

- Procedure which uses SQL%ROWCOUNT to determine the number of rows affected

1) In this example, the procedure takes a product ID and a new price as input parameters and updates the salary of the corresponding product in the "PRODUCT" table. The SQL%ROWCOUNT attribute is used to determine the number of rows updated by the UPDATE statement. The DBMS_OUTPUT.PUT_LINE procedure is used to display the number of rows updated.

(В этом примере процедура принимает идентификатор продукта и новую цену в качестве входных параметров и обновляет зарплату соответствующего продукта в таблице "PRODUCT". Атрибут SQL%ROWCOUNT используется для определения количества строк, обновленных с помощью инструкции UPDATE. DBMS_OUTPUT.Процедура PUT_LINE используется для отображения количества обновленных строк.)

PRODUCT								
Columns Data Indexes Constraints Grants Statistics Triggers Dependencies DDL Sample Queries								
+ Insert Row Columns... Filter... Count Rows Load Data Download Refresh								
	PRODUCT_ID	NAME	QUANTITY	DESCRIPTION	CITY	PRICE	RATING	CATEGORY_ID
	31	Eazzy	90	quam nec dui luctus	Rawa	29614	1	61
	32	Trupe	147	eleifend quam a o...	Curahuasi	27652	4	62
	33	Snaptags	186	arcu libero rutrum...	Ejidal	32321	1	63
	34	Yacero	135	odio curabitur con...	Novi Slankamen	37195	1	64
	35	Oyundu	53	et ultrices posuer...	Shazi	20700	2	65
	36	Twitterbeat	58	sed justo pellente...	Hanjiaji	33628	1	66
	37	Blogtags	200	tortor id nulla ultri...	Olivia	10176	5	67
	38	Linkbridge	140	nunc rhoncus dui ...	Oyskhara	28164	3	68
	39	Twitterbridge	86	eu interdum eu ti...	Butel	19427	2	69
	40	Realmix	96	sed ante vivamus ...	Soanindrarin	39913	4	70

```

1 CREATE OR REPLACE PROCEDURE update_product_price(
2   p_product_id IN NUMBER,
3   p_new_price IN NUMBER
4 )
5 IS
6 BEGIN
7   UPDATE PRODUCT
8   SET PRICE = p_new_price
9   WHERE PRODUCT_ID = p_product_id;
10
11   IF SQL%ROWCOUNT > 0 THEN
12     DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows updated.');

Results Explain Describe Saved SQL History



Procedure created.


```

```

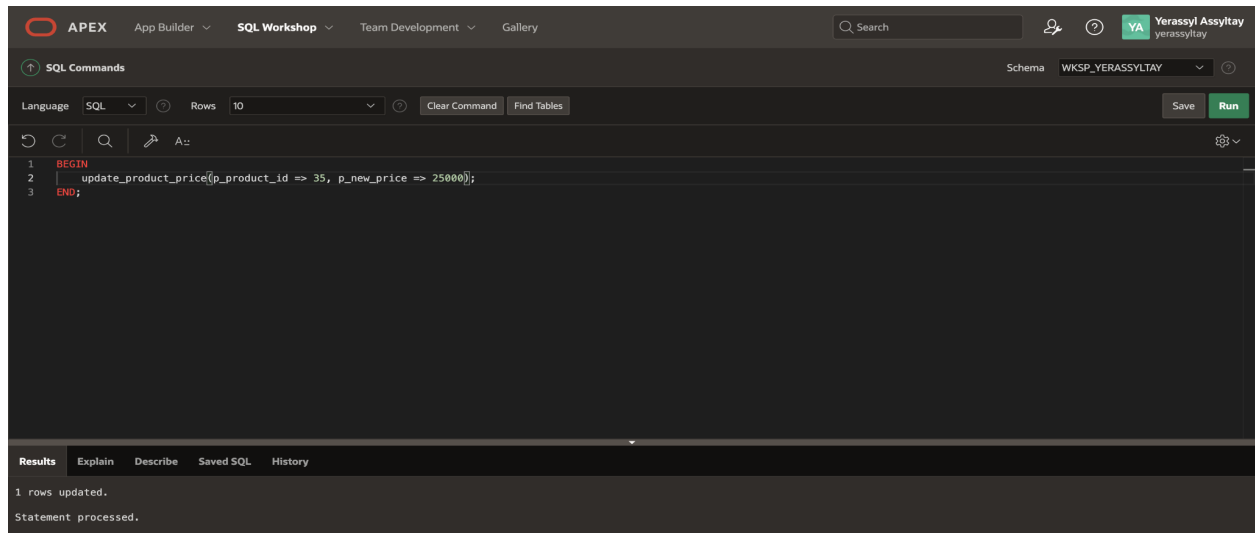
CREATE OR REPLACE PROCEDURE update_product_price(
  p_product_id IN NUMBER,
  p_new_price IN NUMBER)
IS
BEGIN
  UPDATE PRODUCT
  SET PRICE = p_new_price
  WHERE PRODUCT_ID = p_product_id;
  IF SQL%ROWCOUNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows updated.');
```

```
  ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('No rows updated.');
```

```
  END IF;
```

```
END;
```



```

BEGIN
  update_product_price(p_product_id => 35, p_new_price => 25000);
END;
```

PRODUCT									
Columns Data Indexes Constraints Grants Statistics Triggers Dependencies DDL Sample Queries									
+ Insert Row Columns... Filter... Count Rows Load Data Download Refresh									
	PRODUCT_ID	NAME	QUANTITY	DESCRIPTION	CITY	PRICE	RATING	CATEGORY_ID	
	31	Eazzy	90	quam nec dui luctus	Rawa	29614	1	61	
	32	Trupe	147	eleifend quam a o...	Curahuasi	27652	4	62	
	33	Snaptags	186	arcu libero rutrum...	Ejidal	32321	1	63	
	34	Yacero	135	odio curabitur con...	Novi Slankamen	37195	1	64	
	35	Oyundu	53	et ultrices posuer...	Shazi	25000	2	65	
	36	Twitterbeat	58	sed justo pellente...	Hanjiaji	33628	1	66	
	37	Blogtags	200	tortor id nulla ultri...	Olivia	10176	5	67	
	38	Linkbridge	140	nunc rhoncus dui ...	Oyskhara	28164	3	68	
	39	Twitterbridge	86	eu interdum eu ti...	Butel	19427	2	69	
	40	Realmix	96	sed ante vivamus ...	Soanindrarinny	39913	4	70	

2)The procedure delete_deliver_by_name looks correct. It takes the input parameter d_name, which represents the name of the person whose records need to be deleted from the DELIVERS table based on the DELIVER_NAME column. If any rows were affected as a result of deletion, the procedure outputs the number of deleted rows using DBMS_OUTPUT.PUT_LINE.

(Процедура delete_deliver_by_name выглядит верно. Она принимает входной параметр d_name, который представляет имя человека, записи о котором нужно удалить из таблицы DELIVERS на основе столбца DELIVER_NAME. Если в результате удаления были затронуты какие-либо строки, то процедура выводит количество удаленных строк с помощью DBMS_OUTPUT.PUT_LINE.)

DELIVERS								
Columns Data Indexes Constraints Grants Statistics Triggers Dependencies DDL Sample Queries								
+ Insert Row Columns... Filter... Count Rows Load Data Download Refresh								
	ID	USER_ID	SELLER_ID	CUSTOMER_ID	DELIVER_DATE	DELIVER_NAME	PHONE_NUMBER	DELIVER_ID
	1	1	21	11	08/20/0026	Bytecard	983-509-4823	91
	2	2	22	12	08/20/0023	Cardify	648-405-7372	92
	3	3	23	13	06/20/0023	Cardguard	481-642-8807	93
	4	4	24	14	01/20/0004	Y-find	503-750-2303	94
	5	5	25	15	01/20/0009	Bytecard	261-414-6259	95
	6	6	26	16	04/20/0005	Flowdesk	380-594-9802	96
	7	7	27	17	12/20/0023	Solarbreeze	149-299-1418	97
	8	8	28	18	04/20/0020	Temp	947-504-3768	98
	9	9	29	19	07/20/0004	Aerified	923-188-0374	99
	10	10	30	20	04/20/0015	Tempsoft	574-538-8721	100

```

APEX App Builder SQL Workshop Team Development Gallery
Search
YA Yerassyl Assyltay yerassyltay
Schema WKSP_YERASSYLTA
SQL Commands
Language SQL Rows 10 Clear Command Find Tables Save Run
1 CREATE OR REPLACE PROCEDURE delete_deliver_by_name(d_name IN VARCHAR2)
2 IS
3 BEGIN
4     DELETE FROM DELIVERS
5     WHERE DELIVER_NAME = d_name;
6
7     IF SQL%ROWCOUNT > 0 THEN
8         DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows deleted.'); Results Explain Describe Saved SQL History || Procedure created. |  |

```

```

CREATE OR REPLACE PROCEDURE delete_deliver_by_name(d_name IN VARCHAR2)
IS
BEGIN
    DELETE FROM DELIVERS
    WHERE DELIVER_NAME = d_name;

    IF SQL%ROWCOUNT > 0 THEN

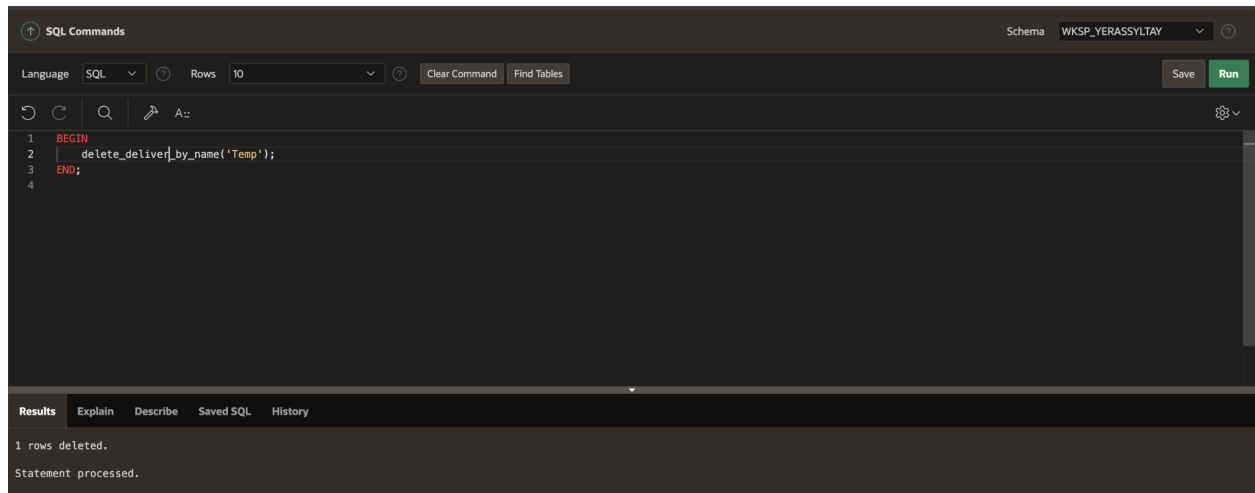
```

```

        DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows deleted. ');
ELSE
    DBMS_OUTPUT.PUT_LINE('No rows deleted. ');
END IF;
END;

```

we can call this procedure by passing it the desired name to delete from the DELIVERS table, for example:



```

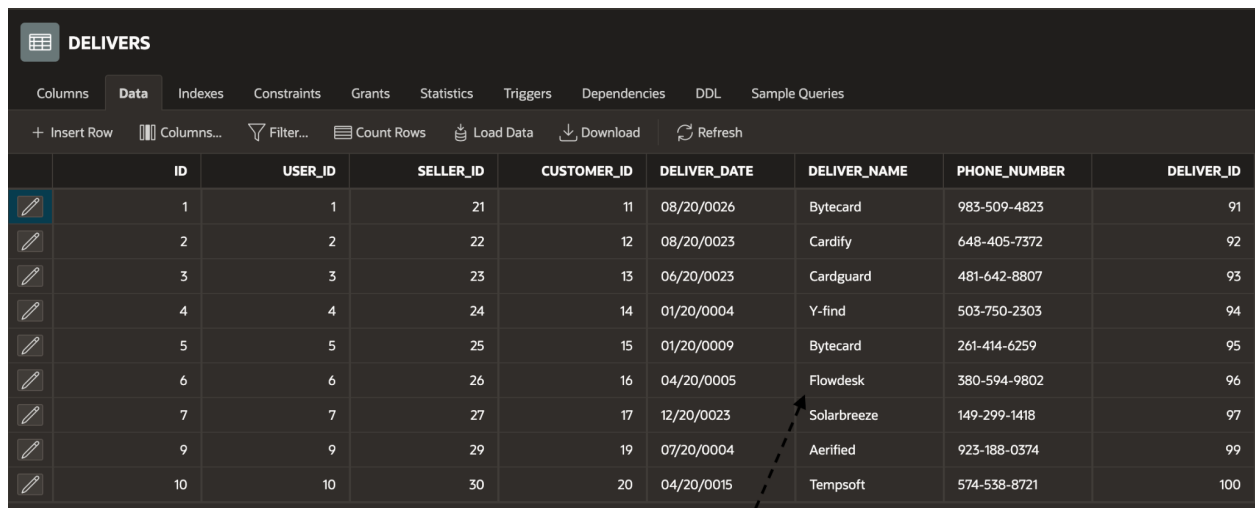
SQL Commands
Schema: WKSP_YERASSYLTA
Language: SQL Rows: 10
1 BEGIN
2   delete_deliver_by_name('Temp');
3 END;
4
Results
1 rows deleted.
Statement processed.

```

```

BEGIN
    delete_deliver_by_name('Temp');
END;

```



	ID	USER_ID	SELLER_ID	CUSTOMER_ID	DELIVER_DATE	DELIVER_NAME	PHONE_NUMBER	DELIVER_ID
	1	1	21	11	08/20/0026	Bytecard	983-509-4823	91
	2	2	22	12	08/20/0023	Cardify	648-405-7372	92
	3	3	23	13	06/20/0023	Cardguard	481-642-8807	93
	4	4	24	14	01/20/0004	Y-find	503-750-2303	94
	5	5	25	15	01/20/0009	Bytecard	261-414-6259	95
	6	6	26	16	04/20/0005	Flowdesk	380-594-9802	96
	7	7	27	17	12/20/0023	Solarbreeze	149-299-1418	97
	9	9	29	19	07/20/0004	Aerified	923-188-0374	99
	10	10	30	20	04/20/0015	Tempsoft	574-538-8721	100