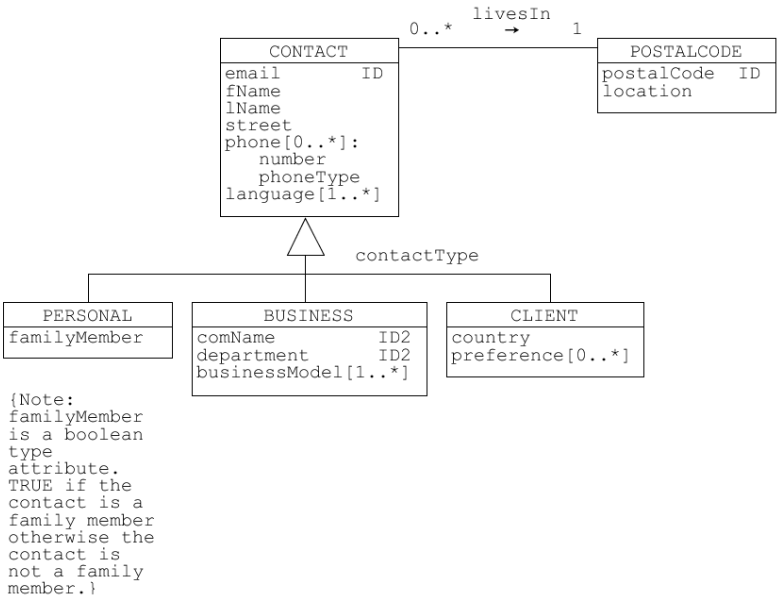
Analysis of relational schemas and normalization

Consider the following conceptual schema of a sample database domain that contains information about ‘contact’.



(i) Simplify the conceptual schema by removing the generalization/specialization association. You may use either the superset, subset, or association method to simplify the schema.

(ii) Find all functional and multivalued dependencies in the relational schemas of the simplified conceptual schema.

(iii) Find all minimal super keys in all the relational schemas. Justify your answer.

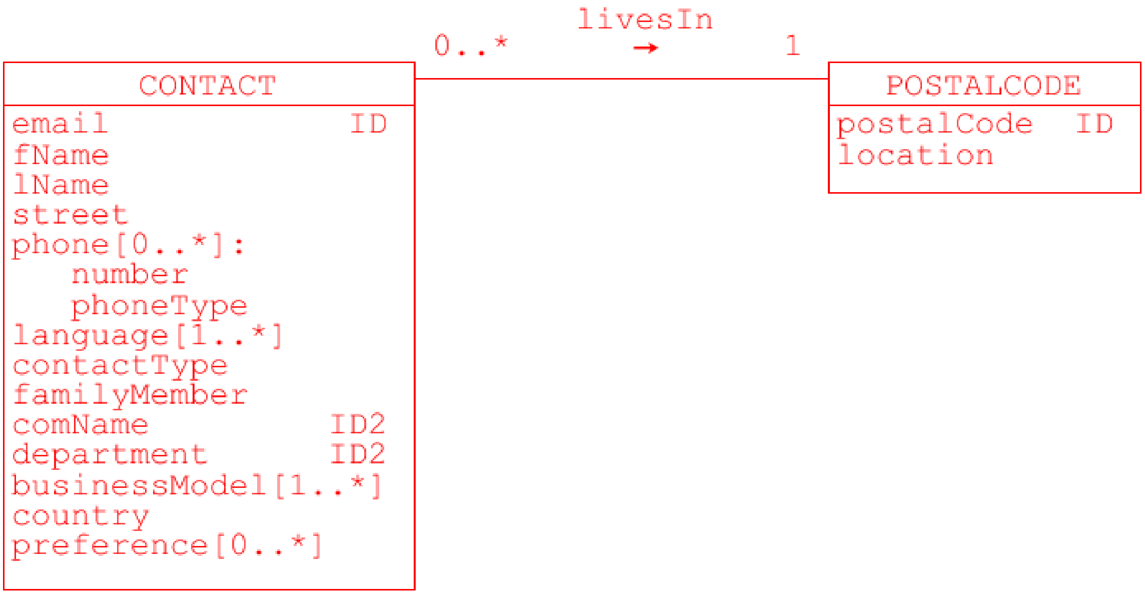
(iv) For each one of the relational schemas, find the highest normal form a schema is in. List the justifications for each highest normal form found.

(v) Decompose all relational schemas that are not in 4NF into 4NF. List all relational schemas obtained from the decompositions. Remember to indicate the primary key and foreign keys (if any).

Sample solution includes but is not limited to the following:

(i) Simplify the conceptual schema by removing the generalization/specialization association. You may use either the superset, subset, or association method to simplify the schema.

I choose to simplify the generalization/specialization using superset method.



(ii) Find all functional and multivalued dependencies in the relational schemas of the simplified conceptual schema.

CONTACT:

$email \rightarrow fName, lName, street, contactType, familyMember, comName, department, country$

$comName, department \twoheadrightarrow phone.number, phone.phoneType, language, businessModel, preference$

POSTALCODE:

$postalCode \rightarrow location$

(iii) Find all minimal super keys in all the relational schemas. Justify your answer.

CONTACT:

Closure of attribute $\{email\}^+$:
$\{email\}^+ = \{email, fName, lName, street, familyMember, comName, department, country\}$

Closure of attribute $\{email, phone.number\}^+$:
$\{email, phone.number\}^+ = \{email, phone.number, phone.phoneType\}$

Closure of attribute $\{email, language\}^+$:
$\{email, language\}^+ = \{email, language\}$

Closure of attribute $\{email, businessModel\}^+$:
$\{email, businessModel\}^+ = \{email, businessModel\}$

Closure of attribute $\{email, preference\}^+$:
$\{email, preference\}^+ = \{email, preference\}$

Since none of the determinant's closure set derives all the attributes, hence, we combine all the closure attributes, using composition rule, and get the closure:

$\{email, phone.number, language, businessModel, preference\}^+ = \{email, fName, lName, street, familyMember, comName, department, country, phone.number, phone.Type, language, businessModel, preference\}$

The closure of attributes $\{email, phone.number, language, busniessModel, preference\}^+$ derives all the attributes, hence, $\{email, phone.number, language, busniessModel, preference\}$ is a minimal super key (or candidate key) of the relational schema CONTACT.

POSTALCODE:

Closure of attribute $\{postalCode\}^+$:
$\{postalCode\}^+ = \{postalCode, location\}$

Since the closure of attribute $\{postalCode\}^+$ derives all the attributes of the relational schema POSTALCODE, hence, $\{postalCode\}$ is a minimal super key (or candidate key) of the relational schema POSTALCODE.

(iv) For each one of the relational schemas, find the highest normal form a schema is in. List the justifications for each highest normal form found.

The relational table CONTACT has the following functional dependencies:

$$(email, phone.number, language, businessModel, preference) \rightarrow (email, fName, lName, street, familyMember, comName, department, country, phone.number, phone.Type, language, businessModel, preference)$$

$$(email, phone.number) \rightarrow (phone.phoneType)$$

$$(email, businessModel) \rightarrow (email, businessModel)$$

$$(email, preference) \rightarrow (email, preference)$$

There exist multi-valued attributes phone, businessModel, language, and preference in the relational schema CONTACT. Existence of multi-value attributes violates the requirement to be in 1NF. Hence, the relational schema CONTACT is in 0NF, that is, not in any of the valid normal forms.

The relational schema POSTALCODE has the following functional dependency:

$$(postalCode) \rightarrow location$$

- There is no multi-value attribute in relational schema POSTALCODE, hence, the relational schema POSTALCODE is in 1NF.
- There is no partial dependency in relational schema POSTALCODE. Hence, the relational schema POSTALCODE is in 2NF.
- There is no transitive dependency in relational schema POSTALCODE. Hence, the relational schema POSTALCODE is in 3NF.
- There is no non-trivial dependency in relational schema POSTALCODE. Hence, the relational schema POSTALCODE is in BCNF.

(v) Decompose all relational schemas that are not in 4NF into 4NF. List all relational schemas obtained from the decompositions. Remember to indicate the primary key and foreign keys (if any).

To decompose the relational schema CONTACT to 4NF, we perform the following series of operations:
Bring the relational schema CONTACT to 1NF by removing the respective multi-valued attributes. Each of the multi-valued attributes that are removed from the relational schema CONTACT, a new separate relational schema is created.

The respective relational schemas are as follows:
PHONE (email, number, phoneType)
FD: $(email, number) \rightarrow phoneType$

LANGUAGE (email, language)

FD: $(email, language)$

PREFERENCE (email, preference)
FD: $(email, preference)$

BUSINESSMODEL (email, businessModel)
FD: $(email, businessModel)$

CONTACT (email, fName, lName, street, contactType, familyMember, comName, department, country)
FD: $(email) \rightarrow fName, lName, street, contactType, familyMember, comName, department, country$

POSTALCODE (postalCode, location)
FD: $postalCode \rightarrow location$

The normalized form:

PHONE (email, number, phoneType)
FD: $(email, number) \rightarrow phoneType$

- There is no multi-valued attribute in the relational schema PHONE, hence, the relational schema is in 1NF.
- There is no partial functional dependency in the relational schema PHONE, hence, the relational schema is in 2NF.
- There is no transitive functional dependency in the relational schema PHONE, hence, the relational schema is in 3NF.
- There is no non-trivial functional dependency in the relational schema PHONE, hence, the relational schema is in BCNF.
- The multi-valued attribute 'number' in the relational schema PHONE is resolved by compositing it with the attribute 'email' and there is no multi-value functional dependency in the relational schema PHONE, hence, the relational schema is in 4NF.

LANGUAGE (email, language)
FD: $(email, language)$

- The multi-valued attribute 'language' is resolved by compositing it to the attribute 'email' to form a single instance value, hence, the relational schema LANGUAGE is in 1NF.
- The relational schema LANGUAGE has no partial functional dependency, no transitive functional dependency, and no non-trivial functional dependency; hence, the relational schema LANGUAGE is in BCNF.
- The multi-valued attribute 'language' in the relational schema LANGUAGE is resolved by compositing it with the attribute 'email' and

there is no multi-value functional dependency in the relational schema LANGUAGE, hence, the relational schema is in 4NF.

PREFERENCE (email, preference)
FD: $(email, preference)$

- The multi-valued attribute ‘preference’ is resolved by compositing it to the attribute ‘email’ to form a single instance value, hence, the relational schema PREFERENCE is in 1NF.
- The relational schema PREFERENCE has no partial functional dependency, no transitive functional dependency, and no non-trivial functional dependency; hence, the relational schema PREFERENCE is in BCNF.
- The multi-valued attribute ‘preference’ in the relational schema PREFERENCE is resolved by compositing it with the attribute ‘email’ and there is no multi-value functional dependency in the relational schema PREFERENCE, hence, the relational schema is in 4NF.

BUSINESSMODEL (email, businessModel)
FD: $(email, businessModel)$

- The multi-valued attribute ‘businessModel’ is resolved by compositing it to the attribute ‘email’ to form a single instance value, hence, the relational schema BUSINESSMODEL is in 1NF.
- The relational schema BUSINESSMODEL has no partial functional dependency, no transitive functional dependency, and no non-trivial functional dependency; hence, the relational schema BUSINESSMODEL is in BCNF.
- The multi-valued attribute ‘businessModel’ in the relational schema BUSINESSMODEL is resolved by compositing it with the attribute ‘email’ and there is no multi-value functional dependency in the relational schema BUSINESSMODEL, hence, the relational schema is in 4NF.

CONTACT (email, fName, lName, street, contactType, familyMember, comName, department, country)
FD: $(email) \rightarrow fName, lName, street, contactType, familyMember, comName, department, country$

- All the attributes in the relational schema CONTACT are atomic, hence, the relational schema CONTACT is in 1NF.
- The relational schema CONTACT has no partial functional dependency, no transitive functional dependency, and no non-trivial functional dependency; hence, the relational schema CONTACT is in BCNF.

POSTALCODE (postalCode, location)
FD: $postalCode \rightarrow location$

- All the attributes in the relational schema POSTALCODE are atomic, hence, the relational schema POSTALCODE is in 1NF.
- The relational schema POSTALCODE has no partial functional dependency, no transitive functional dependency, and no non-trivial functional dependency; hence, the relational schema POSTALCODE is in BCNF.

All the normalized tables:
CONTACT (email, fName, lName, street, contactType, familyMember, comName, department, country)
PK: email
CK: (comName, department)

POSTALCODE (postalCode, location, email)
PK: postalCode
FK: email reference CONTACT (email)

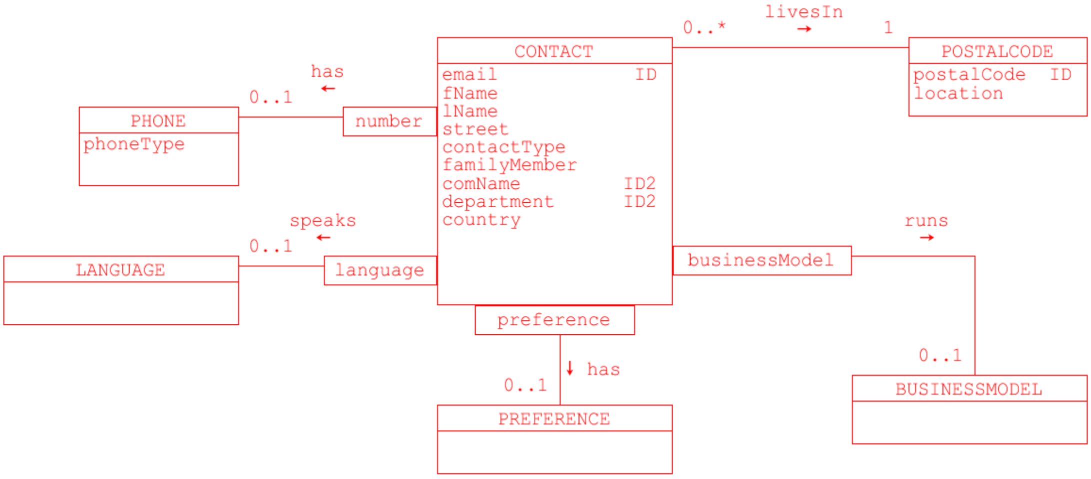
PHONE (email, number, phoneType)
PK: (email, number)
FK: email references CONTACT (email)

LANGUAGE (email, language)
PK: (email, language)
FK: email references CONTACT (email)

PREFERENCE (email, preference)
PK: (email, preference)
FK: email references CONTACT (email)

BUSINESSMODEL (email, businessModel)
PK: (email, businessModel)
FK: email references CONTACT (email)

The normalized relational schema is depicted in the following conceptual diagram.


CONTACT
email ID
fName
lName
street
contactType
familyMember
comName ID2
department ID2
country
livesIn →
0..*
1
POSTALCODE
postalCode ID
location
has ←
0..1
PHONE
phoneType
number
speaks ←
0..1
LANGUAGE
language
runs →
businessModel
0..1
BUSINESSMODEL
preference
↓ has
0..1
PREFERENCE