

گزارش فاز دوم پروژه درس طراحی پایگاه داده ها

محمدجواد احمدپور 400104697 ، امیرمهدی میقانی 400105274 ، محمدشایان شعبانی 400105069

توابع:

1- تابع `distance_of_driver` را داریم که از سیگنیچر آن کار آن مشخص است. در این تابع ابتدا تمام مسیر های یک راننده را پیدا کرده و سپس با `distinct` از مسیر های `overlapping` فاکتور میگیریم و فقط مسیر ها را یکبار حساب میکنیم که عدد درست مسیر را با جمع روی یال ها میدهد و در بازه مشخص شده هم فیلتر میکند.

پرسمان ها:

پرسمان 1: در این پرسمان با داشتن بازه زمانی مورد نظر و شناسه راننده، تعداد سفر های او در بازه زمانی داده شده با شرط اینکه تعداد مسافران زن حداقل 0.6 برابر تعداد مسافر های مرد باشد خروجی داده می شود. در پرسمان طراحی شده ابتدا شرط صدق کردن در بازه زمانی و سپس با پیوند زدن جدول های شهروند، حساب و بلیت سفر، می توانیم به جنسیت مسافران پی ببریم و آن ها را برای خانم ها و آقایان بشماریم و در نهایت شرط خواسته شده را چک کنیم.

پرسمان 2: در بازه زمانی داده شده، هزینه همه خدمات شهروندی را برای شهروندان مختلف از طریق پیوند زدن جدول شهروندان با جدول قبض های مختلف به دست می آوریم. در نهایت با دسته بندی بر اساس سرپرست شهروندان مختلف (که خانواده را تشکیل میدهد) و همچنین مرتب سازی بر اساس مجموع هزینه ها، 5 خانواده خواسته شده را خروجی می دهیم.

پرسمان 3: در این پرسمان با داشتن بازه زمانی مدنظر، جدول وسایل نقلیه عمومی را با همه سفر ها پیوند میزنیم و با به دست آوردن مجموع مسافت های طی شده هر راننده در سفر های مختلف داخل بازه زمانی داده شده (به کمک `group by`) و در نهایت مرتب سازی به کمک `order by` خروجی خواسته شده را تحویل می دهیم.

پرسمان 4: با داشتن شناسه ایستگاه و همچنین بازه زمانی، سفر ها را با وسایل نقلیه عمومی پیوند می زنیم، سپس سفر های درون بازه زمانی داده شده را انتخاب می کنیم و با انتخاب یال های درون مسیر آن سفر، چنانچه ایستگاه داده شده اول یا آخر یک یال باشد، آن را انتخاب می کنیم. توجه کنید که یک وسیله نقلیه فرض شد که در تمامی ایستگاه های مسیر متوقف می شود. (برعکس مسافران که فقط در اولین و آخرین ایستگاه حضور دارند) مانند سیستم BRT تهران که در تمامی ایستگاه ها متوقف می شوند ولی مسافران فقط در ایستگاه ورودی و خروجی حضور دارند. در انتها بر اساس ماه دسته بندی می کنیم و با توجه به تعداد سفر های انجام شده در هر ماه مرتب می کنیم.

پرسمان 5: در این پرسمان با داشتن موقعیت مکانی یک ایستگاه در ساختار داده `Point`، می توانیم همه ایستگاه ها را انتخاب کرده و فاصله اقلیدسی آن ها را با ایستگاه داده شده به دست می آوریم و در نهایت با توجه به فاصله ها به صورت صعودی مرتب می کنیم و 5 تای بالا را با استفاده از کلید واژه `limit` بر می داریم.

پرسمان 6: با توجه به بازه زمانی داده شده، جدول های شهروندان، بلیط های سفر، سفر ها و در نهایت یال های مسیر هر سفر را پیوند می زنیم، برای به دست آوردن ایستگاه هایی که مسافران در مسیر از آن ها بازدید کرده اند، باید فقط ایستگاه های شروع و پایان را حساب کنیم. می دانیم که ایستگاه شروع، تنها ایستگاهی است که برای یال های

مسیر سفر، تنها به عنوان ایستگاه شروع شناخته می شود. ایستگاه پایانی نیز دقیقاً همینطور است، با این تفاوت که تنها ایستگاهی در مسیر است که فقط به عنوان ایستگاه پایانی قرار دارد، با قرار دادن این دو شرط و جمع کردن همه این حالات در یال های پیموده شده در سفر های مختلف که در جدول نهایی پس از پیوند زدن به دست آورده ایم، همه این ایستگاه ها را پیدا می کنیم و در نهایت بر اساس تعداد آن ها را مرتب سازی کرده و خروجی می دهیم.

پرسمان 7: با داشتن بازه زمانی داده شده، جدول شهروندان را با بلیط ها پیوند می زنیم. حال تعداد سفر های داخل بازه زمانی داده شده در دو حالت اینکه با مترو یا اتوبوس انجام شده باشد را حساب می کنیم. اینکار به وسیله پیوند زدن جدول سفر ها با جدول وسایل نقلیه عمومی انجام می شود، در ادامه فقط سفر هایی که بلیطش برای شهروند صادر شده باشد و وسایل نقلیه خاص مانند مترو و اتوبوس را در نظر میگیریم.

پرسمان 8: در این پرسمان ابتدا بازه زمانی و شناسه پارکینگ را دریافت می کنیم، سپس با پیوند زدن جدول پارکینگ و قبض پارکینگ و ماشین های شخصی، می توانیم به ماشین هایی که به آن پارکینگ در بازه زمانی داده شده مراجعه کرده اند دسترسی پیدا کنیم. در نهایت با چسباندن برند و رنگ و شمردن تعداد موارد یکتای آن ها می توانیم خواسته پرسمان را خروجی دهیم.

پرسمان 9:

پرسمان 10:

پرسمان 11:

پرسمان 12: در بخش select، میانگین را با join زدن روی شهروند و قبض سفر به دست میآوریم و در بخش where، با پیدا کردن national_id داخل جدول وسیله نقلیه شخصی ها شرط پرسمان را اعمال میکنیم.

پرسمان 13:

اطلاعات شهروند هایی را میدهد که شناسه آن ها در لیست x وجود دارد. لیست x تشکیل شده از شناسه اکانت هایی است که با جویین زدن parking bill با خودش روی اکانت های یکسان ایا با فاصله زمانی یک روز در پارکینگ رفته اند یا خیر.

پرسمان 14:

پرسمان 15:

اطلاعات افرادی را میدهد که مجموع مسافت طی شده توسط خود آن ها کمتر مقدار مشخص شده است. این مجموع مسافت ها طبق بازه زمانی مشخص شده. و باید در نوع اتوبوس باشد که ایدی در لیست ما وجود داشته باشد.

دیدها:

دید 1: در این دید با کمک تابع distance_of_driver که مسافت راننده در یک بازه را میدهد میتوان به راحتی رانندگانی در بازه که میانگین مسافتشان همان مقدار مطلوب باشد بدهد.

دید 2: برای این دید ابتدا تعداد افرادی که در هر سفر به یک مقصد مشخص و یا مبدا مشخص میروند را پیدا میکنیم سپس این دو تیبیل را با هم اجتماع میگیریم. سپس با لغت جویین از ایستگاه ها باعث میشویم که تمام ایستگاه ها حتی بدون مسافر در جدول حاصل بیایند. سپس با groupby و جمع کردن تعداد مسافران هر ایستگاه و خروجی دادن آن به عنوان یک سطر به جواب میرسیم.

دید 3: برای این دید، روی سه جدول سفر، وسیله نقلیه عمومی و رسید سفر join میزنیم و شرط مترو بودن وسیله نقلیه را اعمال میکنیم و روی id وسیله نقلیه group by میکنیم و سپس شهروندان را بر اساس national_id یونیک میکنیم.

دید 4: در این دید، خانه را با مصرف روزانه join میزنیم و سپس مصارفی را که برای یک ماه اخیرند انتخاب کرده و روی house_id گروپ میکنیم و سپس جمع میکنیم و با این مقدار مشخص مقایسه میکنیم.

رهاناها:

1- رهانایی برای پایان فرایند پارکینگ نوشتیم که هم قبض را کامل میکند و هم برداشت از حساب را انجام میدهد. در ضمن قبل برداشت از حساب چک میکند و اگر حساب موجودی کافی نداشت به جای فرد اولیه یا همان راننده از حساب مالک خرج میکند.

2- دو رهانا برای این ساختیم. یکی برای شروع فرایند قبض پارکینگ و دیگری فرایند سفر ها است. که در صورت بالانس کمتر مساوی 0 ارور میدهد.

3- دو رهانا این کار را انجام میدهند یکی برای قبض پارکینگ و دیگری برای قبض سفر که در صورت اضافه شدن بخش پایان یا زمان پایان مستقیما فاکتور را صادر میکنند و برای بخش هزینه هم محاسبات کامل انجام میدهند تا هزینه به دست بیاید.

4- سه رهانا وظیفه این را به عهده دارند یکی برای خانه یکی برای سفر یکی برای پارکینگ. که از روی مقدار محاسبه شده ی فاکتور صرفا حساب مد نظر را کم میکنند.

5- برای این رهانا ها صرفا ترتیب را بررسی کرده و در صورت وجود مشکل ارور میدهد و اینسرت انجام نمیشود

برای ساختن داده ها از کد پایتون استفاده کردیم و خروجی ان را با کمی تغییر در دیتابیس اجرا کردیم که فایل پایتون و فایل sql را قرار داده ایم.