2022 - Data Analytics for Immersive Environments - CA4 - RDBMS & Linear Regression Project

CA4 Part 2 - Querying Database

Joe O'Regan

2023-01-11

Repo Link

https://github.com/joeaoregan/2022_DAIE_CA4_JOR1

ER Diagram

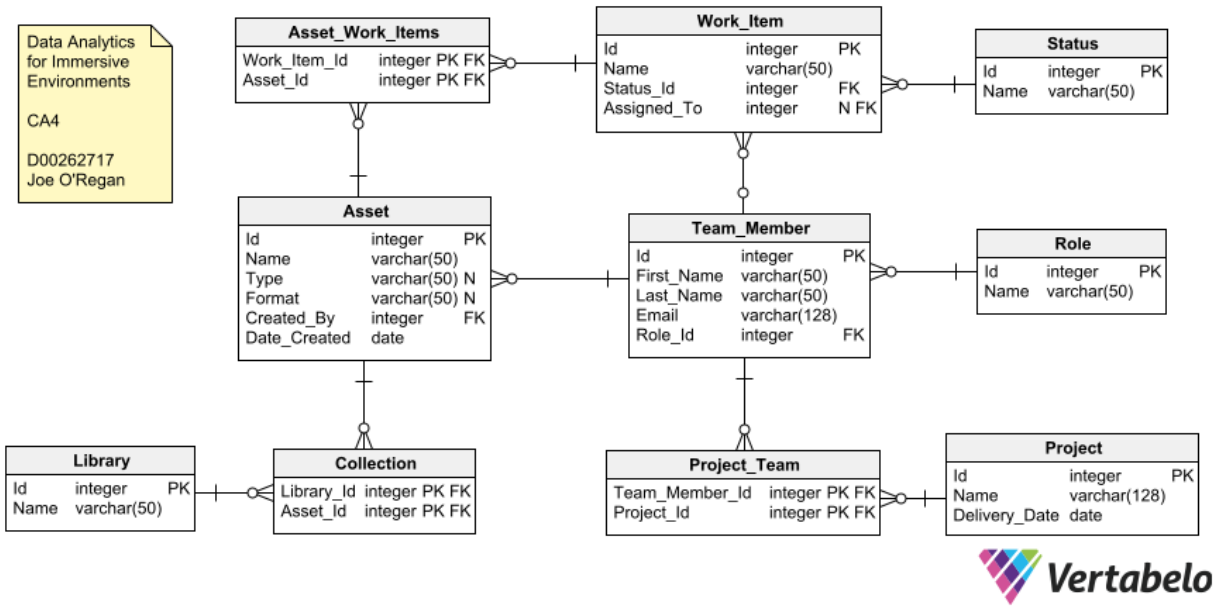

Figure 1: Entity Relationship Diagram

Make Database Connection

Connect to the sqlite database file.

```
# connect to the sqlite database file
conn <- dbConnect(RSQLite::SQLite(), "daie_ca4_data.sqlite")
```

Show Table Data

Contents of tables to check queries against.

Format Table Function

Function to format tables for HTML and PDF output. Display tables using knitr library's kable function and kableExtra to format tables.

```
# type parameter sets class type for different table formatting
# bgcolor parameter sets the table heading colour
# caprt paramter sets the table caption
# paste() used to concatenate strings
# sep - the separator in the concatenated string
# separate formatting is required for html and pdf tables due to css class error
# stripe_color sets alternate row colour for pdf tables and queries
data_format.function <- function(data, capt="", type="table", bgcolour="#28B3F9") {
  data %>%
    {
      if (is_html_output()) { # if the output is HTML add class attribute
        kbl(., caption = capt,
          table.attr=paste("class='",type,"-striped ",type,'-',"hover'", sep="")) %>%
            kable_styling(bootstrap_options = c("striped", "hover"))
      }
      else if (is_latex_output()) { # if the output is PDF ignore class attribute
        kbl(., caption = capt) %>%
          kable_styling(latex_options = c("striped","HOLD_position"),
                        stripe_color=ifelse((type == "table"),"#D3EDF9","#FBBBBB"))
      }
    } %>% # pdf output keep tables in position
        row_spec(0, background = bgcolour) # table heading colour
}
```

Status

Status table data.

```
SELECT * FROM Status -- get all data in Status table
```

```
data_format.function(status_data, "Status") # format data with function above
```

Table 1: Status

Id	Name
1	To Do
2	In Progress
3	Review
4	Done

Role

Role table data.

```
SELECT * FROM Role -- get all data in Role table
```

```
data_format.function(role_data, "Role") # format data with function above
```

Table 2: Role

Id	Name
1	Project Manager
2	Programmer
3	Tester
4	Artist
5	3D Modeller
6	Environment Modeller
7	Animator
8	Shading Artist
9	Concept Artist

Team_Member

Team_Member table data.

```
SELECT * FROM Team_Member -- get all data in Team_Member table
```

```
data_format.function(team_member_data, "Team Member") # format data with function above
```

Table 3: Team Member

Id	First_Name	Last_Name	Email	Role_Id
1	Joe	O'Regan	joe.oregan@daie.ca4	2
2	Derp	McDerp	derpmcderp@daie.ca4	1
3	Herpderp	Derpderpenson	hd.derpderpenson@daie.ca4	3
4	Herpa	Derpderp	herpa.derpderp@daie.ca4	4
5	De	Rpderp	de.rpderp@daie.ca4	5
6	Pred	Prehpred	predprehpred@daie.ca4	6
7	Derpa	Derpa	derpaderpa@daie.ca4	9
8	Herpa	Derpa	herpaderpa@daie.ca4	8
9	HerpaDerpa	McDerpa	herpaderpa.mcderpa@daie.ca4	7
10	Joe	Derp	j.derp@daie.ca4	2
11	Jon	Herpaderp	jherpaderp@daie.ca4	7
12	Joblot	O'Stuff	joblot.ostuff@daie.ca4	3
13	Joderp	Herpderpenson	j.herpderpenson@daie.ca4	4
14	Jo	McQueryfiller	j.mcqueryfiller@daie.ca4	1

Work_Item

Work_Item table data. Formatted differently (HTML output) to test different kableExtra formatting options.

```
SELECT * FROM Work_Item -- get all data in Work_Item table
```

```
# data_format.function(work_item_data, "Work Item") # format data with function above

data <- work_item_data # dataframe
data$Id = row.names(work_item_data)
row.names(data) <- NULL
# in HTML output format the Status_Id column so 1 (To Do) is red, and 4 (Done) is green
if (is_html_output()) {
  data$Status_Id = cell_spec(data$Status_Id, color = "white",
                             background = ifelse(data$Status_Id == 4, "green",
                                                 ifelse(data$Status_Id == 1,
                                                        "red", "black")))
}
data <- data[c("Id", "Name", "Status_Id", "Assigned_To")] # error if not escaped for html
# data <- data.frame(data)

kbl(data, escape=ifelse(is_html_output(), FALSE, TRUE), caption = "Work Item") %>%
  kable_styling(c("striped", "hover")) %>%
  kable_styling(latex_options = "striped", stripe_color="#D3EDF9") %>% # pdf stripe colour
    row_spec(0, background = "#28B3F9") # table heading colour
```

Table 4: Work Item

Id	Name	Status_Id	Assigned_To
1	Art Thingy	2	4
2	Art Test Thingy	3	3
3	Environment Model Thingy	2	6
4	Art Concept Thingy	4	7
5	Art Shading Thingy	4	8
6	Random 3D Model	2	5
7	3D Model Test Obj	2	3
8	Random Blueprint	2	1
9	Blueprint Thingy	2	2
10	Blueprint Test	3	2
11	Art Test	1	12
12	Query Model Thingy	3	11
13	Another Test	4	13

Project

Project table data.

```
SELECT * FROM Project -- get all data in Project table
```

```
data_format.function(project_data, "Project") # format data with function above
```

Table 5: Project

Id	Name	Delivery_Date
1	Art Proj	2023-01-11
2	DAIE CA4	2023-01-20
3	New Project	2023-01-24
4	Old Project	2022-12-14
5	Christmas 2022 Project	2022-12-25
6	Date Range Project	2023-01-17
7	Another Date Range Project	2023-02-01
8	Project Filler	2023-03-01
9	Derp Project	2023-03-17
10	Hmmm I Ran Out of Names	2023-01-20

Project_Team

Project_Team table data.

```
SELECT * FROM Project_Team -- get all data in Project_Team table
```

```
data_format.function(project_team_data, "Project Team") # format data with function above
```

Table 6: Project Team

Team_Member_Id	Project_Id
1	1
2	1
3	1
4	1
5	1
2	2
6	2
7	2
8	2
9	2
2	3
10	3
11	3
14	4
12	4
13	5
14	6
14	7

Asset

Asset table data.

```
SELECT * FROM Asset -- get all data in Asset table
```

```
data_format.function(asset_data, "Asset") # format data with function above
```

Table 7: Asset

Id	Name	Type	Format	Created_By	Date_Created
1	Random Blueprint Asset	Combination of Blueprints	Zip file	1	2023-01-11
2	Random Art Asset	NA	NA	4	2023-01-10
3	Art Asset Thingy	NA	NA	4	2023-01-10
4	Environment Asset Thingy	Tree for use in Environment	NA	4	2023-01-02

Asset_Work_Items

Asset_Work_Items table data.

```
SELECT * FROM Asset_Work_Items -- get all data in Asset_Work_Items table
```

```
data_format.function(asset_work_items_data,
                     "Asset Work Items") # format data with function above
```

Table 8: Asset Work Items

Work_Item_Id	Asset_Id
8	1
9	1
10	1
1	2
2	2
4	3
5	3
3	4
6	4
7	4

Library

Library table data.

```
SELECT * FROM Library -- get all data in Library table
```

```
data_format.function(library_data, "Library") # format data with function above
```

Table 9: Library

Id	Name
1	Programming
2	Models
3	Scenery
4	Characters

Collection

Collection table data.

```
SELECT * FROM Collection -- get all data in Collection table
```

```
data_format.function(collection_data, "Collection") # format data with function above
```

Table 10: Collection

Library_Id	Asset_Id
1	1
2	2
2	3
3	4

Database Querying

Query the above database using SQL queries demonstrating the following SQL concepts:

1. SELECT with WHERE, LIKE, and OR
2. SELECT with DISTINCT and ORDER BY
3. Inner Join
4. Subquery with SELECT
5. SELECT across a date range

1. SELECT with WHERE, LIKE, and OR

1.1 Select with WHERE

Find Team Members who have the first name Joe.

```
SELECT * FROM Team_Member WHERE First_Name = 'Joe';
```

Format the query output as a table with kable and kableExtra function above.

```
data_format.function(query1_select_with_where, "Team members with first name Joe",
                     "query", "#FF0000") # format data changing function defaults
```

Table 11: Team members with first name Joe

Id	First_Name	Last_Name	Email	Role_Id
1	Joe	O'Regan	joe.oregan@daie.ca4	2
10	Joe	Derp	j.derp@daie.ca4	2

1.2 SELECT with LIKE

Using wildcards to substitute one or more characters in a string.

1.2.1 Select with LIKE and '_' wildcard

Find Team Members with first name with 3 characters beginning with "jo" using '_' wildcard.

```
SELECT * FROM Team_Member WHERE First_Name LIKE "jo_";
```

```
data_format.function(query2a_select_with_like,
                     "Team members with name beginning with jo and at least 3 characters",
                     "query", "#FF0000")
```

Table 12: Team members with name beginning with jo and at least 3 characters

Id	First_Name	Last_Name	Email	Role_Id
1	Joe	O'Regan	joe.oregan@daie.ca4	2
10	Joe	Derp	j.derp@daie.ca4	2
11	Jon	Herpaderp	jherpaderp@daie.ca4	7

1.2.2 Select with LIKE and '%' wildcard

Find Team Members with last name containing the string "derp" using '%' wildcard.

```
SELECT * FROM Team_Member WHERE Last_Name LIKE "%derp%";
```

```
data_format.function(query2b_select_with_like,
                     # need to escape \ and %
                     "Team member last name contains 'derp' string using \\% wildcard",
                     "query", "#FF0000")
```

Table 13: Team member last name contains 'derp' string using % wildcard

Id	First_Name	Last_Name	Email	Role_Id
2	Derp	McDerp	derpmcderp@daie.ca4	1
3	Herpderp	Derpderpenson	hd.derpderpenson@daie.ca4	3
4	Herpa	Derpderp	herpa.derpderp@daie.ca4	4
5	De	Rpderp	de.rpderp@daie.ca4	5
7	Derpa	Derpa	derpaderpa@daie.ca4	9
8	Herpa	Derpa	herpaderpa@daie.ca4	8
9	HerpaDerpa	McDerpa	herpaderpa.mcderpa@daie.ca4	7
10	Joe	Derp	j.derp@daie.ca4	2
11	Jon	Herpaderp	jherpaderp@daie.ca4	7
13	Joderp	Herpderpenson	j.herpderpenson@daie.ca4	4

1.2.3 Select with LIKE and '%' and '_' wildcard

Find Team Members with first name beginning with "jo" with at least 3 characters, i.e. excludes "Jo".

```
SELECT * FROM Team_Member WHERE First_Name LIKE "jo_%";
```

```
data_format.function(query2c_select_with_like,
                     "Team member first name of at least 3 characters beginning with 'jo'",
                     "query", "#FF0000")
```

Table 14: Team member first name of at least 3 characters beginning with 'jo'

Id	First_Name	Last_Name	Email	Role_Id
1	Joe	O'Regan	joe.oregan@daie.ca4	2
10	Joe	Derp	j.derp@daie.ca4	2
11	Jon	Herpaderp	jherpaderp@daie.ca4	7
12	Joblot	O'Stuff	joblot.ostuff@daie.ca4	3
13	Joderp	Herpderpenson	j.herpderpenson@daie.ca4	4

1.3 SELECT with OR

1.3.1 OR with numeric comparison

Select Team Members where the Role_Id is 2 OR 7.

```
SELECT * FROM Team_Member WHERE Role_Id = 2 OR Role_Id = 7;
```

```
data_format.function(query3a_select_with_or, "Team member with role id of 2 or 7",
                     "query", "#FF0000")
```

Table 15: Team member with role id of 2 or 7

Id	First_Name	Last_Name	Email	Role_Id
1	Joe	O'Regan	joe.oregan@daie.ca4	2
9	HerpaDerpa	McDerpa	herpaderpa.mcderpa@daie.ca4	7
10	Joe	Derp	j.derp@daie.ca4	2
11	Jon	Herpaderp	jherpaderp@daie.ca4	7

1.3.2 OR with string comparison

Select Team Members whose first name is “Herpa” or last name is “Derpa”.

```
SELECT * FROM Team_Member WHERE First_Name = "Herpa" OR Last_Name = "Derpa";
```

```
data_format.function(query3b_select_with_or,
                     "Team members with first name 'Herpa' or last name 'Derpa'",
                     "query", "#FF0000")
```

Table 16: Team members with first name ’Herpa’ or last name ’Derpa’

Id	First_Name	Last_Name	Email	Role_Id
4	Herpa	Derpderp	herpa.derpderp@daie.ca4	4
7	Derpa	Derpa	derpaderpa@daie.ca4	9
8	Herpa	Derpa	herpaderpa@daie.ca4	8

1.4 SELECT with WHERE, LIKE and OR

Find work items with Name beginning with a string like “art” or have a Status_Id of 3.

```
SELECT * FROM Work_Item WHERE Name LIKE "art%" OR Status_Id = 3;
```

```
data_format.function(query4a_select_with_where_like_or,
                     "Work item with name beginning with the string 'art'",
                     "query", "#FF0000")
```

Table 17: Work item with name beginning with the string ’art’

Id	Name	Status_Id	Assigned_To
1	Art Thingy	2	4
2	Art Test Thingy	3	3
4	Art Concept Thingy	4	7
5	Art Shading Thingy	4	8
10	Blueprint Test	3	2
11	Art Test	1	12
12	Query Model Thingy	3	11

1.5 SELECT with NOT

Find Work Items where the name doesn't contain "Thingy".

```
SELECT * FROM Work_Item WHERE Name NOT LIKE "%thingy%";
```

```
data_format.function(query4b_select_with_not,
                     "Work items that do not containt the string 'thingy'",
                     "query", "#FF0000")
```

Table 18: Work items that do not containt the string 'thingy'

Id	Name	Status_Id	Assigned_To
6	Random 3D Model	2	5
7	3D Model Test Obj	2	3
8	Random Blueprint	2	1
10	Blueprint Test	3	2
11	Art Test	1	12
13	Another Test	4	13

2. SELECT with DISTINCT and ORDER BY

2.1 SELECT with DISTINCT

Find the unique status IDs currently in the Work_Item table.

```
SELECT DISTINCT Status_Id FROM Work_Item;
```

```
data_format.function(query5_select_distinct,
                     "Get the unique values for Status Id in Work Item",
                     "query", "#FF0000") # format query
```

Table 19: Get the unique values for Status Id in Work Item

Status_Id
2
3
4
1

2.2 SELECT with ORDER BY

Display work items ordered by assigned_to (Team_Member.Id).

```
SELECT * FROM Work_Item ORDER BY Assigned_To;
```

```
data_format.function(query6_select_order_by,
                     "Show Work Items ordered by Assigned To ID number",
                     "query", "#FF0000")
```

Table 20: Show Work Items ordered by Assigned To ID number

Id	Name	Status_Id	Assigned_To
8	Random Blueprint	2	1
9	Blueprint Thingy	2	2
10	Blueprint Test	3	2
2	Art Test Thingy	3	3
7	3D Model Test Obj	2	3
1	Art Thingy	2	4
6	Random 3D Model	2	5
3	Environment Model Thingy	2	6
4	Art Concept Thingy	4	7
5	Art Shading Thingy	4	8
12	Query Model Thingy	3	11
11	Art Test	1	12
13	Another Test	4	13

2.3 SELECT with ORDER BY ASC

Display work items ordered by Status_Id (Status.Id) in ascending order.

```
SELECT * FROM Work_Item ORDER BY Status_Id ASC;
```

```
data_format.function(query7_select_order_by_asc,
                     "Show Work Items ordered by Status Id code", "query", "#FF0000")
```

Table 21: Show Work Items ordered by Status Id code

Id	Name	Status_Id	Assigned_To
11	Art Test	1	12
1	Art Thingy	2	4
3	Environment Model Thingy	2	6
6	Random 3D Model	2	5
7	3D Model Test Obj	2	3
8	Random Blueprint	2	1
9	Blueprint Thingy	2	2
2	Art Test Thingy	3	3
10	Blueprint Test	3	2
12	Query Model Thingy	3	11
4	Art Concept Thingy	4	7
5	Art Shading Thingy	4	8
13	Another Test	4	13

2.4 SELECT with ORDER BY DESC

Display work items ordered by Assigned_To (Team_Member.Id) in descending order.

```
SELECT * FROM Work_Item ORDER BY Assigned_To DESC;
```

```
data_format.function(query8_select_order_by_desc,
                     "Work Items ordered by Assigned To ID number in descending order",
                     "query", "#FF0000")
```

Table 22: Work Items ordered by Assigned To ID number in descending order

Id	Name	Status_Id	Assigned_To
13	Another Test	4	13
11	Art Test	1	12
12	Query Model Thingy	3	11
5	Art Shading Thingy	4	8
4	Art Concept Thingy	4	7
3	Environment Model Thingy	2	6
6	Random 3D Model	2	5
1	Art Thingy	2	4
2	Art Test Thingy	3	3
7	3D Model Test Obj	2	3
9	Blueprint Thingy	2	2
10	Blueprint Test	3	2
8	Random Blueprint	2	1

2.5 SELECT with DISTINCT and ORDER By

Display work items ordered by Assigned_To (Team_Member.Id) in descending order.

```
SELECT DISTINCT Assigned_To FROM Work_Item ORDER BY Assigned_To;
```

```
data_format.function(query9_select_distinct_order_by,
                     "Show distinct list of team members with work assigned to them",
                     "query", "#FF0000")
```

Table 23: Show distinct list of team members with work assigned to them

Assigned_To
1
2
3
4
5
6
7
8
11
12
13

2.6 SELECT with DISTINCT and Subquery

Show list of team members who have no work assigned to them (opposite of previous query).

```
SELECT Id, First_Name || ' ' || Last_Name AS "Name"
FROM Team_Member
WHERE Id NOT IN
(SELECT DISTINCT Assigned_To FROM Work_Item);
```

```
data_format.function(query9b,
                     "Show list of team members with no work assigned to them",
                     "query", "#FF0000")
```

Table 24: Show list of team members with no work assigned to them

Id	Name
9	HerpaDerpa McDerpa
10	Joe Derp
14	Jo McQueryfiller

3. Inner Join

3.1 Inner Join 1

Inner Join Team_Member and Role tables via foreign key Team_Member.Role_id corresponding to Role.Id.

First name and last name are concatenated with the || operator as Concat() doesn't work in Sqlite. Using Alias (AS) for column headings and t for Team_Member and r for Role table aliases.

```
 -- no Concat() in sqlite, || = concat operator
SELECT t.Id AS 'Member Id',
t.First_Name || ' ' || t.Last_Name AS 'Full Name',
r.Name AS 'Project Role'
FROM Team_Member t
Inner Join Role r
ON t.Role_Id = r.Id
```

```
data_format.function(query10a_select_inner_join,
                     "Team Member Inner Joins Role to display member name and role",
                     "query", "#FF0000") # format
```

Table 25: Team Member Inner Joins Role to display member name and role

Member Id	Full Name	Project Role
1	Joe O'Regan	Programmer
2	Derp McDerp	Project Manager
3	Herpderp Derpderpenson	Tester
4	Herpa Derpderp	Artist
5	De Rpderp	3D Modeller
6	Pred Prehpred	Environment Modeller
7	Derpa Derpa	Concept Artist
8	Herpa Derpa	Shading Artist
9	HerpaDerpa McDerpa	Animator
10	Joe Derp	Programmer
11	Jon Herpaderp	Animator
12	Joblot O'Stuff	Tester
13	Joderp Herpderpenson	Artist
14	Jo McQueryfiller	Project Manager

3.2 Inner Join 2

Get Projects with no Project Manager.

3.2.1 Select Project Managers from Team Members

Get list of Project Managers from Team_Member table.

```
SELECT * FROM Team_Member WHERE Role_id = 1;
```

```
data_format.function(query10b_part1,
                     "Select Team Members who are Project Managers",
                     "query", "#FF0000")
```

Table 26: Select Team Members who are Project Managers

Id	First_Name	Last_Name	Email	Role_Id
2	Derp	McDerp	derpmcderp@daie.ca4	1
14	Jo	McQueryfiller	j.mcqueryfiller@daie.ca4	1

3.2.2 Show Project Manager and their projects

List of Project managers and the projects assigned to them. Joins Team_Member, Project_Team and Role tables.

```
SELECT First_Name || ' ' || Last_Name as "Member Name",
r.name AS "Role", r.Id AS "Role ID", pt.Project_Id AS "Project ID"
FROM Team_Member tm
INNER JOIN Project_Team pt
ON pt.Team_Member_Id = tm.Id
INNER JOIN Role r
ON tm.Role_Id = r.Id
WHERE tm.Role_Id IN
(SELECT Role_Id FROM Team_Member WHERE Role_id = 1);
```

```
data_format.function(query10b_part2,
                     "Project Managers assigned projects using Inner Join",
                     "query", "#FF0000")
```

Table 27: Project Managers assigned projects using Inner Join

Member Name	Role	Role ID	Project ID
Derp McDerp	Project Manager	1	1
Derp McDerp	Project Manager	1	2
Derp McDerp	Project Manager	1	3
Jo McQueryfiller	Project Manager	1	4
Jo McQueryfiller	Project Manager	1	6
Jo McQueryfiller	Project Manager	1	7

3.2.3 Get Projects with no Project Manager

List of projects with no manager assigned. Joins Project, Team_Member and Role tables.

```
Select p.Id AS "Project Id", p.Name AS "Projects Name"
FROM Project p
WHERE p.Id NOT IN
(SELECT DISTINCT pt.Project_Id
FROM Team_Member tm
INNER JOIN Project_Team pt
ON pt.Team_Member_Id = tm.Id
INNER JOIN Role r
ON tm.Role_Id = r.Id
WHERE tm.Role_Id IN
(SELECT Role_Id FROM Team_Member WHERE Role_id = 1));
```

```
data_format.function(query10b_part3,
                     "List of Projects with no Project Manager assigned",
                     "query", "#FF0000")
```

Table 28: List of Projects with no Project Manager assigned

Project Id	Projects Name
5	Christmas 2022 Project
8	Project Filler
9	Derp Project
10	Hmmm I Ran Out of Names

4. Subquery with SELECT

Nested select query

4.1 Subquery with SELECT

4.1.1 Subquery part 1: Inner query

Select the work items that are at least in Review (Review, or Done), i.e.with a status greater than 2.

```
SELECT * FROM Work_Item WHERE Status_Id > 2
```

```
data_format.function(query11_subquery_part_1,
                     "Get Work Items that are at least In Review",
                     "query", "#FF0000") # format query
```

Table 29: Get Work Items that are at least In Review

Id	Name	Status_Id	Assigned_To
2	Art Test Thingy	3	3
4	Art Concept Thingy	4	7
5	Art Shading Thingy	4	8
10	Blueprint Test	3	2
12	Query Model Thingy	3	11
13	Another Test	4	13

4.1.2 Subquery part 2: Outer query

Select work items that contain the string “test”.

```
SELECT * FROM Work_Item WHERE Name LIKE "%test%"
```

```
data_format.function(query12_subquery_part_2, "Work Items with string 'test'",
                     "query", "#FF0000")
```

Table 30: Work Items with string ’test’

Id	Name	Status_Id	Assigned_To
2	Art Test Thingy	3	3
7	3D Model Test Obj	2	3
10	Blueprint Test	3	2
11	Art Test	1	12
13	Another Test	4	13

4.1.3 Subquery with SELECT and IN

Select work items that contain the string "test" and have a Status_Id greater than 2.

```
SELECT * FROM Work_Item
WHERE Name LIKE "%test%"
AND Status_Id IN
(SELECT Status_Id FROM Work_Item
WHERE Status_Id > 2)
```

```
data_format.function(query13_subquery_with_select,
                     "Test Work Items that are in review or done", "query", "#FF0000")
```

Table 31: Test Work Items that are in review or done

Id	Name	Status_Id	Assigned_To
2	Art Test Thingy	3	3
10	Blueprint Test	3	2
13	Another Test	4	13

4.2 Subquery with SELECT and NOT IN

Select work items that contain the string "test" and have a Status_Id less than 3.

```
SELECT * FROM Work_Item
WHERE Name LIKE "%test%"
AND Status_Id NOT IN
(SELECT Status_Id FROM Work_Item
WHERE Status_Id > 2)
```

```
data_format.function(query13b_subquery_with_select_not_in,
                     "Test Work Items in To Do or In Progress", "query", "#FF0000")
```

Table 32: Test Work Items in To Do or In Progress

Id	Name	Status_Id	Assigned_To
7	3D Model Test Obj	2	3
11	Art Test	1	12

5. SELECT across a date range

5.1 Sorted delivery dates for comparison

Select delivery dates from Project table to compare query against. And order them to make it that much easier to find.

```
SELECT Delivery_Date AS "Project Due Dates" FROM Project
ORDER BY Delivery_Date
```

```
data_format.function(query14_check_dates,
                     "Ordered Project Due Dates",
                     "query", "#FF0000") # Format query data
```

Table 33: Ordered Project Due Dates

Project Due Dates
2022-12-14
2022-12-25
2023-01-11
2023-01-17
2023-01-20
2023-01-20
2023-01-24
2023-02-01
2023-03-01
2023-03-17

5.2 Select across a date range

Select projects with a delivery date in the range 16/01/2023 to 25/01/2023.

```
SELECT * FROM Project
WHERE Delivery_Date
BETWEEN "2023-01-16" AND "2023-01-25";
```

```
data_format.function(query15_select_across_date_range,
                     "Project Due between 16th and 25th of January 2023",
                     "query", "#FF0000")
```

Table 34: Project Due between 16th and 25th of January 2023

Id	Name	Delivery_Date
2	DAIE CA4	2023-01-20
3	New Project	2023-01-24
6	Date Range Project	2023-01-17
10	Hmmm I Ran Out of Names	2023-01-20

Disconnect Database

```
dbDisconnect(conn)
```