

	sup.Fac.	construct	construct	construct	construct	getValue	getValue	getValue	getValue	ctf	ctf	ctf	use
trigger		no	no	yes	yes	no	no	yes	yes	no	no	yes	yes
returning		no	yes	no	yes	no	yes	no	yes	no	yes	no	yes
db2		before	Exception	after	Exception	before	Exception	Exception	Exception	yes	no	no	no
derby	Exception												
hsqldb	Exception												
informix	Exception												
instantdb	Exception												
interbase		before	Exception	after	Exception	before	Exception	Exception	Exception	no	no	no	no
mysql	Exception												
oracle		after	during	after	during	currval	(currval)	currval	(currval)	yes	yes	no	yes
pointbase	Exception												
postgresql		before	before	after	Exception	before	before	after	after	yes	yes	yes	no
progress	Exception												
sapdb		after	Exception	after	Exception	currval	(currval)	currval	(currval)	no	no	no	no
Sql-server	Exception												
sybase	Exception												

Analysis and conclusion

All databases that support returning should use during independent of trigger. One exception is db2 which would need a special implementation.

-> Always use during style with returning with special handling for db2.

When using trigger without returning we can only retrieve generated id after execution of the insert statement.

To get rid of the special handling of postgresql when using after style we should change to use alternate syntax.

-> Always use after style with trigger if not returning.

As reported under CASTOR-2724 sequence key generator on SapDB could have never worked with after style. Instead before style should be used.

For Oracle with no trigger and without returning there is an alternate and better option available using before style.

All other database engines already use or support before style with no trigger and without returning.

-> Always use before style with no trigger and without returning.

Use of colors in table below:

- yellow Syntax as currently supported by Castor which does not change
- orange Syntax as currently supported by Castor which should be changed to the one marked in green
- green New syntax which should be used instead of the one marked in orange
- white Syntax which is not supported but could be added later

database	version	trigger	returning	style	statement
db2	---	no	no	before	1. SELECT NEXTVAL FOR sequence FROM SYSIBM.SYSDUMMY1; 2. INSERT INTO table (id, col) VALUES (?, ?);
db2	V8	no	yes	???	1. SELECT id FROM FINAL TABLE (INSERT (id, col) INTO table VALUES (NEXTVAL FOR sequence, ?));
hsql	V1.7.2	no	no	before	1. SELECT NEXT VALUE FOR sequence FROM SYSTEM_SEQUENCES WHERE SEQUENCE_NAME='sequence'; 2. INSERT INTO table (id, col) VALUES (?, ?);
hsql	V1.7.2	no	no	before	1. SELECT NEXT_VALUE FROM SYSTEM_SEQUENCES WHERE SEQUENCE_NAME='sequence'; 2. INSERT INTO table (id, col) VALUES (NEXT VALUE FOR sequence, ?);
informix	V9.40	no	no	before	1. SELECT sequence.NEXTVAL FROM SYSTABLES WHER TABID=1; 2. INSERT INTO table (id, col) VALUES (?, ?);
informix	V9.40	no	no	after	1. INSERT INTO table (id, col) VALUES (sequence.NEXTVAL, ?); 2. SELECT sequence.CURRVAL FROM SYSTABLES WHER TABID=1;
interbase	---	no	no	before	1. SELECT GEN_ID(sequence, 1) FROM RDB\$DATABASE; 2. INSERT INTO table (id, col) VALUES (?, ?);
oracle	---	no	no	before	1. SELECT sequence.NEXTVAL FROM DUAL; 2. INSERT INTO table (id, col) VALUES (?, ?);
oracle	---	no	no	after	1. INSERT INTO table (id, col) VALUES (sequence.NEXTVAL, ?); 2. SELECT sequence.CURRVAL FROM DUAL;
oracle	V10	no	yes	during	1. INSERT INTO table (id, col) VALUES (sequence.NEXTVAL, ?) RETURNING id INTO ?;
oracle	---	yes	no	after	1. INSERT INTO table (col) VALUES (?); 2. SELECT sequence.CURRVAL FROM DUAL;
oracle	V10	yes	yes	during	1. INSERT INTO table (col) VALUES (?) RETURNING id INTO ?;
postgresql	---	no	no	before	1. SELECT NEXTVAL('sequence'); 2. INSERT INTO table (id, col) VALUES (?, ?);
postgresql	V8.2	no	yes	during	1. INSERT INTO table (id, col) VALUES (NEXTVAL('sequence'), ?) RETURNING id;
postgresql	---	yes	no	after	1. INSERT INTO table (col) VALUES (?); 2. SELECT CURRVAL('sequence');
postgresql	---	yes	no	after	1. INSERT INTO table (col) VALUES (?); 2. see Castor code about how to retrieve col1 value generated by trigger
postgresql	V8.2	yes	yes	during	1. INSERT INTO table (col) VALUES (?) RETURNING id;
sapdb	---	no	no	before	1. SELECT sequence.NEXTVAL FROM DUAL; 2. INSERT INTO table (id, col) VALUES (?, ?);