Asset Integrity Data Control Proposal

Project description

The asset integrity data control project aims to replace existing manual reporting and excel generations of asset integrity mapping of infrastructure. To do this, a hosted SQL database and web application will be the prototype for this work.

This project will reside within the context of other enterprise software. As such, the application is expected to show a prototype of the data management for infrastructure asset management projects, or also called asset integrity projects.

Background

Clients approach our company to undertake integrity assessment of their infrastructure. When engaged, a series of inspections and data collection is carried out to locate damages on their buildings and understand the state of their current defect tracking.

Figure 1 Example of Inspection

Depending on the client's appetite for mapping defects, clients may opt for our company to review their drawings, current photos, records and develop drawings to recommend remediations that can be undertaken to a structure.

Domain

For this project, the application will be focusing on the data collection, and mapping of the asset integrity initial project phase data. (Later phases go into construction works, but this is outside the scope of this work).

Below is a summary of key element of the domain that the application needs to capture:

- Recorded defects within their systems and from our own inspections.

- List of drawings relevant to an asset integrity project. The following must be seen
    - Comments made with an inspection drawing.
    - Identification on whether a drawing has been approved or not.
- Mapping of structures experience during a project.
- Identification of who is submitting defects and remediations.

The domain of the application resides within the standard business practices. As such, standard record of approved remediations, and drawings must be seen. In addition, dates of submissions are important.

Functionality:

Below is a list of functionalities that the asset management team have requested be provided within the application:

- Provide reporting on the number of remediations that have not been identified on the design drawings.
- Provide a metric to the number of defects and remediations are in the system for a given structure.
- To provide record of drawings, remediations and defects for a given project. These will be used to track completed and defects that have been missed.
- To provide list of all drawings that have been issued to the client
- To provide a list of all defects that have been logged in the system.
- To provide a list of all outstanding remediation items.

Overall, they want to see themselves moving away from using excel manual data manipulation and move towards automated reporting.

Technology Architecture

This structural asset management system will be developed upon Python + MySQL.

In more detail, Flask and MyPHPAdmin will be two frameworks used to expediate the delivery of the application.

The reasonings to choosing this technology stack are as follows:

- The company requires a proof of concept prior to any additional funding provided. Python comparative to other technical stacks such as ASP.net infrastructure is simpler to build for small scale applications.
- Within the company, Python & MySQL are commonly known languages. As such, time to learn and develop a minimal viable product is managed when compared to other larger enterprise software's, such as SAP Sybase ASE.
- Both Python and MySQL are openly available software, which allows for low-cost overheads when prototyping this digital business solution.

The use of the two mentioned frameworks will reduce effort in developing the database and web application. Flask provides a high-level package to develop web requests, while MyPHPAdmin assist in the management of the database & provides user friendly tools to assist the development of a database.

Assumptions:

Below are assumptions made to develop the database model:

- This application is for one company to carry out integrity assessments of structures.
- A defect can be highlighted in many inspection drawings. (An example is when a defect was not completed after a past shutdown, therefore it will be captured in the next inspection).
- Not all remediations need to be represented on drawings. Sometimes, remediations are developed prior to development of drawings.
- A remediation does not exist without a defect.
- A project usually starts with a sparse amount of data and little drawings developed.
- Companies may ask workers to be contactable through a sales email account. Unlikely to occur but it has been experienced.
- Overtime, old emails maybe recycled for new staff members.

E/R Diagram

See last page for ER diagram

Schema

Below is a summary form of the schema:

```
TABLE drawing (

  ProjectId int(6) NOT NULL,

  DrawingType enum('DRG','INS') NOT NULL,

  DrawingNumber smallint(4) NOT NULL,

  RevisionNumber varchar(2) NOT NULL,

  DateIssued date DEFAULT NULL,

  RevisionStatus enum('Issued For Review','Issued For Tender','Issued For Construction','Not Issued For Construction','Issued For Information') NOT NULL,

  TitleName varchar(255) NOT NULL,

  RevisionDescription varchar(255) NOT NULL,

  PRIMARY KEY (ProjectId,DrawingType,DrawingNumber,RevisionNumber)

)



TABLE inspection_drawing (

  ProjectId int(6) NOT NULL,

  DrawingType enum('DRG','INS') NOT NULL,

  DrawingNumber smallint(4) NOT NULL,

  RevisionNumber varchar(2) NOT NULL,

  InspectionComment varchar(255) DEFAULT NULL,

  PRIMARY KEY (ProjectId,DrawingType,DrawingNumber,RevisionNumber)

)



TABLE inspection_drawing_highlights_defects (

  ProjectId int(6) NOT NULL,

  DrawingType enum('DRG','INS') NOT NULL,
  DrawingNumber smallint(4) NOT NULL,
  RevisionNumber varchar(2) NOT NULL,
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  PRIMARY KEY (ProjectId, DrawingType, DrawingNumber, RevisionNumber, StructureID, DefectID)
)


table design_drawing (
  ProjectId int(6) NOT NULL,
  DrawingType enum('DRG','INS') NOT NULL,
  DrawingNumber smallint(4) NOT NULL,
  RevisionNumber varchar(2) NOT NULL,
  ApprovedStatus enum('Not Yet Approved','Requesting Approval','Approvial Denied','Approved')
NOT NULL,
  PRIMARY KEY (ProjectId,DrawingType,DrawingNumber,RevisionNumber)
)


table drawing_shows_structure (
  ProjectId int(6) NOT NULL,
  DrawingType enum('DRG','INS') NOT NULL,
  DrawingNumber smallint(4) NOT NULL,
  RevisionNumber varchar(2) NOT NULL,
  StructureID int NOT NULL,
  PRIMARY KEY (ProjectID, DrawingType, DrawingNumber, RevisionNumber, StructureID)
)


table structure (
  StructureID int NOT NULL AUTO_INCREMENT,
  StructureName varchar(255) NOT NULL,
  PRIMARY KEY (StructureID)
)


TABLE structure_apart_of (
  ParentStructureID int NOT NULL,
  ChildStructureID int NOT NULL,
  PRIMARY KEY (ParentStructureID, ChildStructureID)
)


table defect (
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  DefectClassification varchar(2),
  DefectSeverity smallint(1),
  RiskRating enum('high','medium','low'),
  DateGenerated date NOT NULL,
  LastUpdated date DEFAULT NULL,
  PRIMARY KEY (DefectID,StructureID)
)


TABLE defect_comments (
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  Comment varchar(255) NOT NULL,
  PRIMARY KEY (StructureID, DefectID, Comment)
)
```

```
TABLE defect_photos (
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  Photo varchar(255) NOT NULL,
  PRIMARY KEY (StructureID, DefectID, Photo)
)


TABLE worker (
  WorkerID int NOT NULL AUTO_INCREMENT,
  FirstName varchar(50) NOT NULL,
  LastName varchar(50) NOT NULL,
  PhoneNumber varchar(15) DEFAULT NULL,
  Email varchar(255) NOT NULL,
  PRIMARY KEY (WorkerID)
)


TABLE worker_submitted_defect (
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  WorkerID int NOT NULL,
  PRIMARY KEY (StructureID,DefectID,WorkerID)
)


TABLE remediation (
  RemediationID int NOT NULL AUTO_INCREMENT,
  DateSubmitted date NOT NULL,
  DateUpdated date DEFAULT NULL,
  EstimateCost int, /* the estimations are in the thousands. Cents are never used. */
  ApprovedStatus enum('Not Yet Approved','Requesting Approval','Approvial Denied','Approved')
NOT NULL,
  PRIMARY KEY (RemediationID)
)



table design_drawing_proposes_remediation (
  ProjectId int(6) NOT NULL,
  DrawingType enum('DRG','INS') NOT NULL,
  DrawingNumber smallint(4) NOT NULL,
  RevisionNumber varchar(2) NOT NULL,
  RemediationID int NOT NULL,
  PRIMARY KEY (ProjectId,DrawingType,DrawingNumber,RevisionNumber,RemediationID)
)



TABLE worker_generated_remediation (
  RemediationID int NOT NULL,
  WorkerID int NOT NULL,
  PRIMARY KEY (RemediationID, WorkerID)
)



TABLE defect_covered_by_remediation (
  StructureID int NOT NULL,
  DefectID int NOT NULL,
  RemediationID int NOT NULL,
  PRIMARY KEY (StructureID, DefectID, RemediationID)
)
```

Key Mappings:
Note, some have been put into parenthesis to state a grouping. Expanded form may take up too much space.

Defect.StructureID -->Structure.structureID
InspectionDrawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber) -->Drawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)
DesignDrawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber) -->Drawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)
Shows.(ProjectId, DrawingType, DrawingNumber, RevisionNumber) -->Drawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)
Shows.StructureID -->Structure.StructureID
Proposes.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)-->DesignDrawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)
Proposes.RemediationID -->Remediation.RemediationID
CoveredBy.RemediationID -->Remediation.RemediationID
CoveredBy.DefectID -->Defect.DefectID
CoveredBy.StructureID -->Defect.StructureID
structure_apart_of.ParentStructureID -->Structure.StructureID
structure_apart_of.ChildStructureID -->Structure.StructureID
Generated.WorkerID -->Worker.WorkerID
Generated.RemediationID -->Remediation.RemediationID
Repairs.RemediationID -->Remediation.RemediationID
Repairs.StructureID -->Structure.StructureID
Submitted.DefectID -->Defect.DefectID
Submitted.StructureID -->Defect.StructureID
Submitted.WorkderID --> Worker.WorkerID
Highlights.DefectID -->Defect.DefectID
Highlights.StructureID -->Defect.StructureID
Highlights.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)--InspectionDrawing.(ProjectId, DrawingType, DrawingNumber, RevisionNumber)
DefectComments.DefectID -->Defect.DefectID
DefectComments.StructureID -->Defect.StructureID
DefectPhotos.DefectID -->Defect.DefectID
DefectPhotos.StructureID -->Defect.StructureID

Functional Dependencies & Normalization

This section will go through both the defining of functional dependencies and normalization checks to BCNF. BCNF was chosen to make sure no anomalies occur.

Drawing(<u>ProjectId</u>, <u>DrawingType</u>, <u>DrawingNumber</u>, <u>RevisionNumber</u>, DateIssued, RevisionStatus, TitleName, RevisionDescription)

FD:
<u>ProjectId</u>, <u>DrawingType</u>, <u>DrawingNumber</u>, <u>RevisionNumber</u> -> DateIssued, RevisionStatus, TitleName, RevisionDescription

There is the only one functional dependency for this drawing, which is the functional dependency from the PK to the non-prime attributes.

In the UoD, a drawing is uniqualy identified by its drawing ID, which composes of what is listed above. Within each drawing, the date issued, revision status, title name and revision description can be found. These attributes of the drawing give important contextual information when managing contractual obligations or just wanting to understand them.

BCNF check:

- No implicit FD will lead to a non-closure as there is only one FD, the functional dependency made from PK.
    - Augmentations will only make super keys.
    - Reflexive transformation will make trivial FDs.
    - Decomposition can be of the PK, but it will lead to LHS of implicit FD all being equal to the PK.
    - No transatives transformations readily available.
- As all FDs are superkey based on the closures, BCNF is not violated.
- Therefore, the relation is BCNF.

InspectionDrawing(**<u>ProjectId, DrawingType, DrawingNumber, RevisionNumber</u>**, InspectionComment)

FD:
<u>ProjectId, DrawingType, DrawingNumber, RevisionNumber</u> -> InspectionComment

ProjectId, DrawingType, DrawingNumber, RevisionNumber are the foreign PK of the entity. As such, a FD is present between the PK and inspection comment.

BCNF check:

- No implicit FD as there is only one FD with only one prime attribute, which is PK.
- As the FD is a superkey, BCNF is not violated.
- Therefore, the relation is BCNF.

DesignDrawing(**<u>ProjectId, DrawingType, DrawingNumber, RevisionNumber</u>**, ApprovedStatus)

<u>ProjectId, DrawingType, DrawingNumber, RevisionNumber</u> -> ApprovedStatus

ProjectId, DrawingType, DrawingNumber, RevisionNumber are the foreign PK of the entity. As such, a FD is present between the PK and approved status.

BCNF check:

- No implicit FD as there is only one FD with only one prime attribute, which is PK.
- As the FD is a superkey, BCNF is not violated.
- Therefore, the relation is BCNF.

Structure(<u>StructureID</u>, StructureName)

<u>StructureID</u> -> StructureName

StructureID is the PK of the entity. As such, a FD is present between the PK and StructureName.

BCNF check:

- ○ No implicit FD as there is only one FD with only one prime attribute, which is PK.
- ○ As the FD is a superkey, BCNF is not violated.
- ○ Therefore, the relation is BCNF.

Remediation(<u>RemediationID</u>, DateGenerated, DateUpdated, EstimatedCost, ApprovedStatus)

<u>RemediationID</u> -> DateGenerated, DateUpdated, EstimatedCost, ApprovedStatus

RemediationID is the PK of the entity. As such, a FD is present between the PK and the non-prime attributes, which are listed above.

BCNF check:

- ○ No implicit FD as there is only one FD with only one prime attribute, which is PK.
- ○ As the FD is a superkey, BCNF is not violated.
- ○ Therefore, the relation is BCNF.

Defect(**<u>StructureID</u>**, <u>DefectID</u>, DefectClassification, DefectSeverity, RiskRating, DateGenerated, LastUpdated)

<u>StructureID</u>, <u>DefectID</u> -> DefectClassification, DefectSeverity, RiskRating, DateGenerated, LastUpdated

StructureID and DefectID make up the primary key of this entity. As this is a weak entity, a foreign and partial key are needed to make a unique identification of a tuple with this relation.

As such, a FD is present between this nominate key, (i.e. PK), and the non-prime attributes, which are listed above.

BCNF check:

- ○ No implicit FD will lead to a non-closure as there is only one FD, the functional dependency made from PK.
  - Augmentations will only make super keys.
  - Reflexive transformation will make trivial FDs.
  - Decomposition can be of the PK, but it will lead to LHS of implicit FD all being equal to the PK.
  - No transatives transformations readily available.
- ○ As all FDs are superkey based on the closures, BCNF is not violated.
- ○ Therefore, the relation is BCNF.

Worker(<u>WorkerID</u>, FirstName, LastName, PhoneNumber, Email)

<u>WorkerID</u> -> FirstName, LastName, PhoneNumber, Email

WorkerID is the PK of the entity. As such, a FD is present between the PK and the non-prime attributes, which are listed above.

BCNF check:

- No implicit FD as there is only one FD with only one prime attribute, which is PK.
- As the FD is a superkey, BCNF is not violated.
- Therefore, the relation is BCNF.

Shows(**ProjectId, DrawingType, DrawingNumber, RevisionNumber**, **StructureID**)

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Proposes(**ProjectId, DrawingType, DrawingNumber, RevisionNumber**, **RemediationID**)

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

CoveredBy(**RemediationID**, **DefectID**, **StructureID**)

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Generated(**WorkerID**, **RemediationID**)

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Repairs(<u>**RemediationID**</u>, <u>**StructureID**</u>)

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Submitted(<u>**DefectID**</u>, <u>**StructureID**</u>, <u>**WorkerID**</u>)

Trivial: <u>**DefectID**</u>, <u>**StructureID**</u>, <u>**WorkerID**</u> -> <u>**DefectID**</u>, <u>**StructureID**</u>, <u>**WorkerID**</u>

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Highlights(<u>**DefectID**</u>, <u>**StructureID**</u>, <u>**ProjectId, DrawingType, DrawingNumber, RevisionNumber**</u>)

Trivial: <u>**DefectID**</u>, <u>**StructureID**</u>, <u>**ProjectId, DrawingType, DrawingNumber, RevisionNumber**</u> -> <u>**DefectID**</u>, <u>**StructureID**</u>, <u>**ProjectId, DrawingType, DrawingNumber, RevisionNumber**</u>

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

ApartOf(<u>**ParentStructureID**</u> , <u>**ChildStructureID)**</u>

Trivial: <u>**ParentStructureID**</u> , <u>**ChildStructureID**</u> -> <u>**ParentStructureID**</u>, <u>**ChildStructureID**</u>

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

DefectComments(<u>**StructureID**</u>, <u>**DefectID**</u>, <u>Comment)</u>

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

DefectPhotos(**<u>StructureID</u>**, **<u>DefectID</u>**, <u>Photo)</u>

Trivial FD

There is no non-trivial FD that makes up this relation.

BCNF Check:

- Excluding the trivial FD, we have no FD. As there is no non-trivial FDs, there is no redudancy. Therefore BCNF has been not violated and as such satisfied.

Summary table of Nomalized Tables

Below the table is split into three columns, table name, Primary Key and Foreign Keys. Please note, that some tables have to have their primary key be generated from foreign keys. As such, you will see attributes in the FK column being present in the PK. (Parenthesis has been used to highlight tables that have a primary key that is a composite of prime attributes).

<u>Table: List of Table Names and Respective Primary Key and Foreign Keys</u>

Table Name	PK	FK
Drawing	(ProjectId, DrawingType, DrawingNumber, RevisionNumber)	
InspectionDrawing	ProjectId, DrawingType, DrawingNumber, RevisionNumber	ProjectId, DrawingType, DrawingNumber, RevisionNumber
DesignDrawing	ProjectId, DrawingType, DrawingNumber, RevisionNumber	ProjectId, DrawingType, DrawingNumber, RevisionNumber
Structure	StructureID	
Remediation	RemediationID	
Defect	DefectID,StructureID	
Worker	WorkerID	
DrawingShowsStructure	(ProjectId, DrawingType, DrawingNumber, RevisionNumber, StructureID)	ProjectId, DrawingType, DrawingNumber, RevisionNumber, StructureID
DesignDrawing	(ProjectId, DrawingType, DrawingNumber, RevisionNumber, RemediationID)	ProjectId, DrawingType, DrawingNumber, RevisionNumber, RemediationID
DefectCoveredByRemediation	(RemediationID, DefectID, StructureID)	RemediationID, DefectID, StructureID

WorkerGeneratedRemediation	(WorkerID, RemediationID)	WorkerID, RemediationID
RemediationRepairsStructure	(RemediationID, StructureID)	RemediationID, StructureID
WorkerSubmittedDefect	(DefectID, StructureID, WorkerID)	DefectID, StructureID, WorkerID
InspectionDrawingHighlightsDefects	(DefectID, StructureID, ProjectId, DrawingType, DrawingNumber, RevisionNumber)	DefectID, StructureID, ProjectId, DrawingType, DrawingNumber, RevisionNumber
structureApartOf	(ParentStructureID , ChildStructureID)	StructureID , StructureID
DefectComments	(StructureID, DefectID, Comment)	StructureID, DefectID
DefectPhotos	(StructureID, DefectID, Photo)	StructureID, DefectID

SQL dump

Note, it was identified tables are placed in lower case in MySQL. Therefore, there was a shift in capilization of works to all lowercase and underscored.

Below is the SQL dump:

```
-- phpMyAdmin SQL Dump
-- version 5.0.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Apr 17, 2022 at 02:07 PM
-- Server version: 10.4.17-MariaDB
-- PHP Version: 8.0.2

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `test`
--

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

--
-- Table structure for table `defect`
--
```

```sql
CREATE TABLE `defect` (
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL,
  `DefectClassification` varchar(2) DEFAULT NULL,
  `DefectSeverity` smallint(1) DEFAULT NULL,
  `RiskRating` enum('high','medium','low') DEFAULT NULL,
  `DateGenerated` date NOT NULL,
  `LastUpdated` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `defect`
--

INSERT INTO `defect` (`StructureID`, `DefectID`, `DefectClassification`,
`DefectSeverity`, `RiskRating`, `DateGenerated`, `LastUpdated`) VALUES
(1, 1, 'SS', 3, 'medium', '2020-09-13', NULL),
(7, 1, 'SS', 1, 'low', '2020-09-13', '2020-09-14'),
(8, 1, 'SC', 2, 'low', '2020-09-13', '2020-09-14'),
(9, 1, 'SC', 3, 'medium', '2020-09-13', '2020-09-14'),
(1, 2, 'SS', 3, 'medium', '2020-09-13', '2020-09-14'),
(8, 2, 'SC', 3, 'medium', '2020-09-13', '2020-09-14'),
(9, 2, 'SC', 3, 'medium', '2020-09-13', '2020-09-14'),
(1, 3, 'SS', 3, 'high', '2020-09-13', '2020-09-14'),
(1, 4, 'SS', 3, 'high', '2020-09-13', '2020-09-14'),
(1, 5, 'SC', 4, 'high', '2020-09-13', '2020-09-14'),
(1, 6, 'SC', 4, 'high', '2020-09-13', '2020-09-14');

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

--
-- Table structure for table `defect_comments`
--

CREATE TABLE `defect_comments` (
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL,
  `Comment` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `defect_comments`
--

INSERT INTO `defect_comments` (`StructureID`, `DefectID`, `Comment`) VALUES
(1, 1, 'found coroded bolt on major connection.'),
(1, 2, 'identified major corrotion to maj. transfer beam.'),
(1, 2, 'the member in question will need to be closely monitored in future.'),
(1, 3, 'majority of section loss of column.'),
(1, 4, 'corroded nuts of anchors'),
(1, 5, 'spalling of concrete interface to anchors'),
(7, 1, 'broken handrail'),
(8, 1, 'spalling concrete at cover'),
(8, 2, 'spalling next to anchors'),
(9, 1, '1 sqm of spalling concrete'),
(9, 2, 'spalling concrete');

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

--
-- Table structure for table `defect_covered_by_remediation`
--

CREATE TABLE `defect_covered_by_remediation` (
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL,
  `RemediationID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `defect_covered_by_remediation`
--

INSERT INTO `defect_covered_by_remediation` (`StructureID`, `DefectID`,
`RemediationID`) VALUES
(1, 1, 1),
(1, 2, 2),
(1, 3, 3),
(1, 4, 4),
(1, 5, 5),
(1, 6, 6);

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

--
-- Table structure for table `defect_photos`
--

CREATE TABLE `defect_photos` (
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL,
  `Photo` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `defect_photos`
```

```sql
--

INSERT INTO `defect_photos` (`StructureID`, `DefectID`, `Photo`) VALUES
(1, 1, '1.jpg'),
(1, 2, '2.jpg'),
(1, 2, '3.jpg'),
(1, 3, '4.jpg'),
(1, 4, '5.jpg'),
(1, 5, '6.jpg'),
(1, 6, '7.jpg'),
(7, 1, '8.jpg'),
(8, 1, '9.jpg'),
(8, 2, '10.jpg'),
(9, 1, '11.jpg'),
(9, 2, '12.jpg');

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

--
-- Table structure for table `design_drawing`
--

CREATE TABLE `design_drawing` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `ApprovedStatus` enum('Not Yet Approved','Requesting Approval','Approvial
Denied','Approved') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `design_drawing`
--

INSERT INTO `design_drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`,
`RevisionNumber`, `ApprovedStatus`) VALUES
(511923, 'DRG', 1, 'A', 'Not Yet Approved'),
(511923, 'DRG', 2, 'A', 'Requesting Approval'),
(511923, 'DRG', 3, 'A', 'Approvial Denied'),
(511923, 'DRG', 4, '0', 'Approved'),
(511923, 'DRG', 5, '0', 'Approved'),
(511923, 'DRG', 6, '0', 'Approved');

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

--
-- Table structure for table `design_drawing_proposes_remediation`
```

```sql
--

CREATE TABLE `design_drawing_proposes_remediation` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `RemediationID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `design_drawing_proposes_remediation`
--

INSERT INTO `design_drawing_proposes_remediation` (`ProjectId`, `DrawingType`,
`DrawingNumber`, `RevisionNumber`, `RemediationID`) VALUES
(511923, 'DRG', 2, 'A', 1),
(511923, 'DRG', 3, 'A', 2),
(511923, 'DRG', 4, '0', 3),
(511923, 'DRG', 5, '0', 4),
(511923, 'DRG', 6, '0', 5);

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

--
-- Table structure for table `drawing`
--

CREATE TABLE `drawing` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `DateIssued` date DEFAULT NULL,
  `RevisionStatus` enum('Issued For Review','Issued For Tender','Issued For
Construction','Not Issued For Construction','Issued For Information') NOT
NULL,
  `TitleName` varchar(255) NOT NULL,
  `RevisionDescription` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `drawing`
--

INSERT INTO `drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`,
`RevisionNumber`, `DateIssued`, `RevisionStatus`, `TitleName`,
`RevisionDescription`) VALUES
```

```sql
(511923, 'DRG', 1, 'A', NULL, 'Issued For Review', 'Dump Station Steel Work General Notes', 'Initial issue for review'),
(511923, 'DRG', 2, 'A', NULL, 'Issued For Review', 'Dump Station Hopper Steel Work Plans', 'Initial issue for review'),
(511923, 'DRG', 3, 'A', NULL, 'Issued For Review', 'Dump Station Hopper Steel Work Elevations', 'Initial issue for review'),
(511923, 'DRG', 4, '0', '2020-09-14', 'Issued For Construction', 'Dump Station Steel Work Details', 'P543 Issue for construction'),
(511923, 'DRG', 5, '0', '2020-09-14', 'Issued For Construction', 'Dump Station Civil Work General Notes', 'P543 Issue for construction'),
(511923, 'DRG', 6, '0', '2020-09-14', 'Issued For Construction', 'Dump Station Additional Scope', 'P543 Issue for construction'),
(611524, 'INS', 1, 'A', '2020-09-16', 'Issued For Information', 'Silo 1 general inspection plan', 'Issue for information'),
(611524, 'INS', 2, 'A', '2020-09-16', 'Issued For Information', 'Silo 1 inspection foundation sheet 1', 'Issue for information'),
(611524, 'INS', 3, 'A', '2020-09-16', 'Issued For Information', 'Silo 1 inspection foundation sheet 2', 'Issue for information'),
(611524, 'INS', 4, 'A', '2020-09-17', 'Issued For Information', 'Silo 1 inspection walls sheet 1', 'Issue for information'),
(611524, 'INS', 5, 'A', '2020-09-17', 'Issued For Information', 'Silo 1 inspection walls sheet 2', 'Issue for information');

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

--
-- Table structure for table `drawing_shows_structure`
--

CREATE TABLE `drawing_shows_structure` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `StructureID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `drawing_shows_structure`
--

INSERT INTO `drawing_shows_structure` (`ProjectId`, `DrawingType`, `DrawingNumber`, `RevisionNumber`, `StructureID`) VALUES
(511923, 'DRG', 1, 'A', 1),
(511923, 'DRG', 2, 'A', 3),
(511923, 'DRG', 3, 'A', 3),
(511923, 'DRG', 4, '0', 3),
(511923, 'DRG', 5, '0', 1),
(611524, 'INS', 1, 'A', 7),
(611524, 'INS', 2, 'A', 8),
(611524, 'INS', 3, 'A', 8),
(611524, 'INS', 4, 'A', 9),
(611524, 'INS', 5, 'A', 9);

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

--
-- Table structure for table `inspection_drawing`
--

CREATE TABLE `inspection_drawing` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `InspectionComment` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `inspection_drawing`
--

INSERT INTO `inspection_drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`,
`RevisionNumber`, `InspectionComment`) VALUES
(611524, 'INS', 1, 'A', NULL),
(611524, 'INS', 2, 'A', 'underside of foundation was not inspected'),
(611524, 'INS', 3, 'A', 'all anchors were successfully checked'),
(611524, 'INS', 4, 'A', 'lower walls pertained spalling of concrete. '),
(611524, 'INS', 5, 'A', 'upper region of wall partially inspected.');

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

--
-- Table structure for table `inspection_drawing_highlights_defects`
--

CREATE TABLE `inspection_drawing_highlights_defects` (
  `ProjectId` int(6) NOT NULL,
  `DrawingType` enum('DRG','INS') NOT NULL,
  `DrawingNumber` smallint(4) NOT NULL,
  `RevisionNumber` varchar(2) NOT NULL,
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```sql
-- Dumping data for table `inspection_drawing_highlights_defects`
--

INSERT INTO `inspection_drawing_highlights_defects` (`ProjectId`,
`DrawingType`, `DrawingNumber`, `RevisionNumber`, `StructureID`, `DefectID`)
VALUES
(611524, 'INS', 1, 'A', 7, 1),
(611524, 'INS', 2, 'A', 8, 1),
(611524, 'INS', 3, 'A', 8, 2),
(611524, 'INS', 4, 'A', 9, 1),
(611524, 'INS', 5, 'A', 9, 2);

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

--
-- Table structure for table `remediation`
--

CREATE TABLE `remediation` (
  `RemediationID` int(11) NOT NULL,
  `DateSubmitted` date NOT NULL,
  `DateUpdated` date DEFAULT NULL,
  `EstimateCost` int(11) DEFAULT NULL,
  `ApprovedStatus` enum('Not Yet Approved','Requesting Approval','Approvial
Denied','Approved') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `remediation`
--

INSERT INTO `remediation` (`RemediationID`, `DateSubmitted`, `DateUpdated`,
`EstimateCost`, `ApprovedStatus`) VALUES
(1, '2020-09-13', '2020-09-14', 1000, 'Approved'),
(2, '2020-09-13', '2020-09-14', 4000, 'Approved'),
(3, '2020-09-13', '2020-09-14', 3500, 'Approved'),
(4, '2020-09-13', '2020-09-14', 4000, 'Not Yet Approved'),
(5, '2020-09-13', NULL, NULL, 'Not Yet Approved'),
(6, '2020-09-14', NULL, NULL, 'Not Yet Approved');

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

--
-- Table structure for table `structure`
--

CREATE TABLE `structure` (
  `StructureID` int(11) NOT NULL,
  `StructureName` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `structure`
--

INSERT INTO `structure` (`StructureID`, `StructureName`) VALUES
(1, 'Dump Station 1'),
(2, 'DS1 basement'),
(3, 'DS1 hopper'),
(4, 'DS1 hopper columns'),
(5, 'DS1 upper hopper'),
(6, 'DS1 portal frame'),
(7, 'Silo 1'),
(8, 'S1 Foundations'),
(9, 'S1 Walls');

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

--
-- Table structure for table `structure_apart_of`
--

CREATE TABLE `structure_apart_of` (
  `ParentStructureID` int(11) NOT NULL,
  `ChildStructureID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `structure_apart_of`
--

INSERT INTO `structure_apart_of` (`ParentStructureID`, `ChildStructureID`)
VALUES
(1, 2),
(1, 3),
(1, 6),
(3, 4),
(3, 5),
(7, 8),
(7, 9);

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

--
-- Table structure for table `worker`
--
```

```sql
CREATE TABLE `worker` (
  `WorkerID` int(11) NOT NULL,
  `FirstName` varchar(50) NOT NULL,
  `LastName` varchar(50) NOT NULL,
  `PhoneNumber` varchar(15) DEFAULT NULL,
  `Email` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `worker`
--

INSERT INTO `worker` (`WorkerID`, `FirstName`, `LastName`, `PhoneNumber`,
`Email`) VALUES
(1, 'Garry', 'Smith', '+61487912002', 'garry.smith@company.com.au'),
(2, 'Ashleigh', 'Bowman', '+61487912002', 'ashleigh.bowman@company.com.au'),
(3, 'Samuel', 'Barthon', '+61487912002', 'samuel.barthon@company.com.au'),
(4, 'Zack', 'Smith', NULL, 'zack.smith@company.com.au'),
(5, 'Kevin', 'Shell', NULL, 'kevin.shell@company.com.au');

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

--
-- Table structure for table `worker_generated_remediation`
--

CREATE TABLE `worker_generated_remediation` (
  `RemediationID` int(11) NOT NULL,
  `WorkerID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `worker_generated_remediation`
--

INSERT INTO `worker_generated_remediation` (`RemediationID`, `WorkerID`)
VALUES
(1, 1),
(2, 1),
(3, 2),
(4, 2),
(5, 2),
(6, 2);

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

--
```

```sql
-- Table structure for table `worker_submitted_defect`
--

CREATE TABLE `worker_submitted_defect` (
  `StructureID` int(11) NOT NULL,
  `DefectID` int(11) NOT NULL,
  `WorkerID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `worker_submitted_defect`
--

INSERT INTO `worker_submitted_defect` (`StructureID`, `DefectID`, `WorkerID`)
VALUES
(1, 1, 1),
(1, 2, 1),
(1, 3, 2),
(1, 4, 2),
(1, 5, 2),
(1, 6, 2),
(7, 1, 3),
(8, 1, 3),
(8, 2, 4),
(9, 1, 4),
(9, 2, 5);

--
-- Indexes for dumped tables
--

--
-- Indexes for table `defect`
--
ALTER TABLE `defect`
  ADD PRIMARY KEY (`DefectID`,`StructureID`),
  ADD KEY `defect_fk_1` (`StructureID`);

--
-- Indexes for table `defect_comments`
--
ALTER TABLE `defect_comments`
  ADD PRIMARY KEY (`StructureID`,`DefectID`,`Comment`);

--
-- Indexes for table `defect_covered_by_remediation`
--
ALTER TABLE `defect_covered_by_remediation`
```

```sql
  ADD PRIMARY KEY (`StructureID`,`DefectID`,`RemediationID`),
  ADD KEY `defect_covered_by_remediation_fk_1` (`RemediationID`);

--
-- Indexes for table `defect_photos`
--
ALTER TABLE `defect_photos`
  ADD PRIMARY KEY (`StructureID`,`DefectID`,`Photo`);

--
-- Indexes for table `design_drawing`
--
ALTER TABLE `design_drawing`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`);

--
-- Indexes for table `design_drawing_proposes_remediation`
--
ALTER TABLE `design_drawing_proposes_remediation`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`,`RemediationID`),
  ADD KEY `design_drawing_proposes_remediation_fk_1` (`RemediationID`);

--
-- Indexes for table `drawing`
--
ALTER TABLE `drawing`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`);

--
-- Indexes for table `drawing_shows_structure`
--
ALTER TABLE `drawing_shows_structure`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`,`StructureID`),
  ADD KEY `drawing_shows_structure_fk_2` (`StructureID`);

--
-- Indexes for table `inspection_drawing`
--
ALTER TABLE `inspection_drawing`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`);

--
-- Indexes for table `inspection_drawing_highlights_defects`
```

```
--
ALTER TABLE `inspection_drawing_highlights_defects`
  ADD PRIMARY KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`,`StructureID`,`Def
ectID`),
  ADD KEY `inspection_drawing_highlights_defects_fk_2`
(`StructureID`,`DefectID`);

--
-- Indexes for table `remediation`
--
ALTER TABLE `remediation`
  ADD PRIMARY KEY (`RemediationID`);

--
-- Indexes for table `structure`
--
ALTER TABLE `structure`
  ADD PRIMARY KEY (`StructureID`);

--
-- Indexes for table `structure_apart_of`
--
ALTER TABLE `structure_apart_of`
  ADD PRIMARY KEY (`ParentStructureID`,`ChildStructureID`),
  ADD KEY `structure_apart_of_fk_2` (`ChildStructureID`);

--
-- Indexes for table `worker`
--
ALTER TABLE `worker`
  ADD PRIMARY KEY (`WorkerID`);

--
-- Indexes for table `worker_generated_remediation`
--
ALTER TABLE `worker_generated_remediation`
  ADD PRIMARY KEY (`RemediationID`,`WorkerID`),
  ADD KEY `worker_generated_remediation_fk_2` (`WorkerID`);

--
-- Indexes for table `worker_submitted_defect`
--
ALTER TABLE `worker_submitted_defect`
  ADD PRIMARY KEY (`StructureID`,`DefectID`,`WorkerID`),
  ADD KEY `worker_submitted_defect_fk_2` (`WorkerID`);

--
```

```sql
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `remediation`
--
ALTER TABLE `remediation`
  MODIFY `RemediationID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT for table `structure`
--
ALTER TABLE `structure`
  MODIFY `StructureID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

--
-- AUTO_INCREMENT for table `worker`
--
ALTER TABLE `worker`
  MODIFY `WorkerID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `defect`
--
ALTER TABLE `defect`
  ADD CONSTRAINT `defect_fk_1` FOREIGN KEY (`StructureID`) REFERENCES
`structure` (`StructureID`) ON DELETE CASCADE;

--
-- Constraints for table `defect_comments`
--
ALTER TABLE `defect_comments`
  ADD CONSTRAINT `defect_comments_fk_1` FOREIGN KEY (`StructureID`,`DefectID`)
REFERENCES `defect` (`StructureID`, `DefectID`) ON DELETE CASCADE;

--
-- Constraints for table `defect_covered_by_remediation`
--
ALTER TABLE `defect_covered_by_remediation`
  ADD CONSTRAINT `defect_covered_by_remediation_fk_1` FOREIGN KEY
(`RemediationID`) REFERENCES `remediation` (`RemediationID`),
  ADD CONSTRAINT `defect_covered_by_remediation_fk_2` FOREIGN KEY
(`StructureID`,`DefectID`) REFERENCES `defect` (`StructureID`, `DefectID`);
```

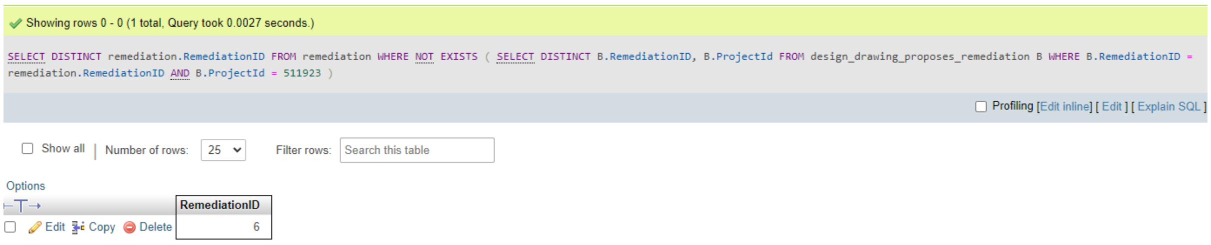
```sql
--
-- Constraints for table `defect_photos`
--
ALTER TABLE `defect_photos`
  ADD CONSTRAINT `defect_photos_fk_1` FOREIGN KEY (`StructureID`,`DefectID`)
REFERENCES `defect` (`StructureID`, `DefectID`) ON DELETE CASCADE;

--
-- Constraints for table `design_drawing`
--
ALTER TABLE `design_drawing`
  ADD CONSTRAINT `design_drawing_fk_1` FOREIGN KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`) REFERENCES
`drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`, `RevisionNumber`) ON
DELETE CASCADE;

--
-- Constraints for table `design_drawing_proposes_remediation`
--
ALTER TABLE `design_drawing_proposes_remediation`
  ADD CONSTRAINT `design_drawing_proposes_remediation_fk_1` FOREIGN KEY
(`RemediationID`) REFERENCES `remediation` (`RemediationID`),
  ADD CONSTRAINT `design_drawing_proposes_remediation_fk_2` FOREIGN KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`) REFERENCES
`design_drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`,
`RevisionNumber`);

--
-- Constraints for table `drawing_shows_structure`
--
ALTER TABLE `drawing_shows_structure`
  ADD CONSTRAINT `drawing_shows_structure_fk_1` FOREIGN KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`) REFERENCES
`drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`, `RevisionNumber`),
  ADD CONSTRAINT `drawing_shows_structure_fk_2` FOREIGN KEY (`StructureID`)
REFERENCES `structure` (`StructureID`);

--
-- Constraints for table `inspection_drawing`
--
ALTER TABLE `inspection_drawing`
  ADD CONSTRAINT `inspection_drawing_fk_1` FOREIGN KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`) REFERENCES
`drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`, `RevisionNumber`) ON
DELETE CASCADE;

--
-- Constraints for table `inspection_drawing_highlights_defects`
```

```sql
--
ALTER TABLE `inspection_drawing_highlights_defects`
  ADD CONSTRAINT `inspection_drawing_highlights_defects_fk_1` FOREIGN KEY
(`ProjectId`,`DrawingType`,`DrawingNumber`,`RevisionNumber`) REFERENCES
`inspection_drawing` (`ProjectId`, `DrawingType`, `DrawingNumber`,
`RevisionNumber`),
  ADD CONSTRAINT `inspection_drawing_highlights_defects_fk_2` FOREIGN KEY
(`StructureID`,`DefectID`) REFERENCES `defect` (`StructureID`, `DefectID`);

--
-- Constraints for table `structure_apart_of`
--
ALTER TABLE `structure_apart_of`
  ADD CONSTRAINT `structure_apart_of_fk_1` FOREIGN KEY (`ParentStructureID`)
REFERENCES `structure` (`StructureID`),
  ADD CONSTRAINT `structure_apart_of_fk_2` FOREIGN KEY (`ChildStructureID`)
REFERENCES `structure` (`StructureID`);

--
-- Constraints for table `worker_generated_remediation`
--
ALTER TABLE `worker_generated_remediation`
  ADD CONSTRAINT `worker_generated_remediation_fk_1` FOREIGN KEY
(`RemediationID`) REFERENCES `remediation` (`RemediationID`),
  ADD CONSTRAINT `worker_generated_remediation_fk_2` FOREIGN KEY (`WorkerID`)
REFERENCES `worker` (`WorkerID`);

--
-- Constraints for table `worker_submitted_defect`
--
ALTER TABLE `worker_submitted_defect`
  ADD CONSTRAINT `worker_submitted_defect_fk_1` FOREIGN KEY
(`StructureID`,`DefectID`) REFERENCES `defect` (`StructureID`, `DefectID`),
  ADD CONSTRAINT `worker_submitted_defect_fk_2` FOREIGN KEY (`WorkerID`)
REFERENCES `worker` (`WorkerID`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Screenshot

During a project, there will be several remediations spread across several drawings. The previous method was to go through all drawings and record all remediations referenced & record all remediations within a system. After, someone would via excel document go through all records and identify what remediations are not covered by the design drawings.

Below is a query to provide an automatic way to list out all remediations that have not yet been captured by design drawings. This is an important task that is needed to assure the company is addressing outstanding remediations.


Showing rows 0 - 0 (1 total, Query took 0.0027 seconds.)

```
SELECT DISTINCT remediation.RemediationID FROM remediation WHERE NOT EXISTS ( SELECT DISTINCT B.RemediationID, B.ProjectId FROM design_drawing_proposes_remediation B WHERE B.RemediationID = remediation.RemediationID AND B.ProjectId = 511923 )
```

Profiling [Edit inline] [Edit] [Explain SQL]

Show all | Number of rows: 25 Filter rows: Search this table

Options

	RemediationID
Edit Copy Delete	6

Below is the query and a short explanation:

Query:

```
SELECT DISTINCT remediation.RemediationID
FROM remediation
WHERE NOT EXISTS (
   SELECT DISTINCT B.RemediationID, B.ProjectId
   FROM design_drawing_proposes_remediation B
   WHERE B.RemediationID = remediation.RemediationID AND B.ProjectId = 511923
   )
```

Explanation of the query:

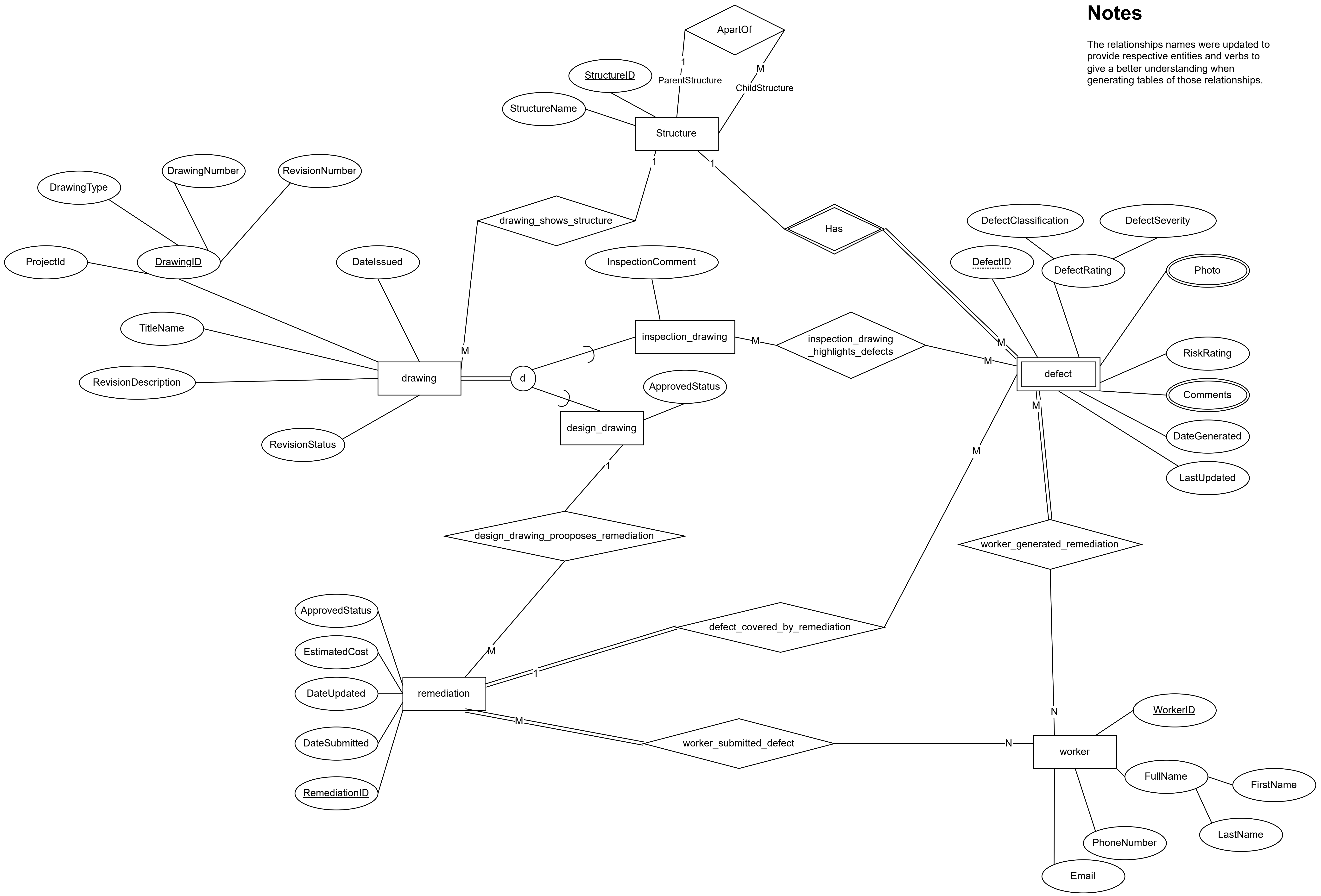
The query as a hole compares the rows of two tables and returns the difference between them.

In this scenario, table remediation will always have more or equal the number of distinct remediationsIDs when compared to design_drawing_proposes_remediation table. This is due to design_drawing_proposes_remediation is a table of FKs representing the relationship between two entities.

The query finds the distinct RemediationID from both tables. After, the NOT EXISTS is used to ignore RemediationIDs that are found in design_drawing_proposes_remediation. (Note, the WHERE segment in the paranthesis is identifying the matches between two distinct RemediationID lists).

To make the query useful for a project, where condition in respect to design_drawing_proposes_remediation table is identifying ProjectIds that are equal to a nominated value. In this case, it is project 511923.

ER DIAGRAM



Notes

The relationships names were updated to provide respective entities and verbs to give a better understanding when generating tables of those relationships.