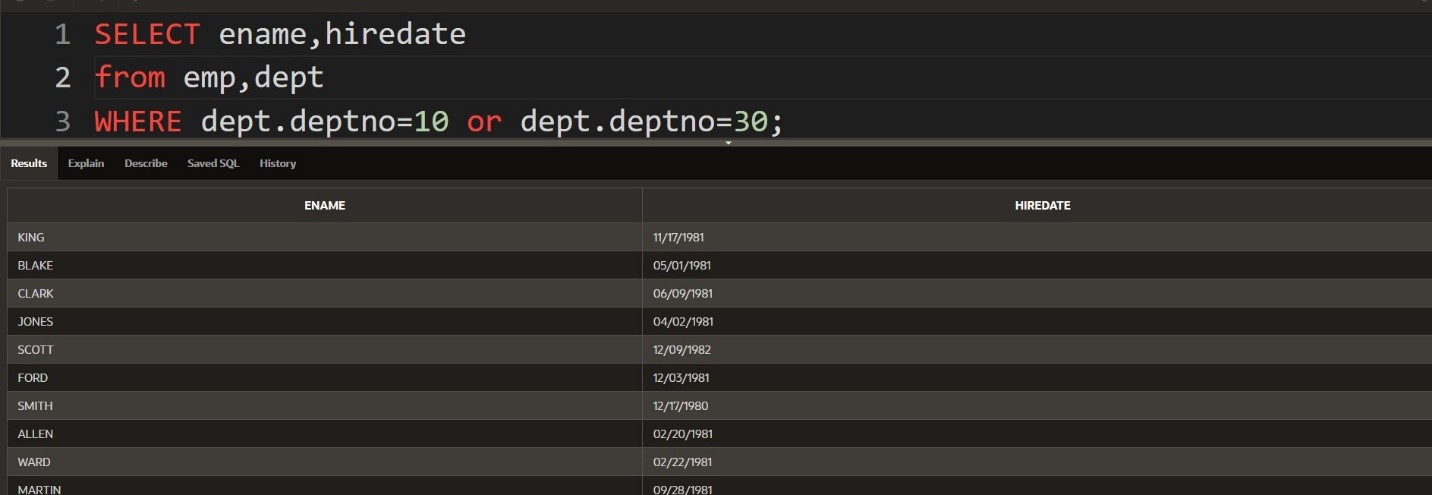
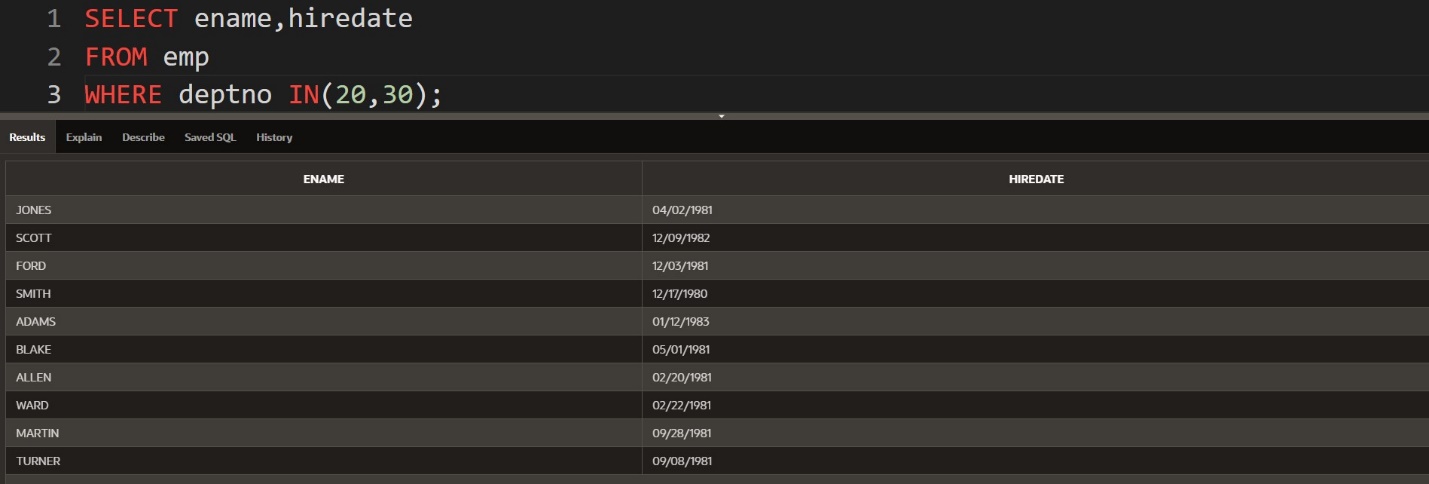
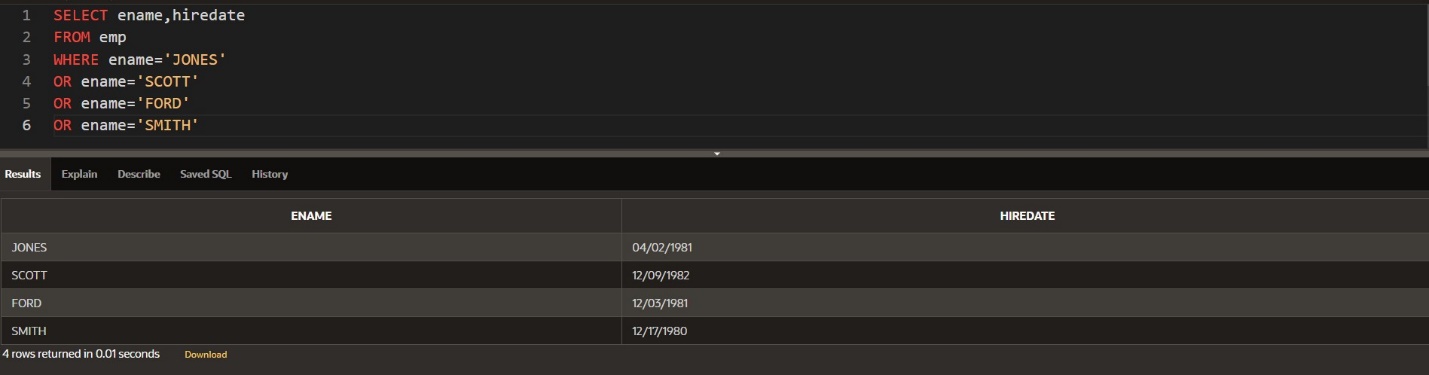
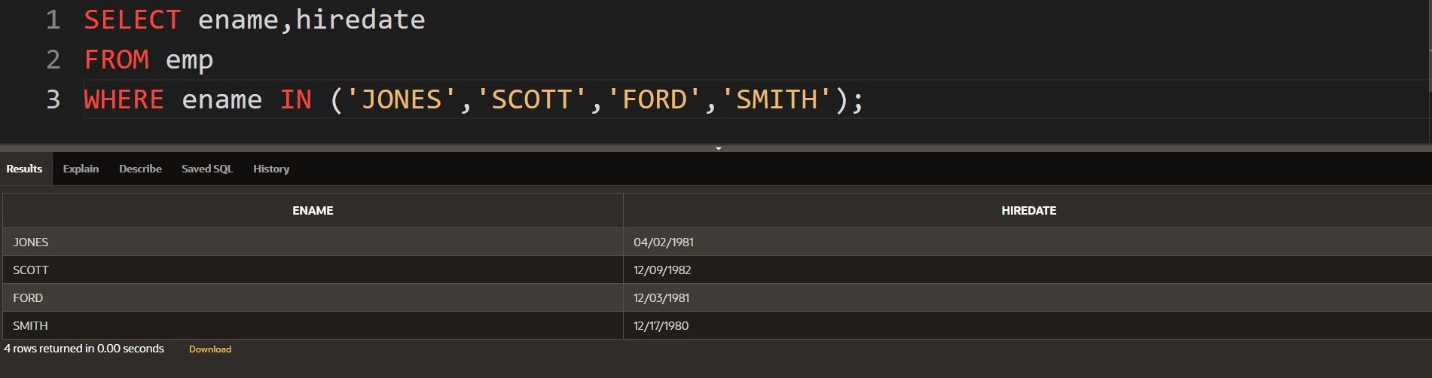
1. Bu dərslikdə biz **SQL** də **IN** və **BETWEEN** clause-larını öyrənmiş olacayıq.
2. Keçən dərslikdə biz sonuncu **challenge** da bir query yazmışdıq aşağıdakı şəkildəki kimi.
3. Yuxarıdakı şəkildə olan query-ni isə bu dəfə daha asand yola gətirib onu flexibile edəcəyik və bunu bu gün öyrənəcəyimiz **IN** clause-ı vasitəsi ilə həyata keçirdəcəyik aşağıdakı şəkildə olan query kimi.

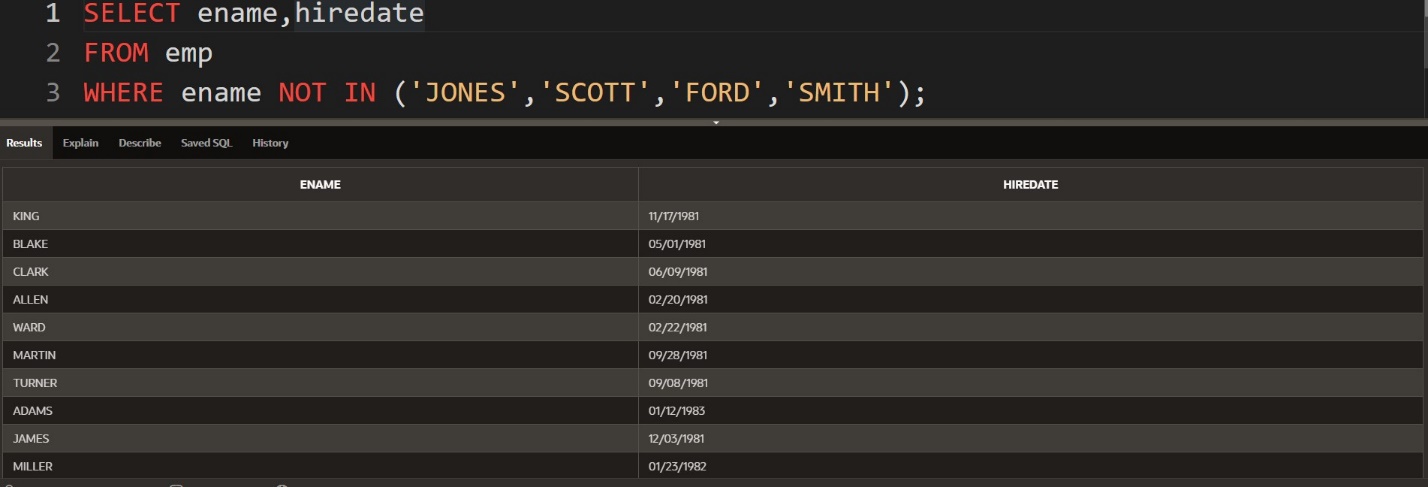
gördüyünüz kimi burada biz bir öncəkindən fərqli olaraq **OR** ları yığışdırıb yerinə **IN** clause-ından istifadə edərək eyni nəticəni əldə etmiş olduq. Sual verə bilərsiniz ki, nəyə görə bunu istifadə etdik fərq nədir axı? Fərq burasındadır ki, **IN** operatoru **SQL** də çoxlu sayda **OR** operatorunun yazılışının qarşısını alaraq bizim query-lərimizi daha flexibile edir. Məsələn fərz edin ki, bizim aşağıdakı kimi bir query-miz vardır və burada çoxlu sayıda **OR** operatorlarından istifadə etmişik.

yuxarıdakı şəkildə olan query-dən gördüyümüz kimi, bu necədə can sıxıcı və ürək bulandırıcıdır. Lakin biz bunun daha alternativ variantını aşağıdakı şəkildə göstərildiyi kimi yaza bilərik.

gördüyünüz kimi yuxarıdakı şəkildəki olan query bir öncəki yuxarıdakı olan şəkildəki query-dən daha alternativ həll yoludur.

* **NOTE: IN** clause-nı anlamağın ən yaxşı yolu ingilis dilində olan conta**IN**s mənası kimi başa düşə bilərsiniz. Yəni qısa deyimlə desək filan sütuna görə filterlama əməliyyatı edərkən bu sütun özündə bu dəyərləri conta**IN**s (saxlayırmı?) .

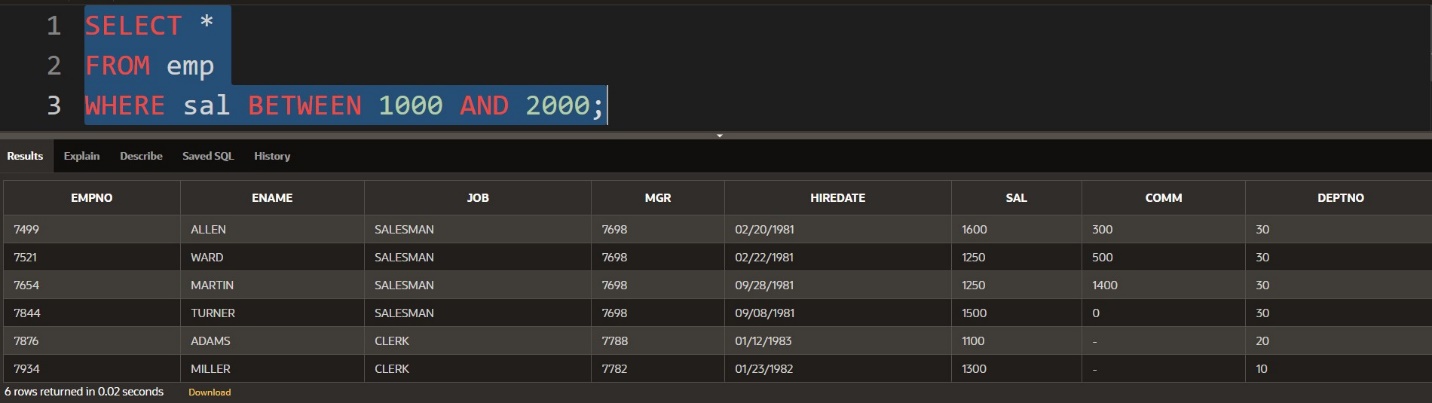
1. Daha sonra elə hal ola bilər ki, biz filan dəyərləri özündə saxlamayanlardan başqa məlumatları almaq istəyə bilərik bu zaman isə **IN** clause-ının inkar variantı olan **NOT IN** clause-ından istifadə edəcəyik. Məsələn aşağıdakı şəkildə olan query kimi.

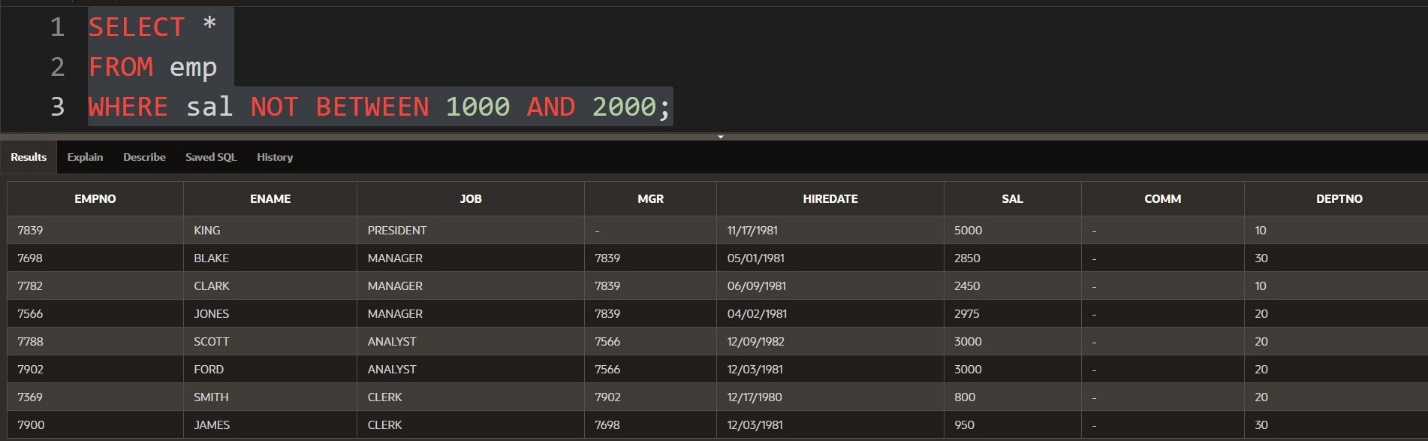
Gördüyünüz kimi yuxarıdakı şəkildə olan query-də biz **ename** column da bu dəyərlərdən başqa hansı dəyərləri özündə conta**IN**s edirsə onları əldə etmiş olduq.

**BETWEEN**

1. **BETWEEN** operatoru **SQL** də verilmiş aralığa əsasən məlumatları gətirməkdədir, əgər aralıqda elan olunan condition özünü doğruldaqcaqdırsa bu zaman həmən aralığ daxilinə tuş gələn şərtə əsasən biz dataları əldə etmiş olacayıq. Elə isə gəlin aşağıdakı şəkildə olan query-ə nəzər yetirək.

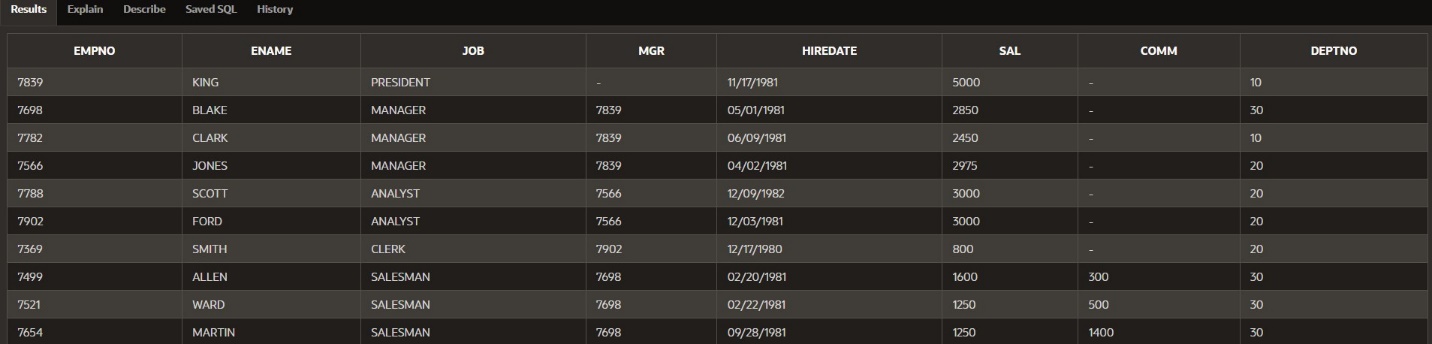
Gördüyümüz kimi burada biz **hiredate** column-a əsasən **BETWEEN** operatorundan istifadə etmiş olduq və burada dedik ki, **bizə 1981-ci ildən 1988-ə qədər olan aralığına tuş gələn** hiredate **sütununda olan dəyərlərə görə employee-ların məlumatlarını ver.** Qeyd edək ki, burada biz **DATE** tipinə xas olan dəyərlərə görə **BETWEEN** operatorunu istifadə etmiş olduq. İndi isə gəlin rəqəmlər ilə necə istifadə olunur aşağıdakı şəkildə olan query-ə baxaq.

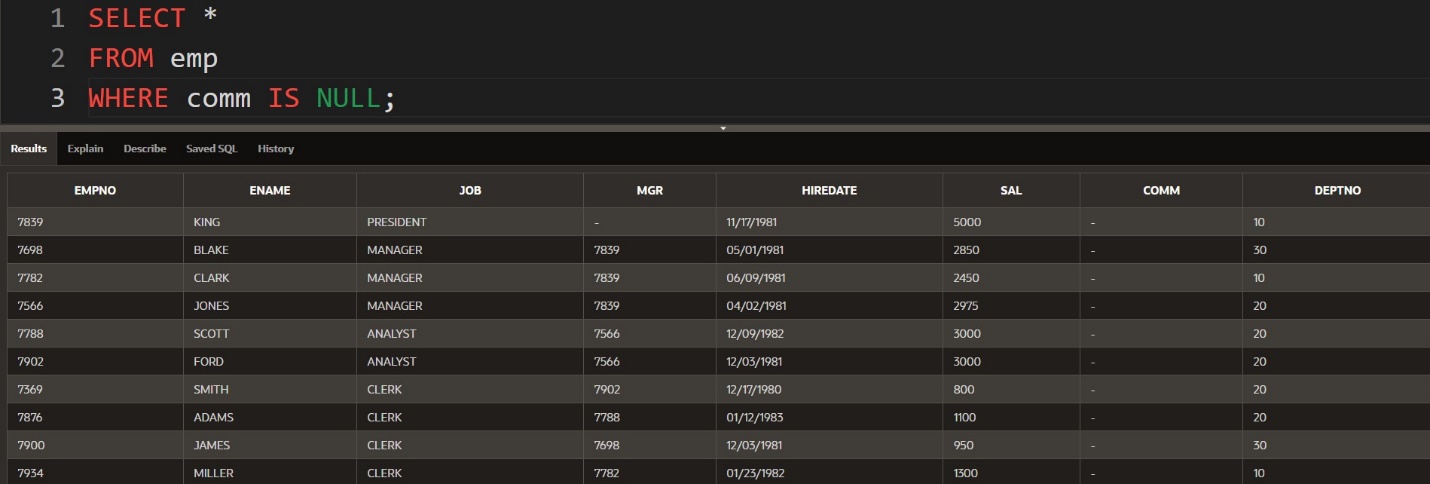
gördüyümüz kimi burada biz **BETWEEN** operatorunu rəqəmlər ilə istifadə qaydasını görmüş olduq.

1. Eynən **IN** də **NOT** operatorunu birlikdə inkar formasında işlətdiyimiz kimi, **BETWEEN** də də bu mümkündür, aşağıdakı şəkildə olan query-dən baxa bilərik.

Gördüyümüz kimi burada biz **NOT BETWEEN** operatorundan istifadə edərək demiş olduq ki, **sal colum-u bu aralıq dəyərlərə sahib olmayan işçilərin məlumatlarını ver.**

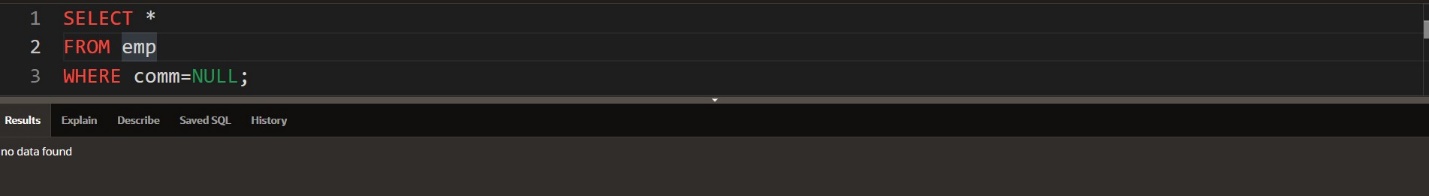
**IS NULL**

1. **IS NULL** filan sütuna görə apardığımız filterlama əməliyyatında yalnızca filterlama apardığımız sütunda **NULL** dəyərə uyğun gələn dataları əldə etmək istədiyimizdə istifadə olunan operatordur. Məsələn aşağıdakı şəkildə olan table-da **comm** sütunu özündə həm **null** və həmçinin **null** olmayan dəyərlər saxlayır, lakin bizim istəyimiz odur ki, burada **comm** sütunu yalnızca null dəyərlərə ****sahib olan dataları əldə etmək istəyirik.

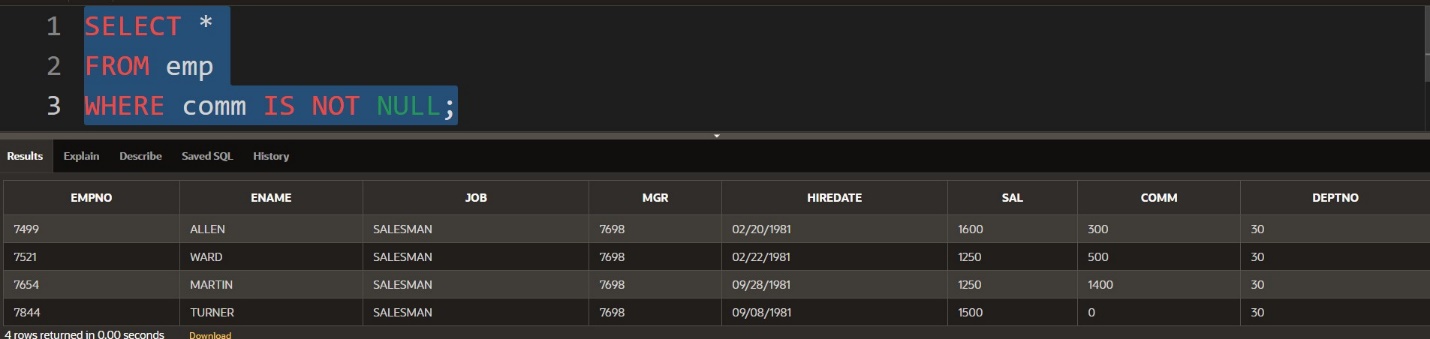
Belə olduğu halda aşağıdakı şəkildən bunun nə demək olduğunu daha yaxşı anlaya bilərik.

Gördüyümüz kimi yuxarıdakı şəkildə olan query-də biz **comm** sütununda null dəyərə sahib olan sətirlərdəki dataları əldə etmiş olduq.

* **NOTE:** SQL də özündə **NULL** saxlayan sütunu **=(equals)** operatoru ilə müqayisə etmək mümkün deyil, bunun əvəzinə **IS NULL** keyword-ündən istifadə edirik. Özündə **NULL** saxlayan sütunu **=(equals)** operatoru ilə müqayisə edə bilməməyimizə səbəb ona görədir ki, **=(equals)** operatoru mövcud dəyərlər üzərində müqayisə aparmaqdadır. Məsələn, aşağıdakı şəkildə olan query-ni çalışdıran biz boş bir **resultset** ilə qarşılaşmış olacayıq.

****

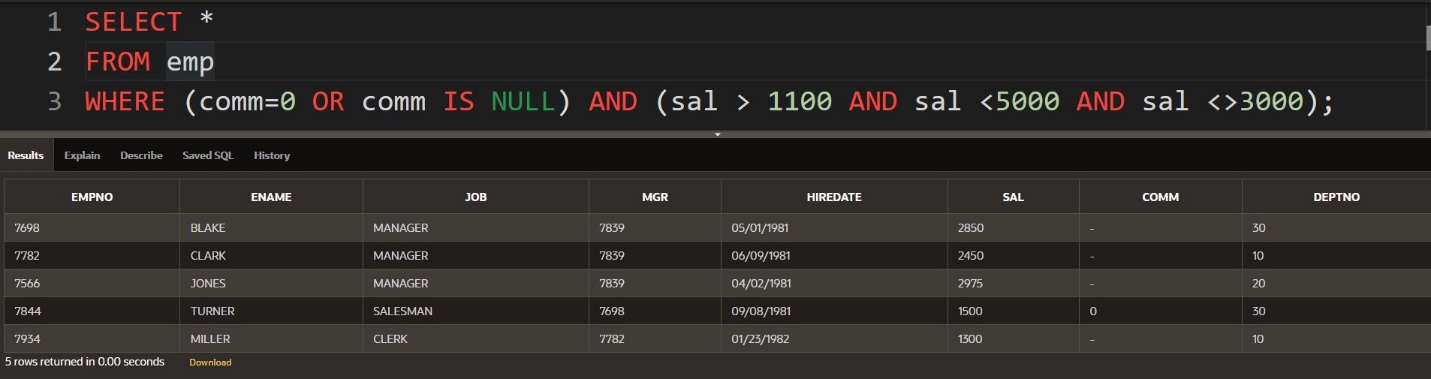
1. Birdə bunun inkar fotmatı var odakı **IS NOT NULL** operatorundan istifadə edilərək yazılmaqdadır, aşağıdakı şəkildə olan query-dən bunu anlaya bilərik.



Gördüyümüz kimi bizə yalnızca **comm** sütununda **null** dəyərə sahib olmayan sətirlərdəki datalar gəlmiş oldu.

**TASK CHALLENGE**

Komisyası olmayan və salary-si 1100 dən böyüg ancaq 5000 dən kiçik və həmçinin salary-si 3000 ə bərabər olmayan işçilərin məlumatlarını qaytaran query yazın.

**TASK SOLVING**