



Instructor:

[**Cihan O'zhan**](#)

Researcher, Developer



Owner:

[**Azamat Nishonov**](#)

SQL Developer

Content

Introduction	4
Create Connection	4
Working with Database Object.....	4
Create Table with SQL.....	4
Creating Tables Using Table Dialog.....	5
Copy Table to Table and SQL Dialog	7
Adding a new column to the table	9
Data access.....	10
Receive Columns	10
Getting rows	11
Oracle Sql Query Builder.....	12
JOIN's Understanding	14
ON CLAUSE with the JOIN operation	15
Writing a Query That Joins Multiple Tables.....	16
Character functions.....	16
Numeric Functions	17
Date/Time Functions	17
Date Formatting	18
NVL Functions	18
Use of Group Functions.....	19
Grouped Data Reports	19
Use the HAVING clause	20
Data Processing	20
Add Single Line Data.....	20
Adding Multi-Line Data	21
Update Data	22
Delete Data	22
Excel to Export.....	23

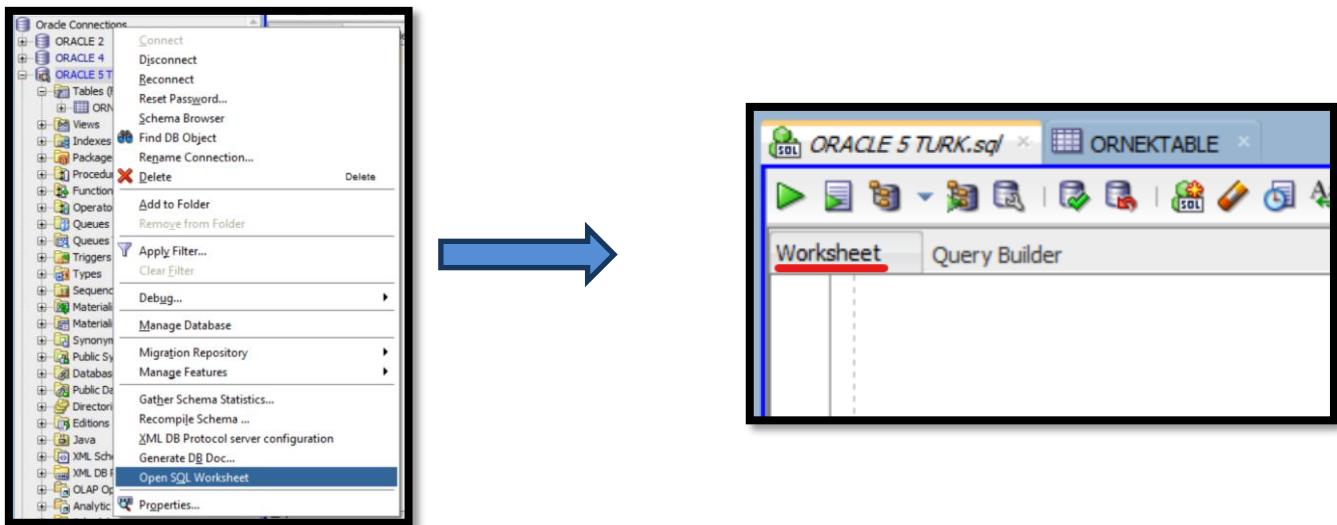
Export to DDL	24
Adding components PL/SQL	27
Create DML Trigger.....	27
Manage Trigger	28
Create Stored Procedure.....	29
Use of Function	31
SYNONYM	32
VIEW.....	34
Sequence.....	35

Introduction

Create Connection

Kod yozish stolini ochish uchun:

(baza ustiga o'ng knopka bosilib, *Open SQL Worksheet* bosiladi)

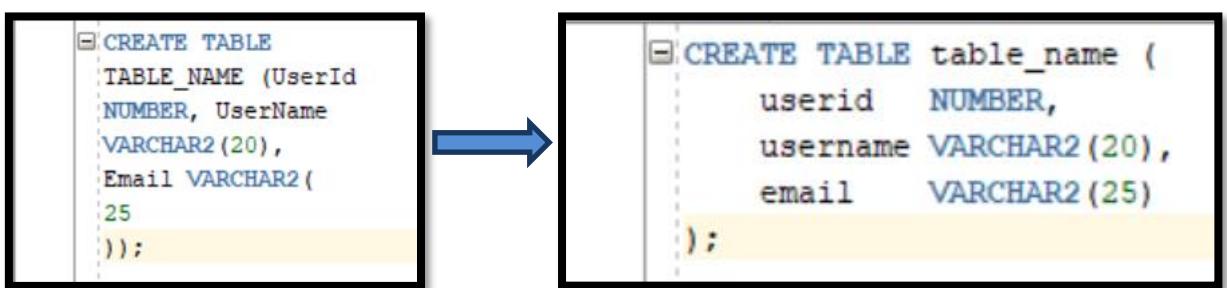


Properties qismida userning profiliga o'zgartirish kiritish mumkin.

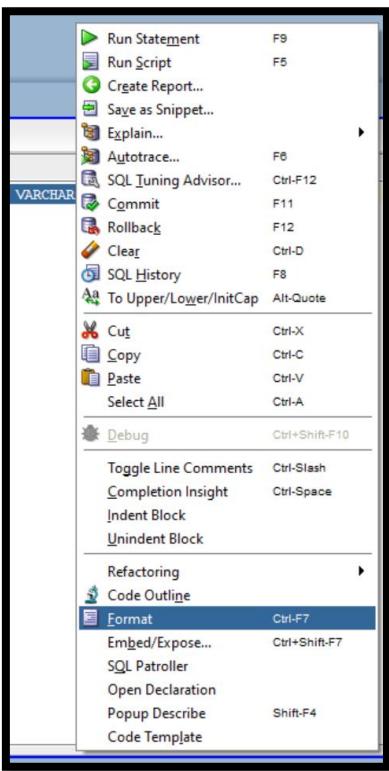
Working with Database Object

Create Table with SQL

Yozilgan kodlarning formatini to'g'irlash uchun: **Ctrl + F7**

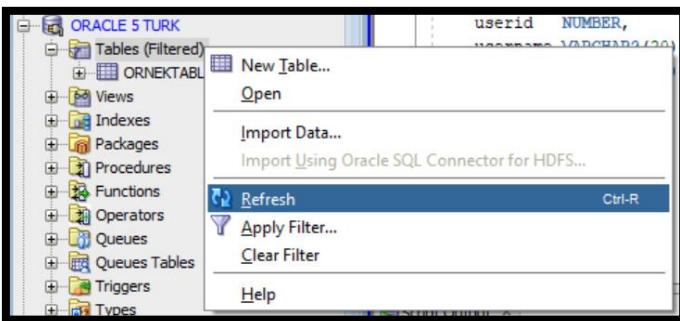


(yoki kodni ustiga o'ng knopka bosiladi va *Format* belgilanadi.)



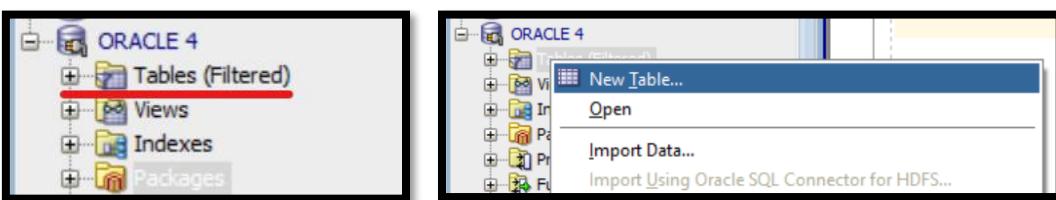
Yaratilgan jadvalni *Tablesda* ko'rish uchun:

(*Tablesga* o'ng knopka bosilib, *Refresh* belgilanadi.)



Creating Tables Using Table Dialog

Jadvalni kodsiz oson yaratish:



Tablesga mishkaning o'ng knopkasi bosilib, *New_Table* bosiladi.

Schema: SYS
Name: ORNEKTABLE

Table DDL

Columns: name

PK	Name	Data Type	Size	Not Null	Default	Comment
	USERID	NUMBER		<input checked="" type="checkbox"/>		
	USERNAME	VARCHAR2	30	<input checked="" type="checkbox"/>		
	EMAIL	VARCHAR2	60	<input checked="" type="checkbox"/>		

Jadval nomi va har xil o'lchovlari kiritiladi.

+ ustun qo'shish

X ustun o'chirish

DDL — ushbu bajarilgan operatsiyaning kodlari.

Schema: SYS
Name: ORNEKTABLE
Table Type: Normal

Search

Constraints: name

Type	Name	Enabled	Deferrable State
New Primary Key Constraint			
New Unique Constraint			
New Foreign Key Constraint			
New Check Constraint			

Constraints => cheklovlar o'rnatish

Constraints: name

Type	Name	Enabled	Deferrable State
Primary Key	ORNEKTABLE_PK	<input checked="" type="checkbox"/>	Not Deferrable

Using Index: <DEFAULT>

Available Columns:

- USERID
- USERNAME
- EMAIL

Selected Columns:

- USERID

Constraints: name

Type	Name	Enabled	Deferrable State
Primary Key	ORNEKTABLE_PK	<input checked="" type="checkbox"/>	Not Deferrable

Using Index: <DEFAULT>

Available Columns:

- USERNAME
- EMAIL

Selected Columns:

- USERNAME

Jadval haqidagi ma'lumotlar:

The screenshot shows the Oracle SQL Developer interface with the title bar "ORACLE 5 TURK.sql" and "ORNEKTABLE". The "Columns" tab is selected in the top navigation bar. Below is a table with the following data:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 USERID	NUMBER	No	(null)	1	UserID ustuni
2 USERNAME	VARCHAR2 (30 BYTE)	No	(null)	2	(null)
3 EMAIL	VARCHAR2 (60 BYTE)	No	(null)	3	(null)

Columns → Ustunlar

Data → Ma'lumotlar

Model → Jadval struktura modeli

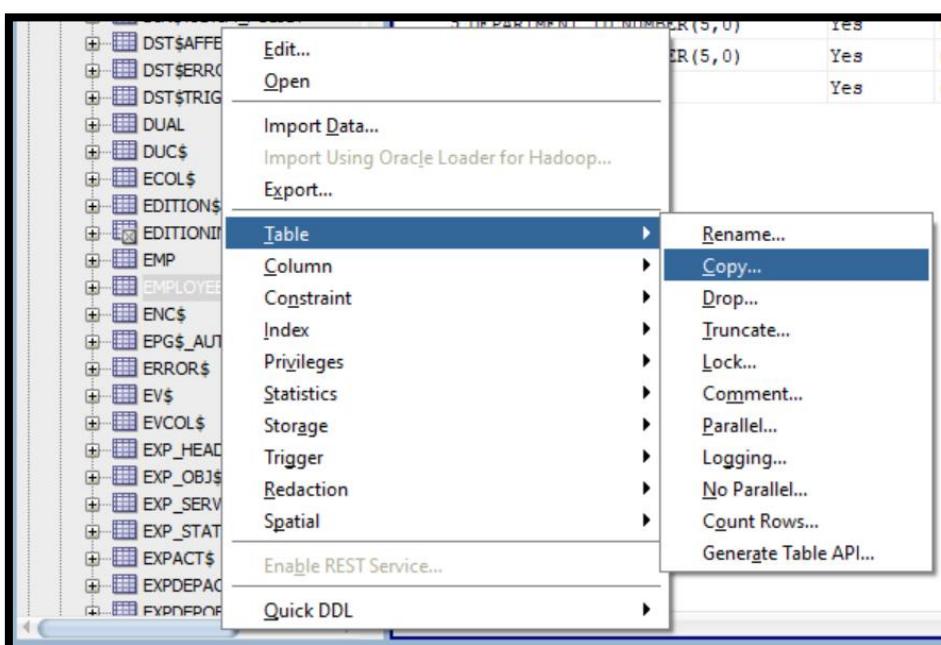
Constraints → Cheklovlar

Grants → Imtiyozlar (ruxsatlar)

Copy Table to Table and SQL Dialog

Jadvaldan nusxa olish:

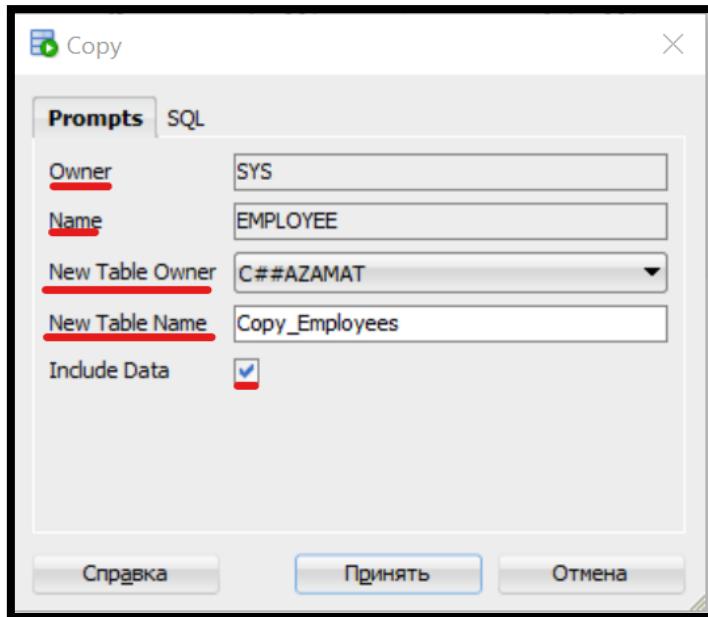
Kerakli jadval ustiga o'ng knopka bosiladi → Table → Copy



Owner ⇔ Egasi (baza nomi)

New Table Name ⇔ Yangi jadval nomi

Include Data ⇔ Ma'lumoti bilan qo'shib nusxa olish



Baza ichidagi jadvaldan nusxa olish kodi:

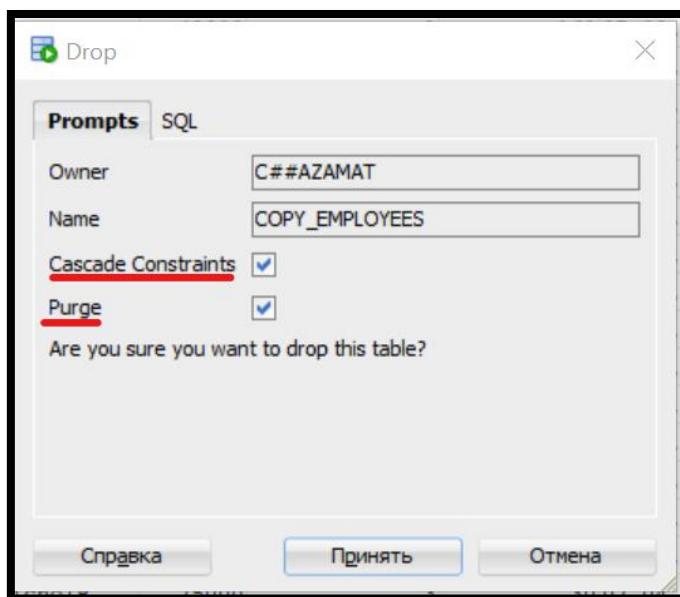
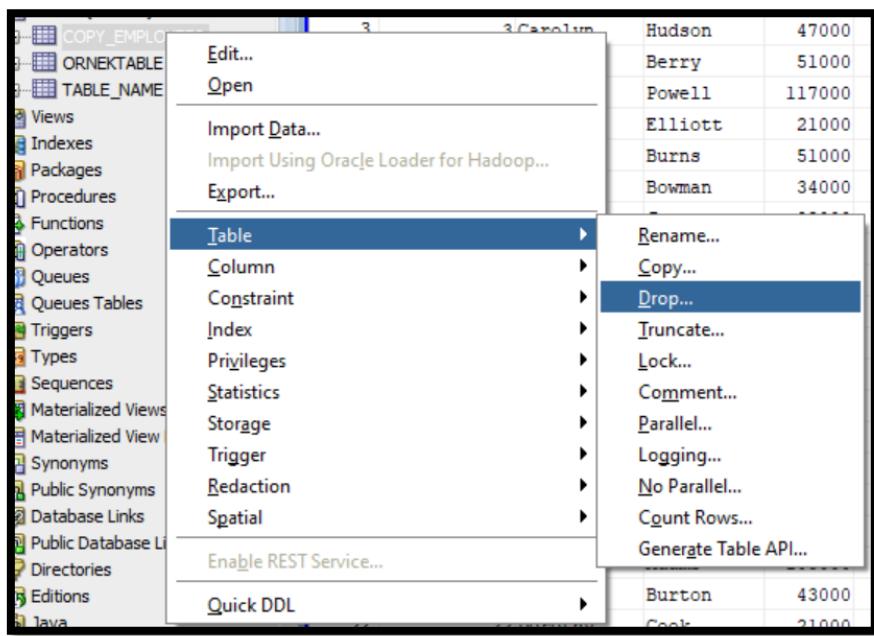
```
create table COPY_EMPLOYEES as
select * from employee;
```

Script Output | Task completed in 0,073 sec

Table COPY_EMPLOYEES created.

Jadvalni o'chirish:

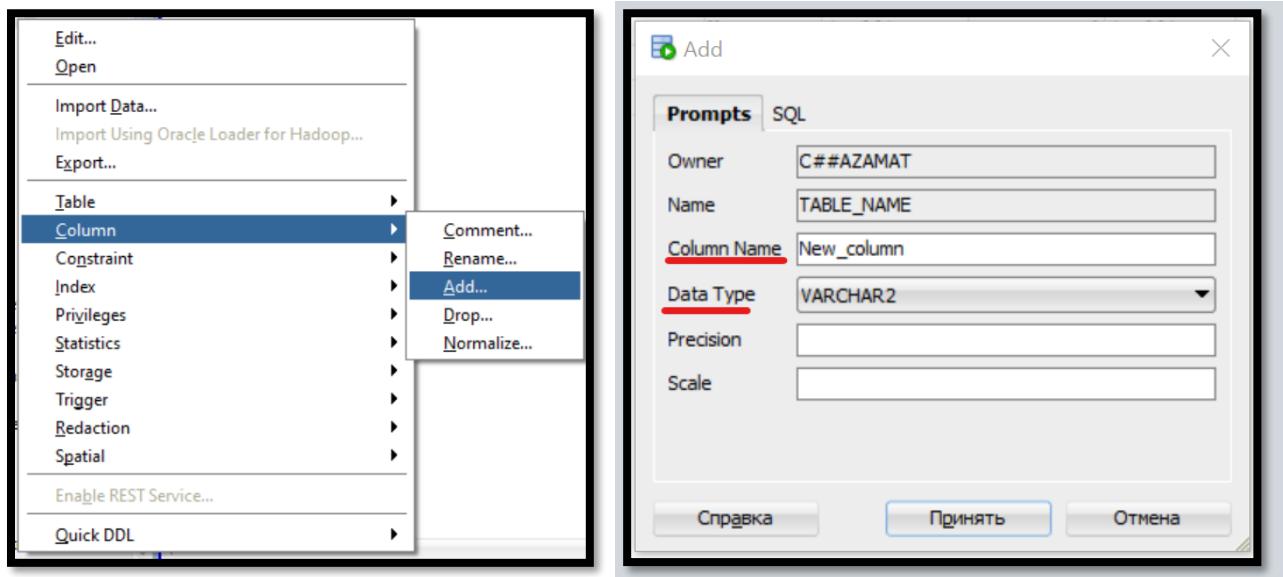
Kerakli jadval ustiga o'ng knopka bosiladi → Table → Drop



Adding a new column to the table

Yangi ustun qo'shish:

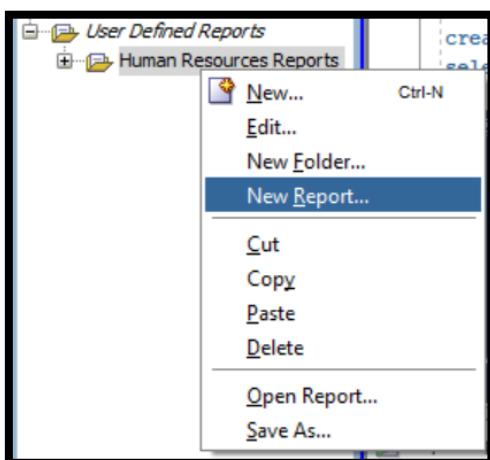
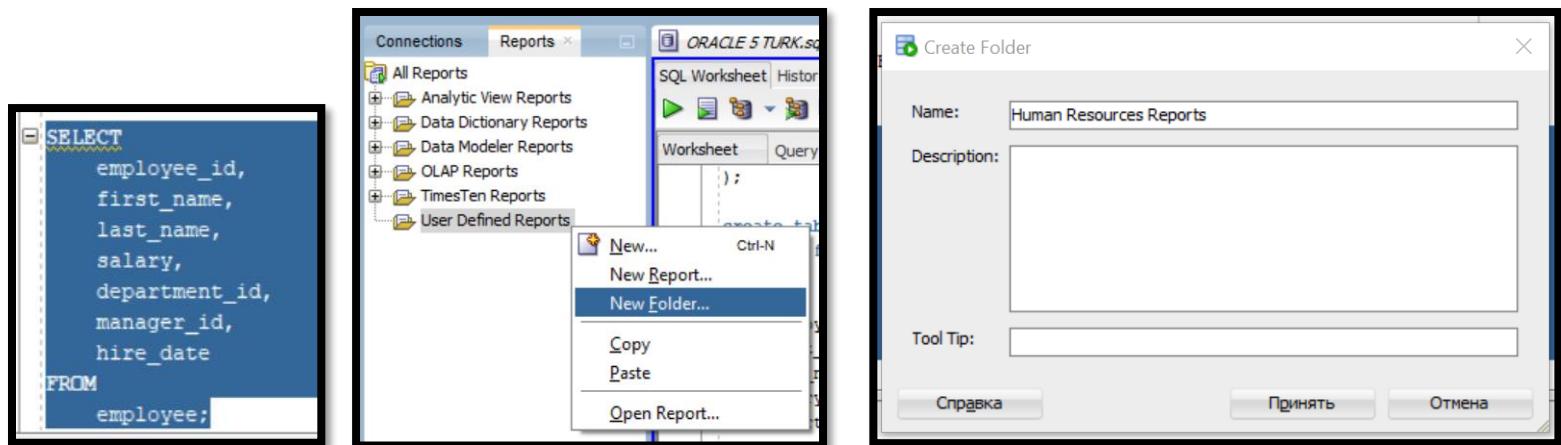
Kerakli jadvalga o'ng knopka bosiladi → Column → Add

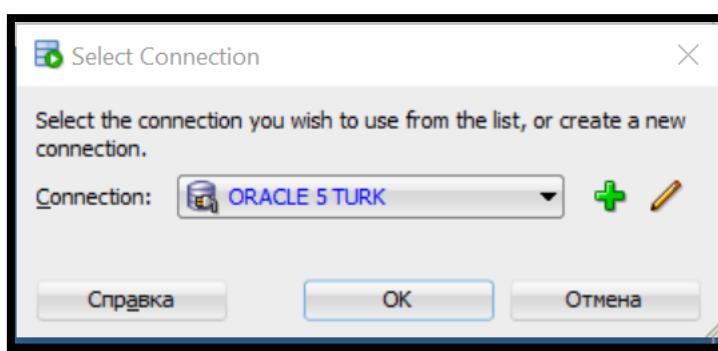
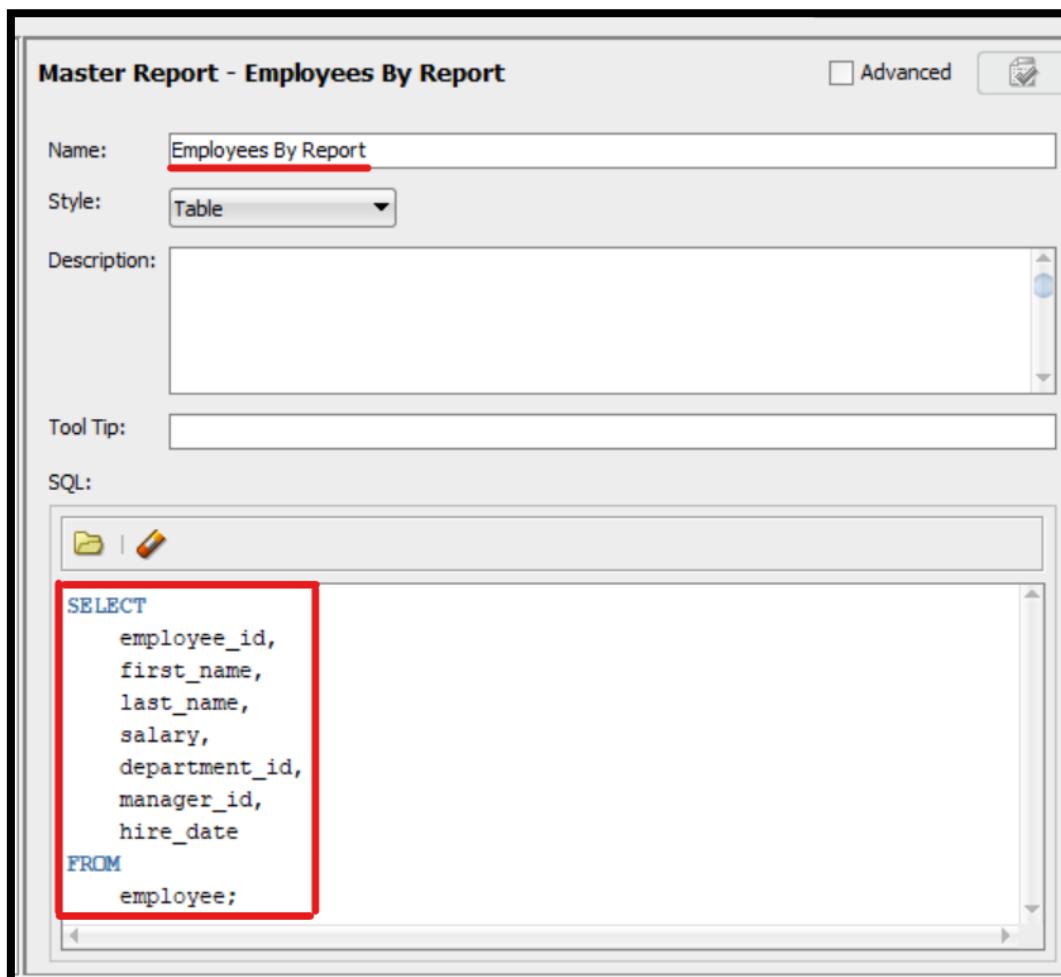


Data access

Receive Columns

Kerakli kod copy qilinadi:





Getting rows

O'zimiz istagan qiymatni tanlab chiqarish: { VALUE or NULL }

```

SELECT
    employee_id,
    first_name,
    last_name,
    salary,
    department_id,
    manager_id,
    hire_date
FROM
    employee
WHERE Department_id = :deptno
    AND Manager_id = :mngrno;

```

deptno → departmenet_id nomerini kiritishni anglatadi.

mngrno → manager_id nomerini kiritishni anglatadi.

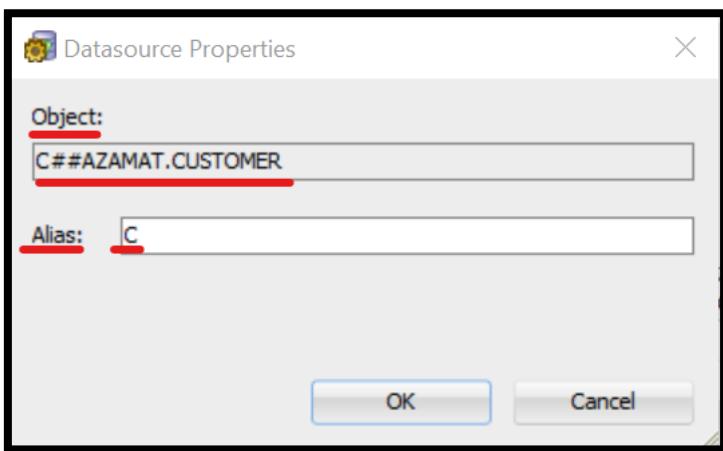
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	MANAGER_ID	HIRE_DATE
1	45 Lori	Jacobs	69000	2	114	15.03.12
2	53 Cynthia	Medina	118000	2	114	16.06.13
3	92 Ruby	Clark	35000	2	114	26.10.16

Oracle Sql Query Builder

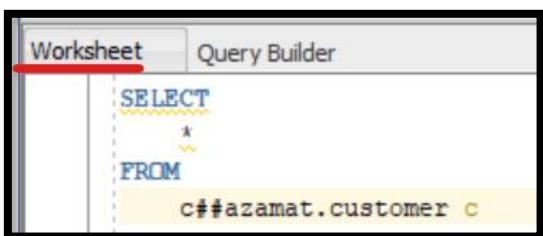
Query builder → kod yozib beruvchi:

{ Kerakli jadval tanlanib, o'ng tarafga olib o'tiladi }

Sarlavhadagi CUSTOMERni 2 marta bosib, unga taxallus qo'yamiz:



Query builder tuzgan kod:

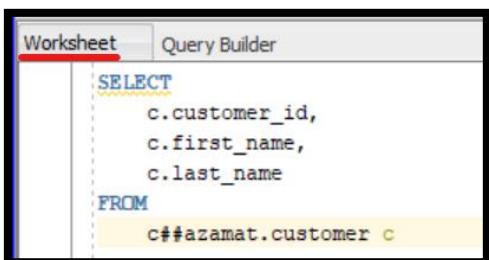


Kerakli ustunlarni tanlab chiqarish va shartlar kiritish:

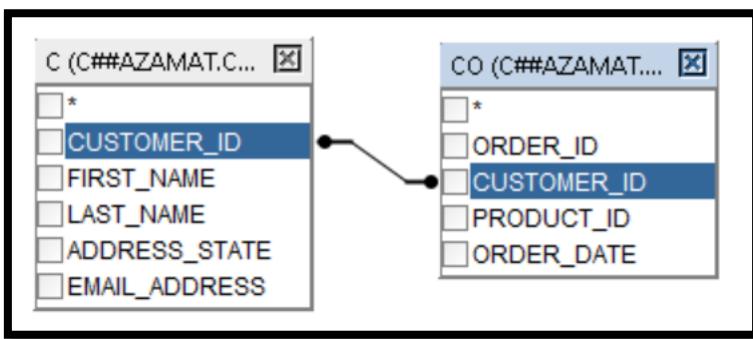
C (C##AZAMAT.C...)

Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or
<input checked="" type="checkbox"/>	C.CUSTOMER_ID					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	C.FIRST_NAME					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	C.LAST_NAME					<input type="checkbox"/>		
<input type="checkbox"/>						<input type="checkbox"/>		

Hosil qilingan kod:

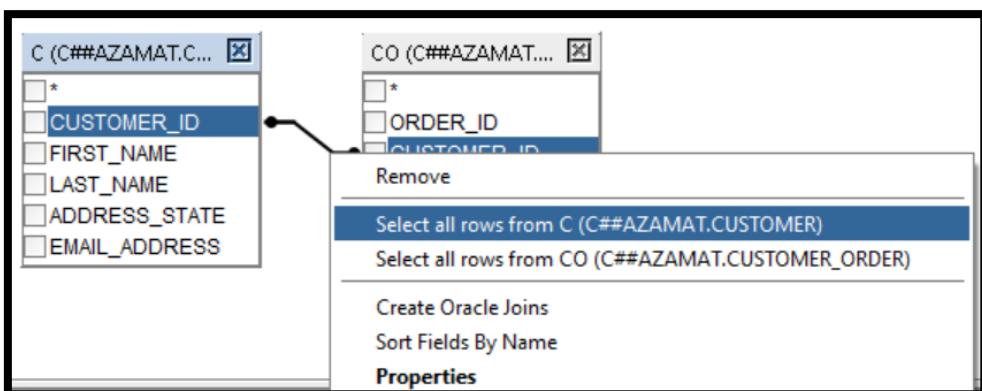


Inner Join qilish: { 1-ustunni ushlab, 2-ustun ustiga tashlaymiz }



Left Join, Right Join va Full Join qilish:

- Left Join → Select all rows from C
- Right Join → Select all rows from CO
- Full Join → ikkovi ham belgilanadi



JOIN's Understanding

Jadvallarni mixlab qo'yish uchun:

The screenshot shows the Oracle Database SQL Developer interface. Two tables are open: 'CUSTOMER' and 'CUSTOMER_ORDER'. The 'CUSTOMER' table has columns: COLUMN_NAME (1 CUSTOMER_ID, 2 FIRST_NAME, 3 LAST_NAME) and DATA_TYPE (NUMBER(5,0), VARCHAR2(100 BYTE), VARCHAR2(100 BYTE)). The 'CUSTOMER_ORDER' table has columns: COLUMN_NAME and DATA_TYPE. The 'Actions...' button is visible at the bottom of the interface.

Natural Join:

```

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

```

Query Result | All Rows Fetched: 9 in 0,008 seconds

	PRODUCT_ID	PRODUCT_NAME	PRICE	DEPARTMENT_ID	DEPARTMENT_NAME
1	12	Large Table	220,5	2	Sales
2	1	Monitor	149,95	4	Hardware Development
3	2	Desk	110,9	4	Hardware Development

ON CLAUSE with the JOIN operation

JOIN operatorinig ON bandi

```

SELECT
    E.employee_id,
    E.first_name,
    E.hire_date,
    D.department_name
FROM employee E
    JOIN department D
        ON e.department_id = d.department_id;

```

Query Result | Fetched 50 rows in 0,011 seconds

	EMPLOYEE_ID	FIRST_NAME	HIRE_DATE	DEPARTMENT_NAME
1	1	Michelle	27.08.11	Legal
2	2	Cheryl	02.01.12	Customer Support
3	3	Carolyn	04.12.16	Finance

Writing a Query That Joins Multiple Tables

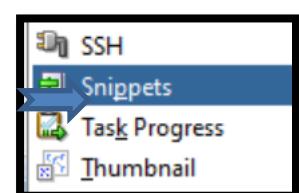
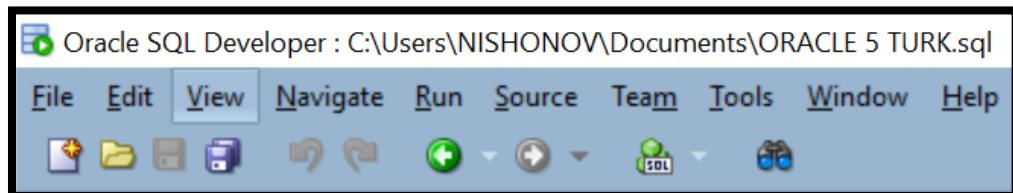
```
SELECT
    co.order_id,
    c.first_name,
    p.product_name,
    p.price,
    co.order_date
FROM
    customer_order co
INNER JOIN customer c ON c.customer_id = co.customer_id
INNER JOIN product p ON p.product_id = co.product_id
ORDER BY c.first_name;
```

Query Result | Fetched 50 rows in 0,014 seconds

ORDER_ID	FIRST_NAME	PRODUCT_NAME	PRICE	ORDER_DATE
1	Dorothy	Microsoft Office	121,15	26.09.16
2	Dorothy	Web Tools 4	202,5	27.06.15
3	Dorothy	Desk	110,9	06.05.15

Character functions

Functions ga kirish: View → Snippets



Snippets

Aggregate Functions

Character Functions

Conversion Functions

Date Formats

Date/Time Functions

Flashback

Number Formats

Character Functions

- CHR(n)
- CONCAT(char1, char2)
- INITCAP(char)
- LOWER(char)
- LPAD(expr1, n, expr2)
- LTRIM(char, set)
- NLS_SORT(char, 'NLS_SORT = lang')
- NLS_INITCAP(char, 'NLS_SORT = lang')
- NLS_LOWER(char, 'NLS_SORT = lang')
- NLS_UPPER(char, 'NLS_SORT = lang')
- RPAD(expr1, n, expr2)
- RTRIM(char, set)
- SOUNDEX(char)
- SUBSTR(char, position, substring_length)
- TRANSLATE(expr, from_string, to_string)
- TRIM([LEADING | TRAILING | BOTH] trim_)
- UPPER(char)

...sql ORACLE ...

Worksheet Query Builder

```
SELECT
  FROM employee;
```

Worksheet Query Builder

```
SELECT
  CONCAT(char1, char2)
  FROM employee;
```

Numeric Functions

Numeric Functions

- ABS(n)
- CEIL(n)
- EXP(n)
- FLOOR(n)
- LOG(n2, n1)
- MOD(n2, n1)
- POWER(n2, n1)
- REMAINDER(n2, n1)
- ROUND(n, integer)
- TRUNC(n1, n2)

```
SELECT
    first_name,
    salary,
    round(salary, - 4)
FROM
    employee;
```

Query Result x

SQL | Fetched 50 rows in 0,038 seconds

	FIRST_NAME	SALARY	ROUND(SALARY, -4)
1	Michelle	48000	50000
2	Cheryl	79000	80000
3	Carolyn	47000	50000

Date/Time Functions

Kunlar farqini hisoblash:

Date/Time Functions

- ADD_MONTHS(date, integer)
- CURRENT_DATE
- CURRENT_TIMESTAMP
- MONTHS_BETWEEN(date, date)
- SYSDATE
- SYSTIMESTAMP
- TO_CHAR(datetime, fmt, nlsparam)
- TO_CHAR(sysdate, 'DD-MON-YYYY HH24:MM')
- TO_TIMESTAMP(char, fmt, nlsparam)
- TRUNC(date, fmt)

```
SELECT
    first_name,
    sysdate,
    hire_date,
    round(sysdate - hire_date) "kunlar farqi"
FROM
    employee;
```

Query Result x

SQL | Fetched 50 rows in 0,006 seconds

	FIRST_NAME	SYSDATE	HIRE_DATE	kunlar farqi
1	Michelle	04.04.23	27.08.11	4238
2	Cheryl	04.04.23	02.01.12	4110
3	Carolyn	04.04.23	04.12.16	2312

Necha yil ishlaganini hisoblash:

```

SELECT
    first_name,
    round(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)) years_worked
FROM employee;

```

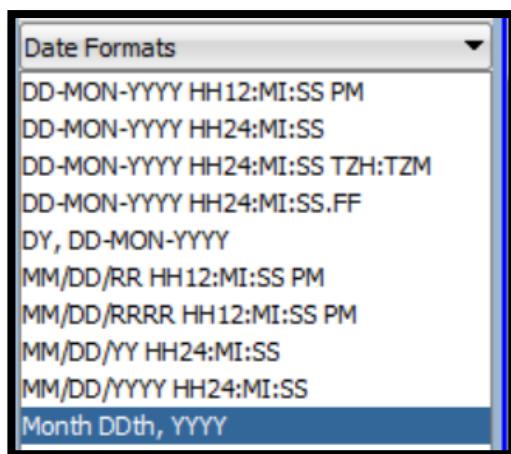
Query Result

Fetched 50 rows in 0,007 seconds

FIRST_NAME	kunlar farqi	YEARS_WORKED
1 Michelle	4238	139
2 Cheryl	4110	135
3 Carolyn	2312	76

Date Formatting

Ishga kirish sanasini belgilar bilan chiqarish:



```

SELECT
    first_name,
    TO_CHAR(hire_date, 'DY, DD-MON-YYYY') start_day
FROM employee;

```

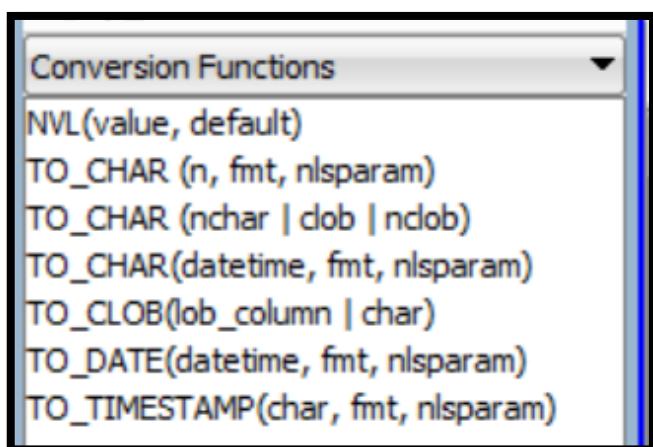
Query Result

Fetched 50 rows in 0,008 seconds

FIRST_NAME	START_DAY
1 Michelle	СБ, 27-АВГ-2011
2 Cheryl	ПН, 02-ЯНВ-2012
3 Carolyn	ВС, 04-ДЕК-2016

NVL Functions

Conversion Functions → NVL Function



```

SELECT
    first_name,
    NVL(TO_CHAR(Manager_id), 'Manageri yo`q') MANAGER
FROM employee;

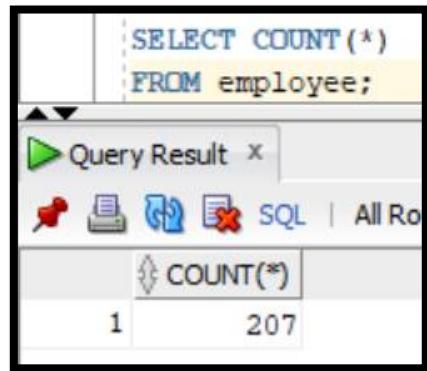
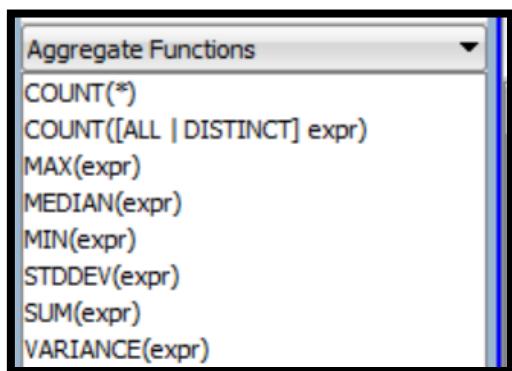
```

Query Result

Fetched 50 rows in 0,006 seconds

FIRST_NAME	MANAGER
4 Patrick	159
5 Doris	Manageri yo`q
6 Jessica	70

Use of Group Functions



Grouped Data Reports

Guruhlangan ma'lumotlar hisobotlari:

SELECT department_id, SUM(salary) FROM employee GROUP BY department_id ORDER BY SUM(salary);		
Query Result x		
		All Rows Fetched: 9 in
DEPARTMENT_ID		SUM(SALARY)
1	(null)	212000
2	1	305000
3	4	1248000

SELECT department_id, manager_id, SUM(salary) FROM employee GROUP BY department_id, manager_id ORDER BY department_id;			
Query Result x			
	All Rows Fetched: 9 in	Fetched 50 rows in 0,012 seconds	
DEPARTMENT_ID	MANAGER_ID		SUM(SALARY)
1	1	5	188000
2	1	(null)	117000
3	2	3	34000

Use the HAVING clause

The screenshot shows a SQL query editor with the following code:

```
SELECT
    department_id,
    MAX(salary)
FROM
    employee
GROUP BY
    department_id
HAVING
    MAX(salary) > 110000;
```

Below the code is a "Query Result" window with the following data:

DEPARTMENT_ID	MAX(SALARY)
1	120000
2	120000
3	114000

Data Processing

Add Single Line Data

Ma'lumotlarni kodsiz kiritish: { Ctrl +I }

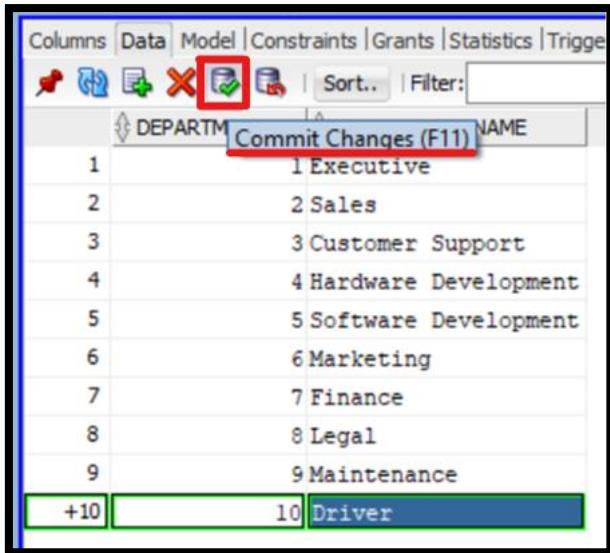
The screenshot shows a table editor for the DEPARTMENT table. The table has two columns: DEPARTMENT_ID and DEPARTMENT_NAME. The rows are numbered 1 to 9. A red box highlights the "Insert Row (Ctrl+I)" button in the toolbar.

DEPARTMENT_ID	DEPARTMENT_NAME
1	Executive
2	Sales
3	Customer Support
4	Hardware Development
5	Software Development
6	Marketing
7	Finance
8	Legal
9	Maintenance

The screenshot shows the same table editor after a new row has been added. The new row is at the bottom of the table, numbered 10, with the values "+10" in the ID column and "Driver" in the name column. A green box highlights the last row.

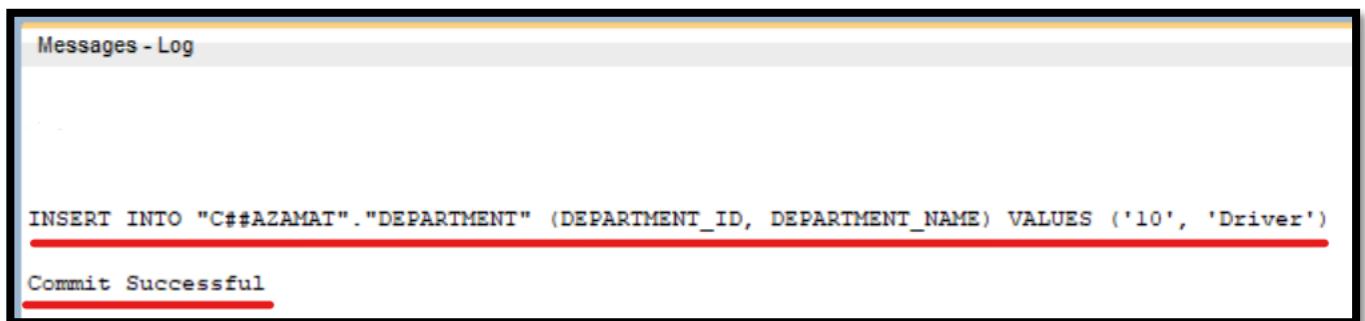
DEPARTMENT_ID	DEPARTMENT_NAME
1	Executive
2	Sales
3	Customer Support
4	Hardware Development
5	Software Development
6	Marketing
7	Finance
8	Legal
9	Maintenance
+10	Driver

O'zgartirishni soxranit qilish: COMMIT { F11 }



A screenshot of the Oracle SQL Developer interface. The main window shows a table named 'DEPARTMENT'. The last row, which has been added ('+10') and has a value '10' in the 'DEPARTMENT_ID' column and 'Driver' in the 'DEPARTMENT_NAME' column, is selected. The status bar at the bottom right shows 'Commit Changes (F11)'. The toolbar above the table has several icons, with the commit icon (a green checkmark inside a box) highlighted by a red box.

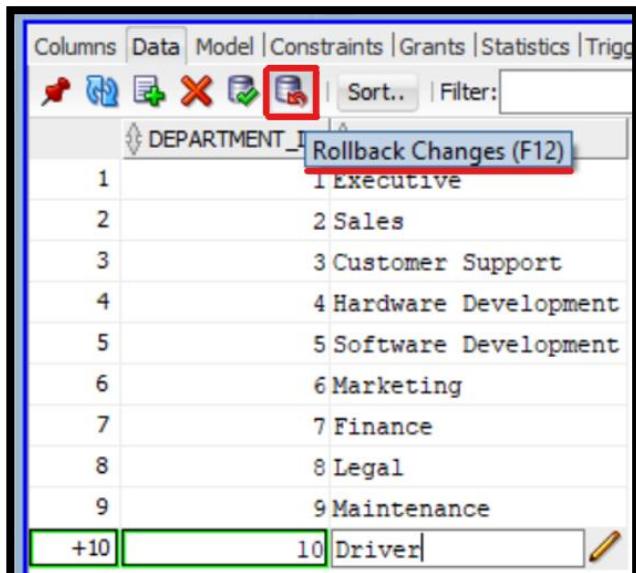
DEPARTMENT_ID	DEPARTMENT_NAME
1	Executive
2	Sales
3	Customer Support
4	Hardware Development
5	Software Development
6	Marketing
7	Finance
8	Legal
9	Maintenance
+10	Driver



The screenshot shows the 'Messages - Log' window in Oracle SQL Developer. It displays two lines of text: 'INSERT INTO "C##AZAMAT"."DEPARTMENT" (DEPARTMENT_ID, DEPARTMENT_NAME) VALUES ('10', 'Driver')' and 'Commit Successful'. Both lines are highlighted with a red underline, indicating they were successfully executed.

```
INSERT INTO "C##AZAMAT"."DEPARTMENT" (DEPARTMENT_ID, DEPARTMENT_NAME) VALUES ('10', 'Driver')
Commit Successful
```

O'zgartirishni bekor qilish: ROLLBACK { F12 }



A screenshot of the Oracle SQL Developer interface, similar to the previous one but showing the result of a rollback. The table 'DEPARTMENT' now has 10 rows, including the original 9 and the new row with ID 10 and name 'Driver'. The status bar at the bottom right shows 'Rollback Changes (F12)'. The toolbar above the table has several icons, with the rollback icon (a red circle with a cross) highlighted by a red box.

DEPARTMENT_ID	DEPARTMENT_NAME
1	Executive
2	Sales
3	Customer Support
4	Hardware Development
5	Software Development
6	Marketing
7	Finance
8	Legal
9	Maintenance
+10	Driver

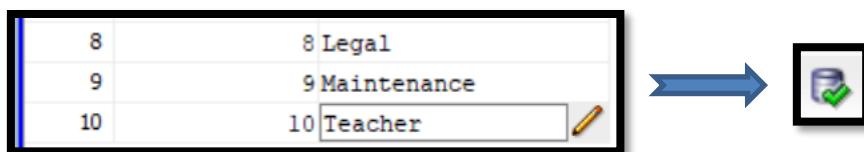
Adding Multi-Line Data

Barcha kodni birdaniga ishlatalish: Run Script → { F5 }

```
INSERT INTO department (department_id, department_name) VALUES (11, 'Coder');  
INSERT INTO department (department_id, department_name) VALUES (12, 'Coder assistant');
```

Update Data

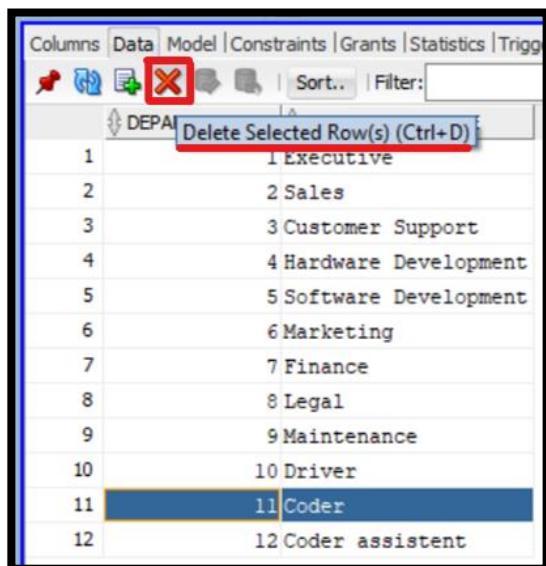
O'zgartirish kiritamiz + Commit:



```
Messages - Log  
  
UPDATE "C##AZAMAT"."DEPARTMENT" SET DEPARTMENT_NAME = 'Teacher' WHERE ROWID = 'AAATdMAAHAAAFA1AAB' AND ORA_ROWSCN = '9789240'  
  
Commit Successful
```

Delete Data

Kodsiz ma'lumotli qatorni o'chirish: { **Ctrl + D** }



```
Messages - Log  
  
DELETE FROM "C##AZAMAT"."DEPARTMENT" WHERE ROWID = 'AAATdMAAHAAAFA1AAA' AND ORA_ROWSCN = '9790255' and ( "DEPARTMENT_ID" is null or "DEPARTMENT_ID" is not null )  
  
Commit Successful
```

Filter

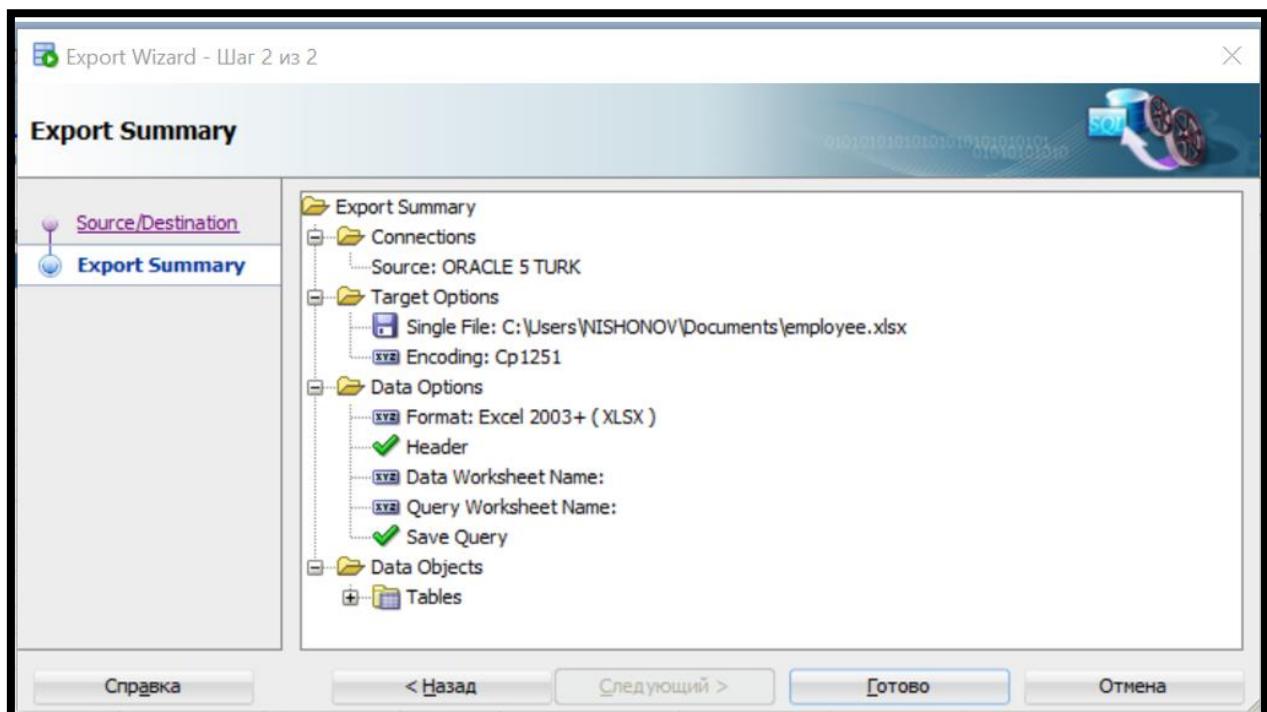
A screenshot of the Oracle SQL Developer interface. The top menu bar shows 'Columns', 'Data', 'Model', 'Constraints', 'Grants', 'Statistics', 'Triggers', 'Flashback', 'Dependencies', 'Details', 'Partitions', 'Indexes', and 'SQL'. Below the menu is a toolbar with icons for edit, insert, delete, copy, and search. A 'Sort...' button is followed by a 'Filter:' button with the value 'department_id = 6' highlighted in red. The main area displays a table with columns 'DEPARTMENT_ID' and 'DEPARTMENT_NAME'. A single row is visible, showing '1' in the first column and 'Marketing' in the second column.

Excel to Export

Data ustiga o'ng knopka bosiladi → Export

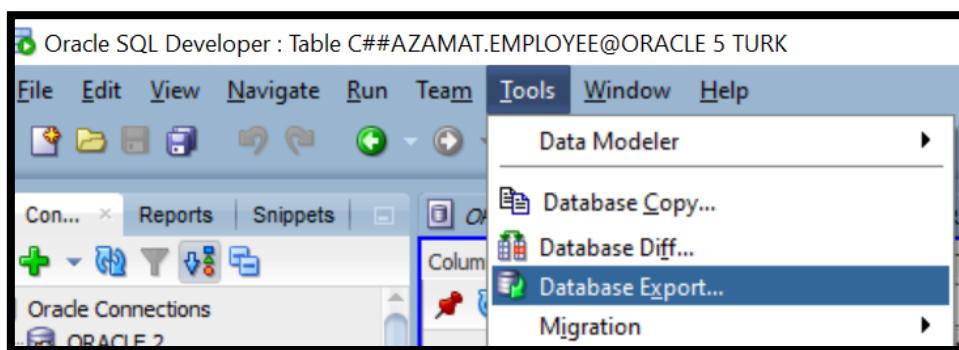
This screenshot shows two windows from Oracle SQL Developer. On the left, a 'Query Result' window displays the output of the query 'SELECT * FROM employee;'. A context menu is open over the first row, listing options: 'Save Grid as Report...', 'Publish to REST', 'Single Record View...', 'Count Rows...', 'Find/Highlight...', and 'Export...'. The 'Export...' option is highlighted with a blue background. On the right, another window shows a table of employee data. A context menu is open over the first row, listing options: 'Save Grid as Report...', 'Publish to REST', 'Single Record View...', 'Count Rows...', 'Find/Highlight...', 'Duplicate Row', 'Refresh', 'Insert Row', 'Delete Selected Row(s)', 'Commit Changes', 'Rollback Changes', and 'Export...'. The 'Export...' option is also highlighted with a blue background.

A screenshot of the 'Export Wizard - Шаг 1 из 2' (Step 1 of 2) dialog box. The title bar says 'Export Wizard - Шаг 1 из 2'. The main area is titled 'Source/Destination'. It shows a connection dropdown set to 'ORACLE 5 TURK'. Under the 'Export Data' section, the 'Format:' dropdown is set to 'csv', 'Header' is checked, and 'Line Terminator:' is set to 'environment default'. The 'Save As' section shows 'Single File' selected, 'Encoding:' set to 'Cp1251', and the 'File:' field containing the path 'C:\Users\NISHONOV\Documents\export.csv'. A 'Browse...' button is next to the file path. At the bottom, there is a checkbox 'Proceed to summary.' and navigation buttons: 'Справка' (Help), '< Назад' (Back), 'Следующий >' (Next), 'Готово' (Done), and 'Отмена' (Cancel).



Export to DDL

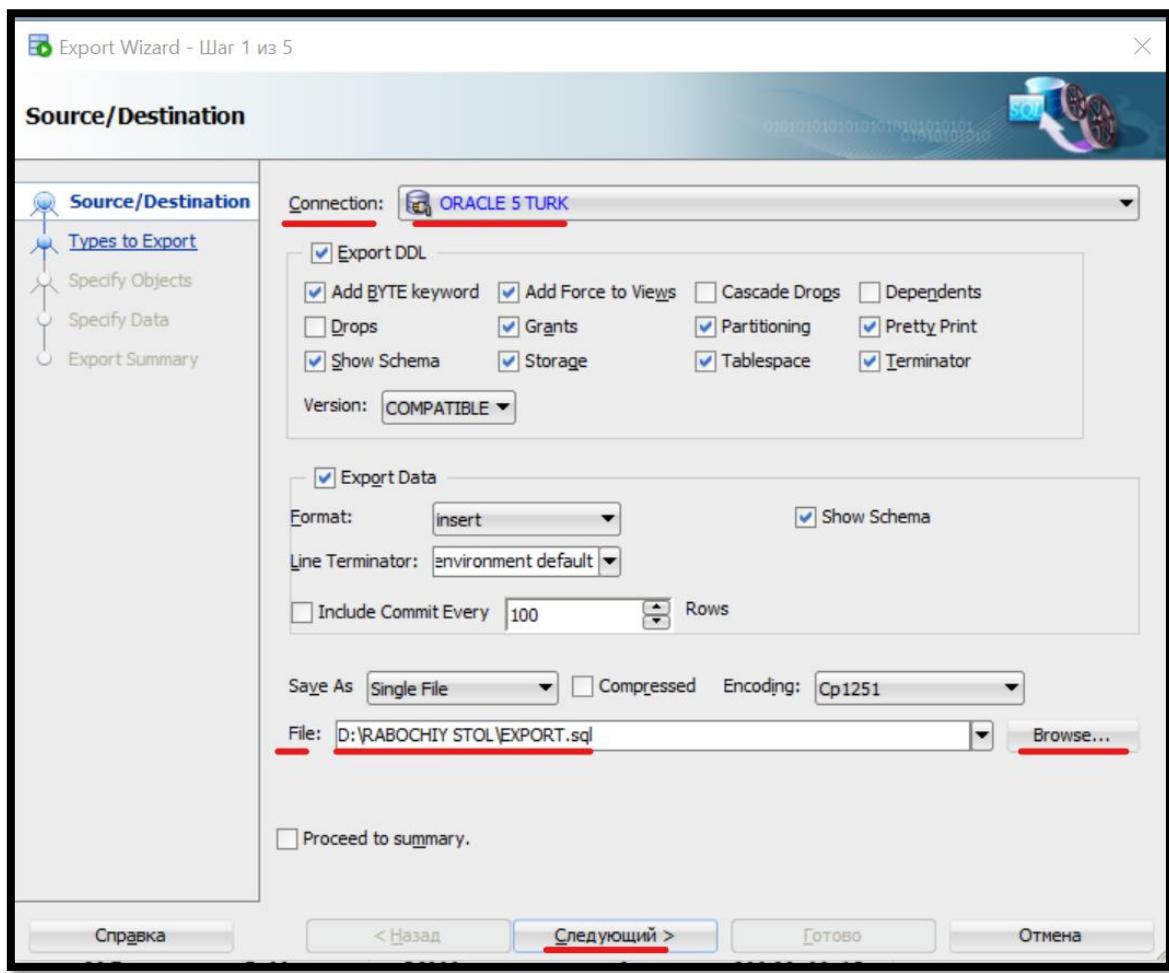
Tools → Database Export



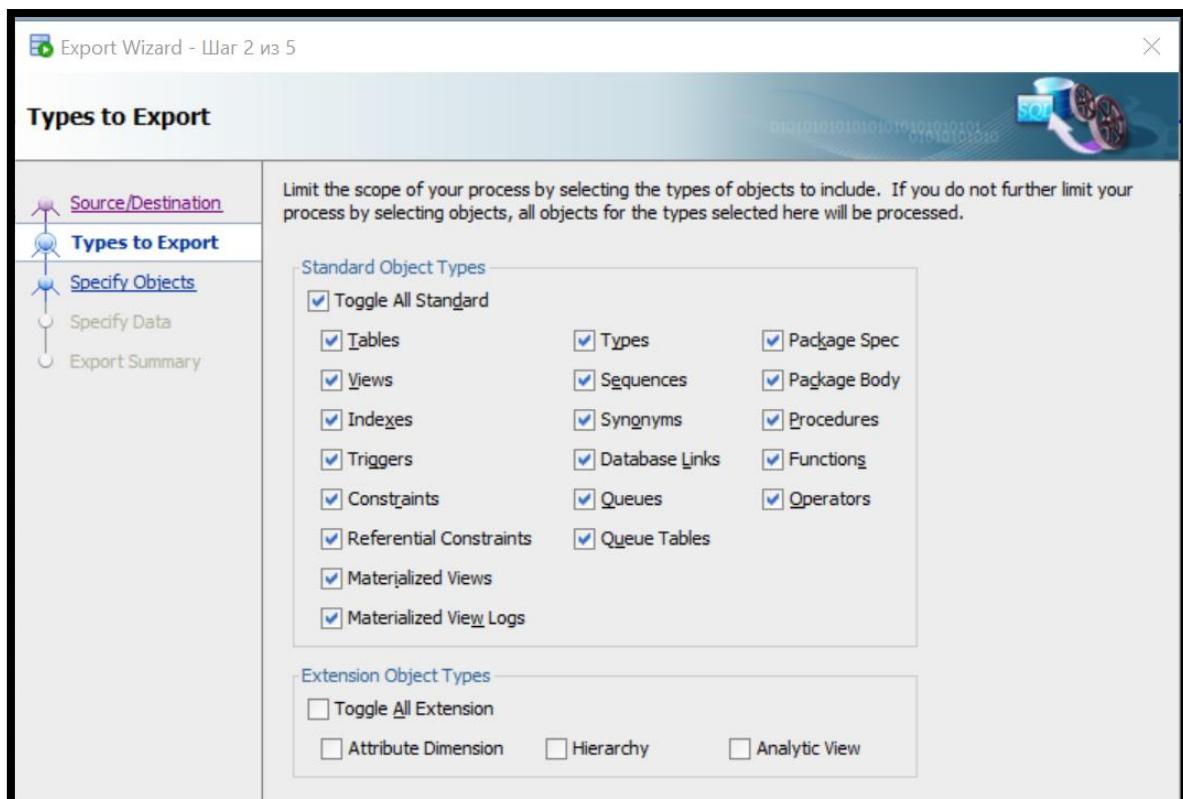
Source / Destination ⇔ Manba / Manzili

Connection ⇔ baza nomi

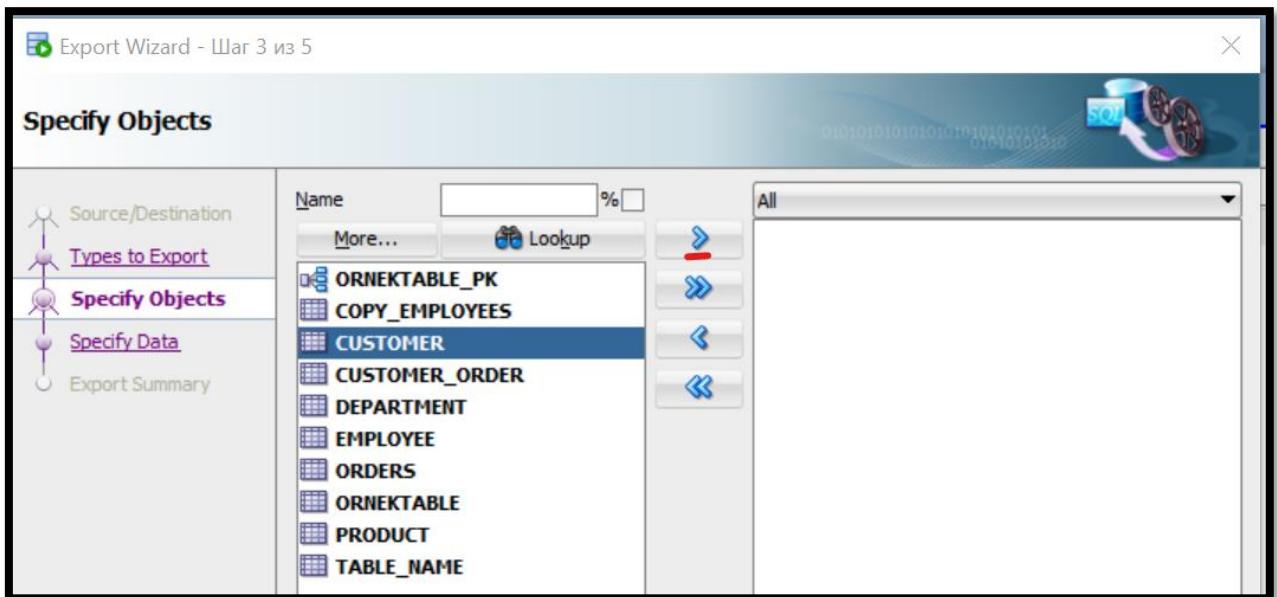
File ⇔ joylashuv joyi



Types to Export ⇔ Export qilinadigan turlar



Spesify Objects ⇔ Obyektlarni belgilash:



Tayyor.

```
-- File created - вторник-апреля-04-2023

-- DDL for Table CUSTOMER

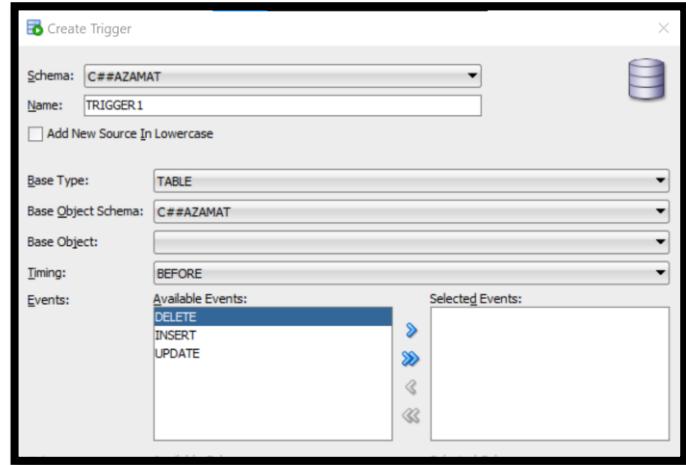
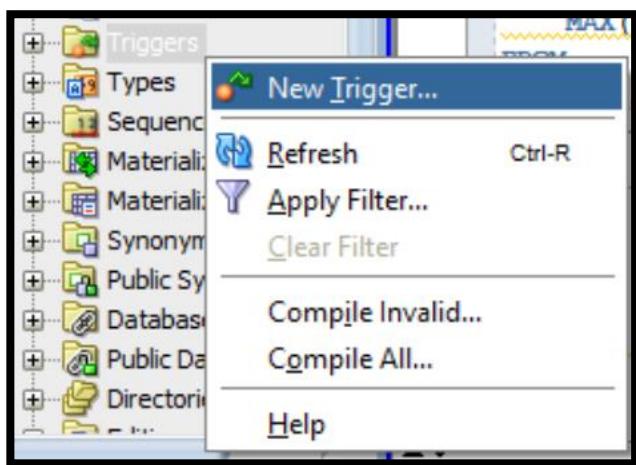
CREATE TABLE "C##AZAMAT"."CUSTOMER"
(
    "CUSTOMER_ID" NUMBER(5,0),
    "FIRST_NAME" VARCHAR2(100 BYTE),
    "LAST_NAME" VARCHAR2(100 BYTE),
    "ADDRESS_STATE" VARCHAR2(10 BYTE),
    "EMAIL_ADDRESS" VARCHAR2(350 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

-- DDL for Table DEPARTMENT
```

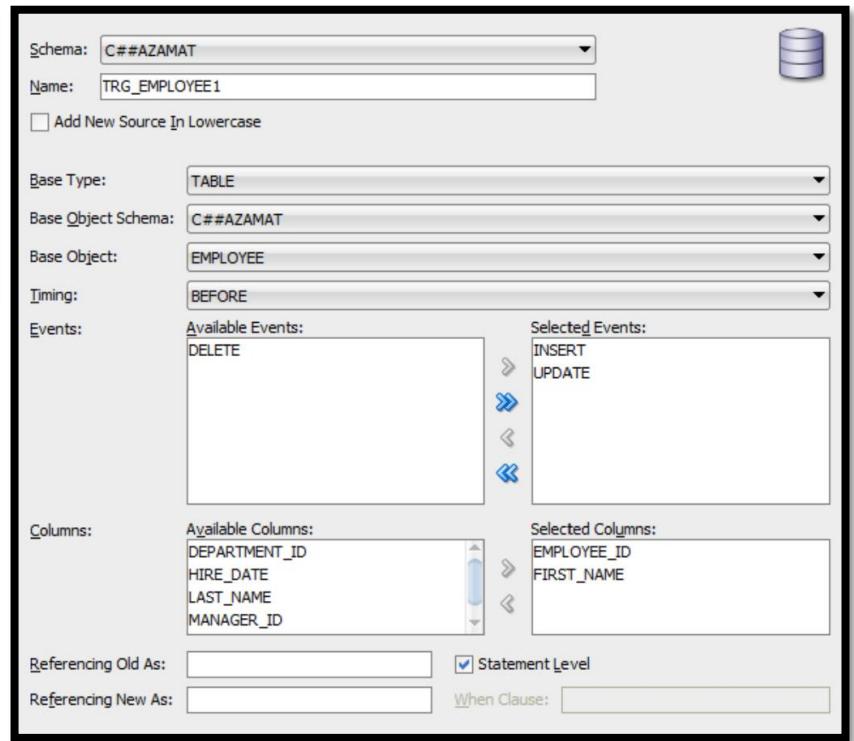
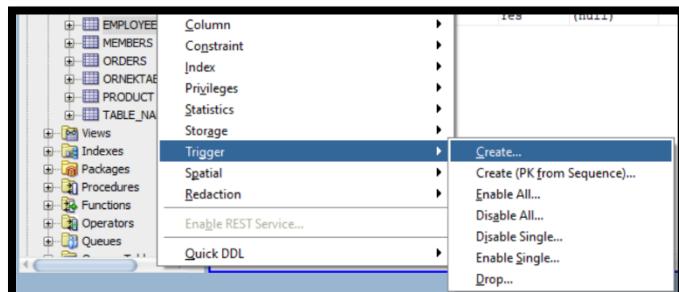
Adding components PL/SQL

Create DML Trigger

Create TRIGGER 1-usul



Create TRIGGER 2-usul

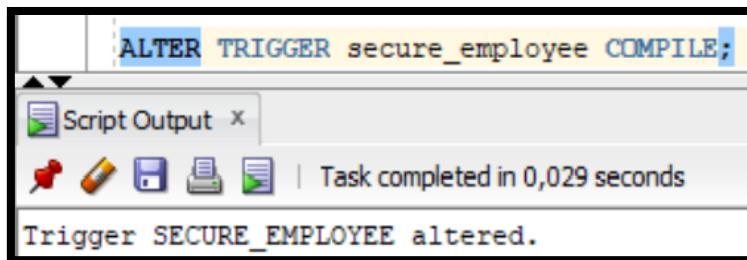


```
CREATE OR REPLACE TRIGGER trg_employee1 BEFORE
  INSERT OR UPDATE OF employee_id, first_name ON employee
BEGIN
  NULL;
END;
```

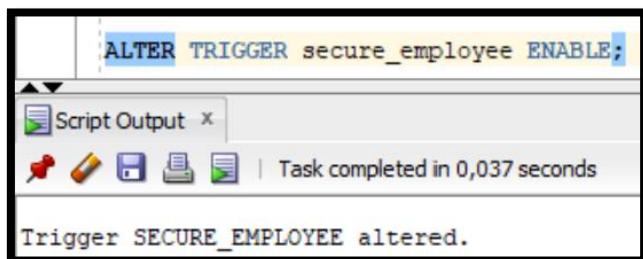
Manage Trigger

```
create or replace TRIGGER secure_employee BEFORE  
    DELETE OR INSERT OR UPDATE ON employee  
BEGIN  
    secure_dml;  
END secure_employee;
```

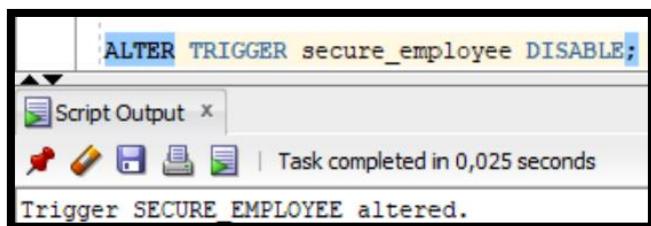
Komplyatsiya qilish:



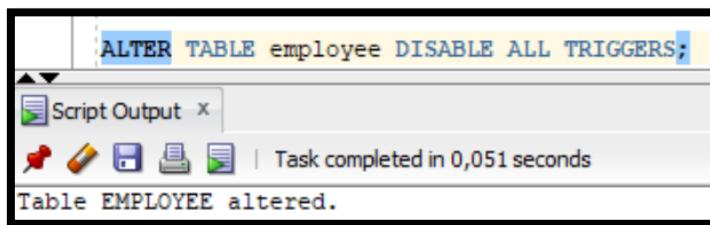
Enable:



Disable:

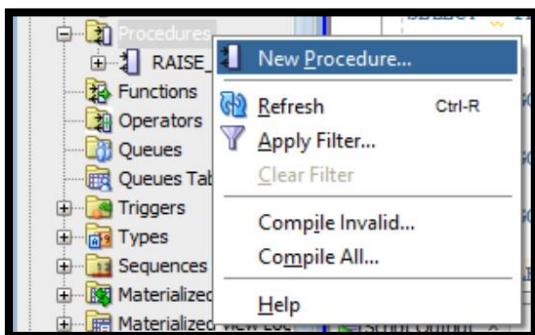


Disable All Triggers:



Create Stored Procedure

Create Procedure:



Schema: C##AZAMAT

Name: RAISE_AMOUNT

Add New Source In Lowercase

Parameters: + X

Name	Mode	No Copy	Data Type	Default Value
P_ID	IN	<input type="checkbox"/>	NUMBER	
P_RAISE_AMT	IN	<input checked="" type="checkbox"/>	NUMBER	

```
create or replace PROCEDURE RAISE_AMOUNT
(
    P_ID IN NUMBER
    , P_RAISE_AMT IN DECIMAL DEFAULT 0
) AS
BEGIN
    UPDATE employee
    SET salary = salary * (1 + p_raise_amt)
    WHERE Employee_ID = P_ID;
END RAISE_AMOUNT;
```

Kompliyatsiya:



Protsedurani ishlatish

Salaryni aniqlaymiz:

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL query:

```
SELECT
    first_name,
    salary
FROM employee
WHERE employee_id = 105;
```

Below the code editor is a toolbar with icons for Script Output, Query Result, and other database operations. The Query Result tab is selected, showing the output of the query:

	FIRST_NAME	SALARY
1	Tammy	45000

Qiymatni o'zgartiramiz:

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following PL/SQL block:

```
BEGIN
    RAISE_AMOUNT(105, 1);
END;
```

Below the code editor is a toolbar with icons for Query Result and Script Output. The Query Result tab is selected, showing the output of the block:

Task completed in 0,055 seconds
PL/SQL procedure successfully completed.

Natijani tekshiramiz: Salary 1 baravar oshdi.

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the same SELECT query as before:

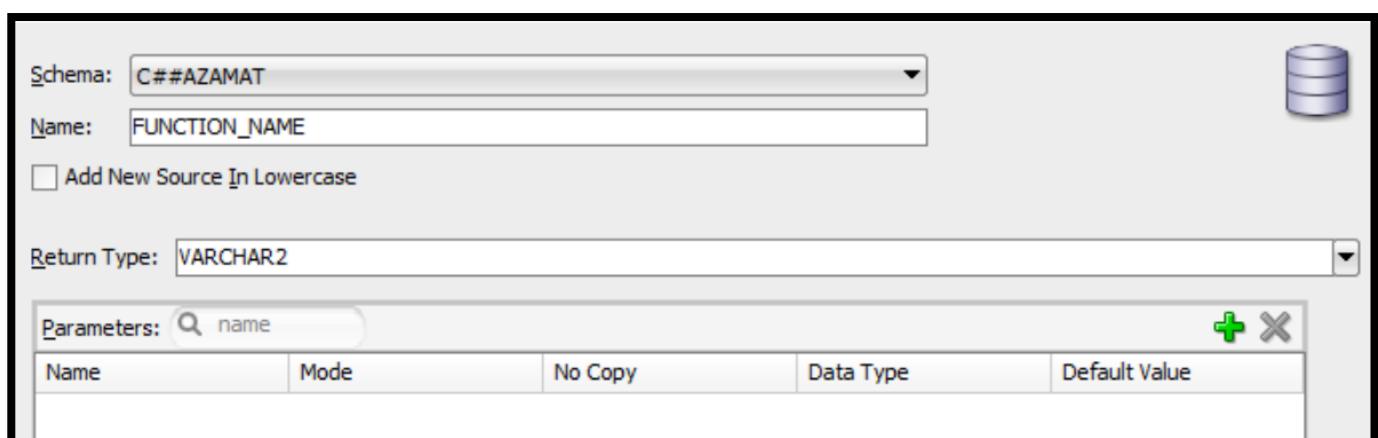
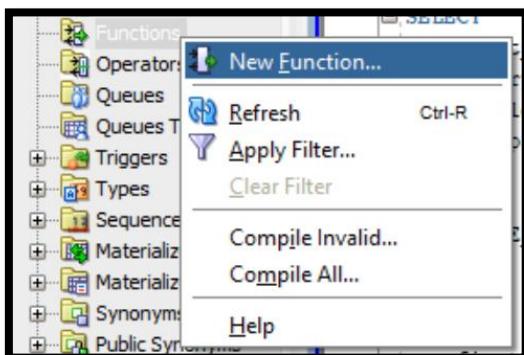
```
SELECT
    first_name,
    salary
FROM employee
WHERE employee_id = 105;
```

Below the code editor is a toolbar with icons for Query Result and Script Output. The Query Result tab is selected, showing the updated output of the query:

	FIRST_NAME	SALARY
1	Tammy	90000

Use of Function

Create Function



```
CREATE OR REPLACE FUNCTION function_name RETURN VARCHAR2 AS
BEGIN
    RETURN NULL;
END function_name;
```

Get_Sal nomli funksiya yaratish

```
CREATE OR REPLACE FUNCTION GET_SAL
(
    P_ID IN Employee.Employee_id%Type
    , P_INCREMENT IN NUMBER := 1
) RETURN NUMBER
IS
    v_sal Employee.Salary%Type := 0;
BEGIN
    SELECT Salary * P_Increment
    INTO v_sal
    FROM Employee
    WHERE Employee_id = P_id;
    RETURN v_sal;
END GET_SAL;
```

Komplyatsiya:



Funksiyani ishlatalamiz: Salary ning qiymati oshirilib taklif qilinmoqda.

The screenshot shows a SQL query window with the following code:

```
SELECT
    employee_id,
    first_name,
    salary,
    GET_SAL(employee_id, 1.4) "Proposed Salary"
FROM employee;
```

Below the query window is a 'Query Result' tab showing the output:

EMPLOYEE_ID	FIRST_NAME	SALARY	Proposed Salary
1	1 Michelle	48000	67200
2	2 Cheryl	79000	110600
3	3 Carolyn	47000	65800

SYNONYM

Create Synonym:

The screenshot shows a SQL script window with the following command:

```
CREATE SYNONYM users
FOR Customer;
```

Below the script window is a 'Query Result' tab showing the output:

Synonym USERS created.

Customer jadvalini chiqaramiz:

The screenshot shows a SQL query window with the following code:

```
SELECT * FROM customer;
```

Below the query window is a 'Query Result' tab showing the output:

CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	1 Teresa	Hudson
2	2 Fred	Montgomery
3	3 Lois	Lawson

Usersni tekshiramiz: ikkovi aynan bir xil

The screenshot shows the SQL developer interface with a query window containing the SQL statement: `SELECT * FROM users;`. Below the query window is a 'Query Result' tab showing the following data:

	CUSTOMER_ID	FIRST_NAME	LAST_NAME
1	1	Teresa	Hudson
2	2	Fred	Montgomery
3	3	Lois	Lawson

Usersga ma'lumot kiritamiz:

The screenshot shows the SQL developer interface with a query window containing the SQL statement: `INSERT INTO Users VALUES (11, 'Cihan', 'O'zhan', 'TK', 'Cihan@gmail.com');`. Below the query window is a 'Query Result' tab showing the following message:

Task completed in 0,044 seconds
1 row inserted.

Endi Customer jadvalini tekshiramiz: yangilanish parallel sodir bo`ldi.

The screenshot shows the SQL developer interface with a query window containing the SQL statement: `SELECT * FROM customer WHERE customer_id = 11;`. Below the query window is a 'Query Result' tab showing the following data:

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	EMAIL_ADDRESS
1	11	Cihan	O'zhan	TK	Cihan@gmail.com

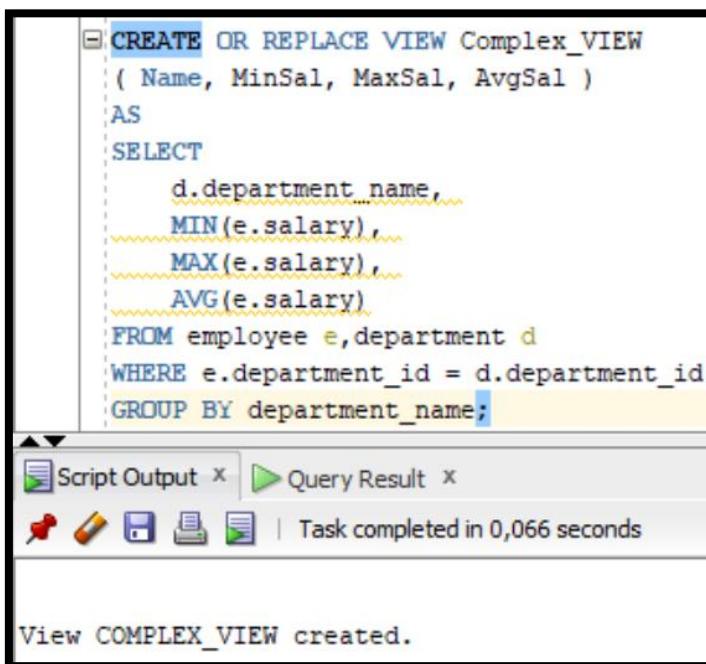
Drop Synonym:

The screenshot shows the SQL developer interface with a query window containing the SQL statement: `DROP SYNONYM users;`. Below the query window is a 'Query Result' tab showing the following message:

Task completed
Synonym USERS dropped.

VIEW

Kompleks View yaratish:

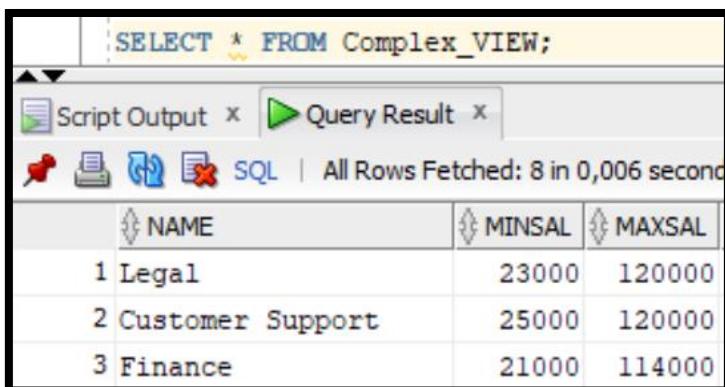


```
CREATE OR REPLACE VIEW Complex_VIEW
( Name, MinSal, MaxSal, AvgSal )
AS
SELECT
    d.department_name,
    MIN(e.salary),
    MAX(e.salary),
    AVG(e.salary)
FROM employee e,department d
WHERE e.department_id = d.department_id
GROUP BY department_name;
```

View COMPLEX_VIEW created.



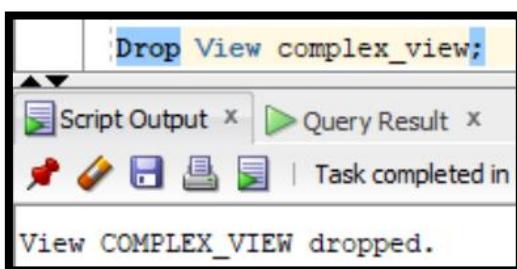
Viewni tekshiramiz:



```
SELECT * FROM Complex_VIEW;
```

NAME	MINSAL	MAXSAL
1 Legal	23000	120000
2 Customer Support	25000	120000
3 Finance	21000	114000

Viewnni o`chirish:



```
Drop View complex_view;
```

View COMPLEX_VIEW dropped.

Sequence

Create Sequence:

```
CREATE SEQUENCE Customer_Seq
INCREMENT BY 5
START WITH 20
MAXVALUE 9999;
```

Sequence CUSTOMER_SEQ created.

Customer jadvaliga ma'lumot kiritamiz:

```
INSERT INTO Customer VALUES(customer_seq.nextval, 'Ali', 'Candan', 'TK', 'Candan@gmail.com');
INSERT INTO Customer VALUES(customer_seq.nextval, 'Elif', 'Eylul', 'TK', 'Eylul@gmail.com');
INSERT INTO Customer VALUES(customer_seq.nextval, 'Mehmet', 'Cakir', 'TK', 'Candan@gmail.com');
```

1 row inserted.

Customer jadvalini tekshiramiz:

```
SELECT * FROM Customer
WHERE Customer_id >= 20;
```

CUSTOMER_ID	FIRST_NAME	LAST_NAME	ADDRESS_STATE	EMAIL_ADDRESS
1	20 Ali	Candan	TK	Candan@gmail.com
2	25 Elif	Eylul	TK	Eylul@gmail.com
3	30 Mehmet	Cakir	TK	Candan@gmail.com

Sequence ning oxirgi qiymatini aniqlash:

```
SELECT Customer_Seq.CURRVAL
FROM dual;
```

CURRVAL
1 30

Userning barcha sequence larini ko`rish:

Hamma sequence larni ko`rish:

SELECT * FROM All_Sequences;		
SEQUENCE_OWNER	SEQUENCE_NAME	MIN_VALUE
1 SYS	DM\$EXPIMP_ID_SEQ	1
2 SYS	SCHEDULER\$_JOBSUFFIX_S	1
3 SYS	PLSQL_CODE_COVERAGE_RUNNUMBER	1

Drop Sequence:

```
DROP SEQUENCE Customer_Seq;
```