

CPSC 304 Project Cover Page

Milestone #: __2__

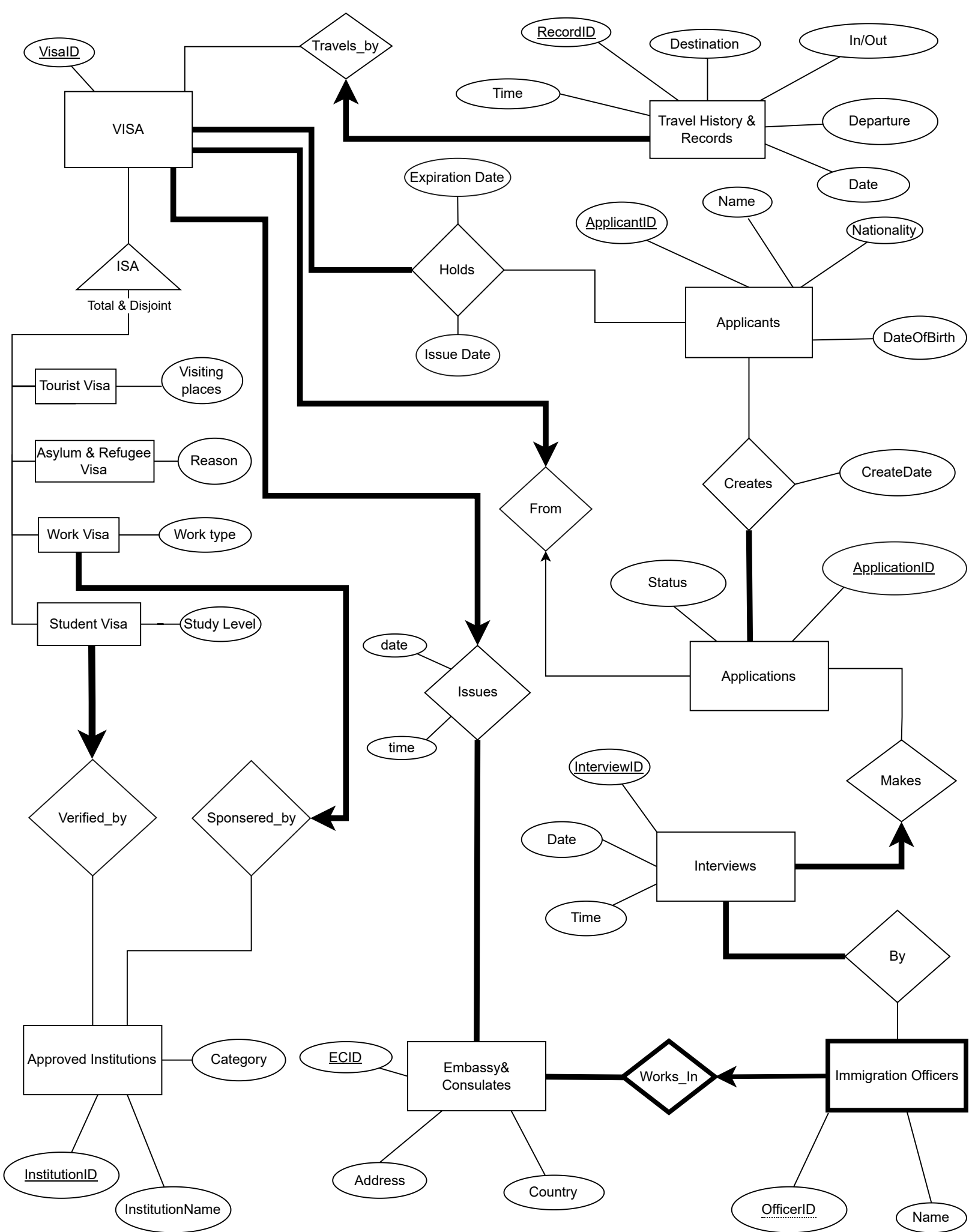
Date: __20-10-2022__

Group Number: __66__

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Shawn Wu	36123040	f3a3g	shawn.xihao.wu@gmail.com
Kunyi Wang	39028535	t5h6p	kunyiwang2652750@gmail.com
Hanze Wang	39408695	o7c8d	hanzewang123@outlook.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia



Changes to ER Diagram

We deleted the thick arrow from Embassy & Consulates to Issues; it is now a thick line.
We deleted the thick line between VISA to Issues; it is now a thick arrow from VISA to Issues.

We made this change because we wanted every visa to correspond to exactly one issuing embassy or consulate.

ER Diagram to Relation Schemas to SQL

Visa_&_From_&_Issue(
 VisaID: String
 VisaType: String,
 ApplicationID: String,
 ECID: String
)

No other candidate keys

Additional Constraints:

The fields of *ApplicationID* and *ECID* cannot be null.

This schema requires a total participation of Embassy_&_Consulates.

Functional Dependencies:

VisaID \rightarrow VisaType, ApplicationID, ECID

Normalization:

It is already in 3NF.

```
CREATE TABLE Visa_&_From_&_Issue (  
    VisaID          VARCHAR(100)          PRIMARY KEY,  
    VisaType        VARCHAR(100),  
    ApplicationID    VARCHAR(100)          NOT NULL,  
    ECID             VARCHAR(100)          NOT NULL,  
    FOREIGN KEY (ApplicationID)  
        REFERENCES Application  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE,  
    FOERIGN KEY (ECID)  
        REFERENCES Embassy_&_Consulates  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Visa_&_From_&_Issue VALUES
('TETY1471', 'TOURIST', '11111111', 'DBOQ4348'),
('TQPX2910', 'TOURIST', '44444444', 'RHPW1067'),
('AEYE2472', 'ASYLUM', '66666666', 'EGHE3624'),
('AXPQ2041', 'ASYLUM', '77777777', 'ECQR2462'),
('WBEY1254', 'WORK', '11111119', 'DBOQ4348'),
('WICN4561', 'WORK', '22222228', 'RHRU2471'),
('SXTR5137', 'STUDENT', '66666664', 'RHRU2471'),
('SRYC2471', 'STUDENT', '77777773', 'ECQR2462');
```

Tourist_Visa(
 VisaID: String,
 VisitingPlaces: String,
)

No other candidate keys

Functional Dependencies:

VisaID \rightarrow VisitingPlaces

Normalization:

It is already in 3NF.

```
CREATE TABLE Tourist_Visa (  
    VisaID          VARCHAR(100)          PRIMARY KEY,  
    VisitingPlaces  VARCHAR(100)  
    FOREIGN KEY (VisaID)  
        REFERENCES Visa_&_From_&_Issue  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Tourist_Visa VALUES  
    ('TETY1471', 'Canada'),  
    ('TQPX2910', 'Singapore');
```

Asylum_Refugee_Visa(
 VisaID: String,
 Reason: String,
)

No other candidate keys

Additional Constraints:

The fields of *Reason* cannot be null.

Functional Dependencies:

VisaID \rightarrow Reason

Normalization:

It is already in 3NF.

```
CREATE TABLE Asylum_Refugee_Visa (  
    VisaID          VARCHAR(100)          PRIMARY KEY,  
    Reason          VARCHAR(100)          NOT NULL,  
    FOREIGN KEY (VisaID)  
        REFERENCES Visa_&_From_&_Issue  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Asylum_Refugee_Visa VALUES  
    ('AEYE2472','Famine'),  
    ('AXPQ2041','Political Prosecution');
```

Work_Visa_&_Sponsered_By(

VisaID: String,

WorkType: String,

InstitutionID: String

)

No other candidate keys

Additional Constraints:

The fields of *WorkType* and *InstitutionID* cannot be null.

Functional Dependencies:

$VisaID \rightarrow WorkType, InstitutionID$

Normalization:

It is already in 3NF.

```
CREATE TABLE Work_Visa_&_Sponsered_By (
  VisaID          VARCHAR(100)      PRIMARY KEY,
  WorkType        VARCHAR(100)      NOT NULL,
  InstitutionID    VARCHAR(100)      NOT NULL,
  FOREIGN KEY (VisaID)
    REFERENCES Visa_&_From_&_Issue
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (InstitutionID)
    REFERENCES Approved_Institutions
    ON DELETE SET DEFAULT
    ON UPDATE CASCADE
);
```

```
INSERT INTO Work_Visa_&_Sponsered_By VALUES
('WBEY1254', 'Programmer', 'AHBG1356'),
('WICN4561', 'Analyst', 'QKCG1920');
```


Student_Visa_&_Verified_By(

VisaID: String,

StudyLevel: String,

InstitutionID: String

)

No other candidate keys

Additional Constraints:

The fields of *StudyLevel*, *InstitutionID* can not be null.

Functional Dependencies:

VisaID → StudyLevel, InstitutionID

Normalization:

It is already in 3NF.

```
CREATE TABLE Student_Visa_&_Verified_By (
  VisaID          VARCHAR(100)      PRIMARY KEY,
  StudyLevel      VARCHAR(100)      NOT NULL,
  InstitutionID   VARCHAR(100)      NOT NULL,
  FOREIGN KEY (VisaID)
    REFERENCES Visa_&_From_&_Issue,
  FOREIGN KEY (InstitutionID)
    REFERENCES Approved_Institutions
    ON DELETE SET DEFAULT
    ON UPDATE CASCADE
);
```

```
INSERT INTO Student_Visa_&_Verified_By VALUES
  ('SXTR5137','UNDERGRADUATE','EHBG1356'),
  ('SRYC2471','UNDERGRADUATE','ENEA2891');
```

Travel_History_&_Records_Travels_By(

RecordID: String,

Time: TIME,

Date: DATE,

Destination: String,

Departure: String,

In/Out: Boolean,

VisaID: String

)

No other candidate keys

Additional Constraints:

None of the fields of attributes can be null.

Functional Dependencies:

RecordID \rightarrow Time, Destination, In/Out, Departure, Date, VisaID

Departure, Destination \rightarrow In/Out

Normalization:

We choose to decompose our schema into 3NF using synthesis.

Step 1: Find the minimal cover of our functional dependencies.

RecordID \rightarrow Time,

RecordID \rightarrow Destination,

RecordID \rightarrow Departure,

RecordID \rightarrow Date,

RecordID \rightarrow VisaID

Destination, Departure \rightarrow In/Out

Step 2: R1(RecordID, Time)

R2(RecordID, Destination)

R3(RecordID, Departure)

R4(RecordID, Date)

R5(RecordID, VisaID)

R6(Destination, Departure, In/Out)

Step 3: No need to add additional relation schema.

Step 4: Optimizing by combining R1, R2, R3, R4, R5.

Final Answer:

R1(RecordID, Time, Date, **Destination**, **Departure**, VisaID)

R6(Destination, Departure, In/Out)

```

CREATE TABLE Travel_History_&_Records_Travels_By (
    RecordID          VARCHAR(100)          PRIMARY KEY,
    Time              TIME                   NOT NULL,
    Date              DATE                   NOT NULL,
    Destination        VARCHAR(100)          NOT NULL,
    Departure          VARCHAR(100)          NOT NULL,
    VisaID            VARCHAR(100)          NOT NULL,
    FOREIGN KEY (Destination, Departure)
        REFERENCES In/Out(Destination, Departure)
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

INSERT INTO Travel_History_&_Records_Travels_By VALUES
    ('WWW111', '14:34:56', '2012-12-06', 'Canada', 'US',
     'TETY1471'),
    ('EEE222', '02:45:23', '2018-09-31', 'China', 'Canada',
     'AEYE2472'),
    ('RRR444', '07:12:06', '2015-03-12', 'US', 'Canada',
     'SXTR5137'),
    ('YYY666', '09:34:50', '2000-12-25', 'Canada', 'Mexico',
     'TQPX2910'),
    ('FFF222', '11:52:09', '2014-04-18', 'Germany', 'Canada',
     'SRYC2471');

```

```

CREATE TABLE In/Out (
    Destination        VARCHAR(100),
    Departure          VARCHAR(100),
    In/Out             BIT              NOT NULL,
    PRIMARY KEY (Destination, Departure)
);

```

```

INSERT INTO In/Out VALUES
    ("Canada", "US", 1),
    ("China", "Canada", 0),
    ("US", "Canada", 0),
    ("Canada", "Mexico", 1),
    ("Germany", "Canada", 0);

```

Applicants(
 ApplicantID: String,
 Name: String,
 Nationality: String,
 DateOfBirth: DATE
 Age: Integer
)

No other candidate keys

Additional Constraints:

None of the fields of attributes can be null.

Functional Dependencies:

ApplicantID \rightarrow Name, Nationality, DateOfBirth, Age

DateOfBirth \rightarrow Age

Normalization:

We choose to decompose our schema into 3NF using method #1.

Step 1: Find the minimal cover.

ApplicantID \rightarrow Name,

ApplicantID \rightarrow Nationality,

ApplicantID \rightarrow DateOfBirth,

DateOfBirth \rightarrow Age

Step 2: Decompose all the way down to BCNF.

Step 2.1: (DateOfBirth \rightarrow Age) violates 3NF.

R1(DateOfBirth, Age)

R2(DateOfBirth, ApplicantID, Name, Nationality)

Step 2.2: No need to decompose further. None of the remaining functional dependencies in R1 and R2 violates BCNF.

Step 3: Identify lost functional dependencies.

ApplicantID \rightarrow Age

Step 4: Construct schemas based on lost functional dependencies and add to decomposition.

R3(ApplicantID, Age)

Final Answer:

R1(DateOfBirth, Age)
R2(ApplicantID, **DateOfBirth**, Name, Nationality)
R3(ApplicantID, Age)

```
CREATE TABLE Applicants (  
    ApplicantID    VARCHAR(100)    PRIMARY KEY,  
    Name           VARCHAR(100)    NOT NULL,  
    Nationality    VARCHAR(100)    NOT NULL,  
    DateOfBirth    DATE            NOT NULL,  
    FOREIGN KEY (DateOfBirth)  
        REFERENCES DateOfBirth_&_Age  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Applicants VALUES  
    ('QOXQMDHO', 'Yitang Zhang', 'China', '1996-08-01'),  
    ('QKXMSNAM', 'Quinn Holmquist', 'Haiti', '1962-08-01'),  
    ('SJZMWISK', 'Kostyantyn Zelensky', 'Ukraine', '1984-12-01'),  
    ('WPSMZKSJ', 'Shinichi Mochizuki', 'Japan', '1935-08-01'),  
    ('QISMZKLP', 'Jingping Xi', 'China', '1953-08-01');
```

```
CREATE TABLE DateOfBirth_&_Age(  
    DateOfBirth    DATE            PRIMARY KEY,  
    Age            INT  
);
```

```
INSERT INTO DateOfBirth_&_Age VALUES  
    ('1996-08-01', 26),  
    ('1962-08-01', 60),  
    ('1984-12-01', 38),  
    ('1935-08-01', 87),  
    ('1953-08-01', 69);
```

```
CREATE TABLE ApplicantID_&_Age (  
    ApplicantID    VARCHAR(100)    PRIMARY KEY,  
    Age            INT,  
    FOREIGN KEY (ApplicantID)  
        REFERENCES Applicants  
        ON DELETE CASCADE
```

```
        ON UPDATE CASCADE  
    );
```

```
INSERT INTO ApplicantID_&_Age VALUES  
    ("QOXQMDHO", 26),  
    ("QKXMSNAM", 60),  
    ("SJZMWISK", 38),  
    ("WPSMZKSJ", 7),  
    ("QISMZKLP", 13);
```

Holds(
 ApplicantID: String,
 VisaID: String,
 IssueDate: DATE,
 ExpirationDate: DATE
 Status: Boolean
)

No other candidate keys

Additional Constraints:

The fields of *IssueDate*, *ExpirationDate*, and *Status* cannot be null.

Each applicant can hold multiple visas.

This schema requires a total participation of Visa_&_From_&_Issue.

Functional Dependencies:

ApplicantID, *VisaID* → *IssueDate*, *ExpirationDate*, *Status*

IssueDate, *ExpirationDate* → *Status*

Normalization:

We choose to decompose our schema into 3NF using method #1.

Step 1: Find the minimal cover.

ApplicantID, *VisaID* → *IssueDate*,

ApplicantID, *VisaID* → *ExpirationDate*,

IssueDate, *ExpirationDate* → *Status*

Step 2: Decompose all the way down to BCNF.

Step 2.1: *IssueDate*, *ExpirationDate* → *Status* violates 3NF.

$R1(\underline{\textit{IssueDate}}, \underline{\textit{ExpirationDate}}, \textit{Status})$

$R2(\underline{\textit{ApplicantID}}, \underline{\textit{VisaID}}, \textit{IssueDate}, \textit{ExpirationDate})$

Step 2.2: No need to decompose further. None of the remaining functional dependencies in R1 and R2 violates BCNF.

Step 3: Identify lost functional dependencies.

ApplicantID, *VisaID* → *Status*

Step 4: Construct schemas based on lost functional dependencies and add to decomposition.

$R3(\underline{\textit{ApplicantID}}, \underline{\textit{VisaID}}, \textit{Status})$

Final Answer:

R1(IssueDate, ExpirationDate, Status)

R2(ApplicantID, VisaID, IssueDate, ExpirationDate)

R3(ApplicantID, VisaID, Status)

```
CREATE TABLE Holds (  
    ApplicantID      VARCHAR(100),  
    VisaID           VARCHAR(100),  
    IssueDate        DATE          NOT NULL,  
    ExpirationDate   DATE          NOT NULL,  
    PRIMARY KEY (ApplicantID, VisaID),  
    FOREIGN KEY (IssueDate, ExpirationDate)  
        REFERENCE IssueDate_&_ExpirationDate_&_Status(IssueDate,  
ExpirationDate)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Holds VALUES  
( 'QOXQMDHO', 'TETY1471', '2012-10-12', '2022-10-12' ),  
( 'QKXMSNAM', 'TQPX2910', '2011-10-12', '2021-10-12' ),  
( 'WPSMZKSJ', 'AEYE2472', '2013-10-12', '2023-10-12' ),  
( 'SJZMWSK', 'AXPQ2041', '2014-10-12', '2024-10-12' ),  
( 'QISMZKLP', 'WBEY1254', '2010-10-12', '2020-10-12' ),  
( 'QOXQMDHO', 'WICN4561', '2012-10-12', '2022-10-12' ),  
( 'QKXMSNAM', 'SXTR5137', '2011-10-12', '2021-10-12' ),  
( 'WPSMZKSJ', 'SRYC2471', '2013-10-12', '2023-10-12' );
```

```
CREATE TABLE IssueDate_&_ExpirationDate_&_Status (  
    IssueDate        DATE,  
    ExpirationDate   DATE,  
    Status           BIT          NOT NULL,  
    PRIMARY KEY (IssueDate, ExpirationDate)  
);
```

```
INSERT INTO IssueDate_&_ExpirationDate_&_Status VALUES  
( '2012-10-12', '2022-10-12', 1 ),  
( '2011-10-12', '2021-10-12', 0 ),  
( '2013-10-12', '2023-10-12', 1 ),  
( '2014-10-12', '2024-10-12', 1 ),
```



```
('2010-10-12','2020-10-12',0);
```

```
CREATE TABLE ApplicantID_&VisaID_&Status (  
    ApplicantID    VARCHAR(100),  
    VisaID         VARCHAR(100),  
    Status         BIT NOT NULL,  
    PRIMARY KEY (ApplicantID, VisaID),  
    FOREIGN KEY (ApplicantID, VisaID)  
        REFERENCES Holds(ApplicantID, VisaID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO ApplicantID_&VisaID_&Status VALUES  
    ('QOXQMDHO', 'TETY1471', 1),  
    ('QKXMSNAM', 'TQPX2910', 0),  
    ('WPSMZKSJ', 'AEYE2472', 1),  
    ('SJZMWISK', 'AXPQ2041', 1),  
    ('QISMZKLP', 'WBEY1254', 0),  
    ('QOXQMDHO', 'WICN4561', 1),  
    ('QKXMSNAM', 'SXTR5137', 0),  
    ('WPSMZKSJ', 'SRYC2471', 1);
```

Creates(

ApplicantID: String,
ApplicationID: String,
CreateDate: DATE

)

No other candidate keys

Additional Constraints:

The fields of *CreateDate* cannot be null.

This schema requires a total participation of Applications.

Functional Dependencies:

ApplicantID, *ApplicationID* → *CreateDate*

Normalization:

It is already in 3NF.

```
CREATE TABLE Creates (
  ApplicantID      VARCHAR(100),
  ApplicationID     VARCHAR(100),
  CreateDate       DATE          NOT NULL,
  PRIMARY KEY (ApplicantID, ApplicationID),
  FOREIGN KEY (ApplicantID)
    REFERENCES Applicants(ApplicantID),
  FOREIGN KEY (ApplicationID)
    REFERENCES Applications(ApplicationID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);x
```

```
INSERT INTO Creates VALUES
('QOXQMDHO', '11111111', '2000-01-01'),
('QOXQMDHO', '44444444', '2000-01-02'),
('QKXMSNAM', '66666666', '2000-01-03'),
('QKXMSNAM', '77777777', '2000-01-04'),
('SJZMWISK', '66666664', '2000-01-05'),
('WPSMZKSJ', '11111119', '2000-01-06'),
('WPSMZKSJ', '22222228', '2000-01-07'),
('QISMZKLP', '77777773', '2000-01-08');
```

Applications(
 ApplicationID: String,
 Status: String,
)

No other candidate keys

Additional Constraints:

The fields of *Status* cannot be null.

Functional Dependencies:

ApplicationID → Status

Normalization:

It is already in 3NF.

```
CREATE TABLE Applications (  
    ApplicationID    VARCHAR(100)        PRIMARY KEY,  
    Status          BIT                 NOT NULL  
);
```

```
INSERT INTO Applications VALUES  
    ('11111111', 1),  
    ('44444444', 1),  
    ('66666666', 1),  
    ('77777777', 1),  
    ('22222228', 1),  
    ('11111119', 1),  
    ('66666664', 1),  
    ('77777773', 1),  
    ('35627464', 0),  
    ('24624724', 0),  
    ('12467146', 0);
```

Interviews_&_Makes(
 InterviewID: String,
 Date: DATE,
 Time: TIME,
 ApplicationID: String
)

No other candidate keys

Additional Constraints:

 The fields of *ApplicationID* cannot be null.

Functional Dependencies:

 InterviewID → Date, Time, ApplicationID

Normalization:

 It is already in 3NF.

```
CREATE TABLE Interviews_&_Makes (  
    InterviewID    VARCHAR(100),  
    Date           DATE,  
    Time           TIME,  
    ApplicationID  VARCHAR(100)    NOT NULL,  
    PRIMARY KEY (InterviewID),  
    FOREIGN KEY (ApplicationID)  
        REFERENCES Applications(ApplicationID)  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Interviews_&_Makes VALUES  
    ('ABCD1111', '2000-02-25', '11:11:11', '11111111'),  
    ('ABCD4444', '2000-03-20', '11:12:12', '44444444'),  
    ('ABCD6666', '2000-03-05', '11:13:13', '66666666'),  
    ('ABCD7777', '2000-07-15', '11:14:14', '77777777'),  
    ('ABCD8888', '2001-12-19', '11:15:15', '77777773');
```

By(

InterviewID: String,

ECID: String,

OfficerID: String

)

No other candidate keys

Additional Constraints:

This schema requires a total participation of Interviews_&_Makes.

Functional Dependencies:

InterviewID → ECID, OfficerID

```
CREATE TABLE By (
  InterviewID      VARCHAR(100),
  ECID             VARCHAR(100),
  OfficerID        VARCHAR(100),
  PRIMARY KEY (InterviewID, ECID, OfficerID),
  FOREIGN KEY (InterviewID)
    REFERENCES Interviews_&_Makes,
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (ECID, OfficerID)
    REFERENCES Immigration_Officers_&_Works_In(ECID, OfficerID)
    ON DELETE SET DEFAULT
    ON UPDATE CASCADE
);
```

```
INSERT INTO By VALUES
  ('ABCD1111', 'RHRU2471', 'AAAA1111'),
  ('ABCD4444', 'RHPW1067', 'BBBB1111'),
  ('ABCD6666', 'DBOQ4348', 'CCCC1111'),
  ('ABCD7777', 'EGHE3624', 'DDDD1111'),
  ('ABCD8888', 'ECQR2462', 'EEEE1111');
```

Immigration_Officers_&_Works_In(

ECID: String,

OfficerID: String,

Name: String

)

No other candidate keys

Additional Constraints:

The fields of *Name* cannot be null.

This schema requires a total participation of Embassy_&_Consulates.

Functional Dependencies:

ECID, OfficerID → Name

Normalization:

It is already in 3NF.

```
CREATE TABLE Immigration_Officers_&_Works_In (  
    ECID          VARCHAR(100),  
    OfficerID     VARCHAR(100),  
    Name          VARCHAR(100)          NOT NULL,  
    PRIMARY KEY (ECID, OfficerID),  
    FOREIGN KEY (ECID)  
        REFERENCES Embassy_&_Consulates(ECID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Immigration_Officers_&_Works_In VALUES  
    ('RHRU2471', 'AAAA1111', 'Jianping Xi'),  
    ('RHPW1067', 'BBBB1111', 'Lelouch vi Britannia'),  
    ('DBOQ4348', 'CCCC1111', 'Shane Dawson'),  
    ('EGHE3624', 'DDDD1111', 'Finn Hudson'),  
    ('ECQR2462', 'EEEE1111', 'Georg Hainz');
```

Embassy_&_Consulates(
 ECID: String,
 Address: String,
 Country: String
)

Additional Candidate Key:

Address

Additional Constraints:

The fields of *Address* and *Country* cannot be null.

Functional Dependencies:

ECID → Address, Country

Address → Country

Normalization:

We choose to decompose our schema into 3NF using synthesis.

Step 1: Find the minimal cover of our functional dependencies.

ECID → Address

Address → Country

Step 2: R1(ECID, Address)

R2(Address, Country)

Step 3: Add in a relation that contains all the attributes of a key, if it doesn't exist.

No need to add additional relation schemas.

Final answer:

R1(ECID, Address)

R2(Address, Country)

```
CREATE TABLE Embassy_&_Consulates (  
    ECID          VARCHAR(100)    PRIMARY KEY,  
    Address       VARCHAR(100)    UNIQUE,  
    FOREIGN KEY (Address)  
        REFERENCES Address_&_Country  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

```
INSERT INTO Embassy_&_Consulates VALUES
('RHRU2471', '1BLVD1Street1House'),
('RHPW1067', '2BLVD2Street2House'),
('DBOQ4348', '3BLVD3Street3House'),
('EGHE3624', '4BLVD4Street4House'),
('ECQR2462', '5BLVD5Street5House');
```

```
CREATE TABLE Address_&_Country (
    Address          VARCHAR(100)    PRIMARY KEY,
    Country          VARCHAR(100)    NOT NULL,
);
```

```
INSERT INTO Address_&_Country VALUES
('1BLVD1Street1House', 'Canada'),
('2BLVD2Street2House', 'Canada'),
('3BLVD3Street3House', 'Canada'),
('4BLVD4Street4House', 'US'),
('5BLVD5Street5House', 'Canada');
```


Approved_Institutions(
 InstitutionID: String,
 InstitutionName: String,
 Category: String
)

No other candidate keys

Functional Dependencies:

InstitutionID → InstitutionName, Category

Normalization:

It is already in 3NF.

```
CREATE TABLE Approved_Institutions (  
    InstitutionID          VARCHAR(100),  
    InstitutionName        VARCHAR(100),  
    Category               VARCHAR(100),  
    PRIMARY KEY (InstitutionID)  
);
```

```
INSERT INTO Approved_Institutions VALUES  
    ("EHBG1356", "University of British Columbia", "university"),  
    ("ENEA2891", "University of Toronto", "university"),  
    ("AHBG1356", "Amazon", "company"),  
    ("QKCG1920", "CIBC", "company"),  
    ("QPXM2917", "Microsoft", "company");
```

SQL Assertions to Enforce Total Participation Constraints

```
CREATE ASSERTION assert_All_Visa_In_Hold
----To enforce the total participation of VISA in Hold
```

```
CREATE ASSERTION assert_All_Applications_In_Creates
---To enforce the total participation of Applications in Creates
```

```
CREATE ASSERTION assert_All_Interviews_In_By
---To enforce the total participation of Interviews in By
```

```
CREATE ASSERTION assert_All_Embassy_In_Works_In
---To enforce the total participation of Embassy_&_Consulates in
Works_In
```

```
CREATE ASSERTION assert_All_Embassy_In_Issues
---To enforce the total participation of Embassy_&_Consulates in
Issues
```