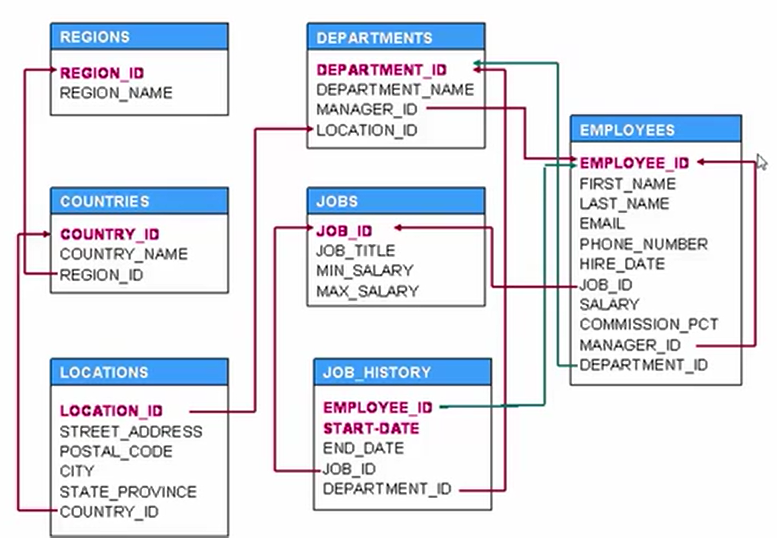


# 1-qadam

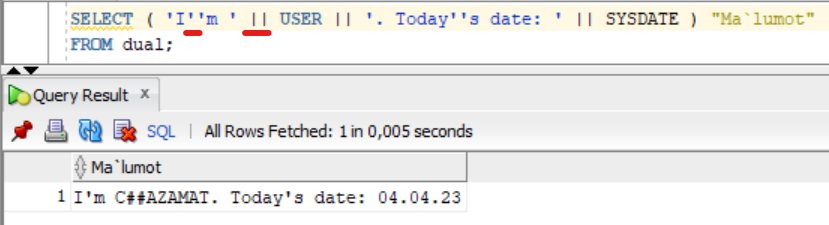
## Ma’lumotlar modeli



Text chiqarish:

{ ' } bu belgi uchun 2ta qo’shtirnoq { '' } yonma yon yoziladi.

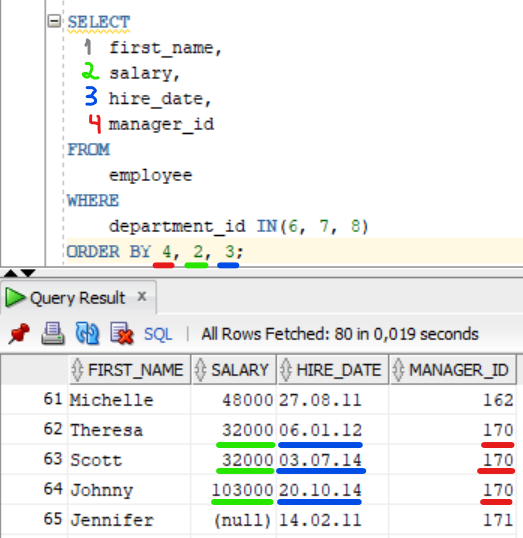
{ **||** } bu belgi ketma-ketliklarni qo’shib chiqar ma’nosini anglatadi.



## ORDER BY bandi

Ustunlarni tartib nomeri bo’yicha saralash:

( Ketma-ketligi: 4-ustun, 2-ustun va 3-ustun bo’yicha tartiblanadi )



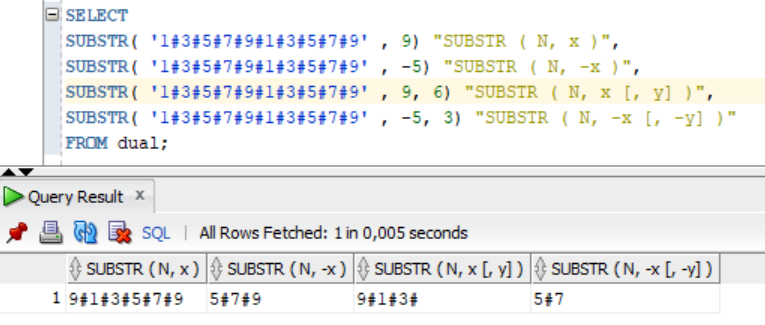
## Substr funksiyasi

SUBSTR ( N, x ) 🡪 N ta belgili matndan x-belgidan boshlab yozadi.

SUBSTR ( N, -x ) 🡪 N ta belgili matndan oxiridan boshiga sanalib, x-belgidan boshlab yozadi.

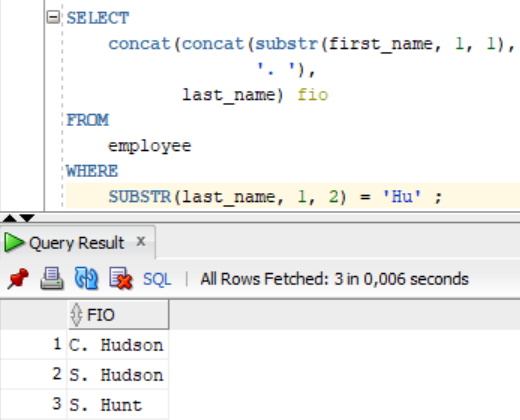
SUBSTR ( N, x [, y] ) 🡪 x-belgidan boshlab, y ta belgi yozadi.

SUBSTR ( N, -x [, y] ) 🡪 oxiridan boshiga sanalib, x-belgidan boshlab, y ta belgi yozadi.



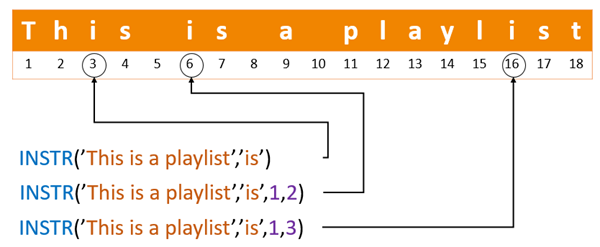
Ism familiyani qisqartirish:

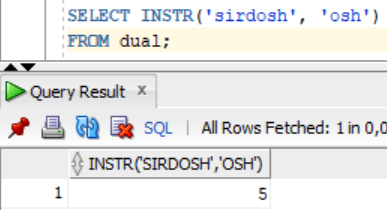
*( SUBSTR*ni *WHERE* bandida ham ishlatsa bo’ladi *)*

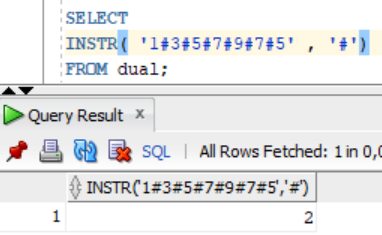
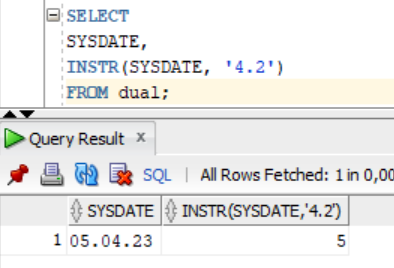


## INSTR()

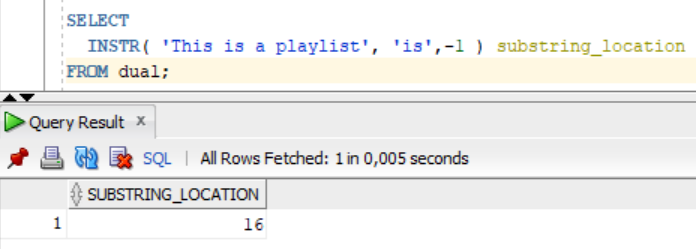
INSTR ⬄ berilgan matndan belgilangan belgilarni tartib nomerini ko’rsatadi.



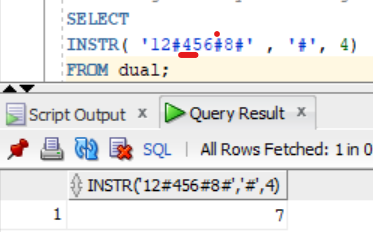




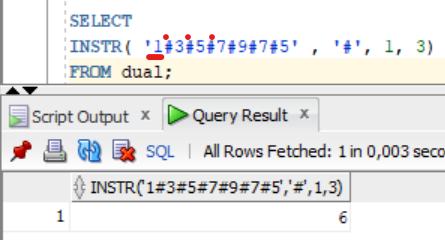
Oxiridan boshiga qarab birinchi takrorlanishni qidirish:



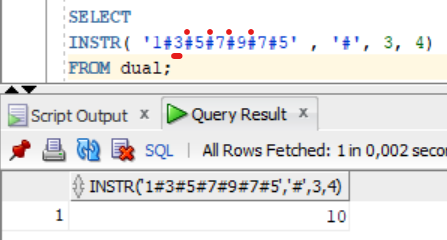
🡻 4- belgidan keyin keladigan # ni tartib nomerini topish:



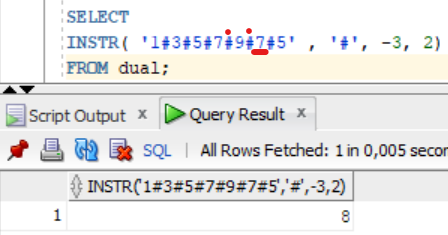
1-belgidan keyin keladigan 3- marta takrorlanadigan # ni tartib nomerini topish:



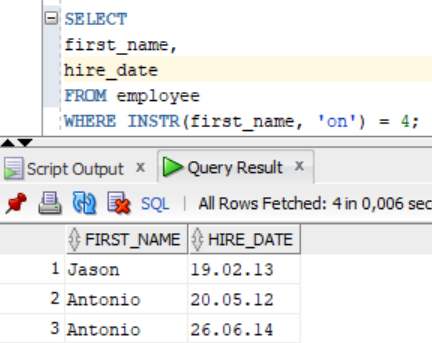
3-belgidan keyin keladigan 4-marta takrorlanadigan # ni tartib nomerini topish:



(-3)-belgidan boshlab, boshiga qarab keladigan 2-marta takrorlanadigan # ni tartib nomerini topish:

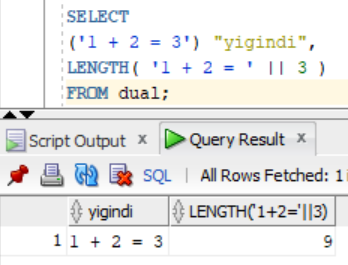
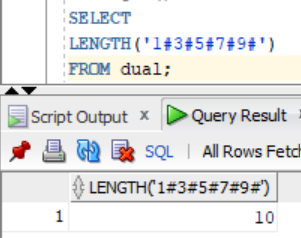


*INSTR* ni *WHERE* bandida ishlatilishi:



## LENGTH() funksiyasi

Belgilar sonini aniqlash:



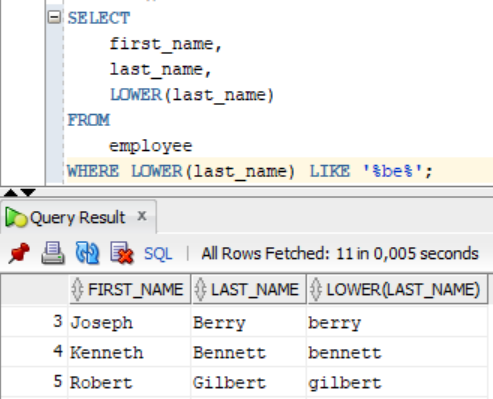
Where bandida ishlatilishi:



## LOWER() Funksiyasi

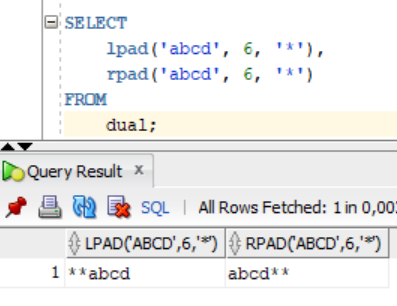
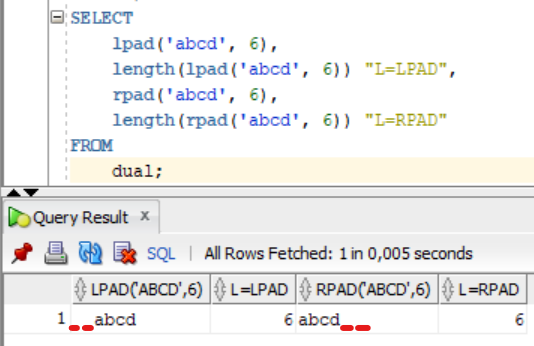
Lower 🡪 kichik harflarga o’tkazadi.

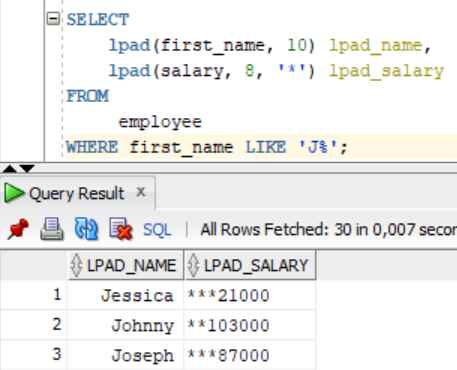
Where bandida ishlatilishi:



## LPAD, RPAD

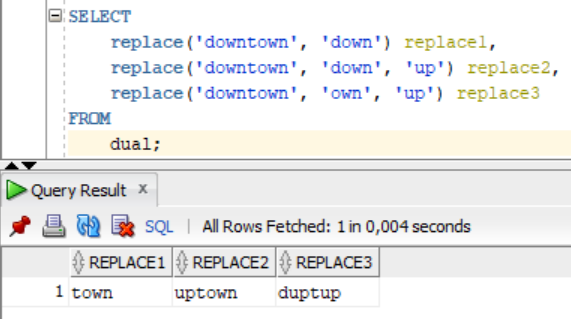
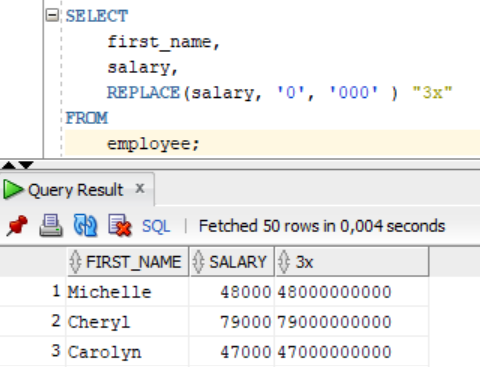
Lpad(c1, x1 [, y1])



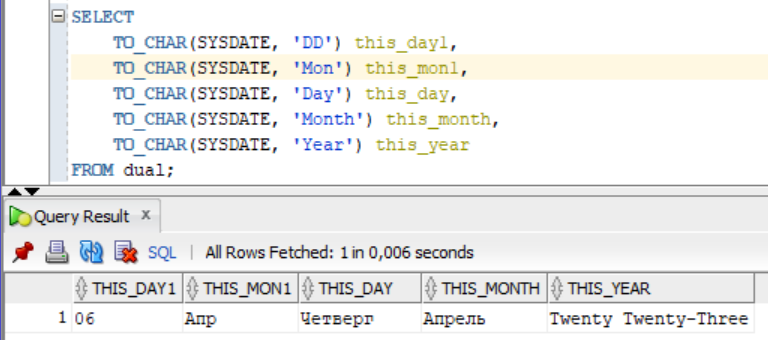


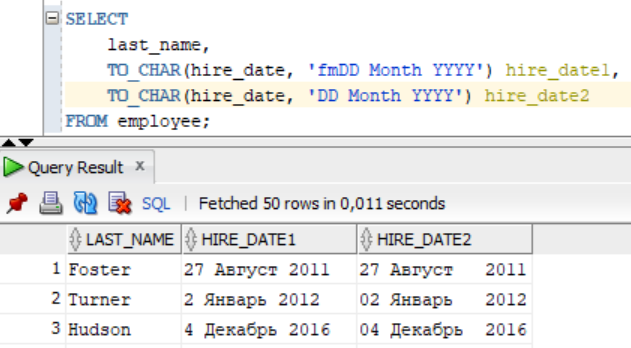
## REPLACE

REPLACE(c1, c2 [, c3]) 🡪 satrdagi belgilangan pastki qatorning barcha takrorlanishini boshqasiga almashtiradi.

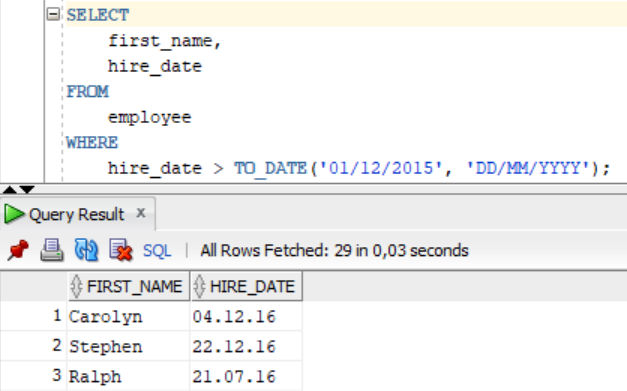
 

## TO\_CHAR

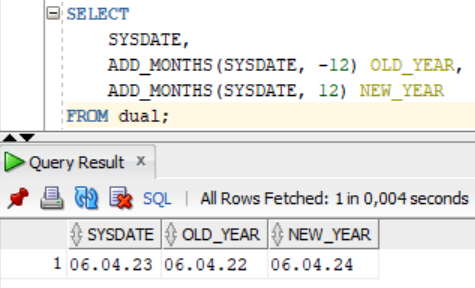




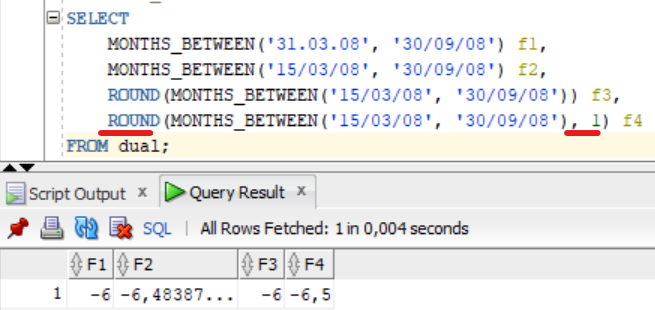
## TO\_DATE



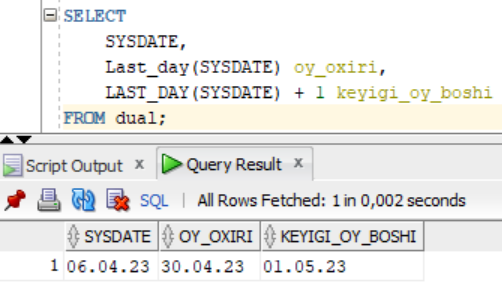
## ADD\_MONTHS



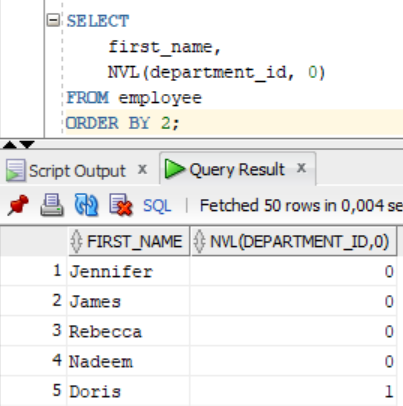
## MONTHS\_BETWEEN

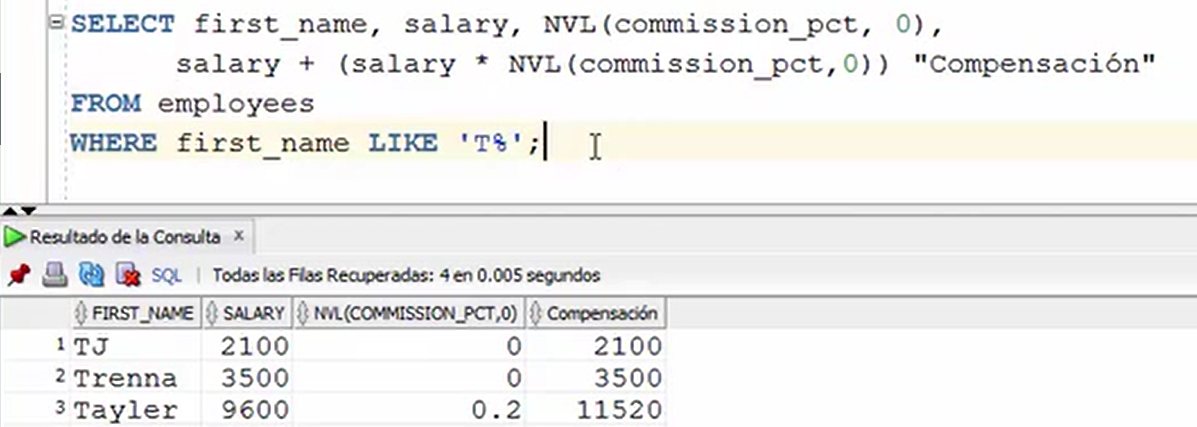


## LAST\_DAY

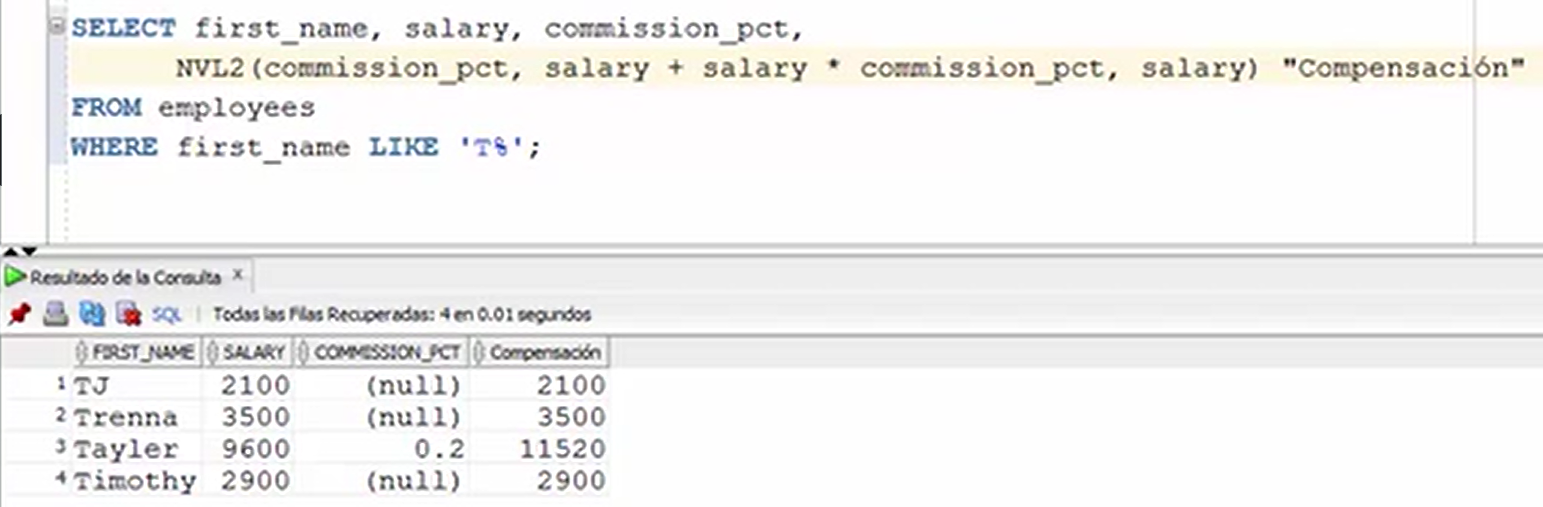


## NVL

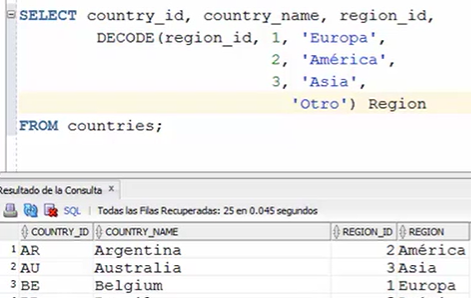




## NVL2

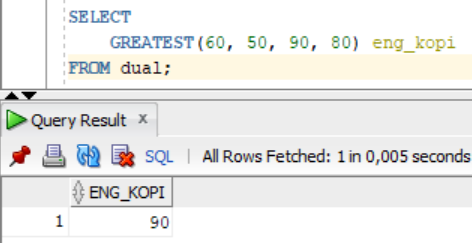


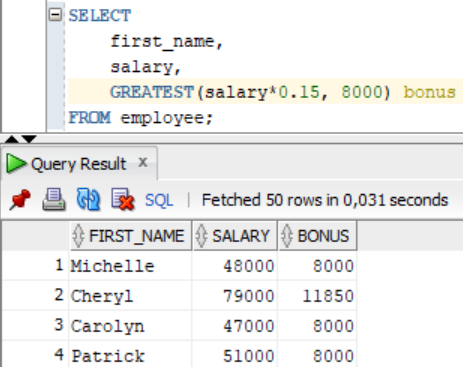
## DECODE()



## GREATEST()

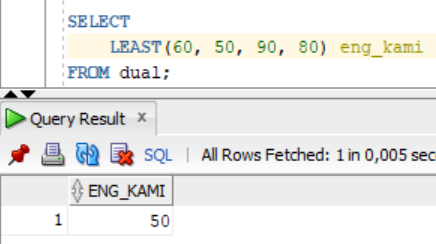
Eng kattasini chiqaradi:

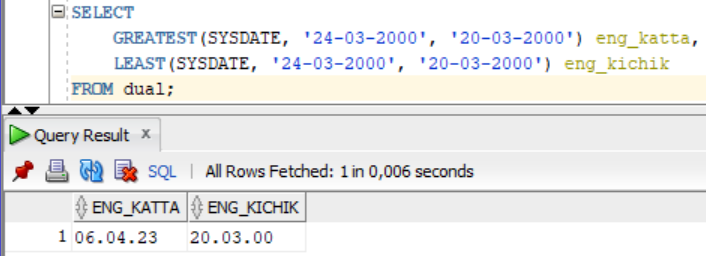




## LEAST()

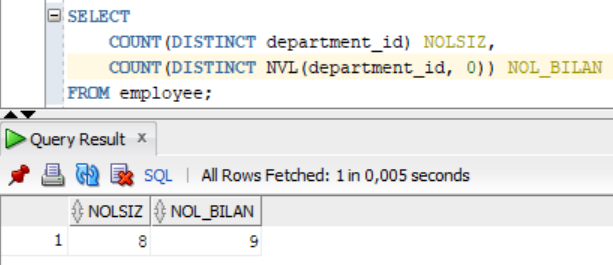
Eng kamini chiqaradi:



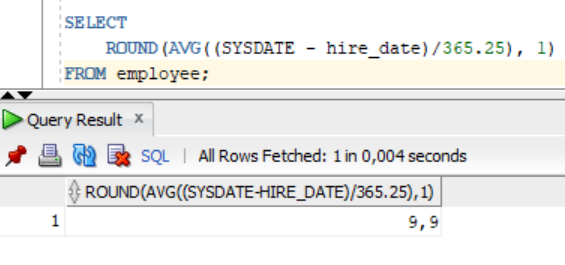


# Group By

## COUNT()

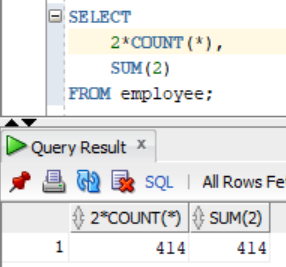


## AVG()

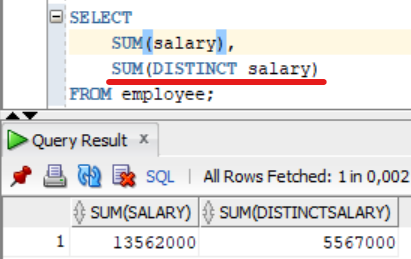


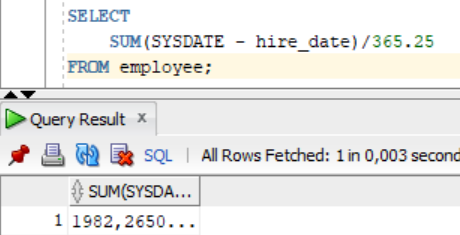
## SUM()

Ikkovi sinonim:

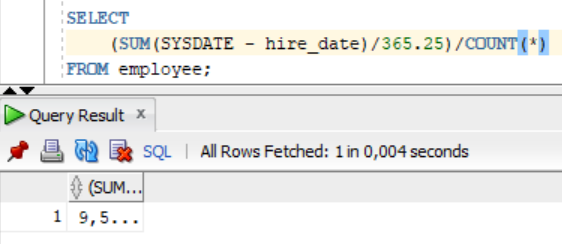


Sum( Distinct salary )



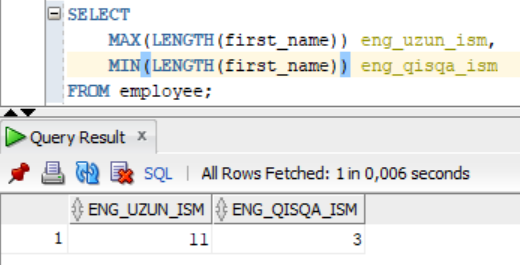


Hodimlarni o’rtacha ishlash yili:



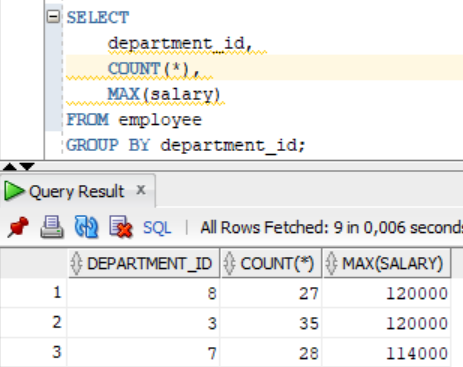
## MIN(), MAX()

Max va Min uzunlikdagi ismlarni topish:

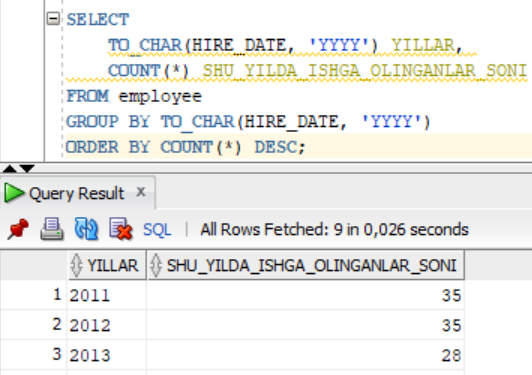


## GROUP BY

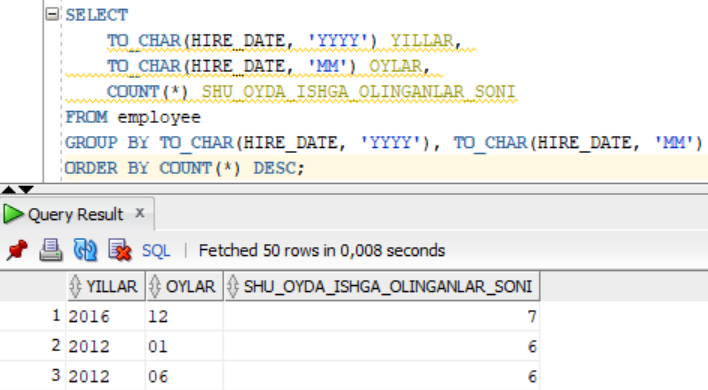
Har bir departament bo’yicha maksimal *salary*ni chiqarish:



Yil bo’yicha ishga olinganlar sonini topish:

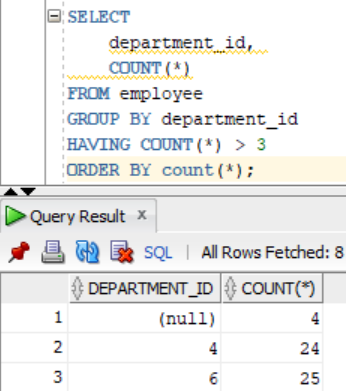


Oy bo’yicha ishga olinganlar sonini topish:

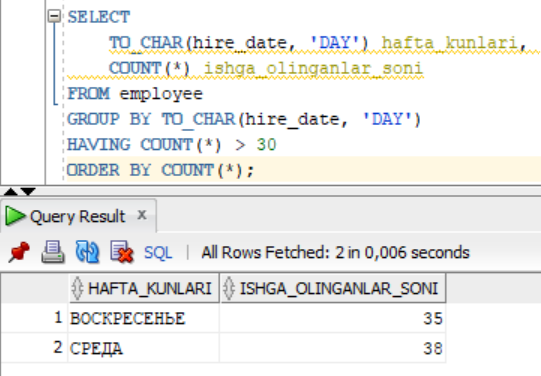


## HAVING bandi

3tadan ko’p hodimi bo’lgan Department\_id ga mansub hodimlar sonini chiqarish:



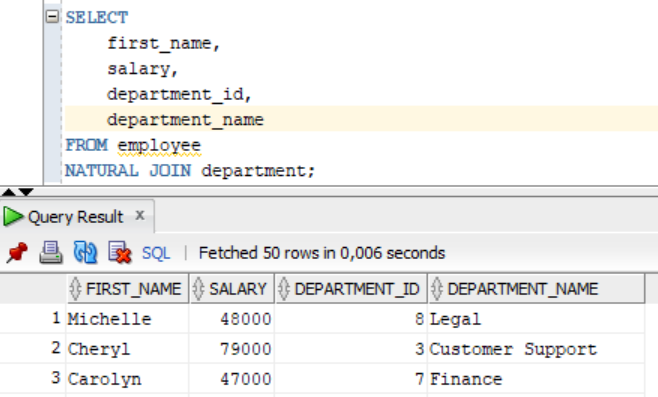
Soni 30 dan oshiq bo’lgan hafta kunlari bo’yicha hodimlar sonini chiqarish:



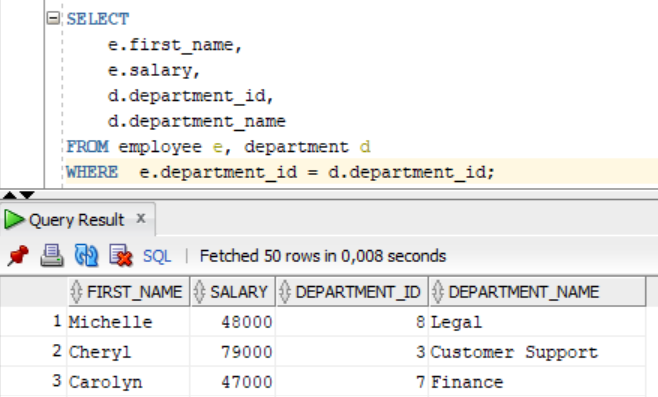
# JOIN

## Natural JOIN

2ta jadvalda bir xil nomli ustun bo’lsa ishlatiladi. Lekin tavsiya etilmaydi. Sababi 2ta jadvalda 2 va undan ortiq bir xil nomli ustun bo’lishi mumkin. Bunday holda aniqlik shart. Misol uchun: employee va customer jadvallarining 2sida ham first\_name va last\_name bor.



Bunga sinonim:

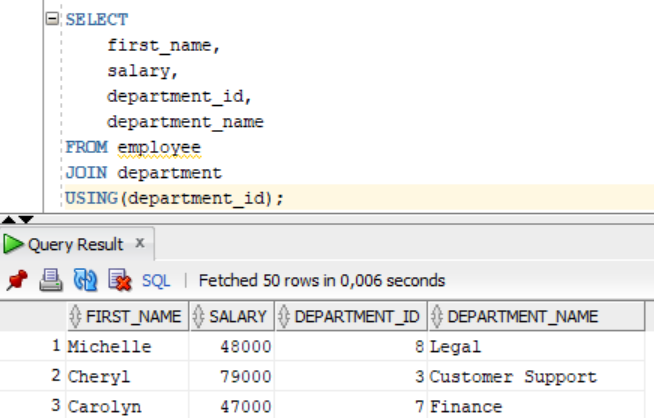


## JOIN

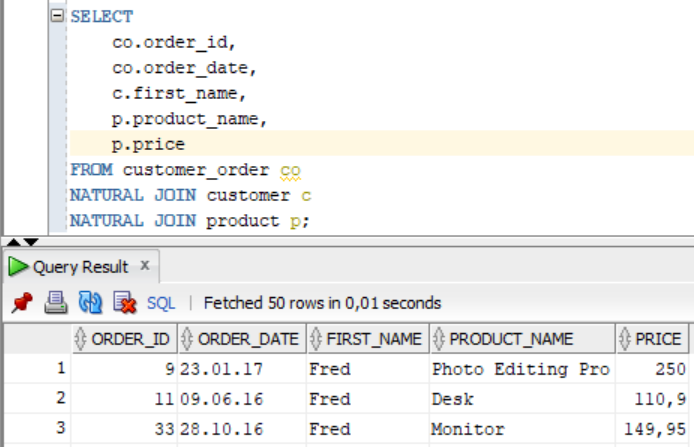
USING ⬄ WHERE e.department\_id = d.department\_id;

USING o’xshash ustunlar nomini kiritish uchun ishlatiladi.

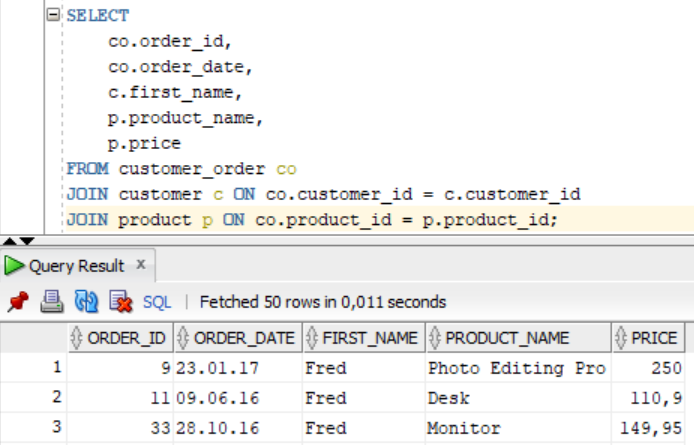
USING ishlatilganda jadvallarga taxallus qo’yilsa xatolik chiqaradi.



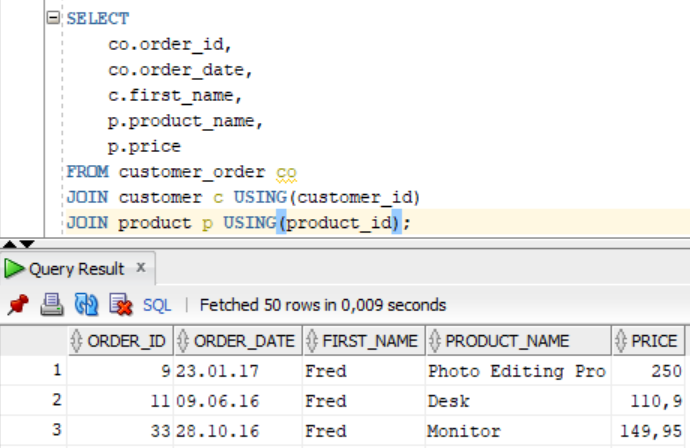
## JOINing Multiple Tables



Bunga sinonim:

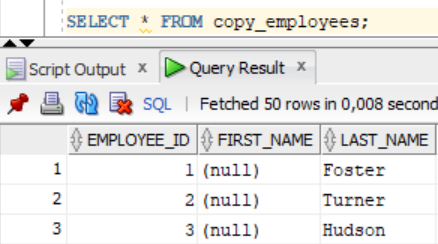
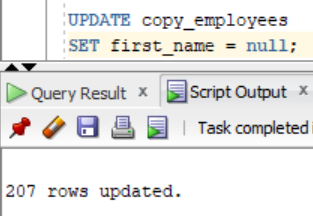


Yana bitta sinonim:

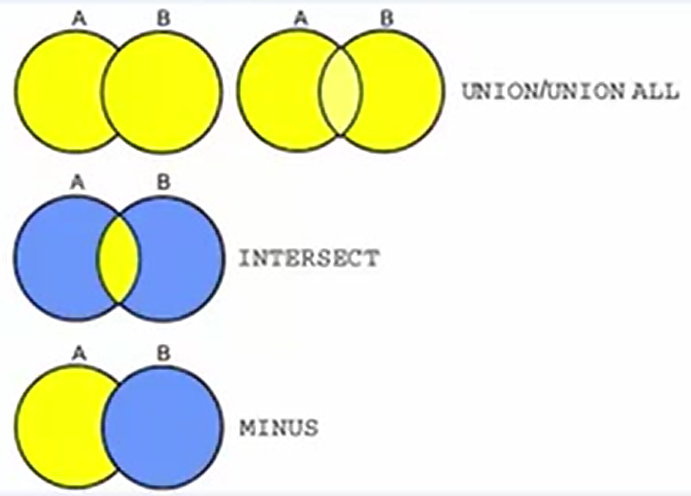


# Insert, Update, Delete

## DML Sentence

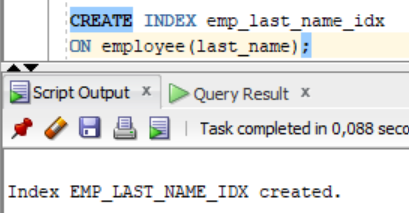


## Operator



## Index

Indeks qatorlarni qidirishni tezlashtiradi.



## Sequence

Sequence ⬄ Auto\_increment

*Sequence*lar jadvallardan ajratilgan holda ishlaydi.

Bir nechta jadval uchun bitta ketma-ketlikdan foydalanishimiz mumkin.

CREATE SEQUENCE sequence\_name

INCREMENT BY interval

START WITH first\_number

MINVALUE min\_value | NOMINVALUE

MAXVALUE max\_value | NOMAXVALUE

CYCLE | NOCYCLE

CACHE cache\_value | NOCACHE

ORDER | NOORDER;

Misol: 50 dan boshlanib, 25ga ortib boruvchi ketma-ketlik. Maksimal qiymati 100. CYCLE bo’lgani uchun 100 dan keyingi qiymat yana boshidagi qiymatga ya’ni 50ga qaytadi.

CREATE SEQUENCE id\_seq

INCREMENT BY 25

START WITH 50

MINVALUE 50

MAXVALUE 100

CYCLE

CACHE 2;

Ketma-ketlikning keyingi qiymatini olish:

SELECT id\_seq.NEXTVAL

FROM dual;

Ketma-ketlikning joriy qiymatini olish:

SELECT id\_seq.CURRVAL

FROM dual;

Drop sequence:

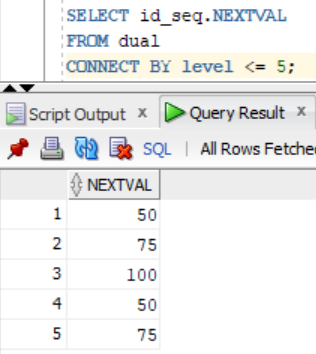
DROP SEQUENCE sequence\_name;

Ushbu SELECT bayonot id\_seq.NEXTVAL qiymatni qayta-qayta ishlatadi:

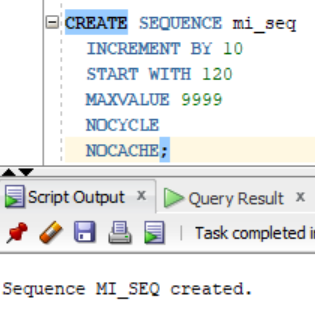
SELECT id\_seq.NEXTVAL

FROM dual

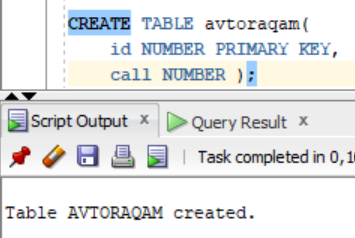
CONNECT BY level <= 5;



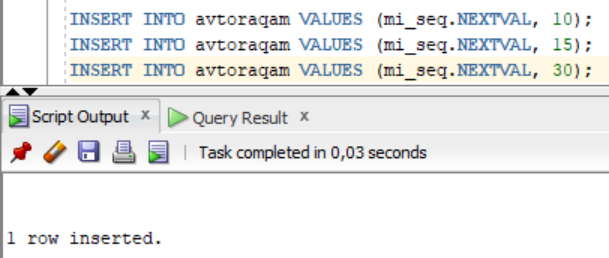
Sequence yaratildi:



Jadval yaratildi:



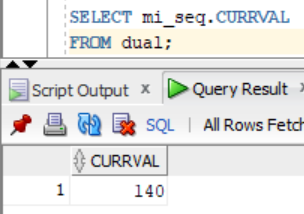
Jadvalga ma’lumotlar kiritildi:



Sequenceni tekshiramiz:



Hozirgi qiymatni aniqlaymiz:



Eskemalar