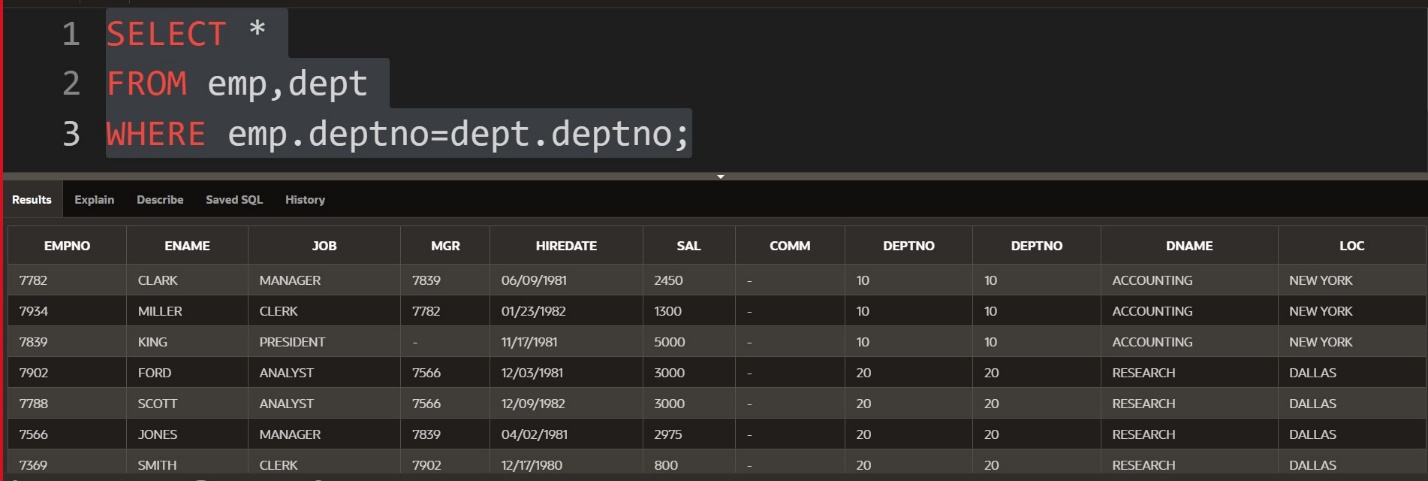
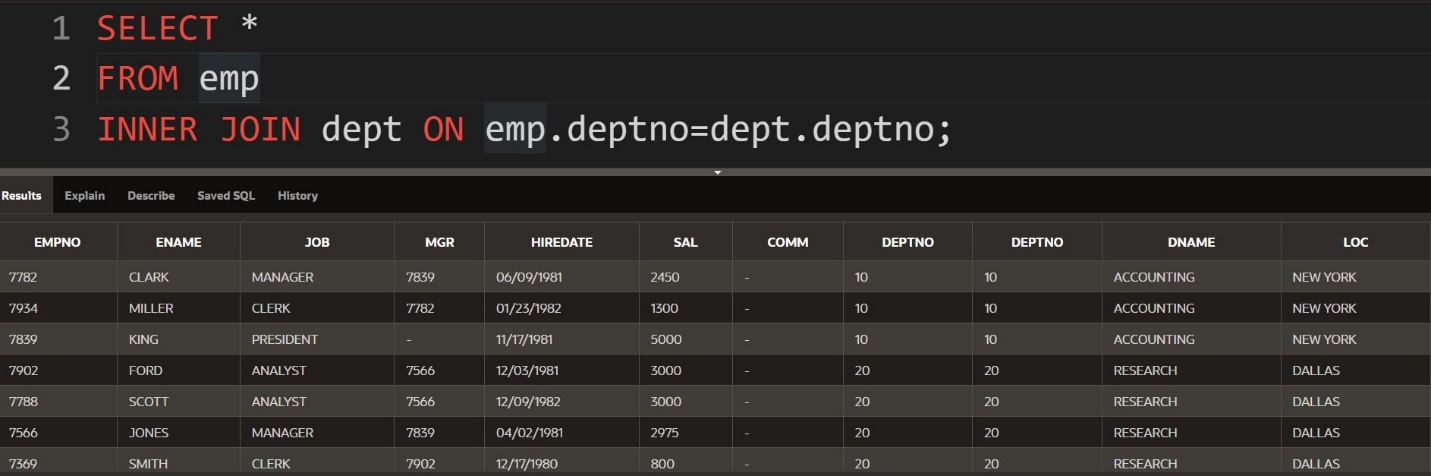
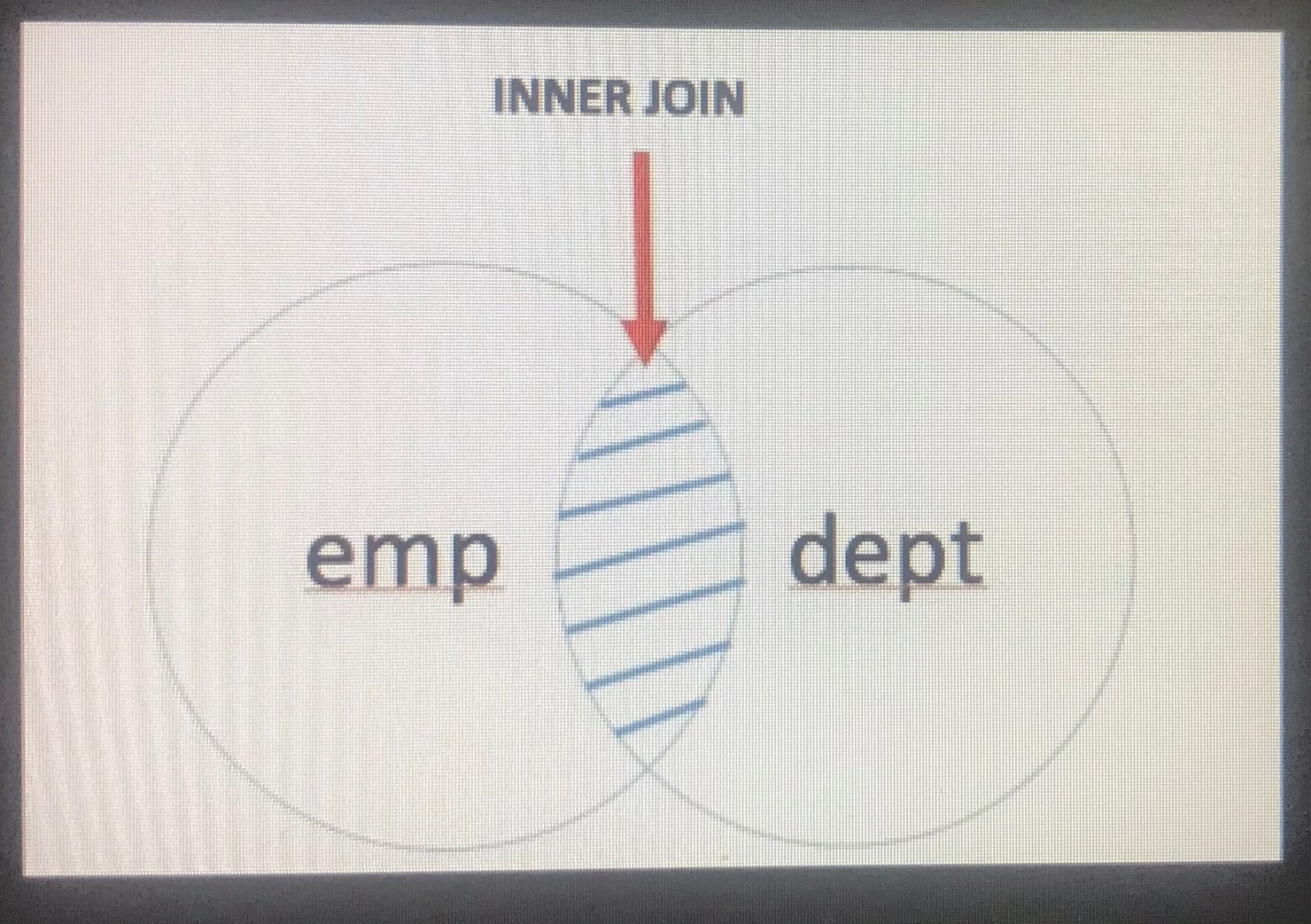
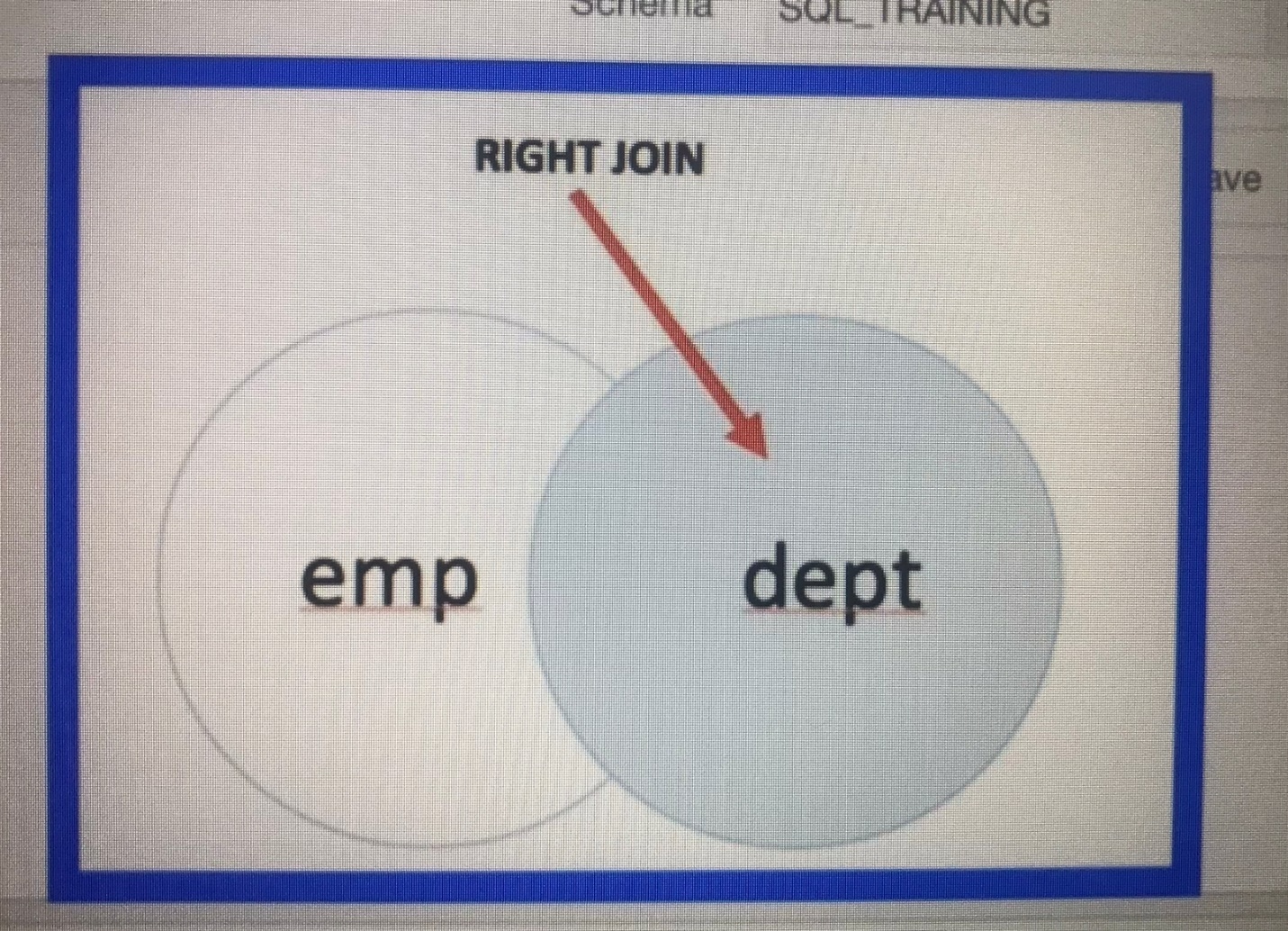
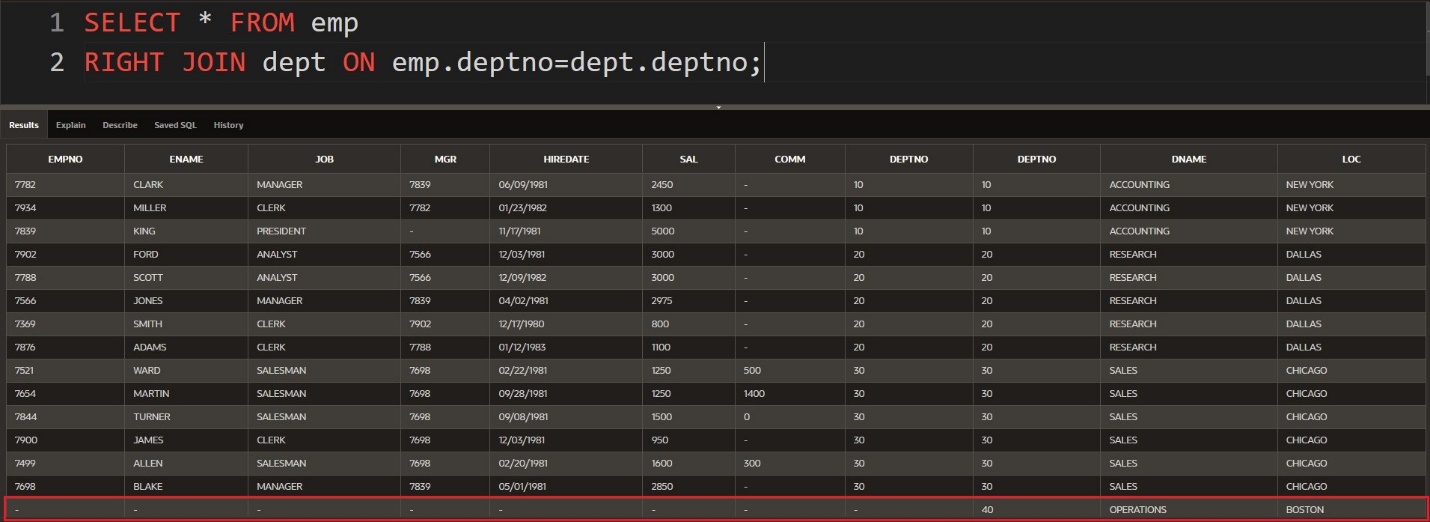
1. Keçən dərslikdə biz belə bir query işlətmişdik, eynən aşağıdakı şəkildə olduğu kimi.

və anlamışdıq ki, bu query məhz hər 2 table-da uyğun gələn sətirləri çəkməkdədir. İndi isə bunun ən alternativ variantı olan **JOIN** lər ilə etmiş olacayıq və yuxarıdakı şəkildə olan query bir növ **INNER JOIN** deməkdir, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi burada biz yeni keyword olan **INNER JOIN** dən istifadə edərək bir öncəki şəkildə göstərdiyim **resultset**-i buradada əldə etmiş olduq. Bəs yaxşı sual verə bilərsiniz ki, ən sonuncu şəkildə əldə etdiyimiz **resultset** elə **birinci** şəkildədə əldə oluna bilinirdi, bunlar arasında fərq nədir kimi sual verə bilərsiniz? Cavab burasındadır ki, siz **birinci** şəkildən fərqli olaraq **ikinci** şəkildə olan query vasitəsi ilə əlavə əməliyyatlar da apara bilərsiniz, təbiiki bunu **JOIN** lər vasitəsi ilə. Həmçinin **INNER JOIN** o join növüdür ki, **JOIN** edəcəyiniz table-lar arasında olan sətirlər biri-birinin qarşılığı şəklində olmalıdır, yəni məsələn **A** table-ında olan sətirdəki dəyər, **B** table-ındada olmalıdır və bu adlanır **INNER JOIN**.

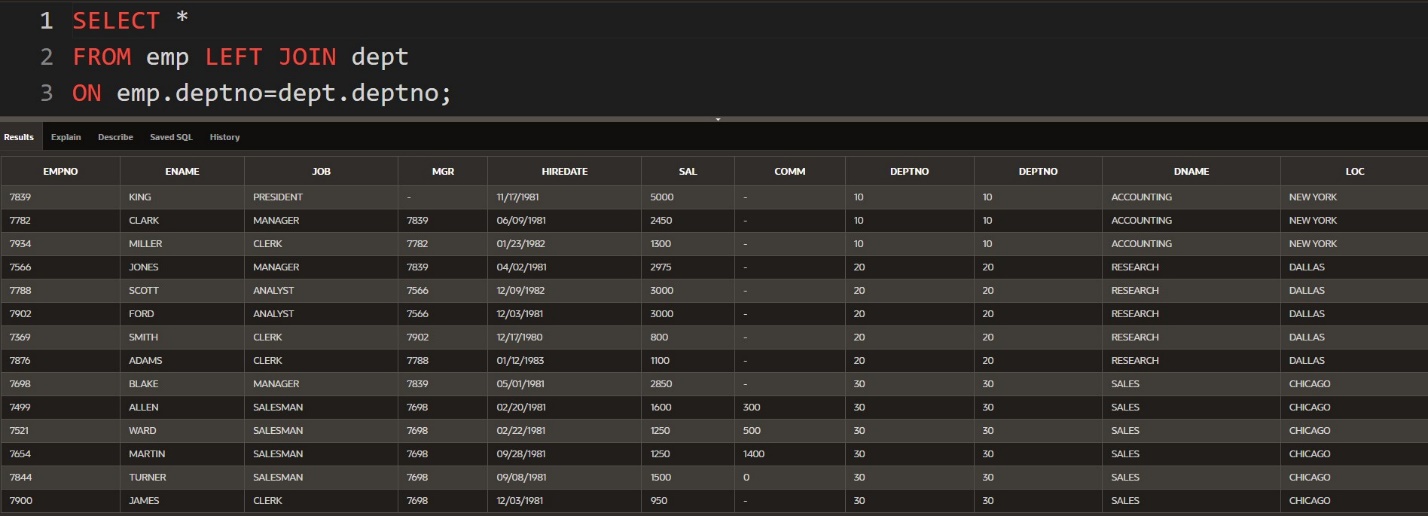
**RIGHT JOIN**

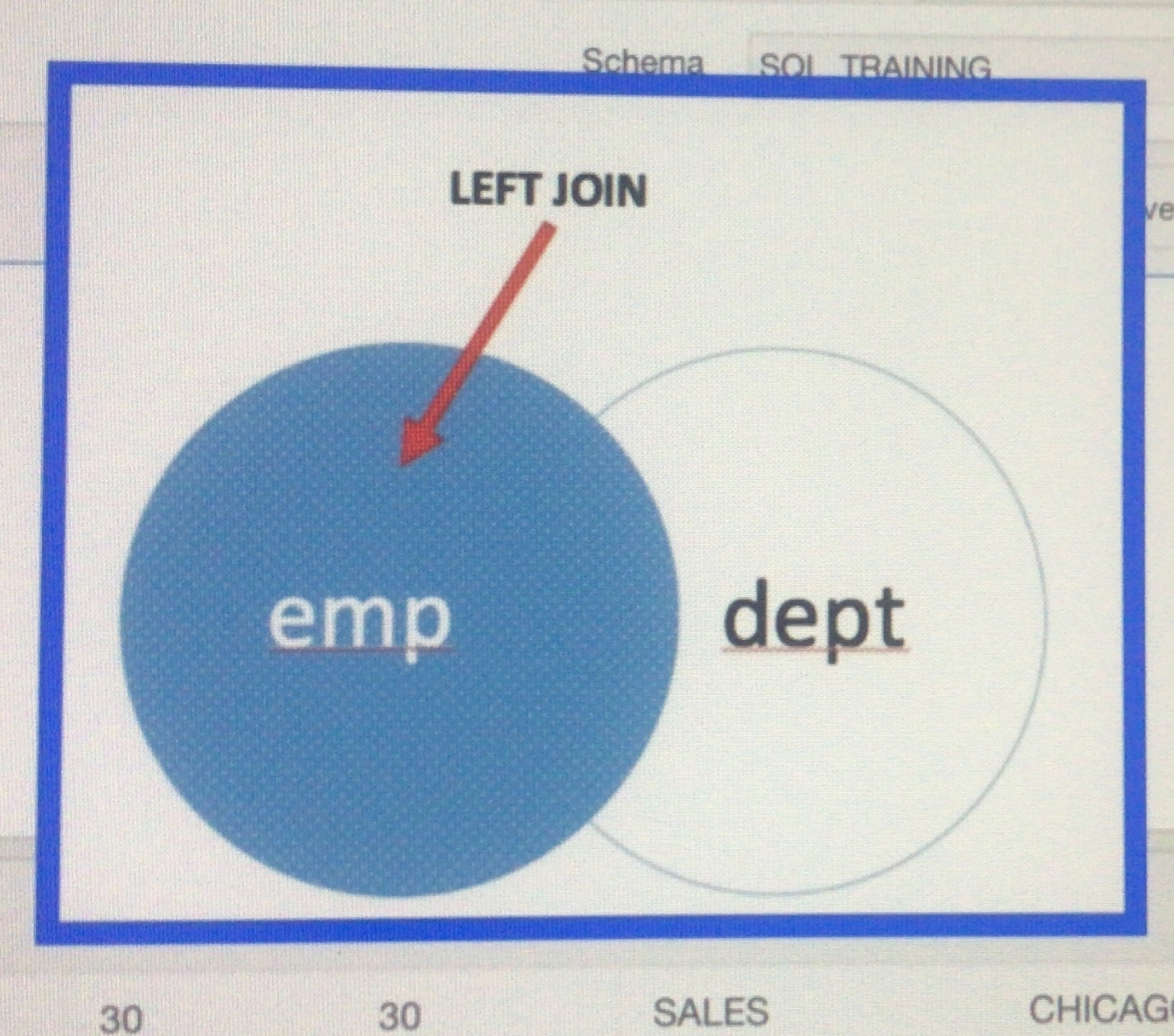
1. Bu **JOIN** növü isə əsasən sağ tərəfdə duran **table** hansıdırsa ondan bütün dataları çəkir, **sol** tərəfə duran table da isə yalnızca sağ tərəfdə uyğun gələn sətirlər göstərilməkdədir, **sol** tərəfdə olubda **sağ** tərəfdə yoxdursa, bu zaman həmin sətirlər aradan qaldırılır.

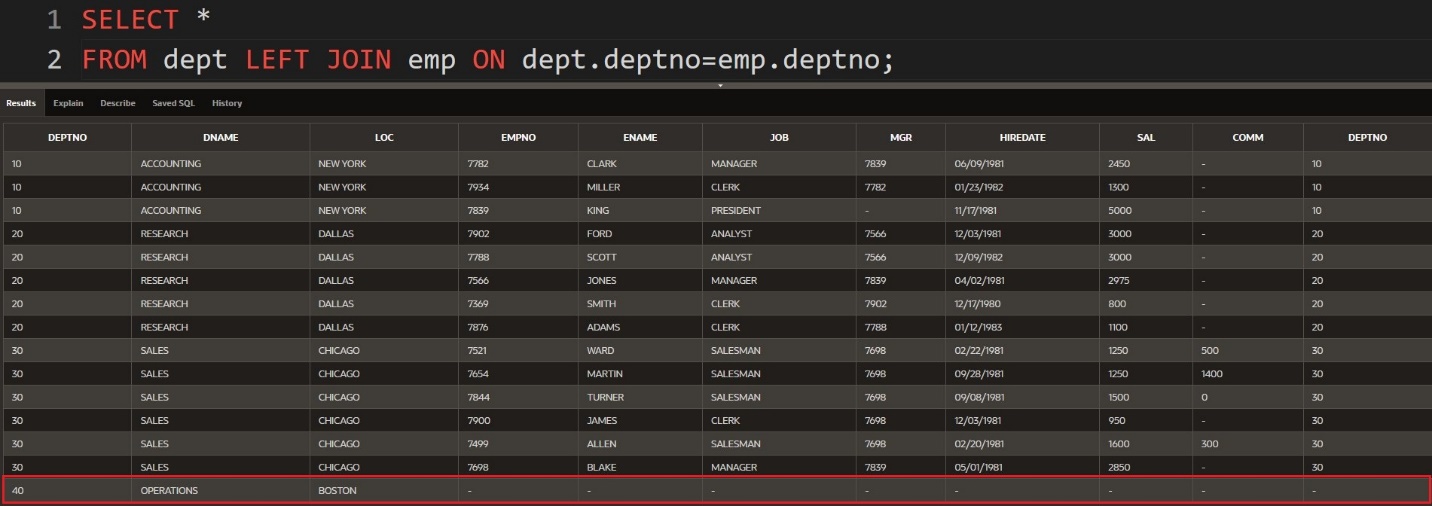
şəkildəndə göründüyünüz kimi, qırmızı rəngli ox əsasən **sağ** da duran **table-a** istinad etməkdədir, buda o deməkdir ki, **SELECT** olunan əsas **table** sağ tərəfdə hansı **table** yerləşdiriləcəkdirsə ora götürülməkdədir. Məsələn gəlin **RIGHT JOIN** a aid bir query yazıb çalışdıraq.

gördüyünüz kimi yuxarıda olan şəkildə biz **RIGHT JOIN** istifadə etmiş olduq. Və buradan alınan resultset də şəkildə altından qırmızı ilə işarələdiyim hissəyə diqqətlə fikir versəniz, yuxarıda izah etdiyimiz **RIGHT JOIN**-in qaydasını daha yaxşı anlaya bilərsiniz. Yəni bu o deməkdir ki, mən əsasən çəkəcəyim hissə **RIGHT** (sağ) table-dan olacaqdır və bizdədə sağ tərəfdə duran table dept olduğundan gördüyümüz kimi dept table-ında olan bütün sətirləri gətirib lakin, sol tərəfdə olan table da sağ tərəfdə olan dept table-ının bu sətiri olmadığından o hissə null olaraq gəlmişdir.

**LEFT JOIN**

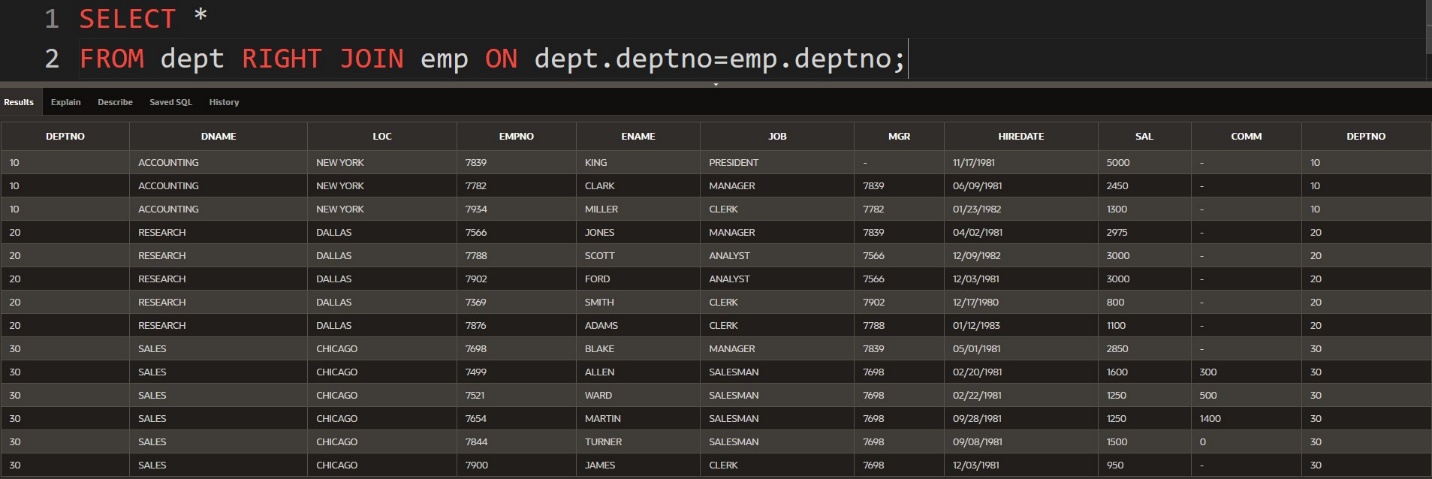
1. **RIGHT JOIN**-in əvəzinə **LEFT JOIN** tam əksin görməkdədir, yəni əsas yük **sol** table-a düşməkdədir, yəni **SELECT** olunacaq hissə ilk öncə **sol** da duran **table**-dan record-ları çəkməkdir, daha sonrasında isə **sağ** da duran o table-dan recordları çəkməkdədir ki, onlar **sol** tərəfdə olan **table** da mütləq şəkildə olsun, əks halda **sağ** tərəfdə olanlar **sol** tərəfdə yoxdursa onlar **resultset** dən **remove** olunacaqdır, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi artıq bizdə yəni **resultset** də bir öncəki şəkildə istifadə olunan **RIGHT JOIN** da ən sonuncu sətirdən çıxan altından qırmızı ilə işarələdiyimiz hissə gəlmədi, buna səbəb bizim burada **LEFT JOIN** dən istifadə etməyimiz oldu, yəni əsas hissə sol tərəfdə olan **emp** table olduğundan oradan recordlar çəkilərək sağa görə müqayisə aparıldı ki, bəs **sağ** tərəfdə olan table-dan o sətirdəki dataları çək ki, onlar həmdə **sol** tərəfdədir, əks halda yəni **sağ** tərəfdə olubda **sol** tərəfdə yoxdursa, bu zaman onlar **remove** olunacaqdır.

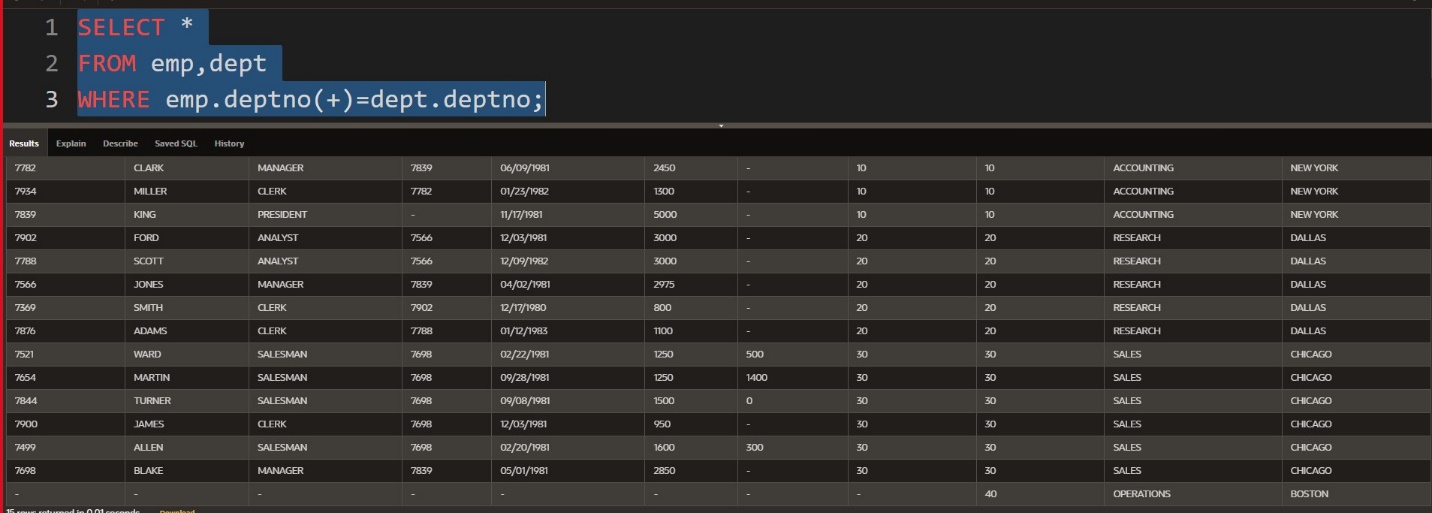
1. Həmçinin biz **LEFT** yaxud **RIGHT** joinlərdən istifadə edərək sadəcə table-ların yerlərini dəyişdirərək, hansı **JOIN** növüdürsə həmən növə uyğun dataları çəkə bilərik, məsələn az öncə biz **RIGHT JOIN** növündən istifadə edərkən sağ tərəfdə **dept** table-ı dayanmışdı və demişdik ki, sağ tərəfdə olan datalar ilk öncə çəkiləcəkdir sol tərəfdəkinə uyğun olaraq və daha sonra isə sol tərəfdə olanlar sağ tərəfdə yoxdursa onlar remove olunacaqdır, indi isə **LEFT JOIN** növündən istifadə edərək sol tərəfə **dept** table-nı əlavə edərək fərqlilik yaradacağam,eynən aşağıdakı şəkildə olduğu kimi.

yuxarıdakı query də biz budəfə sol tərəfə **dept** table-nı əlavə edərək məlumatları çəkmiş olduq.

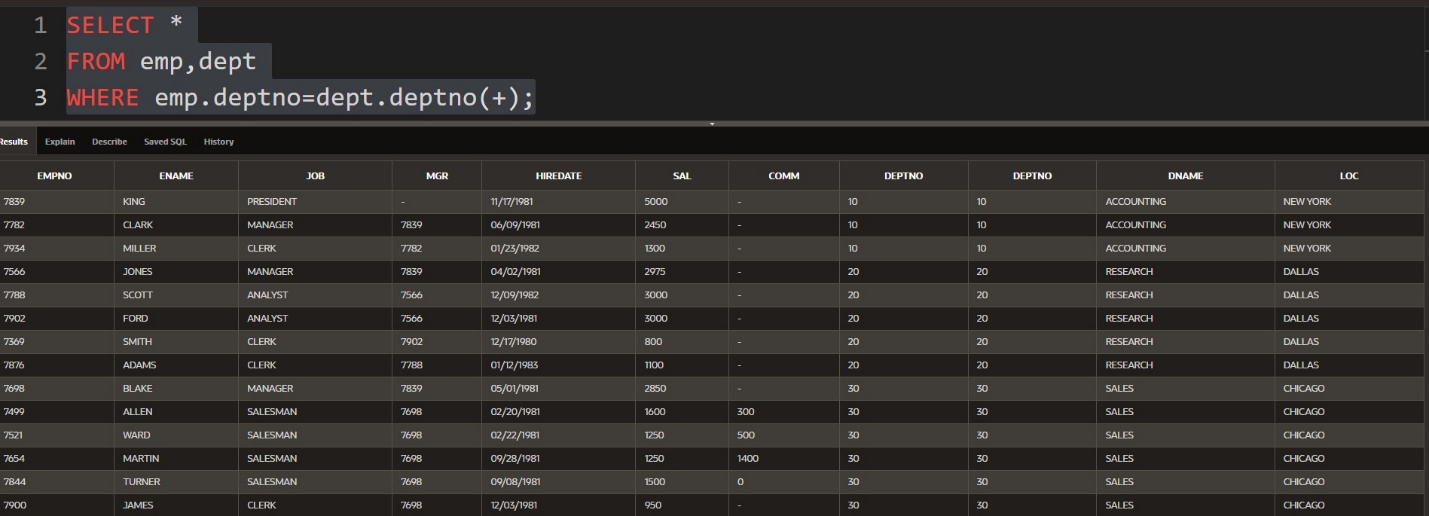
1. Həmçinin belənçinə əks **JOIN** əməliyyatını **RIGHT JOIN** ilədə edə bilərik, eynən aşağıdakı şəkildə olduğu kimi.



**LEGACY RIGHT JOIN TYPE EXPLANATION**

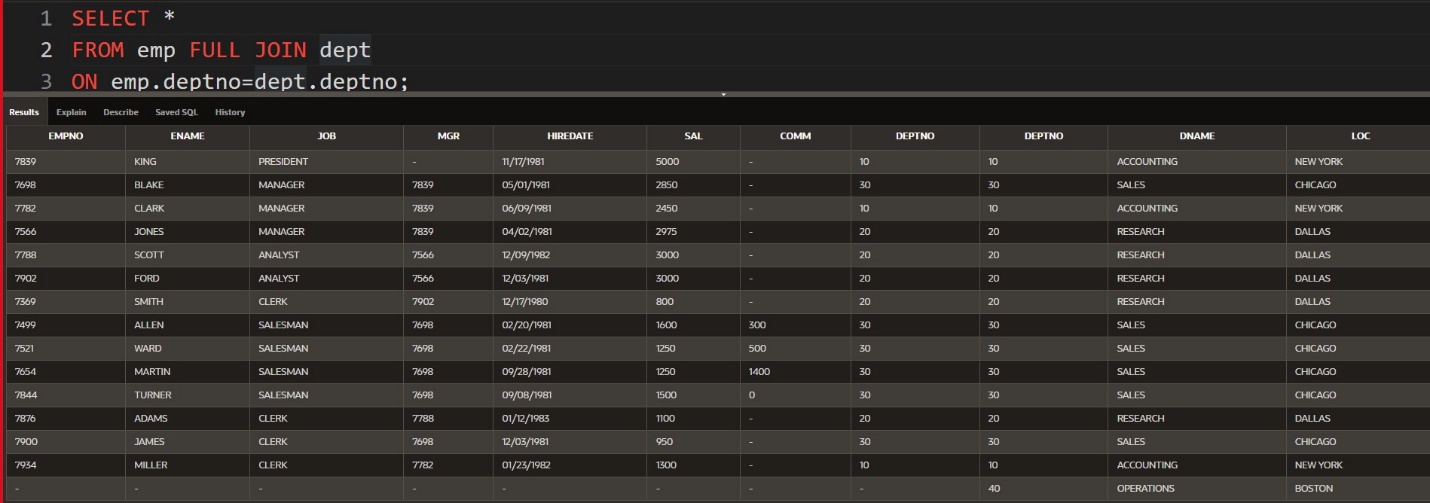
**ORACLE SQL**-in köhnə vaxtlarında **RIGHT JOIN** operatoru olmadığından bu əməliyyatı başqa cür sintaksis ilə həyata keçirdirdilər və bu əsasən **WHERE** clause-ının hesabına baş verirdi, eynən aşağıdakı şəkildə olduğu kimi.

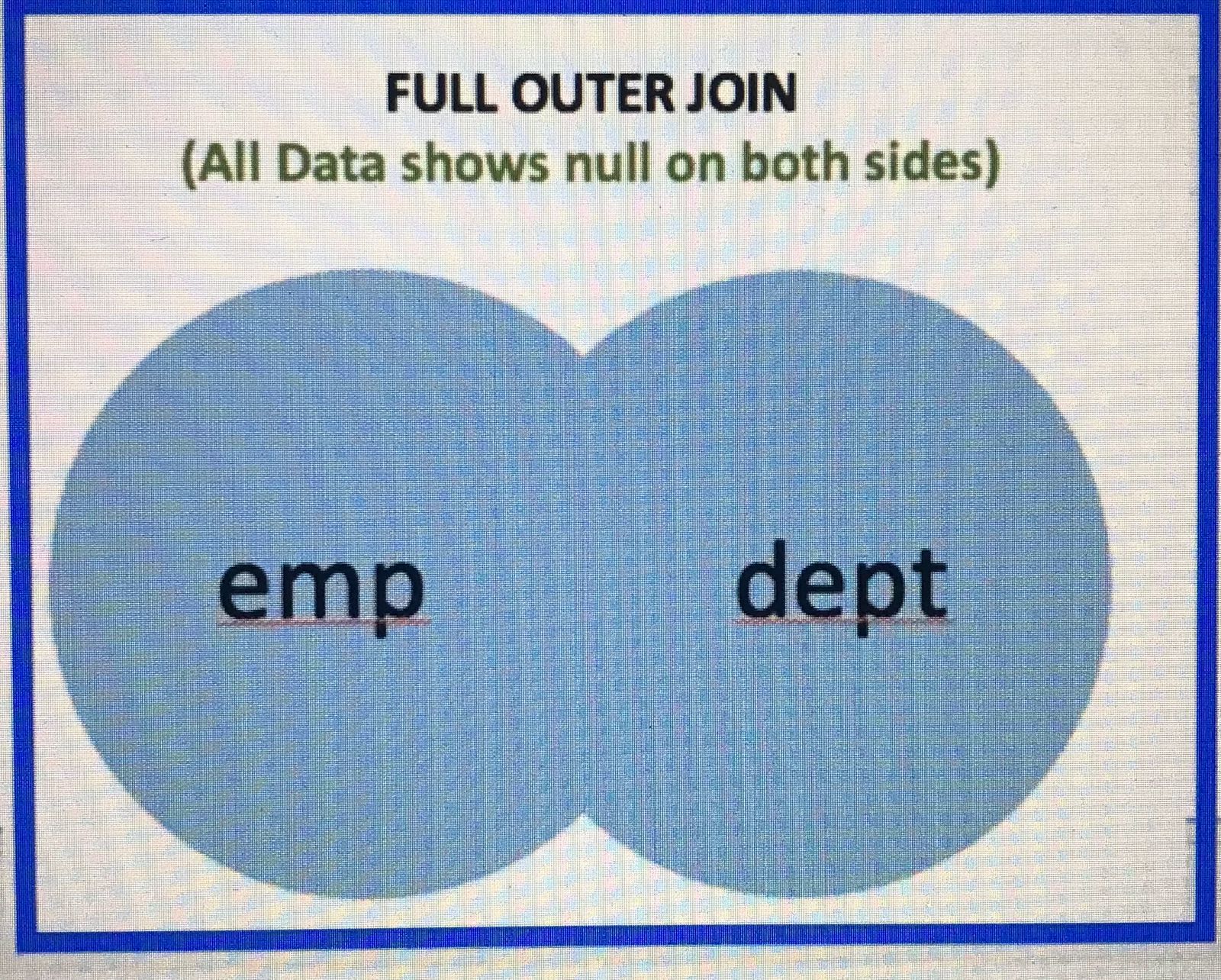
gördüyünüz kimi, yuxarıdakı şəkildə olan query də biz dolayı yol ilə **emp** table-nı **dept** table-na **RIGHT JOIN** etmiş olduq. Əgər **(+)** operatoru sol tərəfə yazılıbsa bu **RIGHT JOIN** adlanır və həmçinin burada **dept** sağ tərəfə düşməsini **WHERE** hissəsində emp.deptno hissəsinə **(+)** operatorunu əlavə etmişik və bu adlanır **RIGHT OUTER JOIN**.

1. Həmçinin biz **LEFT JOIN** istifadə etmədən belə dolayı yol ilə bunu edə bilərik, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi, yuxarıdakı şəkildə olan query də biz dolayı yol ilə **emp** table-na **LEFT OUTER JOIN** etmiş olduq.

**FULL JOIN**

1. Bu **JOIN** növü isə **LEFT** və **RIGHT** join-lərinin qarışığıdır. Yəni join olunacaq table-lar arasında fərqi yoxdur hər hansı birində olan sətir digərində olmasın, elə hər iki tərəfdəndə dəxlisiz şəkildə recordları çəkəcəkdir, eynən aşağıdakı şəkildə olduğu kimi.

gördüyünüz kimi, yuxarıdakı şəkildə olan query də **FULL** **JOIN** istifadə olunaraq hər iki tərəfdəndə recordlar çəkildi.