|  |  |  |
| --- | --- | --- |
| **SQL statement** | **Row permission** | **Column mask (defined for column C1)** |
| INSERT INTO T1(C1, C2) VALUES('A', 'B'); | For each row to be inserted:   * If a user-defined row permission exists, the row can be inserted only when that row can be subsequently retrieved by the authorization ID of the INSERT statement. If the row cannot be inserted, the INSERT statement returns an error. * If no user-defined row permissions exist for the table only the default row permission is applied and no row is inserted. The INSERT statement returns an error.   The ENFORCED FOR ALL ACCESS clause ensures that users cannot insert data that they cannot read. |  |
| INSERT INTO T1(C1) SELECT SUBSTR( T2.C1, 8, 4) FROM T2 WHERE T2.C2 > 10;  CM->not CM  CM->CM (use true value)  Not CM->CM  Not CM->not CM (use true value) | **This should follow before** | * When the columns are used to derive the new values for an INSERT statement, the original column values, not the masked values, are used. If the columns have column masks, those column masks are applied to ensure the evaluation of the access control rules at run time masks the column to itself, not to a constant or an expression. This ensures that the masked values are the same as the original column values. If a column mask does not mask the column to itself, the new row is not inserted and an error is returned at run time.   For example, column T2.C1 is used to derive the value of a new row for INSERT. The column value of T2.C1, not the masked value, is used to derive the new value. Because column T2.C1 has a column mask, the column mask is applied to ensure the evaluation of the access control rule in the column mask masks column T2.C1 to itself, not to a constant or an expression. This ensures the masked value is the same as the original column value. If the column mask of T2.C1 does not mask column T2.C1 to itself, the new value cannot be used and an error is returned at run time.   * The column mask rules that apply to the new value for INSERT are the same as those for SELECT. |
| UPDATE T1 SET C2 = (SELECT SUBSTR(T2.C1, 8, 4) FROM T2 WHERE T2.C2 > 10); | The following rules are applied in the order as shown:   1. Identify candidate rows for updates:    * If a user-defined row permission exists, only the rows of the table that satisfy the row permission can be the candidate rows for UPDATE.    * If no user-defined row permissions exist for the table, only the default row permission is applied and no rows are updated. 2. If there are rows to be updated, for each row to be updated:    * When the columns are used to derive the new values for an UPDATE statement, the original column values, not the masked values, are used. If the columns have column masks, those column masks are applied to ensure that the evaluation of the access control rules at run time masks the column to itself, not to a constant or an expression. This ensures the masked values are the same as the original column values. If a column mask does not mask the column to itself, the new rvalue cannot be used for the update and an error is returned at run time.   For example, column T2.C1 is used to derive the new value for the update. The column value of T2.C1, not the masked value, is used to derive the new value. Because column T2.C1 has a column mask, the column mask is applied to ensure that the evaluation of the access control rule in the column mask masks column T2.C1 to itself, not to a constant or an expression. This ensures that the masked value is the same as the original column value. If the column mask of T2.C1 does not mask column T2.C1 to itself, the new value cannot be used for the update and an error is returned at run time   * + The column mask rules that apply to the new value for UPDATE are the same as those for SELECT.  1. If there are rows to be updated, for each row to be updated:    * If a user-defined row permission exists, the row can be updated only when that row can be subsequently retrieved by the authorization ID of the UPDATE statement. If the row cannot be updated, the UPDATE statement returns an error. The ENFORCED FOR ALL ACCESS clause ensures that users cannot update data that they cannot read.    * The column mask is not applicable in this retrieval. | |
| DELETE | * If a user-defined row permission exists for the table, only the rows that satisfy the permission are the candidate rows for an DELETE statement. * If no user-defined row permissions exist for the table, the default row permission is applied and no row can be deleted. |  |
|  |  |  |
|  |  |  |
|  |  |  |

Ibm db2

https://www.ibm.com/support/knowledgecenter/SSEPEK\_11.0.0/com.ibm.db2z11.doc.seca/src/tpc/db2z\_rules4rowcolaccessctrl.html