ORACLE®

Instructor:

Ben Brumm

IT Software Consultant

Owner:

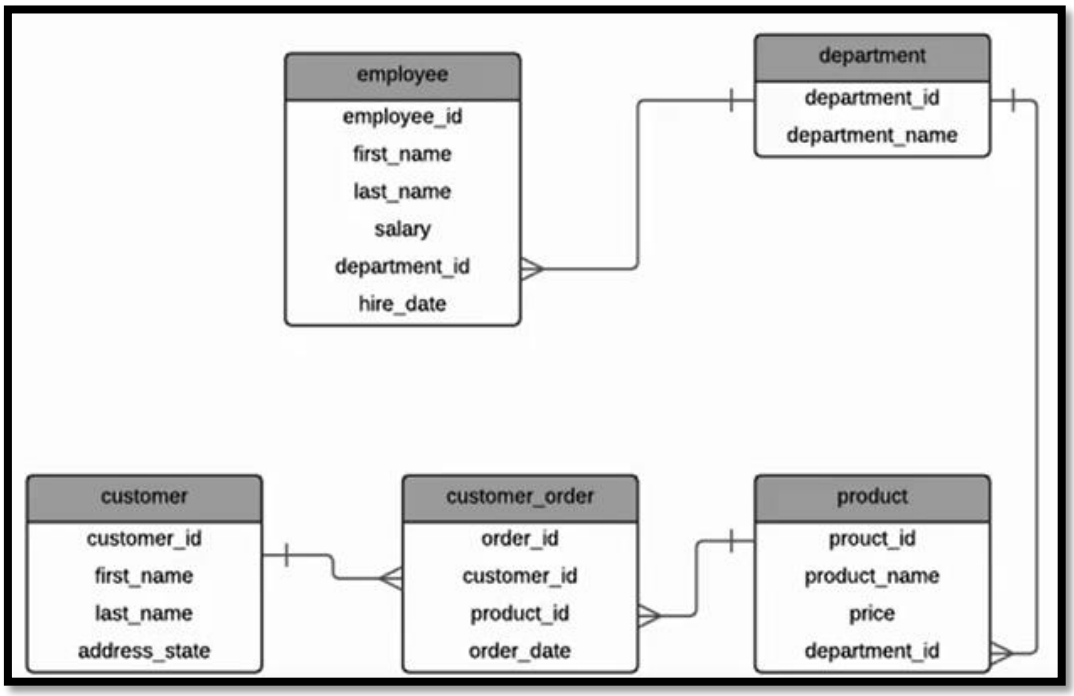
Azamat Nishonov

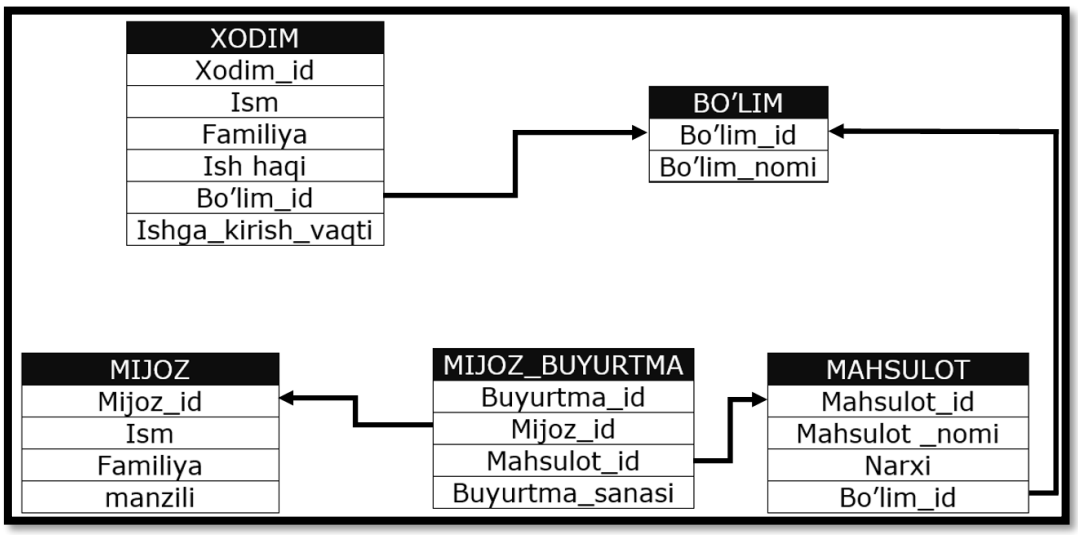
SQL Developer

Content

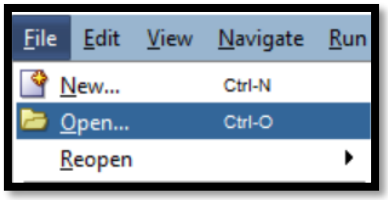
Sample Data

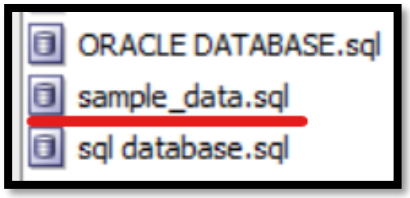
Ma’lumotlar bazasi arxitekturasi



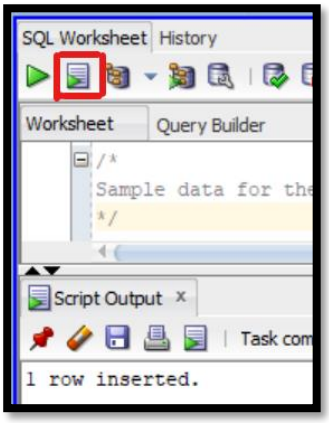


1. Ctrl + O (Open file)





2. Run Script [F5] ni bosamiz va baza o’rnatiladi.



CREATE A NEW USER

Sintaksis:

CREATE USER *user_name*
IDENTIFIED BY *password*
[DEFAULT TABLESPACE tablespace]
[QUOTA {size | UNLIMITED} ON tablespace]
[PROFILE profile]
[PASSWORD EXPIRE]
[ACCOUNT {LOCK | UNLOCK}];

Misol:

CREATE USER *Azamat* IDENTIFIED BY *hayot*;

Agar user_nameni qabul qilmasa unda siz faqat C## prefiks bilan foydalanuvchi yaratishingiz mumkin. Ya’ni:

CREATE USER c##*Azamat* IDENTIFIED BY *hayot*;

Boshqa misol:

CREATE USER intro_user IDENTIFIED BY mypassword;

GRANT CONNECT TO intro_user;

GRANT CREATE SESSION, GRANT ANY PRIVILEGE TO intro_user;

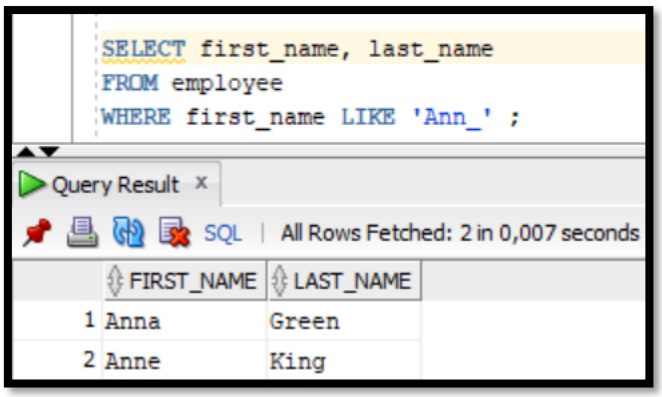
GRANT UNLIMITED TABLESPACE TO intro_user;

GRANT CREATE TABLE TO intro_user;

VIEWING DATA

LIKE

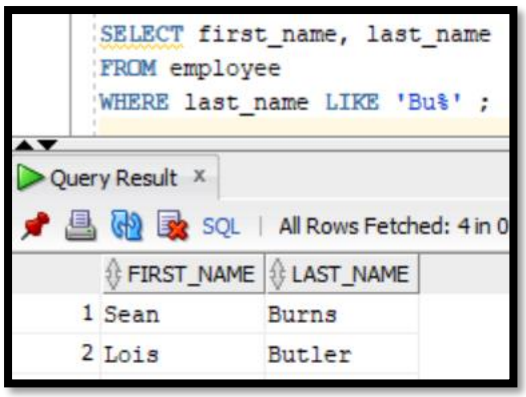
Familiyasi Bu bilan boshlanuvchi shaxslar ma’lumotlari chiqariladi.



```
SELECT first_name, last_name
FROM employee
WHERE first_name LIKE 'Ann_' ;
```

Query Result | SQL | All Rows Fetched: 2 in 0,007 seconds

	FIRST_NAME	LAST_NAME
1	Anna	Green
2	Anne	King

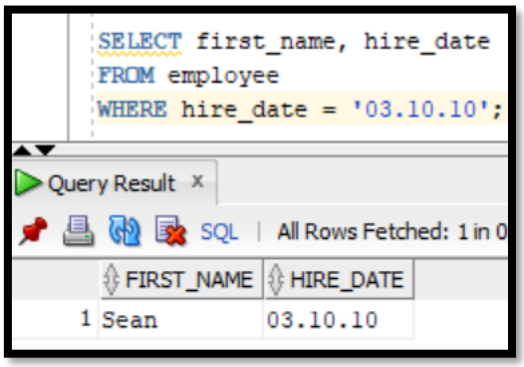


```
SELECT first_name, last_name
FROM employee
WHERE last_name LIKE 'Bu%' ;
```

Query Result | SQL | All Rows Fetched: 4 in 0

	FIRST_NAME	LAST_NAME
1	Sean	Burns
2	Lois	Butler

Sanalar bilan ishlash

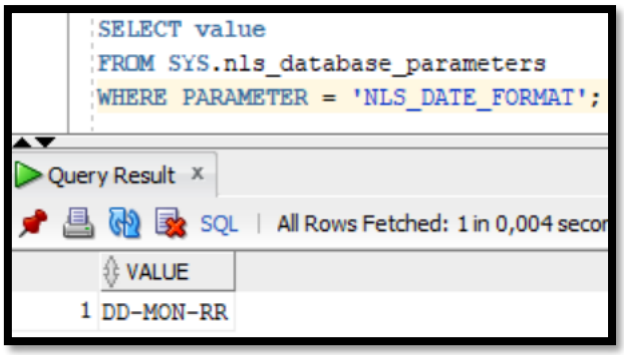


```
SELECT first_name, hire_date
FROM employee
WHERE hire_date = '03.10.10';
```

Query Result | SQL | All Rows Fetched: 1 in 0

	FIRST_NAME	HIRE_DATE
1	Sean	03.10.10

Ma’lumotlar formatini aniqlash



```
SELECT value
FROM SYS.nls_database_parameters
WHERE PARAMETER = 'NLS_DATE_FORMAT';
```

Query Result | SQL | All Rows Fetched: 1 in 0,004 secor

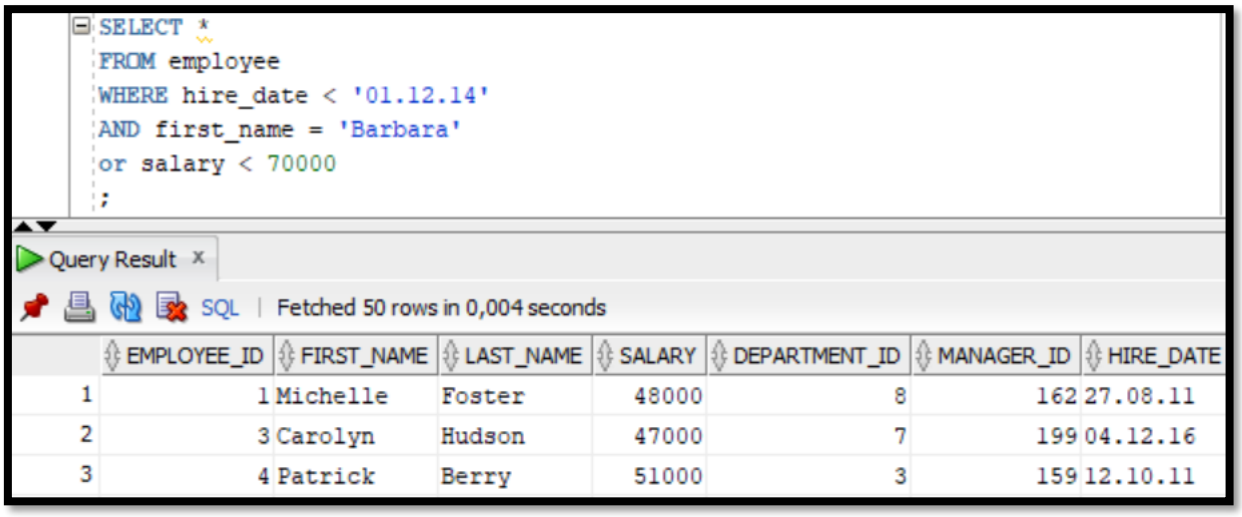
	VALUE
1	DD-MON-RR

AND, OR, NOT

1-holat. Bu yerda ikkovi alohida bajarilib yig’iladi.

1-shart. hire_date < '01.12.14' AND first_name = 'Barbara'

2-shart. salary < 70000



```
SELECT *
FROM employee
WHERE hire_date < '01.12.14'
AND first_name = 'Barbara'
or salary < 70000
;
```

Query Result

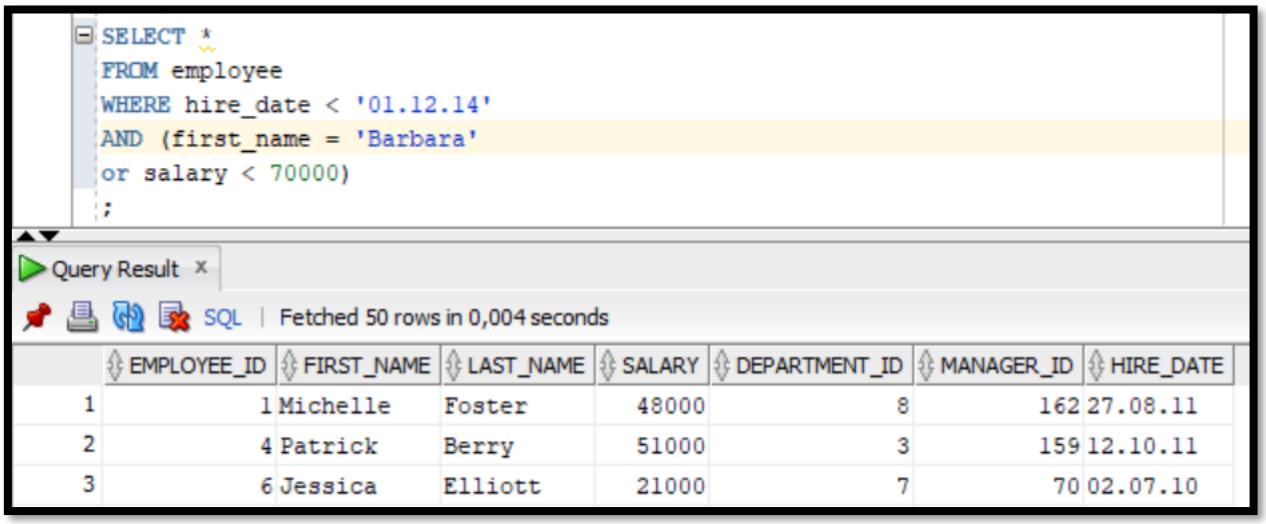
SQL | Fetched 50 rows in 0,004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	MANAGER_ID	HIRE_DATE
1	1	Michelle	Foster	48000	8	162	27.08.11
2	3	Carolyn	Hudson	47000	7	199	04.12.16
3	4	Patrick	Berry	51000	3	159	12.10.11

2-holat. Bu yerda 1-shart aniq bajariladi, 2-shartdan 2 tadan bittasi alohida alohida tanlanadi.

1-shart. hire_date < '01.12.14' AND

2-shart. first_name = 'Barbara' [yoki] salary < 70000



```
SELECT *
FROM employee
WHERE hire_date < '01.12.14'
AND (first_name = 'Barbara'
or salary < 70000)
;
```

Query Result

SQL | Fetched 50 rows in 0,004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	MANAGER_ID	HIRE_DATE
1	1	Michelle	Foster	48000	8	162	27.08.11
2	4	Patrick	Berry	51000	3	159	12.10.11
3	6	Jessica	Elliott	21000	7	70	02.07.10

NULL

NUL qiymatlarni aniqlash [Ochiq (bo’sh)]

```
SELECT first_name, salary
FROM employee
WHERE salary IS NULL
;
```

Query Result

SQL | All Rows Fetched: 2

	FIRST_NAME	SALARY
1	Anthony	(null)
2	Jennifer	(null)

NULL bo’lmagan qiymatlarni aniqlash

```
SELECT first_name, salary
FROM employee
WHERE salary IS NOT NULL
;
```

Query Result

SQL | Fetched 50 rows in

	FIRST_NAME	SALARY
1	Michelle	48000
2	Cheryl	79000
3	Carolyn	47000

DISTINCT

DISTINCT — farqli qiymatlarni chiqarish uchun ishlatiladi. 2ta bir xil qiymatni faqat bittasini chiqaradi.

1-holat. Ismlarni chiqardi. (200 ta qiymat)

2-holat. Faqat farqli ismlarni chiqardi. (126 ta)

3-holat. Ism va familiyasi bir vaqtda farq qiluvchilarni chiqardi. (200 ta)

```
SELECT first_name
FROM employee;
```

Query Result | SQL | All Rows Fetched: 200

	FIRST_NAME
1	Michelle
2	Cheryl
3	Carolyn

```
SELECT DISTINCT first_name
FROM employee;
```

Query Result | SQL | All Rows Fetched: 126

	FIRST_NAME
1	Michelle
2	Cheryl
3	Carolyn

```
SELECT DISTINCT first_name, last_name
FROM employee;
```

Query Result | SQL | All Rows Fetched: 200 in 0,014 secon

	FIRST_NAME	LAST_NAME
1	Michelle	Foster
2	Cheryl	Turner
3	Carolyn	Hudson

MORE OPERATORS

IN

OR ni o'rnini bosuvchi ***IN***

```
SELECT employee_id, last_name
FROM employee
WHERE last_name = 'Foster'
OR last_name = 'Elliott'
OR last_name = 'Mitchell'
;
```

Query Result | SQL | All Rows Fetched: 6 in 0,

	EMPLOYEE_ID	LAST_NAME
1	1	Foster
2	6	Elliott
3	14	Mitchell
4	123	Elliott
5	138	Foster
6	149	Mitchell

```
SELECT employee_id, last_name
FROM employee
WHERE last_name IN ('Foster', 'Elliott', 'Mitchell')
;
```

Query Result | SQL | All Rows Fetched: 6 in 0,006 seconds

	EMPLOYEE_ID	LAST_NAME
1	1	Foster
2	6	Elliott
3	14	Mitchell
4	123	Elliott
5	138	Foster
6	149	Mitchell

BETWEEN

BETWEEN bu chegaraviy oraliqdagi qiymatlar

```
SELECT first_name, hire_date
FROM employee
WHERE hire_date >= '01.05.2016'
AND hire_date <= '31.05.2016'
;
```

Query Result | SQL | All Rows Fetched: 3 in 0,00

	FIRST_NAME	HIRE_DATE
1	John	25.05.16
2	Jacqueline	17.05.16
3	Daniel	15.05.16

```
SELECT first_name, hire_date
FROM employee
WHERE hire_date BETWEEN '01.05.2016' AND '31.05.2016'
;
```

Query Result | SQL | All Rows Fetched: 3 in 0,005 seconds

	FIRST_NAME	HIRE_DATE
1	John	25.05.16
2	Jacqueline	17.05.16
3	Daniel	15.05.16

ALL keywords

ALL — bu barcha shartni qanoatlantiradigan qiymatlar chiqarilsin.

```
SELECT first_name, salary
FROM employee
WHERE salary > 30000
AND salary > 40000
AND salary > 50000
;
```

Query Result | SQL | Fetched 50 rows in 0

	FIRST_NAME	SAL...
1	Patrick	51000
2	Sean	51000
3	Donna	51000
4	Martha	52000

```
SELECT first_name, salary
FROM employee
WHERE salary > ALL(30000, 40000, 50000)
;
```

Query Result | SQL | Fetched 50 rows in 0,007 seconds

	FIRST_NAME	SAL...
1	Patrick	51000
2	Sean	51000
3	Donna	51000
4	Martha	52000

```
SELECT first_name, salary
FROM employee
WHERE salary >= ALL(30000, 40000, 50000)
;
```

Query Result | SQL | Fetched 50 rows in 0,004 seconds

	FIRST_NAME	SAL...
1	Daniel	50000
2	Justin	50000
3	Patrick	51000
4	Sean	51000

```
SELECT first_name, salary
FROM employee
WHERE salary < ALL(30000, 40000, 50000)
;
```

Query Result | SQL | All Rows Fetched: 22 in 0,005 second

	FIRST_NAME	SAL...
1	Kenneth	29000
2	Phyllis	29000
3	Wanda	28000

```
SELECT first_name, salary
FROM employee
WHERE salary <= ALL(30000, 40000, 50000)
;
```

Query Result | SQL | All Rows Fetched: 23 in 0,006 seconds

	FIRST_NAME	SAL...
1	Joseph	30000
2	Kenneth	29000
3	Phyllis	29000

```
SELECT first_name, salary
FROM employee
WHERE salary <> ALL(30000, 40000, 50000)
;
```

Query Result | SQL | Fetched 50 rows in 0,004 seconds

	FIRST_NAME	SALARY
1	Michelle	48000
2	Cheryl	79000
3	Carolyn	47000

```
SELECT first_name, salary
FROM employee
WHERE salary = ALL(30000, 40000, 50000)
;
```

Query Result | SQL | All Rows Fetched: 0 in 0,005 seconds

FIRST_NA...	SALARY

ANY keywords

ANY – bu birorta shartni qanoatlantiradigan qiymatlar chiqarilsin.

```
SELECT first_name, salary
FROM employee
WHERE salary > 30000
OR salary > 40000
OR salary > 50000
;
```

Query Result | SQL | Fetched 50 rows i

	FIRST_NAME	SAL...
1	Amy	31000
2	Susan	31000
3	Theresa	32000

```
SELECT first_name, salary
FROM employee
WHERE salary > ANY(30000, 40000, 50000)
;
```

Query Result | SQL | Fetched 50 rows in 0,007 seconds

	FIRST_NAME	SAL...
1	Amy	31000
2	Susan	31000
3	Theresa	32000

```
SELECT first_name, salary
FROM employee
WHERE salary = ANY(30000, 40000, 50000)
;
```

Query Result | SQL | All Rows Fetched: 6 in 0,005 seconds

	FIRST_NAME	SALARY
1	Roy	40000
2	Justin	50000
3	Christine	40000
4	Joseph	30000
5	Kelly	40000
6	Daniel	50000

SORTING DATA

ORDER BY — bu tartiblash.

ASC – O'sish tartibi, ***DESC*** – kamayish tartibi.

ORDER BY (1ta ustun uchun)

```
SELECT employee_id, last_name
FROM employee
ORDER BY last_name ASC
;
```

Query Result | SQL | Fetched 50 rows in 0,004 s

	EMPLOYEE_ID	LAST_NAME
1	20	Adams
2	71	Allen
3	40	Alvarez

```
SELECT employee_id, last_name
FROM employee
ORDER BY last_name DESC
;
```

Query Result | SQL | Fetched 50 rows in 0,00

	EMPLOYEE_ID	LAST_NAME
1	102	Woods
2	94	Williams
3	95	Williams

```
SELECT last_name, salary
FROM employee
ORDER BY salary ASC
;
```

Query Result | SQL | Fetched 50 rows

	LAST_NAME	SALARY
1	Elliott	21000
2	Cook	21000
3	Sullivan	21000

```
SELECT last_name, salary
FROM employee
ORDER BY salary DESC
;
```

Query Result | SQL | Fetched 50 rows

	LAST_NAME	SALARY
1	Fisher	(null)
2	Rice	(null)
3	Henderson	120000

ORDER BY (bir nechta ustun uchun)

Last_name bo’yicha kamayuvchi, *First_name* bo’yicha o’suvchi holda tartiblash.

First_name bo’yicha tartiblaydi, qachonki, *Last_name* lari bir xil bo’lganlarini.

```
SELECT employee_id, last_name, first_name
FROM employee
ORDER BY last_name DESC, first_name ASC
;
```

Query Result

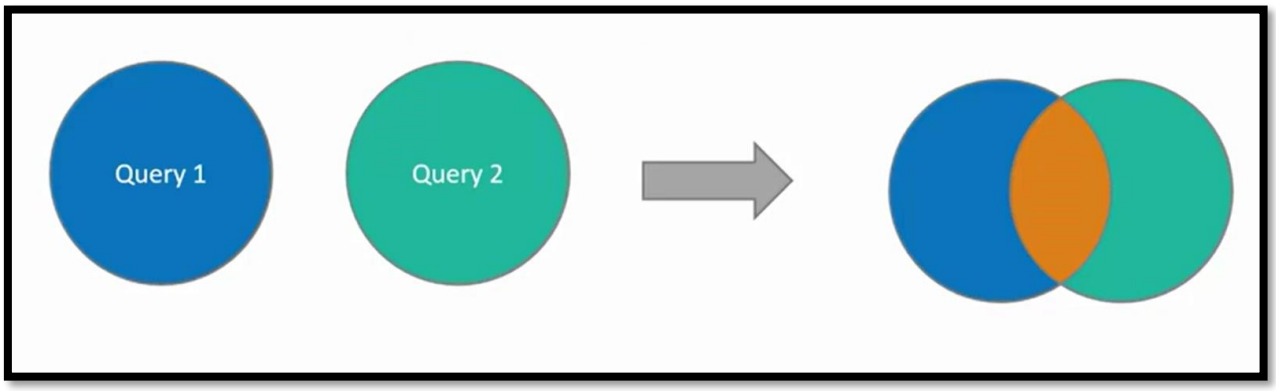
SQL | Fetched 50 rows in 0,004 seconds

	EMPLOYEE_ID	LAST_NAME	FIRST_NAME
1	102	Woods	Ralph
2	61	Williams	Karen
3	94	Williams	Lisa
4	95	Williams	Stephanie

SET OPERATORS

UNION

UNION



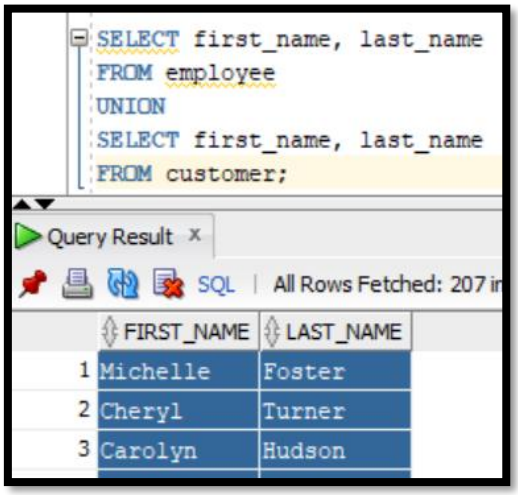
UNION — jadvallarni birlashtirish uchun ishlatiladi.

1-holat. O’xshashlari birlashtirib chiqariladi. (207ta qiymat)

2-holat. Har biri alohida o’z unvoni bilan chiqadi. (210ta qiymat)

3-holat. ***ORDER BY*** bilan tartiblab chiqardik va o'xshashlarini topdik.

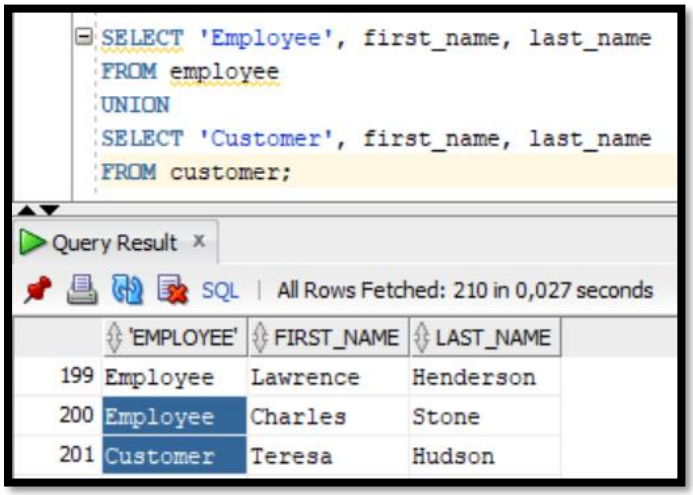
4-holat. ***ORDER BY*** 1, 2 holatida chiqarildi. Oldin mijozlar keyin xodimlar *first_name* bo'yicha tartiblanib chiqarildi. Buni hali batafsil keyinroq o'rganamiz.



```
SELECT first_name, last_name
FROM employee
UNION
SELECT first_name, last_name
FROM customer;
```

Query Result | SQL | All Rows Fetched: 207 in

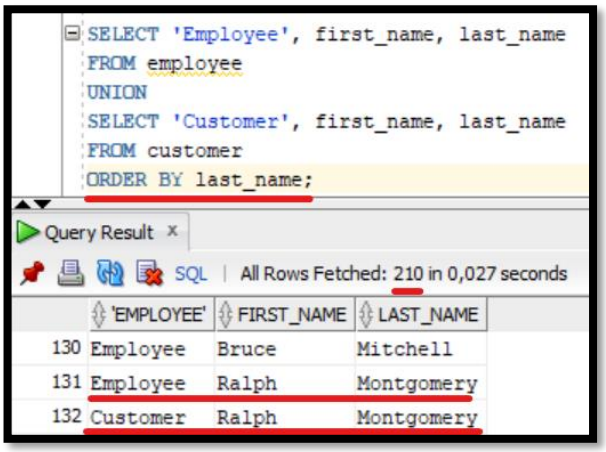
	FIRST_NAME	LAST_NAME
1	Michelle	Foster
2	Cheryl	Turner
3	Carolyn	Hudson



```
SELECT 'Employee', first_name, last_name
FROM employee
UNION
SELECT 'Customer', first_name, last_name
FROM customer;
```

Query Result | SQL | All Rows Fetched: 210 in 0,027 seconds

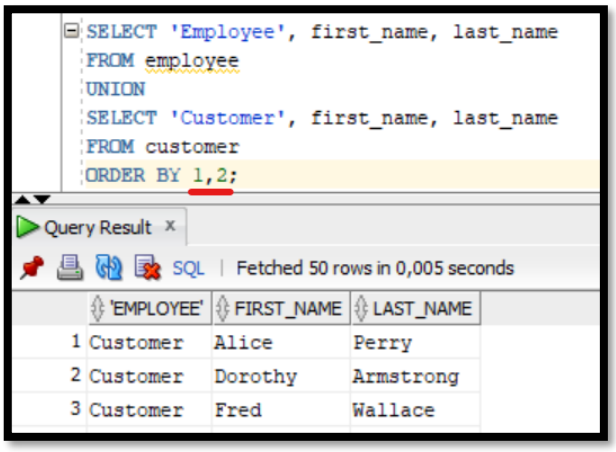
	'EMPLOYEE'	FIRST_NAME	LAST_NAME
199	Employee	Lawrence	Henderson
200	Employee	Charles	Stone
201	Customer	Teresa	Hudson



```
SELECT 'Employee', first_name, last_name
FROM employee
UNION
SELECT 'Customer', first_name, last_name
FROM customer
ORDER BY last_name;
```

Query Result | SQL | All Rows Fetched: 210 in 0,027 seconds

	'EMPLOYEE'	FIRST_NAME	LAST_NAME
130	Employee	Bruce	Mitchell
131	Employee	Ralph	Montgomery
132	Customer	Ralph	Montgomery

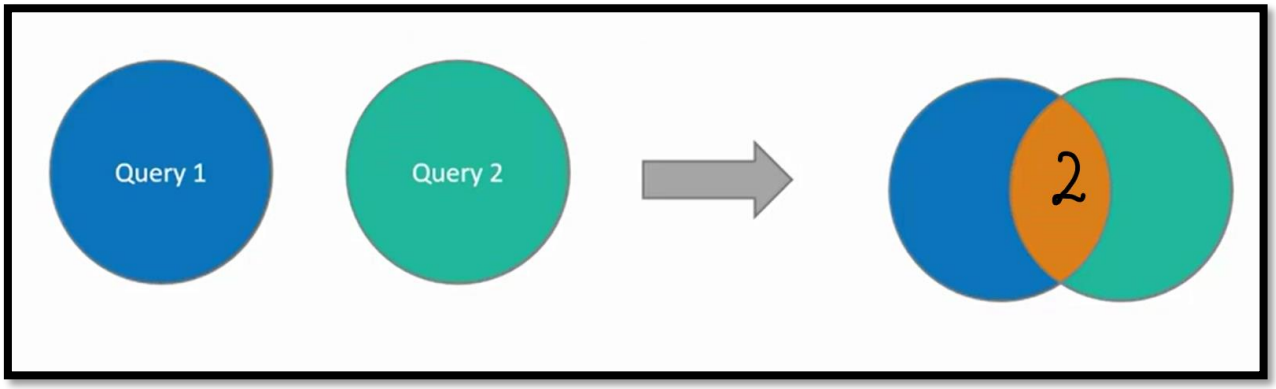


```
SELECT 'Employee', first_name, last_name
FROM employee
UNION
SELECT 'Customer', first_name, last_name
FROM customer
ORDER BY 1,2;
```

Query Result | SQL | Fetched 50 rows in 0,005 seconds

	'EMPLOYEE'	FIRST_NAME	LAST_NAME
1	Customer	Alice	Perry
2	Customer	Dorothy	Armstrong
3	Customer	Fred	Wallace

UNION ALL



UNION ALL — barchasini qo'shib ketma-ket chiqarishdir.

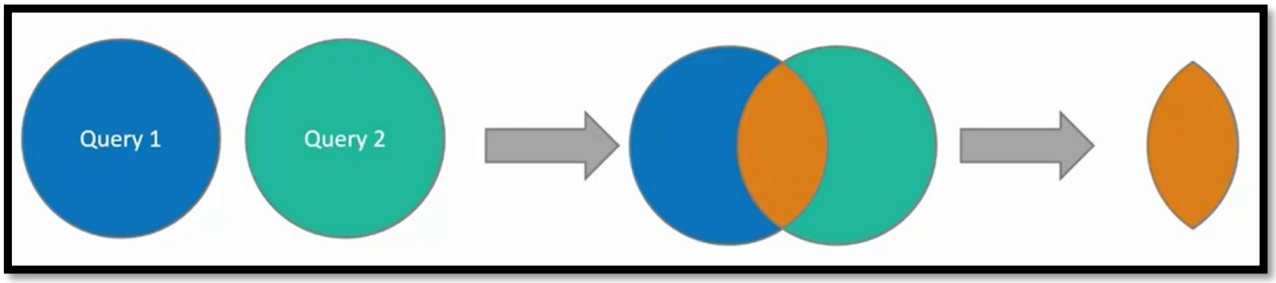
```
SELECT first_name, last_name
FROM employee
UNION ALL
SELECT first_name, last_name
FROM customer
ORDER BY 1,2;
```

Query Result | SQL | All Rows Fetched: 210 in

	FIRST_NAME	LAST_NAME
1	Adam	Stephens
2	Alice	Perry
3	Alice	Perry

INTERSECT

Intersect — bu kesishmani chiqarish. Ya’ni ikki jadvalda ham bir xil takrorlangan qiymatlarni chiqarish.




```
SELECT first_name, last_name
FROM employee
INTERSECT
SELECT first_name, last_name
FROM customer;
```

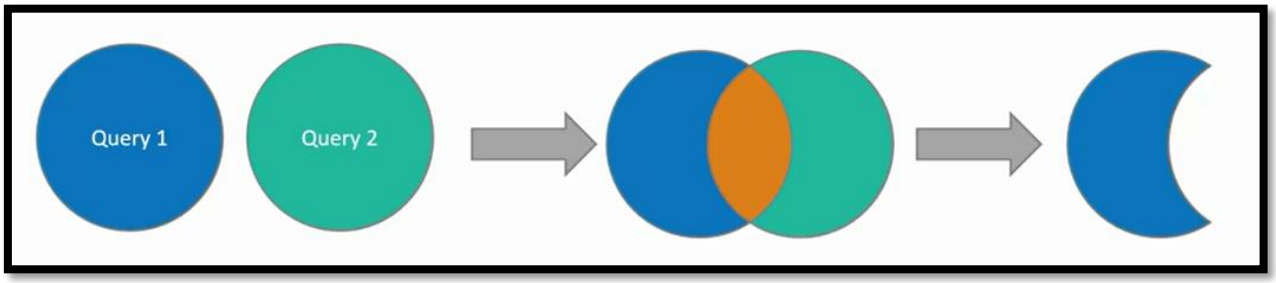
Query Result | SQL | All Rows Fetched: 3 in

	FIRST_NAME	LAST_NAME
1	Alice	Perry
2	Ralph	Montgomery
3	Fred	Wallace

- Query A has records 1, 2, 3, 4
- Query B has records 2, 4, 6, 8, 9
- UNION: 1, 2, 3, 4, 6, 8, 9
- UNION ALL: 1, 2, 2, 3, 4, 4, 6, 8, 9
- INTERSECT: 2, 4

MINUS

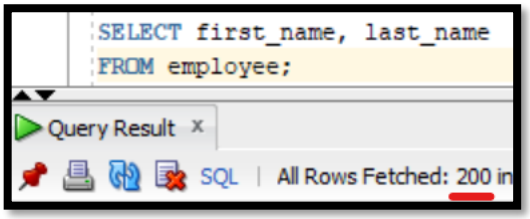
MINUS — bu 1-to'plamdan kesishmani olib tashlab chiqarish.

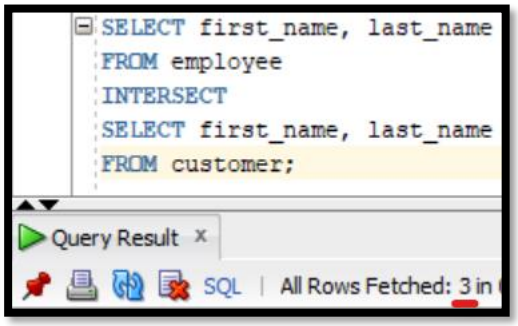


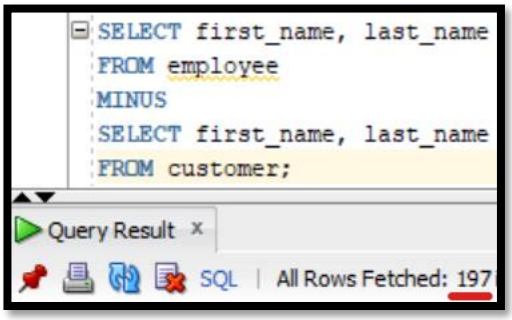
1-holat. Barcha hodimlar. (200 ta)

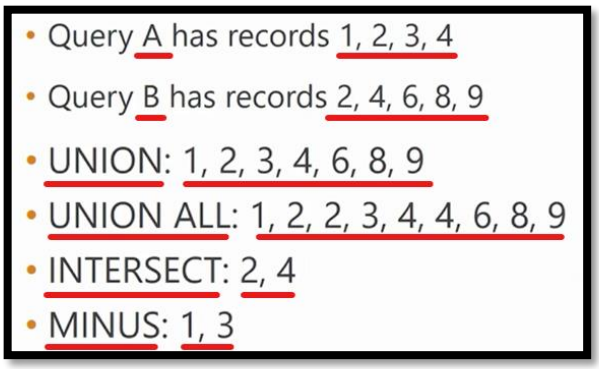
2-holat. Hodimlar va haridorlar kesishmasi. (3 ta)

3-holat. Hodimlardan haridorlar ayirmasi. (200 - 3 = 197 ta)









MORE SETS

SET OPERATOR lar kombinatsiyasi.

1-holat. UNION ([28;40] U [20;30] => [20;40])

2-holat. UNION + INTERSECT ([20;40] () [30;55] => [30;40])

3-holat. UNION + INTERSECT + MINUS (🡻)

([30;40] — [31;33] => [30] U [34;40])

4-holat. INTERSECT + UNION + MINUS

```
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 28000 AND 40000
UNION
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 20000 AND 30000;
```

Query Result — All Rows Fetched: 45 in 0,004 second

	FIRST_NAME	LAST_NAME	SAL...
1	Jessica	Sullivan	21000
2	Jessica	Elliott	21000
3	Dorothy	Cook	21000

```
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 28000 AND 40000
UNION
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 20000 AND 30000
INTERSECT
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 30000 AND 55000
;
```

Query Result — All Rows Fetched: 23 in 0,002 secon

	FIRST_NAME	LAST_NAME	SAL...
1	Joseph	Davis	30000
2	Amy	Wheeler	31000
3	Susan	Henderson	31000

```
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 28000 AND 40000
UNION
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 20000 AND 30000
INTERSECT
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 30000 AND 55000
MINUS
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 31000 AND 33000
;
```

Query Result — All Rows Fetched: 15 in 0,003 seconds

	FIRST_NAME	LAST_NAME	SAL...
1	Joseph	Davis	30000
2	Ann	Bowman	34000
3	Ralph	Anderson	34000

```
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 28000 AND 40000
UNION
(SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 20000 AND 30000
INTERSECT
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 30000 AND 55000)
MINUS
SELECT first_name, last_name, salary
FROM employee
WHERE salary BETWEEN 31000 AND 33000
;
```

Query Result — All Rows Fetched: 18 in 0,006 second

	FIRST_NAME	LAST_NAME	SAL...
1	Wanda	Stewart	28000
2	Phyllis	Larson	29000
3	Kenneth	Little	29000

AGGREGATE FUNCTIONS AND GROUPING

COUNT()

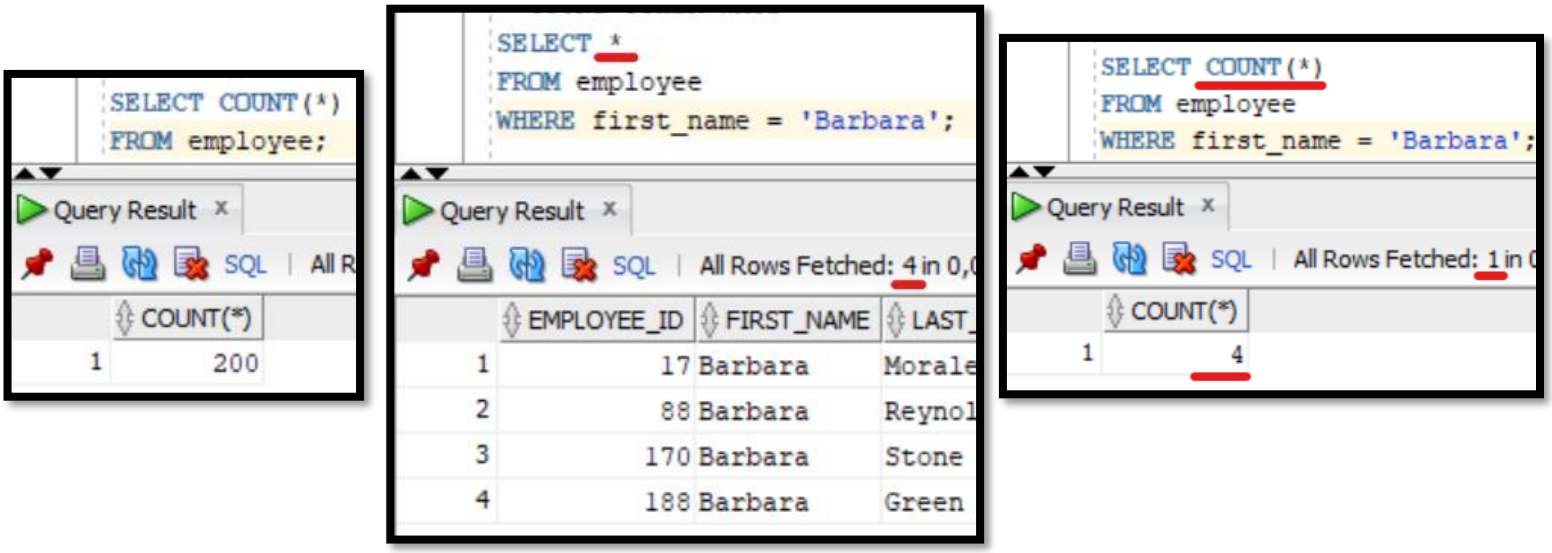
COUNT() — qiymatlar sonini chiqaradi.

COUNT(*) — satrlar sonini chiqaradi.

1-holat. Xodimlar ma’lumotlarini chiqaradi.

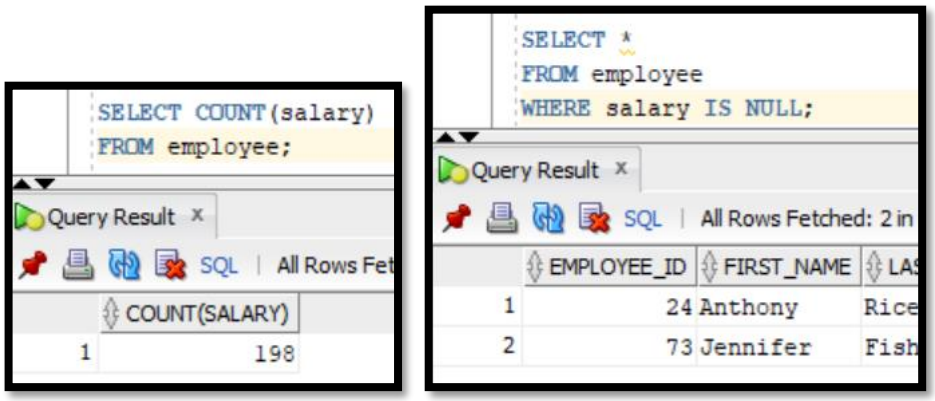
2-holat. Barbara ismli xodimlarni ma’lumotlarini chiqaradi.

3-holat. Barbara ismli xodimlar sonini chiqaradi.



4-holat. Ish haqi oladiganlar soni (198 ta qiymat)

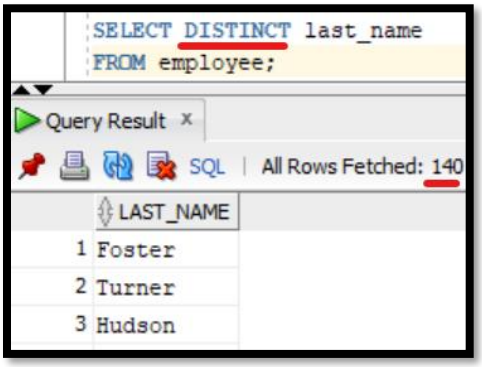
5-holat. Ish haqi olmaydiganlar ma’lumotlari (2 ta ***NULL*** katak)

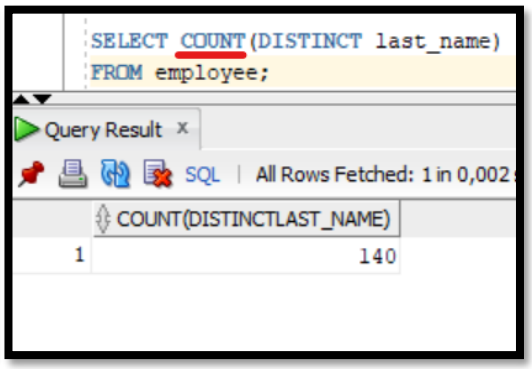


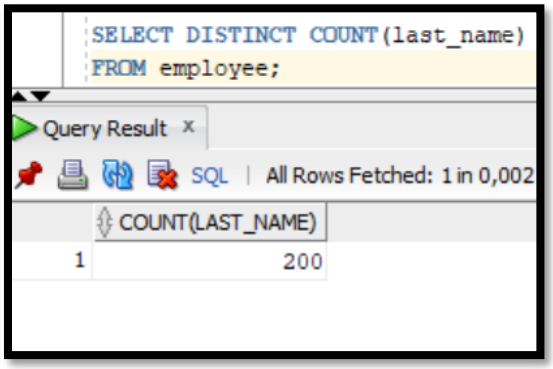
6-holat. Faqrli ismlar chiqarilsin.

7-holat. Farqli ismlar soni chiqarilsin.

8-holat. Qatorlar sonini hisoblandi, va hisobni har xilligi tekshirildi







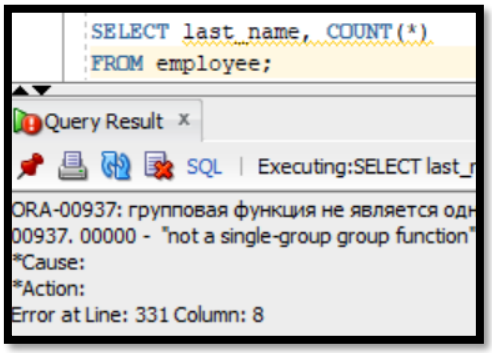
Count data within GROUP BY

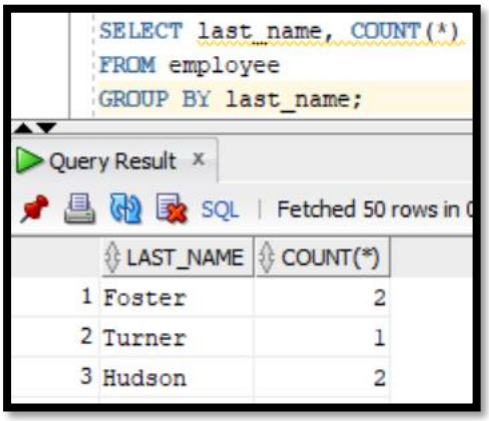
GROUP BY — bu guruhlash

Shart. Familiyalar chiqarilsin, ularning takrorlanish sonlari chiqarilsin.

1-holat. Ishlamaydi. Chunki *last_name*da qator ko'p, *count*da 1ta.

2-holat. Ishlaydi.

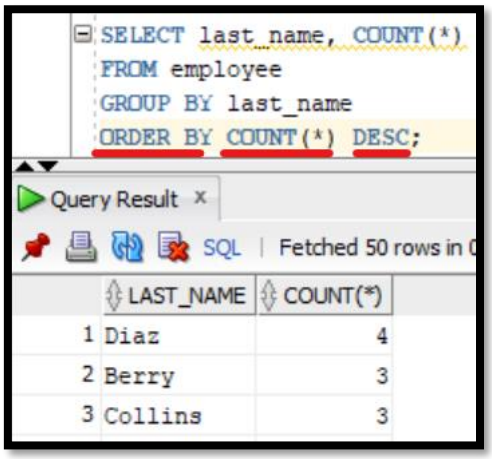


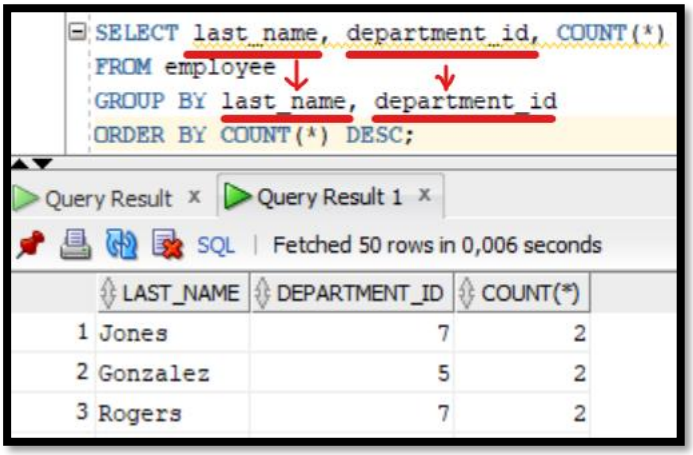


GROUP BY with ORDER BY

1-holat. Familiyalar takrorlanishlar soni kamayuvchi tartibi.

2-holat. *Last_name* va *department_id*lari aynan bir xil bo'lganlar sonini kamayuvchi holatda tartiblash. [*GROUP BY*da *SELECT*da yozilgan ustunlar hammasi yozilishi shart.]





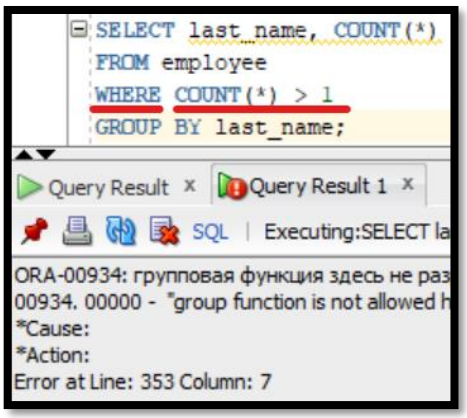
GROUP BY with HAVING

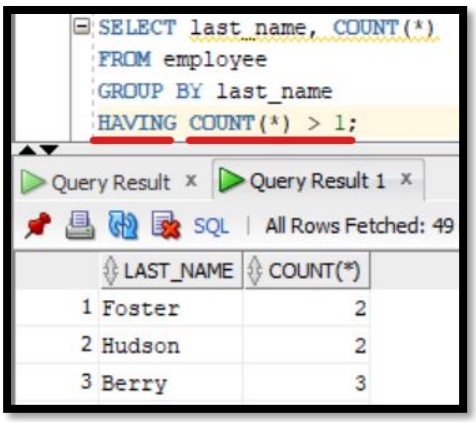
Shart. Familiyalarning takrorlanish sonlari 1dan ortig'i chiqarilsin.

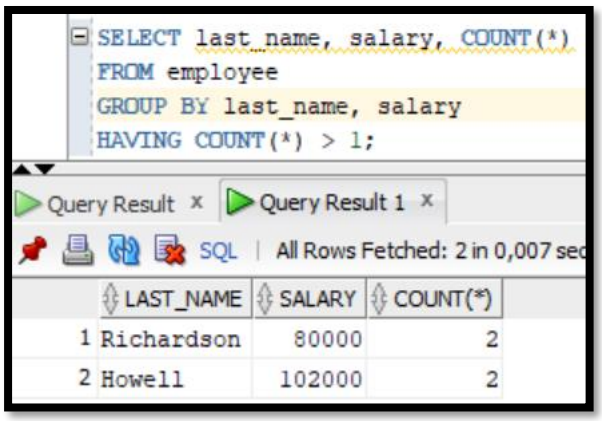
1-holat. Ishlamaydi. Chunki *WHERE*da *Group Function* ishlatilmaydi.

2-holat. Ishlaydi. *HAVING* ishlatilindi.

3-holat. Familiyalari va oyliklari bir xillar soni 2tadan oshig'i chiqarish.





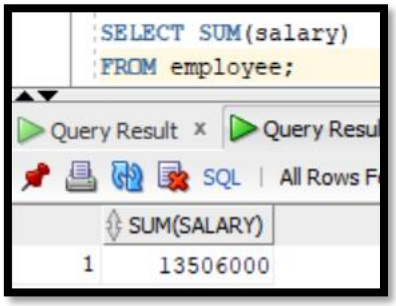


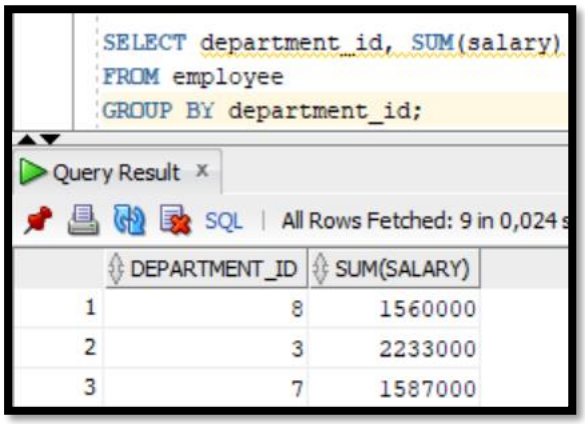
SUM function

SUM — yig'indi.

1-holat. Oyliklar yig'indisi.

2-holat. *Department_id*lari bir xil xodimlarning oyliklari yig'indisi



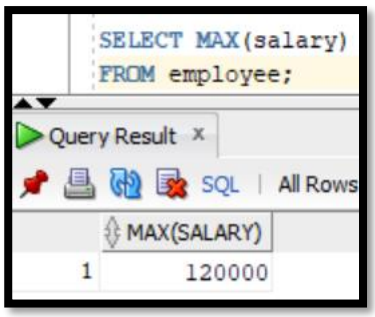


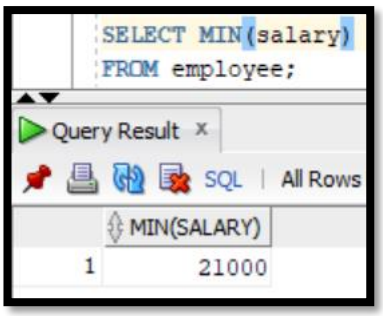
MAX() and MIN() function

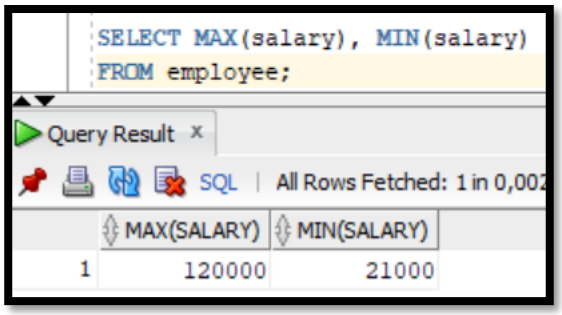
1-holat. Maksimal oylik narxini chiqarish.

2-holat. Minimal oylik narxini chiqarish.

3-holat. Maksimal va Minimal oylik narxini chiqarish.







4-holat. *Department_id*lari o’zaro bir xil bo’lganlarning eng katta, eng kichik va yig’indi oylik narxlari

```
SELECT department_id, MAX(salary), MIN(salary), SUM(salary)
FROM employee
GROUP BY department_id;
```

Query Result | SQL | All Rows Fetched: 9 in 0,007 seconds

	DEPARTMENT_ID	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
1	8	120000	23000	1560000
2	3	120000	25000	2233000
3	7	114000	21000	1587000

5-holat. MAX() va MIN() sanalarda ham ishlaydi.

```
SELECT MIN(hire_date), MAX(hire_date)
FROM employee;
```

Query Result | SQL | All Rows Fetched: 1 in 0,002 second

	MIN(HIRE_DATE)	MAX(HIRE_DATE)
1	03.01.10	24.01.17

6-holat. MAX() va MIN() harflarda ham ishlaydi.

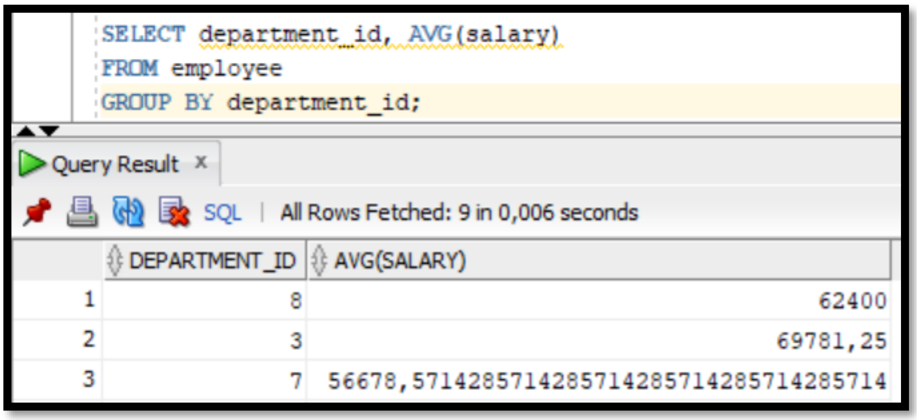
```
SELECT MIN(first_name), MAX(last_name)
FROM employee;
```

Query Result | SQL | All Rows Fetched: 1 in 0,004 seconds

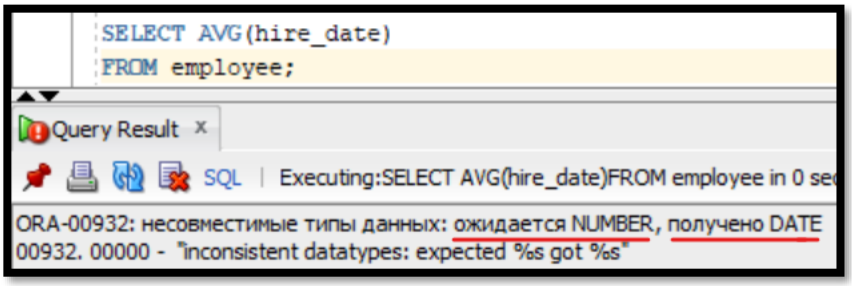
	MIN(FIRST_NAME)	MAX(LAST_NAME)
1	Adam	Woods

AVG function

1-holat. *Department_id*lari o’zaro bir xil bo’lganlarning o’rtacha oylik narxlari



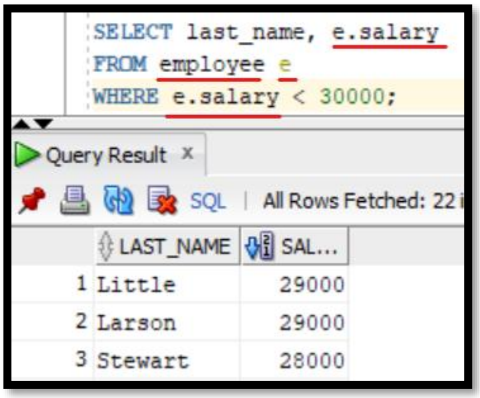
2-holat. Nomuvofiq datatip. Kutilyapti NUMBER, kelayotgani DATE

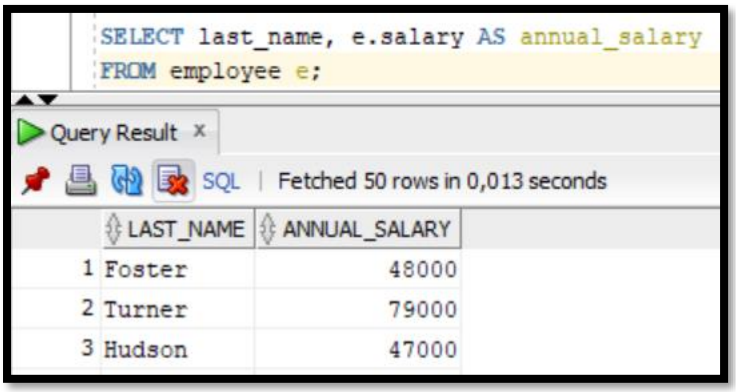


JOINS

TABLE ALIASES

TAXALLUS — uzun nomlarni yozish uchun ketadigan vaqtni tejab beradi.





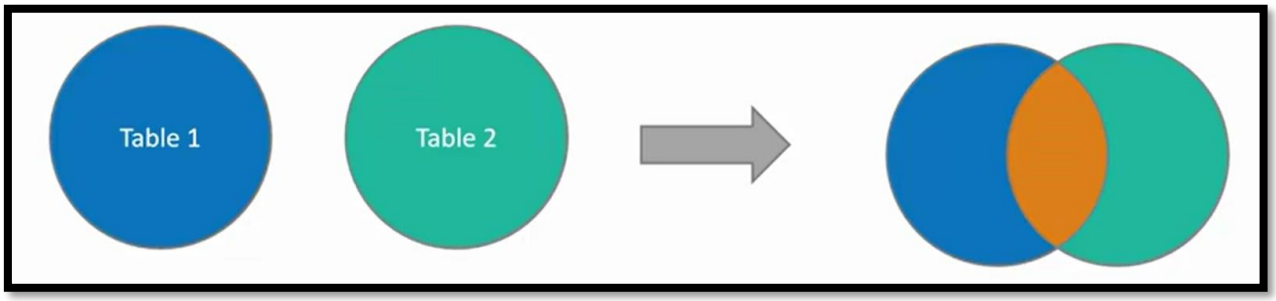
```
SELECT last_name, e.salary AS "Annual Salary", e.salary/12 AS monthly_salary
FROM employee e;
```

Query Result | SQL | Fetched 50 rows in 0,011 seconds

	LAST_NAME	Annual Salary	MONTHLY_SALARY
1	Foster	48000	4000
2	Turner	79000	6583,33333333333333333333333333333333333
3	Hudson	47000	3916,66666666666666666666666666666666667

INNER JOIN

JOIN — 2ta jadvalni qo’shib chiqarishda froydalaniladi. 1 ta so’rovda bir nechta jadval ma’lumotlarini chiqarishda foydalaniladi.




```
SELECT employee_id,
first_name,
last_name,
department_name
FROM employee
JOIN department ON employee.department_id = department.department_id;
```

Query Result | SQL | All Rows Fetched: 197 in 0,048 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
1	1	Michelle	Foster	Legal
2	2	Cheryl	Turner	Customer Support
3	3	Carolyn	Hudson	Finance

Pastda xatolik yuz berdi. Sababi *department_id* ustuni qaysi jadvalga tegishli ekanligi aniqlanmadi.

```
SELECT employee_id,
first_name,
last_name,
department_name,
department_id    -- department.department_id  (to'g'ri holati)
FROM employee
JOIN department ON employee.department_id = department.department_id;
```

Query Result

SQL | Executing:SELECT employee_id, first_name, last_name, department_name,depart

ORA-00918: столбец определен неоднозначно
00918. 00000 - "column ambiguously defined"

Taxalluslar bilan nomlarni qisqartirdik.

```
SELECT e.employee_id,
e.first_name,
e.last_name,
d.department_name,
d.department_id
FROM employee e
JOIN department d ON e.department_id = d.department_id;
```

Query Result

SQL | Fetched 50 rows in 0,017 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	DEPARTMENT_ID
1	1	Michelle	Foster	Legal	8
2	2	Cheryl	Turner	Customer Support	3
3	3	Carolyn	Hudson	Finance	7

Mana shu so’rovga shart kiritamiz.

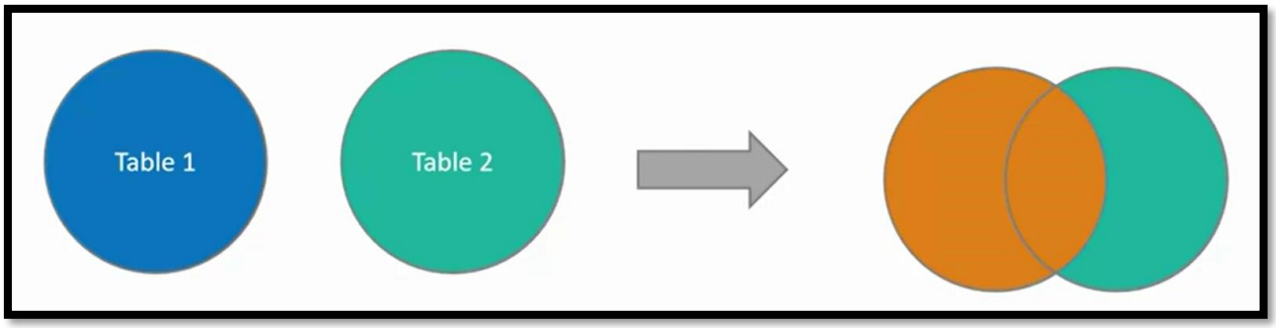
```
SELECT e.employee_id,
e.first_name,
e.last_name,
d.department_name,
d.department_id,
e.salary
FROM employee e
JOIN department d ON e.department_id = d.department_id
WHERE e.salary > 60000;
```

Query Result

SQL | Fetched 50 rows in 0,007 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	DEPARTMENT_ID	SAL...
1	117	Anna	Green	Customer Support	3	61000
2	36	Teresa	Graham	Marketing	6	61000
3	13	Stephen	Hudson	Hardware Development	4	63000

LEFT JOIN



INNER JOIN holatida faqat kesishmalar olinadi. (50ta qiymat)

```
SELECT
c.customer_id,
c.first_name,
c.last_name,
co.order_date
FROM customer c
INNER JOIN customer_order co ON c.customer_id = co.customer_id;
```

Query Result

SQL | All Rows Fetched: 50 in 0,007 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ORDER_DATE
1	3	Lois	Lawson	18.12.16
2	5	Ralph	Montgomery	01.06.15
3	6	Dorothy	Armstrong	26.09.16

LEFT JOIN holatida chapdagi barcha qiymatlar olinadi. Kesishmagan joyi *NULL* bilan to'ldiriladi. (52ta qiymat)

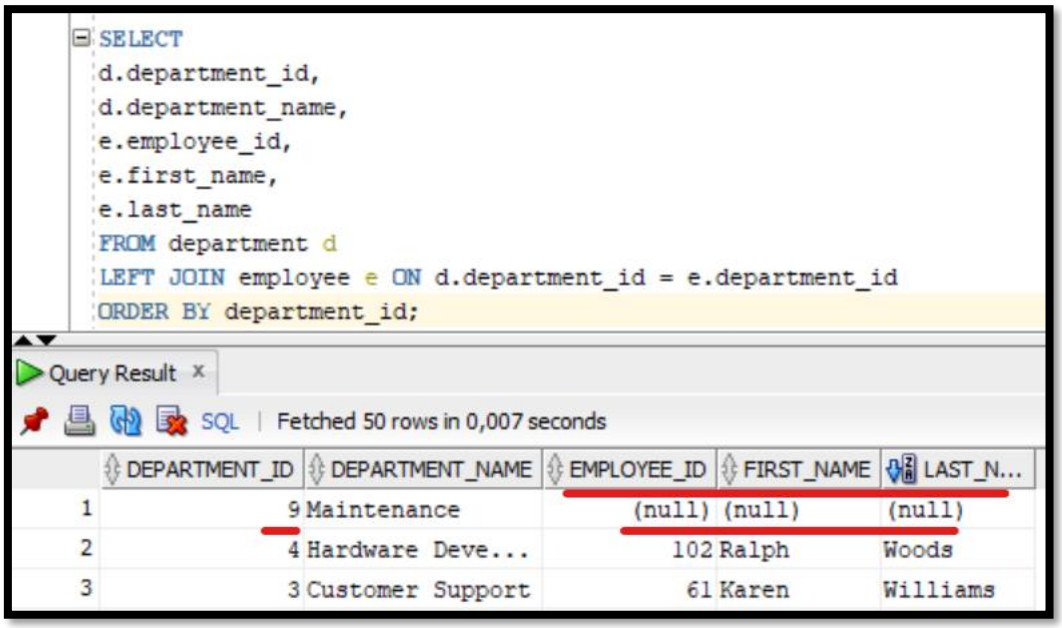
```
SELECT
c.customer_id,
c.first_name,
c.last_name,
co.order_date
FROM customer c
LEFT JOIN customer_order co ON c.customer_id = co.customer_id;
```

Query Result

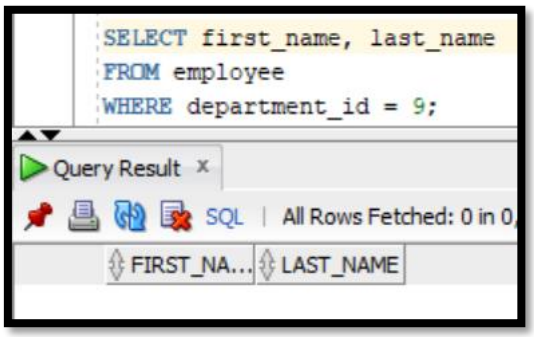
SQL | All Rows Fetched: 52 in 0,009 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ORDER_D...
1	1	Teresa	Hudson	(null)
2	4	Alice	Perry	(null)
3	2	Fred	Montgomery	23.01.17

Department_id = 9 bo’lgan qismda hodimlar yo’qligi aniqlandi.

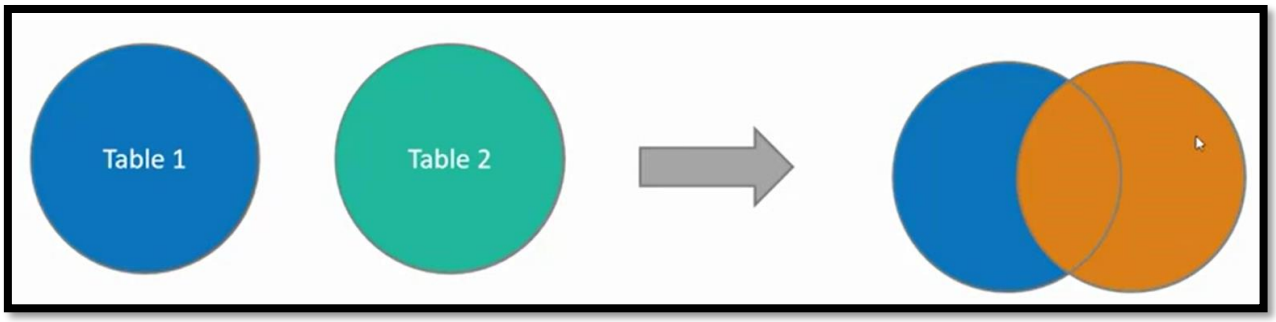


Tekshirib ko’ramiz.



RIGHT JOIN

RIGHT JOIN => *LEFT JOIN*ning huddi o’zginasi. Farqi bu o’ng tarafni oladi xolos.



2ta sinonim holat. Biri *LEFT JOIN* biri *RIGHT JOIN*

```
SELECT
d.department_id,
d.department_name,
e.employee_id,
e.first_name,
e.last_name
FROM department d
LEFT JOIN employee e ON d.department_id = e.department_id
ORDER BY department_id;
```

Query Result

SQL | Fetched 50 rows in 0,007 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEE_ID	FIRST_NAME	LAST_N...
1	9	Maintenance	(null)	(null)	(null)
2	4	Hardware Deve...	102	Ralph	Woods
3	3	Customer Support	61	Karen	Williams

```
SELECT
d.department_id,
d.department_name,
e.employee_id,
e.first_name,
e.last_name
FROM employee e
RIGHT JOIN department d ON e.department_id = d.department_id
ORDER BY department_id;
```

Query Result

SQL | All Rows Fetched: 198 in 0,022 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEE_ID	FIRST_NAME	LAST_N...
1	9	Maintenance	(null)	(null)	(null)
2	4	Hardware Deve...	102	Ralph	Woods
3	3	Customer Support	61	Karen	Williams

Employee va *Department* o’rnini almashtiramiz. Departmentsiz (hech bir bo’limga ega bo’lmagan) hodimlar *NULL* deb chiqarildi.

```sql
SELECT
d.department_id,
d.department_name,
e.employee_id,
e.first_name,
e.last_name
FROM department d
RIGHT JOIN employee e ON d.department_id = e.department_id
ORDER BY department_id;
```

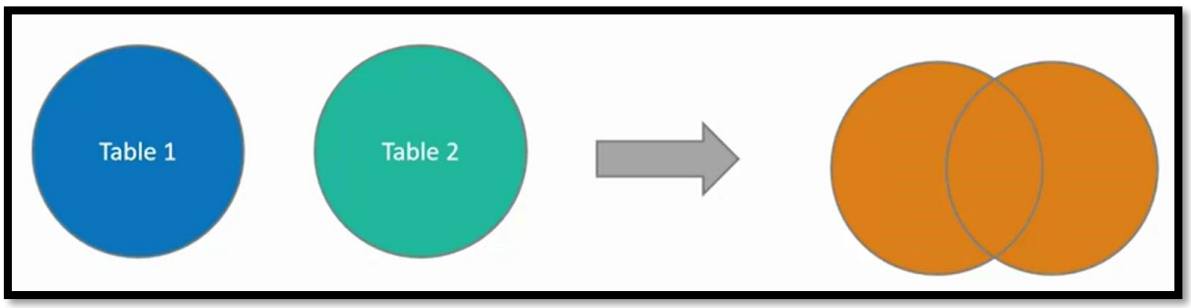
Query Result

SQL | Fetched 50 rows in 0,007 seconds

	DEPARTMENT...	DEPARTMENT_NAME	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	(null)	(null)	84	Jennifer	Long
2	(null)	(null)	140	Rebecca	Chapman
3	(null)	(null)	114	James	Phillips
4	8	Legal	77	David	Thompson

FULL JOIN

FULL JOIN — barcha jadvallarni qamrab oladi.




```sql
SELECT
d.department_id,
d.department_name,
e.employee_id,
e.first_name,
e.last_name
FROM department d
FULL JOIN employee e ON d.department_id = e.department_id
ORDER BY d.department_id, e.employee_id;
```

Query Result

SQL | All Rows Fetched: 201 in 0,018 seconds

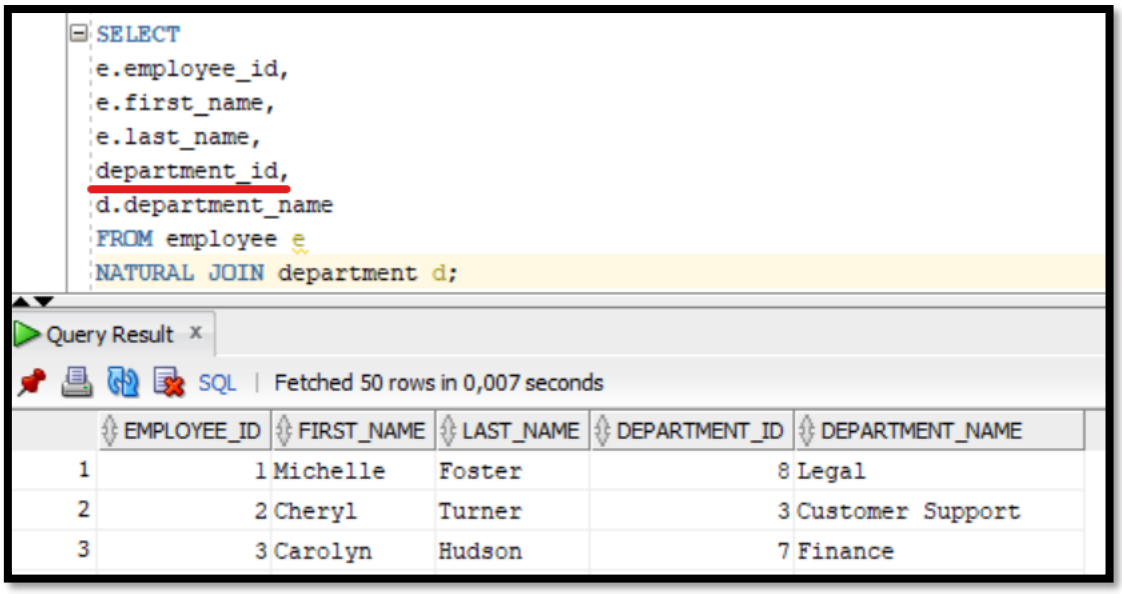
	DEPARTMENT...	DEPARTMENT_NAME	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	(null)	(null)	84	Jennifer	Long
2	(null)	(null)	114	James	Phillips
3	(null)	(null)	140	Rebecca	Chapman
4	9	Maintenance	(null)	(null)	(null)
5	8	Legal	1	Michelle	Foster

Bunda 201ta qiymat aniqlandi sababi. *Employee*da 200ta qiymat bor va 9-bo'limda hodim yo'qligi ko'rsatilgan.

NATURAL JOIN

NATURAL JOIN. Nomlari bir xil bo'lgan ustunlar qo'shilishida va 1-jadvaldagi asosiy kalit 2-jadvaldagi tashqi kalit bilan bil xil bo'lganda foydalaniladi.

Lekin foydalanmaslik tavsiya etiladi.



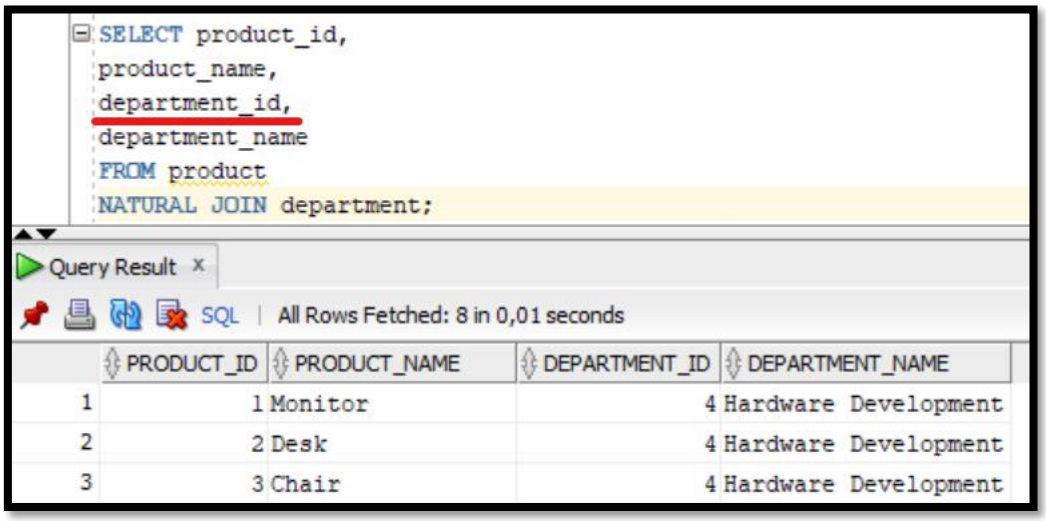
```
SELECT
e.employee_id,
e.first_name,
e.last_name,
department_id,
d.department_name
FROM employee e
NATURAL JOIN department d;
```

Query Result | SQL | Fetched 50 rows in 0,007 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	1	Michelle	Foster	8	Legal
2	2	Cheryl	Turner	3	Customer Support
3	3	Carolyn	Hudson	7	Finance

Ustun nomlari bir xil bo'lgani uchun ishladi.

Yana bir misol:



```
SELECT product_id,
product_name,
department_id,
department_name
FROM product
NATURAL JOIN department;
```

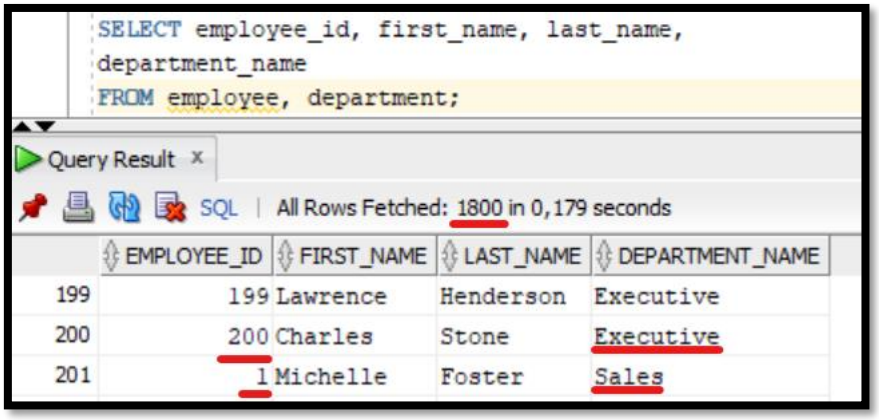
Query Result | SQL | All Rows Fetched: 8 in 0,01 seconds

	PRODUCT_ID	PRODUCT_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	1	Monitor	4	Hardware Development
2	2	Desk	4	Hardware Development
3	3	Chair	4	Hardware Development

CARTESIAN or CROSS JOIN

2ta jadval berilgan satrlari (200 va 9) CROSS JOIN ishlatilsa, natijaviy satrlar soni 200 x 9 = 1800 ta bo'ladi.

Bu yerda 200ta ism executive bo'lib chiqdi va 1-dan boshlab yana 200 tasi sales bo'lib chiqadi va h.z.



SELF JOIN

SELF JOIN — O'z-o'ziga qo'shilish.

Bunda 1ta jadvalda bir ustun boshqa bir ustunga bog'langan bo'ladi.

Misol uchun hodimlar jadvalida 1 hodimning menejeri bor, u menejer ham asli bir hodim, u menejerning ham menejeri bo'lishi mumkin. Ya'ni quyidagicha:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID
1	Michelle	Foster	162
162	Kimberly	Mendoza	191
191	Randy	Spencer	193

Employee jadvalidan misol ko'ramiz.

```
SELECT employee_id,
first_name,
last_name,
manager_id
FROM employee;
```

Query Result | SQL | All Rows Fetched: 200 in 0,019 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID
1	1	Michelle	Foster	162
2	2	Cheryl	Turner	99
3	3	Carolyn	Hudson	199

*SELF JOIN*ni ishlatdik. *Manager_id = employee_id*

```
SELECT emp.employee_id,
emp.first_name,
emp.last_name,
emp.manager_id,
mgr.first_name,
mgr.last_name
FROM employee emp
LEFT JOIN employee mgr ON emp.manager_id = mgr.employee_id;
```

Query Result | SQL | Fetched 50 rows in 0,021 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER...	FIRST_NAME_1	LAST_NAME_1
1	5	Doris	Powell	(null)	(null)	(null)
2	23	Teresa	Bell	200	Charles	Stone
3	133	Pamela	Collins	199	Lawrence	Henderson
4	3	Carolyn	Hudson	199	Lawrence	Henderson

NULL qiymatli qator mavjudligining sababi *id=5* bo'lgan hodimning menejeri mavjud emas. Uning o'zi eng katta menejer yoki shunchaki ish hamkori.

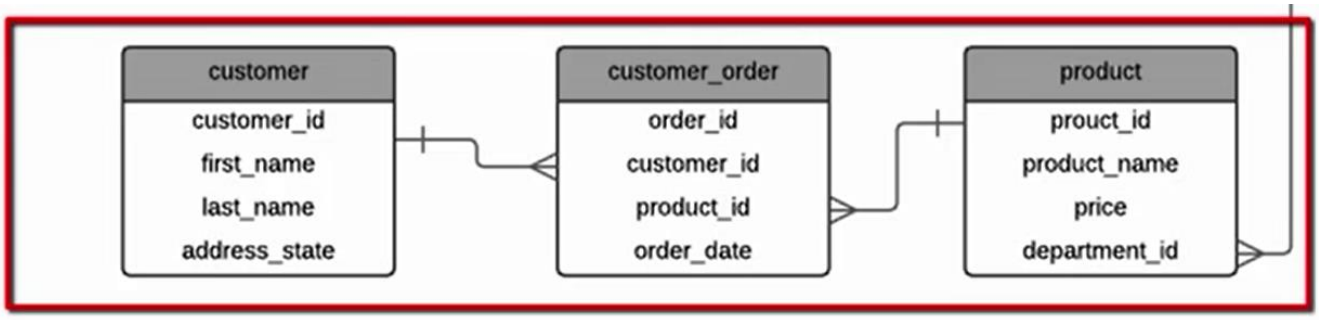
JOINING MANY TABLES

Bir nechta ustunlar uchun JOINlarni qo'llaymiz.

- SELECT ... FROM table1
  JOIN table2 ON table1.col = table2.col
  JOIN table3 ON table2.col = table3.col

- Combination of INNER, LEFT, RIGHT, etc...

Bizning bazamiz arxitekturasining bir qismi:



Bu yerda haridor va haridlarni bir-biriga bog’liq holda chiqardik:

```
SELECT
c.customer_id,
c.first_name,
c.last_name,
c.address_state,
co.order_id,
co.order_date
FROM customer c
JOIN customer_order co ON c.customer_id = co.customer_id;
```

Query Result | SQL | Fetched 50 rows in 0,006 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	ORDER_ID	ORDER_DATE
1	3	Lois	Lawson	OR	1	18.12.16
2	5	Ralph	Montgomery	TX	2	01.06.15
3	6	Dorothy	Armstrong	OR	3	26.09.16

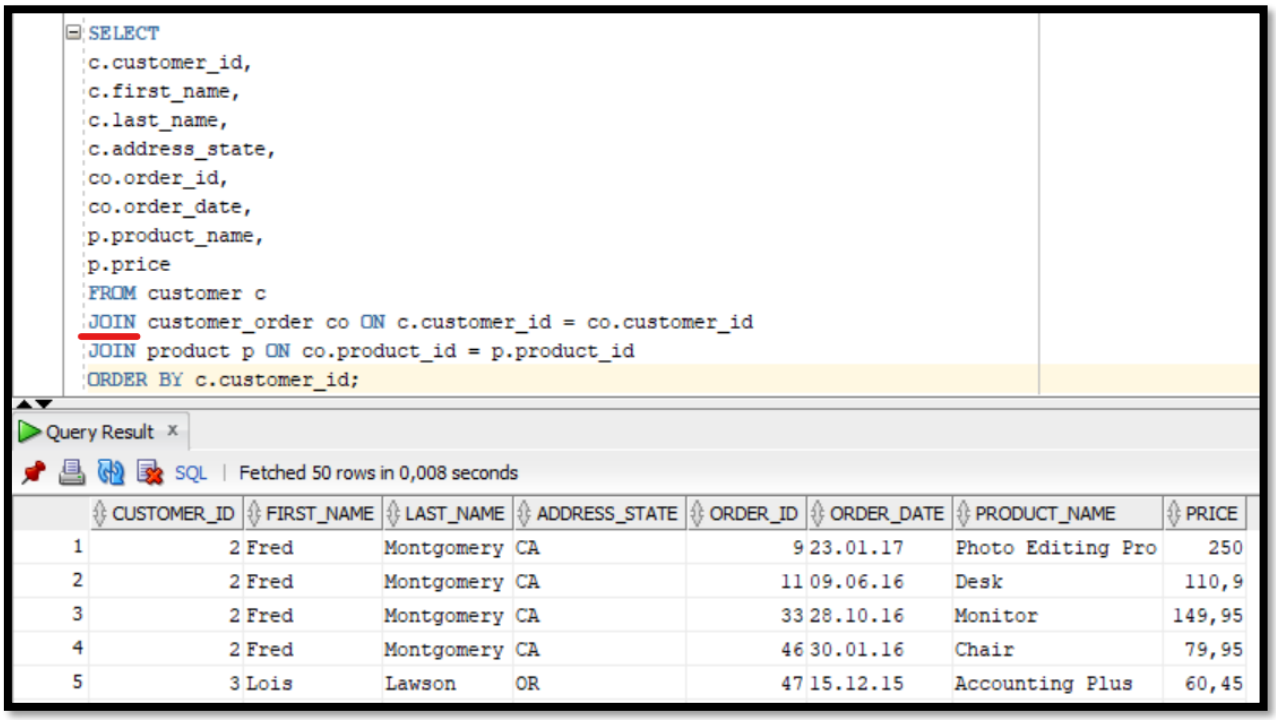
Bu yerda esa Customer Customer_order va Product jadvallarini bog’liq holda chiqardik:

```
SELECT
c.customer_id,
c.first_name,
c.last_name,
c.address_state,
co.order_id,
co.order_date,
p.product_name,
p.price
FROM customer c
JOIN customer_order co ON c.customer_id = co.customer_id
JOIN product p ON co.product_id = p.product_id ;
```

Query Result | SQL | Fetched 50 rows in 0,004 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	ORDER_ID	ORDER_DATE	PRODUCT_NAME	PRICE
1	2	Fred	Montgomery	CA	9	23.01.17	Photo Editing Pro	250
2	2	Fred	Montgomery	CA	11	09.06.16	Desk	110,9
3	2	Fred	Montgomery	CA	33	28.10.16	Monitor	149,95

Bu yerda Customer_id bo’yicha tartiblandi va bundan har bir haridor haridlarini tartib bilan ko’rish mumkin.

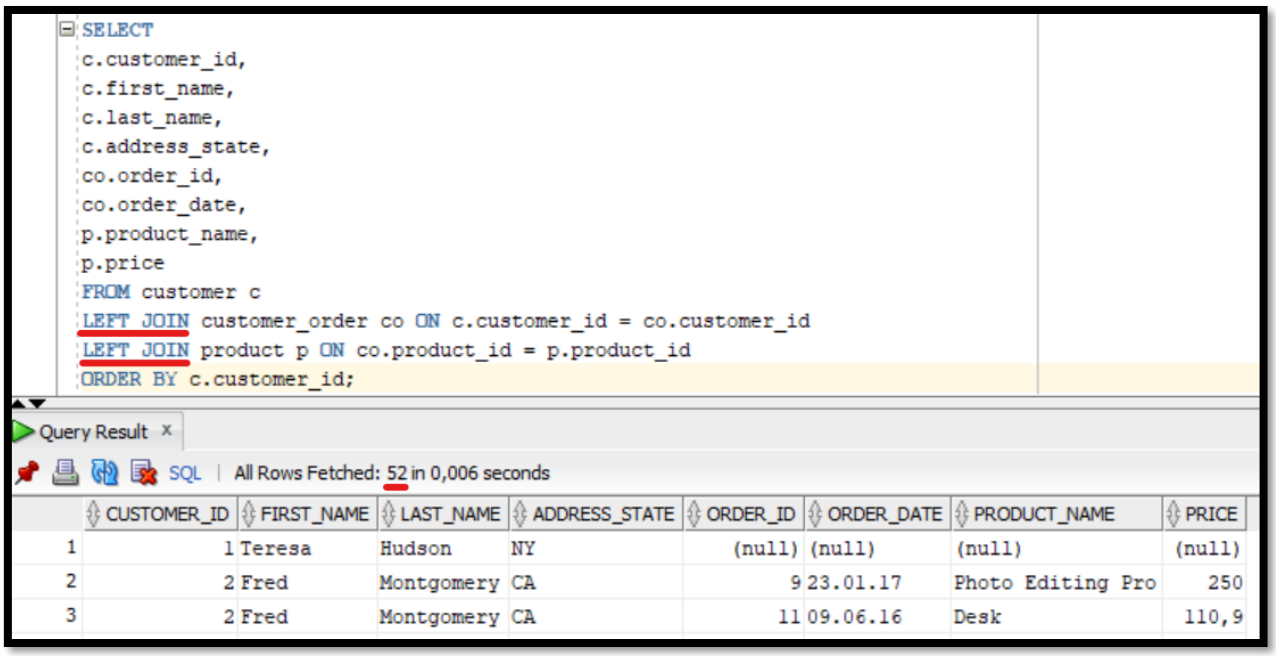


```
SELECT
c.customer_id,
c.first_name,
c.last_name,
c.address_state,
co.order_id,
co.order_date,
p.product_name,
p.price
FROM customer c
JOIN customer_order co ON c.customer_id = co.customer_id
JOIN product p ON co.product_id = p.product_id
ORDER BY c.customer_id;
```

Query Result | SQL | Fetched 50 rows in 0,008 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	ORDER_ID	ORDER_DATE	PRODUCT_NAME	PRICE
1	2	Fred	Montgomery	CA	9	23.01.17	Photo Editing Pro	250
2	2	Fred	Montgomery	CA	11	09.06.16	Desk	110,9
3	2	Fred	Montgomery	CA	33	28.10.16	Monitor	149,95
4	2	Fred	Montgomery	CA	46	30.01.16	Chair	79,95
5	3	Lois	Lawson	OR	47	15.12.15	Accounting Plus	60,45

Bu yerda harid qilinsa qilinmasa, barcha haridorlar ko’rsatilyapti:



```
SELECT
c.customer_id,
c.first_name,
c.last_name,
c.address_state,
co.order_id,
co.order_date,
p.product_name,
p.price
FROM customer c
LEFT JOIN customer_order co ON c.customer_id = co.customer_id
LEFT JOIN product p ON co.product_id = p.product_id
ORDER BY c.customer_id;
```

Query Result | SQL | All Rows Fetched: 52 in 0,006 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	ORDER_ID	ORDER_DATE	PRODUCT_NAME	PRICE
1	1	Teresa	Hudson	NY	(null)	(null)	(null)	(null)
2	2	Fred	Montgomery	CA	9	23.01.17	Photo Editing Pro	250
3	2	Fred	Montgomery	CA	11	09.06.16	Desk	110,9

Bu yerda department jadvali ham qo’shib chiqarildi:

```
SELECT
c.customer_id,
c.first_name,
c.last_name,
c.address_state,
co.order_id,
co.order_date,
p.product_name,
p.price,
d.department_name
FROM customer c
LEFT JOIN customer_order co ON c.customer_id = co.customer_id
LEFT JOIN product p ON co.product_id = p.product_id
LEFT JOIN department d ON p.department_id = d.department_id
ORDER BY c.customer_id;
```

Script Output | Query Result

SQL | All Rows Fetched: 52 in 0,008 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	ORDER_ID	ORDER_DATE	PRODUCT_NAME	PRICE	DEPARTMENT_NAME
1	1	Teresa	Hudson	NY	(null)	(null)	(null)	(null)	(null)
2	2	Fred	Montgomery	CA	9	23.01.17	Photo Editing Pro	250	Software Development
3	2	Fred	Montgomery	CA	11	09.06.16	Desk	110,9	Hardware Development

Alternative JOIN syntax

*JOIN*larning muqobil (o'rindosh yoki sinonim) sintaksisi

```
SELECT
employee_id,
first_name,
last_name,
department_name
FROM employee, department
WHERE employee.department_id = department.department_id;
```

Script Output | Query Result

SQL | All Rows Fetched: 197 in 0,019 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
1	1	Michelle	Foster	Legal
2	2	Cheryl	Turner	Customer Support
3	3	Carolyn	Hudson	Finance

+ belgisini qo'yishdagi o'zgarish.

O'ng tarafga qo'yilsa, o'ng taraf *NULL*lar bilan to'ldiriladi.

Chap tarafga qo'yilsa, chap taraf *NULL*lar bilan to'ldiriladi.

```
SELECT
employee_id,
first_name,
last_name,
department_name
FROM employee, department
WHERE employee.department_id = department.department_id(+);
```

Script Output | Query Result

SQL | All Rows Fetched: 200 in 0,012 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_N...
1	84	Jennifer	Long	(null)
2	140	Rebecca	Chapman	(null)
3	114	James	Phillips	(null)
4	119	Paula	Lynch	Software Deve...

```
SELECT
employee_id,
first_name,
last_name,
department_name
FROM employee, department
WHERE employee.department_id(+) = department.department_id;
```

Script Output | Query Result

SQL | All Rows Fetched: 198 in 0,021 seconds

	EMPLOYEE...	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
1	(null)	(null)	(null)	Maintenance
2	200	Charles	Stone	Software Development
3	199	Lawrence	Henderson	Marketing

Qo’shimcha shartlarni qo’shib ko’ramiz:

```
SELECT
employee_id,
first_name,
last_name,
department_name,
salary,
hire_date
FROM employee, department
WHERE salary > 50000
AND hire_date > '01.01.2012'
AND employee.department_id = department.department_id;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0,004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	SALARY	HIRE_DATE
1	2	Cheryl	Turner	Customer Support	79000	02.01.12
2	9	Kathleen	Jones	Finance	92000	15.03.15
3	11	Norma	Henry	Customer Support	56000	24.07.13

Bitta eng kerakli qator bo'lmasligi *ALTERNATIVE JOIN*ni *CARTESIAN JOIN*ga aylantirib yuboradi.

```
SELECT
employee_id,
first_name,
last_name,
department_name,
salary,
hire_date
FROM employee, department
WHERE salary > 50000
AND hire_date > '01.01.2012'
--AND employee.department_id = department.department_id;
;
```

Script Output | Query Result

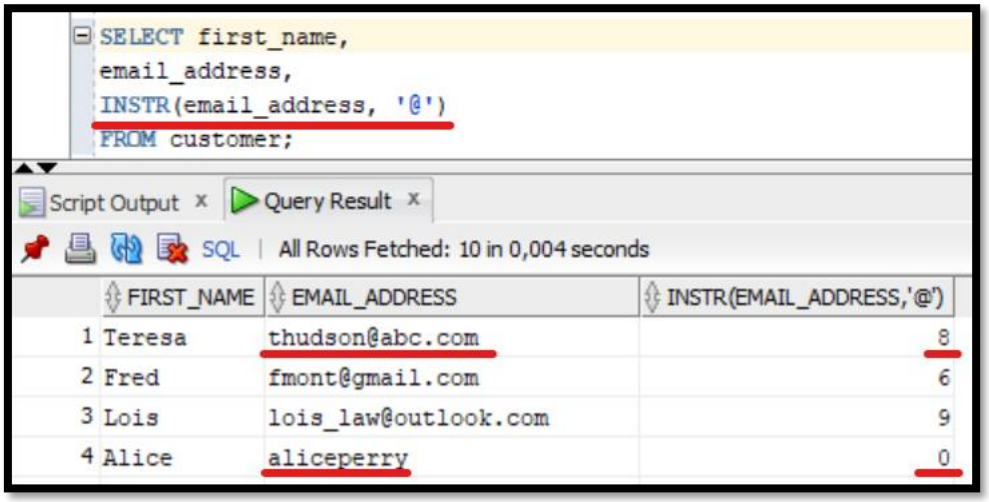
SQL | All Rows Fetched: 891 in 0,069 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	SALARY	HIRE_DATE
1	2	Cheryl	Turner	Executive	79000	02.01.12
2	9	Kathleen	Jones	Executive	92000	15.03.15
3	11	Norma	Henry	Executive	56000	24.07.13

FUNCTIONS

String functions

INSTR(matn, qidirilayotgan belgi) — belgini qidirib, uning tartib raqamini chiqaradi.

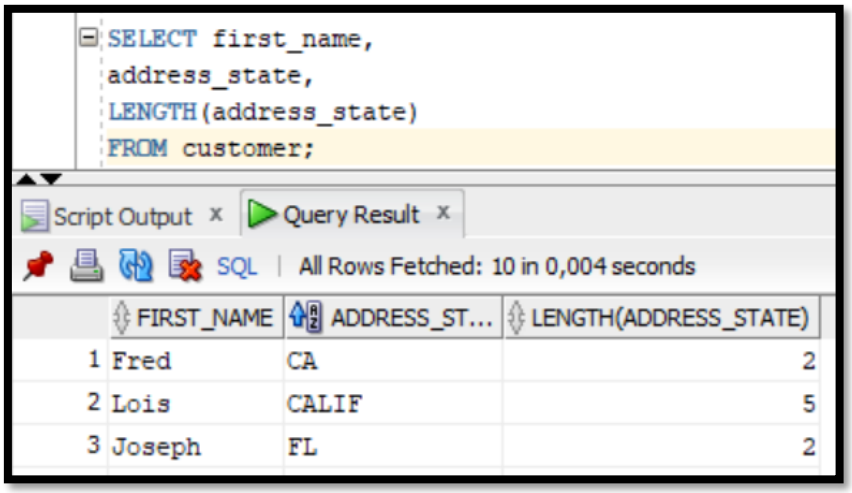


```
SELECT first_name,
email_address,
INSTR(email_address, '@')
FROM customer;
```

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0,004 seconds

	FIRST_NAME	EMAIL_ADDRESS	INSTR(EMAIL_ADDRESS,'@')
1	Teresa	thudson@abc.com	8
2	Fred	fmont@gmail.com	6
3	Lois	lois_law@outlook.com	9
4	Alice	aliceperry	0

LENGTH(matn) — belgilar sonini hisoblaydi.



```
SELECT first_name,
address_state,
LENGTH(address_state)
FROM customer;
```

Script Output | Query Result | SQL | All Rows Fetched: 10 in 0,004 seconds

	FIRST_NAME	ADDRESS_ST...	LENGTH(ADDRESS_STATE)
1	Fred	CA	2
2	Lois	CALIF	5
3	Joseph	FL	2

INSTR va LENGTH aniq qiymat qaytargani uchun raqamli shartga kiritish mumkin:

```
SELECT first_name,
email_address,
INSTR(email_address, '@')
address_state,
LENGTH(address_state)
FROM customer
WHERE INSTR(email_address, '@') > 0
AND LENGTH(address_state) = 2
ORDER BY LENGTH(address_state) DESC;
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0,005 seconds

	FIRST_NAME	EMAIL_ADDRESS	ADDRESS_STATE	LENGTH(ADDRESS_STATE)
1	Ralph	ralph_mont25@gmail.com	13	2
2	Fred	fmont@gmail.com	6	2
3	Lois	lois_law@outlook.com	9	2

Nesting functions within functions

Nesting functions — funksiyalarni kombinatsion ishlatish.

SUBSTR(matn, n-harfdan boshlab, m ta harf chiqarilsin) — Satrdan pastki qatorni ajratib oladi (5-pozitsiyadan boshlab, 3 ta belgini ajratib ol):

SELECT SUBSTR("SqlTutorial", 5, 3) ;
Natija: uto

@ belgisidan boshlab nechta belgi borligini aniqlash:

```
SELECT first_name,
email_address,
INSTR(email_address, '@'),
SUBSTR(email_address, INSTR(email_address, '@'), LENGTH(email_address))
FROM customer;
```

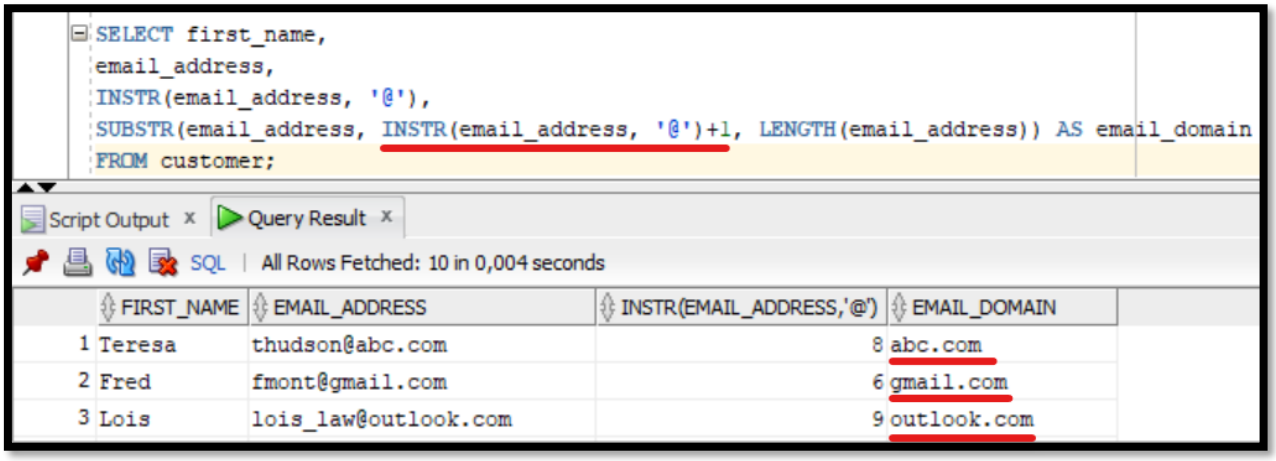
Script Output | Query Result

SQL | All Rows Fetched: 10 in 0,002 seconds

	FIRST_NAME	EMAIL_ADDRESS	INSTR(EMAIL_ADDRESS,'@')	SUBSTR(EMAIL_A
1	Teresa	thudson@abc.com	8	@abc.com
2	Fred	fmont@gmail.com	6	@gmail.com
3	Lois	lois_law@outlook.com	9	@outlook.com

Bunda *SUBSTR*dan foydalanib, *email_address* ustuni tanlandi, *INSTR* orqali @ belgisi aniqlandi, *LENGTH* orqali @ belgisi davomidagi belgilar ajratib olindi.

@ belgisidan keyingi belgilarni aniqlash. Ya'ni elektron pochta domenini aniqlash:



```
SELECT first_name,
email_address,
INSTR(email_address, '@'),
SUBSTR(email_address, INSTR(email_address, '@')+1, LENGTH(email_address)) AS email_domain
FROM customer;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0,004 seconds

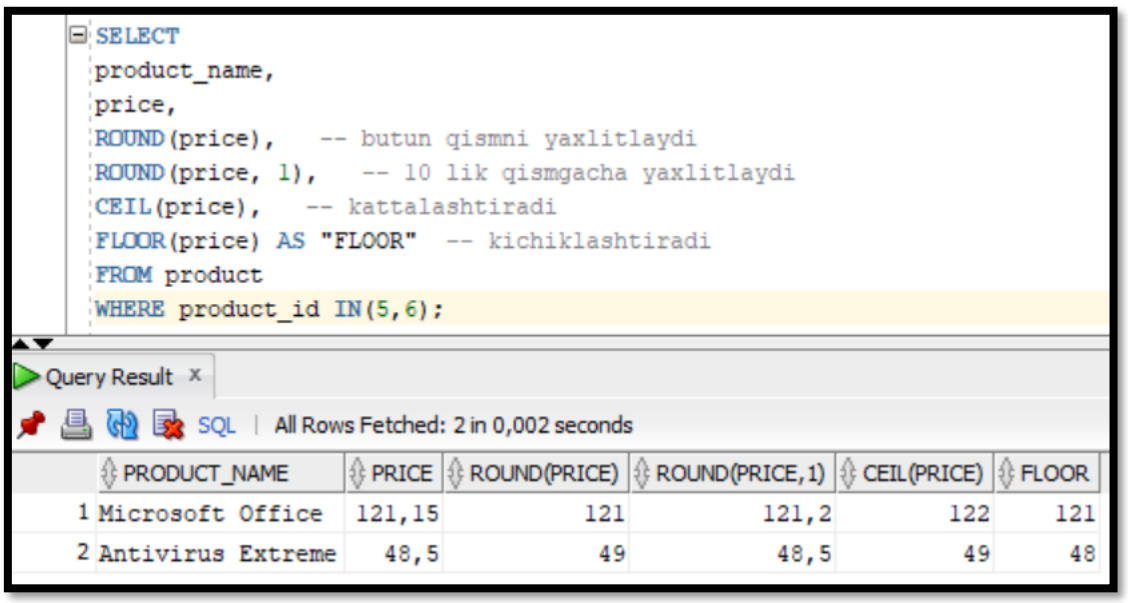
	FIRST_NAME	EMAIL_ADDRESS	INSTR(EMAIL_ADDRESS,'@')	EMAIL_DOMAIN
1	Teresa	thudson@abc.com	8	abc.com
2	Fred	fmont@gmail.com	6	gmail.com
3	Lois	lois_law@outlook.com	9	outlook.com

Number Function

ROUND(number, raqam [10 lik qismigacha]) — sonni yaxlitlaydi.

CEIL(number) — sonni kattasi tomonga yaxlitlaydi.

FLOOR(number) — faqat butun qismini oladi.



```
SELECT
product_name,
price,
ROUND(price),    -- butun qismni yaxlitlaydi
ROUND(price, 1),    -- 10 lik qismgacha yaxlitlaydi
CEIL(price),    -- kattalashtiradi
FLOOR(price) AS "FLOOR"  -- kichiklashtiradi
FROM product
WHERE product_id IN(5,6);
```

Query Result

SQL | All Rows Fetched: 2 in 0,002 seconds

	PRODUCT_NAME	PRICE	ROUND(PRICE)	ROUND(PRICE,1)	CEIL(PRICE)	FLOOR
1	Microsoft Office	121,15	121	121,2	122	121
2	Antivirus Extreme	48,5	49	48,5	49	48

Misol: 9 ta Monitor harid qilingani uchun 85% chegirma berildi. Uni hisoblash quyidagicha turda bo'lishi mumkin:

```
SELECT
product_name,
price,
price*9*0.85 AS NARX,
ROUND(price*9*0.85),
ROUND(price*9*0.85, 2),
CEIL(price*9*0.85),
FLOOR(price*9*0.85)
FROM product
WHERE product_id = 1;
```

Query Result

SQL | All Rows Fetched: 1 in 0,006 seconds

	PRODUCT_NAME	PRICE	NARX	ROUND(PRICE*9*0.85)	ROUND(PRICE*9*0.85,2)	CEIL(PRICE*9*0.85)	FLOOR(PRICE*9*0.85)
1	Monitor	149,95	1147,1175	1147	1147,12	1148	1147

Date Function

SYSDATE — hozirgi kun sanasi

ADD_MONTHS(sana, oylar_soni) — falon sanaga, falon oy qo’shish

MONTHS_BETWEEN(sanadan, sanagacha) — sanalar orasidagi oylarni chiqarib beradi.

```
SELECT
first_name,
last_name,
hire_date,
SYSDATE,
ADD_MONTHS(hire_date, 12) AS rewiev_date,
ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)/12, 1) AS years_with_company
FROM employee;
```

Query Result

SQL | Fetched 50 rows in 0,008 seconds

	FIRST_NAME	LAST_NAME	HIRE_DATE	SYSDATE	REWIEV_DATE	YEARS_WITH_COMPANY
1	Michelle	Foster	27.08.11	01.03.23	27.08.12	11,5
2	Cheryl	Turner	02.01.12	01.03.23	02.01.13	11,2
3	Carolyn	Hudson	04.12.16	01.03.23	04.12.17	6,2

Soddaroq misol:

```
SELECT
SYSDATE,
ADD_MONTHS(SYSDATE, 60),
MONTHS_BETWEEN(ADD_MONTHS(SYSDATE, 60), SYSDATE)/12 AS yillar_farqi
FROM dual;
```

Query Result

SQL | All Rows Fetched: 1 in 0,004 seconds

	SYSDATE	ADD_MONTHS(SYSDATE,60)	YILLAR_FARQI
1	01.03.23	01.03.28	5

Data Types and Conversion Function

- CHAR: character string with fixed size
- VARCHAR2: character string with a variable size
- NUMBER: stores numeric data with optional decimals
- DATE: stores date and time
- TIMESTAMP: stores date, time, and fractional seconds
- CLOB: stores large amounts of text

Converting

- TO_CHAR, TO_DATE, TO_NUMBER

*TO_CHAR*ga misol:

```
SELECT
first_name,
last_name,
hire_date,
TO_CHAR(hire_date, 'YYYY_MM_DD')
FROM employee;
```

Query Result

SQL | Fetched 50 rows in 0,004 seconds

	FIRST_NAME	LAST_NAME	HIRE_DATE	TO_CHAR(HIRE_DATE,'YYYY_MM_DD')
1	Michelle	Foster	27.08.11	2011_08_27
2	Cheryl	Turner	02.01.12	2012_01_02
3	Carolyn	Hudson	04.12.16	2016_12_04

TO_CHAR, TO_DATE va *TO_NUMBER*ga misol:

```
SELECT
TO_CHAR(SYSDATE,'DD_MM_YYYY') AS BUGUN,
TO_DATE('12_11_2010', 'DD_MM_YYYY') AS SANA,
TO_NUMBER('200') AS NOMER
FROM dual;
```

Query Result

SQL | All Rows Fetched: 1 in 0,005 seconds

	BUGUN	SANA	NOMER
1	01_03_2023	12.11.10	200

Case Statement

```
CASE [expression] WHEN condition_1 THEN result_1
WHEN condition_2 THEN result_2
...
WHEN condition_n THEN result_n
ELSE result
END case_name
```

Misol:

```
SELECT
product_id,
product_name,
price,
CASE
WHEN price > 100 THEN 'Over 100'
WHEN price <= 100 THEN  'Less than or under 100'
END price_group
FROM product;
```

Query Result

SQL | All Rows Fetched: 8 in 0,006 seconds

	PRODUCT_ID	PRODUCT_NAME	PRICE	PRICE_GROUP
1	1	Monitor	149,95	Over 100
2	2	Desk	110,9	Over 100
3	3	Chair	79,95	Less than or under 100

Murakkabroq misol:

```
SELECT
product_id,
product_name,
price,
CASE
WHEN price > 200 THEN 'Over 200'
WHEN price > 100 AND price <= 200 THEN  'Between 100 and 200'
WHEN price > 50 AND price <= 100 THEN  'Between 50 and 100'
ELSE 'Under 50'
END price_group
FROM product;
```

Query Result

SQL | All Rows Fetched: 8 in 0,006 seconds

	PRODUCT_ID	PRODUCT_NAME	PRICE	PRICE_GROUP
4	4	Accounting Plus	60,45	Between 50 and 100
5	5	Microsoft Office	121,15	Between 100 and 200
6	6	Antivirus Extreme	48,5	Under 50
7	7	Photo Editing Pro	250	Over 200

Address_state bo’yicha *Case*ni qo’llash:

```
SELECT
first_name,
address_state,
CASE address_state
WHEN 'NY' THEN 'East'
WHEN 'CA' THEN 'West'
WHEN 'OR' THEN 'West'
WHEN 'SC' THEN 'East'
WHEN 'TX' THEN 'West'
WHEN 'FL' THEN 'East'
WHEN 'IN' THEN 'East'
ELSE 'Unknown'
END state_group
FROM customer;
```

Query Result

SQL | All Rows Fetched: 10 in 0,005 seconds

	FIRST_NAME	ADDRESS_STATE	STATE_GR...
5	Teresa	NY	East
6	Lois	CALIF	Unknown
7	Dorothy	OR	West

Huddi shunga sinonim va optimal kod:

```
SELECT
first_name,
address_state,
CASE
WHEN address_state IN ('NY', 'SC', 'FL', 'IN') THEN 'East'
WHEN address_state IN ('CA', 'OR', 'TX') THEN 'West'
ELSE 'Unknown'
END state_group
FROM customer;
```

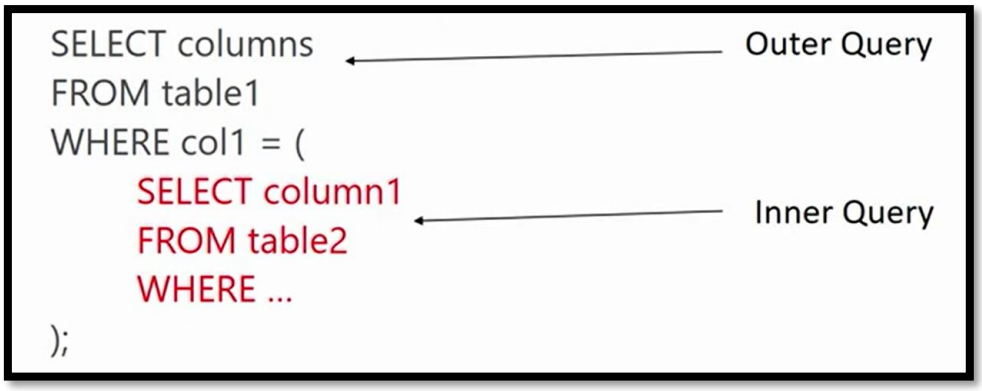
Query Result | SQL | All Rows Fetched: 10 in 0,023 seconds

	FIRST_NAME	ADDRESS_STATE	STATE_GR...
5	Teresa	NY	East
6	Lois	CALIF	Unknown
7	Dorothy	OR	West

Subqueries

About

So’rov osti so’rovlar



Types:

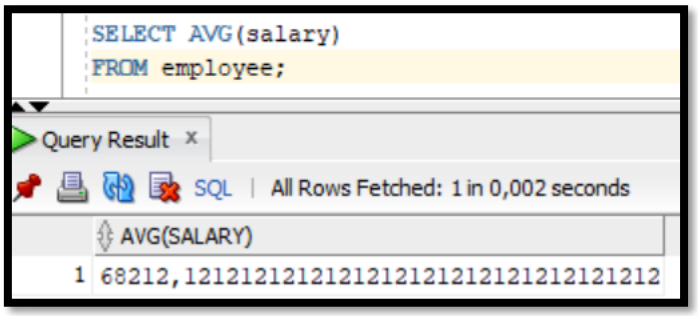
Single row subquery: returns one row (1 qatorli ichki so’rov)

Multi row subquery: returns multiple rows (ko’p qatorli ichki so’rov)

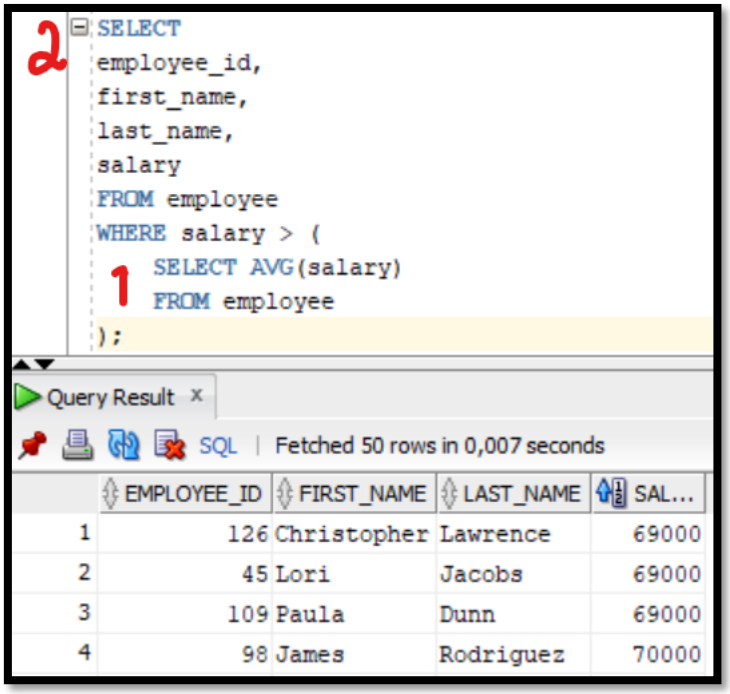
Single row subquery

Examples: AVG() ya’ni sonlarning o’rtachasi yagona qiymat hisoblanadi. Shuning uchun u Single row subquery

Ichki so’rov: maoshlarning o’rtacha qiymati aniqlab olindi.



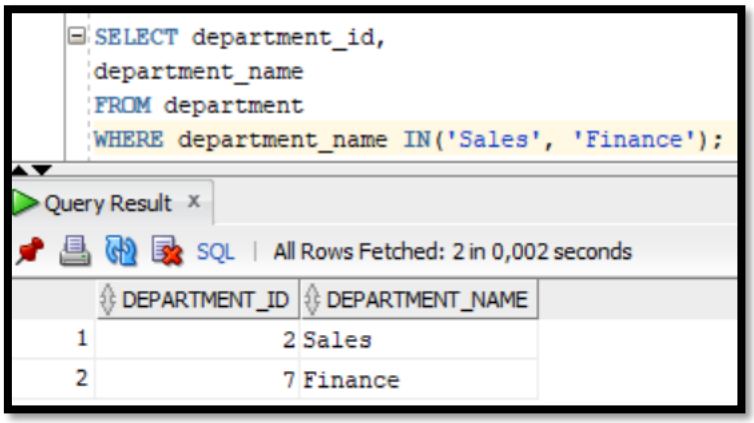
Tashqi so’rov + ichki so’rov: o’rtacha maoshdan yuqori maosh oluvchilar chiqarildi.



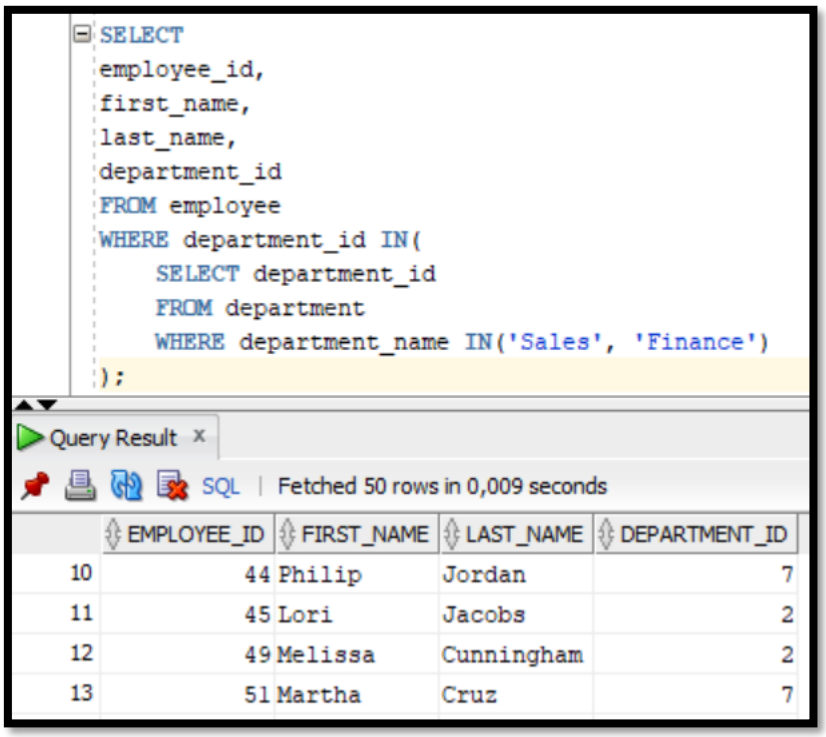
Multi row subquery

Ko’p qatorli ichki so’rovlar quyidagilar bilan ishlatiladi:

Department_name : Sales va Finance bo’lganlarni chiqarib oldik.



Tashqi + ichki so’rovlar jamlandi:



Inserting, Updating an Deleting Data

Inserting Data

Ma’lumotni bittalab kiritish:

```
INSERT INTO table_name [(col1, col2...)]
VALUES (value1, value2);
```

Ma’lumot kiritdik:

```
INSERT INTO employee(employee_id, first_name, last_name, department_id)
VALUES (300, 'John', 'Smith', 3);
```

Script Output | Query Result | Task completed in 0,039 seconds

```
1 row inserted.
```

```
INSERT INTO employee(employee_id, first_name, last_name, department_id, salary, manager_id, hire_date)
VALUES (301, 'Marge', 'Abbot', 2, 31000, 51, '04.01.2017');
```

Script Output | Query Result | Task completed in 0,051 seconds

```
1 row inserted.
```

Ma’lumotni birdaniga ko’p kiritish uchun shablon:

```
INSERT ALL
INTO table_name1(col1, col2) VALUES (val1, val2)
INTO table_name2(col1, col2) VALUES (val1, val2);
```

Misol:

```
INSERT ALL
INTO employee(employee_id, first_name, last_name, department_id) VALUES (303, 'Mark', 'Spencer', 4)
INTO employee(employee_id, first_name, last_name, department_id) VALUES (304, 'Simone', 'Fletcher', 3)
INTO employee(employee_id, first_name, last_name, department_id) VALUES (305, 'Alison', 'Smith', 8)
SELECT * FROM dual;
```

Script Output | Query Result | Task completed in 0,04 seconds

```
3 rows inserted.
```

Pastdagi natijani tozalaydi: Ctrl + Shift + D

Script Output | Task completed in 0,041 seconds

Clear (Ctrl+Shift+D)

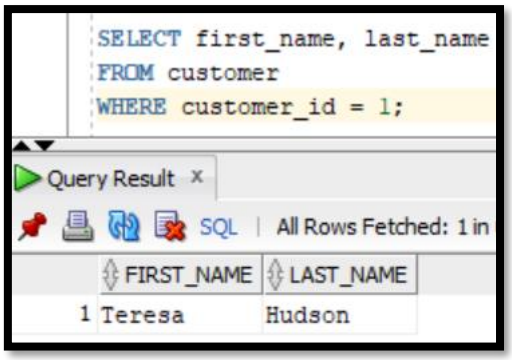
```
Error
SQL Error: ORA-00905: отсутствует ключевое слово
00905. 00000 -  "missing keyword"
```

Inserting Data From Another Table

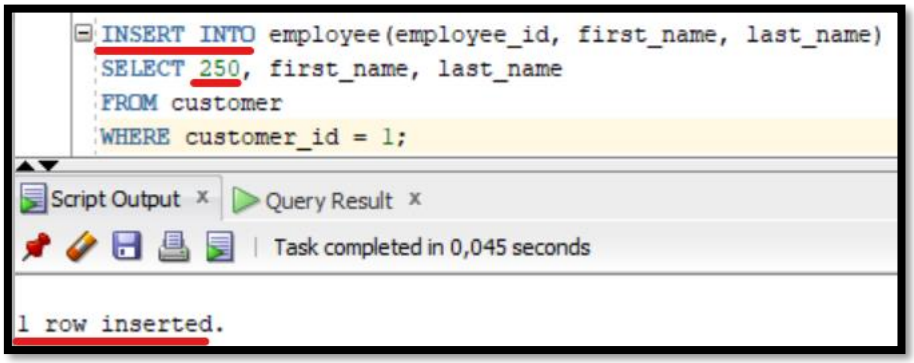
Boshqa jadvaldan ma’lumot qo’shish

- INSERT INTO target_table (columns)
SELECT (columns) FROM source_table...

Misol. Customer jadvalidagi id=1 bo’lgan shaxsni ko’ramiz.



Ushbu shaxsni employee jadvaliga kiritamiz:



Id NOT NULL bo’lgani uchun 250 deb kiritdik.

Updating Data

About

Ma’lumotlarni yangilash:

- UPDATE table
SET col1 = val1,
col2 = val2
[WHERE condition]

Bitta qator ma’lumotini o’zgartirish

Employee_id = 85 bo’lgan shaxsni chiqaramiz:

```
SELECT*
FROM employee
WHERE employee_id = 85;
```

Query Result | SQL | All Rows Fetched: 1 in 0,005 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
1	85	Johnny	Reed	22000

Uni oyligini 10.000 ga oshiramiz:

```
UPDATE employee
SET salary = salary + 10000
WHERE employee_id = 85;
```

Query Result | Script Output | Task completed in 0,04

```
1 row updated.
```

```
SELECT*
FROM employee
WHERE employee_id = 85;
```

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0,003 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
1	85	Johnny	Reed	32000

Ko’p qator ma’lumotlarini o’zgartirish

```
SELECT*
FROM employee
WHERE employee_id IN (102, 59, 16);
```

Script Output | Query Result | SQL | All Rows Fetched: 3 in 0,006 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	MANAGER_ID
1	16	Fred	Nichols	70000	3	186
2	59	Christine	Brooks	40000	6	57
3	102	Ralph	Woods	22000	4	10

```
UPDATE employee
SET manager_id = 30, salary = salary + 5000
WHERE employee_id IN (102, 59, 16);
```

Script Output | Query Result | Task completed in 0,049 seconds

```
3 rows updated.
```

```
SELECT*
FROM employee
WHERE employee_id IN (102, 59, 16);
```

Script Output | Query Result | SQL | All Rows Fetched: 3 in 0,004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	MANAGER_ID
1	16	Fred	Nichols	75000	3	30
2	59	Christine	Brooks	45000	6	30
3	102	Ralph	Woods	27000	4	30

Deleting Data

Ma’lumotlarni o’chirish.

O’chiriladigan ma’lumotni doim tekshirib olish kerak.

```
SELECT*FROM employee
WHERE employee_id = 250;
```

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0,004 sec

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	250	Teresa	Hudson

Endi uni o’chiramiz:

O’chirishda WHERE bandi borligiga ishonch hosil qiling.

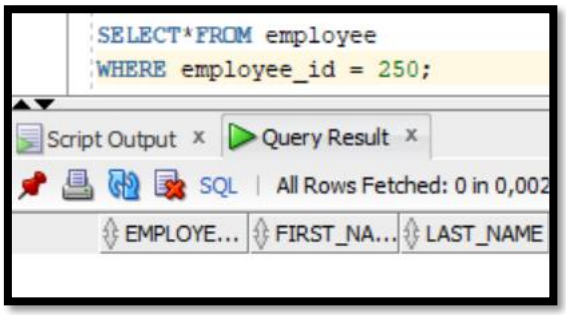
Aks holda barcha ma’lumotlar o’chib ketadi.

```
DELETE FROM employee
WHERE employee_id = 250;
```

Script Output | Query Result | Task completed in 0,0

```
1 row deleted.
```

Tekshiramiz:



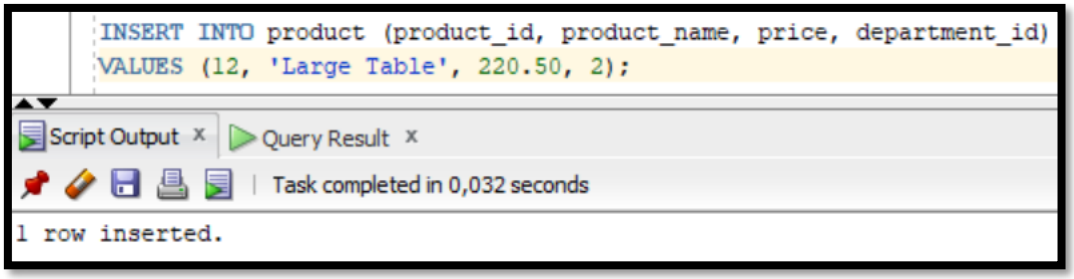
Endi u ma’lumot mavjud emas.

Commit and Rollback

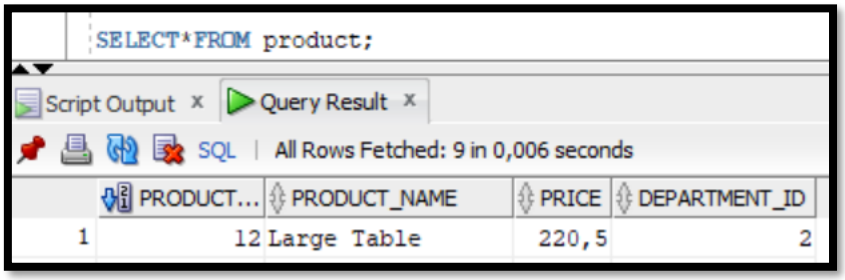
COMMIT — yangilanishni saqlash.

ROLLBACK — avvalgi holatiga qaytarish.

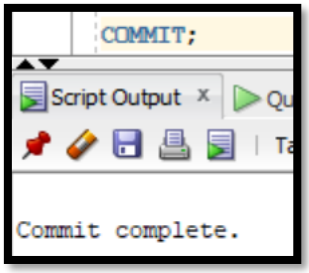
Ma’lumot kiritdik:



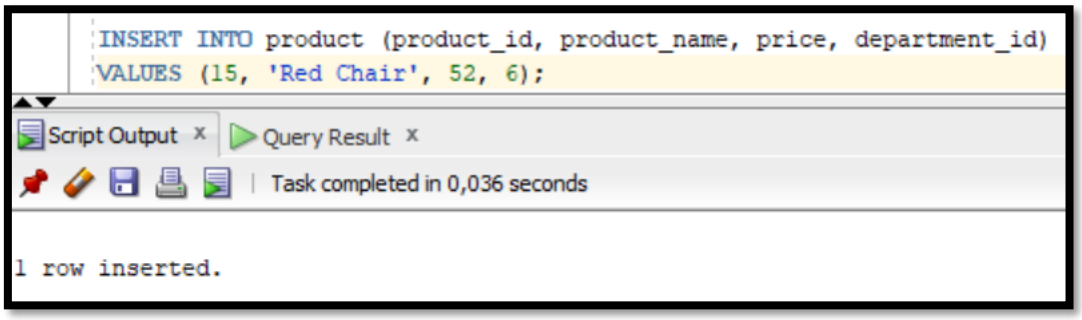
Tekshiramiz:



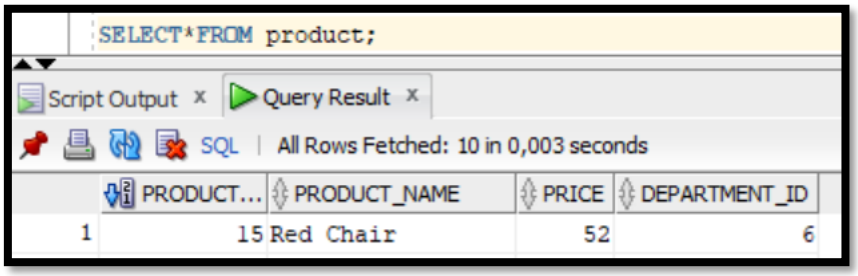
COMMIT: saqlash



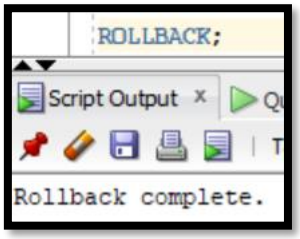
Yana ma’lumot kiritamiz:



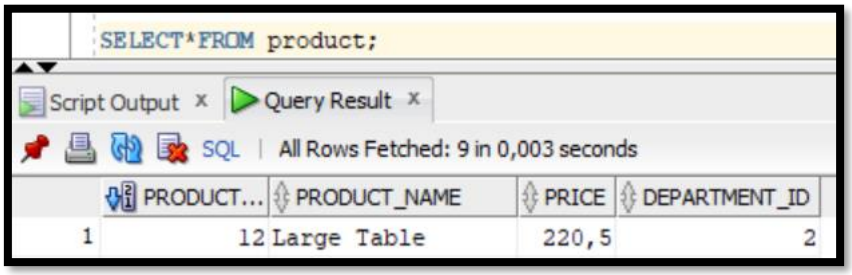
Tekshiramiz:



Rollback: saqlashdan boshlab ortga qaytarish



Tekshiramiz:



Natija: yangilanish avvalgi holatiga qaytarildi.

Truncating Data

Ma’lumotlarni qaytarib bo’lmaydigan qilib o’chirish.

Customer_order jadvalini chiqaramiz:

```
SELECT * FROM customer_order;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0,01 seconds

	ORDER_ID	CUSTOMER_ID	PRODUCT_ID	ORDER_DATE
1	1	3	2	18.12.16
2	2	5	6	01.06.15
3	3	6	5	26.09.16

TRUNCATE TABLE

```
TRUNCATE TABLE customer_order;
```

Query Result | Script Output | Task completed in 0.031 seconds

```
Table CUSTOMER_ORDER truncated.
```

Tekshiramiz:

```
SELECT * FROM customer_order;
```

Script Output | Query Result | SQL | All Rows Fetched: 0 in 0.002 seconds

ORDER_ID	CUSTOME...	PRODUCT...	ORDER_D...

ROLLBACK:

```
ROLLBACK;
```

Script Output | Query Result | Task completed in 0.001 seconds

```
Table CUSTOMER_ORDER truncated.

Rollback complete.
```

Yana tekshiramiz:

```
SELECT * FROM customer_order;
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.002 seconds

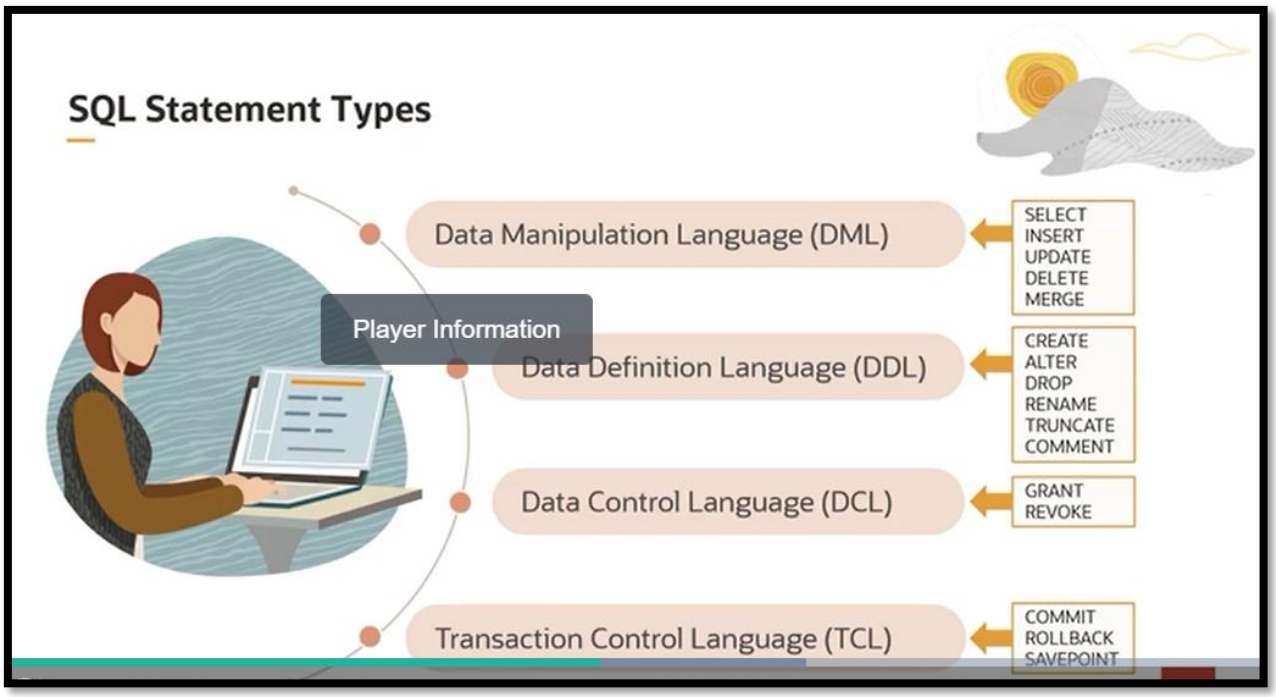
ORDER_ID	CUSTOME...	PRODUCT...	ORDER_D...

Yangilanish avvalgi holatiga qaytarilmadi.

Shuning uchun TRUNCATEni ishlatish havfli hisoblanadi.

Creating, Altering and Dropping Tables

SQL Statement Types



Create a Table

```
CREATE TABLE tablename (
colname data_type constraints
...
);
```

Ma’lumot turi va ustun nomlari 30 belgidan oshmasligi kerak.

Ma’lumot turlari:

VARCHAR2

VARCHAR2(50) — 50 belgi uchun joy ajratadi

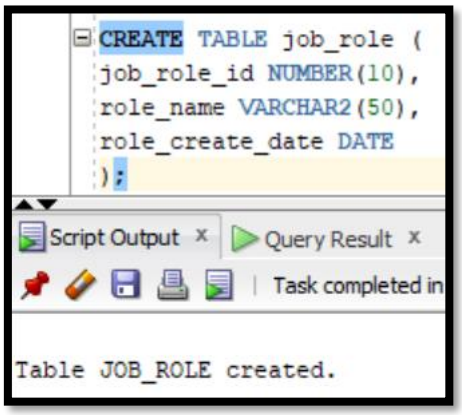
NUMBER

NUMBER(3) = XXX

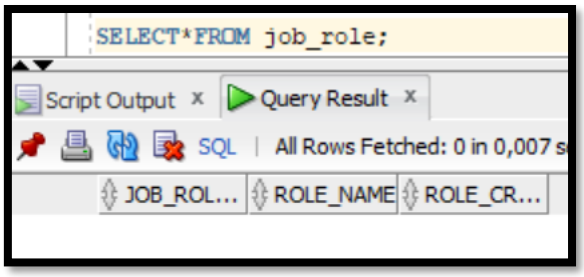
NUMBER(4,2) = XX.XX

[4– raqamlarning umumiy soni, 2 – 10 liklar soni]

Yangi jadval yaratamiz:



Tekshiramiz:



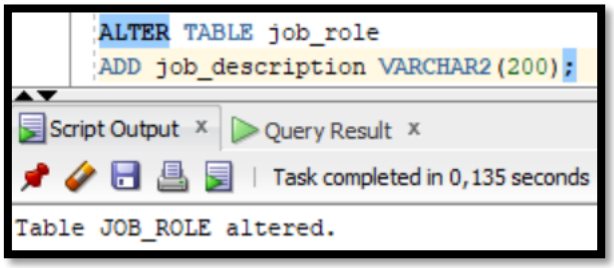
Jadval yaratildi, lekin hali hech qanday ma’lumot kiritilmagan.

Alter Table

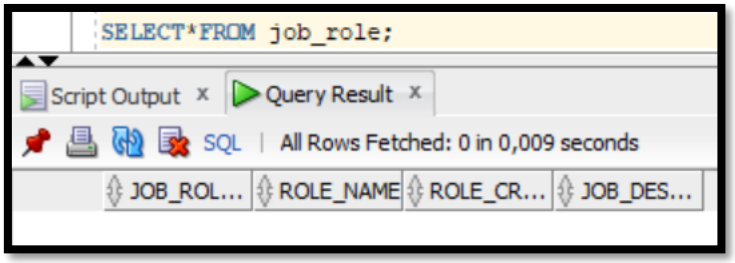
- Add coulmn
- Change data type of column
- Add constraint (cheklov)
- Remove / Rename column
- Rename a table

Add Column

Avvalgi yaratgan bazamizga ustun qo’shdik:

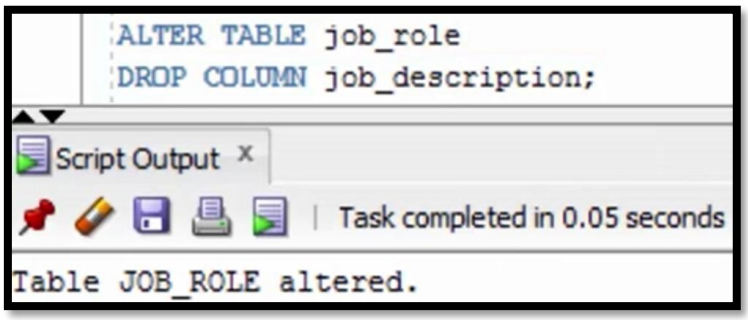


Tekshiramiz:

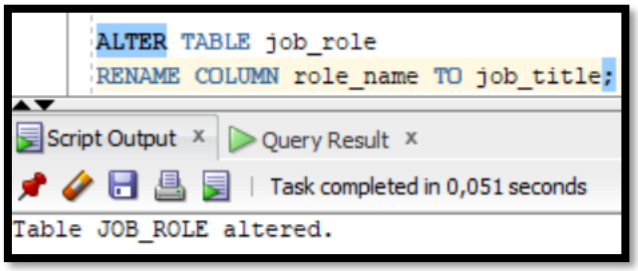


Drop Column

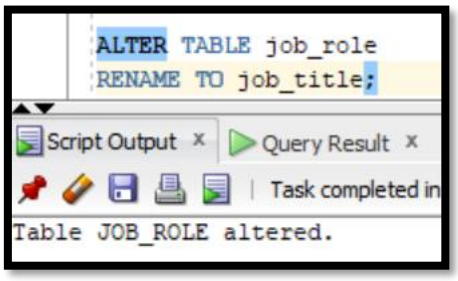
Ustunni o’chiramiz: DROP COLUMN



Rename Column



Rename Table



Drop Table

DROP – faqat obyekt uchun ishlatiladi.