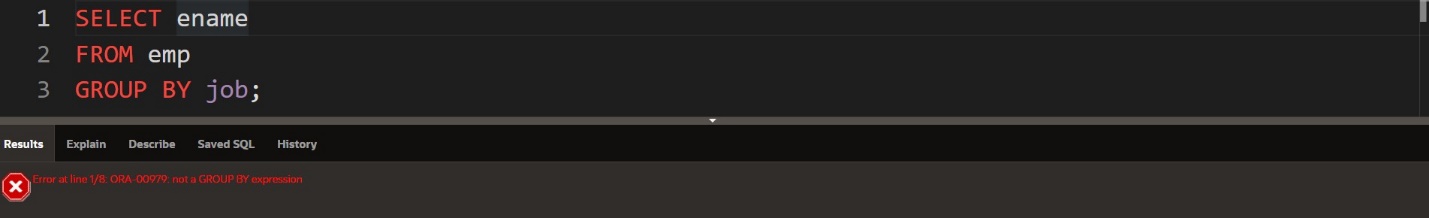
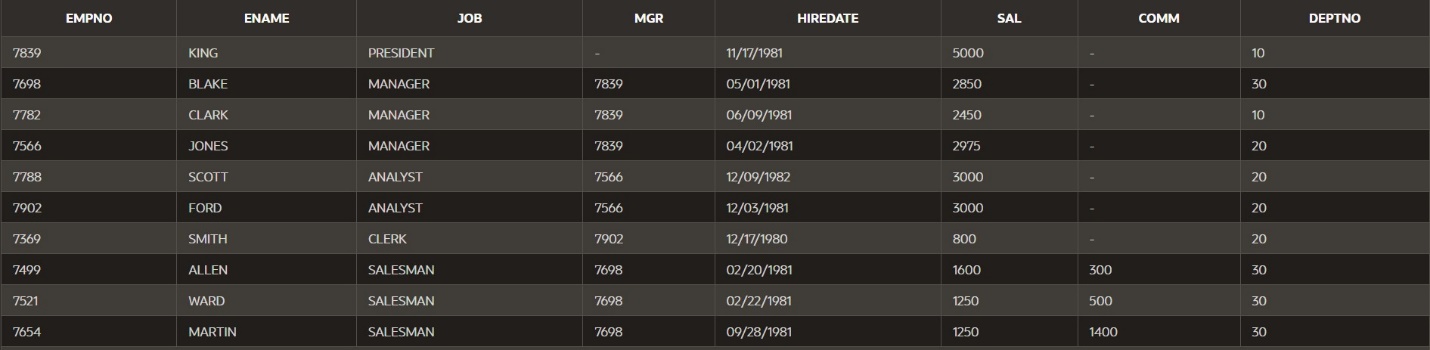
1. Bu dərslikdə biz **aggregate** functionlar ilə birlikdə **GROUP BY** clause-nı öyrənmiş olacayıq. Keçən dərsliyin sonunda bizim qarşımızda nə üçün **GROUP BY** clause-ın istifadə olunduğunu öyrənmişdik və həmçinin öyrənmişdik ki, **GROUP BY** clause bizə hansı üstünlükləri qazandırır.
2. **SQL** də **DISTINCT** keyword-ünü keçdikdə onun nə işə yaradığını öyrənmişdik, **DISTINCT** keyword-ünün gördüyü birə-bir eynisini biz həmçinin **GROUP BY** keyword-ü vasitəsi ilə həyata keçirdə bilərik, eynən aşağıdakı şəkildəki kimi.

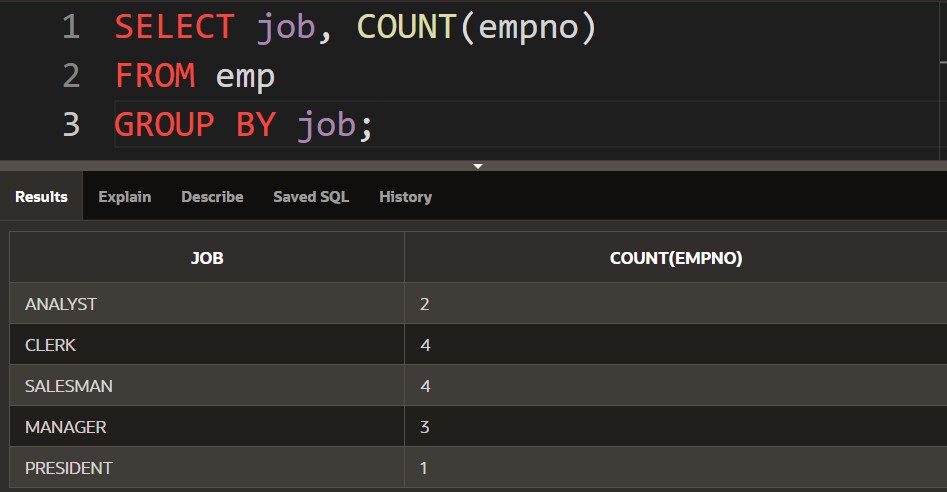
gördüyümüz kimi yuxarıdakı şəkildə olan query vasitəsi ilə biz **job** sütununda olan dəyərlərdən yalnızca **uniqueness** olanları əldə etdik, **duplicate** olanlar isə remove olundu.

* **NOTE: GROUP BY** keyword-ün istifadə edərkən, diqqətli olunmalı məqam budur ki, hansı column-a əsasən **GROUP BY** keyword-ün istifadə edəcəksinizsə, həmən **column SELECT** hissəsində olmalıdır, əks halda xəta ilə qarşılaşacaqsınız, eynən aşağıdakı şəkildə olduğu kimi.

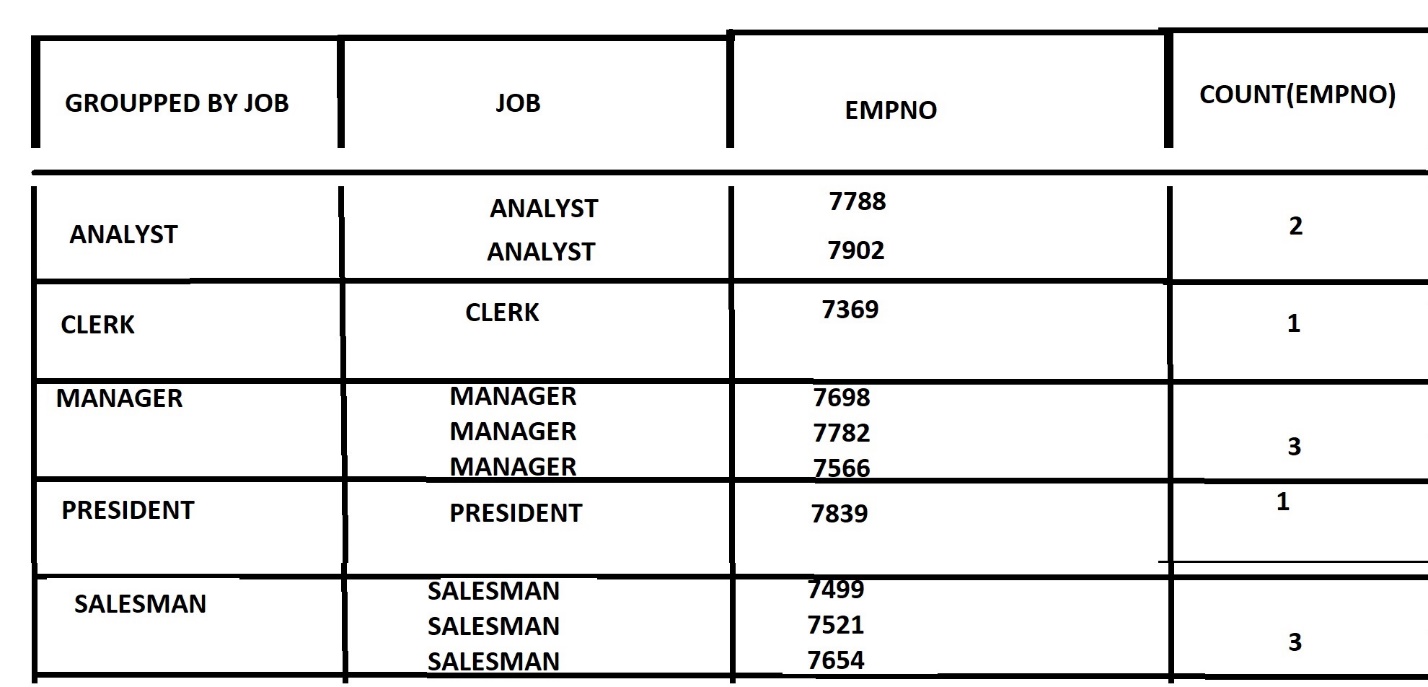
gördüyünüz kimi, yuxarıdakı şəkildə olan query-də biz xəta ilə qarşılaşdıq, buna səbəb biz **SELECT** clause-ında **GROUP BY** da istifadə etdiyimiz **column**-uqeyd etməmişik.

**GROUP BY necə işləyir?**

Fərz eləyin ki, aşağıdakı şəkildəki kimi bizim əlimizdə bir **emp** table-ı vardır.

və biz bilmək istiyirik ki, məsələn **job** sütununda olan dəyərlərə aid neçə sətir record vardır. Yəni ətraflı izah etmək istəsək, belə başa düşə bilərsiniz ki, məsələn, mən bilmək istəyirəm ki, **ANALYST** vəzifəsində neçə employee vardır, **SALESMAN** vəzifəsində neçə employee vardır və s. və ilaxır. Bunu bilmək istəyirik isə aşağıdakı şəkildə olan query-ni yazıb çalışdıra bilərik.

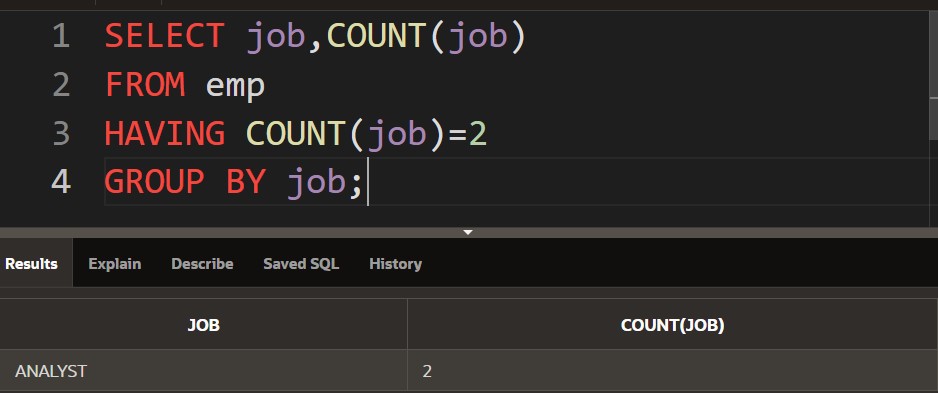
gördüyümüz kimi, yuxarıdakı şəkildə olan query-də olan **resultset** də istədiyimiz nəticə çıxmış oldu. İndi isə gəlin bunun necə baş verdiyini yəni **GROUP BY** keyword-ünün necə işlədiyini anlamağa çalışaq.

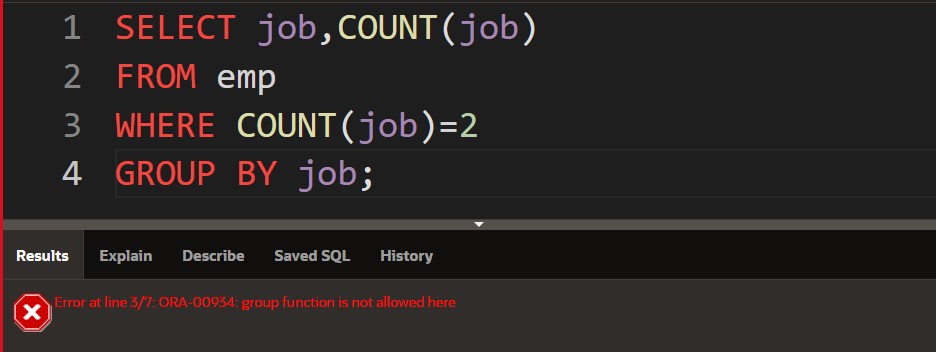
**GROUP BY**-ın işləmə prinsipinə keçməmişdən öncə sizi aşağıdakı şəkil ilə tanış etmək istərdim.

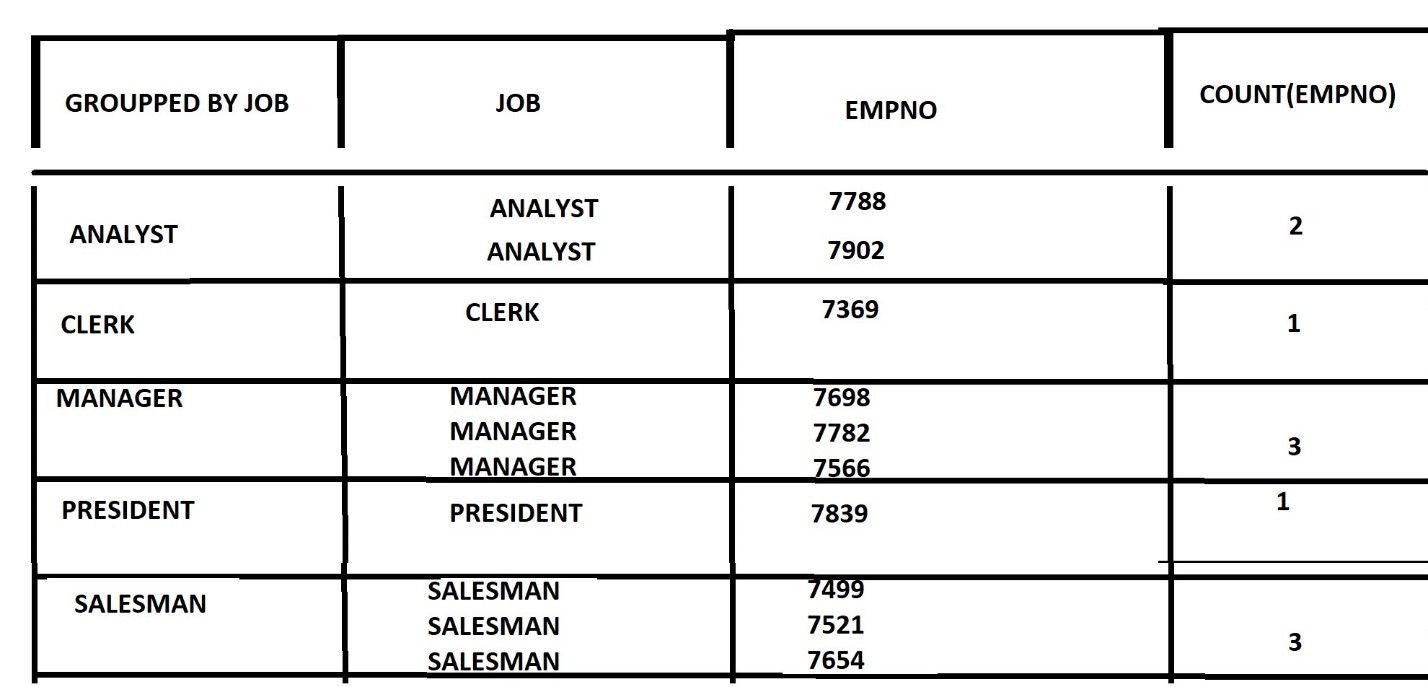
1. Gördüyünüz kimi, **SQL** ilk öncə **GROUP BY** clause-na vermiş olduğunuz sütuna görə əlavə bir sütun yaradır **resultset**-də.
2. Daha sonra əlavə yaradılmış sütunda olan dəyərlərə əsasən neçə sətir uyğun gəlirsə həmən sətirlərin dəyərlərini bir yerdə combined yəni birləşdirərərk onların sayını ən sonda yuxarıdakı şəkildə qeyd etdiyim **COUNT()** funksiyasının vasitəsi ilə həmən sətirlərin sayını hesablıyaraq bizə o nəticə qaytarmış olur.

**TASK CHALLENGE**

Elə bir query yazki, **job** sütununda olan dəyər table-da 2 sətirdədir.

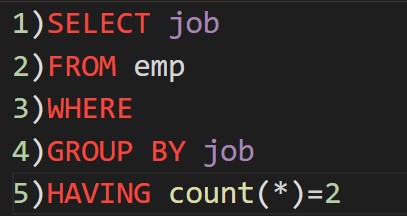
**TASK SOLVING**

* **NOTE:** Burada query-ni yaza bilməməyiniz normaldır, əgər **GROUP BY** da yenisiniz isə, əgər siz yuxarıdakı şəkildə **WHERE** clause-nı işlətmisiniz isə, bu zaman xəta ilə qarşılaşacaqsınız eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi, burada xəta ilə qarşılaşdıq, buna səbəb **WHERE** clause **aggregate functionlar(COUNT(), MIN(),MAX()** və s.**)** qəbul etmir. Lakin bundan əlava yəni, niyə görə **WHERE** clause-nın **aggregate function-lar** ilə işlədilə bilmədiyini bilmək istəyirsinizsə bu zaman bilməlisiniz ki, **aggregate** functionlar **grouping** functionlar olduğundan yəni onlar qrup şəklində olan recordlar daxilində iş gördüyündən və bizdə bilirik ki, **WHERE** clause-ı hər bir sətirə daxil olub filterlama əməliyyatı apardığından, məhz ona görə **WHERE** clause-ı toplu bir qrup içinə girib əməliyyat apara bilməz, eynən aşağıdakı şəkildə olduğu kimi.

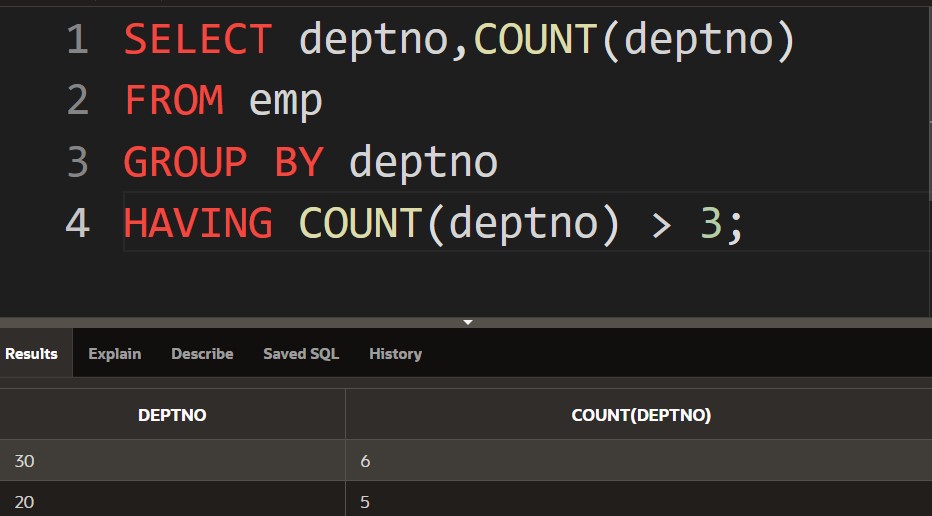
Baxın burada **WHERE** clause toplu bir qrup deyərkən yuxarıda sətir bə sətir ayrılmışlar tək bir sətir olmadığından onlar bir toplu qrup və həmən o qrupun içində sətirlər olduğundan WHERE clause-ı burada işləmir. **HAVING** isə burada **WHERE** clause-ının birə-bir gördüyü əməliyyatı, yəni **filterlama** əməliyyatını həyata keçirdir, lakin **HAVING** clause-ının **WHERE** clause-ından fərqi ondadır ki, o məhz **qrupları** **filterlama** prosesindən keçirdir.

* **NOTE: WHERE** clause-ı **aggregate** funksiyalarnan işləmir demək o demək deyil ki, biz həmən **SELECT** query-sində **WHERE** clause-nı işlədə bilmiyəcəyik xeyr işlədəcəyik, sadəcə olarağ onu **aggregate** funksiyalar ilə işlədə bilmərik və həmçinin bilməli olduğunuz digər məqamda odur ki, **HAVING** clause-ı istifadə yeri optionaldır, yəni istəsəniz **GROUP BY** dan əvvəl gələ bilər, istərsə də **GROUP BY** dan sonra gələ bilər.



**TASK CHALLANGE**

1. 3-dən artıq işçiyə sahib olan departmentin nömrəsini qaytaran query yazın.

**TASK SOLVING**