



پاسخ مسئله‌ی ۱.

بخش ۱

در این بخش باید همه پیتزافروشی‌هایی را پیدا کنیم که حداقل یک مشتری بالای ۸۰ سال دارند.

- انتخاب افرادی که سن آن‌ها بالای ۸۰ سال است:

$$Elderly = \sigma_{age \geq 80}(Person)$$

- پیدا کردن پیتزافروشی‌هایی که این افراد به آنها مراجعه کرده‌اند:

$$ElderlyFrequents = Elderly \bowtie_{Person.ID=Frequents.personID} Frequents$$

- جدا کردن نام پیتزافروشی‌ها:

$$Result = \Pi_{pizzeria}(ElderlyFrequents)$$

بخش ۲

در این بخش باید همه پیتزا فروشی‌هایی را بیابیم که حداقل یک پیتزا را سرو می‌کنند که قیمتی زیر ۲۵۰ هزار تومان دارد و فردی به نام عموحسن آن را می‌خورد.

$$Result = \Pi_{pizzeria}(\sigma_{(price < 250,000) \wedge (name = "AmooHassan")}(Serves \bowtie_{Serves.pizza=Eats.pizza} Eats \bowtie_{Eats.personID=Person.ID} Person))$$

بخش ۳

در این بخش باید پیتزافروشی‌هایی که فقط مشتریان آقا یا فقط مشتریان خانم دارند را پیدا کنیم.

- ابتدا پیتزافروشی‌هایی که حداقل یک مشتری مرد و حداقل یک مشتری زن دارند را پیدا می‌کنیم:

$$HasMale = \Pi_{pizzeria}(Frequents \bowtie_{Frequents.PersonID=Person.ID} (\sigma_{gender=male} Person))$$

$$HasFemale = \Pi_{pizzeria}(Frequents \bowtie_{Frequents.PersonID=Person.ID} (\sigma_{gender=female} Person))$$

- حال با اشتراک و تفریق این دو مجموعه از کل پیتزافروشی‌ها به جواب می‌رسیم:

$$Result = \Pi_{pizzeria}(Frequents) - (HasMale \cap HasFemale)$$

بخش ۴

در این بخش باید آن افرادی که مشتری همه پیتزافروشی‌هایی که پیتزاهای مورد علاقه آن‌ها را دارد شناسایی کرد.

- ابتدا لیست پیتزافروشی‌هایی که حداقل یکی از غذاهای مورد علاقه هر فرد را دارند می‌سازیم:

$$PizzeriasPerson = \prod_{Pizzeria, personID} (Serves \bowtie_{Serves.pizza=Eats.pizza} Eats)$$

- در آخر آن افرادی که شامل این Tuple ها نیستند را شناسایی می‌کنیم و از کل افراد حذف می‌کنیم:

$$Result = \prod_{personID} Frequent - \prod_{personID} (PizzeriasPerson - Frequent)$$

بخش ۵

در این بخش باید پیتزافروشی‌هایی را بیابیم که ارزان‌ترین پیرونی‌ها را دارند.

- ابتدا ارزان‌ترین پیرونی را می‌یابیم:

$$CheapestPrice = \prod_{price} (Serves) - \prod_{price} [\sigma_{P_1.price > P_2.price} [\rho_{P_1}(Serves) \times \rho_{P_2}(Serves)]]$$

- حال آن پیتزافروشی‌هایی که این پیرونی را می‌فروشند را پیدا می‌کنیم:

$$\prod_{pizzeria} [\sigma_{pizza=pepperoni}(Serves) \bowtie_{Serves.price=CheapestPrice.price} CheapestPrice]$$

پاسخ مسئله‌ی ۲.

ما دارای روابط Factory، Product، FP را هستیم و با استفاده از آن‌ها به سوالات پاسخ می‌دهیم.

بخش ۱

در این بخش شماره‌ی کارخانه‌های شهر تهران که وضعیت آنها Open باشد را پیدا کنیم.

- فیلتر کردن کارخانه‌ها با ویژگی‌های گفته شده:

$$\sigma_{Factory.status="Open" \wedge Factory.city="Tehran"}(Factory)$$

- حال شماره‌ی کارخانه‌ها را جدا می‌کنیم:

$$FilteredFactories = \sigma_{Factory.status="Open" \wedge Factory.city="Tehran"}(Factory)$$

$$FilteredFactoriesID's = \Pi_{fID}(FilteredFactories)$$

بخش ۲

در این بخش باید نام کارخانه‌هایی که قطعه‌ی ۴۰۶۷۸ را تولید می‌کنند پیدا کنیم.

- ابتدا رابطه بین محصول و کارخانه را برحسب کد محصول فیلتر می‌کنیم:

$$FilteredFP = \sigma_{FP.pID=40678}(FP)$$

- حال رابطه فیلتر شده را با رابطه‌ی کارخانه‌ها Semi Join می‌کنیم:

$$FilteredFactories = Factory \bowtie_{FilteredFactories.sID=Factory.fID} FilteredFP$$

- حال نام‌ها را جدا کرده و خروجی می‌دهیم:

$$FilteredFactoriesNames = \Pi_{fName}(FilteredFactories)$$

بخش ۳

در این بخش باید نام کارخانه‌هایی که حداقل یک قطعه قرمز تولید می‌کنند را پیدا کنیم.

- ابتدا محصولات قرمز را جدا می‌کنیم:

$$RedProducts = \sigma_{Color='Red'}(Product)$$

- حال کارخانه‌هایی را که قطعات قرمز را تولید می‌کنند را پیدا می‌کنیم:

$$RedProductFP = RedProducts \bowtie_{RedProducts.pID=FP.pID} FP$$

$$Factories = RedProductFP \bowtie_{RedProductFP.sID=Factory.fID} Factory$$

- فیلتر نام کارخانه:

$$FactoryNames = \Pi_{fName} Factories$$

بخش ۴

در این بخش باید نام کارخانه‌هایی که حداقل یک قطعه تهیه شده توسط کارخانه‌ی Iran Sim را تولید می‌کنند را پیدا کنیم.

- ابتدا قطعات تولیدی این کارخانه خاص را پیدا کرده:

$$IransimProducts = \prod_{pID} ((\sigma_{fName='IranSim'} Factory) \bowtie_{Factory.fID=FP.sID} FP)$$

- حال آن کارخانه‌های خاص را پیدا می‌کنیم:

$$FactoriesID = \prod_{sID} (IransimProducts \bowtie_{IransimProducts.pID=FP.pID} FP)$$

- در آخر نام آن‌ها را پیدا می‌کنیم:

$$FactoriesName = \prod_{fName} (FactoriesID \bowtie_{FactoriesID.sID=Factory.fID} Factory)$$

پاسخ مسئله‌ی ۳.

بخش اول

در این بخش باید عبارات جبری را به زبان فارسی بازنویسی کنیم.

۱. این عبارت آن Supplier هایی که حداقل یک کالا به رنگ قرمز و قیمت تمام شده کمتر از ۱۰۰ تولید می‌کنند را نشان می‌دهد. همچنین تمامی هدرهای رابطه Supplier نشان داده می‌شود.
۲. این بخش در ابتدا همان کارهای مورد ۱ را انجام می‌دهد ولی در آخر فقط نام آن Supplier ها را نشان می‌دهد نه همه‌ی ویژگی‌های آن‌ها را.
تفاوت مورد ۱ و ۲ در توضیحات مورد ۲ ذکر شده است.
۳. این عبارت آن Supplier هایی که حداقل یک کالا قرمز با قیمت کمتر ۱۰۰ و حداقل یک کالا سبز با قیمت کمتر از ۱۰۰ تولید می‌کنند را شناسایی کرده و در آخر فقط نام آن‌ها را نشان می‌دهد.

بخش دوم

۱. عملگر تقسیم آن ملوان‌هایی که در رزروهای خود تمامی قایق‌ها را سوار می‌شوند را شناسایی کرده و سپس با استفاده از عملگر join نام آن‌ها را پیدا می‌کنیم.
۲. این عبارت جبری آیدی آن ملوان‌هایی را شناسایی می‌کند که امتیاز آن‌ها از Bob بالاتر است.
۳. این عبارت جبری نام و سن پیرترین ملوان‌ها را بر می‌گرداند.

پاسخ مسئله‌ی ۴.

بخش اول

الف

$$\begin{aligned} & \Pi_{Name, Type, Quantity} (\sigma_{(equipped=True) \wedge (RegistrationDate > 2023)} (\\ & \quad User \bowtie_{User.userid=Inventory.PlayerID} Inventory \\ & \quad \bowtie_{Inventory.InventoryID=InventoryHasItem.InventoryID} InventoryHasItem \\ & \quad \bowtie_{InventoryHasItem.ItemID=Item.ItemID} Item)) \end{aligned}$$

ب

$$\begin{aligned} ItemWeapons &= \Pi_{ItemID} (\sigma_{Type=weapon}(Items)) \\ FilteredIDs &= \Pi_{BuyerID} ((\Pi_{ItemID, BuyerID} Transactions) \div (ItemWeapons)) \\ Usernames &= FilteredIDs \bowtie_{UserIDs.BuyerID=Users.UserID} Users \end{aligned}$$

ج

$$\begin{aligned} & \Pi_{TransactionID} [\sigma_{Transactions.Price > 100\$} (\\ & (Transactions \bowtie_{Transactions.BuyerID=Friends.UserID \wedge Transactions.SellerID=Friends.UserID} Friends) \\ & \cup \\ & (Transactions \bowtie_{Transactions.SellerID=Friends.UserID \wedge Transactions.BuyerID=Friends.UserID} Friends))] \end{aligned}$$

بخش دوم

در این کوئری ابتدا آن بازیکن‌هایی که فروشی نداشته‌اند و XP بزرگتر یا مساوی ۲۵۰ دارند را آیدی‌شان را جدا کرده، سپس کارکترهای آن‌ها را شناسایی کرده، اگر ویژگی آن کارکترها strength بود اسم آن کارکترها را به ما نشان می‌دهد.

بخش سوم

کوئری B بهتر است و هزینه اجرای کمتری دارد، زیرا ابتدا روی هر رابطه یک بار Selection کرده و سپس ضرب کارتازین را انجام داده ایم، ولی در کوئری A، ابتدا ضرب را انجام داده و سپس Selection را انجام داده‌ایم، این کار باعث می‌شود که محاسبات اضافی زیادی را انجام بدهیم و Table خود را خیلی خیلی بزرگ کنیم در صورتی که به آن نیازی نداریم.