

**Question 2c uses this context.**

The equipment table is generating inconsistent information.

Equipment					
AssetNumber	Category	AssetType	LastSafetyTest	Faulty	FaultNotes
Loan-Equip-0003	Display	DLP Projector	18 May 2017	Yes	Bulb to be replaced
Loan-Equip-0004	Display	DLP Projector	18 May 2017	Yes	HDMI socket not working - fixed 20March2017

**2c**

**Analyse** the inconsistent information and how the structure of the database allows these inconsistencies.

**Recommend** how the structure of the database could be improved to prevent inconsistencies.

Type your response in the space provided in the evidence template document.

(9)

**Sample response 1***Analysis of inconsistent information*

There is conflicting information in terms of recording faults. Loan-Equip-0004 has a tick in the Faulty field to signify a fault but the Notes field say that it was fixed on the 20 March 2017. It is impossible to determine whether the tick signifies a new fault and the notes for it have not been updated, or whether the user has forgotten to remove the tick from the faulty field.

*Analysis of how the structure of the database allows these inconsistencies*

The conflicting information is allowed because the fault information is not atomic and has not been normalised correctly. The FaultNotes field is not atomic as it is being used for more than one purpose – to say what the fault is and to identify when it was fixed. The information has not been normalised correctly as there should have been identification that one piece of equipment could have many faults. The current structure only allows for one fault per equipment with the user having to delete the information and replace when a new fault is added.

*Recommendations*

The problem could be prevented by creating a new table (tblFault). This could have the fields

- FaultID – Primary Key – Autonumber
- AssetNumber – Foreign Key
- FaultReportedDate
- FaultResolvedDate
- FaultNotes

The foreign key would ensure there is still a relationship between a piece of equipment and a fault, but it would no longer be a 1:1 relationship and would now correctly use a 1:M relationship meaning a piece of equipment can have many faults without the need to delete the existing fault information to do this. The field Faulty could be replaced with a FaultReportedDate and a FaultResolvedDate. This would allow clear identification of whether a fault has been resolved or not with no conflicts. The FaultNotes column would also now be atomic.