Let us now consider locking concepts. And when we talk about locking, I'm referring to locking data in database tables. And there are two basic types of locking data that we need to bear in mind.

First of all is database locks which lock data in a physical database. So, what I mean is when we update a record. A lock gets set on a record. We can update it and then the lock is released. The lock is there to ensure that once we set it, only we can update the data and once we release the lock, that data can then be accessed by for the people. Now these locks are not sufficient in an SAP system. And they are generally only used. When our record is being modified in a single step dialogue process. Now, what I mean by a single step dialogue is that if we have a screen that contains all the data that we need to work with. And we click a button to update the data. And because everything is held on one screen it can be updated in the database in a single step. Therefore, the database can set a lock, update the data in the table and release the lock all very quickly. As you will with SAP systems, you will realize that database locks are not sufficient. And this is because transactions in an SAP system often occur over multiple steps. So, let's say if we add an employee record to the system, we may have to fill in ten screens of data. The first step may be the name and address, the second step would be the position that they would fill in the system, and so on. And the user when they enter this data. Only want the record to be added to the database at the very end of the transaction when all the data in all the screens has been input. So, think about this. If we tried to save the data that was entered in the first screen into the database, then moved onto the second screen and saved that, what happens if the user quit halfway through? Entering all these ten screens of data. We would have an invalid record in the database. We may have updated a few tables, but we really require, yeah, maybe five other tables filled in with data, for us to have a valid record. So, this is what I mean by using database slots with multi-step dialogue processes is not sufficient. So, SAP of introduced its own locking concept that is completely independent of the database system. SAP have created lock objects that enabled the system to lock data records in multiple database tables for the entire duration of the SAP transaction. Provided that they are linked, in the ABAP dictionary by a foreign key relationship. That we have already looked at. As we're not covering dialog programming in this course, we won't go into detail explaining the step-by-step procedures to follow for creating database locks. But we will continue with a brief overview of the topic before moving on. So, as I previously mentioned, SAP lock objects form the basis of the lock concept. And they are completely independent of database locks. Now a lock object allows you to lock a record for multiple tables at the same time. For the entire duration of an SAP transaction. And for this to work, the tables involved in the transaction must be linked together using foreign keys. If you remember. We introduced foreign keys in an earlier module. Now we used the ABAP dictionary to create lock objects. And they contain the tables and key fields that make up a shared lock. When we create the lock object, the system automatically creates two function modules which we will come to in a later module. These function modules are just simply modularized ABAP programs, that can be culled from other programs. Now the first function module has the action of setting a lock, and the second one releases the lock. And it is the programmer's responsibility to ensure that these function modules are called at the correct place within their programs. When the lock is set a lock record is created in the central lock table for the entire SAP system. Now all programs must adhere to using the SAP lock concept to ensure their programs set, delete and query the lock table that stores the lock records. For the relevant entries. We're not going to go into lock objects any more than this. And for the rest of the course, we will just assume that the programs you create and the tables you access are going to be exclusively used just by yourself.