

Solution 1 - Using the above schema for the LIBRARY database, provide one example for each of the following:

(There are numerous examples of the following components. If you are unsure about the correctness of your answer, please check with a tutor.)

Component	Example
Superkey	<p><i>"BOOK.Book_id, Title"</i></p> <p>Explanation: For the table <i>BOOK</i>, the attributes <i>Book_id</i> and <i>Title</i> can be used to identify each unique tuple, hence making them a valid superkey. However, it can be noted that <i>Book_id</i> alone can be used as a superkey; this makes <i>Title</i> a redundant attribute. Hence, <i>"BOOK.Book_id, Title"</i> could not be a minimal key.</p>
Minimal Key	<p><i>"PUBLISHER.Name"</i></p> <p>Explanation: For <i>PUBLISHER</i>, the attribute <i>"Name"</i> is the minimal key. This means is because if we removed any part of the key it would no longer be able to identify each unique tuple.</p> <p>It can be noted that 'this minimality property is required for a minimal key but is optional for a superkey. Hence, a minimal key is a super key but not always vice versa.'</p>
Foreign Key	<p><i>BOOK_AUTHORS.Book_id</i> (as the arrow shows it references <i>BOOK.Book_id</i>)</p> <p>Explanation: <i>BOOK_AUTHORS.Book_id</i> is the child in the foreign key as it references data from the <i>BOOK</i> table, and thus inherits any data constraints. Whereas, <i>BOOK</i> is the parent table as all the <i>Book_ids</i> are stored in this table.</p> <p>Thus, any <i>Book_ids</i> in the <i>BOOK_AUTHORS</i> must reference a <i>Book_id</i> in the <i>BOOK</i> table or be NULL.</p>
Non-Primary-Key Attribute	<p><i>BORROWER.Phone</i></p> <p>Explanation: <i>BORROWER.Phone</i> is a non-primary-key attribute because it is not a part of the primary key for <i>BORROWER</i>, which is <i>"Card_no"</i>. This means it is not required in identifying each unique tuple.</p>

(Elmasri & Navathe: Chapter 5: 5.2)

Solution 2 - Suppose each of the following update operations is applied directly to the database below. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.

Solution 2a - Insert < 'Robert', 'F', 'Scott', 943775543, '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', 'M', 58000, '888665555', 1 > into EMPLOYEE.

No constraint violations.

Solution 2b - Insert < 'ProductA', 4, 'Bellaire', 2 > into PROJECT.

Violates referential integrity because DNUM=2 and there is no tuple in the DEPARTMENT relation with DNUMBER=2.

We may enforce the constraint by:

- i. rejecting the insertion of the new PROJECT tuple, or,
- ii. changing the value of DNUM in the new PROJECT tuple to an existing DNUMBER value in the DEPARTMENT relation, or,
- iii. inserting a new DEPARTMENT tuple with DNUMBER=2.

Solution 2c - Insert < 'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.

Violates both the key constraint and referential integrity.

Violates the key constraint because there already exists a DEPARTMENT tuple with DNUMBER=4. We may enforce this constraint by:

- i. rejecting the insertion, or
- ii. changing the value of DNUMBER in the new DEPARTMENT tuple to a value that does not violate the key constraint.

Violates referential integrity because MGRSSN='943775543' and there is no tuple in the EMPLOYEE relation with SSN='943775543'. We may enforce the constraint by:

- i. rejecting the insertion, or,
- ii. changing the value of MGRSSN to an existing SSN value in EMPLOYEE, or,
- iii. inserting a new EMPLOYEE tuple with SSN='943775543'.

Solution 2d - Insert < 677678989, null, '40.0' > into WORKS_ON.

Violates both the entity integrity and referential integrity.

Violates entity integrity because PNO, which is part of the primary key of WORKS_ON, is null. We may enforce this constraint by:

- i. rejecting the insertion, or,
- ii. changing the value of PNO in the new WORKS_ON tuple to a value of PNUMBER that exists in the PROJECT relation.

Violates referential integrity because ESSN='677678989' and there is no tuple in the EMPLOYEE relation with SSN='677678989'. We may enforce the constraint by:

- i. rejecting the insertion, or,
- ii. changing the value of ESSN to an existing SSN value in EMPLOYEE, or,
- iii. inserting a new EMPLOYEE tuple with SSN='677678989'.

Solution 2e - Delete the WORKS_ON tuples with ESSN= '333445555'.

No constraint violations.

Solution 2f - Delete the EMPLOYEE tuple with SSN= '987654321'.

Violates referential integrity because several tuples exist in the WORKS_ON, DEPENDENT, DEPARTMENT, and EMPLOYEE relations that reference the tuple being deleted from EMPLOYEE.

We may enforce the constraint by:

- i. rejecting the deletion, or,
- ii. deleting all tuples in the WORKS_ON, DEPENDENT, DEPARTMENT, and EMPLOYEE relations whose values for ESSN, ESSN, MGRSSN, and SUPERSSN, respectively, is equal to '987654321'.

Solution 2g - Modify the MGRSSN and MGRSTARTDATE of the DEPARTMENT tuple with DNUMBER= 5 to '123456789' and '01-OCT-88', respectively.

No constraint violations.

Solution 2h - Modify the SUPERSSN attribute of the EMPLOYEE tuple with SSN= '999887777' to '943775543'.

Violates referential integrity because the new value of SUPERSSN='943775543' and there is no tuple in the EMPLOYEE relation with SSN='943775543'.

We may enforce the constraint by:

- i. rejecting the deletion, or,
- ii. inserting a new EMPLOYEE tuple with SSN='943775543'.