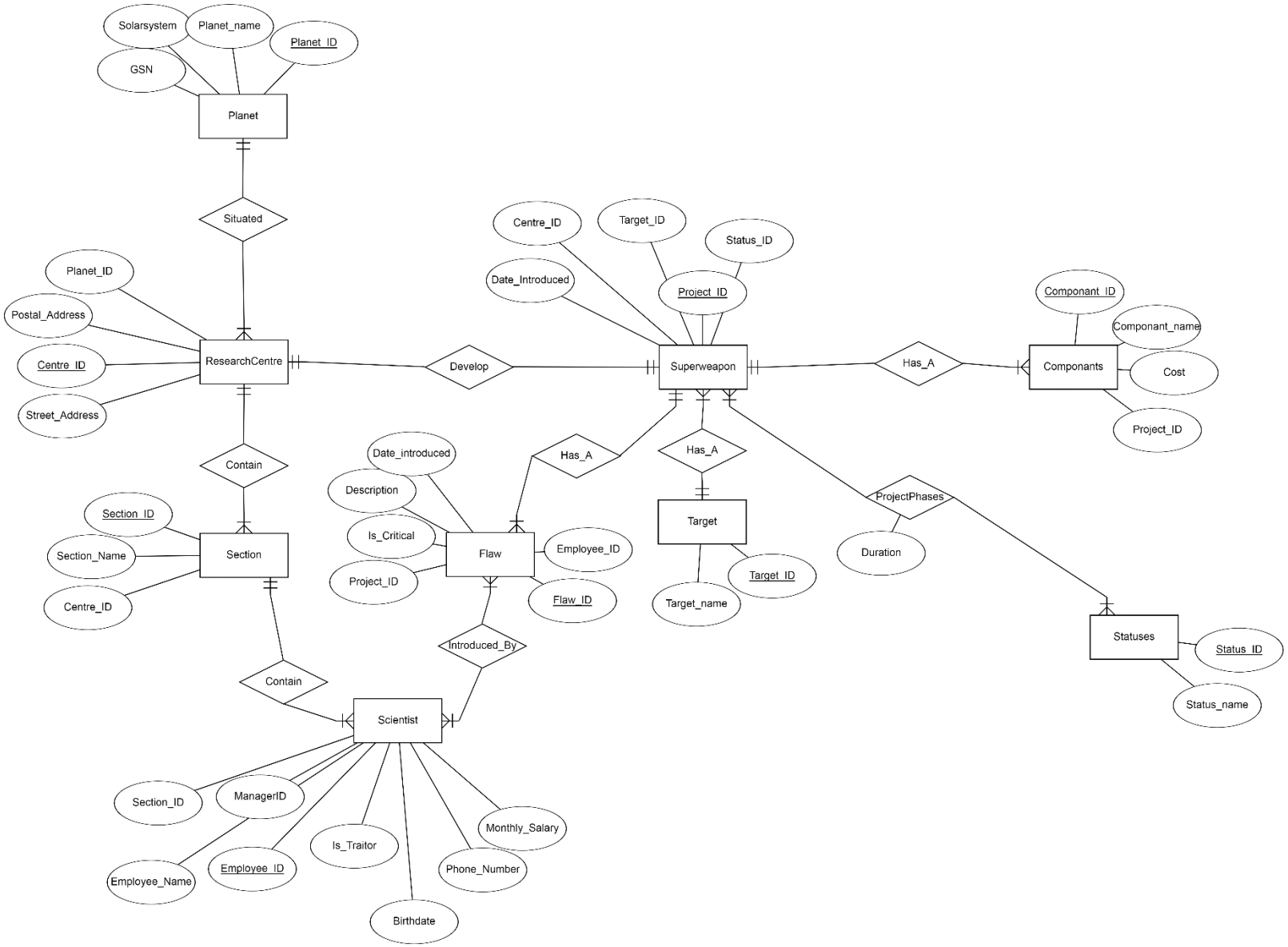
**CITS1402 Relational Database Management Systems**
**Summer semester, Jan 2019**
**Assignment**

Name: Paul O'Sullivan
Student no.: 21492318
Total pages (not including cover or diagram) pages: 5

Extension exercise can be found from line 76 – 101 in Empire.SQL

## 1. Entity-Relationship Diagram for Empire database



## 2. Design Discussion

**Planet:** This table was created to ensure that 3NF could be maintained throughout the database. Attributes for this table include. Planet_ID (PK), Planet_Name, GSN (Galactic sector number) and Solarsystem. 3NF holds in this table as it can be assumed that there is only 1 galaxy and 1 galactic sector number in the universe that the empire wants to control. There are however more than one planet. This table shares a one to many relationship with research centre whereby there can be many research centres on a planet but only one planet per research centre.

**ResearchCentre**: This table is necessary to show at which research centre a super weapon is made, and also to show what sections are contained in a research centre. Every Planet must have at least one research centre. Attributes for this table include Centre_ID(PK) Postal address, Street Address and Planet_ID(FK)

**Section**: Section's relationships with other tables show: the different sections that can be found in a research centre and also show which scientists work at in each section. Attributes for this entity include: Section_ID(PK), Section_Name, Centre_ID(FK)

**Scientist**: Scientist entity allows for one to observe which scientist(s) introduce a flaw, and which section they work in. We will assume that Scientists who are not chief scientists will have the same Section_ID as their chief scientist. We will also assume that HR records constitute the aggregate details of scientist attributes. Attributes for this entity include Empoyee_ID(PK), Employee_Name, Is_Traitor, ManagerID, Birthdate, Monthly Salary, Phone Number, Section_ID(FK)

**Flaw**: The Flaw entity allows for empire staff to observe which flaws are specific to which superweapon (Relationship), which scientist(s) introduced the flaw(s) (relationship), the description of each flaw (attribute), whether or not the flaw is a critical flaw or a major flaw (attribute) and also the date the flaw was introduced (attribute). A flaw can be introduced by many scientists however for this to be the case Is_Critical must be TRUE, If Is_Critical is FALSE (aka, a major flaw), only one scientist has introduced this flaw as being a major flaw. One scientist may also introduce a critical flaw single headedly. Foreign keys for this entity include: Project_ID and Employee_ID.

**Superweapon:** The Superweapon entity allows for users to observe many relationships. The primary key for this entity is Project_ID. Users can observe what target type a superweapon has(relationship), the current and past statuses of each superweapon(relationship), what componants a superweapon has(relationship), which flaws relate to each superweapon(relationship) and at which research centre a superweapon is built (relationship). Users can also see what date the superweapon was introduced (Attribute). Foreign keys are necessary for the above relationships to take place. These include: Target_ID, Status_ID , Centre_ID and Componant_ID.

**Target:** Assigns an ID to each target type to ensure 3NF holds**.** Primary key: Target_ID Attributes: Target_Name . Shares a one to many relationship with superweapon. A superweapon can only be built specifically for one target.

**Statuses:** Records and assigns an ID to each type of status. Primary key: Status_ID Attribute: Status_Name. This entity has a many to many relationship with Superweapon as a Superweapon has to go through each status phase in order to arrive at the last status, thus a superweapon will have at least one phase and can have multiple phases depending on the progress of the project.

**ProjectPhases:** This is a relationship table and records the current and past statuses of a superweapon. It also records the duration of time it took to complete a phase. If the phase is still underway then the duration of the phase should be recorded as NULL: Project_ID and Status_ID are a composite key for this table. Assume admin only records a phase duration when the phase has been completed. Admin will enter in the number of months as an INT to record how long that phase took for each project.

**Components:** Assigns an ID to each target type to ensure 3NF holds**.** Primary key: Component_ID Attributes: Component_Name, Cost. A component is specific to only one type of superweapon, thus will have a one to many relationship with superweapon

Assume that whoever manages this database will continuously take backups to the Empire via cloud storage

# 3.a Schemas

**Guide: 3NF =** Every non-prime attribute is functionally dependant on the primary key and there exists no transitive dependencies between attributes in the schema.

**Planet (Planet_ID,Solarsystem,Planet_Name,GSN).**

- One assumption when creating this database was that there is only one solar system (Milkyway) and one galactic sector number (1101). Thus, this schema is in 3NF as every non-prime attribute is functionally dependant on the primary key "Planet_ID"

**ResearchCentre (Centre_ID, Street_address, Postal_address,Planet_ID)**

- One assumption when creating this database was that there is only one street address and one postal address for each Research Centre. Thus, this schema is in 3NF where non prime attributes are functionally dependent on the primary key being "Centre_ID". Planet_ID is a foreign key in this entity due to the fact that a research centre is situated on a planet and will therefore inherit the planet's attributes.

**Scientist(Employee_ID,Employee_Name,Monthly_salary,Phone_number,Birthdate,Is_Traitor,ManagerID,Section_ID)**

- This schema is in 3NF. All non-prime attributes are functionally dependant on Employee_ID (primary key). Section_ID is a foreign key in this schema due to the fact that employees work inside of a section and therefore inherit the parent (Section) attributes. If a scientist is a chief scientist then "ManagerID" will be recorded as NULL.

**Superweapon(Project_ID,Date_of_commencement,Target_ID,Centre_ID)**

- This schema is in 3NF. Project_ID is the primary key and will determine all other attributes. Centre_ID is also a foreign key as a superweapon can be considered a child of a research centre due to the fact that superweapon is built inside of a research centre and will therefore inherit the attributes of research centre.

**Section(Section_ID,Section_name,Centre_ID)**

- This schema is in 3NF. Its primary key is Section_ID. Centre_ID is a foreign key due to the fact that a section operates inside of a research centre and therefore, a section (child) will inherit these attributes from research centre (parent).

**Target(Target_ID,Target_Name).**

- This schema is in 3NF. Target_ID is the primary key and there are no foreign keys.

**Statuses(Status_ID,Status_name)**

- This table is in 3NF. Status_ID is the primary key and there are no foreign keys.

**ProjectPhases(Status_ID, Duration, Project_ID, CurrentPhase)**

- This is a relationship table. Both, Status_ID and Project_ID are a composite key.

**Componants(Componant_ID,Componant_Name,Cost,Project_ID)**

- This table is in 3NF, Componant_ID is the primary key. Project_ID is a foreign key because a component is specific to only one superweapon.

**Flaw(Flaw_ID,Description,Date_Introduced,Is_Critical,Employee_ID,Project_ID)**

- This table is in 3NF. Flaw_ID is the primary key and functionally determines all other attributes. Project_ID is a foreign key due to the fact that a superweapon can have multiple flaws, while a flaw is specific to each superweapon. Employee_ID is also a foreign key because more than one scientist can introduce a flaw, while each flaw is specific to one superweapon. Having foreign keys in flaw allows for more convenient tables to be created. We do not need to create duplicate superweapon entries in order to show what each type of flaw they have.

## *3.B Data types*

Used datatypes: **INT, BIGINT, VARCHAR, BOOLEAN**

**Widely used datatypes throughout the database:**

In this database and due to the fact that every primary key is classified as "ID", I decided to use a combination of **INT and BIGINT** datatype constrained as unique primary keys. This was used for simplicity and also allowed for 2NF, and ultimately enabled 3NF.

For any naming attributes and descriptions, eg Employee_Name, Section_Name or Flaw Description a **VARCHAR** data type was used. For name attributes, the length of this type was set at 50 characters while 200 characters was allowed for descriptions and address attributes as these can often be longer in length.

**BOOLEAN** was used for attributes that required a TRUE or FALSE value. Examples include: Is_Traitor and Is_Critical. Given that there can only be two types of flaws, **BOOLEAN** was appropriate for IS_Critical due to the fact that a flaw can only be a major flaw or a critical flaw. Therefore, if Is_critical is false, the flaw must be a major flaw.

**Specific or Non trivial decision datatypes:**

Used **BIGINT** for Scientist_ID due to the fact that there could be a very large number of scientists spread out throughout the Milkyway galaxy.

Used **BIGINT** for Planet_ID due to the fact that the total number of planets in the milykway falls close to the largest number that a standard INT can hold and there could still be planets waiting to be discovered.

Used **BIGINT** for Centre_ID and Project_ID due to the fact that there could be an extremely large number of research centres in the Milkyway galaxy, each of which developing a superweapon

Used **VARCHAR** for phone number attribute due to the fact that phone numbers often appear with symbols, eg: () or + .

**Relational Algebra Expressions:** All other queries containED GROUP BY or UPDATE

**Diagram 1 Question 1.**

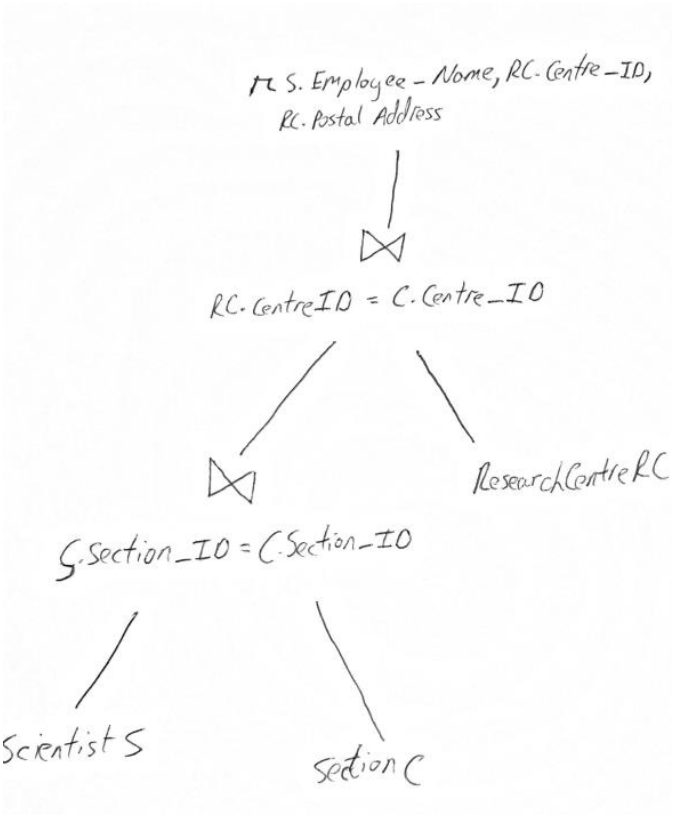



**Diagram 2 Question 2.**