**CONVERT RESULT SET TO XML IN ORACLE**

<https://livesql.oracle.com/apex/livesql/file/content_E9QWIDZT8PCCVA3PMDE33WRK4.html>

# 3 ways to turn SQL query results into XML

[**View All Scripts**](https://livesql.oracle.com/apex/f?p=590:49::::RP,49:P49_TYPES:C)[**Login to Run Script**](https://livesql.oracle.com/apex/f?p=590:41:0:::41:P41_ID:111925498451493177308291977770837922326)

* Script Name3 ways to turn SQL query results into XML
* DescriptionMethods for turning SQL result sets into XML.
* AreaSQL General / XML
* ContributorChris Saxon (Oracle)
* CreatedThursday July 13, 2017
* Statement **1**
* create table toys (
* toy\_id integer,
* toy\_name varchar2(30),
* price number,
* colour varchar2(30)

)

Table created.

* Statement **2**
* begin
* insert into toys values (1, 'Cheapasaurus Rex', 0.99, 'blue');
* insert into toys values (2, 'Costsalottasaurs', 99.99, 'green');
* insert into toys values (3, 'Bluesaurus', 21.99, 'blue');
* commit;

end;

1 row(s) inserted.

* Statement **3**

select \* from toys

|  |  |  |  |
| --- | --- | --- | --- |
| **TOY\_ID** | **TOY\_NAME** | **PRICE** | **COLOUR** |
| 1 | Cheapasaurus Rex | .99 | blue |
| 2 | Costsalottasaurs | 99.99 | green |
| 3 | Bluesaurus | 21.99 | blue |

3 rows selected.

* Statement **4**

XMLForest converts each argument to an XML element. It then combines these into an XML fragment for each row in the input.

select xmlforest(toy\_id, toy\_name, price, colour).getClobVal() xdoc from toys

|  |
| --- |
| **XDOC** |
| <TOY\_ID>1</TOY\_ID><TOY\_NAME>Cheapasaurus Rex</TOY\_NAME><PRICE>.99</PRICE><COLOUR>blue</COLOUR> |
| <TOY\_ID>2</TOY\_ID><TOY\_NAME>Costsalottasaurs</TOY\_NAME><PRICE>99.99</PRICE><COLOUR>green</COLOUR> |
| <TOY\_ID>3</TOY\_ID><TOY\_NAME>Bluesaurus</TOY\_NAME><PRICE>21.99</PRICE><COLOUR>blue</COLOUR> |

3 rows selected.

* Statement **5**

XMLAgg combines multiple XML fragments into a single document. The XMLElement call around XMLForest places each row in it's own ROW element. Without this all rows are at the same depth in the document, for example: 1Cheapasaurus Rex.99blue3Costsalottasaurs99.99green4Bluesaurus21.99blue

select xmlagg(xmlelement("ROW", xmlforest(toy\_id, toy\_name, price, colour))).getClobVal() xdoc

from toys

|  |
| --- |
| **XDOC** |
| <ROW><TOY\_ID>1</TOY\_ID><TOY\_NAME>Cheapasaurus Rex</TOY\_NAME><PRICE>.99</PRICE><COLOUR>blue</COLOUR></ROW><ROW><TOY\_ID>2</TOY\_ID><TOY\_NAME>Costsalottasaurs</TOY\_NAME><PRICE>99.99</PRICE><COLOUR>green</COLOUR></ROW><ROW><TOY\_ID>3</TOY\_ID><TOY\_NAME>Bluesaurus</TOY\_NAME><PRICE>21.99</PRICE><COLOUR>blue</COLOUR></ROW> |

* Statement **6**

Passing a cursor to XMLType will return the results of the query as a single XML document.(Notes by mustafa.sacli:  
empty result set throws error(ORA-06502).)

select xmltype(cursor(select \* from toys)).getClobVal() xdoc from dual

|  |
| --- |
| **XDOC** |
| <?xml version="1.0"?> <ROWSET> <ROW> <TOY\_ID>1</TOY\_ID> <TOY\_NAME>Cheapasaurus Rex</TOY\_NAME> <PRICE>.99</PRICE> <COLOUR>blue</COLOUR> </ROW> <ROW> <TOY\_ID>2</TOY\_ID> <TOY\_NAME>Costsalottasaurs</TOY\_NAME> <PRICE>99.99</PRICE> <COLOUR>green</COLOUR> </ROW> <ROW> <TOY\_ID>3</TOY\_ID> <TOY\_NAME>Bluesaurus</TOY\_NAME> <PRICE>21.99</PRICE> <COLOUR>blue</COLOUR> </ROW> </ROWSET> |

* Statement **7**

dbms\_xmlgen.getxml works in similar way to passing a cursor to XMLType: it executes the query and returns the result as a single XML document.

select dbms\_xmlgen.getxml('select \* from toys') xdoc from dual

|  |
| --- |
| **XDOC** |
| <?xml version="1.0"?> <ROWSET> <ROW> <TOY\_ID>1</TOY\_ID> <TOY\_NAME>Cheapasaurus Rex</TOY\_NAME> <PRICE>.99</PRICE> <COLOUR>blue</COLOUR> </ROW> <ROW> <TOY\_ID>2</TOY\_ID> <TOY\_NAME>Costsalottasaurs</TOY\_NAME> <PRICE>99.99</PRICE> <COLOUR>green</COLOUR> </ROW> <ROW> <TOY\_ID>3</TOY\_ID> <TOY\_NAME>Bluesaurus</TOY\_NAME> <PRICE>21.99</PRICE> <COLOUR>blue</COLOUR> </ROW> </ROWSET> |

# Additional Information

* Database on OTN[SQL and PL/SQL Discussion forums](https://community.oracle.com/tech/developers/categories/sql_and_pl_sql)  
  [Oracle Database](http://otn.oracle.com/database)  
  [Download Oracle Database](https://www.oracle.com/database/technologies/oracle-database-software-downloads.html)