From the createTables.sql file, we see that there are four tables each with their own attributes, as such:

Table 1. Not in any NF form.

Department	Employee	Assigned	Project
deptID	empID	empID	projID
deptName	empName	projID	title
location	job	role	budget
	deptID		funds
	salary		

However, to get it into 1NF, we must make sure each attribute contains only atomic values. The department table's location attribute is not atomic because it is composed of street address, city, region (province or territory), and postal code. The employee table's empName attribute is not atomic because it contains first name, optionally a middle name, and last name. To satisfy the 1NF criteria, we must split these into:

Table 2. In 1NF form.

Department	Employee	Assigned	Project
depID	empID	empID	projID
depName	firstName	projID	title
address	middleName	role	budget
city	lastName		funds
region	job		
postalCode	deptID		
	salary		

Then, to get it into 2NF, the tables and attributes must satisfy 1NF, and additionally, there must be no partial dependencies. To convert the tables to 2NF form we must determine candidate keys and assure that there are no partial dependencies, meaning that an attribute does not rely only on a part of the entire candidate key. We can do this for each table:

1. For the Department table, we choose the candidate key to be { depID, address, city, region, postalCode } to be the candidate keys. The reason every location attribute is being included is because two very small towns can have the same postal code, but also be in a different region since the Northwest territories and Nunavut could share a postal code. Additionally, there could be two identical street addresses in two different cities. Furthermore, two cities with the same name can exist in different regions. The only non-candidate key attribute in this table is depName, but it only relies on depID, thus this is a partial dependency and the table must be split.

2. For the Employee table, the candidate keys are { empID, deptID } because an employee can be in many departments. However, firstName, middleName, lastName, job, and salary rely only on the empID part of the candidate keys. Meaning there is a partial dependency.
3. For the Assigned table, an employee can have many projects and many roles, so the candidate keys are { empID, projID, role }. There are no partial dependencies.
4. For the Project table, the candidate key is { projID }. The title, budget, and funds attributes rely on only projID, meaning there are no partial dependencies.

Therefore, we can modify the tables to the following (with primary keys bolded):

Table 3. In 2NF form.

Department_Main	Department_Location	Employee_Main	Employee_Department	Project_Main	Assigned_Main
deptID	**deptID**	**empID**	**empID**	**projID**	**empID**
deptMain	**address**	firstName	deptID	title	projID
	city	middleName		budget	role
	region	lastName		funds	
	postalCode	job			
		salary			

For 3NF, we must assure that there are no partial dependencies, and that is also follows 2NF. This means that we must assure that every attribute depends on every candidate key attribute and on nothing else. This is true for every attribute in every table in Table 3, so no changes are needed.

For BCNF, we must assure that the tables are already 3NF, and that we should only be able to derive non-candidate keys from candidate keys, and not the other way around. This is already the case with Table 3, so no changes are needed.

When adding foreign keys, the foreign keys for each table are the following:

1. The Department_Main table has no foreign keys.
2. The Department_Location references the Department_Main table's deptID attribute because for a department to have a location, it must exist in the main department table.
3. The Employee_Main table has no foreign keys.
4. The Employee_Department table references Employee_Main table's empID attribute because for a department to have an employee, or vice versa, the employee must exist.
5. The Project_Main table has no foreign keys.
6. The Assigned_Main table references Employee_Main table's empID attribute and Project_Main table's projID attribute, because for an employee to be assigned, the employee and project must exist.