DATABASE SYSTEMS

SQL Project

MANAGING CUSTOMER AND JOBS DATABASE WITH SQL

By

IREKE, Ukiwo Ireke

TABLE OF CONTENTS

PROJECT OBJECTIVES

TASK D Create a corresponding relational schema, and verify it meets first, second and third normal forms.

TASK E Implement your database scheme using MariaDB and load the test data.

TASK F Then write, run and test the report you identified.

1 ER DIAGRAM

Source :

- The Database design selected was the **Customer and Job data model** as shown in figure 1 below:

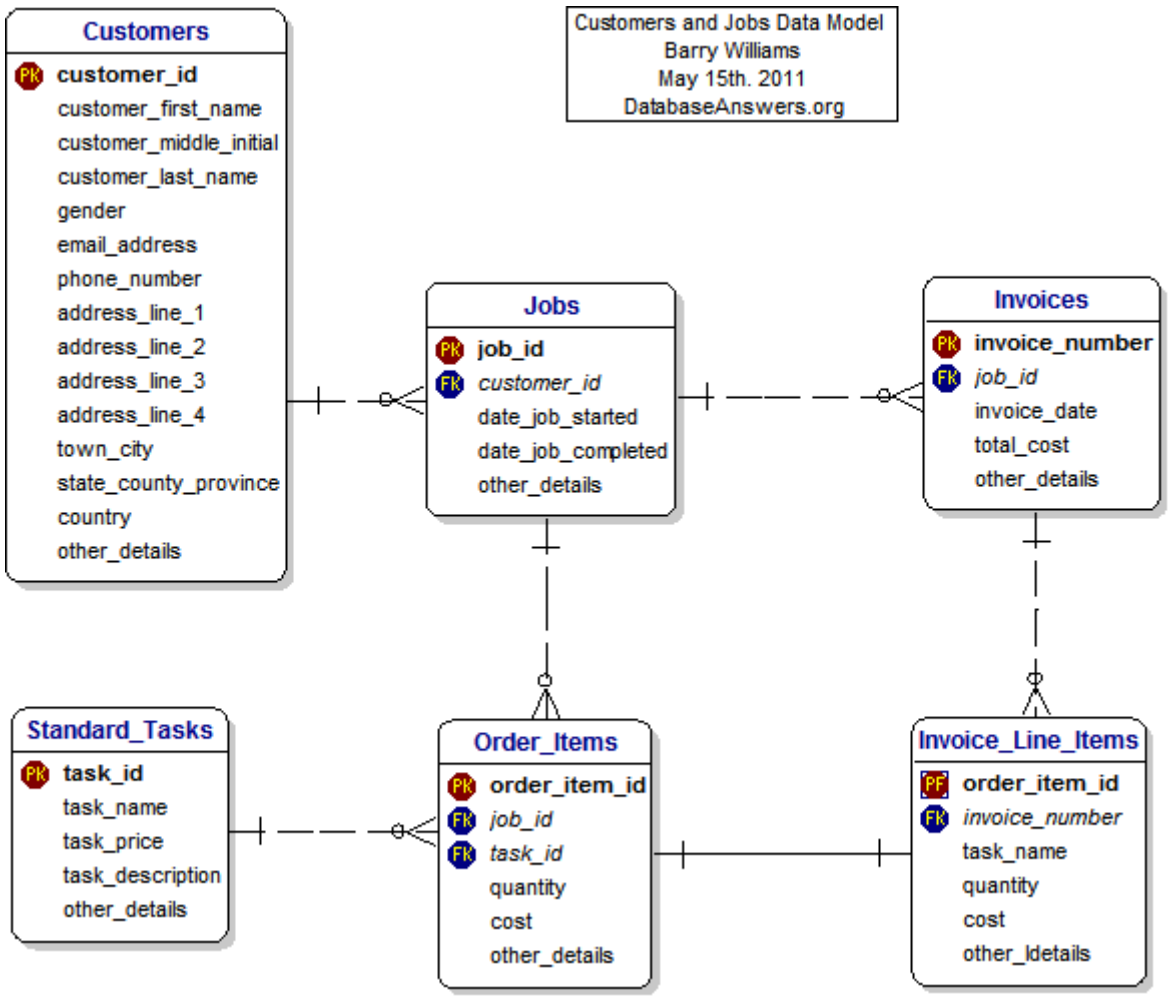

Figure 1: Customer and Job ER Diagram (Source: http://www.databaseanswers.org/data_models/customers_and_jobs/index.htm)

Modifications made to Diagram:

- The Entity Relationship Diagram was transformed from the initial Crow's foot notations into the UML notation making it more appropriate for this assignment.(*Data Model for Customers and Jobs*, n.d.)
- The diagram was clearly annotated with directions included to show the relationships between each entity.
- The ***address_line_1*** to ***address_line_4*** into the ***Customer*** entity was grouped into one single attribute ***address*** for more clarity.
- The ***total_cost*** attribute was removed since this can be calculated from the individual costs
- Some repetitive and redundant relationship was identified such as ***Invoice_Line_item*** and ***Order_items*** as such ***invoice_line_items*** table was merged ***invoice*** table since it explains just one

job . This unnormalized ***invoice*** had ***invoice_number*** attribute as primary key and job_id and order_item_id as foreign key since it is primary key from other tables.

- The ***task_name*** attribute in **invoice_Line_item** was also removed since this is already reflected in the order_item_id key brough in from the Order_Items , thus might be repetitive and unnecessary.
- **Order_item_name** attribute was added to the Order_Items attribute as other details since its important to know the product name.
- **Jobs** in

Customer and Job ER Diagram

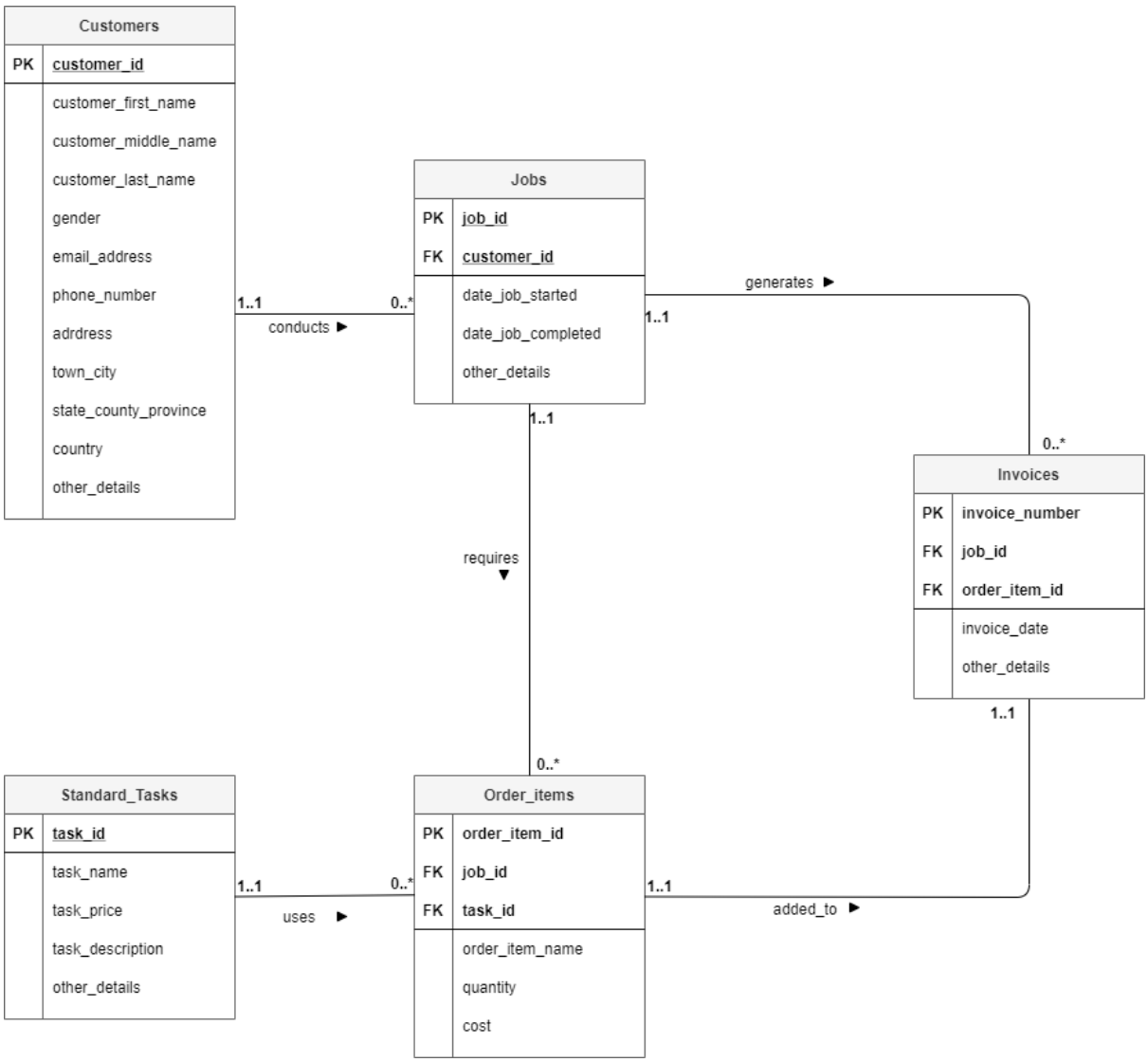

Figure 2: Customer and jobs Entity UML Diagram

2 TASK D

2.1 Relational Schema

For ease of use, attributes from the tables were abbreviated (Abbrv.) with Keys identified from the Entities were underlined with Primary Keys attributes in Bold as shown in table1 below:

Table 1: Summary of Entity Attributes with abbreviations from the considered ER Diagram

CUSTOMERS

Attribute name	Abbrv
customer_id	**cust_id**
customer_firstname	fname
customer_middle_name	mname
customer_last_name	lname
gender	gender
email_address	email
phone_number	phone_no
address	addr
town_city	city
state_county_province	county
country	country
other_details	others

JOBS

Attribute name	Abbrv
job_id	**job_id**
customer_id	cust_id
date_job_started	startdate
date_job_completed	enddate
other_details	others

INVOICES

Attribute name	Abbrv
invoice_number	**invoice_no**
job_id	job_id
order_item_id	item_id
invoice_date	invoice_date
other_details	others

STANDARD TASKS

Attribute name	Abbrv
task_id	**task_id**
task_name	task_name
task_price	task_price
task_description	task_desc
other_details	others

ORDER ITEMS

Attribute name	Abbrv
order_item_id	**item_id**
job_id	job_id
task_id	task_id
quantity	qty
cost	cost
order_item_name	item_name

Proposed Relational Schema for the Customer and Job ER Diagram in Unnormalized Form(UNF) are as follows:

CUSTOMERS(**cust_id**, fname, mname, lname, gender, email, phone_no, addr, city, county, country, others)

JOBS(**job_id**, *cust_id*, startdate, enddate, others)

STANDARD_TASKS(**task_id**, task_name, task_price, task_desc, others)

ORDER_ITEMS(**item_id**, *job_id*, *task_id*, item_name, qty, cost)

INVOICES(**invoice_no**, *job_id*, *item_id*, invoice_date, others)

2.2 Normalization

Some repetitions and redundancies were discovered from this proposed schema. Thus, the following as normalization stages are implemented, the proposed Relational Schema will be consequently modified as well.

2.2.1 First Normal Form (1NF)

To conform with the first normal form(1NF), the table must fulfil the following criteria:

- **All data entries must be atomic. In other words, all multi-value data must be in separate columns.** The address attribute in the CUSTOMERS table does not meet this criterion since an address holds multiple entries such as House Number, Street Name, Post Code. As such the address attribute will be further split and replaced with *house_no, street_name and post_code* attributes. The modified Shema will thus be:

CUSTOMERS(**cust_id**, fname, mname, lname, gender, email, phone_no, house_no, street_name, post_code, city, county, country, others)

JOBS(**job_id**, *cust_id*, startdate, enddate, others)

STANDARD_TASKS(**task_id**, task_name, task_price, task_desc, others)

ORDER_ITEMS(**item_id**, *job_id, task_id*, item_name , qty, cost, others)

INVOICES(**invoice_no**, *job_id, item_id*, invoice_date, others)

- **There must be a primary key.** This criterion is met with all primary keys identified in **bold** with the corresponding attributes dependent on them in each table. Schema is unchanged at this stage.
- **There must be no repeating groups.** This is to ensure no repetition amongst the non-key attributes after primary and foreign keys have been identified. This criterion is also met but the other_details (others) attributes will be dropped. Hence, the modified schema becomes:

CUSTOMERS(**cust_id**, fname, mname, lname, gender, email, phone_no, house_no, street_name , post_code, city, county, country)

JOBS(**job_id**, *cust_id*, startdate, enddate)

STANDARD_TASKS(**task_id**, task_name, task_price, task_desc)

ORDER_ITEMS(**item_id**, *job_id, task_id*, item_name , qty, cost)

INVOICES(**invoice_no**, *job_id, item_id*, invoice_date)

2.2.2 Second Normal Form (2NF)

Here, the schema must be in 1NF then :

- **Every non key attribute should be fully functional and dependent on the primary key.** In other words, there must be no partial dependencies. To meet this criterion, we identify any attribute which depends only on part of a primary key and place them in a different table. Our Schema was well sorted and satisfies the 2NF with no partial dependency identified. Thus, remains unchanged.

2.2.3 Third Normal Form (3NF)

There must be no transitive functional dependencies.

Consequently, with our Schema satisfying these 3 Normalization forms, we can conclude our derived Relational Schema in third normal form is as follows:

CUSTOMERS(**cust_id**, fname, mname, lname, gender, email, phone_no, house_no, street_name, post_code, city, county, country)

JOBS(**job_id**, *cust_id*, startdate, enddate)

STANDARD_TASKS(**task_id**, task_name, task_price, task_desc)

ORDER_ITEMS(**item_id**, *job_id*, *task_id*, item_name , qty, cost)

INVOICES(**invoice_no**, *job_id*, *item_id*, invoice_date)

Note:

- Primary Keys are shown in **bold** underlined.
- Foreign keys are in *italics* and underlined.

3 TASK E

3.1 Data Definition Language (DDL) Statements

The table were created using CREATE statement with PRIMARY and FOREIGN KEYS defined and REFERENCED to their parent tables. DDL code used to create the table is shown below. **Description of table is shown in Appendix.**

```
---DDL
---Creating Customers tables
DROP TABLE customers;
CREATE TABLE  customers(
cust_id INTEGER NOT NULL PRIMARY KEY,
fname VARCHAR(20) NOT NULL,
mname VARCHAR(20),
lname VARCHAR(20) NOT NULL,
gender VARCHAR(10),
email VARCHAR(50),
phone_no INTEGER,
house_no INTEGER,
street_name VARCHAR(50),
post_code VARCHAR(10),
city VARCHAR(20),
county VARCHAR(50),
country VARCHAR(50)
);

---Creating Jobs tables
DROP TABLE jobs;
CREATE TABLE jobs(
job_id INT NOT NULL PRIMARY KEY,
cust_id INT NOT NULL,
start_date DATE,
end_date DATE,
FOREIGN KEY (cust_id) REFERENCES customers(cust_id)
);
```

```
---Creating Standard Tasks tables
DROP TABLE standard_tasks;
CREATE TABLE standard_tasks(
task_id INT NOT NULL PRIMARY KEY,
task_name VARCHAR(50),
task_price FLOAT,
task_desc VARCHAR(50)
);
---Creating order_items tables

DROP TABLE order_items;
CREATE TABLE order_items(
item_id INT NOT NULL PRIMARY KEY,
job_id INT NOT NULL,
task_id INT NOT NULL,
item_name VARCHAR(20),
qty INT,
cost FLOAT,
FOREIGN KEY (job_id ) REFERENCES jobs(job_id ),
FOREIGN KEY (task_id ) REFERENCES standard_tasks(task_id )
);

---Creating invoices tables
DROP TABLE invoices;
CREATE TABLE invoices(
invoice_no INT NOT NULL PRIMARY KEY,
job_id INT NOT NULL,
item_id INT NOT NULL,
invoice_date DATE,
FOREIGN KEY (job_id ) REFERENCES jobs(job_id ),
FOREIGN KEY (item_id) REFERENCES order_items(item_id )
);
```

3.2 DML Statements

DML statements such as INSERT were used to add values to tables shown below;

```
--- DML

---Populate customers table

INSERT INTO customers VALUES (1001, "John " , "Millar ", "Doe ", "male
" ,

"johndoe@gmail.com " , 111222 , 12 , "Princes street " , "E11 5GH " ,
"Edinburgh " ,

"city of edinburgh " , "United Kingdom "),

( 1002, "Stephen " , "Kinnock ", "Rutherford ", "male " ,
"stephenrutherford@gmail.com ",

NULL , 5 , "Leith walk " , "E5 6DS " , "Edinburgh " , "city of edinburgh
" ,

"United Kingdom "),

(1003, "Peter " , "Bush ", "Willett ", "male " , "peterwillett@gmail.com
" , 2162456 ,

43 , "Sighthill " , "E33 5GH " , "Edinburgh " , "east lothian " , "United
Kingdom "),

(1004, "Edward " , "Yi He ", "McCaig ", "male " , "edwardmccaig@gmail.com
" , 123654 ,

5 , "Gordon street " , "E44 5RA " , "Edinburgh " , "west lothian " ,
"United Kingdom "),

(1005, "Duncan " , "Higgitt ", "Begg ", "male " , "duncanbegg@gmail.com
" , 1238965 , 54,

"Leith walk " , "E29 7LM " , "Edinburgh " , "city of edinburgh " , "United
Kingdom "),

(1006, "Helen " , "Clarke ", "Thomson ", "female " ,
"helenthomson@gmail.com " , 3266985,

543 , "Sighthill " , "E33 LK " , "Edinburgh " , "east lothian " , "United
Kingdom "),

( 1007, "Captain " , "Beany ", "Tyrinne ", "male " ,
"captaintyrinne@gmail.com ", 326547,

NULL , "Gordon street " , "E86JY " , "Edinburgh " , "west lothian " ,
"United Kingdom "),

(1008, "Jonathan " , "Tier ", "Christopher ", "male " ,
"jonathanchristopher@gmail.com ",

NULL , 24 , "Leith walk " , "E4 6TR " , "Edinburgh " , "city of edinburgh
","United Kingdom"),

( 1009, "Andrew " , "Jordan ", "Callum ", "male " , "andrewcallum@gmail.com
" , 3625147 ,
```

```
 99 ,  "Sighthill " ,  "E22 8YT " ,  "Edinburgh " ,  "east lothian " ,
"United Kingdom "),

(1010,  "Owen " , "Herbert ",  "Anne ",  "male " , "owenanne@gmail.com " ,
9876554 , 8 ,

"Gordon street " ,  "E28 BH " ,  "Edinburgh " ,  "west lothian " ,  "United
Kingdom "),

(1011,  "Stephen " , "Kinnock ",  "Gray ",  "male " , "stephengray@gmail.com
" , 5555555 ,

 8 ,  "Leith walk " , "E5 6DS " ,  "Edinburgh " ,  "city of edinburgh " ,
"United Kingdom ");


---Populate job table

INSERT INTO jobs VALUES (30, 1004,  "2017-12-29", "2028-03-31"),

(31, 1002, "2017-04-14", "2023-12-21"),

(32, 1008, "2016-02-09", "2030-03-13"),(33, 1008, "2016-05-19", "2024-07-
17"),

(34, 1002, "2015-08-31", "2019-08-01"),(35, 1008, "2016-06-04", "2020-10-
29"),

(36, 1009, "2016-02-21", "2028-06-12"),(37, 1002, "2015-08-26", "2024-06-
19"),

(38, 1010, "2016-05-08", "2019-10-20"),(39, 1003, "2015-04-04", "2029-09-
17"),

(40, 1001, "2015-03-22", "2023-08-13");


---Populate standard_tasks table

INSERT INTO standard_tasks VALUES (234,  "task_p ", 100,  "service  "),

(235,  "task_q ", 119,  "material "),

(236,  "task_r ", 37,  "electrical "),

(237,  "task_s ", 110,  NULL),

(238,  "task_u ", 104,  "service  "),

(239,  "task_v ", 107,  "engineering "),

(240,  "task_w ", 99,  NULL),

(241,  "task_x ", 114,  "engineering "),

(242,  "task_y ", 89,  "material "),

(243,  "task_z ", 60,  "service  ");
```

```
---Populate order_items table
INSERT INTO order_items VALUES
(1, 32, 240,  "plywood ", 13, 4),
(2, 36, 235,  "copper_wires ", 9, 11),
(3, 37, 234,  "washed_sand ", 9, 10),
(4, 39, 236,  "ceramic_tiles ", 6, 10),
(5, 40, 241,  "marble ", 8, 3),
(6, 37, 239,  "frames ", 6, 7),
(7, 34, 236,  "laptop ", 7, 10),
(8, 33, 237,  "sockets ", 7, 3),
(9, 30, 242,  "sinks ", 3, 5),
(10, 39, 237,  "interior_paint ", 7, 3),
(11, 38, 238,  "exterior_paint ", 2, 3),
(12, 37, 240,  "screw_drivers ", 10, 7),
(13, 40, 234,  "drills ", 14, 11),
(14, 38, 238,  "alluminium ", 14, 3),
(15, 34, 238,  "pipes ", 11, 6),
(16, 36, 240,  "masks ", 0, 3),
(17, 39, 236,  "gloves ", 11, 10),
(18, 34, 234,  "safety_boots ", 1, 11);

---Populate invoices table
INSERT INTO invoices VALUES
(571, 33, 12, "2019-06-13"),
(572, 37, 2, "2019-07-25"),
(573, 40, 17, "2020-08-05"),
(574, 36, 14, "2019-06-11"),
(575, 38, 13, "2020-01-22"),
(576, 39, 15, "2019-07-30"),
(577, 32, 7, "2020-04-17"),
(578, 37, 9, "2019-05-30"),
(579, 34, 7, "2019-10-11"),
(580, 37, 1, "2019-06-13");
```

Afterwards, the SELECT statement was used to view our table as shown below;

```
SELECT * FROM customers;
```

```
MariaDB [40478776]> select * from customers;
+---------+-----------+----------+--------------+--------+--------------------------------+----------+----------+-----------------+-----------+-----------+--------------------+-----------+
| cust_id | fname     | mname    | lname        | gender | email                          | phone_no | house_no | street_name     | post_code | city      | county             | country   |
+---------+-----------+----------+--------------+--------+--------------------------------+----------+----------+-----------------+-----------+-----------+--------------------+-----------+
|    1001 | John      | Millar   | Doe          | male   | johndoe@gmail.com              |   111222 |       12 | Princes street  | E11 5GH   | Edinburgh | city of edinburgh  | United Kingdom |
|    1002 | Stephen   | Kinnock  | Rutherford   | male   | stephenrutherford@gmail.com    |     NULL |        5 | Leith walk      | E5 6DS    | Edinburgh | city of edinburgh  | United Kingdom |
|    1003 | Peter     | Bush     | Willett      | male   | peterwillett@gmail.com         |  2162456 |       43 | Sighthill       | E33 5GH   | Edinburgh | east lothian       | United Kingdom |
|    1004 | Edward    | Yi He    | McCaig       | male   | edwardmccaig@gmail.com         |   123654 |        5 | Gordon street   | E44 5RA   | Edinburgh | west lothian       | United Kingdom |
|    1005 | Duncan    | Higgitt  | Begg         | male   | duncanbegg@gmail.com           |  1238965 |       54 | Leith walk      | E29 7LM   | Edinburgh | city of edinburgh  | United Kingdom |
|    1006 | Helen     | Clarke   | Thomson      | female | helenthomson@gmail.com         |  3266985 |      543 | Sighthill       | E33 LK    | Edinburgh | east lothian       | United Kingdom |
|    1007 | Captain   | Beany    | Tyrinne      | male   | captaintyrinne@gmail.com       |   326547 |     NULL | Gordon street   | E86JY     | Edinburgh | west lothian       | United Kingdom |
|    1008 | Jonathan  | Tier     | Christopher  | male   | jonathanchristopher@gmail.com  |     NULL |       24 | Leith walk      | E4 6TR    | Edinburgh | city of edinburgh  | United Kingdom |
|    1009 | Andrew    | Jordan   | Callum       | male   | andrewcallum@gmail.com         |  3625147 |       99 | Sighthill       | E22 8YT   | Edinburgh | east lothian       | United Kingdom |
|    1010 | Owen      | Herbert  | Anne         | male   | owenanne@gmail.com             |  9876554 |        8 | Gordon street   | E28 8H    | Edinburgh | west lothian       | United Kingdom |
|    1011 | Stephen   | Kinnock  | Gray         | male   | stephengray@gmail.com          |  5555555 |        8 | Leith walk      | E5 6DS    | Edinburgh | city of edinburgh  | United Kingdom |
+---------+-----------+----------+--------------+--------+--------------------------------+----------+----------+-----------------+-----------+-----------+--------------------+-----------+
11 rows in set (0.111 sec)
```

Figure 3: customers table viewed in MariaDB using the SELECT statement

```
SELECT * FROM jobs;
```

```
MariaDB [40478776]> select * from jobs;
+--------+---------+------------+------------+
| job_id | cust_id | start_date | end_date   |
+--------+---------+------------+------------+
|     30 |    1004 | 2017-12-29 | 2028-03-31 |
|     31 |    1002 | 2017-04-14 | 2023-12-21 |
|     32 |    1008 | 2016-02-09 | 2030-03-13 |
|     33 |    1008 | 2016-05-19 | 2024-07-17 |
|     34 |    1002 | 2015-08-31 | 2019-08-01 |
|     35 |    1008 | 2016-06-04 | 2020-10-29 |
|     36 |    1009 | 2016-02-21 | 2028-06-12 |
|     37 |    1002 | 2015-08-26 | 2024-06-19 |
|     38 |    1010 | 2016-05-08 | 2019-10-20 |
|     39 |    1003 | 2015-04-04 | 2029-09-17 |
|     40 |    1001 | 2015-03-22 | 2023-08-13 |
+--------+---------+------------+------------+
11 rows in set (0.000 sec)
```

Figure 4: jobs table viewed in MariaDB using the SELECT statement

```
SELECT * FROM order_items;
```

```
MariaDB [40478776]> select * from order_items;
+---------+--------+---------+-----------------+------+------+
| item_id | job_id | task_id | item_name       | qty  | cost |
+---------+--------+---------+-----------------+------+------+
|       1 |     32 |     240 | plywood         |   13 |    4 |
|       2 |     36 |     235 | copper_wires    |    9 |   11 |
|       3 |     37 |     234 | washed_sand     |    9 |   10 |
|       4 |     39 |     236 | ceramic_tiles   |    6 |   10 |
|       5 |     40 |     241 | marble          |    8 |    3 |
|       6 |     37 |     239 | frames          |    6 |    7 |
|       7 |     34 |     236 | laptop          |    7 |   10 |
|       8 |     33 |     237 | sockets         |    7 |    3 |
|       9 |     30 |     242 | sinks           |    3 |    5 |
|      10 |     39 |     237 | interior_paint  |    7 |    3 |
|      11 |     38 |     238 | exterior_paint  |    2 |    3 |
|      12 |     37 |     240 | screw_drivers   |   10 |    7 |
|      13 |     40 |     234 | drills          |   14 |   11 |
|      14 |     38 |     238 | alluminium      |   14 |    3 |
|      15 |     34 |     238 | pipes           |   11 |    6 |
|      16 |     36 |     240 | masks           |    0 |    3 |
|      17 |     39 |     236 | gloves          |   11 |   10 |
|      18 |     34 |     234 | safety_boots    |    1 |   11 |
+---------+--------+---------+-----------------+------+------+
18 rows in set (0.001 sec)
```

Figure 5: order_items table viewed in MariaDB using the SELECT statement

```
SELECT * standard_tasks;
```

```
MariaDB [40478776]> select * from standard_tasks;
+---------+-----------+------------+--------------+
| task_id | task_name | task_price | task_desc    |
+---------+-----------+------------+--------------+
|     234 | task_p    |        100 | service      |
|     235 | task_q    |        119 | material     |
|     236 | task_r    |         37 | electrical   |
|     237 | task_s    |        110 | NULL         |
|     238 | task_u    |        104 | service      |
|     239 | task_v    |        107 | engineering  |
|     240 | task_w    |         99 | NULL         |
|     241 | task_x    |        114 | engineering  |
|     242 | task_y    |         89 | material     |
|     243 | task_z    |         60 | service      |
+---------+-----------+------------+--------------+
```

Figure 6: standard_tasks table viewed in MariaDB using the SELECT statement

```
SELECT FROM invoices;
```

```
MariaDB [40478776]> select * from invoices;
+------------+--------+---------+--------------+
| invoice_no | job_id | item_id | invoice_date |
+------------+--------+---------+--------------+
|        571 |     33 |      12 | 2019-06-13   |
|        572 |     37 |       2 | 2019-07-25   |
|        573 |     40 |      17 | 2020-08-05   |
|        574 |     36 |      14 | 2019-06-11   |
|        575 |     38 |      13 | 2020-01-22   |
|        576 |     39 |      15 | 2019-07-30   |
|        577 |     32 |       7 | 2020-04-17   |
|        578 |     37 |       9 | 2019-05-30   |
|        579 |     34 |       7 | 2019-10-11   |
|        580 |     37 |       1 | 2019-06-13   |
+------------+--------+---------+--------------+
10 rows in set (0.000 sec)
```

Figure 7: invoices table viewed in MariaDB using the SELECT statement

4 TASK F

4.1 Modified Report Query

The changes and modification made on the ER Diagram led to the drafting of a more suitable report query for this work ensuring at least 3 tables are used. Therefore, the considered report query is;

What is the total quantity of items required to conduct jobs in which Customer Stephen Kinnock Rutherford is involved in?

To be able to develop this report query we will be using the fname = "Stephen ", mname = " Kinnock" and *lname = "Rutherford"* from **CUSTOMERS** table to obtain the customer_id of the customer. This will ensure we obtain the right customer_id since a customer can have the same first name and last name. Afterwards, find the jobs they are involved in from the **JOBS** table. Finally, we will use the **ORDER_ITEMS** table to sum all *quantity (qty) of items associated with job_id* . The best way of achieving this will be to make use of subqueries.

4.2 Implementation of Report Query

Firstly, we had to find the customer id (cust_id) of the Customer called Stephen Kinnock Rutherford using the first name (*fname*), middle name (*mname*) and last name (*lname*) as shown below;

```
select cust_id from customers where fname like "Stephen%" and mname
like "Kinnock%" and lname like "Rutherford%";
```

```
MariaDB [40478776]>  select cust_id from customers where fname like "Stephen%" and
    -> mname like "Kinnock%" and lname like "Rutherford%";
+---------+
| cust_id |
+---------+
|    1002 |
+---------+
1 row in set (0.004 sec)
```

Figure 8: query result showing customer Stephen Kinnock Rutherford has customer id 1002

Next, we then must find the job_id of jobs in which the Customer is involved in using the previous code as subquery as shown by the following query;

```
select job_id from jobs where cust_id = (select cust_id from customers
where fname like  "Stephen%" and mname like "Kinnock%" and lname like
"Rutherford%");
```

```
MariaDB [40478776]> select job_id from jobs where cust_id =
    -> (select cust_id from customers where fname like  "Stephen%" and
    -> mname like "Kinnock%" and lname like "Rutherford%");
+--------+
| job_id |
+--------+
|     31 |
|     34 |
|     37 |
+--------+
3 rows in set (0.001 sec)
```

Figure 9: query result showing jobs in which customer Stephen Kinnock Rutherford is involved in are job_id 31, 34, 37.

Finally, to implement our report query we then use the previous query as sub query to find the total quantity of items for the job_id 31, 34 and 37. As shown below;

```
select sum(qty) from order_items where job_id in (select job_id from
jobs where cust_id = (select cust_id from customers where fname like
"Stephen%" and mname like "Kinnock%"and lname like "Rutherford%"));
```

```
MariaDB [40478776]> select sum(qty) from order_items where job_id in
    -> (select job_id from jobs where cust_id =
    -> (select cust_id from customers where fname like "Stephen%"
    -> and mname like "Kinnock%"and lname like "Rutherford%"));
+----------+
| sum(qty) |
+----------+
|       44 |
+----------+
1 row in set (0.001 sec)
```

Figure 10: Report Query result showing total quantity of items to conduct jobs in which Customer Stephen Kinnock Rutherford is involved in.

The result showed that a total quantity of items used to conduct jobs in which customer Stephen Kinnock Rutherford is involved in to be 44.

4.3 Test Plan Review

A good test plan to verify the validity and correctness of our report query will be to execute each sub-query and compare the obtained results against what was expected ensuring we obtained the correct values. This was achieved during our implementation process by building to final query in bits of sub-queries executing each of them to ensure correctness.

Firstly, we had to ensure the customer id from our first subquery was that of Customer *Stephen Kinnock Rutherford*. This was achieved by extracting ensuring the first name(fname), middle name (*mname*) and last names(*lname*) corresponds to that of the described customer since we had another customer STEPHEN KINNOCK GRAY with cust_id 1011 who had the same first name and middle name but different last name. So, the result returned from our query was **cust_id 1002** which is correct.

Next, we had to obtain job_id of jobs in which the customer was involved in using another sub-query. Looking at jobs tables we expected the result to return **job_id 31, 34 and 37** which was eventually returned confirming correctness of this second part of our query.

Finally, we used to these sub-queries to build our final report query to calculating the total number of quantities required for jobs with job_id 31, 34 and 37. Table 2 below summarizes the items required for the jobs from the order_item table.

Table 2: Summary of sum of quantity of items used for jobs 31, 34 and 37 from order_items table

item_id	job_id	task_id	item_name	qty	cost
3	37	234	washed_sand	9	10
6	37	239	frames	6	7
7	34	236	laptop	7	10
12	37	240	screw_drivers	10	7
15	34	238	pipes	11	6
18	34	234	safety_boots	1	11
			SUM	44	

Since our report query showed the total quantity to be **44** thus validated our report query

REFERENCES

Data Model for Customers and Jobs. (2011). Retrieved 1 November 2020, from http://www.databaseanswers.org/data_models/customers_and_jobs/index.htm

APPENDIX

Tables data format;

```
DESCRIBE customers;
```

```
MariaDB [40478776]> describe customers;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| cust_id     | int(11)     | NO   | PRI | NULL    |       |
| fname       | varchar(20) | NO   |     | NULL    |       |
| mname       | varchar(20) | YES  |     | NULL    |       |
| lname       | varchar(20) | NO   |     | NULL    |       |
| gender      | varchar(10) | YES  |     | NULL    |       |
| email       | varchar(50) | YES  |     | NULL    |       |
| phone_no    | int(11)     | YES  |     | NULL    |       |
| house_no    | int(11)     | YES  |     | NULL    |       |
| street_name | varchar(50) | YES  |     | NULL    |       |
| post_code   | varchar(10) | YES  |     | NULL    |       |
| city        | varchar(20) | YES  |     | NULL    |       |
| county      | varchar(50) | YES  |     | NULL    |       |
| country     | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
13 rows in set (0.046 sec)
```

```
DESCRIBE jobs;
```

```
MariaDB [40478776]> describe jobs;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| job_id     | int(11) | NO   | PRI | NULL    |       |
| cust_id    | int(11) | NO   | MUL | NULL    |       |
| start_date | date    | YES  |     | NULL    |       |
| end_date   | date    | YES  |     | NULL    |       |
+------------+---------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

```
DESCRIBE standard_tasks;
```

```
MariaDB [40478776]> describe standard_tasks;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| task_id    | int(11)     | NO   | PRI | NULL    |       |
| task_name  | varchar(50) | YES  |     | NULL    |       |
| task_price | float       | YES  |     | NULL    |       |
| task_desc  | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

DESCRIBE order_items;

```
MariaDB [40478776]> describe order_items;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| item_id   | int(11)     | NO   | PRI | NULL    |       |
| job_id    | int(11)     | NO   | MUL | NULL    |       |
| task_id   | int(11)     | NO   | MUL | NULL    |       |
| item_name | varchar(20) | YES  |     | NULL    |       |
| qty       | int(11)     | YES  |     | NULL    |       |
| cost      | float       | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.002 sec)
```

DESCRIBE invoices;

```
MariaDB [40478776]> describe invoices;
+--------------+---------+------+-----+---------+-------+
| Field        | Type    | Null | Key | Default | Extra |
+--------------+---------+------+-----+---------+-------+
| invoice_no   | int(11) | NO   | PRI | NULL    |       |
| job_id       | int(11) | NO   | MUL | NULL    |       |
| item_id      | int(11) | NO   | MUL | NULL    |       |
| invoice_date | date    | YES  |     | NULL    |       |
+--------------+---------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```