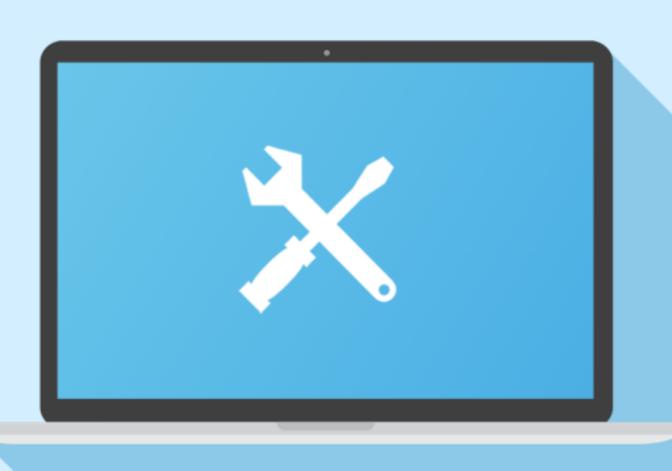


Section 10 - Exercises



#### Contents

This workbook will provide practice problems for you to work through. Each section will have a number of problems and some sections will have advanced problems. The *advanced* problems may require additional research and/or scripting. If you cannot complete these problems, do not be frustrated. We did not cover these topics and are meant as a challenge, rather than a test of your knowledge.

## (1) Section

Exercises

1. Create a ServiceNow community account

community.servicenow.com

2. Checkout the ServiceNow product documentation

docs.servicenow.com

3. Watch some video tutorials

youtube.com/user/servicenowdemo/featured

Exercises

1. Obtain a personal instance or demo instance

community.servicenow.com

2. Login to your personal instance

3. Navigate into, and play around with the Incident and Problem applications

Exercises

1. Configure the Incident list view layout to include the following fields

Number, Caller, Category, Priority

- 2. Create a filter for all incidents that were opened today and are unassigned
- 3. Save the filter as *Today's* unassigned incidents
- 4. Add the *reassignment count* field to the default incident form view

Exercises

1. Create a business rule which sets a problem's assignment group to the Administrators group, if a group is not selected on insert

- 2. Create a UI policy which will *hide* the *priority* field if *impact* and *urgency* are set to 3 Low
- 3. Activate the *Save* UI action only for the *Incident* form view

**Advanced** Exercises

#### Advanced

1. Create a problem task record if a problem record is saved with a priority of 1 - Critical, and relate it to the problem record

2. Create a client script which informs the user that a problem with a priority of 1 - Critical will create a problem task

Exercises

1. Create a car table with the following fields:

```
make (choice)
model (string)
release_date (date)
cost (currency)
```

2. Create 5 choices for the make field

3. Create 2 car records

4. Configure the *form* to your liking and add the *created* field to the form

Exercises

1. Create users, groups, roles and access controls for the following use case

3 users belong to the *Incident Team* group. This group has the *incident\_team* role which gives them the ability to add and modify the caller and contact type fields on an incident

2. Create a new *access control* rule which gives permissions to change the *last name* of any *user record* 

# (7) Section

Exercises

- 1. Create an incident record, incident A
- 2. Create another incident record, incident B, and link incident A to incident B
- 3. Create a new *emergency change* request
- 4. From incident A, create a new problem
- 5. Link the *emergency change* request to the problem record
- 6. Create a new iPhone X catalog item
- 7. Create a new *knowledge article* and publish it

Exercises

- 1. Create a new *workflow* for the catalog item you created in section 7
- 2. Add the *sc\_req\_item* table to the workflow
- 3. Add a *condition* so that this workflow is only ran when this item is *selected*
- 4. Add an *approval* and *notification* to the workflow
- 5. Request the item and view the notification in the *emails* log

Exercises

- 1. Create the Rotten Potatoes application in your personal instance
- 2. Add a *Owned* true/false field to the *video game* table
- 3. If a review is given 1 or 2 stars, require the comments field
- 4. Add an *Image* field to the *video* game table

#### Advanced

9 Section

**Advanced** Exercises

1. Create a workflow for the video game table which will execute only for video games that have not yet been released, and send a notification to a review group once the video game release date is 1 week after today's date