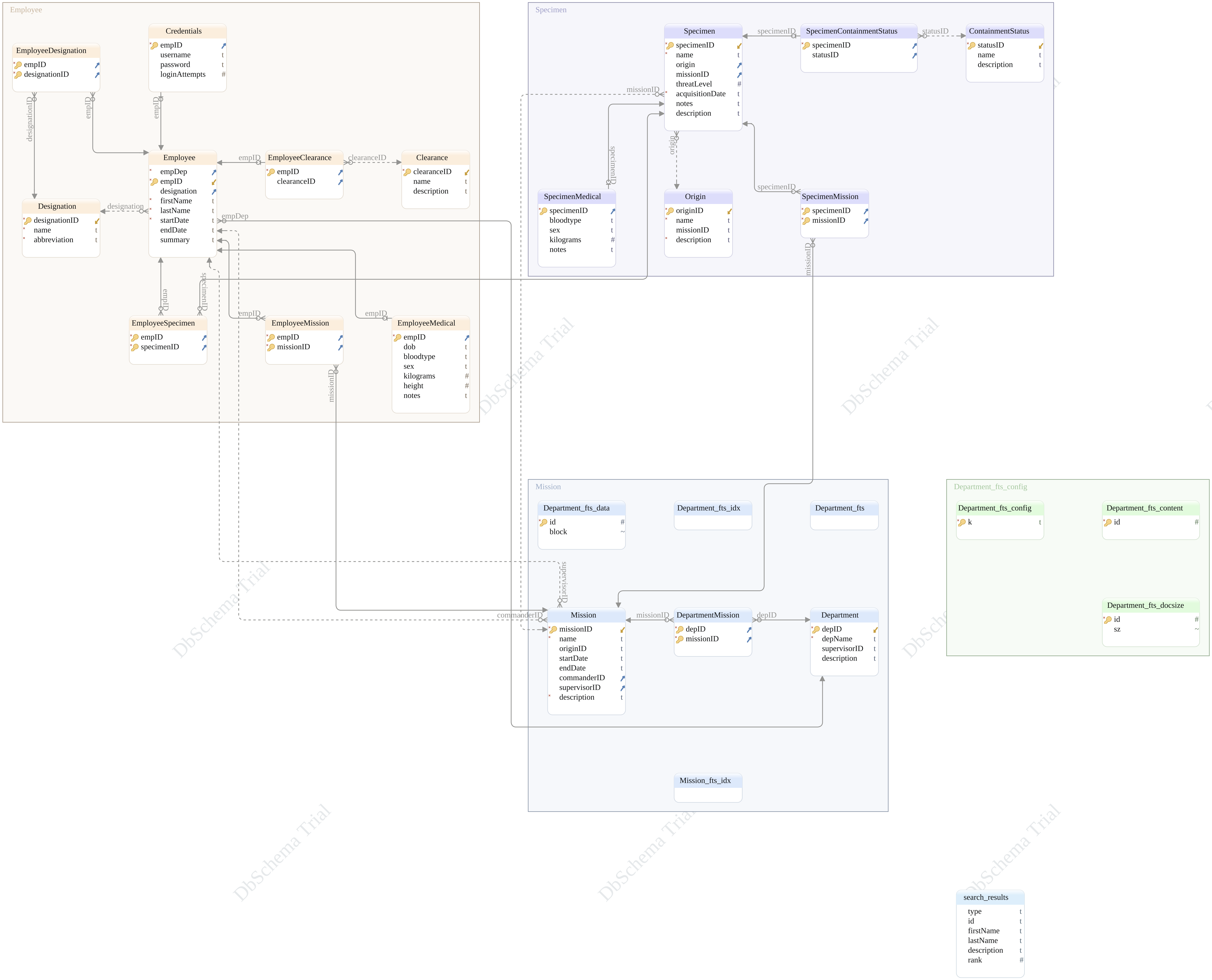
Employee
Credentials
empID
username
password
loginAttempts
EmployeeDesignation
empID
designationID
designationID
empID
empID
Employee
empDep
empID
designation
firstName
lastName
startDate
endDate
summary
empID
EmployeeClearance
empID
clearanceID
clearanceID
Clearance
clearanceID
name
description
Designation
designationID
name
abbreviation
designation
empDep
empID
specimenID
empID
empID
EmployeeSpecimen
empID
specimenID
EmployeeMission
empID
missionID
EmployeeMedical
empID
dob
bloodtype
sex
kilograms
height
notes
missionID
Specimen
Specimen
specimenID
name
origin
missionID
threatLevel
acquisitionDate
notes
description
specimenID
SpecimenContainmentStatus
specimenID
statusID
statusID
ContainmentStatus
statusID
name
description
missionID
specimenID
origin
specimenID
SpecimenMedical
specimenID
bloodtype
sex
kilograms
notes
Origin
originID
name
missionID
description
SpecimenMission
specimenID
missionID
missionID
Mission
Department_fts_data
id
block
Department_fts_idx
Department_fts
supervisorID
commanderID
Mission
missionID
name
originID
startDate
endDate
commanderID
supervisorID
description
missionID
DepartmentMission
depID
missionID
depID
Department
depID
depName
supervisorID
description
Mission_fts_idx
Department_fts_config
Department_fts_config
k
Department_fts_content
id
Department_fts_docsize
id
sz
search_results
type
id
firstName
lastName
description
rank

Main Layout

Table Clearance

* Pk	clearanceID	INTEGER
	name	TEXT
	description	TEXT
Indexes		
Unq	unq_Clearance_clearanceID	clearanceID
Pk	pk_Clearance	clearanceID

Table ContainmentStatus

* Pk	statusID	INTEGER
	name	TEXT
	description	TEXT
Indexes		
Unq	unq_ContainmentStatus_statusID	statusID
Pk	pk_ContainmentStatus	statusID

Table Credentials

* Pk	empID	TEXT
	username	TEXT
	password	TEXT
	loginAttempts	INTEGER
Indexes		
Unq	unq_Credentials_empID	empID
Pk	pk_Credentials	empID
Foreign Keys		
	employeeID (empID) ref Employee (empID)	

Table Department

* Pk	depID	INTEGER
*	depName	TEXT
	supervisorID	TEXT
	description	TEXT
Indexes		
Unq	unq_Department_depID	depID
Pk	pk_Department	depID
Triggers		

Table Department

department_fts_delete

```
CREATE TRIGGER department_fts_delete AFTER DELETE ON Department
          BEGIN
            DELETE FROM Department_fts
            WHERE depID = old.depID;
          END
```

department_fts_insert

```
CREATE TRIGGER department_fts_insert AFTER INSERT ON Department
          BEGIN
            INSERT INTO Department_fts (depID, depName, supervisorID, description)
            VALUES (new.depID, new.depName, new.supervisorID, new.description);
          END
```

department_fts_update

```
CREATE TRIGGER department_fts_update AFTER UPDATE ON Department
          BEGIN
            UPDATE Department_fts
            SET depID = new.depID,
              depName = new.depName,
              supervisorID = new.supervisorID,
              description = new.description
            WHERE depID = old.depID;
          END
```

Table DepartmentMission

* Pk	depID	TEXT
* Pk	missionID	TEXT
Indexes		
Pk	pk_DepartmentMission	depID, missionID
Foreign Keys		
	depID (depID) ref Department (depID)	
	missionID (missionID) ref Mission (missionID)	

Table Department_fts_config

* Pk	k	VARCHAR
Indexes		
Pk	pk_Department_fts_config	k

Table Department_fts_content

* Pk	id	INTEGER
Indexes		
Pk	pk_Department_fts_content	id

Table Department_fts_data

* Pk	id	INTEGER
	block	BLOB
Indexes		
Pk	pk_Department_fts_data	id

Table Department_fts_docsize

* Pk	id	INTEGER
	sz	BLOB
Indexes		
Pk	pk_Department_fts_docsize	id

Table Designation

* Pk	designationID	INTEGER
*	name	TEXT
*	abbreviation	TEXT
Indexes		
Unq	unq_Designation_designationID	designationID
Pk	pk_Designation	designationID

Table Employee

*	empDep	INTEGER
* Pk	empID	TEXT
	designation	INTEGER
*	firstName	TEXT
*	lastName	TEXT
*	startDate	TEXT
	endDate	TEXT
	summary	TEXT
Indexes		
Unq	unq_Employee_empID	empID
Pk	pk_Employee	empID
Foreign Keys		
	FK_Employee Department (empDep) ref Department (depID)	
	FK_Employee Designation (designation) ref Designation (designationID)	
Constraints		
	Cns_Employee_firstName	LENGTH(firstName) <= 50

Table Employee

	Cns_Employee_lastName	LENGTH(lastName) <= 50

Triggers

emp_deletes

```
CREATE TRIGGER emp_deletes AFTER DELETE ON Employee
          BEGIN
            DELETE FROM Employee_fts
            WHERE empID = old.empID;
          END
```

emp_fts_update

```
CREATE TRIGGER emp_fts_update AFTER UPDATE ON Employee
          BEGIN
            UPDATE Employee_fts
            SET empID = new.empID,
              empDep = new.empDep,
              designation = new.designation,
              firstName = new.firstName,
              lastName = new.lastName,
              summary = new.summary
            WHERE empID = old.empID;
          END
```

emp_inserts

```
CREATE TRIGGER emp_inserts AFTER INSERT ON Employee
          BEGIN
            INSERT INTO Employee_fts (empID, empDep, designation, firstName, lastName, summary)
            VALUES (new.empID, new.empDep, new.designation, new.firstName, new.lastName, new.summary);
            INSERT INTO EmployeeMedical(empID)
            VALUES (new.empID);
            INSERT INTO Credentials (empID)
            VALUES (new.empID);
            INSERT INTO EmployeeClearance (empID)
            VALUES (new.empID);
          END
```

Table EmployeeClearance

* Pk	empID	TEXT
	clearanceID	INTEGER

Indexes

Unq	unq_EmployeeClearance_empID	empID
Pk	pk_EmployeeClearance	empID

Foreign Keys

	employeeID (empID) ref Employee (empID)
	clearanceID (clearanceID) ref Clearance (clearanceID)

Table EmployeeDesignation

* Pk	empID	TEXT
* Pk	designationID	INTEGER

Table EmployeeDesignation

Indexes		
Pk	pk_EmployeeDesignation	empID, designationID
Foreign Keys		
	FK_EmployeeDesignation Employee (empID) ref Employee (empID)	
	FK_EmployeeDesignation Designation (designationID) ref Designation (designationID)	

Table EmployeeMedical

* Pk	empID	TEXT
	dob	TEXT
	bloodtype	TEXT
	sex	TEXT
	kilograms	REAL
	height	REAL
	notes	TEXT
Indexes		
Unq	unq_EmployeeMedical_empID	empID
Pk	pk_EmployeeMedical	empID
Foreign Keys		
	empID (empID) ref Employee (empID)	
Constraints		
	Cns_EmployeeMedical_bloodtype	bloodtype IN ('A+', 'O+', 'B+', 'AB+', 'A-', 'O-', 'B-', 'AB-', 'V-', 'V+', 'BF', 'undefined')
	Cns_EmployeeMedical_sex	sex IN ('m', 'f', 'inter', 'unknown', 'undefined')

Table EmployeeMission

* Pk	empID	TEXT
* Pk	missionID	TEXT
Indexes		
Pk	pk_EmployeeMission	empID, missionID
Foreign Keys		
	empID (empID) ref Employee (empID)	
	missionID (missionID) ref Mission (missionID)	

Table EmployeeSpecimen

* Pk	empID	TEXT
* Pk	specimenID	TEXT
Indexes		

Table EmployeeSpecimen

Pk	pk_EmployeeSpecimen	empID, specimenID

Foreign Keys

	employeeID (empID) ref Employee (empID)
	specimenID (specimenID) ref Specimen (specimenID)

Table Mission

* Pk	missionID	TEXT
*	name	TEXT
	originID	TEXT
	startDate	TEXT
	endDate	TEXT
	commanderID	TEXT
	supervisorID	TEXT
*	description	TEXT

Indexes

Unq	unq_Mission_missionID	missionID
Pk	pk_Mission	missionID

Foreign Keys

	FK_Mission Employee (commanderID) ref Employee (empID)
	FK_Mission Employee_001 (supervisorID) ref Employee (empID)

Triggers

mission_fts_delete

```
CREATE TRIGGER mission_fts_delete AFTER DELETE ON Mission
          BEGIN
            DELETE FROM Mission_fts
            WHERE missionID = old.missionID;
          END
```

mission_fts_insert

```
CREATE TRIGGER mission_fts_insert AFTER INSERT ON Mission
          BEGIN
            INSERT INTO Mission_fts (missionID, name, originID, startDate, endDate, commanderID, supervisorID, description)
            VALUES (new.missionID, new.name, new.originID, new.startDate, new.endDate, new.commanderID, new.supervisorID, new.description);
          END
```

mission_fts_update

Table Mission

```
CREATE TRIGGER mission_fts_update AFTER UPDATE ON Mission
          BEGIN
            UPDATE Mission_fts
            SET missionID = new.missionID,
              name = new.name,
              originID = new.originID,
              startDate = new.startDate,
              endDate = new.endDate,
              commanderID = new.commanderID,
              supervisorID = new.supervisorID,
              description = new.description
            WHERE missionID = old.missionID;
          END
```

Table Origin

* Pk	originID	TEXT
*	name	TEXT
	missionID	TEXT
*	description	TEXT

Indexes

Unq	unq_Origin_originID	originID
Pk	pk_Origin	originID

Triggers

origin_fts_delete

```
CREATE TRIGGER origin_fts_delete AFTER DELETE ON Origin
          BEGIN
            DELETE FROM Origin_fts
            WHERE originID = old.originID;
          END
```

origin_fts_insert

```
CREATE TRIGGER origin_fts_insert AFTER INSERT ON Origin
          BEGIN
            INSERT INTO Origin_fts (originID, name, missionID, description)
            VALUES (new.originID, new.name, new.missionID, new.description);
          END
```

origin_fts_update

```
CREATE TRIGGER origin_fts_update AFTER UPDATE ON Origin
          BEGIN
            UPDATE Origin_fts
            SET originID = new.originID,
              name = new.name,
              missionID = new.missionID,
              description = new.description
            WHERE originID = old.originID;
          END
```

Table Specimen

* Pk	specimenID	TEXT

Table Specimen

*	name	TEXT
	origin	TEXT
	missionID	TEXT
	threatLevel	REAL
*	acquisitionDate	TEXT
	notes	TEXT
	description	TEXT

Indexes

Unq	unq_Specimen_specimenID	specimenID
Pk	pk_Specimen	specimenID

Foreign Keys

originID (origin) ref Origin (originID)

missionID (missionID) ref Mission (missionID)

Triggers

specimen_fts_delete

```
CREATE TRIGGER specimen_fts_delete AFTER DELETE ON Specimen
        BEGIN
           DELETE FROM Specimen_fts
           WHERE specimenID = old.specimenID;
        END
```

specimen_fts_update

```
CREATE TRIGGER specimen_fts_update AFTER UPDATE ON Specimen
        BEGIN
           UPDATE Specimen_fts
           SET specimenID = new.specimenID,
              name = new.name,
              origin = new.origin,
              missionID = new.missionID,
              threatLevel = new.threatLevel,
              acquisitionDate = new.acquisitionDate,
              notes = new.notes,
              description = new.description
           WHERE specimenID = old.specimenID;
           UPDATE SpecimenMedical
           SET specimenID = new.specimenID
           WHERE specimenID = old.specimenID;
        END
```

specimen_inserts

```
CREATE TRIGGER specimen_inserts AFTER INSERT ON Specimen
        BEGIN
           INSERT INTO Specimen_fts (specimenID, name, origin, missionID, threatLevel, acquisitionDate, notes, description)
           VALUES (new.specimenID, new.name, new.origin, new.missionID, new.threatLevel, new.acquisitionDate, new.notes, new.description);
           INSERT INTO SpecimenMedical (specimenID)
           VALUES (new.specimenID);
           INSERT INTO SpecimenContainmentStatus (specimenID)
           VALUES (new.specimenID);
        END
```

Table SpecimenContainmentStatus

* Pk	specimenID	TEXT
	statusID	INTEGER
Indexes		
Unq	unq_SpecimenContainmentStatus_specimenID	specimenID
Pk	pk_SpecimenContainmentStatus	specimenID
Foreign Keys		
	SpecimenID (specimenID) ref Specimen (specimenID)	
	statusID (statusID) ref ContainmentStatus (statusID)	

Table SpecimenMedical

* Pk	specimenID	TEXT
	bloodtype	TEXT
	sex	TEXT
	kilograms	REAL
	notes	TEXT
Indexes		
Pk	pk_SpecimenMedical	specimenID
Foreign Keys		
	specimenID (specimenID) ref Specimen (specimenID)	
Constraints		
	Cns_SpecimenMedical_bloodtype	bloodtype IN ('A+', 'O+', 'B+', 'AB+', 'A-', 'O-', 'B-', 'AB-', 'V-', 'V+', 'BF', 'undefined')
	Cns_SpecimenMedical_sex	sex IN ('m', 'f', 'inter', 'unknown', 'undefined')

Table SpecimenMission

* Pk	specimenID	TEXT
* Pk	missionID	TEXT
Indexes		
Pk	pk_SpecimenMission	specimenID, missionID
Foreign Keys		
	specimenID (specimenID) ref Specimen (specimenID)	
	missionID (missionID) ref Mission (missionID)	

Table search_results

	type	TEXT
	id	TEXT
	firstName	TEXT

Table search_results

lastName	TEXT
description	TEXT
rank	INTEGER