Database Design – Powerstation Scenario

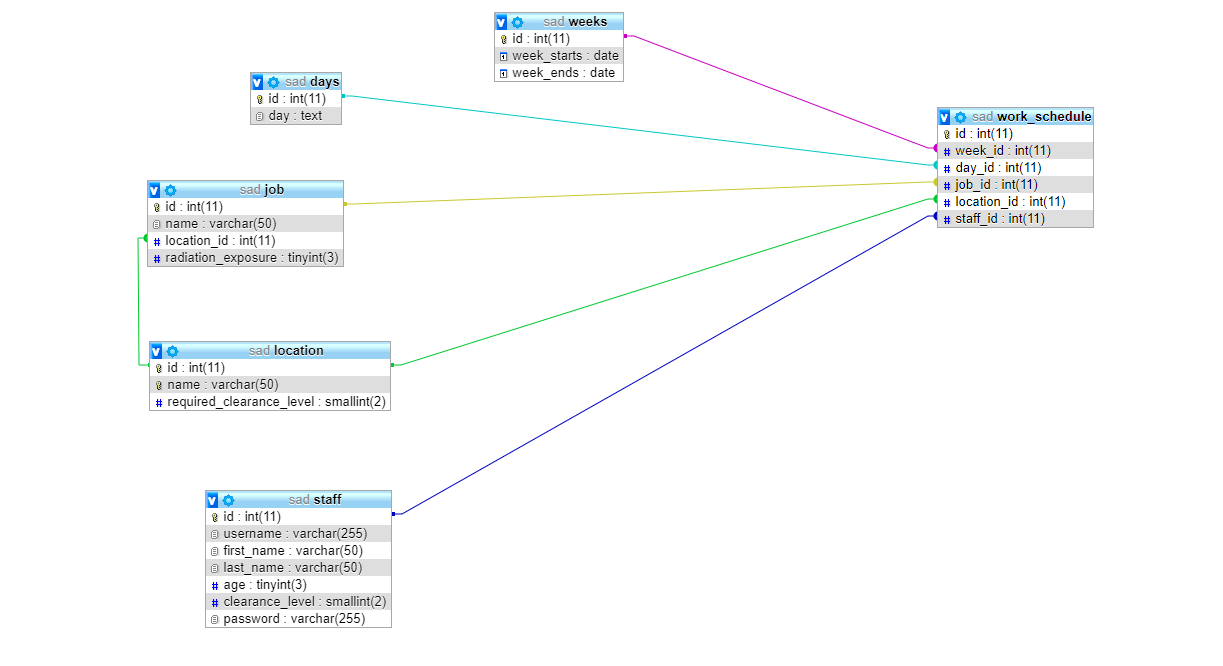
<https://dd-a02.knightstone.io> - working example

Before you start

When building JOINS remember to look for the foreign keys and tackle them one join at a time.

When building queries for the parts listed below you will be given the column\_name aliases to use. This is because the php code uses the aliases you will be proved with.

Prepare the database

You will need to create a database called "SAD" and import the supplied SQL files found in the **sql\_data** directory. Finally using the graph below you need to create the table called "work\_schedule" and build the connections depicted (foreign keys):  
  


Prepare the system

**Part 1:** *\_connect.php line 6* Make sure this file can connect to the SAD database.

Enable the login system

**Part 2:** *\_auth.php line 23* You need to use $username and $password from above to find a matching staff member.

Activate the admin system

**Part 3:** *utlis.php line 17* Starting from work\_schedule, use JOINS to gather the required data from:

* work\_schedule
* staff
* job
* location

You will need to use the php variables **$week** and **$day** in your WHERE clause to ensure you collect the appropriate data.

**Part 4:** *staff\_raditaion\_exposure.php line 37* Starting from work\_schedule, use JOINS to gather the required data from:

* staff
* job

The query will need to calculate (SUM) the total amount of radiation exposure staff members are exposed to. The query requires JOINS and it is a good idea to start from the work\_schedule table

You are supplied with the WHERE clause stored within a php variable **$staff\_check**, you will NEED to use this in you your query.

You will also need to GROUP BY staff.id

Ordering by exposure levels would be nice touch :)

**Part 5:** *delete\_work\_schedule.php line 11* Create a SQL query that can DELETE a work\_schedule entry based on the php variable **$id**

**Part 6:** *utils.php line 120* Create a query to SELECT the listed data from the staff table.

**Part 7:** *utils.php line 138* Create a query to SELECT the listed data from the job and location tables, you will need to use JOINS.

**Part 8:** *utils.php line 163* Create a query to SELECT the listed data from weeks.

**Part 9:** *utils.php line 178* Create a query to SELECT the listed data from days.

**Part 10:** *appoint\_job.php line 21* You need to create an INSERT query here to create a new work\_schedule based on the provided php variables:

* **$work\_week\_id**
* **$work\_day\_id**
* **$work\_staff\_id**

To acquire the location\_id you will need to create a nested SELECT query using the php variable **$work\_job\_id**.

**Part 11:** *edit\_job\_appointment.php line 19* You need to create an UPDATE query here to edit a work\_schedule based on the provided php variables:

* **$work\_week\_id**
* **$work\_day\_id**
* **$work\_staff\_id**

To acquire the location\_id you will need to create a nested SELECT query using the php variable **$work\_job\_id**.

Make sure you use the **$appointment\_id** from line 8 in your WHERE clause.

**Part 12:** *utils.php line 191* Create a query to SELECT a work\_schedule entry based on the id. Make sure to use the php variable **$id** in the WHERE clause.

**Part 13:** *staff\_schedule.php line 11* Make sure you use **$staff\_id** variable in your WHERE clause to SELECT staff member based on ID and select the required data.

**Part 14:** *staff\_schedule.php line 35* Starting from work\_schedule, use JOINS to gather the required data from:

* work\_schedule
* staff
* job
* location
* weeks
* days

You will need to use the php variable **$staff\_id** and **$day** in your WHERE clause to select WHERE staff.id = **'$staff\_id**'