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

1. **What is a (database) transaction?**

A database transaction is viewed as a ‘logical unit of work,’ where all the operations of a transaction must be completed before it deemed as a successful transaction. This process means that a transaction can only be entirely completed or entirely aborted, steps within the transaction cannot be skipped or bypassed.

1. **Why do we need (database) transactions?**

We need Database Transactions due to data integrity. Database Transactions enables the implementation of business logic at Database level. This means that a consistent database state must be adhered, ie. A database is taken from one consistent database state to another via transaction. This consistent database state is a fundamental principle in programming as failure to adhere to these standards can result in not only serious problems but also be deemed as illegal.

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?**

Whether due to a technical error, database constraint, business rule or other reason, when a transaction is partially completed, then the transaction is entirely aborted, this abortion is because a transaction must either be entirely completed or entirely aborted.

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?**

The keyword/action that tells a database that a transaction has been successfully completed and the results of the transaction can be made permanent in the database is:

COMMIT.

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 that might require a database transaction can be a theatre booking system or a train ticket system.

These systems require transactions because a train ticket purchase or seat booking must be completed or aborted so that there is no double purchase or double booking in case of customer confusion.

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

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

In database terms, the acronym ACID stands for:

**Atomicity** - Where a transaction is indivisible, ie. Performed entirely or not at all.

- Implemented by COMMIT and ROLLBACK.

**Consistency** - Where a transaction transforms the database from one consistent state to another.

* Implemented by database integrity constraints.

**Isolation** - Where transactions that can work correctly by itself is guaranteed to work correctly when run

multiple times at the same time or with other transactions.

* Implemented by locking and/or multiversioning.

**Durability** – Ensures that the results of the committed transaction can never be lost even in the event of a

hardware, memory or system failure.

* Implemented by transaction logging and recovery.