**University of Strathclyde**

**Emanuel Felipe dos Santos – cyb14138**

**I) Final EER model solution**

**a) and b)**

**Entities Table.**

Employee (employee number, name, address, telephone)

\* Employees of the properties

Site (site number)

\*Sites of the properties

Unit (unit number, area, zoning)

\*Units of the properties

Company (company\_id, company name)

\* The companies responsible for the maintenance

Job(job\_number, description)

Emergency(job\_number)

Routine(job\_number)

Preventative(job\_number)

\*Types of jobs during a maintenance

Maintenance(maintenance\_id, start\_date, end\_date, month\_payment)

**c)**

**Relationships**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Entities** | **Degree** | **Optionality** |
| Manages | Employee, Site | N:N | Obligatory on Both |
| Supervises | Employee, Employee | 1:N | Obligatory on Both |
| Occurs | Site, Unit | N:N | Obligaroty on Both |
| UM | Unit, Maintenance | 1:N | Obligatory on Both |
| CM | Company, Maintenance | 1:N | Obligatory on Both |
| JM | Job, Maintenance | 1:N | Obligatory on Both |

Manages (employee\_number, site\_number)

employee\_number references employee(employee\_number)

site\_number references site(site\_number)

Supervises(supervisor\_number, employee\_number)

supervisor\_number references employee(employee\_number)

employee\_number references employee(employee\_number)

Occurs(site\_number, unit\_number)

site\_number references site(site\_number)

unit\_number references unit(unit\_number)

UM (unit\_number, maintenance\_id)

unit\_number references unit(unit\_number)

maintenance\_id references maintenance(maintenance\_id)

CM (company\_id,maintenance\_id)

company\_id references company(company\_id)

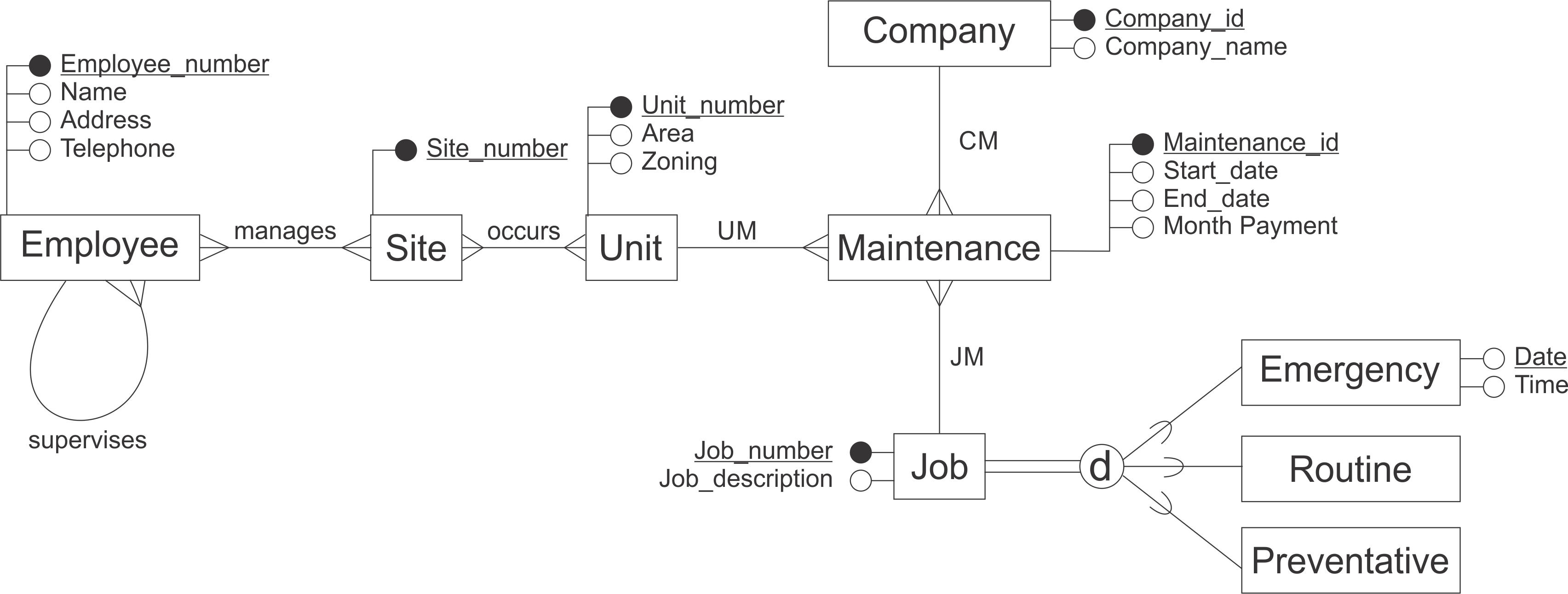
maintenance\_id references maintenance(maintenance\_id)

JM(job\_number, maintenance\_id)

job\_number references job(job\_number)

maintenance\_id references maintenance(maintenance\_id)

d) Enhanced Diagram



e)

**Assumptions**

* For the entity Employee, the key is the attribute employee\_number which is unique. Also, the employee\_number as an identifier to supervisor.
* The Supervisor relationship relates two employees, which one is the supervisor id and the other one is the employee supervised.
* For the entity Unit, the key is the attribute unit\_number, which is unique. The zoning is given as an attribute instead of one entity for each type of zoning.
* For the Company, the key is the attribute company\_id which is unique. Also for each company was given a name.
* The entities Unit, Company and Job are related by a ternary relationship called maintenance.
* For the maintenance relationship was created an entity relationship which also stores the maintenance\_id, start and end dates of the maintenance, the month payment.
* For the relationship between Unit and Maintenance was created the relationship UM, which stores the identifier of each entity.
* For the relationship between Job and Maintenance was created the relationship JM, which stores the identifier of each entity
* For the relationship between Company and Maintenance was created the relationship CM, which stores the identifier of each entity
* For the entity Job, the key is the attribute job\_number, which is unique. Also is stored the description of the job.
* For each type of job was created an sub-entity as well, which are Routine, Preventative and Emergency. And they use the identifier of the sup-entity.

**III) Create table statements**

**EMPLOYEE**

CREATE TABLE cw\_employee (employee\_number INT(10) NOT NULL, name VARCHAR(255) NOT NULL, address VARCHAR (255) NOT NULL, telephone VARCHAR(255) NOT NULL, PRIMARY KEY (employee\_number))

**SITE**

CREATE TABLE cw\_site(site\_number INT(10) NOT NULL, PRIMARY KEY (site\_number))

**UNIT**

CREATE TABLE cw\_unit (unit\_number INT(10) NOT NULL, area INT(255) NOT NULL, zoning VARCHAR(255) NOT NULL, PRIMARY KEY (unit\_number))

**COMPANY**

CREATE TABLE cw\_company (company\_id INT(10) NOT NULL, company\_name VARCHAR(255) NOT NULL, PRIMARY KEY (company\_id))

**JOB**

CREATE TABLE cw\_job (job\_number INT(10) NOT NULL, job\_description VARCHAR(255) NOT NULL, PRIMARY KEY (job\_number))

**MAINTENANCE**

CREATE TABLE cw\_maintenance (maintenance\_id INT(10) NOT NULL, start\_date DATE NOT NULL, end\_date DATE NOT NULL, month\_payment INT(255) NOT NULL, PRIMARY KEY (maintenance\_id))

**MANAGES**

CREATE TABLE cw\_manages (employee\_number INT(10) NOT NULL, site\_number INT(10) NOT NULL, FOREIGN KEY (employee\_number) REFERENCES cw\_employee(employee\_number), FOREIGN KEY (site\_number) REFERENCES cw\_site(site\_number) )

**OCCURS**

CREATE TABLE cw\_occurs (site\_number INT(10) NOT NULL, unit\_number INT(10) NOT NULL, FOREIGN KEY (site\_number) REFERENCES cw\_site(site\_number), FOREIGN KEY (unit\_number) REFERENCES cw\_unit(unit\_number) )

**SUPERVISES**

CREATE TABLE cw\_supervises (supervisor\_number INT(10) NOT NULL, employee\_number INT(10) NOT NULL, FOREIGN KEY (supervisor\_number) REFERENCES cw\_employee(employee\_number), FOREIGN KEY (employee\_number) REFERENCES cw\_employee(employee\_number))

**UM**

CREATE TABLE cw\_um (unit\_number INT(10) NOT NULL, maintenance\_id INT(10) NOT NULL, FOREIGN KEY (unit\_number) REFERENCES cw\_unit(unit\_number), FOREIGN KEY (maintenance\_id) REFERENCES cw\_maintenance(maintenance\_id))

**CM**

CREATE TABLE cw\_cm (company\_id INT(10) NOT NULL, maintenance\_id INT(10) NOT NULL, FOREIGN KEY (company\_id) REFERENCES cw\_company(company\_id), FOREIGN KEY (maintenance\_id) REFERENCES cw\_maintenance(maintenance\_id))

**JM**

CREATE TABLE cw\_jm (job\_number INT(10) NOT NULL, maintenance\_id INT(10) NOT NULL, FOREIGN KEY (job\_number) REFERENCES cw\_job(job\_number), FOREIGN KEY (maintenance\_id) REFERENCES cw\_maintenance(maintenance\_id))

**ROUTINE**

CREATE TABLE cw\_routine (job\_number INT(10) NOT NULL, FOREIGN KEY (job\_number) REFERENCES cw\_job(job\_number))

**PREVENTATIVE**

CREATE TABLE cw\_preventative (job\_number INT(10) NOT NULL, FOREIGN KEY (job\_number) REFERENCES cw\_job(job\_number))

**EMERGENCY**

CREATE TABLE cw\_emergency (job\_number INT(10) NOT NULL, FOREIGN KEY (job\_number) REFERENCES cw\_job(job\_number), date DATE NOT NULL, time TIME NOT NULL)

**IV) Insert Statements**

**Employee**

INSERT INTO `cyb14138`.`cw\_employee` (`employee\_number`, `name`, `address`, `telephone`) VALUES ('1', 'Emanuel', '49 Rottenrow East C10-4', '+4407474283542'), ('2', 'Lucas', '49 Rottenrow East C10-3', '+4407474231342'), ('3', 'Ivo', '49 Rottenrow East C10-2', '+4407474986254 '), ('4', 'Carlos', '49 Rottenrow East C10-1', '+4407474332242'), ('5', 'Arthur', '47 Rottenrow East D9-1', '+4407474553142'), ('6', 'Andre', '47 Rottenrow East D9-1', '+4407474312432'), ('7', 'Gabriel', '47 Rottenrow East D9-1', '+4407424453142'), ('8', 'Claudio', '47 Rottenrow East D9-1', '+4407474897564');

**Site**

INSERT INTO `cyb14138`.`cw\_site` (`site\_number`) VALUES ('1'), ('2'), ('3'), ('4'), ('5'), ('6'), ('7'), ('8'), ('9'), ('10');

**Supervises**

INSERT INTO `cyb14138`.`cw\_supervises` (`supervisor\_number`, `employee\_number`) VALUES ('1', '2'), ('1', '3'), ('1', '4'), ('5', '6'), ('5', '7'), ('5', '8'), ('4', '7'), ('3', '8'), ('6', '3'), ('5', '3');

**Unit**

INSERT INTO `cyb14138`.`cw\_unit` (`unit\_number`, `area`, `zoning`) VALUES ('1', '532', ''), ('2', '5115', ''), ('3', '18550', ''), ('4', '22561', ''), ('5', '864', ''), ('6', '7050', ''), ('7', '6330', ''), ('8', '23163', ''), ('9', '65589', ''), ('10', '24411', '');

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Distribution Center' WHERE `cw\_unit`.`unit\_number` =1;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Light Industry' WHERE `cw\_unit`.`unit\_number` =2;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Hazardous Storage' WHERE `cw\_unit`.`unit\_number` =3;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Heavy Industry' WHERE `cw\_unit`.`unit\_number` =4;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Light Industry' WHERE `cw\_unit`.`unit\_number` =5;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Light Industry' WHERE `cw\_unit`.`unit\_number` =6;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Distribution Center' WHERE `cw\_unit`.`unit\_number` =7;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Hazardous Storage' WHERE `cw\_unit`.`unit\_number` =8;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Heavy Industry' WHERE `cw\_unit`.`unit\_number` =9;

UPDATE `cyb14138`.`cw\_unit` SET `zoning` = 'Hazardous Storage' WHERE `cw\_unit`.`unit\_number` =10;

**Manages**

INSERT INTO `cyb14138`.`cw\_manages` (`employee\_number`, `site\_number`) VALUES ('1', '2'), ('1', '4'), ('1', '7'), ('5', '1'), ('5', '6'), ('7', '3'), ('8', '5'), ('8', '9');

**Occurs**

INSERT INTO `cyb14138`.`cw\_occurs` (`site\_number`, `unit\_number`) VALUES ('1', '1'), ('1', '8'), ('2', '7'), ('9', '3'), ('10', '4'), ('9', '6'), ('2', '5'), ('2', '10'), ('2', '2'), ('2', '9');

**Company**

INSERT INTO `cyb14138`.`cw\_company` (`company\_id`, `company\_name`) VALUES ('1', 'ChoiceRepair'), ('2', 'ServicesHere'), ('3', 'HoustonFix'), ('4', 'Repairtime'), ('5', 'MonsterRepair'), ('6', 'VacFix'), ('7', 'ParkFix Ltda.'), ('8', 'Solve Me');

**Job**

INSERT INTO `cyb14138`.`cw\_job` (`job\_number`, `job\_description`) VALUES ('1', 'Painting'), ('2', 'Repair windows'), ('3', 'Repair toilet'), ('4', 'Moving furniture'), ('5', 'Clean lamps'), ('6', 'Test fire alarm'), ('7', 'Check roof and ceiliing'), ('8', 'Repair lift'), ('9', 'Disinfect drains'), ('10', 'Empty bins');

**Maintenance**

INSERT INTO `cyb14138`.`cw\_maintenance` (`maintenance\_id`, `start\_date`, `end\_date`, `month\_payment`) VALUES ('1', '2015-03-04', '2015-05-04', '350'), ('2', '2015-02-01', '2015-03-01', '600'), ('3', '2015-05-06', '2015-05-28', '500'), ('4', '2015-05-01', '2015-07-01', '600'), ('5', '2015-02-16', '2015-08-20', '580'), ('6', '2015-07-01', '2015-07-30', '450'), ('7', '2015-10-21', '2016-03-23', '600'), ('8', '2015-08-07', '2015-12-17', '900'), ('9', '2015-05-01', '2015-06-01', '650'), ('10', '2015-02-02', '2015-07-16', '600'), ('11', '2015-06-12', '2015-08-21', '800'), ('12', '2015-02-09', '2015-08-21', '1200'), ('13', '2015-05-30', '2015-07-16', '600'), ('14', '2015-06-18', '2015-09-22', '460'), ('15', '2015-07-15', '2015-09-28', '900'), ('16', '2015-07-03', '2015-10-23', '1000'), ('17', '2015-03-16', '2015-08-20', '800'), ('18', '2015-05-01', '2015-08-20', '380'), ('19', '2015-01-13', '2016-01-22', '500'), ('20', '2014-09-09', '2015-11-20', '360'), ('21', '2015-01-04', '2015-08-20', '700'), ('22', '2015-06-10', '2015-10-16', '350'), ('23', '2014-11-18', '2015-02-11', '400'), ('24', '2014-11-10', '2015-10-16', '350');

**UM(Unit - Maintenance)**

INSERT INTO `cyb14138`.`cw\_um` (`unit\_number`, `maintenance\_id`) VALUES ('1', '1'), ('2', '2'), ('5', '3'), ('3', '4'), ('10', '5'), ('2', '6'), ('10', '7'), ('8', '8'), ('4', '9'), ('8', '10'), ('9', '11'), ('3', '12'), ('9', '13'), ('4', '14'), ('2', '15'), ('10', '16'), ('9', '17'), ('8', '18'), ('7', '19'), ('8', '20'), ('7', '21'), ('4', '22'), ('3', '23'), ('6', '24');

**CM(Company - Maintenance)**

INSERT INTO `cyb14138`.`cw\_cm` (`company\_id`, `maintenance\_id`) VALUES ('1', '1'), ('1', '2'), ('1', '3'), ('1', '4'), ('3', '5'), ('3', '6'), ('4', '7'), ('4', '8'), ('4', '9'), ('5', '10'), ('5', '11'), ('6', '12'), ('6', '13'), ('6', '14'), ('6', '15'), ('7', '16'), ('7', '17'), ('7', '18'), ('8', '19'), ('8', '20'), ('8', '21'), ('8', '22'), ('8', '23'), ('8', '24');

**JM (Job – Maintenance)**

INSERT INTO `cyb14138`.`cw\_jm` (`job\_number`, `maintenance\_id`) VALUES ('1', '1'), ('2', '2'), ('1', '3'), ('5', '4'), ('5', '5'), ('5', '6'), ('7', '7'), ('6', '8'), ('4', '9'), ('8', '10'), ('8', '11'), ('6', '12'), ('3', '13'), ('3', '14'), ('8', '15'), ('7', '16'), ('9', '17'), ('10', '18'), ('10', '19'), ('10', '20'), ('9', '21'), ('10', '22'), ('10', '23'), ('10', '24');

**Routine**

INSERT INTO `cyb14138`.`cw\_routine` (`job\_number`) VALUES ('5'), ('7'), ('10'), ('9');

**Preventative**

INSERT INTO `cyb14138`.`cw\_preventative` (`job\_number`) VALUES ('8'), ('1'), ('3'), ('8'), ('6');

**Emergency**

INSERT INTO `cyb14138`.`cw\_emergency` (`job\_number`, `date`, `time`) VALUES ('6', '2015-04-07', '12:00'), ('8', '2015-04-15', '07:00'), ('6', '2015-03-16', '05:00'), ('1', '2015-05-23', '18:00');

**V) Queries, Queries output and explanation**

* Since the output of the query depends on the value(s) entered by the user, I will give for each query an example of output with a predefined input.

**Query 1**

“Select all the employees who supervises units area bigger than 500 and also show the information about the unit.”

**Input required:** “Area”.

**Input given by the user:** 500

SELECT DISTINCT E.employee\_number, E.name, E.address, E.telephone, U.unit\_number, U.zoning , U.area FROM cw\_employee E, cw\_manages M, cw\_site S, cw\_occurs O, cw\_unit U WHERE E.employee\_number = M.employee\_number AND M.site\_number = S.site\_number AND S.site\_number = O.site\_number AND O.unit\_number = U.unit\_number AND U.area > 500 GROUP BY U.unit\_number ORDER BY E.employee\_number

**Query 1 – Output (cw\_employee.odt)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| employee\_number | name | address | telephone | unit\_number | zoning | area |
| 1 | Emanuel | 49 Rottenrow East C10-4 | +4407474283542 | 7 | Distribution Center | 6330 |
| 1 | Emanuel | 49 Rottenrow East C10-4 | +4407474283542 | 5 | Light Industry | 864 |
| 1 | Emanuel | 49 Rottenrow East C10-4 | +4407474283542 | 10 | Hazardous Storage | 24411 |
| 1 | Emanuel | 49 Rottenrow East C10-4 | +4407474283542 | 2 | Light Industry | 5115 |
| 1 | Emanuel | 49 Rottenrow East C10-4 | +4407474283542 | 9 | Heavy Industry | 65589 |
| 5 | Arthur | 47 Rottenrow East D9-1 | +4407474553142 | 1 | Distribution Center | 532 |
| 5 | Arthur | 47 Rottenrow East D9-1 | +4407474553142 | 8 | Hazardous Storage | 23163 |
| 8 | Claudio | 47 Rottenrow East D9-1 | +4407474897564 | 3 | Hazardous Storage | 18550 |
| 8 | Claudio | 47 Rottenrow East D9-1 | +4407474897564 | 6 | Light Industry | 7050 |

**Query 2**

“Select all the jobs of the company number 6 and have the month payment bigger than the average”.

**Input required:** “Company ID”.

**Input given by the user:** 6

SELECT C.company\_id, C.company\_name, J.job\_number, J.job\_description, M.start\_date, M.end\_date, M.month\_payment FROM cw\_job J, cw\_maintenance M, cw\_company C, cw\_jm JM, cw\_cm CM WHERE JM.job\_number = J.job\_number AND M.maintenance\_id = JM.maintenance\_id AND CM.company\_id = C.company\_id AND CM.maintenance\_id = M.maintenance\_id AND C.company\_id = 6 AND M.month\_payment > (SELECT AVG(month\_payment) FROM cw\_maintenance)

**Query 2 – Output (cw\_job.odt)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| company\_id | company\_name | job\_number | job\_description | start\_date | end\_date | month\_payment |
| 6 | VacFix | 6 | Test fire alarm | 2015-02-09 | 2015-08-21 | 1200 |
| 6 | VacFix | 8 | Repair lift | 2015-07-15 | 2015-09-28 | 900 |

**Query 3**

“Select all the jobs in progress on the year 2015 and month 05 and have the month payment bigger than the average month payment inside the company itself.”

**First input required:** “Year”.

**First input given by the user:** 2015

**Second input required:** “Month”.

**Second input given by the user:** 05

SELECT C1.company\_id ,C1.company\_name, J1.job\_number, J1.job\_description, M1.start\_date, M1.end\_date, M1.month\_payment FROM cw\_company C1, cw\_maintenance M1, cw\_cm CM1, cw\_job J1, cw\_jm JM1 WHERE C1.company\_id = CM1.company\_id AND CM1.maintenance\_id = M1.maintenance\_id AND J1.job\_number = JM1.job\_number AND JM1.maintenance\_id = M1.maintenance\_id AND M1.start\_date <= '2015-05-\_\_' AND M1.end\_date >= '2015-05-\_\_' AND M1.month\_payment >(SELECT avg(M2.month\_payment) FROM cw\_company C2, cw\_maintenance M2, cw\_cm CM2, cw\_job J2, cw\_jm JM2 WHERE C2.company\_id = CM2.company\_id AND CM2.maintenance\_id = M2.maintenance\_id AND J2.job\_number = JM2.job\_number AND JM2.maintenance\_id = M2.maintenance\_id AND C1.company\_id = C2.company\_id GROUP BY C2.company\_id)

**Query 3 – Output (cw\_company.odt)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| company\_id | company\_name | job\_number | job\_description | start\_date | end\_date | month\_payment |
| 1 | ChoiceRepair | 5 | Clean lamps | 2015-05-01 | 2015-07-01 | 600 |
| 3 | HoustonFix | 5 | Clean lamps | 2015-02-16 | 2015-08-20 | 580 |
| 6 | VacFix | 6 | Test fire alarm | 2015-02-09 | 2015-08-21 | 1200 |
| 7 | ParkFix Ltda. | 9 | Disinfect drains | 2015-03-16 | 2015-08-20 | 800 |
| 8 | Solve Me | 10 | Empty bins | 2015-01-13 | 2016-01-22 | 500 |
| 8 | Solve Me | 9 | Disinfect drains | 2015-01-04 | 2015-08-20 | 700 |

**Query 4**

“Select all the employees who are supervised by the supervisor number 6.”

**Input required:** “Supervisor Number”.

**Input given by the user:** 1

SELECT S.employee\_number, E.name, E.address, E.telephone FROM cw\_supervises S, cw\_employee E WHERE E.employee\_number = S.employee\_number AND S.supervisor\_number = 1

**Query 4 – Output (cw\_supervises.odt)**

|  |  |  |  |
| --- | --- | --- | --- |
| employee\_number | name | address | telephone |
| 2 | Lucas | 49 Rottenrow East C10-3 | +4407474231342 |
| 3 | Ivo | 49 Rottenrow East C10-2 | +4407474986254 |
| 4 | Carlos | 49 Rottenrow East C10-1 | +4407474332242 |

**VI) Critique of implementation**

**Database:**

In my implementation I used a methodology, which was to represent each entity and each relationship with one table. With this approach I reduced the redundancy of rows with the same value and improve the performance as well. Also this approach give you more mobility, you can correlate different and tables that seems that there’s no relation between then, by just joining them. However, the modeling being efficient and easy to understand, I believe that with this design the queries can be quite confusing, because it is necessary to join several tables, sub-queries and conditions to relate several tables.

This division of tables, made my work easier when I did the task of making self join query, because I used an intermediate table to do it, so I just joined two tables.  
On the entity employee I used the type VARCHAR for the attribute telephone instead of a number.

I decided not to create new entities for each type of zoning, because there’s no relevant information that was suppose to be added to each type of zoning, just its name, so there’s no necessity for a new entity.

For the company entity I added the company name to make the table easier to read instead of only numbers.

To store the information of the maintenance, the start date, the end date and the month payment are part attributes of the entity maintenance. So this way, every entity which relates with maintenance will have access to these information pretty easily.

**Web Development:**

For the web development my implementation was pretty linear. As required, I kept all the pages on the same format, which made the design and CSS styling much easier.

Is listed on the index page, all the queries and what it is suppose to return for the user. This way the user doesn’t need to click in every query page to see what is required.

This methodology is good because maximize the efficiency of the webpage by directing the user specifically to where he wants to go.

I don’t see any big weaknesses in my implementation. The home page shows clearly all the possible information that can be retrieved with the queries. The menu is well-located and easy to read and access. Inside each query page, there is a table, which is direct related to what the query need, and also the table contain pretty much all the information need for the user to complete the query with success.

For the logo it is not specified that it must be an image, so just using CSS I was able to work with font size, positioning and styling to give a pleasant visual.