

# MyBatis Generator Generated SQL Map Files

MyBatis Generator (MBG) generates SQL Map files that conform to the MyBatis or iBATIS SQL Map DTD. The files contain many different elements based on the characteristics of the table, and on the configuration options you specify. MBG generates a different SQL Map file for every table you specify. The name space of the SQL Map is the name of the table (qualified by schema and catalog if present). MBG does not add the SQL Map entries to the MyBatis/iBATIS configuration files - you must do that manually (or you may use a plugin to cause MBG to generate a skeleton configuration file if you wish).

Every generated XML element contains an XML comment section that contains the string `@mbggenerated`. On subsequent runs, every element that contains a comment with the string `@mbggenerated` will be deleted and replaced. Any other element in the XML file will remain untouched. With this in mind, you can add other elements to the file without fear of losing them in subsequent runs - simply do not include the string `@mbggenerated` in any element comment.

The following sections describe the elements that will be generated.

Note: in the following descriptions, the term "BLOB" is used to refer to any column with a data type of BLOB, CLOB, LONGVARCHAR, or LONGVARBINARY.

## Result Map

This element is used to map table columns to properties of the generated Java model object. The result map (and corresponding select statements) will not contain:

- Any field that has been ignored by the `<ignoreColumn>` configuration element
- Any BLOB field from the table (see the result map with BLOBs element)

The columns will be mapped according the configuration element `<columnOverride>` if it exists for the specific column. If the override does not exist, then a default property name and JDBC type will be used.

It is acceptable to extend this result map if you code any custom join queries in the SQL map. This is a common use case and is expected. If you plan to reuse this result map with other join queries, you may wish to have MBG generate a prefix for the fields in the result map. See the [<table>](#) reference page for information about generating a prefix.

This element will be generated if either the select by example, or select by primary key statements are enabled.

## Result Map With BLOBs

The element extends the base result map, and adds any BLOB fields that exist in the table. We do this because we offer different versions of the select by example statement depending on whether or not you want to return the BLOB fields in those queries.

The result map (and corresponding select statements) will not contain:

- Any field that has been ignored by the `<ignoreColumn>` configuration element

The columns will be mapped according the configuration element `<columnOverride>` if it exists for the specific column. If the override does not exist, then a default property name and JDBC type will be used.

It is acceptable to extend this result map if you code any custom join queries in the SQL map. This is a common use case and is expected. If you plan to reuse this result map with other join queries, you may wish to have MBG generate a prefix for the fields in the result map. See the [<table>](#) reference page for information about generating a prefix.

This element will be generated if that table contains BLOB fields, and either the select by example, or select by primary key statements are enabled.

## SQL Where Clause 可重用的 where 子句

This element contains a reusable where clause that is used by the "by example" methods. The where clause will not include any BLOB fields if they exist in the table. Most databases do not support BLOB fields in the WHERE clause.

This element will be generated if any of the "by example" statements are enabled.

## Select By Primary Key

This element contains a select statement that will return one row - designated by the primary key. The returned row will include BLOB fields if they exist in the table.

This element will be generated if the table has a primary key and the select by primary key statement is enabled.

## Select by Example 选择匹配的所有行

This element contains a select statement with rows that match the example object. This implements a simple "query by example" functionality that can be used to generate many different database queries. The returned rows will not include any BLOB fields that exist in the table (see the select by example with BLOBs statement below). 不包括任何的BLOB字段

**Important:** If the example class is null, or no criteria have been set, then all rows in the table will be selected.

This element will be generated if the select by example statement is enabled.

## Select by Example With BLOBs

This element contains a select statement with rows that match the example object. This implements a simple "query by example" functionality that can be used to generate many different database queries. The returned rows will include any BLOB fields that exist in the table.

**Important:** If the example class is null, or no criteria have been set, then **all** rows in the table will be selected.

This element will be generated if the table contains BLOB fields, and the select by example statement is enabled.

## Insert 插入表中所有的字段（包括 BLOB类型的字段）

This element is an insert statement that includes all fields in the table (including BLOBs), unless the field is specifically ignored with the `<ignoreColumn>` configuration element.

If the table has an auto generated key (an identity column or value from a sequence), and the `<generatedKey>` configuration element is specified, then MBG will generate an appropriate <selectKey> element.

**Important Note:** the insert method works differently in iBATIS2 and MyBatis3. The following table summarizes the differences:

Runtime	Behavior
iBATIS2	If a <code>&lt;generatedKey&gt;</code> element is specified, the method will return the newly generated key. If no <code>&lt;generatedKey&gt;</code> element is specified, the method will be <code>void</code> .

MyBatis3	The method will return the number of rows inserted (typically either 0 or 1). If a <code>&lt;generatedKey&gt;</code> element is specified, the value of the newly generated key will be set in the corresponding property of the parameter object.
----------	--

This element will be generated if the insert statement is enabled.

## Insert Selective

This element is an insert statement that includes all fields in the table (including BLOBs), unless the field is specifically ignored with the `<ignoreColumn>` configuration element. However, this statement will not include fields that are `null` in the parameter object. This allows you to use database defaults for columns, if they exist. This element will not allow the insert of `null` into any field - for that you must use the regular insert statement. **Important:** any field mapped to a Java primitive is always inserted by this method.

If the table has an auto generated key (an identity column or value from a sequence), and the `<generatedKey>` configuration element is specified, then MBG will generate an appropriate `<selectKey>` element.

**Important Note:** the insertSelective method works differently in iBATIS2 and MyBatis3. The following table summarizes the differences:

Runtime	Behavior
iBATIS2	If a <code>&lt;generatedKey&gt;</code> element is specified, the method will return the newly generated key. If no <code>&lt;generatedKey&gt;</code> element is specified, the method will be <code>void</code> .
MyBatis3	The method will return the number of rows inserted (typically either 0 or 1). If a <code>&lt;generatedKey&gt;</code> element is specified, the value of the newly generated key will be set in the corresponding property of the parameter object.

This element will be generated if the insert statement is enabled.

## Update By Primary Key

This element is an update statement that will update one row - designated by the primary key. The update statement will update all fields in the table unless:

- The field has been ignored by the `<ignoreColumn>` configuration element
- The field is a **BLOB field** (see the update by primary key with BLOBs element)

This element will be generated if the table has a primary key, and the update by primary key statement is enabled.

## Update By Primary Key With BLOBs

This element is an update statement that will update one row - designated by the primary key. The update statement will update all fields in the table (including BLOB fields) unless:

- The field has been ignored by the `<ignoreColumn>` configuration element

This element will be generated if the table has a primary key, the table has BLOB columns, and the update by primary key statement is enabled.

## Update By Primary Key Selective      部分字段更新

This element is an update statement that will update one row - designated by the primary key. The update statement will update only the fields in the table whose corresponding property in the parameter object is non-null. This statement can be used to update certain columns in a record without affecting all columns in the record. **Important:** if the column has been mapped to a primitive type, then the column will always be updated.

“原始类型”总是会被更新

This element will be generated if the table has a primary key, and the update by primary key statement is enabled.

## Delete By Primary Key

This element is a delete statement that will delete one row in the table - designated by the primary key.

This element will be generated if the table has a primary key, and the delete by primary key statement is enabled.

## Delete By Example

This element is a delete statement that will delete one or more rows in the table - designated by the example object.

**Important:** If the example class is null, or no criteria have been set, then all rows in the table will be deleted.

This element will be generated if the delete by example statement is enabled.

## Count By Example

返回表中匹配的记录行数

This element is a select count(\*) statement that will return the number of rows in the table that match the specified example object.

**Important:** If the example class is null, or no criteria have been set, then the select will return the number of rows in the entire table.

This element will be generated if the count by example statement is enabled.

## Update By Example

更新表中所有匹配的行

This element is an update statement that will update all rows in a table that match the specified example. The update statement will update all fields in the table unless:

更新表中所有的字段

- The field has been ignored by the `<ignoreColumn>` configuration element
- The field is a BLOB field (see the update by example with BLOBs element)

**Important:** If the example class is null, or no criteria have been set, then all rows in the table will be updated.

This element will be generated if the update by example statement is enabled.

## Update By Example With BLOBs

This element is an update statement that will update all rows in a table that match the specified example. The update statement will update all fields in the table (including BLOB fields) unless:

- The field has been ignored by the `<ignoreColumn>` configuration element

**Important:** If the example class is null, or no criteria have been set, then all rows in the table will be updated.

This element will be generated if the table contains BLOB columns, and the update by example statement is enabled.

## 选择性地更新“非 null”的字段

### Update By Example Selective

更新表中所有匹配的行

This element is an update statement that will update all rows in a table that match the specified example. The update statement will update only the fields in the table whose corresponding property in the parameter object is non-null. This statement can be used to update certain columns in certain records without affecting all columns in the records. **Important:** if the column has been mapped to a primitive type, then the column will always be updated.

“原始类型”总是会被更新

**Important:** If the example class is null, or no criteria have been set, then all rows in the table will be updated.

This element will be generated if the update by example statement is enabled.