Research Higher Degree Logical Model

Prepared by:  
Thursdays **Group 4**

Sam Deane dean0109

Andrew Zschorn zsch0003

Version 1.0

15/5/2014

Created as part of the requirements for Advanced Database GE 2014

Contents

[1. Derive relations 1](#_Toc387938467)

[1.1. Conceptual E-R diagram 1](#_Toc387938468)

[1.2. Strong Entity types 2](#_Toc387938469)

[1.3. Weak Entity types 2](#_Toc387938470)

[1.4. One-to-many binary relationships 3](#_Toc387938471)

[1.5. One-to-one binary relationships 4](#_Toc387938472)

[1.5.1. Mandatory on both sides 4](#_Toc387938473)

[1.5.2. Mandatory on one side 4](#_Toc387938474)

[1.5.3. Optional on both sides 4](#_Toc387938475)

[1.6. One-to-one recursive relationships 4](#_Toc387938476)

[1.7. Superclass/subclass relationships 4](#_Toc387938477)

[1.8. Many-to-many binary relationship types 5](#_Toc387938478)

[1.8.1. University Staff Member *works in* Research Area 5](#_Toc387938479)

[1.8.2. University Staff Member *oversees* Research Area 5](#_Toc387938480)

[1.8.3. Application *in* Research Area 6](#_Toc387938481)

[1.8.4. University Staff Member *flags* Application 6](#_Toc387938482)

[1.9. Complex relationship types 6](#_Toc387938483)

[1.10. Multi-valued attributes 7](#_Toc387938484)

[1.11. Document relations and foreign key attributes 7](#_Toc387938485)

[2. Normalisation 8](#_Toc387938486)

[2.1. Lookup relations 10](#_Toc387938487)

[2.1.1. Lookup relationships 11](#_Toc387938488)

[2.2. Normalised Relations 12](#_Toc387938489)

[2.3. Logical E-R Diagram 14](#_Toc387938490)

[3. User transaction validation 15](#_Toc387938491)

[3.1. Transaction pathways 16](#_Toc387938492)

[4. Check integrity constraints 17](#_Toc387938493)

[4.1. Required data 17](#_Toc387938494)

[4.2. Attribute domain constraints 17](#_Toc387938495)

[4.3. Multiplicity 17](#_Toc387938496)

[4.4. Entity integrity 17](#_Toc387938497)

[4.5. Referential integrity 18](#_Toc387938498)

[4.5.1. Null attributes (~insert rules) 18](#_Toc387938499)

[4.5.2. Updates & Deletes 18](#_Toc387938500)

[4.6. General constraints. 18](#_Toc387938501)

[4.7. Document all integrity constraints 18](#_Toc387938502)

[5. Review logical data model 19](#_Toc387938503)

[6. Check for future growth 19](#_Toc387938504)

[7. Develop Test Plan 20](#_Toc387938505)

[7.1. Populate staff members 20](#_Toc387938506)

[7.2. Populate lookup tables 21](#_Toc387938507)

[7.2.1. Populate Country 24](#_Toc387938508)

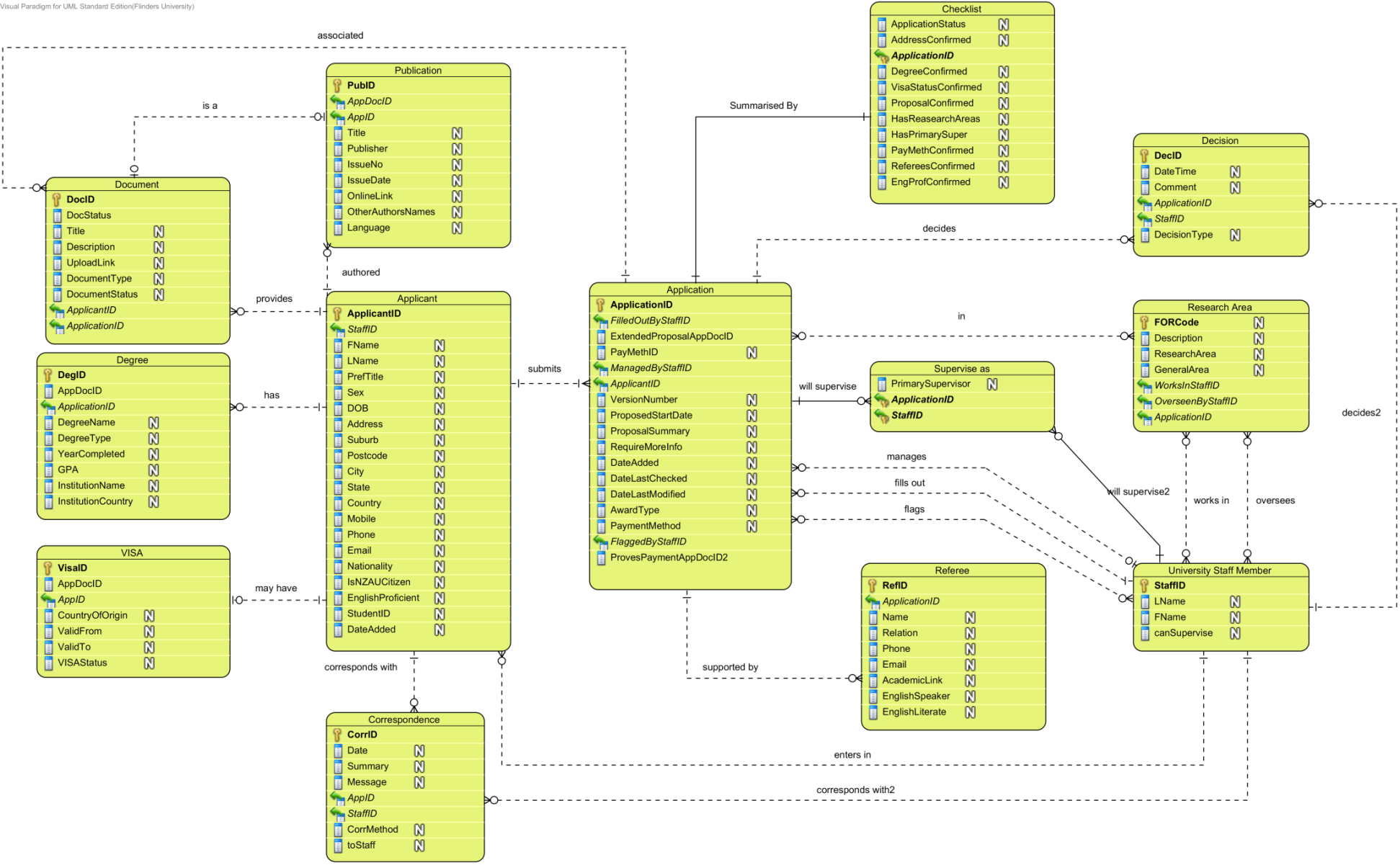
[7.3. Populate Applicant and Application 33](#_Toc387938509)

[7.4. Test user transactions 38](#_Toc387938510)

[8. Data dictionary 42](#_Toc387938511)

# Derive relations

## Conceptual E-R diagram



## Strong Entity types

|  |
| --- |
| **Applicant** (ApplicantID, FName, LName, PrefTitle, Sex, DOB, StreetAddress, Suburb, Postcode, City, State, Country, Mobile, Phone, Email, Nationality, isNZAUCitizen, EnglishProficient, StudentID, DateAdded)  **Primary key** ApplicantID |
| **Application** (ApplicationID, ProposedStartDate, ProposalSummary, RequireMoreInfo, DateAdded, DateLastChecked, DateLastModified, AwardType, flindersCampus, fullTime, PaymentMethod)  **Primary key** ApplicationID |
| **Correspondence** (CorrID, Date, Summary, Message, CorrMeth, toStaff)  **Primary key** CorrID |
| **Decision** (DecID, DateTime, Comment, DecisionType)  **Primary key** DecID |
| **Degree** (DegID, DegreeName, DegreeType, YearCompleted, GPA, InstitutionName, InstituitonCountry)  **Primary key** DegID |
| **Document** (DocID, UploadLink, DocStatus, DocumentType, Title, Description)  **Primary key** UploadLink |
| **Publication** (PubID, ApplicantID, Title, Publication, IssueNo, IssueDate, OnlineLink, OtherAuthorsNames, Language)  **Primary key** PubID |
| **Referee** (RefID, Name, Relation, Phone, Email, AcademicLink, EnglishSpeaker, EnglishLiterate)  **Primary key** RefID |
| **Research Area** (FORCode, Description, ResearchArea, GeneralArea)  **Primary key** FORCode |
| **University Staff Member** (StaffID, FName, LName, canSupervise)  **Primary key** StaffID |
| **Visa** (VisaID, VISAStatus, CountryOfOrigin, ValidFrom, ValidTo)  **Primary key** VisaID |

## Weak Entity types

|  |
| --- |
| **Checklist** (ApplicationStatus, AddressConfirmed, DegreeConfirmed, VisaStatusConfirmed, ProposalConfirmed, HasResearchArea, HasPrimarySuper, PayMethConfirmed, RefrereesConfirmed, EngProfConfirmed)  **Primary key** None (at this stage) |
| **Supervise as** (PrimarySupervisor)\*  **Primary key** None (at this stage) |

\* This entity was created during the conceptual phase to remove a relationship attribute

Very few natural alternate keys were found as many attributes are nullable due the limited information received at the time of initial user entry.

## One-to-many binary relationships

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| associated | No | Application | 0..1 | ApplicationID | Document | 0..\* |
| authored | No | Applicant | 1..1 | ApplicantID | Publication | 0..\* |
| corresponds with | No | Application | 1..1 | ApplicationID | Correspondence | 0..\* |
| corresponds with2 | No | University Staff Member | 1..1 | StaffID | Correspondence | 0..\* |
| decides | No | Application | 1..1 | ApplicationID | Decision | 0..\* |
| decides2 | No | University Staff Member | 1..1 | StaffID | Decision | 0..\* |
| has | Yes | Applicant | 1..1 | ApplicantID | Degree | 0..\* |
| last to modify |  | University Staff Member | 1..1 | StaffID → LastToModifyStaffID | Applicant | 0..\* |
| last to update | No | University Staff Member | 1..1 | StaffID →LastToModifyStaffID | Application | 0..\* |
| manages | No | University Staff Member | 0..1 | StaffID → ManagedByStaffID | Application | 0..\* |
| provides | No | Applicant | 1..1 | ApplicantID | Document | 0..\* |
| submits | Yes | Applicant | 1..1 | ApplicantID | Application | 0..\* |
| supported by | No | Application | 1..1 | ApplicationID | Referee | 1 |
| will supervise | Yes | Application | 1..1 | ApplicationID | Supervise as | 0..\* |
| will supervise2 | Yes | University Staff Member | 1..1 | StaffID | Supervise as | 0..\* |

## One-to-one binary relationships

### Mandatory on both sides

For the **Application *summarised by* Checklist** relationship, we bring all the attributes of **Checklist** into **Application**. The primary key remains ApplicationID; **Checklist** was a weak entity and never had a primary key. The **Checklist** relation is dropped.

|  |
| --- |
| **Application** (ApplicationID, ApplicationStatus, AddressConfirmed, DegreeConfirmed, VisaStatusConfirmed, ProposalConfirmed, HasResearchAreas, HasPrimarySuper, PayMethConfirmed, EngProfConfirmed, RefereesConfirmed, LastToModifyStaffID, ProposedStartDate, ProposalSummary, RequireMoreInfo, DateAdded, DateLastChecked, DateLastModified, AwardType, PaymentMethod, flindersCampus, fullTime)  **Primary key** ApplicationID |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| Summarised by | Yes | Application | 1 | ApplicationID | Checklist | 1 |

### Mandatory on one side

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| may have | No | Applicant | 1 | ApplicantID | VISA | 0..1 |

### Optional on both sides

None in this case.

## One-to-one recursive relationships

None in this case.

## Superclass/subclass relationships

None in this case.

## Many-to-many binary relationship types

### University Staff Member *works in* Research Area

Introduce the following relation:

|  |
| --- |
| **University Staff Member\_Research Area** (StaffID, FORCode)  **Primary key** StaffID, FORCode  **Foreign key** StaffID **references** University Staff Member(StaffID)  **Foreign key** FORCode **references** Research Area(FORCode) |

Introduce the following relationships:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| works in | Yes | University Staff Member | 1 | StaffID | University Staff Member\_Research Area | 0..\* |
| works in2 |  | Research Area | 1 | FORCode | University Staff Member\_Research Area | 0..\* |

### University Staff Member *oversees* Research Area

Introduce the following relation:

|  |
| --- |
| **University Staff Member\_Research Area2** (StaffID, FORCode)  **Primary key** StaffID, FORCode  **Foreign key** StaffID **references** University Staff Member(StaffID)  **Foreign key** FORCode **references** Research Area(FORCode) |

Introduce the following relationships:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| Oversees | Yes | University Staff Member | 1 | StaffID | University Staff Member\_Research Area2 | 0..\* |
| Oversees2 | Yes | Research Area | 1 | FORCode | University Staff Member\_Research Area2 | 0..\* |

### Application *in* Research Area

Introduce the following relation:

|  |
| --- |
| **Application\_Research Area** (ApplicationID, FORCode)  **Primary key** ApplicationID, FORCode  **Foreign key** ApplicationID **references** Application(ApplicationID)  **Foreign key** FORCode **references** Research Area(FORCode) |

Introduce the following relationships:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| in |  | Application | 1 | ApplicationID | Application\_Research Area | 1..\* |
| in2 |  | Research Area | 1 | FORCode | Application\_Research Area | 0..\* |

### University Staff Member *flags* Application

Introduce the following relation:

|  |
| --- |
| **University Staff Member\_Application** (StaffID, ApplicationID)  **Primary key** StaffID, ApplicationID  **Foreign key** StaffID **references** University Staff Member(StaffID)  **Foreign key** ApplicationID **references** Application(ApplicationID) |

Introduce the following relationships:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| flags | Yes | University Staff Member | 1 | StaffID | University Staff Member\_Application | 0..\* |
| flags2 | Yes | Application | 1 | ApplicationID | University Staff Member\_Application | 0..\* |

## Complex relationship types

None in this case

## Multi-valued attributes

None in this case

## Document relations and foreign key attributes

Since the relationships ‘will supervise’ and ‘will supervise 2’ have been defined in section 1.3, foreign keys are introduced into the weak entity ‘supervise as’ such that

|  |
| --- |
| **Supervise as** (PrimarySupervisor, StaffID, ApplicationID)  **Primary key** StaffID, ApplicationID  **Foreign key** StaffID **references** University Staff Member(StaffID)  **Foreign key** ApplicationID **references** Application(ApplicationID) |

To save space, a full list of intermediate relationships are not listed here, instead please see section 2.2 for a full list of normalised relations.

# Normalisation

|  |  |  |
| --- | --- | --- |
| **Relation** | **Functional Dependencies** | **Remarks** |
| Applicant | ApplicantID → FName, LName, PrefTitle, Sex, DOB, StreetAddress, Suburb, Postcode, City, State, Country, Mobile, Phone, Email, Nationality, isNZAUCitizen, EnglishProficient, StudentID, DateAdded, LastToModifyStaffID  FName, LName, DOB, StreetAddress, Suburb, Postcode, City, State, Country → ApplicantID, PrefTitle, Mobile, Phone, Email, Nationality, isNZAUCitizen, EnglishProficient, StudentID, DateAdded, LastToModifyStaffID  FName, LName, DOB, Email → ApplicantID, PrefTitle, StreetAddress, Suburb, Postcode, City, State, Country Mobile, Phone, Nationality, isNZAUCitizen, EnglishProficient, StudentID, DateAdded, LastToModifyStaffID | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Application | ApplicationID → ApplicantID, AddressConfirmed, DegreeConfirmed, VisaStatusConfirmed, ProposalConfirmed, HasResearchAreas, HasPrimarySuper, PayMethConfirmed, EngProfConfirmed, RefereesConfirmed, LastToModifyStaffID, ProposalDocID, AwardType, ManagedByStaffID, ProposedStartDate, ProposedSummary, RequireMoreInfo, DateAdded, DateLastChecked, DateLastModified, PayDocID, PaymentMethod, ApplicationStatus, flindersCampus, fullTime | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Correspondence | CorrID → Date, Summary, Message, ApplicationID, StaffID, CorrMethID, toStaff | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Decision | DecID → Date, Comment, ApplicationID, StaffID | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Degree | DegID →ApplicantID, Name, Type, YearCompleted, GPA, InstitutionName, InstituitonCountry  ApplicantID, Name → DegID, Type, YearCompleted, GPA, InstitutionName, InstituitonCountry | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Document | DocID → UploadLink, DocStatus, DocumentType, Title, Description, ApplicantID, ApplicationID  UploadLink → DocID, DocStatus, DocumentType, Title, Description, ApplicantID, ApplicationID | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Publication | PubID → ApplicantID, DocID, Title, Publication, IssueNo, IssueDate, OnlineLink, OtherAuthorsNames, Language | Attributes Publication, IssueNo, IssueDate, OnlineLink and OtherAuthorsNames are optional, therefore should not be considered as candidate keys.  No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Referee | RefID → ApplicationID, Name, Relation, Phone, Email, Profession, AcademicLink, EnglishSpeaker, EnglishLiterate | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| ResearchArea | FORCode → Description, ResearchArea, GeneralArea | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| University Staff Member | StaffID → LName, FName, canSupervise | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |
| Visa | VisaID → ApplicantID, VisaStatus, OriginCountry, ValidFrom, ValidTo | No repeating groups.  No partial dependencies.  No transitive dependencies.  Primary key fully determines non-primary-key attributes  ⇒ in 3NF |

## Lookup relations[[1]](#footnote-1)

In this schema there are several attributes with domains that allow only a small set of possible values. These attributes are Applicant(Nationality), Applicant(Country), Application(ApplicationStatus), Application(PaymentMethod), Document(DocumentStatus), Document(DocumentType), Visa(VisaStatus), Decision(DecisionType), Correspondence(CorrespondencMethod), and Award(AwardType).

In order to reduce the chance of update anomalies from keying-in incorrect values, we choose to create a lookup relation for each of these. In all cases, these relations are in one-to-many relationships with the relations containing their corresponding attributes. So, in each case, the lookup relation is designated the parent relation, the other relation is designated the child, and the primary key of the lookup table is posted to the child relation in place of the attribute.

This approach also has the benefit that users can: alter the associated description/types easily; or extend these domains to cover new cases as they arise.

|  |
| --- |
| **ApplicationStatus** (Status, Description)  **Primary key** Status |
| **Decision Type** (Type)  **Primary key** Type |
| **CorrespondenceMethod** (method)  **Primary key** Method |
| **AwardType** (Type, Description, Method)  **Primary key** Type |
| **Payment Method** (Method)  **Primary key** Method |
| **DocumentStatus** (Status, Description)  **Primary key** Status |
| **DocumentType** (Type, Description)  **Primary key** Type |
| **Visa Status** (Status, Description)  **Primary key** Status |
| **Country** (CountryISOCode, Name)  **Primary key** CountryISOCode |

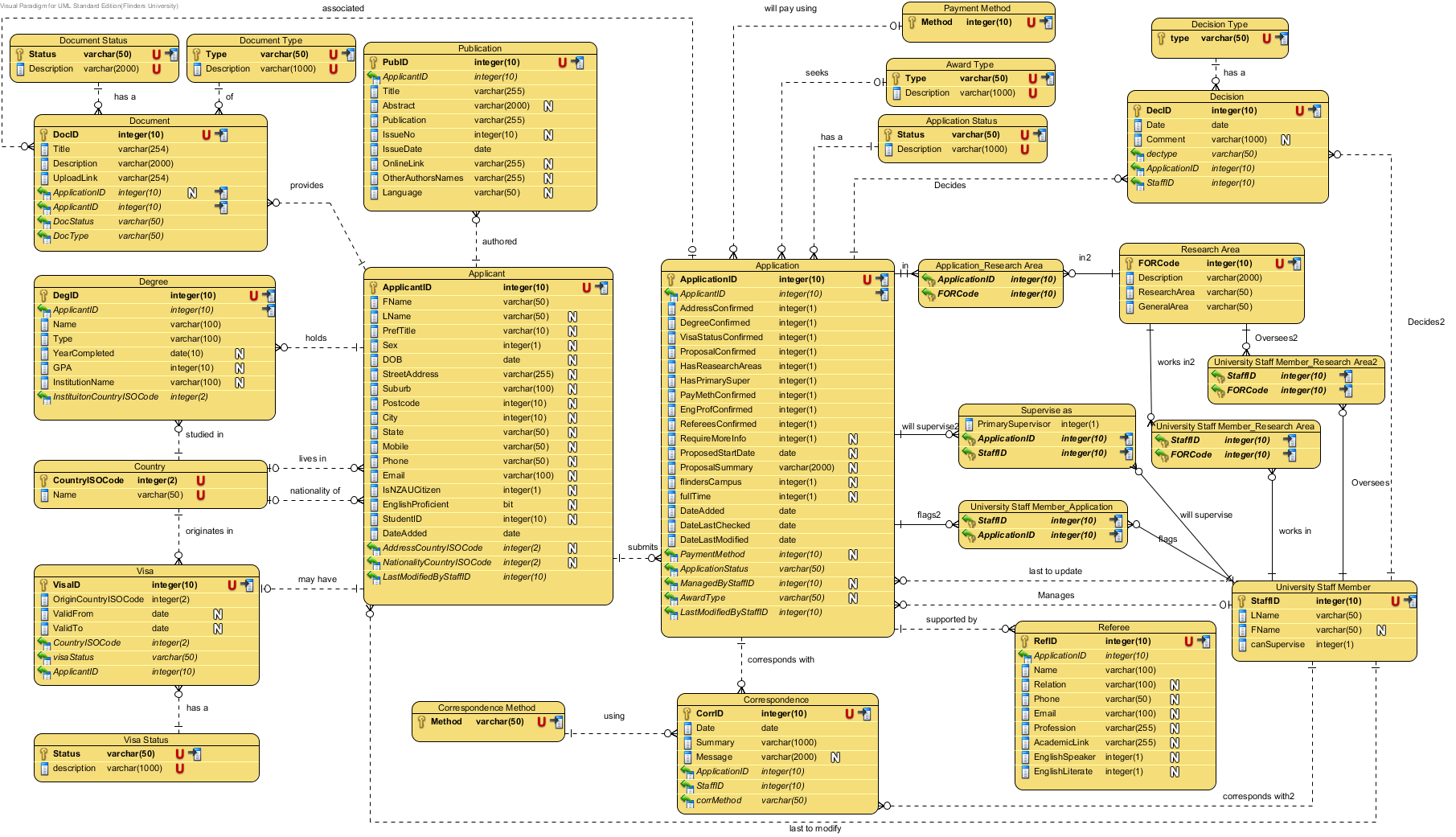
### Lookup relationships[[2]](#footnote-2)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Relationship name** | **Identifying** | **Parent relation** | **Parent multiplicity** | **Foreign Keys** | **Child relation** | **Child multiplicity** |
| lives in | No | Country | 1..1 | CountryISOCode→ AddressCountryISOCode | Applicant | 0..\* |
| nationality of | No | Country | 0..1 | CountryISOCode→ NationalityCountryISOCode | Applicant | 0..\* |
| has a | No | Visa Status | 1..1 | VisaStatus | Visa | 0..\* |
| originates in | No | Country | 1..1 | CountryISOCode → OriginCountryISOCode | Visa | 0..\* |
| has a | No | Document Status | 1..1 | Status → DocStat | Document | 0..\* |
| of | No | Document Type | 1..1 | Type → DocType | Document | 0..\* |
| will pay using | No | Payment Method | 0..1 | Method →paymentMethod | Application | 0..\* |
| has a | No | Application Status | 1..1 | Status → applicationStatus | Application | 0..\* |
| has a | No | Decision Type | 1..1 | type → decType | Decision | 0..\* |

## Normalised Relations

|  |
| --- |
| **Applicant** (ApplicantID, FName, LName, PrefTitle, Sex, DOB, StreetAddress, Suburb, Postcode, City, State, AddressCountryISOCode, Mobile, Phone, Email, NationalityCountryISOCode, isNZAUCitizen, EnglishProficient, StudentID, DateAdded, LastToModifyStaffID)  **Primary key** ApplicantID  **Alternate key** (FName, LName, DOB, StreetAddress, Suburb, Postcode, City, State, Country)  **Alternate key** (FName, LName, DOB, Email)  **Foreign key** LastToModifyStaffID **references** University Staff Member(StaffID)  **Foreign key** AddressCountryISOCode **references** Country(CountryISOCode)  **Foreign key** NationalityCountryISOCode **references** Country(CountryISOCode) |
| **Application** (ApplicationID, applicationStatus, AddressConfirmed, DegreeConfirmed, VisaStatusConfirmed, ProposalConfirmed, HasResearchAreas, HasPrimarySuper, PayMethConfirmed, EngProfConfirmed, RefereesConfirmed, LastToModifyStaffID, ProposedStartDate, ProposalSummary, RequireMoreInfo, DateAdded, DateLastChecked, DateLastModified, awardType, flindersCampus, fullTime, paymentMethod)  **Primary key** ApplicationID  **Foreign key** ApplicantID **references** Application(ApplicationID)  **Foreign key** LastToModifyStaffID **references** University Staff Member(StaffID)  **Foreign key** ManagedByStaffID **references** University Staff Member(StaffID)  **Foreign key** awardType **references** AwardType(Type)  **Foreign key** applicationStatus **references** Application Status(Status)  **Foreign key** paymentMethod **references** Payment Method(Method) |
| **Application\_Research Area** (ApplicationID, FORCode)  **Primary key** ApplicationID, FORCode  **Foreign key** ApplicantID **references** Application(ApplicationID)  **Foreign key** FORCode **references** Research Area(FORCode) |
| **Application Status** (Status, Description)  **Primary key** Status |
| **AwardType** (Type, Description, Method)  **Primary key** Type |
| **Correspondence** (CorrID, Date, Summary, Message, ApplicantID, StaffID, CorrMeth, toStaff  **Primary key** CorrID  **Foreign key** ApplicantID **references** Applicant(ApplicantID)  **Foreign key** StaffID **references** UniversityStaffMember(StaffID)  **Foreign key** CorrMethod **references** Correspondence Method(Method) |
| **Correspondence Method** (Method)  **Primary key** Method |
| **Country** (CountryISOCode, Name)  **Primary key** CountryISOCode |
| **Decision** (DecID, Date, Comment, ApplicationID, StaffID, decType)  **Primary key** DecID  **Foreign key** ApplicationID **references** Application(ApplicationID)  **Foreign key** StaffID **references** UniversityStaffMember(StaffID)  **Foreign key** DecTypeID **references** Decision Type(type) |
| **Decision Type** (type)  **Primary key** type |
| **Degree** (DegID, ApplicantID, Name, Type, YearCompleted, GPA, InstitutionName, InstitutionCountryISOCode)  **Primary key** DegID  **Foreign key** ApplicantID **references** Applicant(ApplicantID)  **Foreign key** InstituitonCountryISOCode **references** Country(CountryISOCode) |
| **Document** (DocID, UploadLink, Title, Description, DocType, DocStat, ApplicantID, ApplicationID)  **Primary key** UploadLink  **Foreign key** DocType **references** DocumentType(Type)  **Foreign key** DocStat **references** DocumentStatus(Status)  **Foreign key** ApplicantID **references** Applicant(ApplicantID)  **Foreign key** ApplicationID **references** Application(ApplicationID) |
| **DocumentStatus** (Status, Description)  **Primary key** Status |
| **DocumentType** (Type, Description)  **Primary key** Type |
| **Payment Method** (Method)  **Primary key** Method |
| **Publication** (PubID, ApplicantID, Title, Publication, IssueNo, IssueDate, OnlineLink, OtherAuthorsNames, Language)  **Primary key** PubID  **Foreign key** ApplicantID **references** Applicant(ApplicantID) |
| **Referee** (RefID, ApplicationID, Name, Relation, Phone, Email, AcademicLink, EnglishSpeaker, EnglishLiterate)  **Primary key** RefID  **Foreign key** ApplicationID **references** Application(ApplicationID) |
| **Research Area** (FORCode, Description, ResearchArea, GeneralArea)  **Primary key** FORCode |
| **Supervise as** (StaffID, ApplicationID, PrimarySupervisor)  **Primary key** StaffID, ApplicationID  **Foreign key** ApplicationID **references** Application(ApplicationID)  **Foreign key** StaffID **references** University Staff Member(StaffID) |
| **University Staff Member** (StaffID, FName, LName, canSupervise)  **Primary key** StaffID |
| **University Staff Member\_Application** (StaffID, ApplicationID)  **Primary key** StaffID, ApplicationID  **Foreign key** StaffID **references** University Staff Member(StaffID)  **Foreign key** ApplicationID **references** Application(ApplicationID) |
| **University Staff Member\_Research Area** (StaffID, FORCode)  **Primary key** StaffID, FORCode  **Foreign key** StaffID **references** University Staff Member (StaffID)  **Foreign key** FORCode **references** Research Area(FORCode) |
| **University Staff Member\_Research Area2** (StaffID, FORCode)  **Primary key** StaffID, FORCode  **Foreign key** StaffID **references** University Staff Member (StaffID)  **Foreign key** FORCode **references** Research Area(FORCode) |
| **Visa** (VisaID, ApplicantID, OriginCountryISOCode, VISAStatus, ValidFrom, ValidTo)  **Primary key** VisaID  **Foreign key** ApplicantID **references** Applicant(ApplicantID)  **Foreign key** OriginCountryISOCode **references** Country(CountryISOCode) |
| **Visa Status** (Status, Description)  **Primary key** Status |

## Logical E-R Diagram



# User transaction validation

University Staff Members

|  |  |
| --- | --- |
| # | pathway |
|  | Look up applicant + publications + degrees + visa Status + Associated documents by applicant name |
|  | Look up applicant’s applications by applicant name |
|  | Look up applicant’s applications by applicant email |
|  | Look up incomplete applications |
|  | Look up all correspondences relevant to an application |
|  | Create new applicant and associated application records |
|  | Look up which staff member updated an Application most recently |
|  | Check for any decision recorded about an application |
|  | Look up an existing application and attach a new standard type document to an application |
|  | Look up an existing application and attached a new exceptional type document to an application |
|  | Look up an existing application and list outstanding information (checklist). |
|  | Update the checklist to confirm that a mandatory information requirement has been met |
|  | Retrieve all on-going applications for which the user has made the most recent correspondence |
|  | Record making a decision about an application |
|  | Update the status of an application |

Academic Staff

|  |  |
| --- | --- |
| # | pathway |
|  | Look up, add to, and delete from own current research areas |
|  | Search for all applications in certain research areas that have been added since a certain time |
|  | Flag interest in an application |

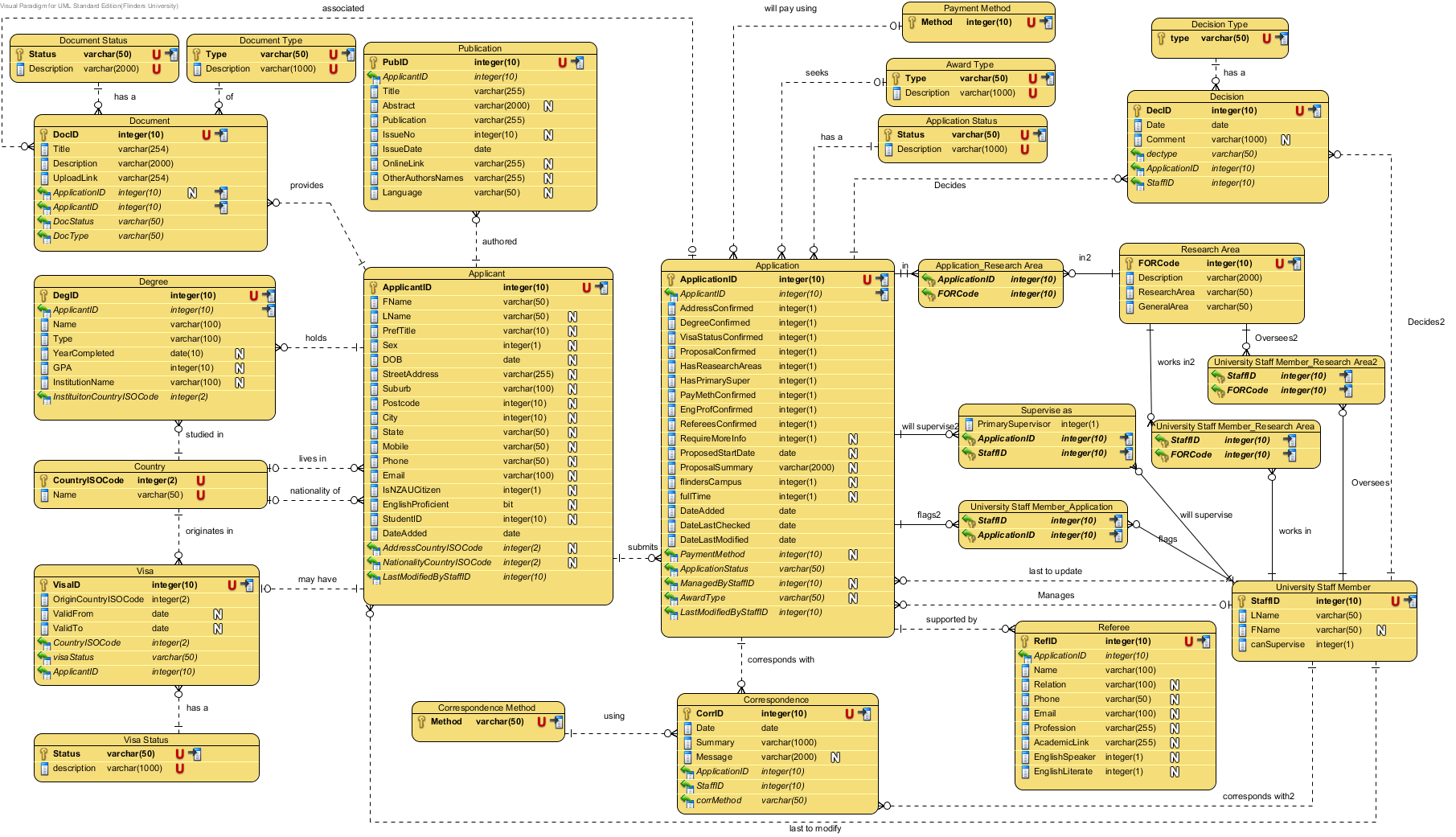
RHD Co-ordination Staff

|  |  |
| --- | --- |
| # | pathway |
|  | Retrieve all staff who have flagged an application, or have edited an application or applicant record most recently |
|  | Retrieve all ongoing applications |

Note that the following modelling changes were triggered when validating user transactions:

* (e) – we had Correspondence in relationship with Applicant, but that could not record which correspondence was in regard to which application. We changed this to have Correspondence in relationship Application.

## Transaction pathways



a1

a2

a3

a4

a5, i, j

b, c,

f, i

e, m

r

s

g, s

r

s

p

p

q

q

h, n

d, t, o

k, l, o

# Check integrity constraints

The following type of integrity constraints have been added to the logical model to protect the database from becoming incomplete, inaccurate, or inconsistent.

## Required data

Required data has been specified through the use of not-null attributes identified in section 3 of the conceptual Documentation. These have been reviewed and included in the logical diagram. Since attributes in VP-UML are null-able by default, the following attributes were changed to not null, that is are required:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Applicant** | | **Degree** | | **ResearchArea** | |
| fName | 1.1 | name | 4.1 | FORCode | 9.1 |
| sex | 1.4 | type | 4.2 | Description | 9.2 |
| email | 1.15 | **Document** | | researchArea | 9.3 |
| dateAdded | 1.20 | Title | 5.1 | generalArea | 9.4 |
| **Application** | | uploadLink | 5.3 | **University Staff Member** | |
| dateAdded | 2.3 | documentType | 5.4 | staffID | 12.1 |
| dateLastChecked | 2.4 | documentStatus | 5.5 | lName | 12.3 |
| dateLastModified | 2.5 | **Publication** | | **Visa** | |
| awardType | 2.7 | title | 7.1 | VisaStatus | 13.3 |
| **Checklist** | | Publication | 7.3 | **Correspondence** | |
| applicationStatus | 3.1 | issueDate | 7.5 | date | 14.1 |
| addressConfirmed | 3.2 | **Referee** | | Summary/message | 14.2 |
| degreeConfirmed | 3.3 | name | 8.1 | corrMeth | 14.4 |
| visaConfirmed | 3.4 | relation | 8.2 | toStaff | 14.5 |
| proposalConfirmed | 3.5 | email | 8.4 | **Decision** | |
| engProfConfirmed | 3.6 | Profession | 8.5 | date | 15.1 |
| hasResearchAreas | 3.7 |  |  | decType | 15.2 |
| hasPrimarySuper | 3.8 |  |  | **SuperviseAs** | |
| payMethConfirmed | 3.9 |  |  | primarySupervisor | 16.1 |
| refereesConfirmed | 3.10 |  |  |  |  |

All other attributes not mentioned here are required to maintain integrity.

Note that some of these attributes are moved to separate entities in the next section, in these cases the foreign key will be not null. Furthermore any attributes added to one of these entities will also be not null and unique.

## Attribute domain constraints

The attribute domains identified in section 4 of the conceptual documentation, have been introduced in the form of lookup relations, as produced in sections 2.2 and 2.3, and have been added to the logical diagram.

## Multiplicity

The multiplicities identified in section 2 of the conceptual documentation and above in sections 1.3-17, have been added to the diagram. The additional multiplicities for the lookup tables identified in sections 2.2 and 2.3 have also been added to the logical diagram.

## Entity integrity

The primary keys identified in section 5 of the conceptual documentation, have been reviewed in section 1 and added to/updated in the diagram. The additional primary keys for the attribute domain entities identified in section 4.2 have also been identified in section 1 and added to the diagram.

## Referential integrity

The primary key of each parent relation has been added to the child relation in the form of a foreign key as specified by the Strong and weak entity table in sections 1.1, 1.2 and 1.7.

### Null attributes (~insert rules)

It should be noted however that the attribute domain relations Identified in section 4.1 that replace null-able attributes will add their primary key as a null-able foreign keys in the child relation. More specifically Award Type, Country, Payment Method and Study Load and Location can all be null in the applicant/applicant entities because they may not be known at the time of the initial input of the entry.

### Updates & Deletes

It is expected that there will be a significant amount of updates as an application (and applicant) progresses through the process. As such all entities have been set to cascade upon any update of the parent table. Even though it is expected that deletions will be rare (if at all) functionality has been included to maintain the integrity of the database. Deletions are performed such that

* All weak entities, that are children of applicant and or application only, CASCADE on delete excluding document which requires applicant to be deleted and
* All lookup tables and University Staff member relationships RESTRICT on all deletes.

This will enable all application/applicant specific material to be removed upon the deletion of an applicant and all application material removed upon a delete. Correspondingly the removal of statuses and other lookup tables (or staff tables) will not leave any of its children such as document without a status. This is particularly relevant for application status as the deletion of a status, will make it lose its point in the process.

RESTRICT has been chosen as it reflects the requirement that nothing should be deleted, since it is possible that mistakes and repeated data can be entered some deletion is enable in this way.

## General constraints.

Currently there are no high level constraints that will affect the validity of the logical model. If any are later realised that will be discovered during Review or the final checks of the physical diagram.

There are however pseudo constraints through the checklist (now merged into the application) whereby an application cannot have certain application statuses unless the corresponding checklist attributes are true.

## Document all integrity constraints

All integrity constraints have been applied to the logical model, which can be used to produce a data dictionary when required.

# Review logical data model

The Logical data model will be reviewed with Paul, the head of the Research Higher Degree Office to ensure that it meets all their requirements and is a true (or as close to possible) representation of the data requirements as specified by the Research Higher Degree Office and the staff members who will use the database.

# Check for future growth

It has been found that there are several possible areas that the logical database model may need to include in future. These include but are not limited to

* Expanding research areas, within the school of CSEM or the wider university,
* The possible inclusion of a housing & transport weak entity/attribute for the application/applicant,
* The inclusion of some form of disability/issues entity and or
* The addition of other statuses, types or other lookup tuples.

It has been found that these areas can be accounted for, in the form of adding new entities or attributes. However some possible future growths areas such as the reuse of referees for anther applications (by the same applicant) will break the model and force a redesign. Even though the redesign will be minor, creating a RefereeApplicaiton entity of referee and application IDs and replacing applicationID with applicantID (if maintaining connection with the applicant), it will mean the database and any programs will have to go offline while the modifications are made.

It has been assumed that such cases occur rarely and as such re-entering details will not be too inefficient. This will ensure that any detail changes that occur between applications are added as different entries.

# Develop Test Plan

We will develop a series of SQL scripts to: create the database tables; populate the table with a baseline set of data; and test scripts to query and update the data and verify the results.

We will use the STK-Unit framework to check test results match the actual results.

Descriptions and pseudo-code implementations of these scripts follow.

## Populate staff members

Populate a set of University Staff Member information plus related Field of Research codes.

--------------------------------------------------------------------------------

-- UNIVERSITY STAFF MEMBERS

--------------------------------------------------------------------------------

-- Denise de Vries

INSERT INTO UniversityStaffMember(StaffID, LName, FName, canSupervise)

VALUES (1000, 'de Vries', 'Denise', 1);

-- FOR 'computer software not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1000, 080399);

-- FOR 'data format not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1000, 080499);

-- FOR 'information systems not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1000, 080699);

-- FOR 'information and computing sciences not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1000, 089999);

--------------------------------------------------------------------------------

-- Paul Calder

INSERT INTO UniversityStaffMember(StaffID, LName, FName, canSupervise)

VALUES (1001, 'Calder', 'Paul', 1);

-- FOR 'artificial intelligence and image processing not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1001, 080199);

-- FOR 'computer software not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1001, 080399);

--------------------------------------------------------------------------------

-- John Roddick

INSERT INTO UniversityStaffMember(StaffID, LName, FName, canSupervise)

VALUES (1002, 'Roddick', 'John', 1);

-- FOR 'artificial intelligence and image processing not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1002, 080199);

-- FOR 'information systems not elsewhere classified'

INSERT INTO UniversityStaffMember\_ResearchArea (StaffID, FORCode)

VALUES (1002, 080699);

--------------------------------------------------------------------------------

-- Jennie Brand

INSERT INTO UniversityStaffMember(StaffID, LName, FName, canSupervise)

VALUES (1003, 'Brand', 'Jennie', 0);

## Populate lookup tables

Populate valid values for lookup tables, so that the related application tables may be easily populated for testing.

--------------------------------------------------------------------------------

-- ApplicationStatus

INSERT INTO ApplicationStatus (Status, Description)

VALUES ('ongoing', 'Application/information gathering is currently ongoing' ) ;

INSERT INTO ApplicationStatus (Status, Description)

VALUES ('complete.accepted', 'Application accepted. Elevated to RHD office.' ) ;

INSERT INTO ApplicationStatus (Status, Description)

VALUES ('complete.declined', 'Application declined. School chooses not to pursue.' ) ;

INSERT INTO ApplicationStatus (Status, Description)

VALUES ('complete.withdrawn', 'Application withdrawn by applicant.' ) ;

INSERT INTO ApplicationStatus (Status, Description)

VALUES ('complete.lapsed', 'No activity for a significant period.' ) ;

--------------------------------------------------------------------------------

-- DocumentType

INSERT INTO DocumentType (Type, Description)

VALUES ('application', 'A completed RHD application form.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('cv', 'CV') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('resume', 'resume') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('faculty\_assessment\_memo', 'Faculty assessment memo') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('certificate', 'A certificate of a previous degree.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('transcript', 'An academic transcript of a previous degree.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('thesis', 'An previous degree major work.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('proposal', 'An application proposal.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('reference', 'A character/academic reference of the applicant.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('publication', 'A scientific publication authored by the applicant.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('financial', 'A document relating to how the RHD place is to be funded.') ;

INSERT INTO DocumentType (Type, Description)

VALUES ('general', 'A type of document other than decribed above.') ;

--------------------------------------------------------------------------------

-- DocumentStatus

INSERT INTO DocumentStatus (Status, Description)

VALUES ('original.english', 'Original document in English');

INSERT INTO DocumentStatus (Status, Description)

VALUES ('original.lote', 'Original document in a language other than English.');

INSERT INTO DocumentStatus (Status, Description)

VALUES ('translation', 'Official translation into English of original document');

INSERT INTO DocumentStatus (Status, Description)

VALUES ('summary', 'Not an original source document.');

--------------------------------------------------------------------------------

-- AwardType

INSERT INTO AwardType (Type, Description)

VALUES ('PhD', 'PhD in any field in the school');

INSERT INTO AwardType (Type, Description)

VALUES ('PhD (Comp Sc)', 'PhD in Computer Science');

INSERT INTO AwardType (Type, Description)

VALUES ('masters', 'Masters by research in any field in the school');

INSERT INTO AwardType (Type, Description)

VALUES ('MIT', 'Masters IT');

--------------------------------------------------------------------------------

-- ResearchArea

INSERT INTO ResearchArea (FORCode, Description)

VALUES (100503, 'computer communication networks') ;

INSERT INTO ResearchArea (FORCode, Description)

VALUES (080199, 'artificial intelligence and image processing not elsewhere classified');

INSERT INTO ResearchArea (FORCode, Description)

VALUES (080399, 'computer software not elsewhere classified');

INSERT INTO ResearchArea (FORCode, Description)

VALUES (080499, 'data format not elsewhere classified');

INSERT INTO ResearchArea (FORCode, Description)

VALUES (080699, 'information systems not elsewhere classified');

INSERT INTO ResearchArea (FORCode, Description)

VALUES (089999, 'information and computing sciences not elsewhere classified');

--------------------------------------------------------------------------------

-- DecisionTypes

INSERT INTO DecisionType (type)

VALUES ('GPA too low');

INSERT INTO DecisionType (type)

VALUES ('TOEFL/IELTS score too low');

INSERT INTO DecisionType (type)

VALUES ('Hand over');

INSERT INTO DecisionType (type)

VALUES ('RFI');

### Populate Country

The Country table is another lookup table.

--------------------------------------------------------------------------------

-- Country

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AD', 'Andorra') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AE', 'United Arab Emirates') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AF', 'Afghanistan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AG', 'Antigua and Barbuda') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AI', 'Anguilla') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AL', 'Albania') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AM', 'Armenia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AO', 'Angola') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AQ', 'Antarctica') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AR', 'Argentina') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AS', 'American Samoa') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AT', 'Austria') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AU', 'Australia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AW', 'Aruba') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AX', 'Åland Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('AZ', 'Azerbaijan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BA', 'Bosnia and Herzegovina') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BB', 'Barbados') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BD', 'Bangladesh') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BE', 'Belgium') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BF', 'Burkina Faso') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BG', 'Bulgaria') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BH', 'Bahrain') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BI', 'Burundi') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BJ', 'Benin') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BL', 'Saint Barthélemy') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BM', 'Bermuda') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BN', 'Brunei Darussalam') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BO', 'Bolivia, Plurinational State of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BQ', 'Bonaire, Sint Eustatius and Saba') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BR', 'Brazil') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BS', 'Bahamas') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BT', 'Bhutan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BV', 'Bouvet Island') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BW', 'Botswana') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BY', 'Belarus') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('BZ', 'Belize') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CA', 'Canada') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CC', 'Cocos (Keeling) Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CD', 'Congo, the Democratic Republic of the') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CF', 'Central African Republic') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CG', 'Congo') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CH', 'Switzerland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CI', 'Côte d''Ivoire') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CK', 'Cook Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CL', 'Chile') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CM', 'Cameroon') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CN', 'China') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CO', 'Colombia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CR', 'Costa Rica') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CU', 'Cuba') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CV', 'Cabo Verde') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CW', 'Curaçao') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CX', 'Christmas Island') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CY', 'Cyprus') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('CZ', 'Czech Republic') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DE', 'Germany') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DJ', 'Djibouti') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DK', 'Denmark') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DM', 'Dominica') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DO', 'Dominican Republic') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('DZ', 'Algeria') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('EC', 'Ecuador') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('EE', 'Estonia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('EG', 'Egypt') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('EH', 'Western Sahara') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ER', 'Eritrea') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ES', 'Spain') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ET', 'Ethiopia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FI', 'Finland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FJ', 'Fiji') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FK', 'Falkland Islands (Malvinas)') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FM', 'Micronesia, Federated States of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FO', 'Faroe Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('FR', 'France') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GA', 'Gabon') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GB', 'United Kingdom') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GD', 'Grenada') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GE', 'Georgia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GF', 'French Guiana') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GG', 'Guernsey') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GH', 'Ghana') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GI', 'Gibraltar') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GL', 'Greenland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GM', 'Gambia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GN', 'Guinea') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GP', 'Guadeloupe') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GQ', 'Equatorial Guinea') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GR', 'Greece') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GS', 'South Georgia and the South Sandwich Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GT', 'Guatemala') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GU', 'Guam') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GW', 'Guinea-Bissau') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('GY', 'Guyana') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HK', 'Hong Kong') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HM', 'Heard Island and McDonald Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HN', 'Honduras') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HR', 'Croatia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HT', 'Haiti') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('HU', 'Hungary') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ID', 'Indonesia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IE', 'Ireland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IL', 'Israel') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IM', 'Isle of Man') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IN', 'India') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IO', 'British Indian Ocean Territory') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IQ', 'Iraq') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IR', 'Iran, Islamic Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IS', 'Iceland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('IT', 'Italy') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('JE', 'Jersey') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('JM', 'Jamaica') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('JO', 'Jordan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('JP', 'Japan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KE', 'Kenya') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KG', 'Kyrgyzstan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KH', 'Cambodia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KI', 'Kiribati') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KM', 'Comoros') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KN', 'Saint Kitts and Nevis') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KP', 'Korea, Democratic People''s Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KR', 'Korea, Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KW', 'Kuwait') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KY', 'Cayman Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('KZ', 'Kazakhstan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LA', 'Lao People''s Democratic Republic') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LB', 'Lebanon') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LC', 'Saint Lucia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LI', 'Liechtenstein') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LK', 'Sri Lanka') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LR', 'Liberia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LS', 'Lesotho') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LT', 'Lithuania') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LU', 'Luxembourg') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LV', 'Latvia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('LY', 'Libya') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MA', 'Morocco') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MC', 'Monaco') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MD', 'Moldova, Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ME', 'Montenegro') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MF', 'Saint Martin (French part)') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MG', 'Madagascar') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MH', 'Marshall Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MK', 'Macedonia, the former Yugoslav Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ML', 'Mali') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MM', 'Myanmar') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MN', 'Mongolia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MO', 'Macao') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MP', 'Northern Mariana Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MQ', 'Martinique') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MR', 'Mauritania') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MS', 'Montserrat') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MT', 'Malta') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MU', 'Mauritius') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MV', 'Maldives') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MW', 'Malawi') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MX', 'Mexico') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MY', 'Malaysia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('MZ', 'Mozambique') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NA', 'Namibia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NC', 'New Caledonia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NE', 'Niger') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NF', 'Norfolk Island') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NG', 'Nigeria') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NI', 'Nicaragua') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NL', 'Netherlands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NO', 'Norway') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NP', 'Nepal') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NR', 'Nauru') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NU', 'Niue') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('NZ', 'New Zealand') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('OM', 'Oman') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PA', 'Panama') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PE', 'Peru') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PF', 'French Polynesia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PG', 'Papua New Guinea') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PH', 'Philippines') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PK', 'Pakistan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PL', 'Poland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PM', 'Saint Pierre and Miquelon') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PN', 'Pitcairn') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PR', 'Puerto Rico') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PS', 'Palestine, State of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PT', 'Portugal') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PW', 'Palau') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('PY', 'Paraguay') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('QA', 'Qatar') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('RE', 'Réunion') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('RO', 'Romania') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('RS', 'Serbia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('RU', 'Russian Federation') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('RW', 'Rwanda') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SA', 'Saudi Arabia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SB', 'Solomon Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SC', 'Seychelles') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SD', 'Sudan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SE', 'Sweden') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SG', 'Singapore') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SH', 'Saint Helena, Ascension and Tristan da Cunha') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SI', 'Slovenia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SJ', 'Svalbard and Jan Mayen') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SK', 'Slovakia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SL', 'Sierra Leone') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SM', 'San Marino') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SN', 'Senegal') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SO', 'Somalia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SR', 'Suriname') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SS', 'South Sudan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ST', 'Sao Tome and Principe') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SV', 'El Salvador') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SX', 'Sint Maarten (Dutch part)') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SY', 'Syrian Arab Republic') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('SZ', 'Swaziland') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TC', 'Turks and Caicos Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TD', 'Chad') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TF', 'French Southern Territories') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TG', 'Togo') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TH', 'Thailand') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TJ', 'Tajikistan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TK', 'Tokelau') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TL', 'Timor-Leste') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TM', 'Turkmenistan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TN', 'Tunisia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TO', 'Tonga') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TR', 'Turkey') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TT', 'Trinidad and Tobago') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TV', 'Tuvalu') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TW', 'Taiwan, Province of China') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('TZ', 'Tanzania, United Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('UA', 'Ukraine') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('UG', 'Uganda') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('UM', 'United States Minor Outlying Islands') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('US', 'United States') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('UY', 'Uruguay') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('UZ', 'Uzbekistan') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VA', 'Holy See (Vatican City State)') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VC', 'Saint Vincent and the Grenadines') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VE', 'Venezuela, Bolivarian Republic of') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VG', 'Virgin Islands, British') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VI', 'Virgin Islands, U.S.') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VN', 'Viet Nam') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('VU', 'Vanuatu') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('WF', 'Wallis and Futuna') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('WS', 'Samoa') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('YE', 'Yemen') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('YT', 'Mayotte') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ZA', 'South Africa') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ZM', 'Zambia') ;

INSERT INTO Country (CountryISOCode, Name)

VALUES ('ZW', 'Zimbabwe') ;

## Populate Applicant and Application

Here we populate a set of fictional applicants, and their applications, degrees, research areas and documents.

This information is based on actual, anonymised information supplied by the RHD office in order to achieve realistic testing.

--------------------------------------------------------------------------------

-- Application for PhD studies.txt

-- 01

INSERT INTO Applicant (ApplicantID, FName, LName, Email, Mobile, StreetAddress,

City, Postcode, AddressCountryISOCode)

VALUES (01, 'Shirin', 'Ebadi', 'shirin.ebadi@keb.com.de',

'+49 (176) 6488 9999', 'Zwillingstr 99, App 099', 'Munich', 80807, 'DE');

INSERT INTO Degree (DegID, ApplicantID, Name, Type, YearCompleted,

InstitutionName, InstitutionCountryCode)

VALUES (121, 01, 'Electrical Engineering with expertise in Control

Theory and Reat-Time Applications', 'bachelor', 2005, 'University of Tabriz',

'IR') ;

INSERT INTO Degree (ApplicantID, Name, Type, YearCompleted, InstitutionName,

InstitutionCountryCode)

VALUES (122, 'Electrical Engineering with expertise in Control

Theory and Reat-Time Applications', 'masters', 2008, 'University of Tabriz',

'IR') ;

INSERT INTO Application (ApplicationID, ApplicantID, awardType)

VALUES (111, 'PhD') ;

INSERT INTO Document (DocID, UploadLink, ApplicantID, DocType)

VALUES (131, '/mnt/data/rhd/01/CV/001.pdf', 01, 'CV' );

--------------------------------------------------------------------------------

-- FW Your kind supervision for my intended PhD.txt

-- 2

-- TODO: 2 emails (also memalki@uqu.edu.sa)

INSERT INTO Applicant(2, FName, LName, Email, Mobile, Phone)

VALUES ('Mohammad', 'Almalki', 'don.memo@hotmail.com', '+966 565907070',

'+966 12 527000000 ext 4951');

INSERT INTO Application (ApplicantID, ApplicationID, awardType)

VALUES (2, 211, 'PhD') ;

INSERT INTO Degree (DegID, ApplicantID, Name, Type, YearCompleted,

InstitutionName, InstitutionCountryCode)

VALUES (221, 2, 'IT', 'master', 2010,

'University of Technology Sydney', 'AU') ;

-- TODO: how to capture 'e-government in higher education organizations' ?

--INSERT INTO Application\_ResearchArea (ApplicationID, FORCode)

--VALUES (ma\_application\_id, );

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + ma\_applicant\_id + '/CV/001.pdf',

ma\_applicant\_id, 'CV' ) ;

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + ma\_applicant\_id + '/transcript/001.pdf',

ma\_applicant\_id, 'transcript', 'transcript of masters course') ;

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + ma\_applicant\_id + '/certificate/001.pdf',

ma\_applicant\_id, 'certificate', 'bachelors certificate') ;

INSERT INTO Document (UploadLink, ApplicantID, ApplicationID, DocType)

VALUES ('/mnt/data/rhd/' + ma\_applicant\_id + '/proposal/001.pdf',

ma\_applicant\_id, ma\_application\_id, 'proposal') ;

--------------------------------------------------------------------------------

-- Fwd Flinders Application - PhD (Comp Sc) - Sem 2 2015.txt

-- 3

INSERT INTO Applicant(ApplicantID, FName, LName, StudentID)

VALUES (03, 'Azzam', 'Alwash', '1234567') ;

INSERT INTO Application(ApplicationID, ApplicantID, awardType, ProposalSummary,

ProposedStartDate, LastModifiedByStaffID)

VALUES (311, 03, 'PhD (Comp Sc)',

'Genetic algorithms for Arabic character recognition', '2015/07', 1001) ;

INSERT INTO Degree (ApplicantID, Name, Type, GPA, InstitutionCountryCode)

VALUES (03, 'IS and CS', 'bachelors', 4.27, 'IQ') ;

INSERT INTO Degree (ApplicantID, Name, Type, GPA, InstitutionCountryCode)

VALUES (03, 'IT', 'masters', 6.79, 'MY') ;

INSERT INTO Document (UploadLink, ApplicantID, ApplicationID, DocType)

VALUES ('/mnt/data/rhd/' + 03 + '/application/0001.pdf',

03, 311, 'application');

--------------------------------------------------------------------------------

-- Fwd Flinders Application - PhD (Computer Science) Sem 2 2014 .txt

-- 4

INSERT INTO Applicant(ApplicantID, FName, LName, StudentID)

VALUES (04, 'Mustafa', 'Al Lami', 2130106);

INSERT INTO Application(ApplicationID, ApplicantID, awardType, ProposalSummary)

VALUES (411, 04, 'PhD',

'Cloud computing for large scale data analysis') ;

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 04 +

'/faculty\_assessment\_memo/0001.pdf',

04, 'faculty\_assessment\_memo');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 04 +

'/general/0001.pdf',

04, 'general');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 04 +

'/general/0002.pdf', 04, 'general');

-- Paul Colder asks for more info about masters

INSERT INTO Decision ( DecID, Date, dectype, Comment, StaffID)

VALUES (1, '2014.05.15', 'RFI',

'Have asked for more info about Masters project', 1001 );

--------------------------------------------------------------------------------

-- Fwd Flinders Application.txt

-- 5

INSERT INTO Applicant(ApplicantID, FName, StudentID)

VALUES (05, 'Ena', '2345678') ;

INSERT INTO Application(ApplicationID, ApplicantID, ProposedStartDate)

VALUES (511, 05, '2014/07') ;

INSERT INTO Document (UploadLink, ApplicantID, ApplicationID, DocType)

VALUES ('/mnt/data/rhd/' + 05 + '/proposal/0001.pdf',

05, 511, 'proposal');

INSERT INTO Document (UploadLink, ApplicantID, ApplicationID, DocType)

VALUES ('/mnt/data/rhd/' + 05 + '/application/0001.pdf',

05, 511, 'application');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 05 + '/transcript/0001.pdf',

05, 'transcript', 'bachelor certificate and transcript');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 05 + '/transcript/0002.pdf',

05, 'transcript', 'master certificate and transcript');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 05 + '/financial/0001.pdf',

05, 'financial');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 05 + '/reference/0001.pdf',

05, 511, 'reference',

'recommendation letters');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 05 + '/general/0001.pdf',

05, 511, 'general', 'training certificate');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 05 +

'/faculty\_assessment\_memo/0001.pdf', 05, 'faculty\_assessment\_memo');

--------------------------------------------------------------------------------

-- Fwd PhD inquiry.txt

-- 6

INSERT INTO Applicant (ApplicantID, FName, LName, Sex)

VALUES (06, 'Fakhri', 'Bazzaz', 'M') ;

INSERT INTO Application (ApplicationID, ApplicantID, awardType)

VALUES (611, 06, 'PhD');

-- 100503 Computer Communications Networks

INSERT INTO Application\_ResearchArea (ApplicationID, FORCode)

VALUES (611, 100503) ;

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 06 + '/general/0001.pdf',

06, 'general', 'Award of Deg');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 06 + '/certificate/0001.pdf',

06, 611, 'certificate');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 06 + '/cv/0001.pdf',

06, 'cv') ;

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 06 + '/certificate/0002.pdf',

06, 'certificate', 'English cert');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 06 + '/publication/0001.pdf',

06, 'publication', 'masters thesis');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 06 + '/transcript/0001.pdf',

06, 'transcript', 'masters transcript');

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 06 + '/reference/0001.pdf',

06, 611, 'reference') ;

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 06 + '/reference/0002.pdf',

06, 611, 'reference') ;

--------------------------------------------------------------------------------

-- Fwd Requesting for PhD supervision.txt

-- 7

INSERT INTO Applicant (ApplicantID, FName)

VALUES (07, 'Nemo') ;

INSERT INTO Degree (DegID, ApplicantID, Name, Type, InstitutionName,

InstitutionCountryCode)

VALUES (721, 07, 'IT', 'bachelors', 'Manipal University', 'IN') ;

INSERT INTO Degree (DegID, ApplicantID, Name, Type, InstitutionName,

InstitutionCountryCode)

VALUES (722, 07, 'Software development and engineering',

'masters', 'West Bengal University', 'IN') ;

INSERT INTO Application (ApplicationID, ApplicantID, Type, Email,

LastToModifyStaffID)

VALUE (711, 07, 'PhD', 'nemo@gmail.com', 1000) ;

INSERT INTO Document (UploadLink, ApplicantID, DocType)

VALUES ('/mnt/data/rhd/' + 07 + '/resume/0001.pdf',

07, 'resume');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 07 + '/publication/0001.pptx',

07, 'publication',

'Six-plus presenting a six-element framework for sentiment analysis.pptx');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 07 + '/publication/0002.docx',

07, 'publication',

'The price prediction framework based upon representation of the opinions');

INSERT INTO Document (UploadLink, ApplicantID, DocType, Description)

VALUES ('/mnt/data/rhd/' + 07 + '/publication/0003.docx',

07, 'publication',

'The classification framework for the representation of opinions.docx');

--------------------------------------------------------------------------------

-- PhD Student.txt

-- 8

INSERT INTO Applicant(ApplicantID, FName, Email)

VALUES (08, 'Sara', 'sara@gmail.com') ;

INSERT INTO Degree (DegID, ApplicantID, Name, Type, YearCompleted, GPA,

InstitutionName, InstitutionCountryCode)

VALUES (821, 08, 'Masters Comp Sc (Information Security)', 'masters', 2013,

6.79);

INSERT INTO Application (ApplicationID, ApplicantID, AwardType)

VALUES (811, 08, 'PhD');

--------------------------------------------------------------------------------

-- PhD Student1.txt

-- 9

INSERT INTO Applicant(ApplicantID, FName, LName, Email)

VALUES (09, 'Abdul-Allah', 'Al-Sadhan', 'abdul-allah.al-sadhan@gmail.com') ;

INSERT INTO Application (ApplicationID, ApplicantID, AwardType)

VALUES (911, 09, 'PhD');

--------------------------------------------------------------------------------

-- PhD Student2.txt

-- 10

INSERT INTO Applicant(ApplicantID, FName, LName, Email)

VALUES (10, 'Fahd', 'Al-Hayyan', 'fahd.al-hayyan@ut.edu.sa') ;

INSERT INTO Application (ApplicationID, ApplicantID, AwardType)

VALUES (1011, 10, 'PhD');

--------------------------------------------------------------------------------

-- Request for PhD Supervision.txt

-- 11

INSERT INTO Applicant(ApplicantID, FName, LName, Email)

VALUES (11, 'Venkatraman', 'Ramakrishnan',

'venkatraman.ramakrishnan@gmail.com');

INSERT INTO Application (ApplicationID, ApplicantID, AwardType)

VALUES (1111, 11, 'PhD');

## Test user transactions

Each user transaction (see section 3) is translated into one or more queries to simulate the application in use.

Expected result counts and values are in comments below.

This will be translated into executable test code suitable for the STK Unit framework in the physical model development stage.

--------------------------------------------------------------------------------

-- USER TRANSACTION TESTING

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

-- a) Look up applicant + publications + degrees + visa Status + Associated

-- documents by applicant name

SELECT \* FROM Applicant

WHERE Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect COUNT = 1

SELECT \* FROM Publication, Applicant

WHERE Publication.ApplicantID = Applicant.ApplicantID

AND Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect COUNT = 1

SELECT \* FROM Degree, Applicant

WHERE Degree.ApplicantID = Applicant.ApplicantID

AND Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect COUNT = 1

SELECT Visa.VisaStatus FROM Visa, Applicant

WHERE Visa.ApplicantID = Applicant.ApplicantID

AND Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect COUNT = 1

SELECT \* FROM Document, Applicant

WHERE Document.ApplicantID = Applicant.ApplicantID

AND Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect COUNT = 1

--------------------------------------------------------------------------------

-- b) Look up applicant’s applications by applicant name

SELECT \* FROM Application, Applicant

WHERE Application.ApplicantID = Applicant.ApplicantID

AND Applicant.FName = 'Shirin'

AND Applicant.LName = 'Ebadi' ;

-- expect ApplicationID = 111, Type = 'PhD'

--------------------------------------------------------------------------------

-- c) Look up applicant’s applications by applicant email

SELECT \* FROM Application, Applicant

WHERE Application.ApplicantID = Applicant.ApplicantID

AND Applicant.Email = 'don.memo@hotmail.com' ;

-- expect ApplicationID = 211, Type = 'PhD'

--------------------------------------------------------------------------------

-- d) Look up incomplete applications

SELECT \* FROM Application

WHERE Application.applicationStatus <> 'complete.accepted'

AND Application.applicationStatus <> 'complete.declined'

AND Application.applicationStatus <> 'complete.lapsed' ;

-- expect all 10 applications

--------------------------------------------------------------------------------

-- e) Look up all correspondences relevant to an application

SELECT \* FROM Correspondence

WHERE Correspondence.ApplicationID = 2 ;

-- expect COUNT = 1

--------------------------------------------------------------------------------

-- f) Create new applicant and associated application records

--

-- tested in populate.sql script

--------------------------------------------------------------------------------

-- g) Look up which staff member updated an Application most recently

SELECT UniversityStaffMember.FName, UniversityStaffMember.LName

FROM UniversityStaffMember, Application

WHERE Application.ApplicationID = 3

AND Application.LastToModifyStaffID = UniversityStaffMember.StaffID ;

-- expect 'Paul' 'Calder'

--------------------------------------------------------------------------------

-- h) Check for any decision recorded about an application

SELECT \* FROM Decision

WHERE Decision.ApplicationID = 4 ;

-- expect COUNT = 1

-- expect StaffID = 1001

-- expect dectype = 'RFI'

--------------------------------------------------------------------------------

-- i) Look up an existing application and attach a new standard type document to

-- an application

-- tested by populate.sql script

--------------------------------------------------------------------------------

-- j) Look up an existing application and attached a new exceptional type

-- document to an application

--

-- tested by populate.sql script

--------------------------------------------------------------------------------

-- k) Look up an existing application and list outstanding information

-- (checklist).

SELECT applicationStatus, AddressConfirmed, DegreeConfirmed,

VisaStatusConfirmed, ProposalConfirmed, HasResearchAreas, HasPrimarySuper,

PayMethConfirmed, EngProfConfirmed

FROM Application

WHERE Application.ApplicationID = 611 ;

-- expect COUNT = 1

--------------------------------------------------------------------------------

-- l) Update the checklist to confirm that a mandatory information requirement

-- has been met

UPDATE Application

SET AddressConfirmed = 1

WHERE Application.ApplicationID = 611 ;

-- expect success

--------------------------------------------------------------------------------

-- m) Retrieve all on-going applications for which the user has made the most

-- recent correspondence

SELECT \*

FROM Application

WHERE Application.applicationStatus = 'ongoing'

AND Application.LastToModifyStaffID = 1 ;

-- expect COUNT = 1

--------------------------------------------------------------------------------

-- n) Record making a decision about an application

-- tested in populate.sql script

--------------------------------------------------------------------------------

-- o) Update the status of an application

UPDATE Application

SET Application.ApplicationStatus = 'complete.declined'

WHERE Application.ApplicationID = 611 ;

-- expect success

--------------------------------------------------------------------------------

-- p) Look up, add to, and delete from own current research areas

SELECT \*

FROM ResearchArea, UniversityStaffMember\_ResearchArea

WHERE ResearchArea.FORCode = UniversityStaffMember\_ResearchArea.FORCode

AND UniversityStaffMember\_ResearchArea.StaffID = 1000 ;

-- expect COUNT = 4

--------------------------------------------------------------------------------

-- q) Search for all applications in certain research areas that have been added

-- since a certain time

-- TODO: dates

SELECT \*

FROM Application, Application\_ResearchArea

WHERE Application.DateAdded >= '2014.05.14'

AND Application.ApplicationID = Application\_ResearchArea.ApplicationID

AND Application\_ResearchArea.FORCode = 100503 ;

-- expect COUNT = 1

--------------------------------------------------------------------------------

-- r) Flag interest in an application

INSERT INTO UniversityStaffMember\_Application (StaffID, ApplicationID)

VALUES (1000, 07);

-- expect success

INSERT INTO UniversityStaffMember\_Application (StaffID, ApplicationID)

VALUES (1002, 07);

-- expect success

--------------------------------------------------------------------------------

-- s) Retrieve all staff who have flagged an application, or have edited an

-- application or applicant record most recently

SELECT UniversityStaffMember.FName, UniversityStaffMember.LName

FROM UniversityStaffMember, UniversityStaffMember\_Application

WHERE UniversityStaffMember\_Application.ApplicationID = 7

AND UniversityStaffMember\_Application.StaffID = UniversityStaffMember.StaffID;

-- expect Denise de Vries and John Roddick

SELECT UniversityStaffMember.FName, UniversityStaffMember.LName

FROM Application, UniversityStaffMember

WHERE Application.ApplicationID = 7

AND Application.LastToModifyStaffID = UniversityStaffMember.StaffID ;

-- expect Denise de Vries

--------------------------------------------------------------------------------

-- t) Retrieve all ongoing applications

SELECT \*

FROM Application

WHERE Application.applicationStatus = 'ongoing' ;

-- expect COUNT = 10

# Data dictionary

1. applied methodology as described in ‘Chapter 18 Methodology – Monitoring and Tuning the Operational System Step 7.2 Duplicating non-key attributes in one-to-many (1:\*) relationships to reduce joins’ [↑](#footnote-ref-1)
2. applied methodology as described in ‘Chapter 18 Methodology – Monitoring and Tuning the Operational System Step 7.2 Duplicating non-key attributes in one-to-many (1:\*) relationships to reduce joins’ [↑](#footnote-ref-2)