Spatial Data Management Assignment 2 – Database Design and Creation

Contents

Part A – Database Creation

Although explanations of each element of the report are not required, some explanation has been added where it has been considered necessary, particularly where there have been minor deviations from the requirements outlined in assignments 1A and 1B to accommodate PostgreSQL limitations.

1. Conceptual UML Diagram

One alteration has been made to the conceptual UML diagram submitted in assignment 1B (Figure 1): The relationship cardinality between entities 'Drillhole Type' and 'Drillhole' should have be listed as one to many (Figure 2).

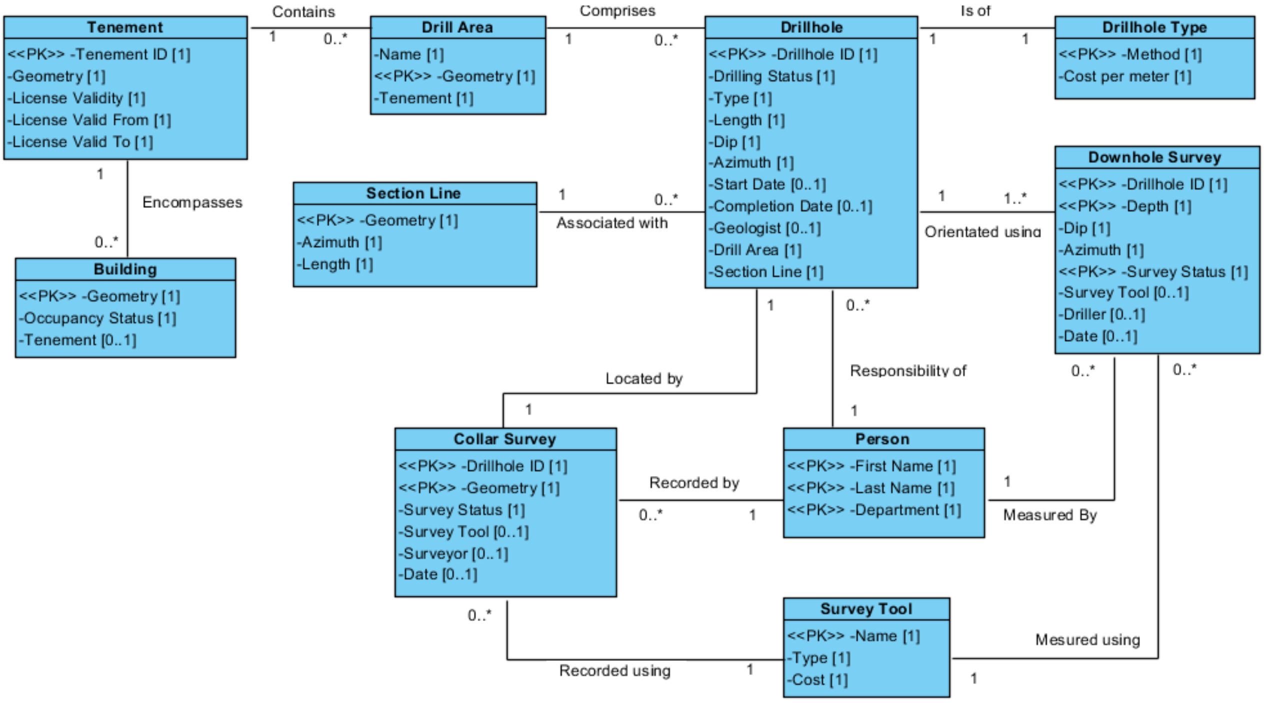


Figure 1: Conceptual UML diagram submitted in assignment 1B.

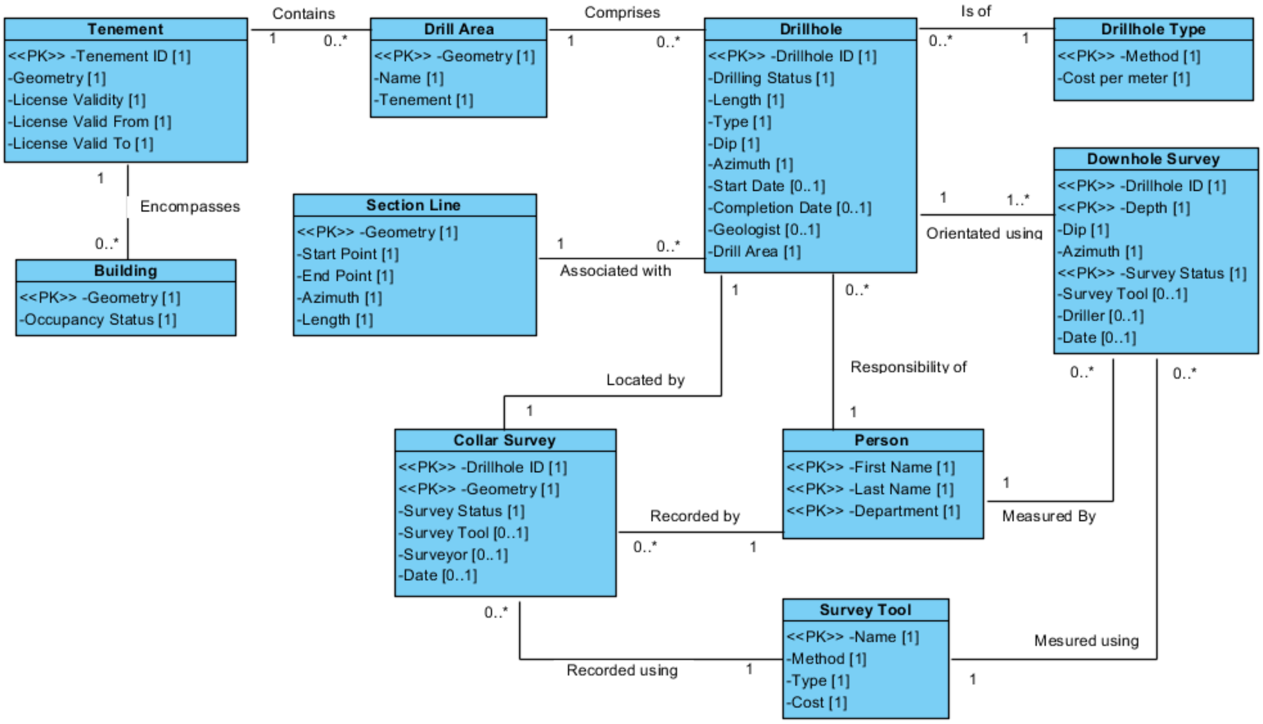


Figure 2: Updated conceptual UML diagram.

2. Functional Requirements – Mineral Exploration Drilling Program Tracking System

The database created allows the following questions to be answered through SQL queries. These queries use a combination of both spatial functions and joins. In total, 8 queries utilise spatial functions, and 6 utilise joins. The methods used to answer each functional requirement are listed in Table 1. Some of the required spatial functions have been revised since they were detailed in the system specification: where the required spatial functions have been replaced or new functions added, this has been highlighted.

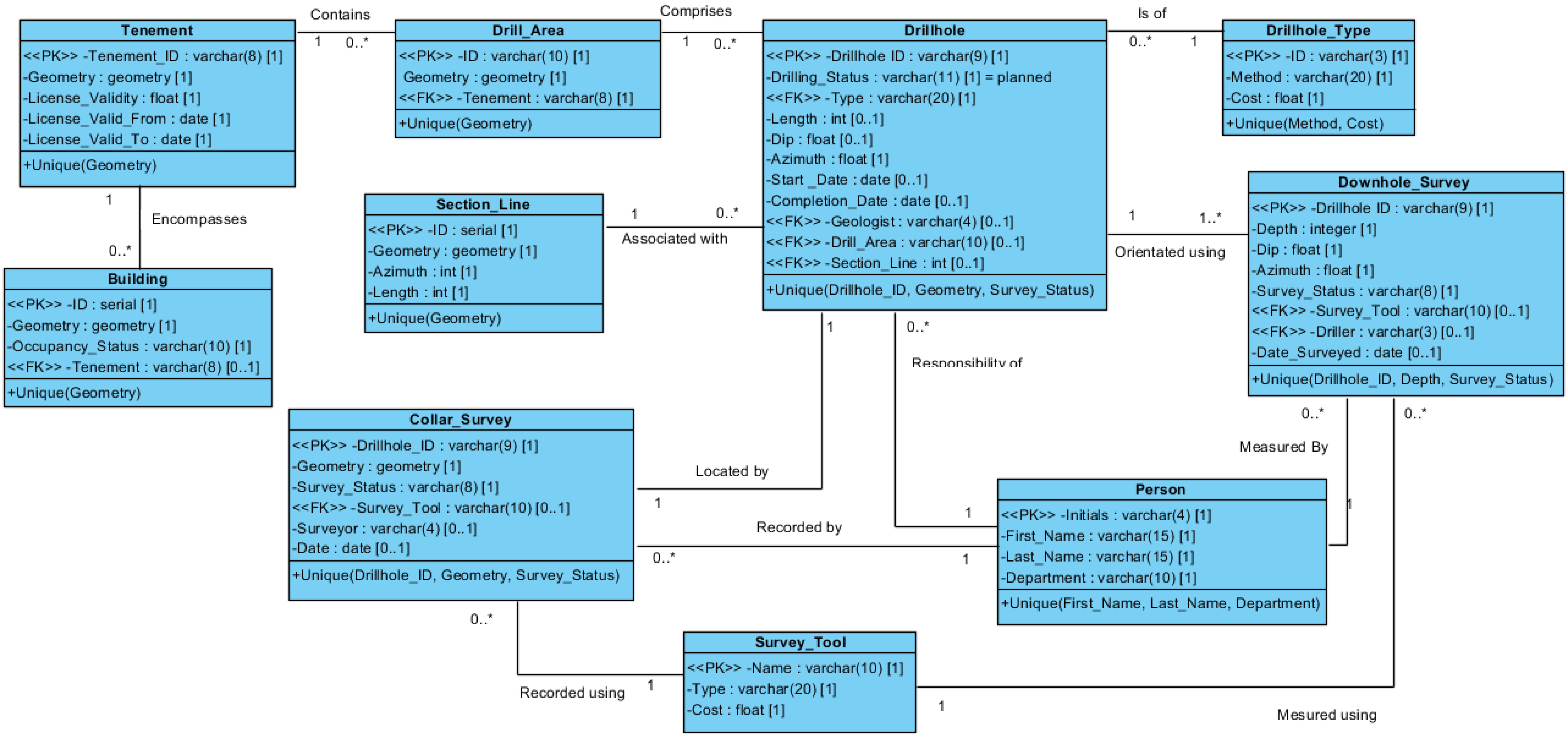
Table 1: Functional requirements of the database detailed in this report. Whether the SQL query required to meet this requirement will require either a join, or spatial function, or both, is detailed, in addition to the spatial operation required. Spatial operations required to meet these requirements have been revised.

	Functional Requirement	**Join**	**Spatial**	**3D**	**Required Operations**
1	Which tenement has the largest area?		✓		ST_Area
2	How many drillholes have been planned, are in progress, or have been completed in each drill area, and in which tenement are they located?	✓			
3	What percentage of all holes have been cancelled?				
4	What is the average number of holes drilled per tenement, per month?	✓			
5	Do any planned drillholes fall within 100 m of the occupied buildings, or 50 m of the unoccupied buildings? Which geologist is responsible?		✓	✓	~~ST_3DIntersects~~ ST_3Ddistance

					ST_3DDWithin ST_Transform
6	Do any of the drill areas overlap each other, or extend outside of the tenement area?		✓		ST_Overlaps
7	Do all drillholes fall within a drill area?		✓		ST_Contains
8	What is the longest section line and how many drillholes does it have on it (of any drilling status)?		✓		ST_Length
9	What is the largest distance between adjacent drillholes on each section line, as measured from their collar positions?	✓	✓		ST_Distance ST_Transform
10	What is the total drilling expense per tenement, per year, taking into account both hole type and downhole survey expenses?	✓			
11	Which drillholes fall more than 10 metres from their intended section lines? Are these inaccuracies the fault of one particular geologist?	✓	✓		ST_ShortestLine ST_Transform
12	How many drillholes in each tenement have been drilled (i.e. have a completion date), but are missing GPS or DGPS collar surveys?	✓			
13	What is the maximum distance a geologist had to travel between drillholes for which he/she was responsible for at one time?	✓	✓		ST_Distance ST_Transform

3. Logical UML Diagram

Attributes and entity names comprising more than one word have been altered to replace the space with an underscore to create table and column names respectively. Check constraints have not been added as they contained syntax that was not permissible by Visual Paradigm, however they remain as detailed in assignment 1B.


Tenement
<<PK>> -Tenement_ID : varchar(8) [1]
-Geometry : geometry [1]
-License_Validity : float [1]
-License_Valid_From : date [1]
-License_Valid_To : date [1]
+Unique(Geometry)
Contains
1
0..*
Drill_Area
<<PK>> -ID : varchar(10) [1]
Geometry : geometry [1]
<<FK>> -Tenement : varchar(8) [1]
+Unique(Geometry)
Comprises
1
0..*
Drillhole
<<PK>> -Drillhole ID : varchar(9) [1]
-Drilling_Status : varchar(11) [1] = planned
<<FK>> -Type : varchar(20) [1]
-Length : int [0..1]
-Dip : float [0..1]
-Azimuth : float [1]
-Start _Date : date [0..1]
-Completion_Date : date [0..1]
<<FK>> -Geologist : varchar(4) [0..1]
<<FK>> -Drill_Area : varchar(10) [0..1]
<<FK>> -Section_Line : int [0..1]
+Unique(Drillhole_ID, Geometry, Survey_Status)
Is of
0..*
1
Drillhole_Type
<<PK>> -ID : varchar(3) [1]
-Method : varchar(20) [1]
-Cost : float [1]
+Unique(Method, Cost)
1
Encompasses
0..*
Building
<<PK>> -ID : serial [1]
-Geometry : geometry [1]
-Occupancy_Status : varchar(10) [1]
<<FK>> -Tenement : varchar(8) [0..1]
+Unique(Geometry)
Section_Line
<<PK>> -ID : serial [1]
-Geometry : geometry [1]
-Azimuth : int [1]
-Length : int [1]
+Unique(Geometry)
1
0..*
Associated with
1
1..*
Orientated using
Downhole_Survey
<<PK>> -Drillhole ID : varchar(9) [1]
-Depth : integer [1]
-Dip : float [1]
-Azimuth : float [1]
-Survey_Status : varchar(8) [1]
<<FK>> -Survey_Tool : varchar(10) [0..1]
<<FK>> -Driller : varchar(3) [0..1]
-Date_Surveyed : date [0..1]
+Unique(Drillhole_ID, Depth, Survey_Status)
1
0..*
Responsibility of
0..*
0..*
Measured By
Collar_Survey
<<PK>> -Drillhole_ID : varchar(9) [1]
-Geometry : geometry [1]
-Survey_Status : varchar(8) [1]
<<FK>> -Survey_Tool : varchar(10) [0..1]
-Surveyor : varchar(4) [0..1]
-Date : date [0..1]
+Unique(Drillhole_ID, Geometry, Survey_Status)
Located by
1
Recorded by
0..*
1
1
1
Person
<<PK>> -Initials : varchar(4) [1]
-First_Name : varchar(15) [1]
-Last_Name : varchar(15) [1]
-Department : varchar(10) [1]
+Unique(First_Name, Last_Name, Department)
0..*
Recorded using
1
Survey_Tool
<<PK>> -Name : varchar(10) [1]
-Type : varchar(20) [1]
-Cost : float [1]
1
Mesured using

4. SQL Scripts

4.1. Schema Creation

The tables for this database have been created in a new schema called *drilldb*, created using the following script:

```
create schema drilldb;
```

A new schema was created to keep these new tables separate from tables currently existing in the public schema. In a commercial setting, this would also allow schema-specific user permissions to be assigned for each schema, so if additional database tables were to be added, such as geochemical assay information or geological mapping data, these could also be stored in separate schemas with different user permissions assigned.

4.2. Table Creation

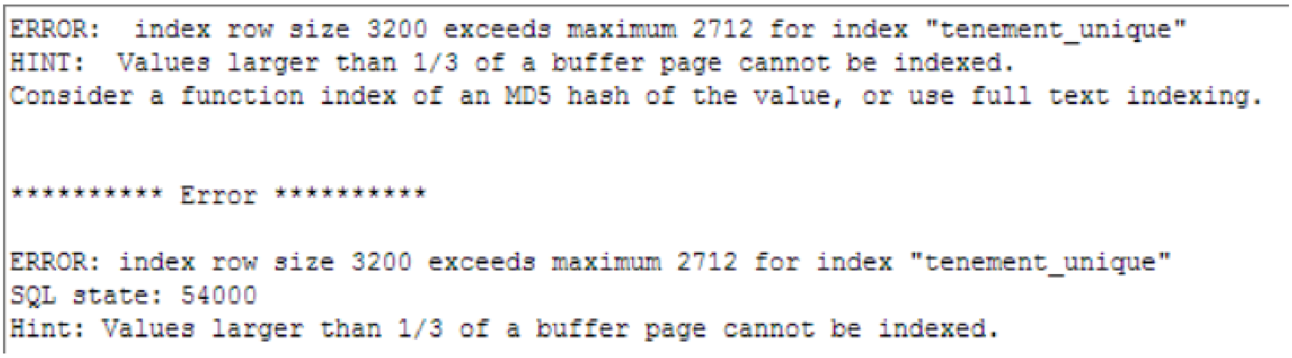
These SQL scripts are presented in the order in which they are required for successfully building the database, listed under the appropriate headings. Primary and foreign keys are assigned within the table creation script, and where possible, constraints are also assigned in this manner. Where it was not possible to implement attribute cardinality or constraints previously outlined, this is detailed.

Tenement

Table 'tenement' is based on the entity 'Tenement'.

```
create table drilldb.tenement
    (
    tenement_id character varying(8) not null primary key,
    license_validity float not null,
    license_valid_from date not null,
    license_valid_to date not null,

    constraint tenement_licence_check check (license_validity <= 5 and
license_valid_from < license_valid_to)

    );

select AddGeometryColumn ('drilldb','tenement','geometry',4326,'POLYGON',2,1);
--alter table drilldb.tenement add constraint tenement_unique unique(geometry);
```

It was necessary to remove the unique constraint on the geometry as the complexity of the tenement polygons caused the following error:



```
ERROR:  index row size 3200 exceeds maximum 2712 for index "tenement_unique"
HINT:  Values larger than 1/3 of a buffer page cannot be indexed.
Consider a function index of an MD5 hash of the value, or use full text indexing.


********** Error **********

ERROR: index row size 3200 exceeds maximum 2712 for index "tenement_unique"
SQL state: 54000
Hint: Values larger than 1/3 of a buffer page cannot be indexed.
```

Building

Table *'building'* is based on the entity *'Building'*.

```
create table drilldb.building
    (
    id serial not null primary key,
    occupancy_status character varying(10) not null,
    tenement character varying(8),

    foreign key (tenement) references drilldb.tenement (tenement_id)
    );

select AddGeometryColumn ('drilldb','building','geometry',4283,
    'POLYHEDRALSURFACE',3,1);

alter table drilldb.building add constraint building_unique unique(geometry);
```

Drill Area

Table *'drill_area'* is based on the entity *'Drill Area'*.

```
create table drilldb.drill_area
    (
    id character varying(10) not null primary key,
    tenement character varying(8) not null,

    foreign key (tenement) references drilldb.tenement (tenement_id)
    );

select AddGeometryColumn ('drilldb','drill_area','geometry',4326,'POLYGON',2,1);

alter table drilldb.drill_area add constraint drill_area_unique unique(geometry);
```

Section Line

Table *'section_line'* based on entity *'Section Line'*.

```
create table drilldb.section_line
    (
    id serial not null primary key,
    azimuth float not null,
    length float not null
    );

select AddGeometryColumn
     ('drilldb','section_line','geometry',4326,'LINESTRING',2,1);

alter table drilldb.section_line
    add constraint section_line_unique unique(geometry);
alter table drilldb.section_line
    add constraint section_line_npoints_check check (st_npoints(geometry) = 2);
```

Drillhole Type

Table *'drillhole_type'* based on entity *'Drillhole Type'*.

```
create table drilldb.drillhole_type
    (
    id character varying(3) not null primary key,
    method character varying(20) not null,
    cost float not null

    constraint cost_check check (cost > 0),
    constraint type_unique unique (method, cost)
    );
```

Survey Tool

Table *'survey_tool'* based on entity *'Survey Tool'*.

```
create table drilldb.survey_tool
    (
    name character varying(10) not null primary key,
    type character varying(20) not null,
    cost float not null,

    constraint cost_check check (cost > 0),
    constraint type_check check (lower(type) = 'orientation' or lower(type) = 
'location')
    );
```

Person

Table *'person'* based on entity *'Person'*.

```
create table drilldb.person
    (
    initials character varying(4) not null primary key,
    first_name character varying(15) not null,
    last_name character varying(15) not null,
    department character varying(10) not null,

    constraint person_unique unique(first_name, last_name, department),

    constraint department_check check (lower(department) = 'geology'
        or lower(department) = 'survey' or lower(department) = 'drill')
    );
```

Drillhole

Table *'drillhole'* based on entity *'Drillhole'*.

Since the values for dip, azimuth, drill_area, and section_line are derived from records existing in tables *collar_survey* and *downhole_survey*, is it not possible for these values to be 'not null' in the table *'drillhole'*, since this would prevent drillhole records from being entered before their corresponding downhole and collar surveys were loaded. Loading downhole and collar survey records before drillhole records is not possible, since it violates the foreign key requirement on these tables. Instead, these values are set using trigger functions upon loading of downhole and collar survey records.

```
create table drilldb.drillhole
    (
    drillhole_id character varying(9) not null primary key,
    drilling_status character varying(11) default 'planned',
    length integer not null,
    type character varying(20) not null,
    dip float, --not null,
    azimuth float, --not null,
    start_date date,
    completion_date date,
    geologist character varying(4),
    drill_area character varying(10), --not null,
    section_line integer, --not null,

    foreign key (drill_area) references drilldb.drill_area (id),
    foreign key (section_line) references drilldb.section_line (id),
    foreign key (type) references drilldb.drillhole_type (id),
    foreign key (geologist) references drilldb.person (initials),

    constraint drillhole_length_check check (length >= 0),
    constraint drillhole_dip_check check (dip >= -90 and dip <= 90),
    constraint drillhole_azimuth_check check(azimuth >= 0 and azimuth < 360),
    constraint drillhole_status_check check(lower(drilling_status) = 'planned'
        or lower(drilling_status) = 'in progress'
        or lower(drilling_status) = 'complete'
        or lower(drilling_status) = 'cancelled'
        or lower(drilling_status) = 'collapsed'),

    constraint drillhole_start_date_check check (start_date = NULL or start_date
        >= current_date),
    constraint drillhole_completion_date_check check (completion_date = NULL or
        completion_date >= current_date),

    constraint drillhole_status_check check
        (
          ((lower(drilling_status) = 'planned' or lower(drilling_status) =
            'cancelled') and start_date is null and geologist is null)
          or (lower(drilling_status) = 'in progress')
          or ((lower(drilling_status) = 'complete' or lower(drilling_status) =
             'collapsed') and (start_date is not null and completion_date is not
             null and geologist is not null))
        )

    );
```

Collar Survey

Table *'collar_survey'* based on entity *'Collar Survey'*.

```
create table drilldb.collar_survey
    (
    id serial not null primary key,
    drillhole_id character varying(9) not null,
    survey_status character varying(8) not null,
    survey_tool character varying(10),
    surveyor character varying(4),
    date_surveyed date,

    foreign key (drillhole_id) references drilldb.drillhole (drillhole_id),
    foreign key (surveyor) references drilldb.person (initials),
    foreign key (survey_tool) references drilldb.survey_tool (name),

    constraint collar_date_check check (date_surveyed <= current_date),
    constraint collar_status_check check (lower(survey_status) = 'planned' or
        lower(survey_status) = 'surveyed'),

    constraint collar_check check
        (
          lower(survey_status) = 'planned'
          or (lower(survey_status) <> 'planned'
          and survey_tool is not null
          and surveyor is not null
          and date_surveyed is not null)
        )
    );

select AddGeometryColumn('drilldb','collar_survey','geometry',4326,'POINT',3,1);

alter table drilldb.collar_survey add constraint collar_survey_unique
    unique(drillhole_id, geometry, survey_status);
```

Downhole Survey

Table *'downhole_survey'* based on entity *'Downhole Survey'*.

```
create table drilldb.downhole_survey
    (
    id serial not null primary key,
    drillhole_id character varying(9) not null,
    depth integer not null,
    dip float not null,
    azimuth float not null,
    survey_status character varying(8) not null,
    survey_tool character varying(10),
    driller character varying(4),
    date_surveyed date,

    foreign key (drillhole_id) references drilldb.drillhole (drillhole_id),
    foreign key (driller) references drilldb.person (initials),
    foreign key (survey_tool) references drilldb.survey_tool (name),

    constraint dhsurvey_unique unique(drillhole_id, depth, survey_status),

    constraint dhsurvey_depth_check check (depth >= 0),
    constraint dhsurvey_dip_check check (dip >= -90 and dip <= 90),
    constraint dhsurvey_status_check check (lower(survey_status) = 'planned' or
          lower(survey_status) = 'surveyed'),
    constraint dhsurvey_azimuth_check check (azimuth >= 0 and azimuth < 360),
    constraint dhsurvey_date_check check (date_surveyed <= current_date),

    constraint dhsurvey_check check
        (
          lower(survey_status) = 'planned'
          or (lower(survey_status) <> 'planned'
          and survey_tool is not null
          and driller is not null
          and date_surveyed is not null)
        )

    );
```

4.3. Indexes

The following spatial and non-spatial were created to reduce query speed, since is expected that this database will contain a large number of records.

```
create index downhole_survey_idx on drilldb.downhole_survey(drillhole_id);
create index collar_survey_idx on drilldb.collar_survey(drillhole_id);

create index tenement_gidx on drilldb.tenement using GIST(geometry);
create index drill_area_gidx on drilldb.drill_area using GIST(geometry);
create index section_line_gidx on drilldb.section_line using GIST(geometry);
create index collar_survey_gidx on drilldb.collar_survey using GIST(geometry);
create index building_gidx on drilldb.building using GIST(geometry);
```

4.4. Triggers

Triggers have been utilised primarily to calculate derived information before or after a record is updated, or a new record inserted into a table. Many of the derived values are compulsory in their respective tables, and so it is necessary to utilise a trigger function to calculate these values before the record is inserted, else the insert will fail due to missing compulsory values.

Trigger functions are also used to update information in tables other than those into which the data is being inserted/updated. For example, the drilling status of the drillhole is dynamically updated as records are inserted into the collar survey and downhole survey tables, according to the requirements listed in the system specification. The drillhole dip, azimuth, drill area, and section line are also updated in this way.

Set *tenement validity* in table *tenement* before record is inserted or updated:

```
create trigger tenement_validity_set
    before insert or update on drilldb.tenement
    for each row
    execute procedure set_validity();

create or replace function set_validity() returns trigger as
$BODY$
     begin
        new.license_validity = (new.license_valid_to -
            new.license_valid_from)/365.0::float;
        return new;
     end;

$BODY$ LANGUAGE 'plpgsql';
```

Set azimuth and length of section line before insertion or update of record in table *section_line*:

```
create trigger section_line_update
    before insert or update on drilldb.section_line
    for each row
    execute procedure update_section_line();

create or replace function update_section_line() returns trigger as
$BODY$
     begin
        new.azimuth = st_azimuth(st_startpoint(new.geometry),
            st_endpoint(new.geometry))/((2*pi())*360);
        new.length = st_length(st_transform(new.geometry,3107));
        return new;
     end;

$BODY$ LANGUAGE 'plpgsql';
```

Set drill area in table *drillhole* upon insertion or update of record into table *collar_survey*:

```
create trigger drillhole_update_area
    after insert on drilldb.collar_survey
    for each row
    execute procedure set_drill_area();

create or replace function set_drill_area() returns trigger as
$BODY$
     begin
        update drilldb.drillhole a set drill_area = s.drill_area
        from (select d.id as drill_area, c.drillhole_id
            from drilldb.drill_area d, drilldb.collar_survey c
            where st_contains(d.geometry, c.geometry)) as s
        where s.drillhole_id = a.drillhole_id;
        return null;
     end;

$BODY$ LANGUAGE 'plpgsql';
```

Update azimuth and dip in table *drillhole* upon insertion of record into table *downhole_survey*:

```
create trigger drillhole_update_azidip
    after insert on drilldb.downhole_survey
    for each row
    execute procedure set_dip();

create or replace function set_dip() returns trigger as
$BODY$
    begin
    if new.survey_status = 'planned' then
        update drilldb.drillhole a set (dip, azimuth) = (new.dip, new.azimuth)
        where a.drillhole_id = new.drillhole_id and a.dip is null
            and a.azimuth is null;
    end if;
    if new.survey_status = 'surveyed' then
        update drilldb.drillhole a set (dip, azimuth) = (s.dip, s.azimuth)
            from (select drillhole_id, depth, dip, azimuth
                  from drilldb.downhole_survey b
                  where survey_status = 'surveyed'
                  group by drillhole_id, depth, dip, azimuth
                  having depth = (select min(depth)
                                    from drilldb.downhole_survey
                                    where survey_status = 'surveyed'
                                    and drillhole_id = b.drillhole_id)) as s
                  where a.drillhole_id = s.drillhole_id;
    end if;
    return null;
    end;

$BODY$ LANGUAGE 'plpgsql';
```

Set section line in table *drillhole* upon insertion or update of record in table *collar_survey*:

```
create trigger drillhole_set_section_line
    after insert on drilldb.collar_survey
    for each row
    execute procedure set_section_line();

create or replace function set_section_line() returns trigger as
$BODY$
    begin
    if new.survey_status = 'planned' then
        update drilldb.drillhole a set section_line = s.section_line
        from (select distinct on (c.drillhole_id) c.drillhole_id
            as drillhole_id, l.id as section_line,
            st_distance(l.geometry, c.geometry) as distance
            from drilldb.section_line l, drilldb.collar_survey c
            order by c.drillhole_id, st_distance(l.geometry, c.geometry)) as s
        where s.drillhole_id = a.drillhole_id;
        end if;
        return null;
     end;

$BODY$ LANGUAGE 'plpgsql';
```

Update drilling status in table *drillhole* before record is inserted or updated, based upon records present in tables *collar_survey* and *downhole_survey*. This trigger is activated every time a *collar_survey* or *downhole_survey* record is inserted or updated, since both tables have triggers that prompt updates on the *drillhole* table, therefore prompting this trigger to run.

```
create trigger drillhole_update_status
    before insert or update on drilldb.drillhole
    for each row
    execute procedure update_status();

create or replace function update_status() returns trigger as
$BODY$
    begin
    if new.start_date is not null and new.drilling_status = 'planned' then
        new.drilling_status = 'in progress';
    end if;

    if new.start_date is not null and new.completion_date is not null
```

```
        and new.drillhole_id not in (select drillhole_id
            from drilldb.collar_survey where survey_status = 'surveyed')
        and new.drillhole_id not in (select drillhole_id
            from drilldb.downhole_survey where survey_status = 'surveyed') then
        new.drilling_status = 'in progress';
    end if;

    if new.start_date is not null and new.completion_date is not null
        and new.drillhole_id in (select drillhole_id
            from drilldb.collar_survey where survey_status = 'surveyed')
            and new.drillhole_id in (select drillhole_id
                  from drilldb.downhole_survey where survey_status = 'surveyed')
        then
        new.drilling_status = 'complete';
    end if;

    if new.start_date is not null and new.completion_date is not null
        and new.drilling_status = 'collapsed' and new.drillhole_id not in
            (select drillhole_id
            from drilldb.collar_survey where survey_status = 'surveyed')
        then
        new.drilling_status = 'in progress';
    end if;
    return new;
    end;

$BODY$ LANGUAGE 'plpgsql';
```

4.5. Table Population

```
-- Add data to Person table

-- Six geologists
insert into drilldb.person values ('LAB', 'Lucille', 'Ablett', 'Geology');
insert into drilldb.person values ('LAL', 'Lucy', 'Aldridge', 'Geology');
insert into drilldb.person values ('WS', 'William', 'Shaw', 'Geology');
insert into drilldb.person values ('KG', 'Katie', 'Grahame', 'Geology');
insert into drilldb.person values ('JW', 'John', 'Williams', 'Geology');
insert into drilldb.person values ('FM', 'Fiona', 'Murray', 'Geology');

-- Three surveyors
insert into drilldb.person values ('MF', 'Michael', 'Fisher', 'Survey');
insert into drilldb.person values ('JPR', 'Jonathan', 'Price', 'Survey');
insert into drilldb.person values ('JPL', 'Josephine', 'Plummer', 'Survey');

-- Nine drillers
insert into drilldb.person values ('LM', 'Lauren', 'Munroe', 'Drill');
insert into drilldb.person values ('CW', 'Craig', 'Wood', 'Drill');
insert into drilldb.person values ('DC', 'Daniel', 'Cleaver', 'Drill');
insert into drilldb.person values ('NS', 'Nathan', 'Stone', 'Drill');
insert into drilldb.person values ('AAT', 'Andrew', 'Atwood', 'Drill');
insert into drilldb.person values ('YC', 'Yolandi', 'Cooper', 'Drill');
insert into drilldb.person values ('AAD', 'Ana', 'Adams', 'Drill');
insert into drilldb.person values ('IREY', 'Isabel', 'Reynolds', 'Drill');
insert into drilldb.person values ('IREI', 'Isaac', 'Reid', 'Drill');

-- Add data to Survey_Tool table
insert into drilldb.survey_tool values ('EMS', 'orientation', 5.69);
insert into drilldb.survey_tool values ('EZ-Shot', 'orientation', 4.38);

insert into drilldb.survey_tool values ('Flexit', 'location', 9.67);
insert into drilldb.survey_tool values ('DGPS', 'location', 2.96);
insert into drilldb.survey_tool values ('GPS', 'location', 1.00);


-- Add data to Drillhole_Type table
insert into drilldb.drillhole_type values ('RVC', 'Reverse circulation', 38.42);
insert into drilldb.drillhole_type values ('DIA', 'Diamond drillcore', 49.54);


-- Add data to Tenement table
insert into drilldb.tenement (tenement_id, geometry, license_valid_from,
license_valid_to) values ('EL5815', st_geomfromtext('
POLYGON((136.13469771033214784 -32.96522300743509959,136.00136578500405449 -
32.96522445714225569,136.00136669511800847 -
33.02635413573403866,136.00130089262324873 -
33.02629315180678304,136.00131399664473975 -
33.03189149359678822,135.90136786623509124 -
33.03189256199141255,135.90136862076633406 -
33.08189284427857046,135.83470266889412414 -
33.08189355744099203,135.82591630059789622 -
33.08189364827251211,135.82590857362288261 -
33.07794794294244412,135.82580176294209195 -
33.07795175876589155,135.82580175484815754 -
33.077402439370303,135.82589330673181394 -
33.07740243847098327,135.82586245728771246 -
33.0553915098095672,135.77309813297358687 -
33.05570488217381353,135.7739983579389218 -
33.05364874149546495,135.7724114232496504 -
33.05114248833439206,135.76835257781033306 -
33.04777032024333039,135.76800159849608463 -
33.0456989315596914,135.7796590946925334 7 -
33.03950751745719572,135.77947997941862468 -
33.01522710194842603,135.77947561680741728 -
33.01463555218822421,135.7794754009701137 -
33.01434182101746728,135.77946011429389728 -
32.9985870315654779,135.7794137029810599 4 -
32.9563580941499481,135.7513683467728924 4 -
32.9564499796820769,135.7210036830746275 9 -
32.95654945459290275,135.72112618962307806 -
32.9856328712592699,135.7202411820859424 6 -
32.985133152172466,135.717540394774459 86 -
32.98449612279705434,135.71604504974038719 -
32.98451902763025601,135.71255081576862267 -
32.98395448630975579,135.70969746296862013 -
32.9846297224829641,135.7079732332791763 8 -
32.98431693377870033,135.69764314721453502 -
32.98456881589737577,135.68793873298727704 -
32.98935639188323421,135.68047739611608904 -
32.99808454886209574,135.68047739611608904 -
32.99811506645642112,135.68038127028046347 -
32.99856139369262564,135.68037059352911911 -
32.99861098141093407,135.68014173135694023 -
32.99978973180992625,135.67866166490500746 -
33.00114015829092295,135.67643389670809029 -
33.00052982708911031,135.67400775065823382 -
32.99893530032312583,135.67307697032697433 -
32.99863013247386334,135.67303119393557154 -
32.99861487457604881,135.67287018561171408 -
32.99856147373225213,135.67069661235984768 -
32.99784051152960274,135.66803652476687603 -
32.99808438788346621,135.66694299142707791 -
32.99811139902107016,135.66448633317884287 -
32.99680679119967408,135.66354028505725182 -
32.99592560128473906,135.66323508842970114 -
32.99435394068103733,135.66238059848694775 -
32.99386566546735367,135.66109887481434271 -
32.99392290012099238,135.65993920443179377 -
32.99295397414584841,135.65511748748315313 -
32.9935567510424903,135.65311859944495154 -
32.99306467349515515,135.65316436774241993 -
32.99251153917544599,135.65312901269487611 -
32.94856145494645716,135.6013698219608159 -
32.94856220767900368,135.56803608160100794 -
32.9485626897156294,135.53470387098798255 -
32.94856316365832072,135.46803184779264484 -
32.94856412683225244,135.46803218593777274 -
32.93189660194587987,135.46803264009531631 -
32.89856307472535946,135.46803358977945209 -
32.84856280233071857,135.46803448820219273 -
32.79856196875908836,135.46803516449233484 -
32.76522898293046637,135.33470322747302816 -
32.76523040745661319,135.31803673410911415 -
32.7652305873210139,135.31803572776766487 -
32.69856355896030209,135.36803521865147104 -
32.69856300497792745,135.38470170931748271 -
32.69856282601284647,135.38470195753041025 -
32.71522958917347523,135.40136845089443796 -
32.71522941110765714,135.40136869371133344 -
32.73189617786556482,135.41803518617598456 -
32.73189599979980358,135.41803544248273283 -
32.74856275306785847,135.55136736781082618 -
32.7485613438302039,135.58470034914284952 -
32.74856098410140248,135.58470087884359145 -
32.78189447175174109,135.63470032116401853 -
32.78189396633274555,135.66803332587835484 -
32.7818935886174927,135.66803306957160657 -
32.76522683534943781,135.68470055848513312 -
32.76522667706871772,135.6847005809681832 -
32.76643186933455354,135.71291378214641554 -
32.76634001527872897,135.71299030186207801 -
32.78189267580557953,135.70136323152905788 -
32.78189274685200871,135.70136347344669048 -
32.7985592609003902,135.75136599234781443 -
32.79855872940106565,135.75136623876198882 -
32.81522625698539741,135.75136648877355583 -
32.83189301744806698,135.75136698789731327 -
32.86522653837329244,135.75136748792033359 -
32.89856158364943894,135.78470046385643855 -
32.89856121582670312,135.86803494245975799 -
32.89856030121615049,135.90135977953127622 -
32.89855986234704233,135.90135853397021037 -
32.81522466248742376,135.90135803574582951 -
32.78189078902795472,135.91803067237867708 -
32.78189064513640005,135.91803571397815631 -
32.78189064513640005,135.96803009491407011 -
32.78189017389166793,136.05136558075798803 -
32.7818893950787924,136.10136809516257017 -
32.78188851554182293,136.10136883800259966 -
32.83188930414729612,136.10136908711479009 -
32.84855453216516707,136.16802893348960879 -
32.84855400786040036,136.16802943351262911 -
32.88188886337957229,136.16803018174857698 -
32.9318891411701884,136.16803068446961333 -
32.96522265580017574,136.13469771033214784 -32.96522300743509959))
',4326), '2017-01-01', '2021-12-31');
```

```
insert into drilldb.tenement (tenement_id, geometry, license_valid_from, 
license_valid_to) values ('EL5120', st_geomfromtext('
POLYGON((136.53469909898535661 -33.3485542505774788,136.53470012691047941 -
33.41522127354221539,136.53470139225657931 -
33.49855507495601614,136.50136843340760606 -
33.4985554328861781,136.48470195083541512 -
33.4985556127505788,136.4680354664646984 -
33.49855579171566689,136.46803520656067121 -
33.48188902945440759,136.41803576334086756 -
33.48188956634965052,136.41803525432464994 -
33.44855604272646588,136.401368771752459 -
33.44855622169154685,136.40136826363550426 -
33.41522270256490401,136.40136801002677203 -
33.39855593760569263,136.36803503948658545 -
33.39855629553585459,136.36803529399469426 -
33.41522305869642651,136.31803584358044645 -
33.41522359379308682,136.31803609898793184 -
33.43189035515501928,136.35136906503146292 -
33.43188999722485732,136.35136957404768054 -
33.46522352084804197,136.31803660800414946 -
33.46522387877820393,136.31803711702048076 -
33.49855740420009198,136.30137063354902693 -
33.49855758316516585,136.2847041500775731 -
33.49855776213024683,136.26803766840475873 -
33.49855794199464754,136.25137119932253427 -
33.49855812095978536,136.21803819011142878 -
33.49855849237974326,136.21803386886904264 -
33.21522355462258957,136.21803366921960787 -
33.2018370092698091‌9,136.23331863304952094 -
33.20184824809740576,136.23434299323560026 -
33.20184715542114873,136.23456174163220567 -
33.20184871574485896,136.23470017607337468 -
33.20184876610693436,136.25136666134346797 -
33.20183693462610108,136.25283002198901272 -
33.2018365739979231,136.25283640717555045 -
33.19855639755894572,136.25338908823698603 -
33.19855639126370761,136.33469904272772055 -
33.19855552251857489,136.46802379056657628 -
33.198554100690444‌1,136.50136385226107905 -
33.1985537562500781,136.50136611945197274 -
33.34855460940696048,136.53469909898535661 -33.3485542505774788))
',4326), '2016-06-01', '2019-12-31');

insert into drilldb.tenement (tenement_id, geometry, license_valid_from, 
license_valid_to) values ('EL5659', st_geomfromtext('
POLYGON((136.13467648813048072 -31.49854863171321639,136.13467695038207239 -
31.53188213015545216,136.21800940730940965 -
31.53188127669881169,136.23467589078086348 -
31.53188110582760828,136.23467658865479279 -
31.58188135573919908,136.23467670646596162 -
31.5901858250150589,136.23467752754697813 -
31.64854835622088203,136.23340035084663668 -
31.64854836791209891,136.21801104767280322 -
31.64854852619277281,136.05134613561676815 -
31.64855024209925105,135.88468122266135651 -
31.64855195530770615,135.88468239717599317 -
31.73188571625206933,135.82994863941928543 -
31.73188581607678671,135.82992197811790902 -
31.73188627832831799,135.82778100798827836 -
31.73123199096283997,135.8234990083736875 -
31.73400199099830843,135.81893500834507904 -
31.73239799077390444,135.81893372411320797 -
31.73239117481210769,135.81871764040784001 -
```

```
31.73123884019958041,135.81827100830150812 -
31.7288569911564764,135.81319100785981391 -
31.72984499085287879,135.80901300944390186 -
31.73006099092123122,135.79954400894996525 -
31.72754999113839958,135.79956041258412824 -
31.72754572745258983,135.80381000872603181 -
31.72643999121185843,135.80380180511031085 -
31.72643567986199642,135.79853000895388959 -
31.72365899066915063,135.79404500897115327 -
31.72629399077431245,135.789487009219215 -
31.72664699086516293,135.78405700882501606 -
31.72602299086758748,135.77977400916427086 -
31.72382899131412159,135.77432500879319832 -
31.71967599135314231,135.77434070556023471 -
31.71967690056777656,135.7788650086143889 -
31.71993699079871831,135.78344500938078454 -
31.7173379914543716,135.78324300905876498 -
31.7132979913084796,135.78775000825737607 -
31.71084299141017482,135.78775077987563691 -
31.71083651089549349,135.7881837135087153 -
31.70722196202319765,135.7911160078984949 -
31.70332499095480117,135.79609100799473254 -
31.70387999136773161,135.79608851057741958 -
31.70387346858495903,135.79483300843764937 -
31.69989099087576179,135.79503017040713075 -
31.69901981940836677,135.79566700822692837 -
31.69620599091297564,135.79767107276290972 -
31.6924754843553842,135.79774101933367092 -
31.69154707463701115,135.79795500851787438 -
31.68866699128784603,135.79794749648090146 -
31.68866397586106487,135.79306400779887554 -
31.68670299064973506,135.78839200818572408 -
31.68819399105910861,135.78839034264129282 -
31.68818746377968765,135.78749893642782354 -
31.68468076070780626,135.78867100846139238 -
31.68114699074027385,135.79054900783160065 -
31.67753499144595608,135.7921450076870542 -
31.67305899097891597,135.7921437423409543 -
31.67305198436082847,135.79140476132181448 -
31.6689408604316327,135.78748100844291002 -
31.66405799138243893,135.7824440081855073 -
31.66082199144097231,135.77671500659334924 -
31.66074872637188165,135.77670874551324687 -
31.66074855819863387,135.77237700799366849 -
31.66044999137324112,135.7723761680268808 -
31.66044321767958536,135.77208400797167087 -
31.65686099134092402,135.77208575805241253 -
31.65685353865910656,135.77282200783054122 -
31.65295999130978544,135.77314200819822076 -
31.64837399116601446,135.77528535253804876 -
31.6447975951321645,135.77997300844867823 -
31.64790899130616708,135.78412300827130821 -
31.64607899145152814,135.78903900834461638 -
31.64749299100799362,135.79269600791553785 -
31.64348199130012063,135.79270040470100867 -
31.64347342975423416,135.79517400797521987 -
31.6396959911763247,135.79871900507907867 -
31.63608299453380823,135.80295900815283971 -
31.63369199129158815,135.80660700551004538 -
31.63005199430336134,135.81162300839616819 -
31.62844099105814166,135.81497600833722572 -
31.62482799081834983,135.81829300841673103 -
31.6207739909261818,135.823003007983516 -
```

```
31.61680199121814994,135.82798700849480156 -
31.61538899080852616,135.83223000810949088 -
31.61230299058979654,135.83222849275182398 -
31.61229970716499338,135.83036073665971344 -
31.60822765967799342,135.82584800788038137 -
31.60710199086344119,135.8215030080584711 -
31.60689599125623772,135.81823700853226455 -
31.60231599138916181,135.8136460081578889 -
31.60047699111948205,135.81377307606771865 -
31.59927110097976666,135.81403600815633581 -
31.59677599131816805,135.81403107807284414 -
31.59676748373163946,135.81171200800383758 -
31.59253199075800822,135.80674400823045289 -
31.59463899079605653,135.80307800824459719 -
31.59854299096548402,135.79886200807629848 -
31.59613299134372255,135.7940620079564269 -
31.5944119911197312,135.79406653514354275 -
31.59440777419865753,135.79836939723099931 -
31.59039772237622401,135.7983675572180573 -
31.59039692018097867,135.79422300818634994 -
31.58857499084240317,135.78914029269139974 -
31.59189912614300155,135.78900739537607478 -
31.59220007527249763,135.78739100788982341 -
31.59586999090657855,135.78468153672463359 -
31.59640171326771352,135.78302400795269023 -
31.59672699085718151,135.77876800773844934 -
31.59456099079631031,135.77557800811962352 -
31.59102499140948339,135.77409194678705262 -
31.58640436287027242,135.77408296885505479 -
31.58637648568549139,135.77385759695164325 -
31.5856771287956839,135.7710414320175687 -
31.58155270389340785,135.77103841569146425 -
31.58154795637233292,135.76883500833002927 -
31.57726199074750184,135.76605300774122043 -
31.57342699106203199,135.76402600779510976 -
31.56913499098629927,135.76413100814056634 -
31.56515599095547842,135.76413462431457901 -
31.56515595498257198,135.76882000853765931 -
31.56510099111710588,135.77320300831331679 -
31.56502899139405827,135.77319862591696165 -
31.56502178692517191,135.77106200850596451 -
31.56098199092537371,135.76762700838082765 -
31.55714999137825316,135.7629900076838112 -
31.55696199080250253,135.76107900769113712 -
31.55266099121104162,135.76108429390615129 -
31.55265583000181095,135.76511400850574773 -
31.54872099118006545,135.7651049865070263 -
31.54871817630203168,135.76457241518596675 -
31.5485515148398008,135.76089900838360336 -
31.5474019906086447,135.76089993108803355 -
31.54739423755324879,135.7614171923506774 -
31.54368201511215375,135.75948000769733426 -
31.53989499066051394,135.7547350083150377 -
31.5419709910676147,135.75011000817141849 -
31.54460499112667904,135.74817364549846843 -
31.5482718229855017,135.74816900769465633 -
31.54827999062831978,135.74753409262541481 -
31.54855160657064417,135.74393800773395924 -
31.55008999135924341,135.73851900784700319 -
31.55235999082015397,135.7338650081646847 -
31.55263899109581871,135.72770400823446835 -
31.55161799077689011,135.72770888076138363 -
31.55161175667649687,135.73010464771243733 -
```

```
31.54855170010011989,135.7307190078770418 -
31.54776699125289952,135.73498200841402195 -
31.54461199144946804,135.73497855771543641 -
31.54460278149241859,135.73309900791400651 -
31.53957999142272683,135.72890000852964931 -
31.5374559906007299,135.72414400774084697 -
31.53952199144600854,135.72414154539706033 -
31.53951338583334874,135.72286900829922729 -
31.53539699097757776,135.72230378709116394 -
31.53305100959568463,135.72196600802590183 -
31.53164899080752548,135.72173700825737797 -
31.5278249907299859,135.72173059429258046 -
31.52781746160582088,135.71850500850041499 -
31.52399999060634173,135.71801455962531691 -
31.52398280546134401,135.71428432286430166 -
31.52385214386061207,135.71042394071344006 -
31.52750095960084309,135.70963056420032444 -
31.53267563167429444,135.70862355913106967 -
31.5372285061898805,135.706472160462908 5 -
31.54210373639756426,135.70441231187976427 -
31.54685117294252805,135.70282748969793829 -
31.54855230534382926,135.70071977919553774 -
31.55081470103686669,135.69711880019360706 -
31.55493844785030433,135.69484533924207881 -
31.56022948320333299,135.69363996801124017 -
31.56459353156549597,135.69362473799230884 -
31.56522016477805792,135.61801616511502289 -
31.5652209471882123 6,135.6180159312913247 -
31.54855419481947365,135.61801523071937936 -
31.49855393861264474,135.61801499689568118 -
31.48188718264663066,135.61801464616007706 -
31.45667845814358543,135.61801444830928176 -
31.44254513195153677,135.61801434488722862 -
31.43514539324456791,135.61801406609731657 -
31.4152201785676084,135.61801360474521516 -
31.38188667383019492,135.61801353639668832 -
31.37716632756320578,135.61801336912276383 -
31.36521992955533733,135.61801267574549001 -
31.31521967154987252,135.61801152011662452 -
31.23188591869945085,135.65134450594530335 -
31.23188557875567639,135.66801133745434527 -
31.23188540878385311,135.70084255554058927 -
31.2318850733366844,135.83467593294903963 -
31.23188370726654739,135.88467541304100905 -
31.23188319735089991,136.00134085759555091 -
31.23188200844720086,136.00134017231221151 -
31.18188176393152844,136.03467315364423484 -
31.18188142398776108,136.13467210303625166 -
31.18188040955249463,136.13467302574065343 -
31.24854739924234082,136.13467348709286853 -
31.28188089588587317,136.13467417867150289 -
31.33188141649344516,136.13467648813048072 -31.49854863171321639))
',4326), '2017-08-01', '2018-12-31');


-- Add data to Building table
insert into drilldb.building (occupancy_status, geometry) values ('unoccupied',
st_geomfromtext('
POLYHEDRALSURFACE(  ((135.810836451 -32.9301153355 135,
                        135.813109762 -32.9301153355 135,
                        135.813109762 -32.9315807757 135,
                        135.810836451 -32.9315807757 135,
                        135.810836451 -32.9301153355 135)),
                    ((135.810836451 -32.9301153355 135,
                        135.813109762 -32.9301153355 135,
                        135.813109762 -32.9301153355 140,
                        135.810836451 -32.9301153355 140,
                        135.810836451 -32.9301153355 135)),
                    ((135.810836451 -32.9301153355 135,
                        135.810836451 -32.9315807757 135,
                        135.810836451 -32.9315807757 140,
                        135.810836451 -32.9301153355 140,
                        135.810836451 -32.9301153355 135)),
                    ((135.813109762 -32.9315807757 135,
                        135.813109762 -32.9301153355 135,
                        135.813109762 -32.9301153355 140,
                        135.813109762 -32.9315807757 140,
                        135.813109762 -32.9315807757 135)),
                    ((135.813109762 -32.9315807757 135,
                        135.810836451 -32.9315807757 135,
                        135.810836451 -32.9315807757 140,
                        135.813109762 -32.9315807757 140,
                        135.813109762 -32.9315807757 135)),
                    ((135.810836451 -32.9301153355 140,
                        135.813109762 -32.9301153355 140,
                        135.813109762 -32.9315807757 140,
                        135.810836451 -32.9315807757 140,
                        135.810836451 -32.9301153355 140))
)',4283));


insert into drilldb.building (occupancy_status, geometry) values ('unoccupied',
st_geomfromtext('
POLYHEDRALSURFACE(  ((136.089833717 -31.1984910926 135,
                        136.092107028 -31.1984910926 135,
                        136.092107028 -31.1999565328 135,
                        136.089833717 -31.1999565328 135,
                        136.089833717 -31.1984910926 135)),
                    ((136.089833717 -31.1984910926 135,
                        136.092107028 -31.1984910926 135,
                        136.092107028 -31.1984910926 140,
                        136.089833717 -31.1984910926 140,
                        136.089833717 -31.1984910926 135)),
                    ((136.089833717 -31.1984910926 135,
                        136.089833717 -31.1999565328 135,
                        136.089833717 -31.1999565328 140,
                        136.089833717 -31.1984910926 140,
                        136.089833717 -31.1984910926 135)),
                    ((136.092107028 -31.1999565328 135,
                        136.092107028 -31.1984910926 135,
                        136.092107028 -31.1984910926 140,
                        136.092107028 -31.1999565328 140,
                        136.092107028 -31.1999565328 135)),
                    ((136.092107028 -31.1999565328 135,
                        136.089833717 -31.1999565328 135,
                        136.089833717 -31.1999565328 140,
                        136.092107028 -31.1999565328 140,
                        136.092107028 -31.1999565328 135)),
                    ((136.089833717 -31.1984910926 140,
                        136.092107028 -31.1984910926 140,
                        136.092107028 -31.1999565328 140,
                        136.089833717 -31.1999565328 140,
                        136.089833717 -31.1984910926 140))
)',4283));
```

```
insert into drilldb.building (occupancy_status, geometry) values ('unoccupied',
st_geomfromtext('
POLYHEDRALSURFACE(  ((136.141537452 -31.5832630796 135,
                        136.143810763 -31.5832630796 135,
                        136.143810763 -31.5847285198 135,
                        136.141537452 -31.5847285198 135,
                        136.141537452 -31.5832630796 135)),
                    ((136.141537452 -31.5832630796 135,
                        136.143810763 -31.5832630796 135,
                        136.143810763 -31.5832630796 140,
                        136.141537452 -31.5832630796 140,
                        136.141537452 -31.5832630796 135)),
                    ((136.141537452 -31.5832630796 135,
                        136.141537452 -31.5847285198 135,
                        136.141537452 -31.5847285198 140,
                        136.141537452 -31.5832630796 140,
                        136.141537452 -31.5832630796 135)),
                    ((136.143810763 -31.5847285198 135,
                        136.143810763 -31.5832630796 135,
                        136.143810763 -31.5832630796 140,
                        136.143810763 -31.5847285198 140,
                        136.143810763 -31.5847285198 135)),
                    ((136.143810763 -31.5847285198 135,
                        136.141537452 -31.5847285198 135,
                        136.141537452 -31.5847285198 140,
                        136.143810763 -31.5847285198 140,
                        136.143810763 -31.5847285198 135)),
                    ((136.141537452 -31.5832630796 140,
                        136.143810763 -31.5832630796 140,
                        136.143810763 -31.5847285198 140,
                        136.141537452 -31.5847285198 140,
                        136.141537452 -31.5832630796 140))
)',4283));

insert into drilldb.building (occupancy_status, geometry) values ('occupied',
st_geomfromtext('
POLYHEDRALSURFACE(  ((136.427166774 -33.4246450348 300,
                        136.429440085 -33.4246450348 300,
                        136.429440085 -33.4261104750 300,
                        136.427166774 -33.4261104750 300,
                        136.427166774 -33.4246450348 300)),
                    ((136.427166774 -33.4246450348 300,
                        136.429440085 -33.4246450348 300,
                        136.429440085 -33.4246450348 305,
                        136.427166774 -33.4246450348 305,
                        136.427166774 -33.4246450348 300)),
                    ((136.427166774 -33.4246450348 300,
                        136.427166774 -33.4261104750 300,
                        136.427166774 -33.4261104750 305,
                        136.427166774 -33.4246450348 305,
                        136.427166774 -33.4246450348 300)),
                    ((136.429440085 -33.4261104750 300,
                        136.429440085 -33.4246450348 300,
                        136.429440085 -33.4246450348 305,
                        136.429440085 -33.4261104750 305,
                        136.429440085 -33.4261104750 300)),
                    ((136.429440085 -33.4261104750 300,
                        136.427166774 -33.4261104750 300,
                        136.427166774 -33.4261104750 305,
                        136.429440085 -33.4261104750 305,
                        136.429440085 -33.4261104750 300)),
                    ((136.427166774 -33.4246450348 305,
                        136.429440085 -33.4246450348 305,
                         136.429440085 -33.4261104750 305,
                         136.427166774 -33.4261104750 305,
                         136.427166774 -33.4246450348 305))
)',4283));


-- Add data to Drill_Area table
insert into drilldb.drill_area (id, geometry) values
('EXAA0001',st_geomfromtext('POLYGON((135.66769542903773527 -
32.78465837710358244,135.66769542903773527 -
32.94578164664756059,135.47110099194486565 -
32.94578164664756059,135.47110099194486565 -
32.78465837710358244,135.66769542903773527 -32.78465837710358244))',4326));
insert into drilldb.drill_area (id, geometry) values
('EXAA0002',st_geomfromtext('POLYGON((135.66747324321150359 -
32.89880639159374454,135.90128158968576599 -
32.89871971874576673,135.90156465236833583 -
32.96682386468501846,135.77951978775107023 -
32.96599720611924766,135.77959493852978312 -
32.95622760488758018,135.66769234818536916 -
32.95621891994604624,135.66747324321150359 -32.89880639159374454))',4326));
insert into drilldb.drill_area (id, geometry) values
('EXBA0001',st_geomfromtext('POLYGON((135.62200375558487053 -
31.33214412628925061,136.133630257009969 -
31.33214412628925061,136.1342314632395869 -
31.4992794581296387,135.62320616804416318 -
31.50168428304820978,135.62200375558487053 -31.33214412628925061))',4326));
insert into drilldb.drill_area (id, geometry) values
('EXCA0001',st_geomfromtext('POLYGON((136.53422148289811844 -
33.34904072517969098,136.53407118134069265 -
33.49813987013082794,136.46869000386715243 -
33.49798956857342347,136.46838940075235769 -
33.48115579414345433,136.41848928369208238 -
33.48115579414345433,136.41833898213465659 -
33.44778884839834632,136.40165550926212745 -
33.44778884839833921,136.40225671549177378 -
33.38225736936739452,136.4719966381302072 -
33.3816561631377553,136.4719966381302072 -
33.34874012206486071,136.53422148289811844 -33.34904072517969098))',4326));


-- Add data to Section_Line table
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.48030696233627168 -32.7348334108219845,
135.4815093747955359 -32.95487489087090438)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.53982637907083131 -32.73423220459233818,
135.54102879153009553 -32.95427368464125806)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.60235182695356571 -32.73783944197018769,
135.60355423941282993 -32.95788092201910757)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.66607968729562117 -31.18672736949461566,
135.67209174959199913 -33.01920395744257775)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.7358196099340546 -31.18913219441319384,
135.73702202239331882 -33.02160878236114883)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.81217280109856915 -31.18432254457605879,
135.81337521355783338 -33.02641843219828388)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(135.89032961095202268 -31.19454305047996812,
135.8915320234112869 -33.0270196384279231)',4326));
```

```
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(136.42600436156294563 -33.1911489391201826,
136.42720677402220986 -33.53143166509745043)',4326));
insert into drilldb.section_line (geometry) values
(st_geomfromtext('LINESTRING(136.50175634649778544 -33.19355376403875368,
136.50175634649775702 -33.54345578969029162)',4326));


-- Add data to Drillhole table
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC001', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC002', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC003', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC004', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC005', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC006', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC007', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC008', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC009', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXARVC010', 'planned', 51, 'RVC');

insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC001', 'planned', 69, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC002', 'planned', 69, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC003', 'planned', 66, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC004', 'planned', 60, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC005', 'planned', 54, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC006', 'planned', 51, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC007', 'planned', 48, 'RVC');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXBRVC008', 'planned', 48, 'RVC');

insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXCDIA001', 'planned', 99, 'DIA');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXCDIA002', 'planned', 102, 'DIA');
insert into drilldb.drillhole (drillhole_id, drilling_status, length, type)
      values ('EXCDIA003', 'planned', 105, 'DIA');

-- Update some holes with start and completion dates and geologists
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
02-01', '2017-02-05', 'LAB') where drillhole_id = 'EXARVC001';
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
03-02', '2017-03-05', 'LAL') where drillhole_id = 'EXBRVC001';
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
03-02', '2017-03-06', 'LAL') where drillhole_id = 'EXBRVC002';
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
03-14', '2017-03-21', 'KG') where drillhole_id = 'EXBRVC003';
```

```
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
04-01', '2017-04-07', 'WS') where drillhole_id = 'EXCDIA001';
update drilldb.drillhole set (start_date, completion_date, geologist) = ('2017-
04-05', '2017-04-10', 'WS') where drillhole_id = 'EXCDIA002';

-- Update some holes with start dates and geologists
update drilldb.drillhole set (start_date, geologist) = ('2017-02-05', 'FM')
where drillhole_id = 'EXARVC002';
update drilldb.drillhole set (start_date, geologist) = ('2017-03-05', 'FM')
where drillhole_id = 'EXARVC003';
update drilldb.drillhole set (start_date, geologist) = ('2017-04-03', 'JW')
where drillhole_id = 'EXARVC004';
update drilldb.drillhole set (start_date, geologist) = ('2017-03-20', 'JW')
where drillhole_id = 'EXBRVC004';
update drilldb.drillhole set (start_date, geologist) = ('2017-03-27', 'WS')
where drillhole_id = 'EXBRVC005';
update drilldb.drillhole set (start_date, geologist) = ('2017-04-12', 'LAB')
where drillhole_id = 'EXCDIA003';


-- Cancel some holes
update drilldb.drillhole set drilling_status = 'cancelled' where drillhole_id =
'EXBRVC006';
update drilldb.drillhole set drilling_status = 'cancelled' where drillhole_id =
'EXBRVC006';
update drilldb.drillhole set drilling_status = 'cancelled' where drillhole_id =
'EXARVC005';

-- Add data to Collar_Survey table
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC001', 'planned', st_geomfromtext('POINT(135.48412375995042112 -
32.82052361486449854 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC002', 'planned', st_geomfromtext('POINT(135.4799153163429537 -
32.90649610570326189 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC003', 'planned', st_geomfromtext('POINT(135.53943473307748491 -
32.81992240863485222 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC004', 'planned', st_geomfromtext('POINT(135.54123835176639545 -
32.90529369324397635 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC005', 'planned', st_geomfromtext('POINT(135.60316259341948353 -
32.82112482109413776 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC006', 'planned', st_geomfromtext('POINT(135.60316259341948353 -
32.90769851816254032 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC007', 'planned', st_geomfromtext('POINT(135.67290251605791696 -
32.92934194242964452 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC008', 'planned', st_geomfromtext('POINT(135.7366303763999 44 -
32.92813952997035898 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC009', 'planned', st_geomfromtext('POINT(135.81358477379407645 -
32.93054435488893006 135)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXARVC010', 'planned', st_geomfromtext('POINT(135.89234278987711946 -
32.92994314865929084 135)',4326));

insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC001', 'planned', st_geomfromtext('POINT(135.66749165999110005 -
31.3758250450355014 160)',4326));
```

```
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC002', 'planned', st_geomfromtext('POINT(135.66749165999110005 -
31.4527794424296452 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC003', 'planned', st_geomfromtext('POINT(135.73723158262953348 -
31.3758250450355014 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC004', 'planned', st_geomfromtext('POINT(135.73602917017026925 -
31.45157702997035898 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC005', 'planned', st_geomfromtext('POINT(135.81178115510510906 -
31.37342022011694098 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC006', 'planned', st_geomfromtext('POINT(135.81178115510510906 -
31.45157702997035898 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC007', 'planned', st_geomfromtext('POINT(135.89114037741782681 -
31.37462263257622297 160)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXBRVC008', 'planned', st_geomfromtext('POINT(135.89114037741782681 -
31.44796979259250946 160)',4326));

insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXCDIA001', 'planned', st_geomfromtext('POINT(136.42861874671748978 -
33.42413466942474543 300)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXCDIA002', 'planned', st_geomfromtext('POINT(136.50376952542271169 -
33.39347315171301744 300)',4326));
insert into drilldb.collar_survey (drillhole_id, survey_status, geometry) values
('EXCDIA003', 'planned', st_geomfromtext('POINT(136.5013647005041264 -
33.46381428058109719 300)',4326));

-- Update some collar surveys to 'surveyed'
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPR', '2017-02-04',
st_geomfromtext('POINT(135.48412375995042112 -32.82052361486449854 135)',4326))
where drillhole_id = 'EXARVC001';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPR', '2017-02-07',
st_geomfromtext('POINT(135.4799153163429537 -32.90649610570326189 135)',4326))
where drillhole_id = 'EXARVC002';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'MF', '2017-02-09',
st_geomfromtext('POINT(135.53943473307748491 -32.81992240863485222 135)',4326))
where drillhole_id = 'EXARVC003';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'MF', '2017-02-15',
st_geomfromtext('POINT(135.54123835176639545 -32.90529369324397635 135)',4326))
where drillhole_id = 'EXARVC004';

update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPR', '2017-01-21',
st_geomfromtext('POINT(135.66749165999110005 -31.3758250450355014 160)',4326))
where drillhole_id = 'EXBRVC001';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPR', '2017-01-24',
st_geomfromtext('POINT(135.66749165999110005 -31.4527794424296452 160)',4326))
where drillhole_id = 'EXBRVC002';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPR', '2017-02-04',
st_geomfromtext('POINT(135.73602917017026925 -31.45157702997035898 160)',4326))
where drillhole_id = 'EXBRVC004';
```

```
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPL', '2017-02-10',
st_geomfromtext('POINT(136.42861874671748978 -33.42413466942474543 300)',4326))
where drillhole_id = 'EXCDIA001';
update drilldb.collar_survey set (survey_status, survey_tool, surveyor,
date_surveyed, geometry) = ('surveyed', 'DGPS', 'JPL', '2017-02-10',
st_geomfromtext('POINT(136.50376952542271169 -33.39347315171301744 300)',4326))
where drillhole_id = 'EXCDIA002';


-- Add data to Downhole_Survey table

-- ‘Planned’ surveys
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC001', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC002', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC003', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC004', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC005', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC006', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC007', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC008', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC009', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXARVC010', 0, 70, 180, 'planned');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC001', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC002', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC003', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC004', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC005', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC006', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC007', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXBRVC008', 0, 70, 180, 'planned');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXCDIA001', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXCDIA002', 0, 70, 180, 'planned');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status) values ('EXCDIA003', 0, 70, 180, 'planned');

-- ‘Surveyed’ surveys
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC001', 5,
69.5, 181.5, 'surveyed', 'EMS', 'AAT', '2017-02-03');
```

```
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC001', 25,
70.5, 184.5, 'surveyed', 'EMS', 'LM', '2017-02-03');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC001', 50,
71.6, 187.0, 'surveyed', 'EMS', 'CW', '2017-02-04');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 5,
68.5, 181.0, 'surveyed', 'EMS', 'AAT', '2017-02-03');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 25,
75.6, 184.4, 'surveyed', 'EMS', 'LM', '2017-02-03');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 50,
79.7, 184.6, 'surveyed', 'EMS', 'LM', '2017-02-04');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 75,
80.4, 185.7, 'surveyed', 'EMS', 'CW', '2017-02-04');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 100,
80.2, 186.3, 'surveyed', 'EMS', 'CW', '2017-02-04');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXARVC002', 125,
81.1, 187.2, 'surveyed', 'EMS', 'CW', '2017-02-04');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC001', 5,
65.2, 180.1, 'surveyed', 'Flexit', 'IREI', '2017-01-19');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC001', 25,
70.3, 185.2, 'surveyed', 'Flexit', 'IREI', '2017-01-19');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC001', 50,
75.4, 190.6, 'surveyed', 'Flexit', 'IREI', '2017-01-21');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC003', 5,
65.1, 181.2, 'surveyed', 'Flexit', 'DC', '2017-01-31');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC003', 25,
66.3, 182.4, 'surveyed', 'Flexit', 'DC', '2017-01-31');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXBRVC003', 50,
67.5, 184.6, 'surveyed', 'Flexit', 'DC', '2017-02-01');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA001', 5,
72.2, 183.1, 'surveyed', 'EZ-Shot', 'IREI', '2017-02-01');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA001', 25,
71.4, 185.0, 'surveyed', 'EZ-Shot', 'IREI', '2017-02-01');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA001', 50,
70.6, 186.3, 'surveyed', 'EZ-Shot', 'YC', '2017-02-02');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA001', 75,
70.6, 186.3, 'surveyed', 'EZ-Shot', 'YC', '2017-02-02');

insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA002', 5,
72.2, 183.1, 'surveyed', 'EZ-Shot', 'AAD', '2017-02-03');
```

```
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA002', 25,
71.3, 185.4, 'surveyed', 'EZ-Shot', 'AAD', '2017-02-03');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA002', 50,
70.4, 186.9, 'surveyed', 'EZ-Shot', 'AAD', '2017-02-04');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA002', 75,
70.3, 186.9, 'surveyed', 'EZ-Shot', 'NS', '2017-02-04');
insert into drilldb.downhole_survey (drillhole_id, depth, dip, azimuth,
survey_status, survey_tool, driller, date_surveyed) values ('EXCDIA002', 100,
70.4, 186.9, 'surveyed', 'EZ-Shot', 'NS', '2017-02-05');
```

5. Map of Spatial Data

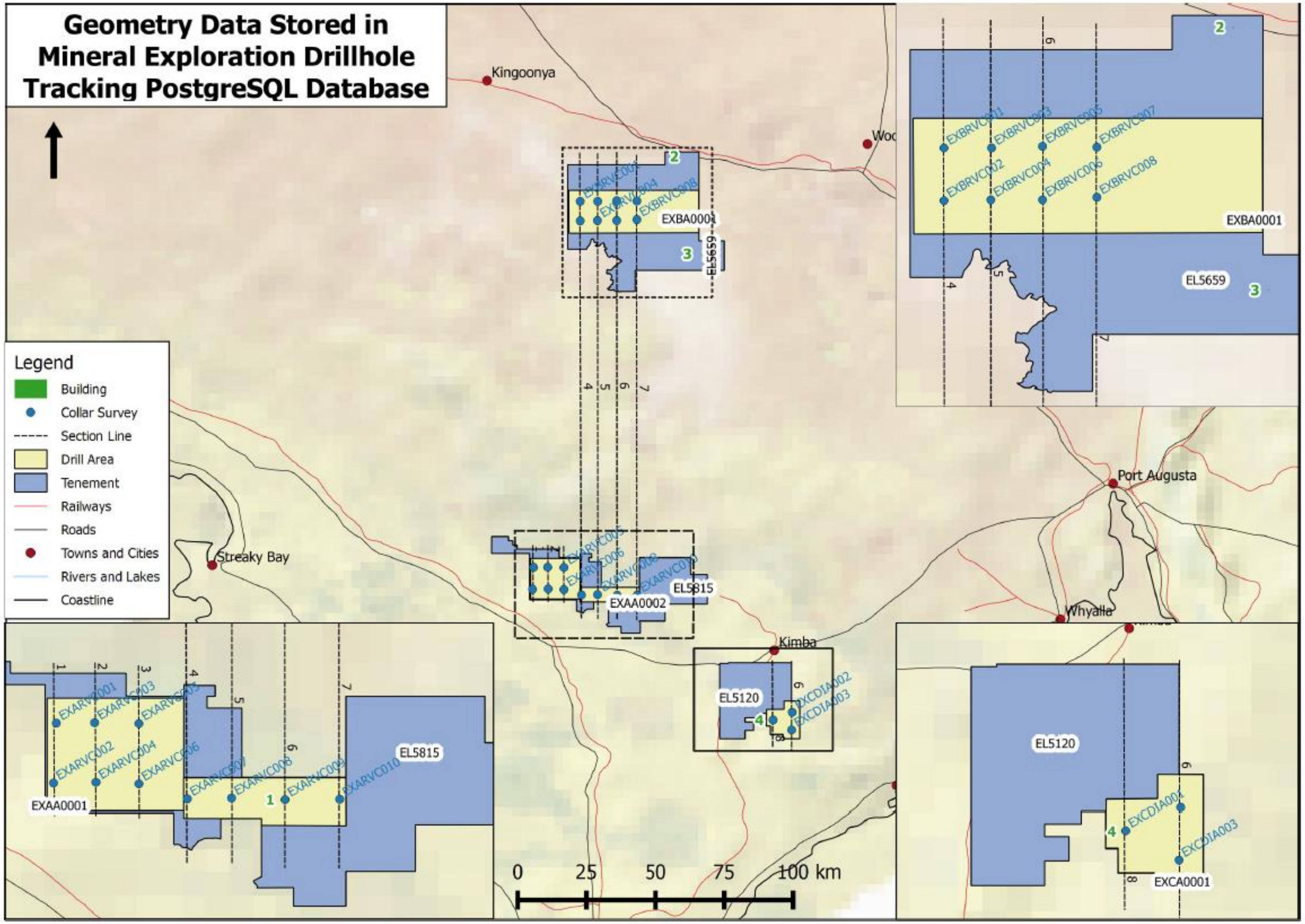


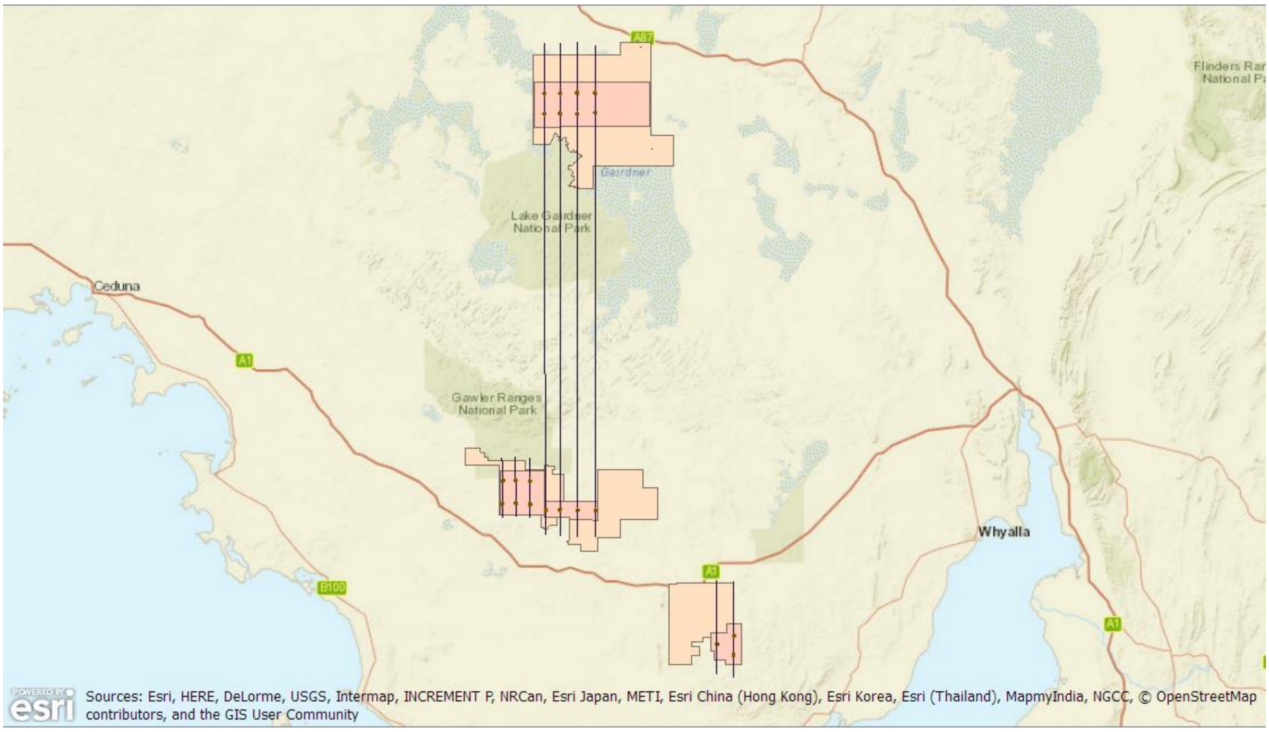
Figure 3: Spatial data mapped using QGIS, with areas of interest mapped at higher zoom levels. Some labels do not seem to show at certain zoom levels.

6. 3D Screenshot from FME with appropriate background mapping

As background mapping is not available in 3D mode, Figure 4 shows the 3D data viewed in 2D mode, along with the 2D data, with appropriate background mapping. Figure 5 shows the 3D data viewed in 3D mode.

When viewing the 3D data in FME, a problem was encountered in that neither WGS84 or GDA94 are currently intended for capturing height information. The vertical height components of both the collar surveys and building have assumed to be in meters, however these coordinate systems measure vertical height in degrees, which is not ideal when dealing with 3D datasets. The four drill areas EXAA0001, EXAA0002, EXBA0001, and EXCA0001, and their contained drillholes and buildings, are located at 135 m, 135m, 160 m, and 300 m above sea level respectively, however this is incorrectly rendered in FME, since it is not able to determine that these values are actually in meters, rather than degrees. The buildings are therefore rendered as vastly taller than their true heights of 5 m, and the collar surveys are difficult to view simultaneously at a suitable zoom level, since they are interpreted to be many kilometres apart, rather than a maximum of 140 m (Figure 5). The small size of the geometry objects compared to the geographic area in which they are located further complicates the viewing of all objects at the same time.

It should be noted that Figure 5 is only showing one building, although four are present in the database. This is likely a side-effect of the incorrect height measurements, and the subsequent difficulty in rendering the building objects, since all four buildings can be seen in 2D view (Figure 2), and the number of buildings displaying in 3D view varied between FME Data Inspector sessions.



Sources: Esri, HERE, DeLorme, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), MapmyIndia, NGCC, © OpenStreetMap contributors, and the GIS User Community

Figure 4: Spatial data viewed in FME, in 2D mode, with ESRI World Street Map background mapping.

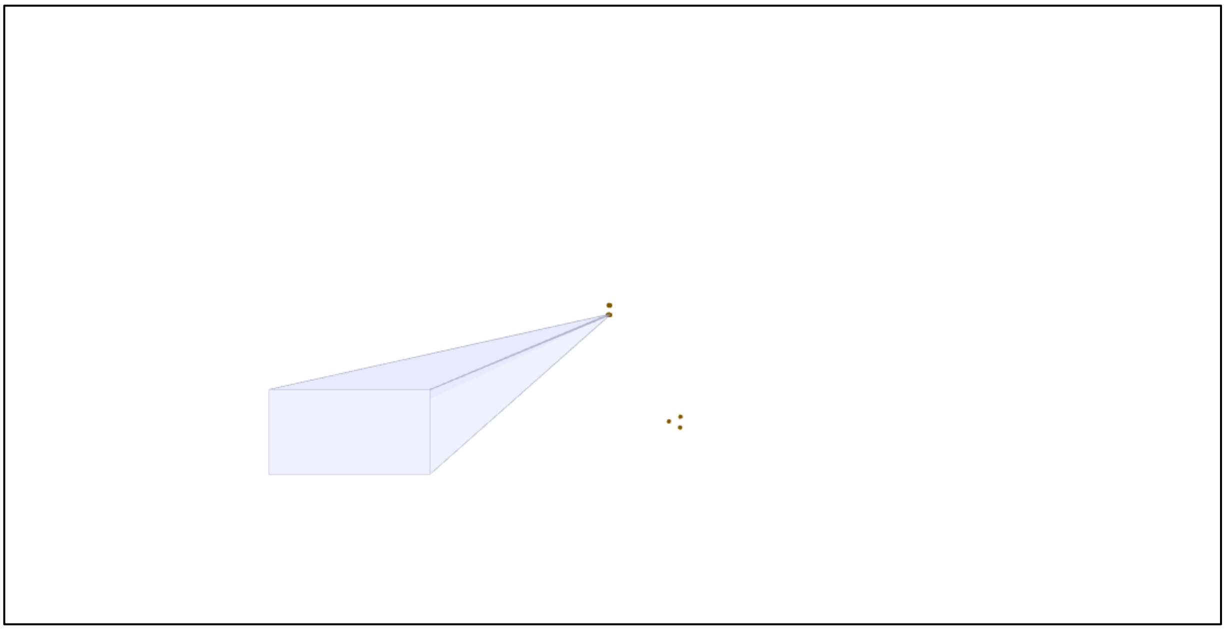


Figure 5: Screenshot of 3D data viewed in FME: the point data are collar surveys, and the cuboid is one of the four buildings that rendered properly during this particular FME Data Inspector session. The 3 points in a triangle to the bottom-right of the screen-shot are the collar position of drillholes in drill area EXCA0001. Due to the inability WGS85 and GDA94 to recognise height in meters, the data does not render correctly.

7. Functional Requirements

All functional requirements were successfully answered through querying the database created using the above scripts (Table 2). It should be noted that the spatial aspect of functional requirement 8 was calculated using triggers, and so is not required in the query itself.

Table 2: Functional requirements successfully answered.

	Functional Requirement	**Successfully Answered?**
1	Which tenement has the largest area?	Yes
2	How many drillholes have been planned, are in progress, or have been completed in each drill area, and in which tenement are they located?	Yes
3	What percentage of all holes have been cancelled?	Yes
4	What is the average number of holes drilled per tenement, per month?	Yes
5	Do any planned drillholes fall within 100 m of the occupied buildings, or 50 m of the unoccupied buildings? Which geologist is responsible?	Yes
6	Do any of the drill areas overlap each other, or extend outside of the tenement area?	Yes
7	Do all drillholes fall within a drill area?	Yes

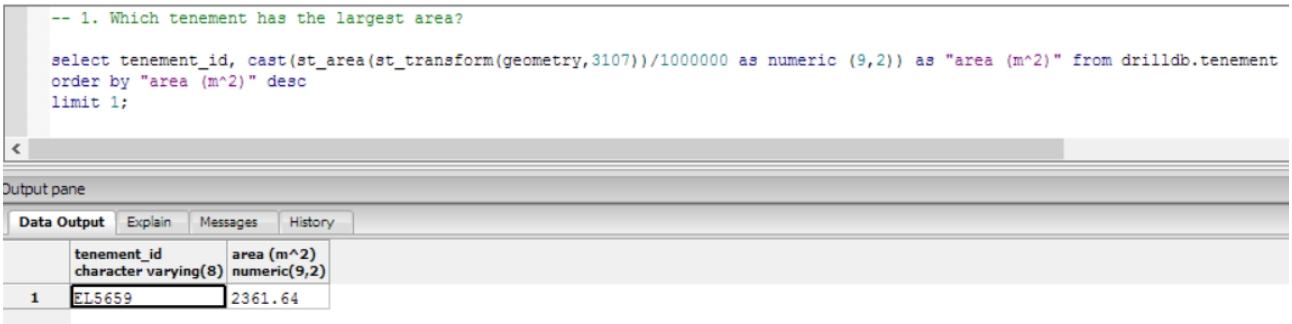
8	What is the longest section line and how many drillholes does it have on it (of any drilling status)?	Yes
9	What is the largest distance between adjacent drillholes on each section line, as measured from their collar positions?	Yes
10	What is the total drilling expense per tenement, per year, taking into account both hole type and downhole survey expenses?	Yes
11	Which drillholes fall more than 10 metres from their intended section lines? Are these inaccuracies the fault of one particular geologist?	Yes
12	How many drillholes in each tenement have been drilled (i.e. have a completion date), but are missing GPS or DGPS collar surveys?	Yes
13	What is the maximum distance a geologist had to travel between drillholes for which he/she was responsible for at one time?	Yes

8. SQL Query Scripts

The functional requirements were answered using the following scripts, giving the following answers.

1. Which tenement has the largest area?

```
select tenement_id,
      cast(st_area(st_transform(geometry,3107))/1000000  as  numeric  (9,2))  as
      "area (m^2)"
    from drilldb.tenement
order by "area (m^2)" desc
limit 1;
```

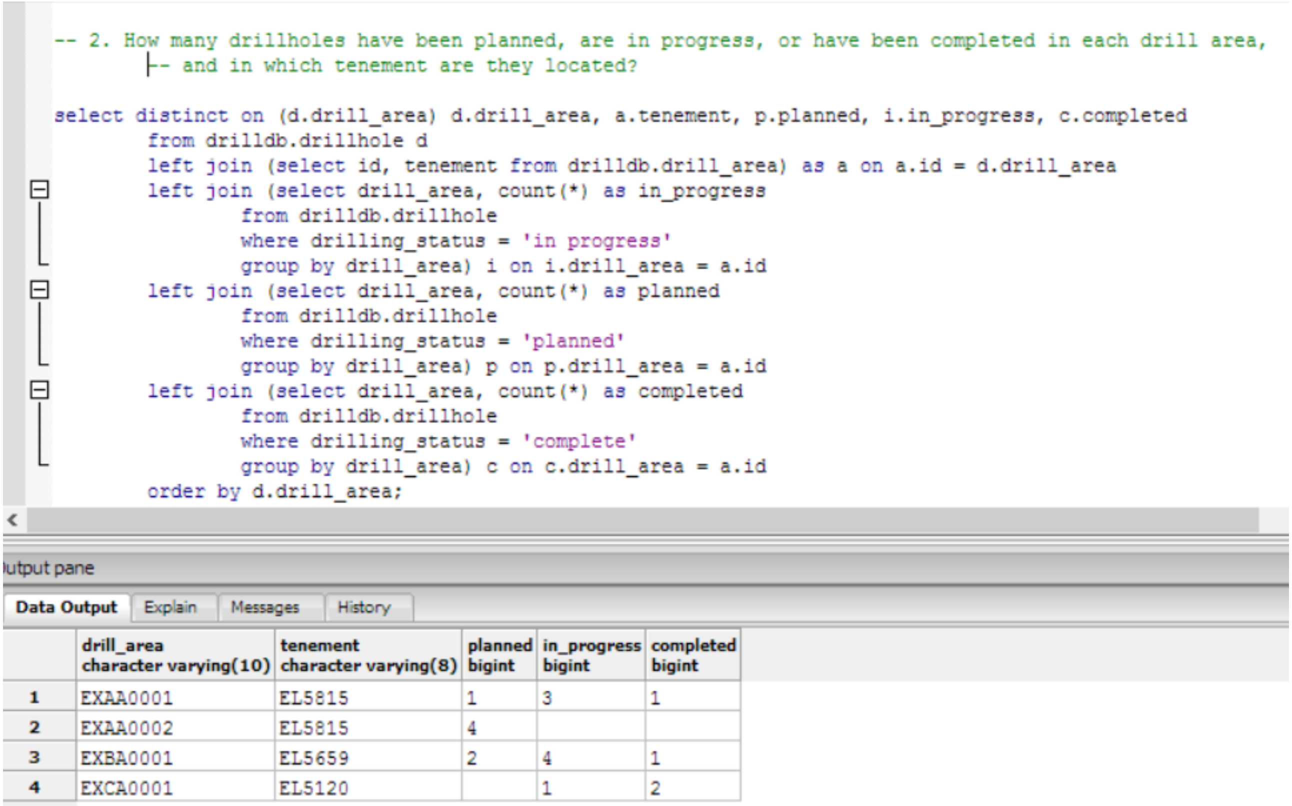


```
-- 1. Which tenement has the largest area?

select tenement_id, cast(st_area(st_transform(geometry,3107))/1000000 as numeric (9,2)) as "area (m^2)" from drilldb.tenement
order by "area (m^2)" desc
limit 1;
```

```
     tenement_id          area (m^2)
     character varying(8) numeric(9,2)
1    EL5659               2361.64
```

2. How many drillholes have been planned, are in progress, or have been completed in each drill area, and in which tenement are they located?

```
select distinct on (d.drill_area) d.drill_area, a.tenement, p.planned,
      i.in_progress, c.completed
    from drilldb.drillhole d
    left join (select id, tenement from drilldb.drill_area) as a
        on a.id = d.drill_area
    left join (select drill_area, count(*) as in_progress
        from drilldb.drillhole
        where drilling_status = 'in progress'
        group by drill_area) i on i.drill_area = a.id
    left join (select drill_area, count(*) as planned
        from drilldb.drillhole
        where drilling_status = 'planned'
        group by drill_area) p on p.drill_area = a.id
    left join (select drill_area, count(*) as completed
        from drilldb.drillhole
        where drilling_status = 'complete'
        group by drill_area) c on c.drill_area = a.id
order by d.drill_area;
```

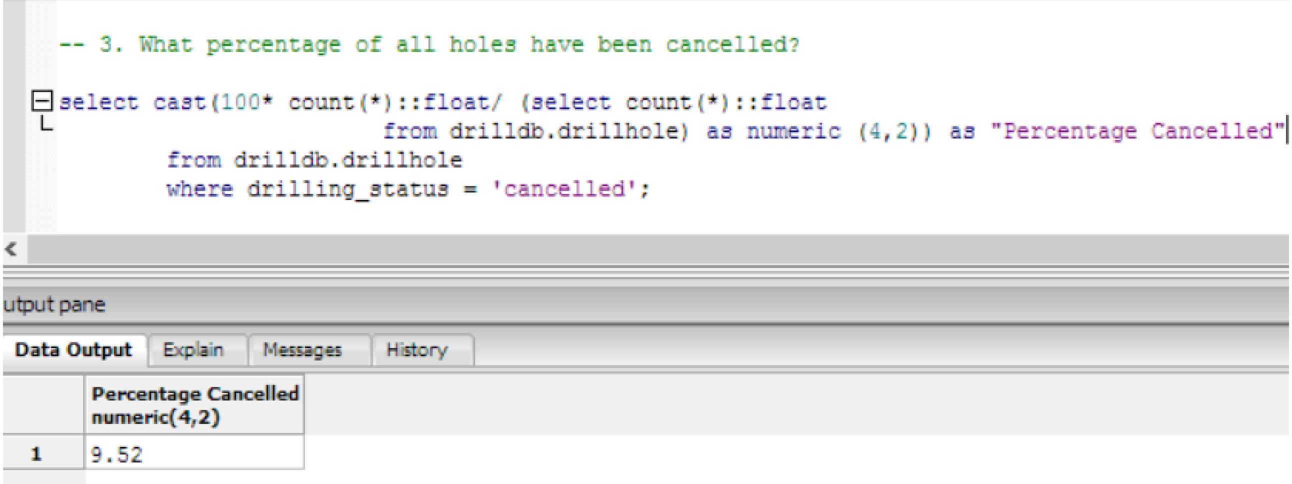


```
-- 2. How many drillholes have been planned, are in progress, or have been completed in each drill area,
      -- and in which tenement are they located?

select distinct on (d.drill_area) d.drill_area, a.tenement, p.planned, i.in_progress, c.completed
      from drilldb.drillhole d
      left join (select id, tenement from drilldb.drill_area) as a on a.id = d.drill_area
      left join (select drill_area, count(*) as in_progress
            from drilldb.drillhole
            where drilling_status = 'in progress'
            group by drill_area) i on i.drill_area = a.id
      left join (select drill_area, count(*) as planned
            from drilldb.drillhole
            where drilling_status = 'planned'
            group by drill_area) p on p.drill_area = a.id
      left join (select drill_area, count(*) as completed
            from drilldb.drillhole
            where drilling_status = 'complete'
            group by drill_area) c on c.drill_area = a.id
      order by d.drill_area;
```

	drill_area character varying(10)	tenement character varying(8)	planned bigint	in_progress bigint	completed bigint
1	EXAA0001	EL5815	1	3	1
2	EXAA0002	EL5815	4		
3	EXBA0001	EL5659	2	4	1
4	EXCA0001	EL5120		1	2

3. What percentage of all holes have been cancelled?

```
select cast(100* count(*)::float/(select count(*)::float
            from drilldb.drillhole) as numeric (4,2)) as "Percentage Cancelled"
    from drilldb.drillhole
    where drilling_status = 'cancelled';
```

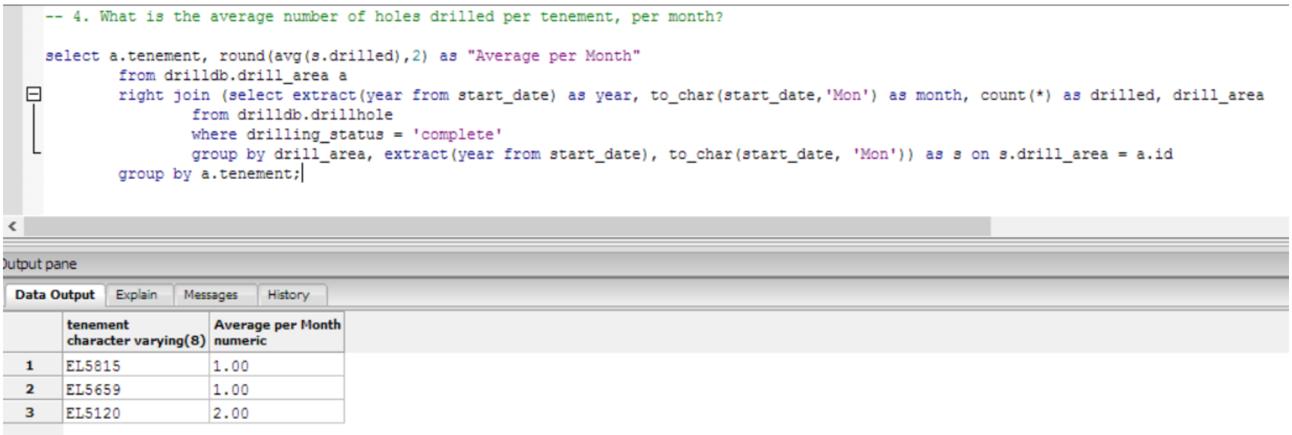


```
-- 3. What percentage of all holes have been cancelled?

select cast(100* count(*)::float/ (select count(*)::float
                                   from drilldb.drillhole) as numeric (4,2)) as "Percentage Cancelled"
         from drilldb.drillhole
         where drilling_status = 'cancelled';
```

```
   Percentage Cancelled
   numeric(4,2)
1  9.52
```

4. What is the average number of holes drilled per tenement, per month?

```
select a.tenement, round(avg(s.drilled),2) as "Average per Month"
    from drilldb.drill_area a
    right join (select extract(year from start_date) as year,
        to_char(start_date,'Mon') as month, count(*) as drilled, drill_area
        from drilldb.drillhole
        where drilling_status = 'complete'
        group by drill_area, extract(year from start_date),
        to_char(start_date, 'Mon')) as s on s.drill_area = a.id
group by a.tenement;
```

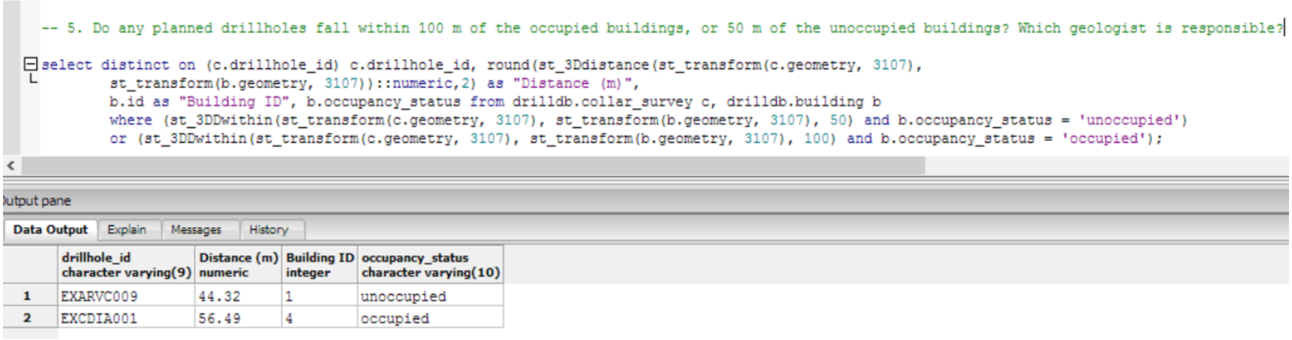


```
-- 4. What is the average number of holes drilled per tenement, per month?

select a.tenement, round(avg(s.drilled),2) as "Average per Month"
     from drilldb.drill_area a
     right join (select extract(year from start_date) as year, to_char(start_date,'Mon') as month, count(*) as drilled, drill_area
          from drilldb.drillhole
          where drilling_status = 'complete'
          group by drill_area, extract(year from start_date), to_char(start_date, 'Mon')) as s on s.drill_area = a.id
     group by a.tenement;
```

```
   tenement              Average per Month
   character varying(8)  numeric
1  EL5815                1.00
2  EL5659                1.00
3  EL5120                2.00
```

5. Do any planned drillholes fall within 100 m of the occupied buildings, or 50 m of the unoccupied buildings? Which geologist is responsible?

```
select distinct on (c.drillhole_id) c.drillhole_id,
    round(st_3Ddistance(st_transform(c.geometry, 3107),
    st_transform(b.geometry, 3107))::numeric,2) as "Distance (m)",
    b.id as "Building ID", b.occupancy_status
    from drilldb.collar_survey c, drilldb.building b
    where (st_3DDwithin(st_transform(c.geometry, 3107),
        st_transform(b.geometry, 3107), 50)
        and b.occupancy_status = 'unoccupied')
        or (st_3DDwithin(st_transform(c.geometry, 3107),
         st_transform(b.geometry, 3107), 100)
        and b.occupancy_status = 'occupied');
```

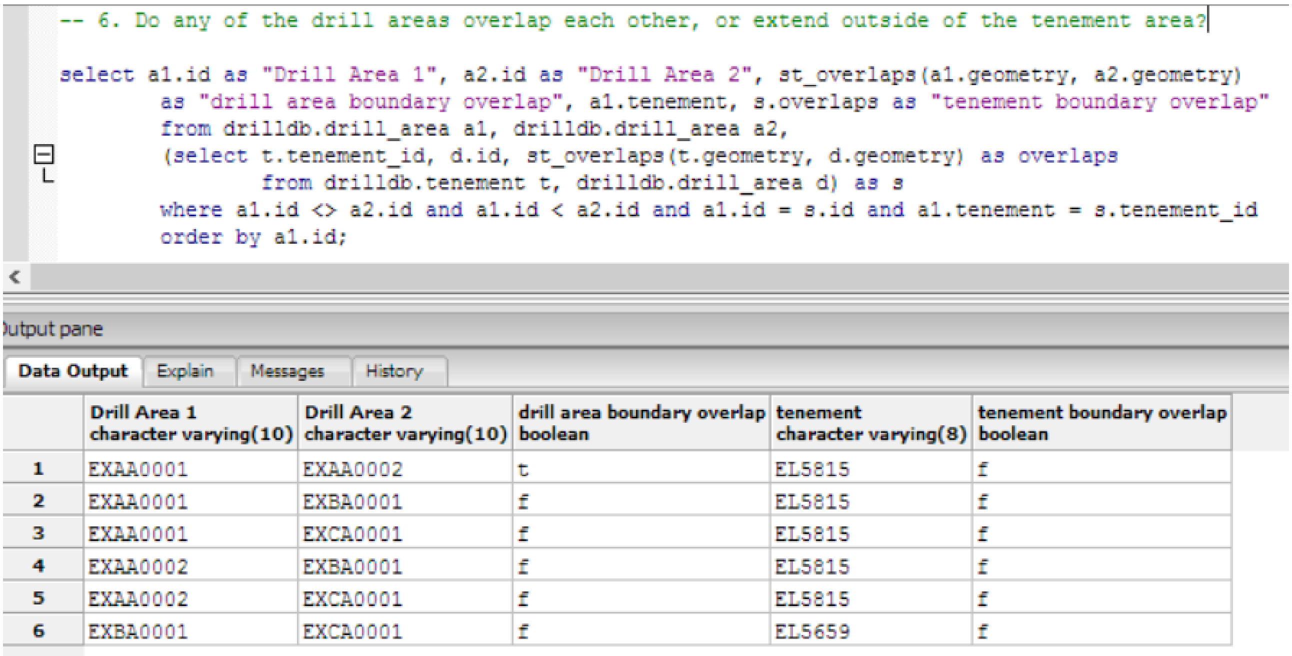


```
-- 5. Do any planned drillholes fall within 100 m of the occupied buildings, or 50 m of the unoccupied buildings? Which geologist is responsible?

select distinct on (c.drillhole_id) c.drillhole_id, round(st_3Ddistance(st_transform(c.geometry, 3107),
        st_transform(b.geometry, 3107))::numeric,2) as "Distance (m)",
        b.id as "Building ID", b.occupancy_status from drilldb.collar_survey c, drilldb.building b
        where (st_3DDwithin(st_transform(c.geometry, 3107), st_transform(b.geometry, 3107), 50) and b.occupancy_status = 'unoccupied')
        or (st_3DDwithin(st_transform(c.geometry, 3107), st_transform(b.geometry, 3107), 100) and b.occupancy_status = 'occupied');
```

	drillhole_id character varying(9)	Distance (m) numeric	Building ID integer	occupancy_status character varying(10)
1	EXARVC009	44.32	1	unoccupied
2	EXCDIA001	56.49	4	occupied

6. Do any of the drill areas overlap each other, or extend outside of the tenement area?

```
select a1.id as "Drill Area 1", a2.id as "Drill Area 2",
    st_overlaps(a1.geometry, a2.geometry) as "drill area boundary overlap",
    a1.tenement, s.overlaps as "tenement boundary overlap"
    from drilldb.drill_area a1, drilldb.drill_area a2,
    (select t.tenement_id, d.id, st_overlaps(t.geometry, d.geometry) as overlaps
        from drilldb.tenement t, drilldb.drill_area d) as s
    where a1.id <> a2.id and a1.id < a2.id and a1.id = s.id and a1.tenement =
        s.tenement_id
order by a1.id;
```



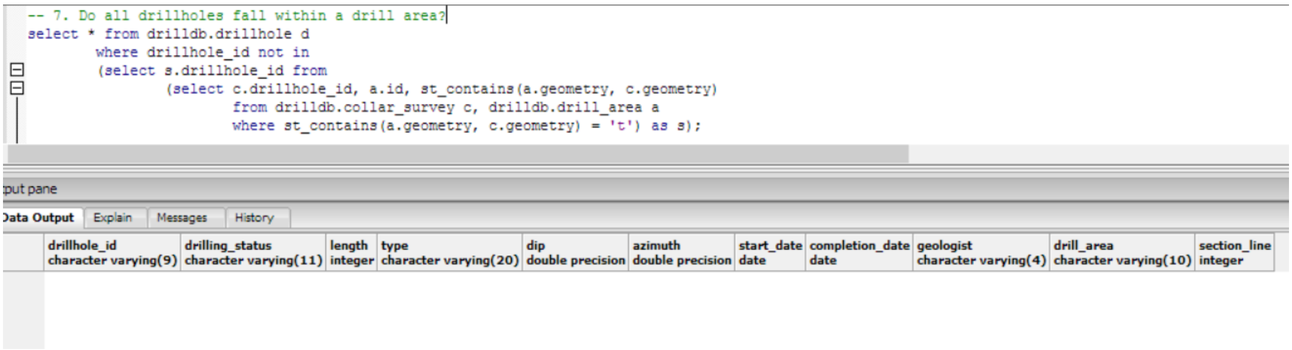
```
-- 6. Do any of the drill areas overlap each other, or extend outside of the tenement area?

select a1.id as "Drill Area 1", a2.id as "Drill Area 2", st_overlaps(a1.geometry, a2.geometry)
        as "drill area boundary overlap", a1.tenement, s.overlaps as "tenement boundary overlap"
        from drilldb.drill_area a1, drilldb.drill_area a2,
        (select t.tenement_id, d.id, st_overlaps(t.geometry, d.geometry) as overlaps
            from drilldb.tenement t, drilldb.drill_area d) as s
        where a1.id <> a2.id and a1.id < a2.id and a1.id = s.id and a1.tenement = s.tenement_id
        order by a1.id;
```

	Drill Area 1 character varying(10)	Drill Area 2 character varying(10)	drill area boundary overlap boolean	tenement character varying(8)	tenement boundary overlap boolean
1	EXAA0001	EXAA0002	t	EL5815	f
2	EXAA0001	EXBA0001	f	EL5815	f
3	EXAA0001	EXCA0001	f	EL5815	f
4	EXAA0002	EXBA0001	f	EL5815	f
5	EXAA0002	EXCA0001	f	EL5815	f
6	EXBA0001	EXCA0001	f	EL5659	f

The overlap between drill areas EXAA0001 and EXAA0002 can be seen in Figure 3.

7. Do all drillholes fall within a drill area?

```
select * from drilldb.drillhole d
    where drillhole_id not in
    (select s.drillhole_id from
        (select c.drillhole_id, a.id, st_contains(a.geometry, c.geometry)
            from drilldb.collar_survey c, drilldb.drill_area a
            where st_contains(a.geometry, c.geometry) = 't') as s);
```



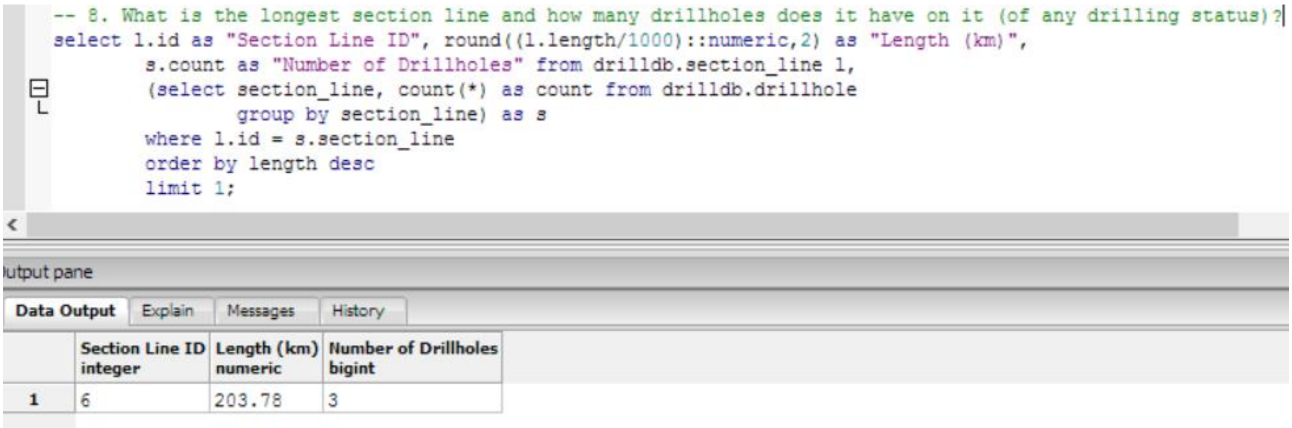
```
-- 7. Do all drillholes fall within a drill area?
select * from drilldb.drillhole d
        where drillhole_id not in
        (select s.drillhole_id from
                (select c.drillhole_id, a.id, st_contains(a.geometry, c.geometry)
                        from drilldb.collar_survey c, drilldb.drill_area a
                        where st_contains(a.geometry, c.geometry) = 't') as s);
```

drillhole_id character varying(9)	drilling_status character varying(11)	length integer	type character varying(20)	dip double precision	azimuth double precision	start_date date	completion_date date	geologist character varying(4)	drill_area character varying(10)	section_line integer

The absence of returned records shows that all drillholes fall within a drill area.

8. What is the longest section line and how many drillholes does it have on it (of any drilling status)?

```
select l.id as "Section Line ID",
    round((l.length/1000)::numeric,2) as "Length (km)",
    s.count as "Number of Drillholes" from drilldb.section_line l,
    (select section_line, count(*) as count
        from drilldb.drillhole
        group by section_line) as s
    where l.id = s.section_line
order by length desc
limit 1;
```

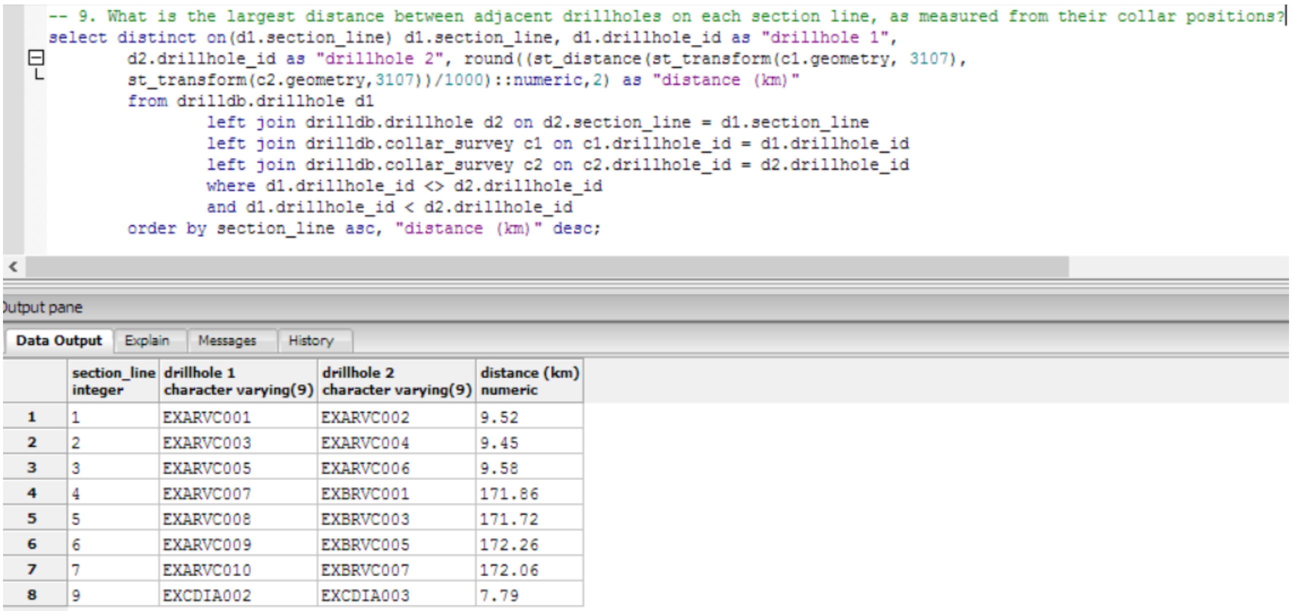


```
-- 8. What is the longest section line and how many drillholes does it have on it (of any drilling status)?
select l.id as "Section Line ID", round((l.length/1000)::numeric,2) as "Length (km)",
        s.count as "Number of Drillholes" from drilldb.section_line l,
        (select section_line, count(*) as count from drilldb.drillhole
                group by section_line) as s
        where l.id = s.section_line
        order by length desc
        limit 1;
```

	Section Line ID integer	Length (km) numeric	Number of Drillholes bigint
1	6	203.78	3

9. What is the largest distance between adjacent drillholes on each section line, as measured from their collar positions?

```
select distinct on(d1.section_line) d1.section_line,
    d1.drillhole_id as "drillhole 1", d2.drillhole_id as "drillhole 2",
    round((st_distance(st_transform(c1.geometry, 3107),
    st_transform(c2.geometry,3107))/1000)::numeric,2) as "distance (km)"
    from drilldb.drillhole d1
        left join drilldb.drillhole d2 on d2.section_line = d1.section_line
        left join drilldb.collar_survey c1 on c1.drillhole_id = d1.drillhole_id
        left join drilldb.collar_survey c2 on c2.drillhole_id = d2.drillhole_id
    where d1.drillhole_id <> d2.drillhole_id
        and d1.drillhole_id < d2.drillhole_id
order by section_line asc, "distance (km)" desc;
```

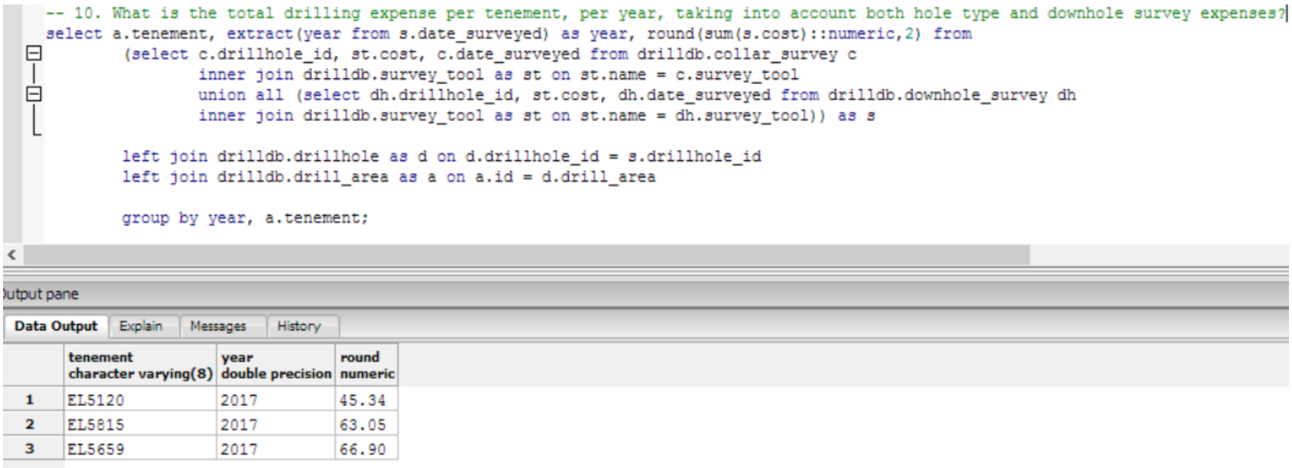


```
-- 9. What is the largest distance between adjacent drillholes on each section line, as measured from their collar positions?
select distinct on(d1.section_line) d1.section_line, d1.drillhole_id as "drillhole 1",
        d2.drillhole_id as "drillhole 2", round((st_distance(st_transform(c1.geometry, 3107),
        st_transform(c2.geometry,3107))/1000)::numeric,2) as "distance (km)"
        from drilldb.drillhole d1
                left join drilldb.drillhole d2 on d2.section_line = d1.section_line
                left join drilldb.collar_survey c1 on c1.drillhole_id = d1.drillhole_id
                left join drilldb.collar_survey c2 on c2.drillhole_id = d2.drillhole_id
                where d1.drillhole_id <> d2.drillhole_id
                and d1.drillhole_id < d2.drillhole_id
        order by section_line asc, "distance (km)" desc;
```

	section_line integer	drillhole 1 character varying(9)	drillhole 2 character varying(9)	distance (km) numeric
1	1	EXARVC001	EXARVC002	9.52
2	2	EXARVC003	EXARVC004	9.45
3	3	EXARVC005	EXARVC006	9.58
4	4	EXARVC007	EXBRVC001	171.86
5	5	EXARVC008	EXBRVC003	171.72
6	6	EXARVC009	EXBRVC005	172.26
7	7	EXARVC010	EXBRVC007	172.06
8	9	EXCDIA002	EXCDIA003	7.79

10. What is the total drilling expense per tenement, per year, taking into account both hole type and downhole survey expenses?

```
select a.tenement, extract(year from s.date_surveyed) as year,
    round(sum(s.cost)::numeric,2)
    from (select c.drillhole_id, st.cost, c.date_surveyed
        from drilldb.collar_survey c
        inner join drilldb.survey_tool as st on st.name = c.survey_tool
        union all (select dh.drillhole_id, st.cost, dh.date_surveyed
            from drilldb.downhole_survey dh
            inner join drilldb.survey_tool as st on st.name = dh.survey_tool))
                as s
    left join drilldb.drillhole as d on d.drillhole_id = s.drillhole_id
    left join drilldb.drill_area as a on a.id = d.drill_area
group by year, a.tenement;
```



```
-- 10. What is the total drilling expense per tenement, per year, taking into account both hole type and downhole survey expenses?
select a.tenement, extract(year from s.date_surveyed) as year, round(sum(s.cost)::numeric,2) from
        (select c.drillhole_id, st.cost, c.date_surveyed from drilldb.collar_survey c
                inner join drilldb.survey_tool as st on st.name = c.survey_tool
                union all (select dh.drillhole_id, st.cost, dh.date_surveyed from drilldb.downhole_survey dh
                inner join drilldb.survey_tool as st on st.name = dh.survey_tool)) as s

        left join drilldb.drillhole as d on d.drillhole_id = s.drillhole_id
        left join drilldb.drill_area as a on a.id = d.drill_area

        group by year, a.tenement;
```

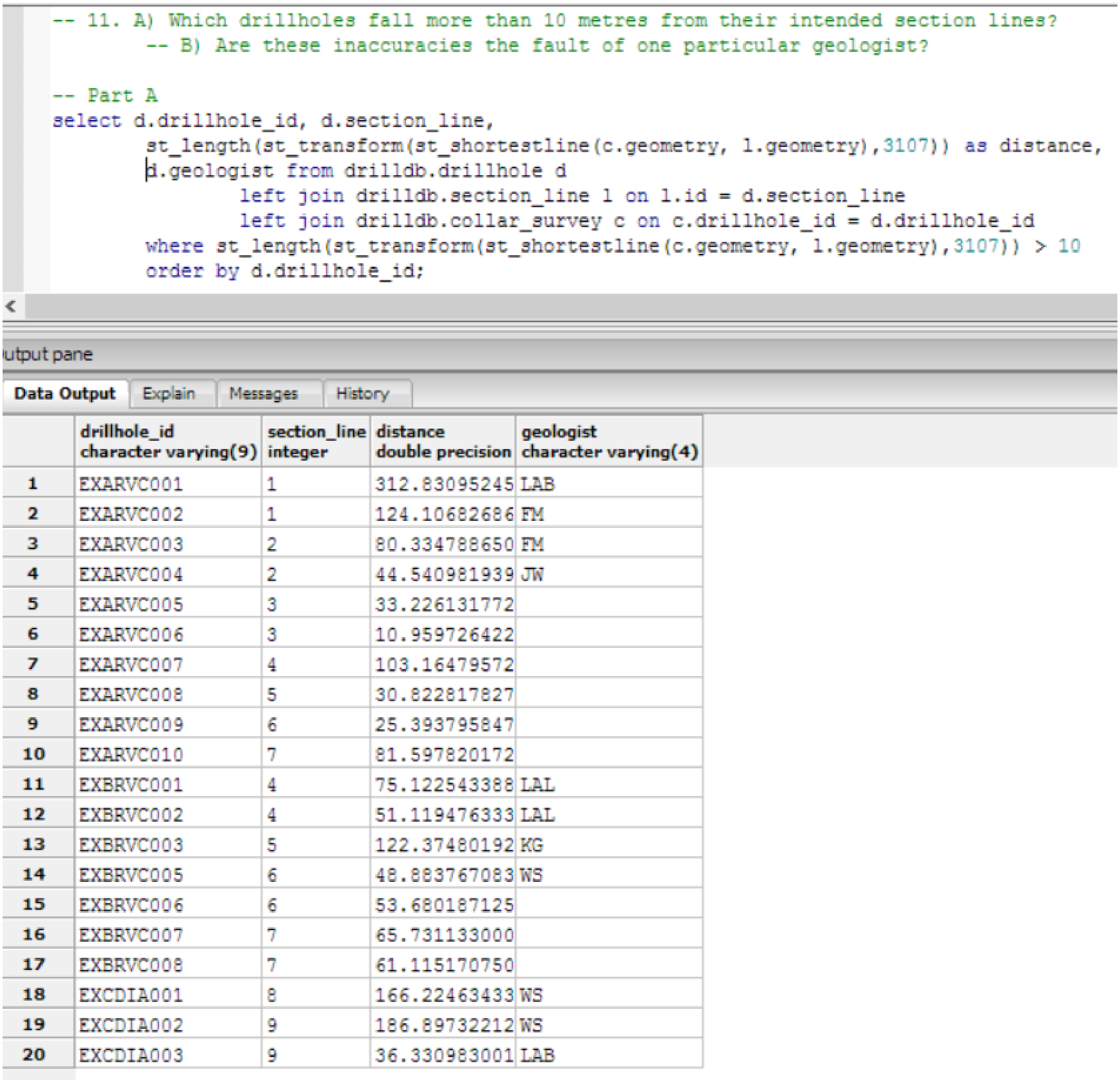
	tenement character varying(8)	year double precision	round numeric
1	EL5120	2017	45.34
2	EL5815	2017	63.05
3	EL5659	2017	66.90

11. A) Which drillholes fall more than 10 metres from their intended section lines? B) Are these inaccuracies the fault of one particular geologist?

Please note that it is not possible to meet this functional requirement in one single query since it is essentially two questions, and so it has been answered in two parts: part A is the subquery required to answer part B.

A)

```
select d.drillhole_id, d.section_line,
    st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107))
        as distance, d.geologist
    from drilldb.drillhole d
    left join drilldb.section_line l on l.id = d.section_line
    left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
    where st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107))
        > 10
order by d.drillhole_id;
```

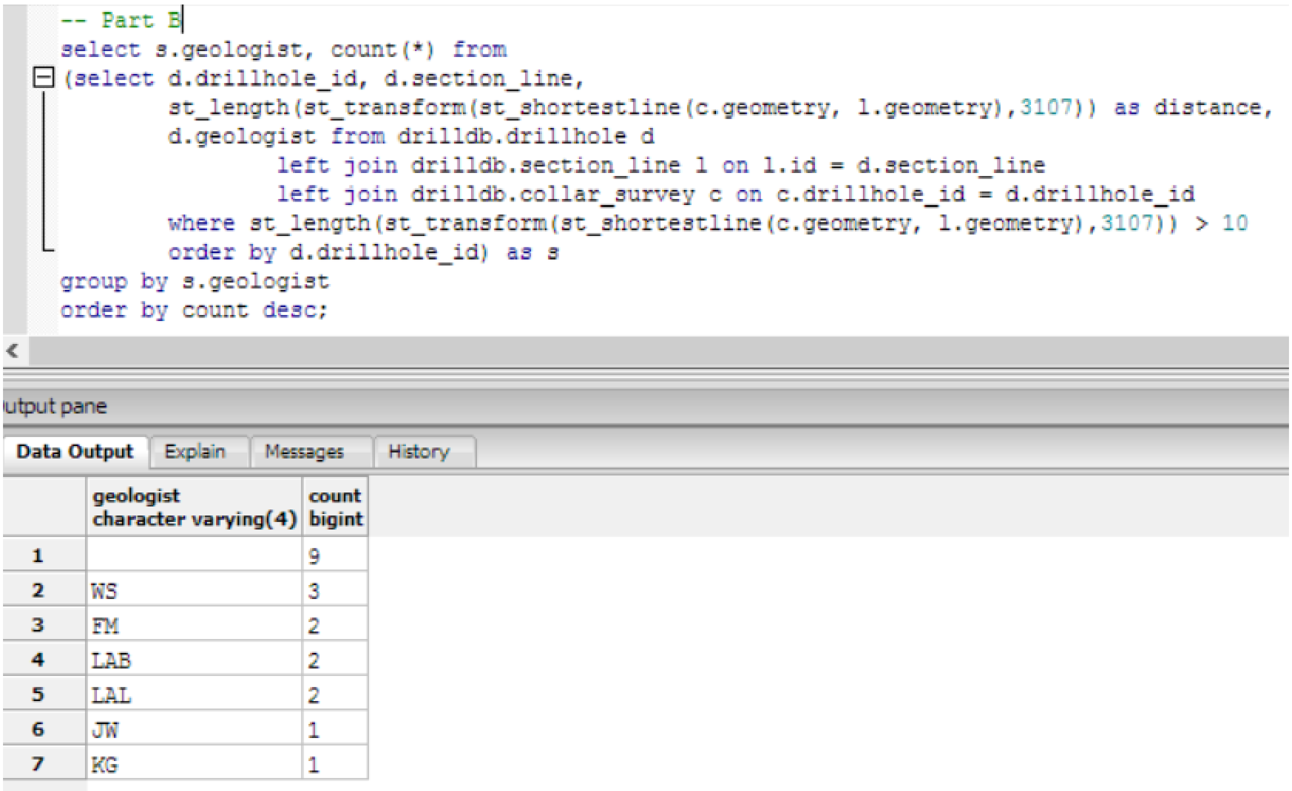


```
-- 11. A) Which drillholes fall more than 10 metres from their intended section lines?
        -- B) Are these inaccuracies the fault of one particular geologist?

-- Part A
select d.drillhole_id, d.section_line,
        st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107)) as distance,
        d.geologist from drilldb.drillhole d
                left join drilldb.section_line l on l.id = d.section_line
                left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
        where st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107)) > 10
        order by d.drillhole_id;
```

	drillhole_id character varying(9)	section_line integer	distance double precision	geologist character varying(4)
1	EXARVC001	1	312.83095245	LAB
2	EXARVC002	1	124.10682686	FM
3	EXARVC003	2	80.334788650	FM
4	EXARVC004	2	44.540981939	JW
5	EXARVC005	3	33.226131772	
6	EXARVC006	3	10.959726422	
7	EXARVC007	4	103.16479572	
8	EXARVC008	5	30.822817827	
9	EXARVC009	6	25.393795847	
10	EXARVC010	7	81.597820172	
11	EXBRVC001	4	75.122543388	LAL
12	EXBRVC002	4	51.119476333	LAL
13	EXBRVC003	5	122.37480192	KG
14	EXBRVC005	6	48.883767083	WS
15	EXBRVC006	6	53.680187125	
16	EXBRVC007	7	65.731133000	
17	EXBRVC008	7	61.115170750	
18	EXCDIA001	8	166.22463433	WS
19	EXCDIA002	9	186.89732212	WS
20	EXCDIA003	9	36.330983001	LAB

B)

```
select s.geologist, count(*)
    from (select d.drillhole_id, d.section_line,
    st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107))
        as distance, d.geologist
        from drilldb.drillhole d
    left join drilldb.section_line l on l.id = d.section_line
    left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
    where st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107))
         > 10
    order by d.drillhole_id) as s
group by s.geologist
order by count desc;
```

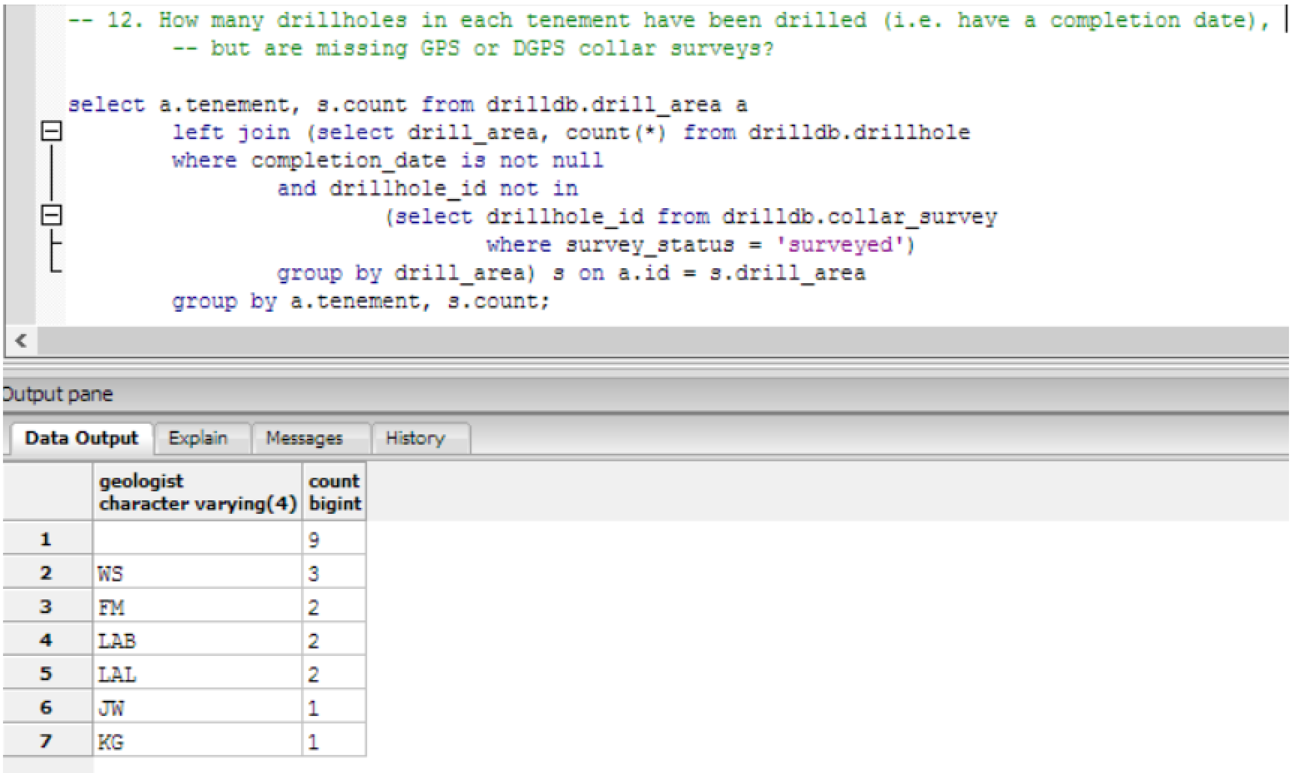


```
-- Part B
select s.geologist, count(*) from
(select d.drillhole_id, d.section_line,
        st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107)) as distance,
        d.geologist from drilldb.drillhole d
                left join drilldb.section_line l on l.id = d.section_line
                left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
        where st_length(st_transform(st_shortestline(c.geometry, l.geometry),3107)) > 10
        order by d.drillhole_id) as s
group by s.geologist
order by count desc;
```

	geologist character varying(4)	count bigint
1		9
2	WS	3
3	FM	2
4	LAB	2
5	LAL	2
6	JW	1
7	KG	1

12. How many drillholes in each tenement have been drilled (i.e. have a completion date), but are missing GPS or DGPS collar surveys?

```
select a.tenement, s.count
    from drilldb.drill_area a
    left join (select drill_area, count(*)
        from drilldb.drillhole
        where completion_date is not null and drillhole_id not in
            (select drillhole_id
            from drilldb.collar_survey
            where survey_status = 'surveyed')
        group by drill_area) s on a.id = s.drill_area
group by a.tenement, s.count
```

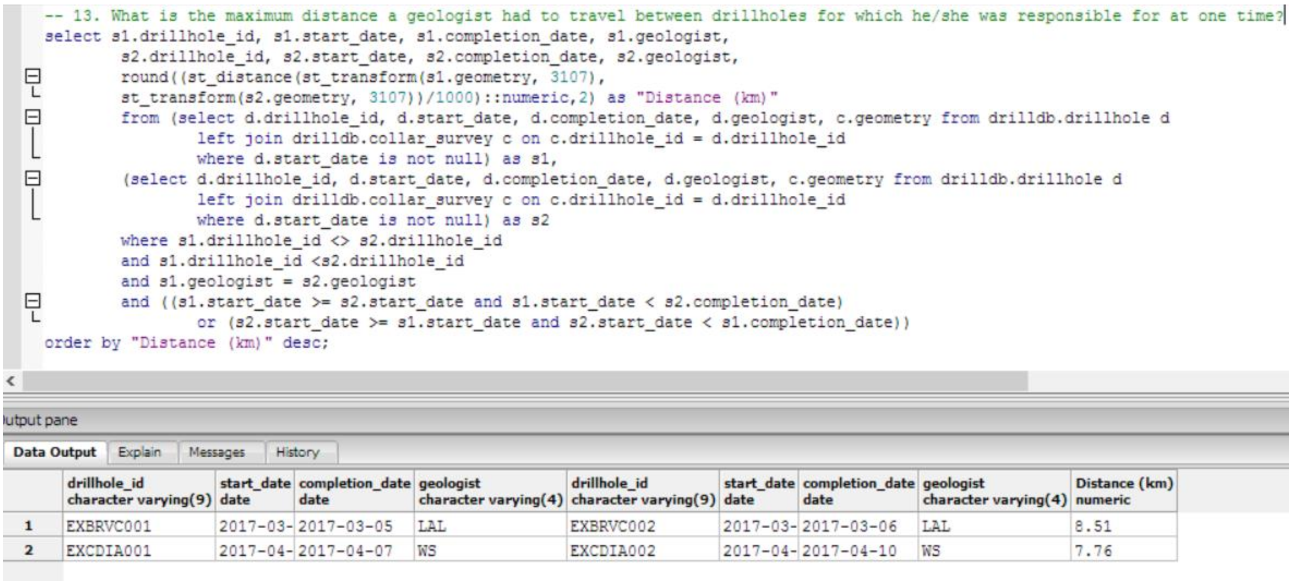


```
-- 12. How many drillholes in each tenement have been drilled (i.e. have a completion date),
        -- but are missing GPS or DGPS collar surveys?

select a.tenement, s.count from drilldb.drill_area a
        left join (select drill_area, count(*) from drilldb.drillhole
        where completion_date is not null
                and drillhole_id not in
                        (select drillhole_id from drilldb.collar_survey
                                where survey_status = 'surveyed')
                group by drill_area) s on a.id = s.drill_area
        group by a.tenement, s.count;
```

	geologist character varying(4)	count bigint
1		9
2	WS	3
3	FM	2
4	LAB	2
5	LAL	2
6	JW	1
7	KG	1

13. What is the maximum distance a geologist had to travel between drillholes for which he/she was responsible for at one time?

```
select s1.drillhole_id, s1.start_date, s1.completion_date, s1.geologist,
    s2.drillhole_id, s2.start_date, s2.completion_date, s2.geologist,
    round((st_distance(st_transform(s1.geometry, 3107),
    st_transform(s2.geometry, 3107))/1000)::numeric,2) as "Distance (km)"
    from (select d.drillhole_id, d.start_date, d.completion_date, d.geologist,
        c.geometry
        from drilldb.drillhole d
        left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
            where d.start_date is not null) as s1,
    (select d.drillhole_id, d.start_date, d.completion_date, d.geologist,
        c.geometry
        from drilldb.drillhole d
        left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
            where d.start_date is not null) as s2
    where s1.drillhole_id <> s2.drillhole_id
    and s1.drillhole_id <s2.drillhole_id
    and s1.geologist = s2.geologist
    and ((s1.start_date >= s2.start_date and s1.start_date < s2.completion_date)
        or (s2.start_date >= s1.start_date
        and s2.start_date < s1.completion_date))
order by "Distance (km)" desc;
```



```
-- 13. What is the maximum distance a geologist had to travel between drillholes for which he/she was responsible for at one time?
select s1.drillhole_id, s1.start_date, s1.completion_date, s1.geologist,
        s2.drillhole_id, s2.start_date, s2.completion_date, s2.geologist,
        round((st_distance(st_transform(s1.geometry, 3107),
        st_transform(s2.geometry, 3107))/1000)::numeric,2) as "Distance (km)"
        from (select d.drillhole_id, d.start_date, d.completion_date, d.geologist, c.geometry from drilldb.drillhole d
                left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
                where d.start_date is not null) as s1,
        (select d.drillhole_id, d.start_date, d.completion_date, d.geologist, c.geometry from drilldb.drillhole d
                left join drilldb.collar_survey c on c.drillhole_id = d.drillhole_id
                where d.start_date is not null) as s2
        where s1.drillhole_id <> s2.drillhole_id
        and s1.drillhole_id <s2.drillhole_id
        and s1.geologist = s2.geologist
        and ((s1.start_date >= s2.start_date and s1.start_date < s2.completion_date)
                or (s2.start_date >= s1.start_date and s2.start_date < s1.completion_date))
order by "Distance (km)" desc;
```

	drillhole_id character varying(9)	start_date date	completion_date date	geologist character varying(4)	drillhole_id character varying(9)	start_date date	completion_date date	geologist character varying(4)	Distance (km) numeric
1	EXBRVC001	2017-03-	2017-03-05	LAL	EXBRVC002	2017-03-	2017-03-06	LAL	8.51
2	EXCDIA001	2017-04-	2017-04-07	WS	EXCDIA002	2017-04-	2017-04-10	WS	7.76

Part B – 3D GIS Support in MapInfo Discover 3D

MapInfo Discover 3D is a GIS package, produced by Pitney Bowes, designed for geoscientific applications within the natural resources industry, primarily targeted at the mining and exploration industry. Its main purpose is to aid in the visualisation, integration, and combined analysis of multiple datasets of differing data sources, from geophysical surveys to drillhole data and field mapping. The package comprises a number of MapInfo products built upon MapInfo Pro, Pitney Bowes' industry-non-specific Desktop GIS solution. Discover 3D is one of these extensions that adds 3D functionality to what is, natively, a 2D GIS (Pitney Bowes, 2016c).

The main advantage that Discover 3D has over traditional 2D GIS packages is that it allows data to be visualised, modelled, and analysed in three dimensions. Visualisation in 3D allows users to gain new insights into datasets that may not have been apparent when viewed in 2D (Figure 6). 'On-the-fly' reprojection of coordinates in 3D space is supported, as is the functionality to export and share 3D views though Discover Viewer, and produce movies of data 'fly-throughs' (Pitney Bowes, 2016b).

MapInfo Discover 3D does not require all data to be captured in 3D, since existing 2D datasets, including high-resolution raster grids, can be draped over terrain models to obtain Z-coordinates. 2D gridded surfaces, such as aero-magnetics or gravity, can be transformed into 3D based on relative grid values, and 3D objects created through vertical extrusion of points, lines, and polygons, although these features must exist in the same XY plane. Although draping and extrusion of 2D objects is a 2.5D GIS function (ESRI, 2016), MapInfo differs in that it stores the resulting surfaces as 3D objects. Unlike 2.5D surfaces, where the vertical component is stored as an attribute to a single XY location, these 3D objects comprise points located in full 3D space. This enables multiple Z values for the same X and Y coordinates (Mugumbu, 2000). These 3D objects can be viewed an manipulated is the same way as other 3D objects (Pitney Bowes, 2015).

The extrusion function can also be used to create 3D buildings from footprints, however the level of detail is minimal with no option to add roof structure or external features. 3D image objects, such as vegetation, vehicles, and signposts can be added to the environment, and buildings coloured with simple textures (Figure 7) (Pitney Bowes, 2015).

Another selling-point of MapInfo Discover 3D is its ability to display drillhole traces spatially, with multiple downhole attributes shown concurrently (Figure 5) (Pitney Bowes, 2016b). Simultaneous visualisation of all relevant data significantly aids with trend-detection, leading to more accurate interpretation wireframes, and therefore more accurate geological models. Wireframes are 3D solids that contain a volume, and that are created from polygon interpretations using the '3D solid generator' function. Polygons can be imported from 2D cross-section interpretations, or through digitising interpretations directly into the 3D environment (Figure 8). 3D solids can be reshaped using various manipulation functions, such as cutting by other intersecting planes or solids, and their volumes calculated. 3D solids can also be imported from external sources (Pitney Bowes, 2015).

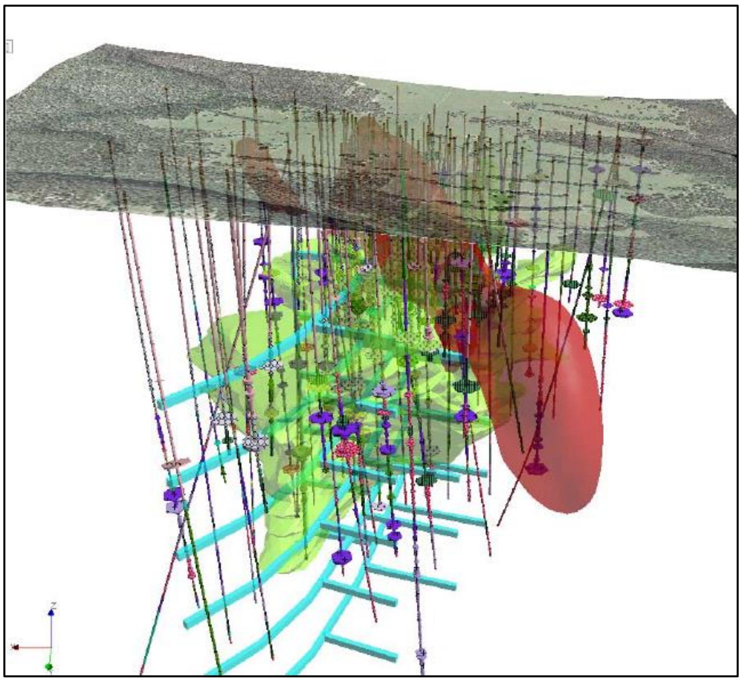
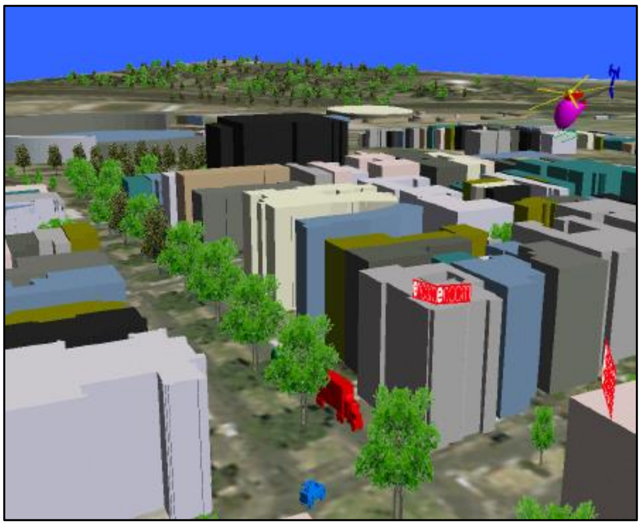


Figure 6: 3D visualisation of data using MapInfo Discover 3D, showing underground mine workings (blue) and a wireframe of the ore body (green) as 3D vector files, along with drillholes, a georeferenced image, and an isosurface from a voxel model (red) (Pitney Bowes, 2015). Since 3D visualisation is closer to reality, it makes real-life problems easier to comprehend, thus improving decision-making.



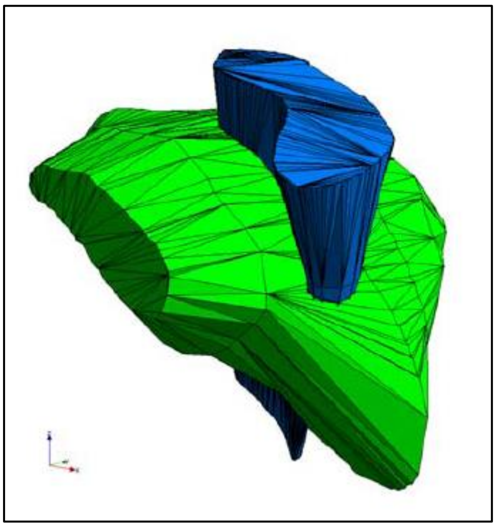


Figure 7 (left): Buildings created through extrusion of polygons, along with 3D image objects of trees, vehicles, and signposts (Pitney Bowes, 2015).
Figure 8 (right): Example of two intersecting 3D wireframes created using the '3D solid generator' (Pitney Bowes, 2015).

MapInfo Discover 3D also offers the functionality to create block/voxel models using either inverse distance weighting or kriging. Although both of these methods are available in 2D GIS packages, MapInfo uses these interpolation techniques to estimate the data into 3D 'blocks' of a user-specified size. The resulting models can then be clipped by objects, such as terrain models or 3D solids, or viewed as isosurfaces. Block models from other mining packages can also be imported and manipulated in the 3D environment (Pitney Bowes, 2015).

A side-effect of this increase in dimensionality is that file sizes are also enlarged. This requires both increased storage capabilities, and higher performance computers to handle the visualisation of these larger datasets.

Not only is it necessary to store this data in a range of coordinate reference systems, but also transform between projections. This is particularly important when projecting 2D information into 3D, as the 2D dataset may not be stored in a 3D enabled CRS. MapInfo utilises the Open Geospatial Consortium GeoPackage as its principal database, that allows storage of data as a 3D geometry dataset (Pitney Bowes, 2016c). 3D data, particularly drillholes, are often stored as a collection of 2D spatial, and non-spatial tables that are formed into 3D objects for visualisation an analysis purposes. Unlike traditional spatially and non-spatially-enabled databases that often have size limits of a few gigabytes, GeoPackage databases can store up to 100TB of data. It also has the ability to increase processing speeds through the reduction of storage space consumed by empty fields within the tables themselves (Pitney Bowes, 2016c).

Whilst the 3D extension is an improvement over traditional 2D GIS packages, the 3D functionality is still limited to 3D specific files. Whilst it allows for some 3D functionality, much of the 3D information is acquired through draping or extrusion of 2D surfaces or grids, or georeferencing of 2D sections along planes, rather than enforcing 3D data collection. Discover 3D extension is also marketed, exclusively, as a geoscience solution, and its 3D support would be of little use to urban planners. Although there is reference to its existence on the Pitney Bowes website, the product webpage can only be found through the site's search engine (Pitney Bowes, 2017). This suggests that, although industry-specific 3D GIS support exists, there is little demand for generic, or urban-orientated, 3D GIS solutions.

Geoscience is a discipline that is inherently 3D, and the ability to both visualise and analyse geological data in 3D is essential to the understanding of the geology. Although the features offered by MapInfo Discover 3D are powerful, they are an extension of what is, natively, a 2D package. In an industry where 3D data collection is essential, particularly during the mining phase, this 2D functionality is somewhat obsolete, making native 3D software packages such as Leapfrog Geo (ARANZ Geo Limited, 2017) and Datamine Studio EM/RM (Datamine, 2016) more suitable for these purposes.

Word count: 991

Bibliography

ARANZ Geo Limited, 2017. *Leapfrog Geo.* [Online]
Available at: http://www.leapfrog3d.com/products/leapfrog-geo
[Accessed April 2017].

Datamine, 2016. *Resource Modelling Software.* [Online]
Available at: http://www.dataminesoftware.com/software/resource-modelling-software/
[Accessed April 2017].

ESRI, 2016. *About using extrusion as 3D symbology.* [Online]
Available at: http://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/about-using-extrusion-as-3d-symbology.htm
[Accessed April 2017].

Mugumbu, W., 2000. *Why 3D GIS?.* [Online]
Available at: http://www.hbp.usm.my/thesis/HeritageGIS/master/
[Accessed 2017].

Pitney Bowes, 2015. *MapInfo Discover 3D 2015 User Guide,* s.l.: Pitney Bowes.

Pitney Bowes, 2016a. *MapInfo Pro™ v16 Data Sheet.* [Online]
Available at: http://www.pitneybowes.com/content/dam/pitneybowes/us/en/location-intelligence/geographic-information-systems/mapinfo-pro-v16/16-DCS-06822%20MI%20Pro-DataSht_US_web.pdf
[Accessed April 2017].

Pitney Bowes, 2016b. *MapInfo Discover™ 2016 and MapInfo Discover™ 3D 2016 64-bit bundles.* [Online]
Available at:
http://du8h850wz9371.cloudfront.net/documentation/discover/MapInfoDiscover2016_DataSheet.pdf
[Accessed April 2017].

Pitney Bowes, 2016c. *MapInfo Discover 2016 Release Notes.* [Online]
Available at:
http://du8h850wz9371.cloudfront.net/documentation/discover/MapInfoDiscover2016_ReleaseNotes.pdf
[Accessed April 2017].

Pitney Bowes, 2017. *Location Analytics & Geographical Information Systems.* [Online]
Available at: http://www.pitneybowes.com/us/location-intelligence/geographic-information-systems.html
[Accessed April 2017].