

SOFTENG 351: Lab #3

April 4, 2020

Aiden Burgess

abur970 - 600280511

Q1

a)

Area code is missing. e.g. Auckland has area code 09.

b)

Add the information to the attributes, as these two pieces of information are not that useful by themselves. If these pieces of information are critical and at some point may be retrieved individually (without their respective phone numbers), then for efficiency they should be added as columns.

c)

Access to first, middle and last name is easier if they are in separate columns. Searching is also easier if names are in separate columns. There is less confusion when a person has several middle names or first names as it is contained within separate columns.

If the names are combined, retrieving the full name for a STUDENT is faster.

d)

Split the information if the segments have meaning. If the information is generally retrieved fully (such as a mobile number) then there is no need to split the information.

Q2

Book_# is a foreign key in the TEXT table, which references the BOOK_ADOPTION entity.

Course# in ENROLL table is a foreign key for a COURSE entity.

Course# in BOOK_ADOPTION table is a foreign key for a COURSE entity.

SSN is foreign key for the ENROLL entity.

Q3

a)

No integrity constraints violated.

b)

Dnum references a DEPARTMENT with Dnumber 2. No DEPARTMENT exists with Dnumber 2, so a referential integrity constraint is violated.

Ways to fix:

- Ignore this transaction and provide error message.
- Before this transaction, add a DEPARTMENT with Dnumber of 2.
- Use a Dnumber that exists in the DEPARTMENT table.

c)

There already exists a DEPARTMENT with Dnumber 4, as Dnumber is a key attribute, this violates the key integrity constraint.

Mgr_ssn references an EMPLOYEE with Ssn of 943775543. No EMPLOYEE exists with Ssn 943775543, so a referential integrity constraint is violated.

Ways to fix both:

- Ignore this transaction and provide error message.

Ways to fix key integrity:

- Use a Dnumber that is unique.

Ways to fix referential integrity:

- Use a Ssn of an existing EMPLOYEE.
- Before this transaction, add a new EMPLOYEE with Ssn of 943775543.

d)

Inserting null into the Pno attribute column violates the key integrity constraint, as the Pno attribute is part of the key.

Essn references an EMPLOYEE with Ssn of 677678989. No EMPLOYEE exists with Ssn of 677678989, so referential integrity is also violated.

Ways to fix both:

- Ignore this transaction and provide error message.

Ways to fix key integrity:

- Use a Pno of an existing PROJECT.

Ways to fix referential integrity:

- Use a Ssn of an existing EMPLOYEE.
- Before this transaction, add a new EMPLOYEE with Ssn of 677678989.

e)

No integrity constraints violated.

f)

No integrity constraints violated.

g)

This EMPLOYEE row is referenced in the EMPLOYEE, DEPARTMENT, WORKS_ON, and DEPENDENT tables. Therefore, the deletion of this row will violate referential integrity constraints.

Ways to fix:

- Ignore this transaction and provide error message.
- Delete row that references this EMPLOYEE in EMPLOYEE, DEPARTMENT, WORKS_ON, and DEPENDENT tables, and cascade this change to all other affected rows/tables.

h)

This PROJECT Pnumber row is referenced in the WORKS_ON table. Therefore, the deletion of this row will violate referential integrity constraints.

Ways to fix:

- Ignore this transaction and provide error message.
- Delete all rows in WORKS_ON that reference the Pnumber (1) of 'ProjectX'.

i)

No integrity constraints violated.

j)

There does not exist an EMPLOYEE with Ssn 943775543. Therefore, this modification transaction violates a referential integrity constraint.

Ways to fix:

- Ignore this transaction and provide error message.
- Before this transaction, add a new EMPLOYEE with Ssn of 943775543.

k)

No integrity constraints violated.

Q4

