



















Why Multiple Implementations? //To support databases that span networks which include machines with different architectures //To support mixed-language and mixed-compiler environments

//To meet different performance versus space versus recoverability trade-offs (e.g., set as a B-tree or linked list)

//Which implementation an object uses is specified at object creation time

11

Û





























Transactions
 //The object model supports a nested transaction model simple way of undoing changes made to stored persistent data method to protect against network failure for remote operations
Transaction::begin()> t:Transaction;
 Transaction::begin()> x:Transaction;
 Transaction::begin()> y:Transaction;
 if minor_error then y.abort() if major_error then y.abort_to_top_level()
 y.commit()
 x.commit()
 t.commit() 26

