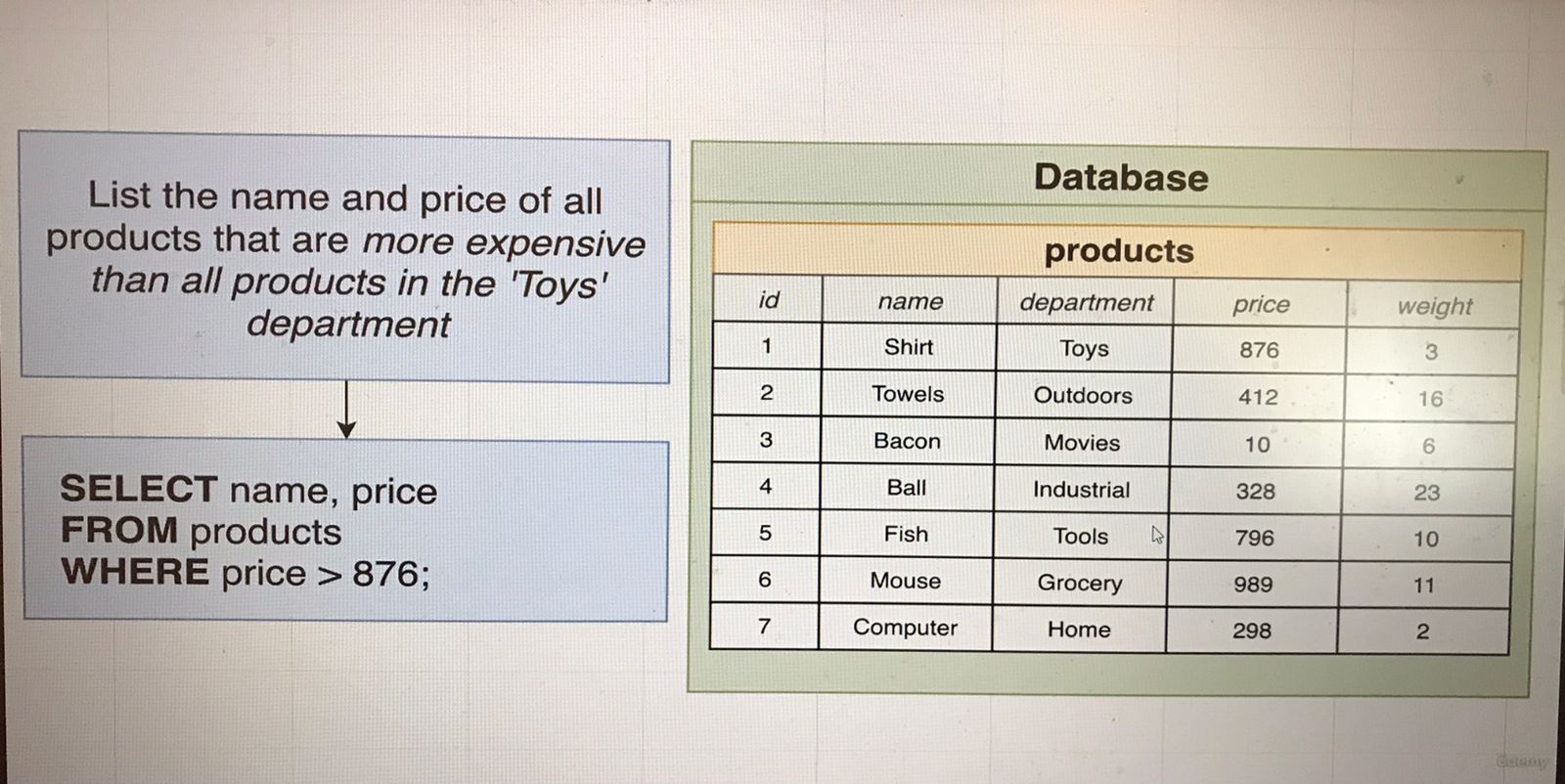
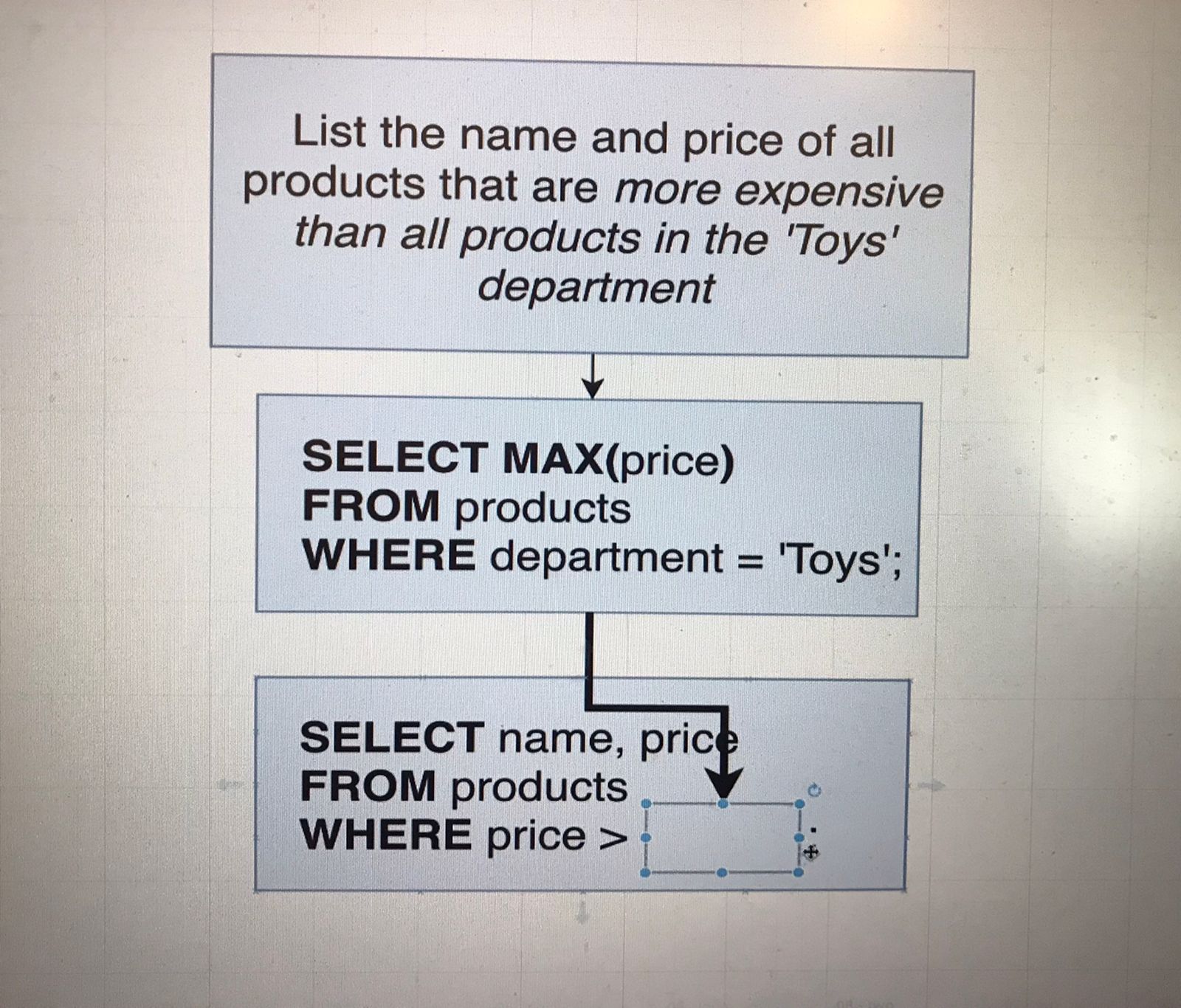
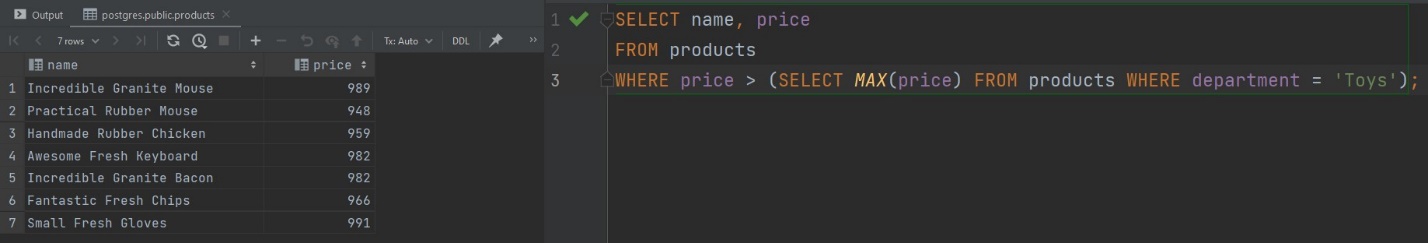
1. Bu mövzudan etibarən biz **SQL** də **subquery**-lərin nə olduğunu anlamağa çalışacayıq. Lakin mövzuya keçid etməmişdən öncə sizinlə aşağıdakı şəkildə göstərilmiş problemi həll etmək istəyirəm. Deməli problem ondan ibarətdir ki, **products** table-ından **Toys** departamentində olan ən bahalı **product**-dan da bahalı olan **product**-ları tapmalıyıq. Əgər biz bunu sadə yolla etmiş olsaydıq, aşağıdakı şəkildə göstərilmiş kimi bir query yazardıq.

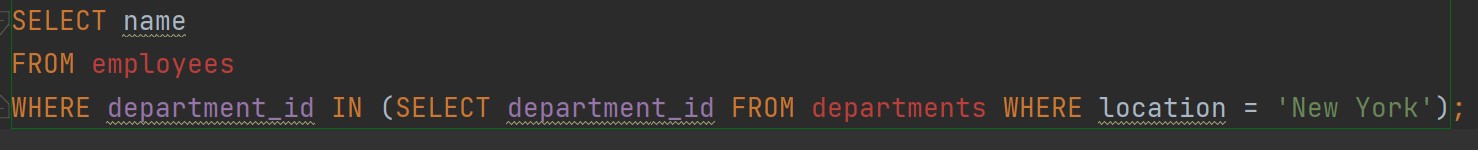
gördüyümüz kimi, yuxarıdakı şəkildə də göstərildiyi kimi, biz **price > 876** yazmışıq, yəni, bu o deməkdir ki, az öncədə dediyimiz kimi **Toys** departamentində olan **product**-dan da ən bahalı **product**­ı tapmalıyıq, **lakin** bu yol əlverişli deyil, ona görəki, bizim table-mızda milyonlarla sətirlər ola bilər və bunu tapmağımız qeyri mümkün və məntiqsizdir, lakin query-ni hal hazırda belə asand yoldan yazmağımıza səbəb odur ki, hal hazırda bizim qarşımızda kiçik bir table-ı vizual olaraq gördüyümüz üçün bunu rahat bir şəkildə yaza bilirik ki, **Toys** departamentin-də olan product-dan da ən bahalı product-ın name və price sütunlarındakı məlumatlarını çəkə bilirik.

1. Bu problemin query-sini yaza bilməyimiz üçün biz bu gün öyrənəcəyimiz **subquery**-lər ilə etmiş olacayıq. İlk öncə biz ayrı-ayrı query-lər yazaraq bu problemi həll etmiş olacayıq.
2. İlk öncə **products** table-ından **Toys** departamentində ən bahalı **product**-ı tapan query-ni yazacayıq.
3. Daha sonra isə yuxarıdakı query-ni başqa bir query daxilində istifadə edəcəyik.

bu söyləntilər aşağıdakı şəkildə göstərilmişdir.

gördüyünüz kimi, ilk öncə **products** table-ından **Toys** departamentində ən bahalı **product**-ı tapırıq, ardından isə həmən bu query-ni aşağıdakı **query** daxilində **WHERE** hissəsində istifadə etmiş oluruq. Bu cürnə query-lər bizə o şəraiti yaradır ki, eyni anda bir neçə query yazaraq eyni anda çalışdıra bilməyimizi təmin edir, beləliklə siz ayrı-ayrı query-lər yazmaq əvəzinə eyni anda çalışa bilən bir neçə query yaza bilirsiniz və bu adlanır **subquery**. Elə isə gəlin bu query-ni yazıb çalışdıraq.

gördüyünüz kimi yuxarıdakı şəkildə olan query-də biz **Toys** departamentində olan **product**-lardanda ən bahalı olan **product**-ların name və price dəyərlərini çapa vermiş olduq.

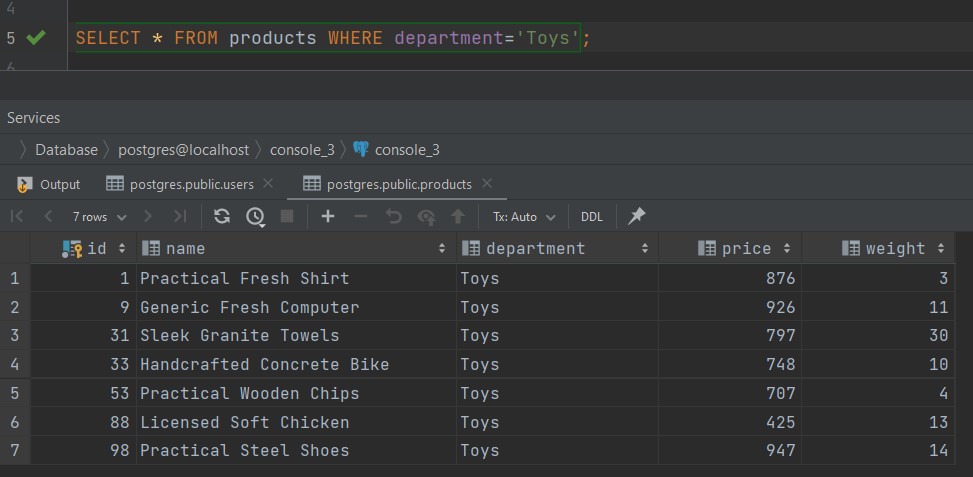
1. Əgər bir query içərisində başqa bir query istifadə olunmuşdursa, xüsusəndə bu query mötərizə daxilindədirsə adlanır **subquery**. **Subquery**-ə həmçinin **inner** yaxud **nested** query də deyilməkdədir. **Subquery** ilk işə düşən hissədir, buna səbəb ondan qayıdan nəticə **outher** query yaxud **main** query-ə lazımlıdır. **Subquery**-lər fərqli-fərqli **SQL** statement-lərdə,**SELECT,FROM,WHERE** yaxud **HAVING** kimi yerlərdə istifadə oluna bilinir. **Subquery**-lər mötərizələr daxilində yazılır, eynən aşağıdakı şəkildə olduğu kimi.

yuxarıdakı şəkildə **(SELECT department\_id FROM departments WHERE location = 'New York');** olan hissə **subquery**-dir və o **New York** departmentində yerləşən department-lərin id-lərini qaytarmaqdadır. Daha sonra **outher** query həmən bu **subquery-**dən qayıdan **department\_id** dəyərlərini alıb **WHERE** hissəsində **department\_id** sütununa görə filterləmə əməliyyatını apararaq, **subquery**-dən qayıdan dəyərlərin bu sütunda olub-olmadığını uzlaşdıraraq **employees** table-ından adları qaytaracaqdır.**subquery**-lər complex query-ləri parçalara ayıraraq işləri rahatlaşdırma baxımından faydalıdır.

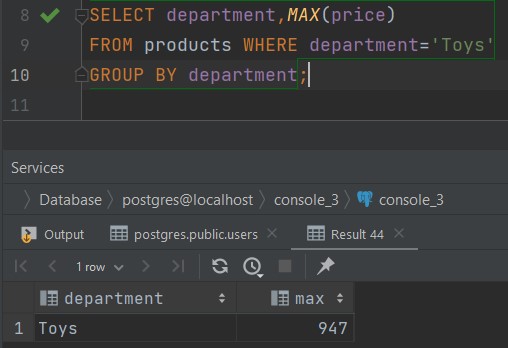
1. **SQL** da **subquery**-lərin bir neçə fərqli tipi vardır və onlar aşağıdakılardır:

* **Scalar subqueries** geriyə tək bir dəyər yaxud tək bir sətir yaxud tək bir sütun qaytarmaqdadır.
* **Multirow subqueries** geriyə ya bir sütundan ibarət çoxlu sətur yadakı çoxlu sütundan ibarət çoxlu sətir qaytarmaqdadır.
* **Correlated subqueries** isə **outher** query-dən gələn məlumat əsasında geriyə nəticə qaytarmaqdadır.

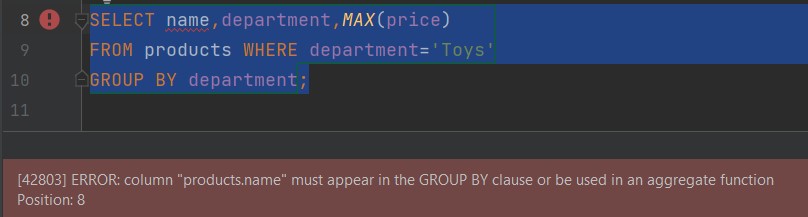
Əgər bir **subquery** yazmaq istəyirik isə biz ilk öncə birbaşa **subquery** yazmaq əvəzinə yazmaq istədiyimiz **subquery**-ni parçalara ayırıb ayrı-ayrılıqda çalışdıraraq işə salıb test nəticələrinə baxıb istədiyimiz nəticəni əldə ediriksə həmən parçalara ayırdığımız query-ləri birləşdirib **subquery** vəziyyətinə salmış oluruq. Məsələn gəlin **products** table-ından **department**-i **Toys** olan ən bahalı **product**-ı tapmağa çalışaq.

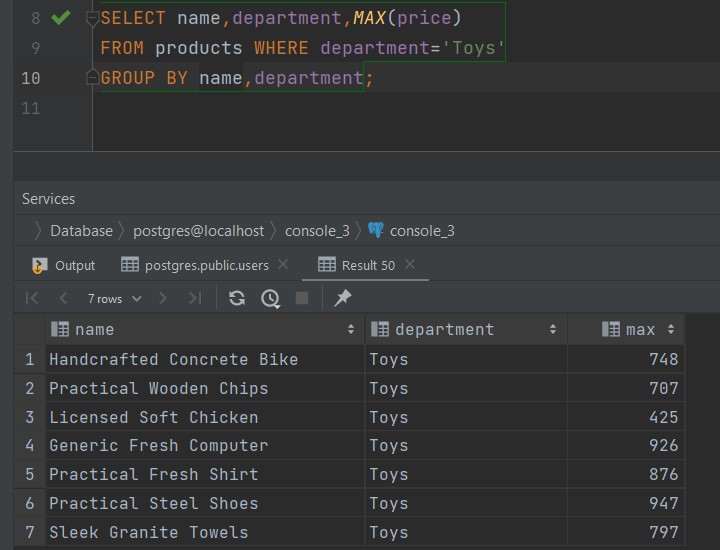
1. İlk öncə etməli olduğumuz əməliyyat budur ki, **products** table-ından **department**-i **Toys** olan **product**-ları çəkməyə çalışaq, eynən aşağıdakı şəkildə olan query kimi.

yuxarıdakı şəkildə olan **resultset** də gördüyümüz kimi **department**-i **Toys** olan ən bahalı **product** 2-ci sətirdə olan məlumatdır, bunu **price** sütununda olan dəyərə görə təyin etdik. İndi biz elə bir query yazmalıyıq ki, məhz bu sətirdəki məlumatı əldə edə bilək. Bilirəm çoxlarınız deyəcək ki, **WHERE** operatoruna **price** sütununa ən çox dəyər hansıdırsa onu set edərək əldə edə bilərik. Lakin bu düzgün yanaşma deyil, ona görəki, sizin table-nızda heçvax bilə bilməzsinizki, hansı product max **price** dəyərinə sahibdir. Gəlin elə isə bir-bir hansı mümkün yollar var onu tapmağa çalışaq.

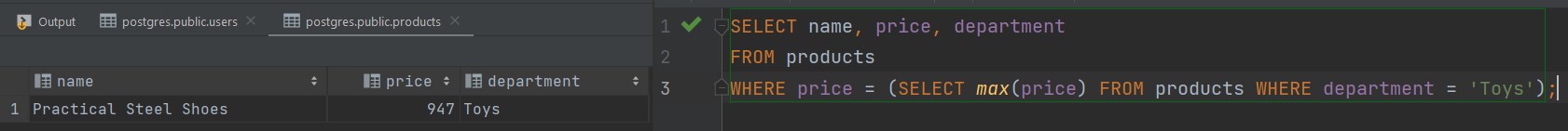
1. İlk öncə, tələb olunan problemə baxaraq orada nə istənildiyini və bu istədiklərini **SQL** də hansı operatorlardan istifadə edərək tapa bilərəm onu yadıma salıram. Məsələn məndən ən bahalı **product** tapılmasını istəyirsə, mən bilirəmki **product**-ın qiyməti **price** sütununda yerləşməkdədir və mən bir ədədin max dəyər olduğunu bilmək istəyirəmsə, əvvəlki mövzularda öyrəndiyim **aggregate** funksiyalardan **max()** funksiyasına müraciət edə bilərəm. O zaman gəlin yazmağa çalışaq.

yuxarıdakı şəkildəndə gördüyünüz kimi **products** table-ından **department**-i **Toys** olan ən bahalı **product**-ı tapdıq. Lakin bunu tapmağımıza rəğmən bizim tam istədiyimiz kimi deyil, çünki bizdən istənilən həmdə bu **product**-ın adınında göstərilməsi lazımdır.

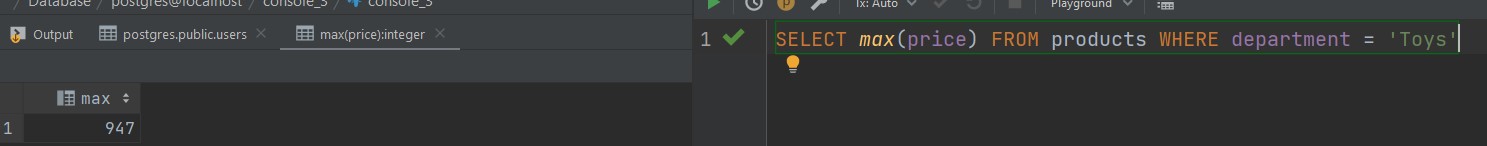
1. **product**-ın adını da göstərə bilmək üçün biz artıq, **subquery** yazmağımız gərək duyulmaqdadır. Deyə bilərsiniz ki, yuxarıdakı query-ə **name** sütununda artırmış olsaydıq işləməzdi?? deyirsinizsə, bildiyimiz kimi həmən sütunlar **GROUP BY**-a əlavə olunmadığından xəta ilə qarşılaşacayıq, eynən aşağıdakı şəkildə olduğu kimi.

yox əgər deyirsinizsə ki, həmən əlavə etdiyimiz sütunuda **GROUP BY**-a əlavə edək, o zaman istədiyimiz case-i ala bilmiyəcəyik, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi bu bizim istədiyimiz case deyil bizə lazım olan yalnızca bir sətirdə məlumatların gətirilməsidir.

1. Məhz ona görə bizim istədiyimiz case-ə uyğun query eynən aşağıdakı şəkildə olduğu kimidir.

gördüyünüz kimi yuxarıdakı query də **WHERE** hissəsində biz bir **subquery** istifadə etmiş olduq. Bunu daha yaxşı anlıya bilmək üçün gəlin aşağıdakı şəkillərdən yola çıxaraq anlamağa çalışaq.

1. İlk öncə etməli olduğunuz step **subquery**-ni ayrılıqda çalışdırmanızdır.

gördüyünüz kimi yuxarıdakı query-ni tək çalışdırdığımızda bizə tək bir dəyər qaytarmaqdadır və bu dəyər **department**-i **Toys** olan **product**-ın max dəyərini çıxartmaqdadır.

1. Daha sonra biz bunu **WHERE** hissəsində **price** sütununa görə müqayisə əməliyyatını aparmaqdayıq. Birdaha yadımıza salaq, **WHERE** hissəsi vermiş olduğumuz sütun adına görə üzərində sorğu əməliyyatı aparacağımız table üzərində sətir bə sətir **price** sütununa görə axtarış əməliyyatı aparacaqdır və **WHERE** də elan etdiyimiz **price** sütununda hansı dəyər bu **subquery** dən qayıdan dəyərə bərabərdirsə yəni **products** table-ında **price** sütununda hansı dəyər 947-dirsə həmən şərtə uyğun gələn sətiri saxlayıb digərlərini qırağa tullayaraq **resultset**-ə həmən sətirdəki **SELECT** hissəsində əlavə etdiyimiz sütunlardakı məlumatı çıxardacaqdır.