Homework 3

Short answers

1. A user is cleared for <secret>. Can he have access to data objects labeled <classified>? Why?

- Yes, this user should have access to <classified> data objects because his clearance level is higher than that. However, this user should not be permitted to write to data objects at this level, as he risks exposing information that is above this clearance level.

2. The user wants to pass a file to a user at <top secret> level, how could he do that?

- The user could add the file to the data base, it would have a level of <secret>, and the <top secret> level user would be able to access this file because they have a higher clearance than <secret>.

3. The user wants to pass a file to a user at <unclassified> level, how could he do that?

- With the current arrangement, this would not be possible. The <unclassified> user would have to be promoted to the <secret> level because the original user is cleared at a <secret> level, meaning anything they post to the database is also flagged with a <secret> level.

4. In a database access control mechanism, privilege *r* refers to read, and privilege *w* refers to write. User *A* is the owner of relation *R1*, user B is the owner of relation *R2*. The following sequence of actions are taken:

Step	By	Action
1	*A*	GRANT *R1.r* TO *B*, *C* WITH GRANT OPTION
2	*B*	GRANT *R1.w*, *R1.r* TO *C*, *D* WITH GRANT OPTION
3	*B*	GRANT *R2.w*, *R2.r* TO *C* WITH GRANT OPTION
4	*C*	GRANT *R1.r* TO *D*
5	*D*	GRANT *R1.r* TO C
6	*A*	GRANT *R2.r* TO C
7	*B*	REVOKE *R1.r* FROM *D* CASCADE
8	*A*	REVOKE *R1.r* FROM *B* CASCADE

Which of the above steps will be denied? Why?

1. A gives read access to R1 to B, C with grant
2. B gives read & write access to R1 to C, D with grant
3. B gives read & write access to R2 to C with grant
4. C gives read access to R1 to D
5. D gives read access to R1 to C
6. A gives read access to R2 to C
7. B takes read access to R1 from D and cascades
8. A takes read access to R1 from B and cascades

1. This step passes, A owns R1, so now B & C have read access with grant option on R1.
2. **This step fails** because B does not have write access to R1 so it cannot grant write permissions.
3. This step passes, B owns R2, so now C has read/write access with grant option on R2.
4. This step passes, C has read access with grant option on R1, so it can grant the same permission to D.
5. **This step fails**, D does have read access on R1, however, D does not have the grant option, so it cannot grant anyone permission to read this file.
6. **This step fails**, A does not own R2/have access to read R2, let alone grant a read permission to C.
7. This step passes. B has read access with grant option on R1. This revokes D's read access to R1 FROM B, but D may still be able to read R1 because it was given access from C (depends on the database system). Since the grant option was never given to D, the cascade stops here.
8. This step passes. B has read access with grant option on R1. This revokes B's read access to R1, and any users that B has given R1 read access to, which includes C, which, if D still had read access (again, based on the database system), would remove D's access as well.