Referential Integrity

(i.e. references between tables)

One of the goals of good database design is to ensure data in the database remains consistent. This means ensuring that:

* All redundant data is removed to avoid cases where conflicting values exist for the same thing.
* All associations implemented with foreign keys remain consistent at both ends. For example, a loan record does not exist without there being a corresponding user that it was loaned to.

The association implemented in the database with foreign keys must obey the cardinalities (multiplicities) specified in the class diagram (or logical database design).

To keep the database consistent during operation the following must be taken into account:

1. When a parent record’s primary key is modified, the associated foreign key field in all child records must be updated to the new value.
2. When a parent record is deleted the associated child records must also be deleted.

When creating foreign keys, SQL includes the following option to specify how foreign keys should be treated:

1. On update – specifies what should be done to the child foreign keys when a parent’s primary key value is changed
   * + Cascade: Child’s value is updated to the same as the parent’s
     + Set Null: Child’s value is set to null
     + Restrict: Prevents updates on parent field
2. On delete - specifies what should be done to the child foreign keys when a parent’s primary key value is deleted
   * Cascade: Child value is deleted when parent value is deleted
   * Set Null: Same as above
   * Restrict: Same as above