**In your own words answer the following questions:**

1. What is a (database) transaction?

A database transaction represents a set of procedures which interact with the database and will often, but not always, modify its data.

1. Why do we need (database) transactions?

Database transactions are important in order to uphold the data integrity of a database. When the database is updated using a transaction, it will ensure the necessary tables are updated and adhere to the database structural rules. They also ensure that data integrity is upheld in the case of encountering an error.

1. What needs to happen when (whether due to a technical error, database constraint, business rule or other reason) a transaction can only be partially completed and why does this need to happen?

In this case, the transaction must be rolled back to prevent changes being made to only partial sets of the data. In Transact-SQL the command ROLLBACK TRANSACTION can be used in a catch-block to mitigate changes that had already been made during the transaction.

1. What keyword/ action tells a database that a transaction has been successfully completed and the results of the transaction can be made permanent in the database?

In Transact-SQL, the keyword for this is COMMIT TRANSACTION.

1. Give an example of a business operation that might require a database transaction (other than those given in the lecture) and explain WHY it would need the transaction

An example of a business operation is a soccer-league updating the results of each round. A transaction would be required as it must update the game scores one-by-one which requires the ladder to be updated as well as any statistics, such as goals and assists.

1. What does the acronym ACID stand for (in database terms)?

Provide a brief explanation of the meaning of each word in the Acronym.

ACID stands for Atomicity, Consistency, Isolation, and Durability.

* **Atomicity** refers to ensuring that if a transaction is to succeed, all processes must be successful. Otherwise the transaction should be reverted.
* **Consistency** guarantees that all changes to data will remain consistent and adhere to all defines rules of the database.
* **Isolation** guarantees that all transactions are processed in isolation from one another. No transaction can read data from another transaction that is not yet complete.
* **Durability** ensures that after a transaction is complete, the changes will remain in the database even in the event of a crash directly after the transaction has completed.