# Detailed algorithms for Mandatory access control

As usual, implementation of rules throws up unexpected complications. Tables have classifications as do the records they contain, and the interplay between them and user clearances is far from simple. The main purpose of the classification information for a table is to specify the set of groups and references that will apply to records classified above D. It can also specify a minimum clearance level for access to the table. The SA can completely specify or modify the classification of any record in the table (but for best results should use subsets of the groups and references that they have specified for the table).

I have the following for users other than the SA in my first implementation. (As usual in Pyrrho, any exception will roll back the transaction.)

## Read

1. If the user does not have select privilege on any of the columns selected or select \* has been specified and the user does not have select privilege for any columns, throw an informative exception (such as “User cannot select column x”, or “user cannot access any columns”).

2. If Select is enforced and the user’s clearance level does not exceed the table’s classification level, report that the table does not exist.

Even if the table contains rows to which the user’s clearance would give them access.

3. If Select is enforced by the table and the user’s clearance does not allow access to a given record, skip the record.

4. If Select is enforced and any records with classification above D are accessed, an audit record is added to the database immediately, whether or not the user’s transaction commits.

## Insert

1. If Insert is enforced by the table and the user does not have insert privilege or the user’s clearance does not exceed the table’s classification, throw an Access Denied exception.

2. If Insert is enforced by the table and the user has insert privilege, construct a record whose classification is equal to the user’s clearance, and insert it.

The new record’s classification label will have the user’s minimum clearance level: if this is above D, the groups will be the subset of the user’s groups that are in the table classification, and the references will be the same as the table (a subset of the user’s references).

If Insert is not enforced and the user has insert privilege, the record inserted will have level D classification.

## Update

1. If the user does not have update privilege for the table, throw an Access Denied exception.

2. If Update is not enforced the record’s classification will be unaffected (presumably it will be level D).

3. If Update is enforced by the table and the user’s clearance does not allow access to the table, throw an Access Denied exception.

Even if the update would access records that would match the user’s clearance.

4. If Update is enforced by the table, and a record selected for update is not one to which the user has clearance or does not match the user’s clearance level, throw an Access Denied exception.

Even if the user has a higher clearance than the record’s classification.

5. The updated record must have the same classification as the old record.

## Delete

1. If the user does not have delete permission for the table, throw and Access Denied exception.

Even if the user has a high security clearance.

2. If Delete is enforced by the table for the table or the user’s clearance does not exceed the table’s classification, throw an Access Denied exception.

Even if the delete would actually only remove records that match the user’s clearance.

3. If Delete is enforced by the table and the user has delete privilege for the table, but the record to be deleted has a classification level different from the user or the clearance does not allow access to the record, throw an Access Denied exception.

Even if the delete is attempting to remove an unclassified record.