگان هم می توانند به مشتریان خود امتیاز بدهند. مشخصات خودرو ها نیز در این سامانه ثبت می شود.
- در بازه های زمانی خاص مدیر سیستم ممکن است درصدی از هزینه را به مشتریان تخفیف بدهد. همچنین مدیر سیستم ممکن است بر اساس تعداد سفر یا میزان مبلغ و یا مسافت طی شده برای مشتری و یا راننده جوایزی نقدی در نظر بگیرد که به حساب اشان شارژ می شود. تعیین نرخ پایه هر شهر برای هر کیلومتر سفر، فعال و غیر فعال کردن رانندگان نیز توسط مدیر سیستم انجام می شود.



۵- الف) جداول طراحی شده خود را از نظر درجه نرمال سازی بررسی کنید. این بررسی باید از سطح 1NF تا 4NF انجام شود. برای این منظور با ذکر دلیل مشخص کنید چرا جداول طراحی شده شما تا سطح 4NF نرمال سازی شده اند و اگر شرایط هر کدام از فرم های نرمال را دارا نیست، باید آن ها را به فرم نرمال 4NF ببرید. (در نهایت تمامی جداول شما باید به فرم نرمال 4NF باشند).

ب) وابستگی تراگذری یا transitive dependency را برای جداول یک پایگاه داده شرح دهید و ضمن بیان اینکه این وابستگی مربوط به کدام درجه نرمال سازی است نوع آن را نیز از نظر وابستگی تابعی بررسی کنید.

ج) برای نرمال سازی چه معایی می توان متصور شد؟ بیان کنید.

الف) برای نرمال سازی جدول CabDriver را تعریف کردم و کلیدهای اصلی جدول Destination را اصلاح کردم.

روند بررسی سطوح نرمال سازی :

: NF1

چون Multi value attribute نداریم و از جداول جداگانه و کلیدهای خارجی استفاده شده

: NF2

چون partial Dependency در جداول دیده نمیشود و کلید(های) اصلی برای شناسایی یک ویژگی کفایت میکند

: NF3

چون وابستگی تراگذری دیده نمیشود .

: BCNF

چون هیچ کلید کاندیدی وجود ندارد که جزو کلیدهای اصلی نباشد .

: NF4

چون در کل جداول ، ویژگی ها فقط به کلید اصلی وابسته اند و وابستگی چند مقدره نداریم .

(ب)

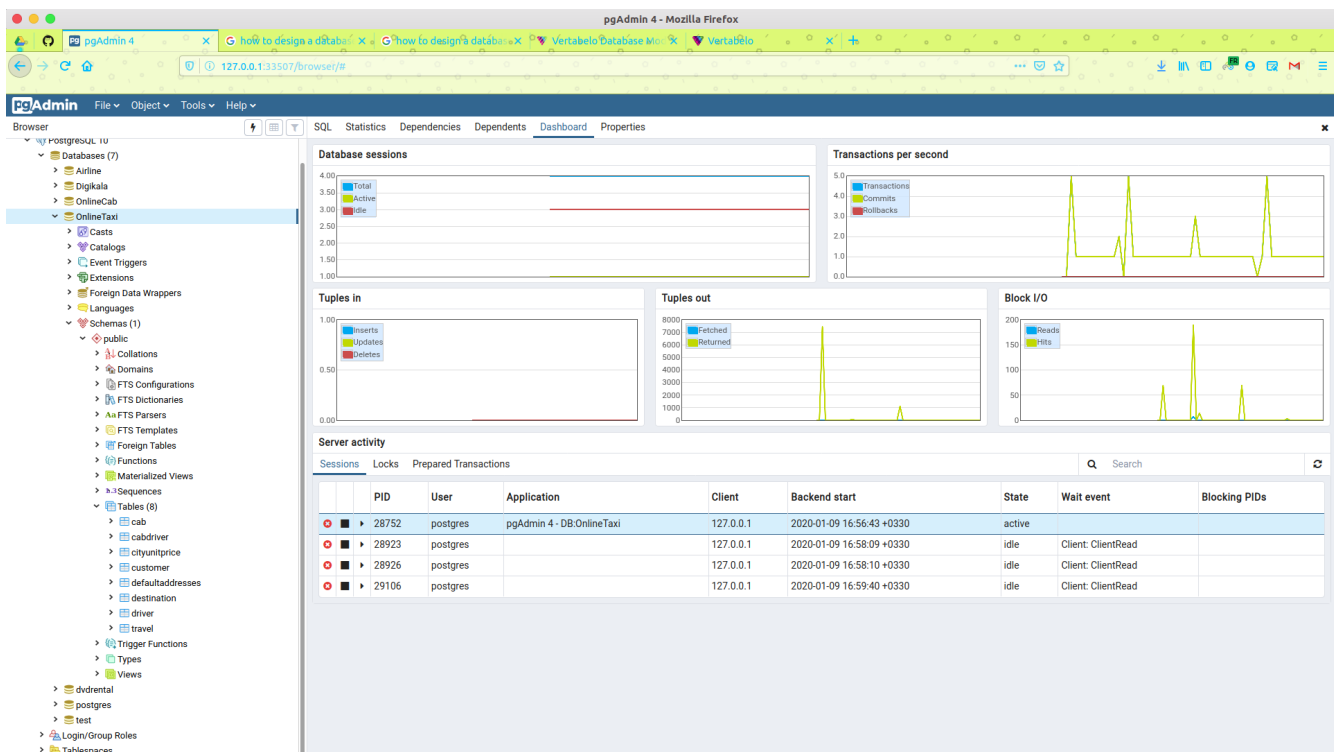
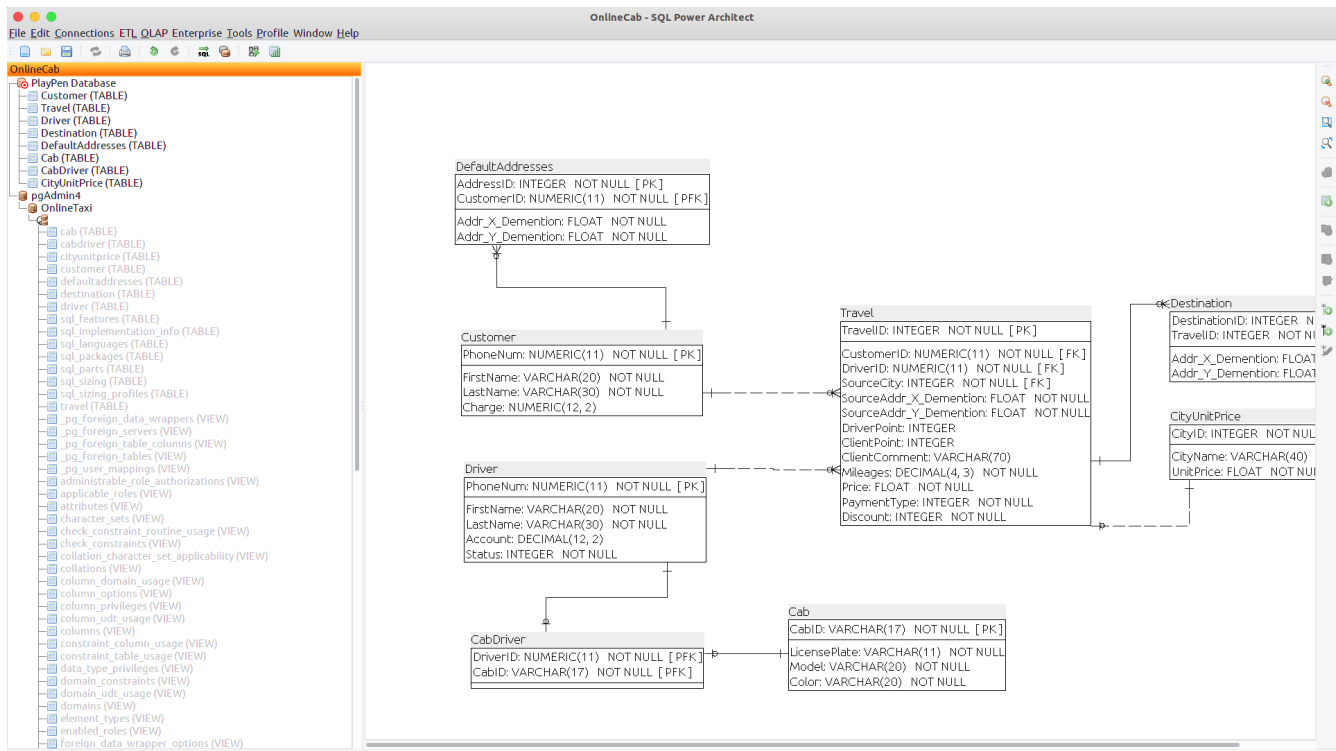
Non-key attributes are not allowed to define other non-key attributes.
Does not allow transitive dependencies in which one non-key attribute is functionally dependent on another.→ Third Normal Form
functional dependencies can be checked on individual relations without computing a join.

(ج)

Since data is not duplicated, table joins are required. This makes queries more complicated, and thus read times are slower.
Since joins are required, indexing does not work as efficiently.
Again, this makes read times slower because the joins don't typically work well with indexing.

۶- با استفاده از SQL Power Architect و امکان Forward Engineer از روی نمودار ER که تهیه کرده اید، جداول را در PostgreSQL ایجاد کنید.

خروجی ۶: متن دستورات تولید شده توسط Forward Engineer به همراه تصویری از جداول ساخته شده سمت PostgreSQL (کد های SQL در یک فایل جدا ذخیره شوند)



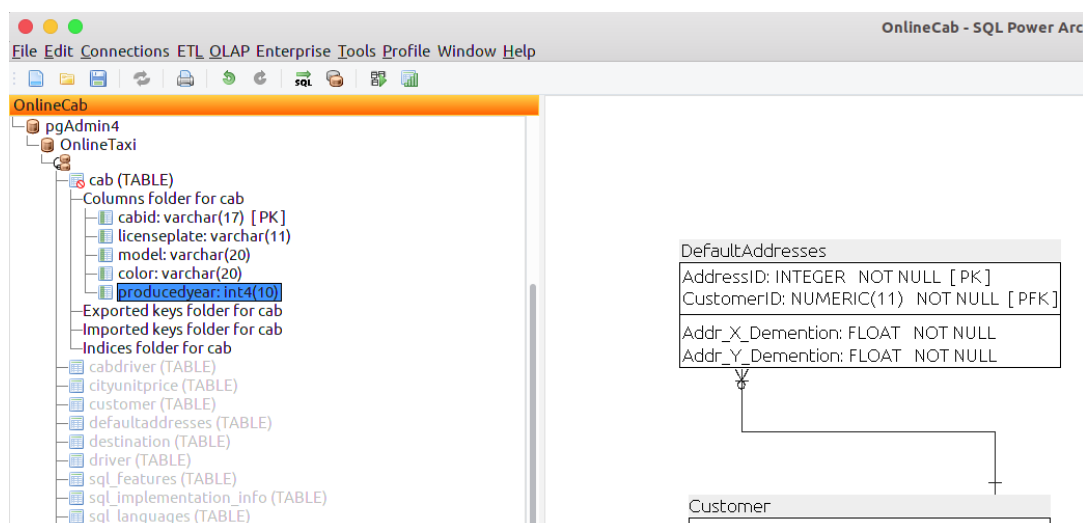
۷- اگر تغییری در طراحی جداول (علاوه بر تغییرات احتمالی که در قسمت نرمال سازی انجام داده اید) در سوال ۵-الف ایجاد کرده اید، این تغییر را در سمت PostgreSQL اعمال کنید. سپس سعی کنید این تغییرات را به محیط SQL Power Architect منتقل کنید (هدف بروز کردن نمودار ER از روی جداول توسط Reverse Engineering می باشد)

خروجی ۷: ذکر عملیات لازم برای Reverse Engineering به همراه تصویری از نمودار ER بروز شده بطوری که تغییر نسبت به ER قبلی مشخص باشد.

بعد از اجرای اسکریپت زیر در PostgreSQL مشاهده میشود که ستون جدید در کانکشن pgAdmin4 که قبلا

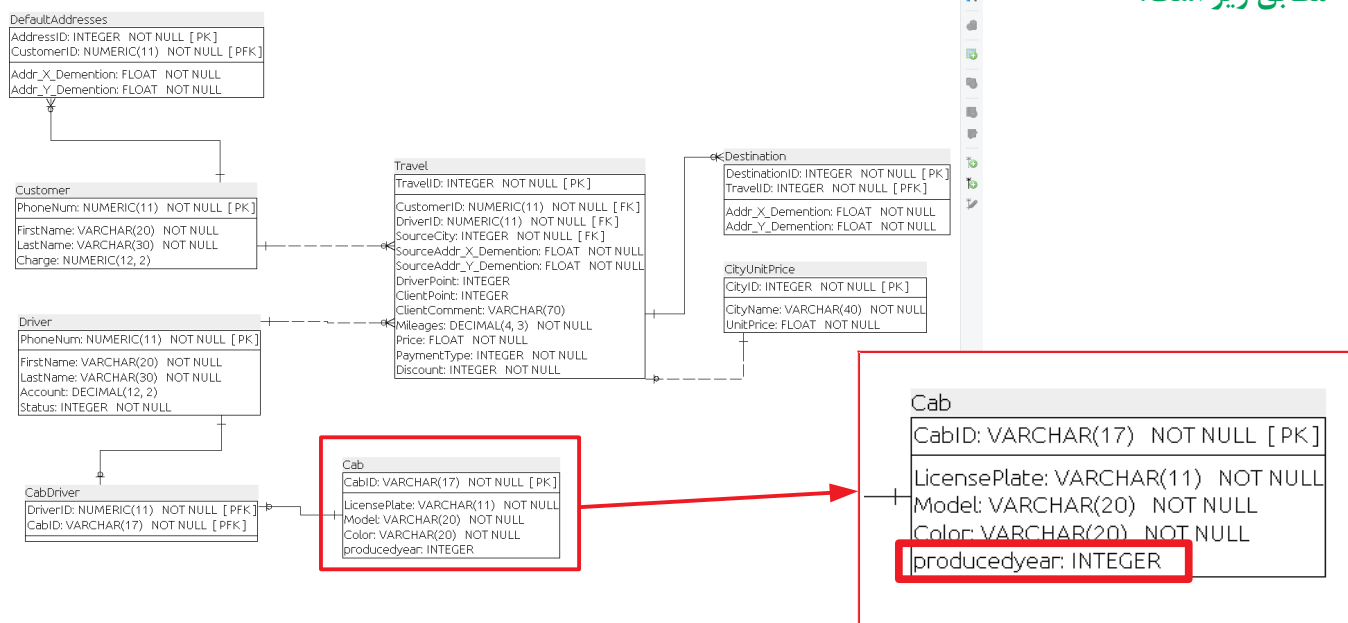
ساخته بودیم و در جدول مربوطه اضافه شده است مطابق تصویر زیر :

ALTER TABLE cab
ADD COLUMN ProducedYear INTEGER



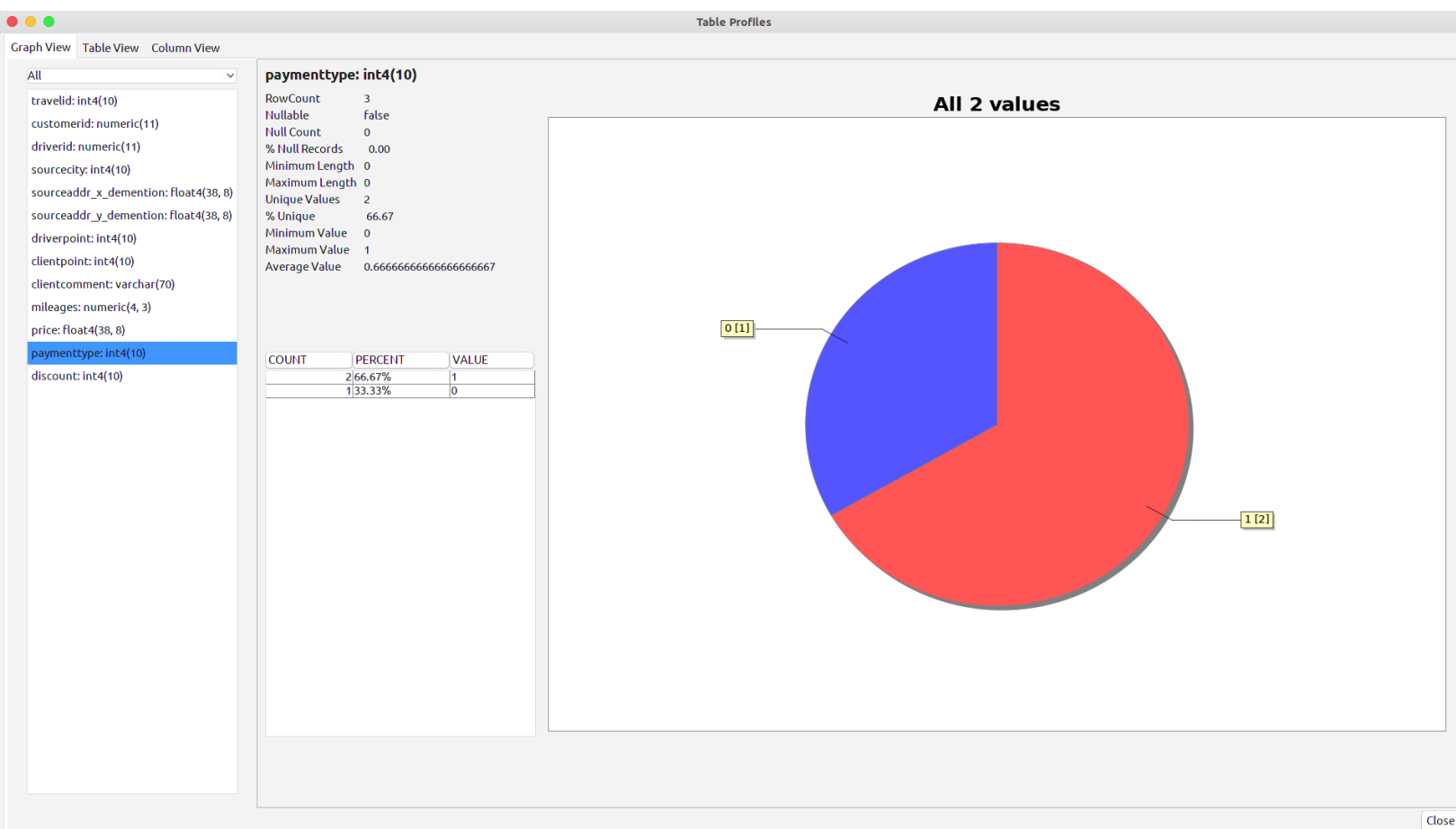
حالا با Drag and Drop این ستون جدید را به نمودار ER مان و در جدول Cab اضافه میکنیم و نمودار ER جدید

مطابق زیر است:



۸- با استفاده از امکان Profile در ابزار SQL Power Architect تحلیلی روی سفر های ثبت شده در سیستم از نظر نوع پرداخت (نقدی یا الکترونیکی) انجام دهید. برای انجام این تحلیل نیاز است تا به مقدار کافی داده وارد جداول خود کنید.

خروجی ۸: تصویری از نتیجه Profile به همراه شرحی بر تصویر.



بعد از انتخاب جدول مورد نظر (Travel) روی آن کلیک راست کرده و گزینه ی Profile را انتخاب میکنیم . در ادامه پنجره ای مشابه فوق باز میشود ، حالا در ستون سمت چپ ، روی ستونی که قصد تحلیل دیتاهایش را داریم کلیک میکنیم.....

سپس گراف و داده های آماری مربوط به این ستون از جدول در سمت راست ظاهر میشود ... من در کل سه سطر به جدول Travel اضافه کرده ام که از این بین پرداخت هزینه ی دو سفر ، به صورت آنلاین و تنها یک مورد به صورت نقدی بوده که گراف و داده های آماری خروجی Profile هم گواهی بر این ادعاست.