Discuss the entity integrity and referential integrity constraints. Why is each considered important?

Entity integrity constraint: no primary key value can be NULL. Primary key value is used to identify individual tuples in a relation. Having NULL values for the primary keys implies some tuples cannot be identified. Example: if two or more tuples have NULL for their primary keys, they cannot be distinguished from each other if they are referenced from other relations. (p. 163)

Ensures there are no repetitions of data, helps retrieve data faster because each record has a unique primary key (<https://www.geeksforgeeks.org/difference-between-entity-constraints-referential-constraints-and-semantic-constraints/>)

Without a unique primary key, an individual tuple cannot be identified (<https://opendsa.cs.vt.edu/ODSA/Books/Database/html/RDDConstraints.html#:~:text=Advantage%20of%20Using%20Integrity%20Constraints,to%20a%20reliable%20database%20design>.)

**The entity integrity constraint states that no primary key can be NULL. This is because the primary key value is used to identify individual tuples in a relation. Tuples have a unique primary key, which allows distinguishable references between relations and efficient searching for specific records. Having NULL value implies that some tuples cannot be identified, resulting in indistinguishable references between relations, longer search times, and possible repetitions of data since there is no unique primary key value.**

Referential integrity constraint: specified between two relations and is used to maintain the consistency among tuples in the two relations. States that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation

Maintains data relationships and references between tables (<https://www.geeksforgeeks.org/difference-between-entity-constraints-referential-constraints-and-semantic-constraints/>)

Consistency among tuples in two relations (https://opendsa.cs.vt.edu/ODSA/Books/Database/html/RDDConstraints.html#:~:text=Advantage%20of%20Using%20Integrity%20Constraints,to%20a%20reliable%20database%20design.)

Allows for cascading updates or deletes as that deletion of records from a parent table, which possesses the primary key, results in deletion of the data from the child table, which possesses the foreign key. (<https://www.geeksforgeeks.org/difference-between-entity-constraints-referential-constraints-and-semantic-constraints/> and <https://www.ibm.com/docs/en/informix-servers/14.10?topic=integrity-referential>)

**The referential integrity constraint is one that is specified between two relations and states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. This constraint maintains data relationships and references between tables. It ensures that data is consistent across tables and helps prevent inaccurate data from being added to the database by checking that a foreign key in one relation corresponds to a primary key value in a tuple of another relation. Referential integrity also allows for cascading updates or deletes as that changes in records of relations containing the primary keys will also change records of relations that reference the primary keys in one of their foreign keys.**