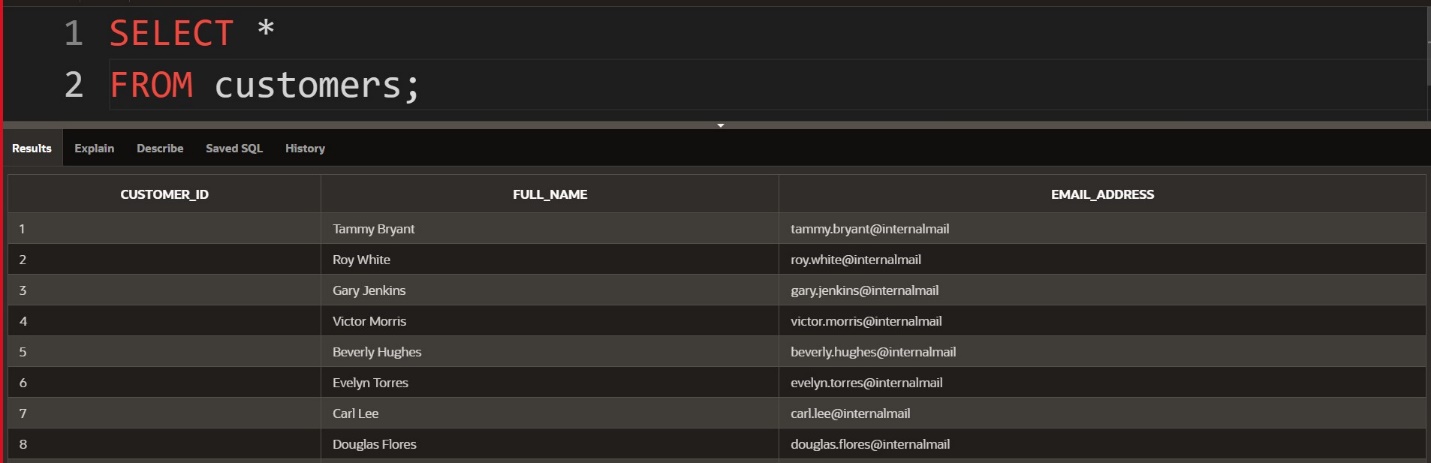
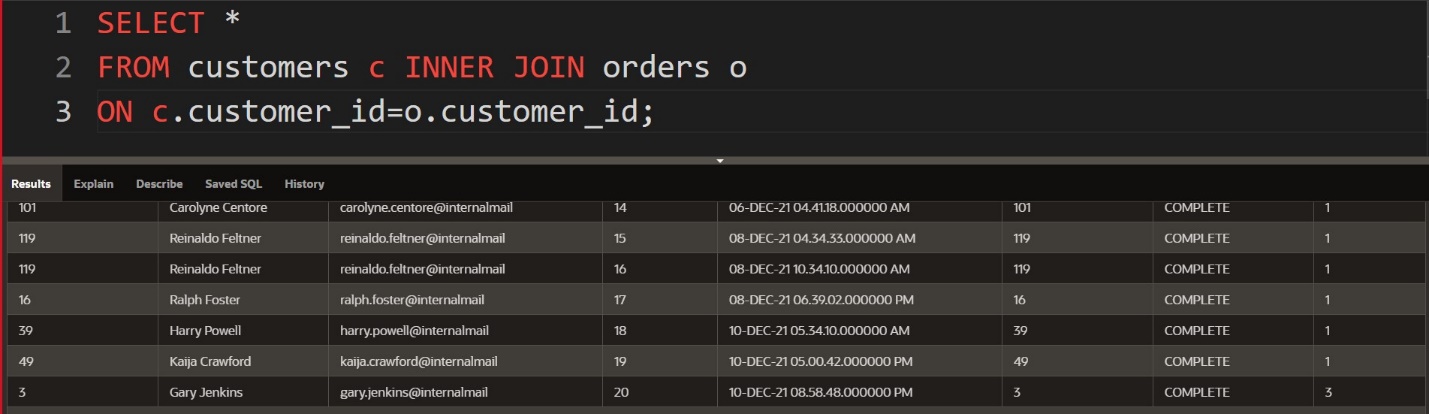
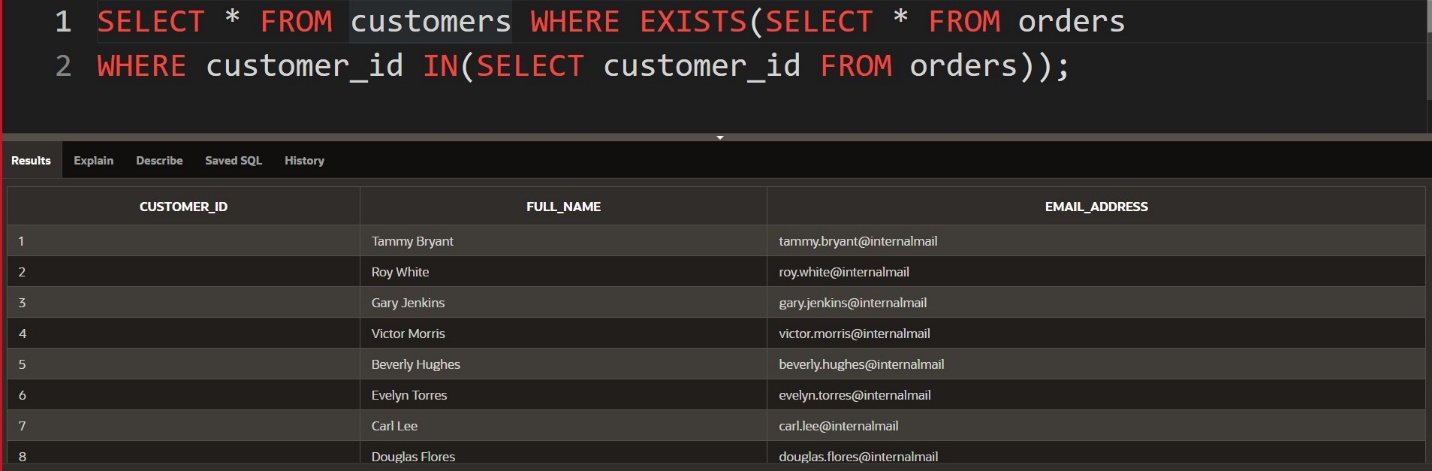
1. **SQL** də **EXISTS** operatoru **otuher** query ilə **sub query** arasında müqayisə etmək üçün istifadə olunur. Daha aydın olması üçün, aşağıdakı misallara baxaraq anlaya bilərik.

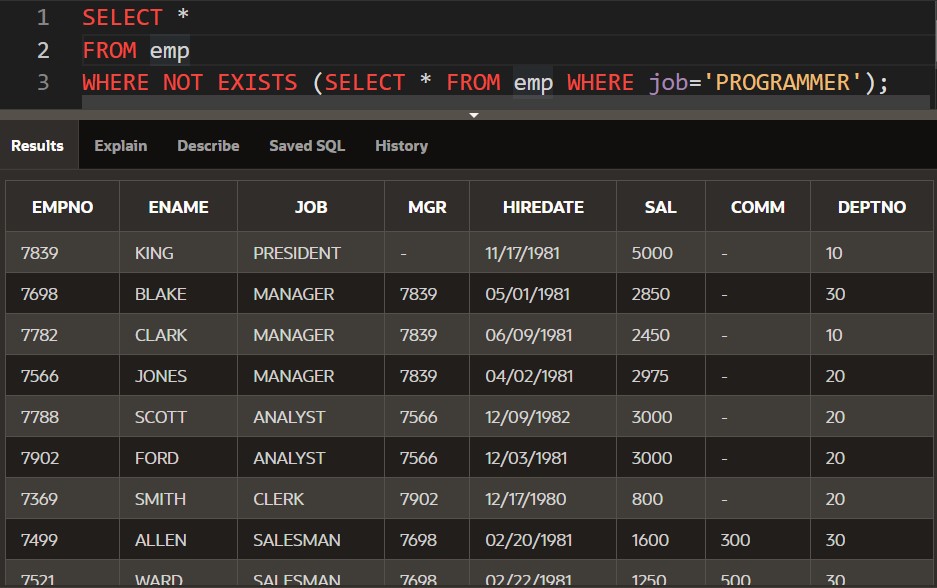
Məsələn fərz eləyin sizin bir databazavız vardır və orada **customers** və **orders** adında 2 ədəd table vardır. Və fərz eləyin ki, günlərin bir günü direktoruvuz gəlir deyir ki, mənə mağazamızda o müştərilərin məlumatlarını göstər ki, heç olmasa ən azından 1 ədəd sifarişi olsun. Bu halda siz əgər **customers** table-ından aşağıdakı şəkildə olan query-ni yazıb çalışdırsanız, sizin qarşınızısa sadəcə olaraq **customer**-ların məlumatları çıxmış olacaqdır, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi, yuxarıdakı şəkildə olan query də biz sadəcə bütün müştərilərin məlumatlarını əldə etmiş olduq, lakin biz əminlik ilə deyə bilmərik ki, burada olan müştərilər həqiqətəndəmi sifarişləri varmı?! Sifarişə sahib olan müştərilərin məlumatlarını öyrənmək istəyiriksə, əminəmki siz gedib də aşağıdakı query-ni yazıb çalışdıracaqsınız.

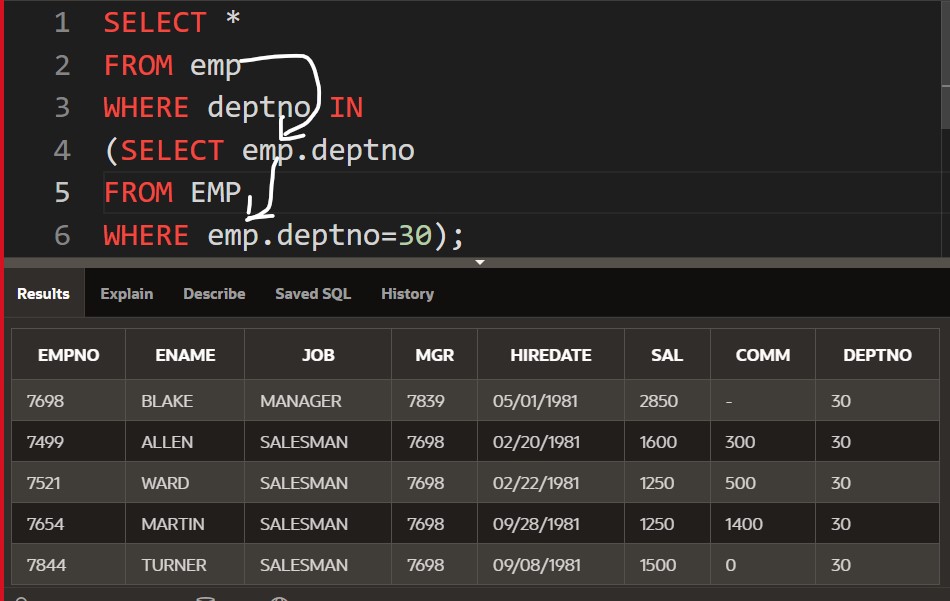
gördüyünüz kimi, yuxarıdakı şəkildə olan query də biz **INNER JOIN** növündən istifadə edərək, istədiyimiz nəticəni əldə etmiş olduq. Gəlin indi isə bunu **EXISTS** operatoru vasitəsi ilə necə edərik onu anlamağa çalışaq, eynən aşağıdakı şəkildə olduğu kimi.

gördüyümüz kimi, bu dəfədə biz istədiyimiz nəticəni **EXISTS** operatorundan istifadə edərək əldə etmiş olduq.

**NOT EXISTS**

1. **EXISTS** dən fərqli olaraq **NOT EXISTS** subquery record qaytarmadığı təqdirdə outher query çalışmağa başlıyacaqdır, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi, **emp** table-ında **job** sütununda **PROGRAMMER** dəyərinə aid **sətir** olmadığından, yəni **NOT EXISTS** şərti ödənildiyindən bizə **emp** table-ından datalar qayıtmış oldu.

* **NOTE: SQL** də correlated query deyilən bir anlayış vardır ki hansıki, aşağıdakı şəkildən nə olduğunu anlaya bilərsiniz.

gördüyünüz kimi, yuxarıdakı şəkildə altından ağ ilə işarələdiyim hissə correlated query adlandı. Bu o deməkdir ki, siz **sub-query** daxilində **outher-query** üzvündən istifadə edə bilərsiniz, ancaq **outher-query** daxilində **sub-query** üzvündən istifadə edə bilməzsiniz.