Solution 8.1 - Fill out the following table.

(Elmasri & Navathe: Chapter 30: 1)

(There are numerous answers for the following table. If you are unsure about the correctness of your answer, please check with a tutor.)

Threats to Databases	Explanation	Possible Causes	Possible Control Measures
Loss of Integrity	Database integrity extends from relations to the file-system, and trustworthiness of the data. Data must remain consistent with their integrity constraints, must not be modified by unauthorised users, and must be reliably written and fetched from disk.	**(1)** Information is modified to an inconsistent value. **(2)** If an unauthorised user accesses the system and makes changes to data. **(3)** If incorrect (or no) integrity constraints were set	**Flow control**: prevents information from flowing in such a way that it reaches unauthorised users. **Routine (encrypted) backups:** enable point in time recovery
Loss of Availability	If the database is unavailable to authorised users or the authorised users cannot perform a task which they have permission to do so. This includes any database administrators, and end-users (e.g. HR being unable to connect to complete timesheets, a website being unable to connect).	**(1)** If anything occurs which can cause unavailability such as: server failure, database corruption, network connectivity issues, hardware failure, or incorrect permissions assigned to objects. **(2)** More advanced issues include query blocking.	**Disaster Recovery Plan:** can have multiple instances of the same database run in parallel and kept synchronous. In event of a server failure the active server can be failed-over to the secondary. **Permissions:** Ensure correct permissions are set on objects for users/groups.
Loss of confidentiality	Unauthorized disclosure of confidential information	**(1)** Poor database design can cause sensitive information e.g. passwords stored in plain text, to be correlated with a person. **(2)** If authorised or unauthorised users can view information outside of their permission level this could also lead to sensitive information being breached.	**Data/ Backup encryption:** Encoded to provide additional protection. **Privilege:** setting correct privilege security levels. Permissions should always be set on a least-privilege basis (get the permission if you need it), **Auditing:** Auditing on sensitive tables should also be included to record who has viewed particular information.

Solution 8.2 - Using your knowledge of SQL Injection, what input can be entered so that the 'USERS' table is dropped?

Input: X"; DROP TABLE Users;

This will result in:

```
SELECT *
FROM USERS
WHERE USERNAME = "X";
DROP TABLE Users;
"
```

Explanation: The **"** after *X* is used to finish the string prematurely. This means the rest of the input is taken as an SQL command rather than a string and implemented on the database.

Solution 8.3 - Rewrite the UserID input value using an SQL Injection so that it selects information from all users.

105 OR 1=1

This will result in:

```
SELECT *
FROM USERS
WHERE UserID = 101 OR 1=1
```

Explanation: *OR 1=1* is interpreted as an additional part of the SQL query. Since *1=1* is always true, the query will select every user. Any condition which is always true can replace *1=1 in this solution*.