Busy Bee Project

Chapter	16-17
Created	@Dec 4, 2020 3:48 AM
Materials	
Recall Date 1	
Recall Date 2	
Reviewed	☐
Type	Project
Week	

Source

GitHub Code - https://github.com/isaacattuah/CSC423-Final-Project

Notion Page - https://www.notion.so/Busy-Bee-Project-1933633b3a224254aa05ded539af8d3a

isaacattuah/CSC423-Final-Project

A database final project based on the BusyBee Cleaning Company GitHub is home to over 50 million developers working together to host and review code, manage projects,

https://github.com/isaacattuah/CSC423-Final-Project

Case Study 2: BusyBee Cleaning Company

The BusyBee Cleaning Company specializes in providing cleaning services for clients. **Each type of client** has a set of requirements. For example, **The Cardboard Box Company** requires cleaning services from Monday to Friday

7am until 9am and 5pm until 7pm each day, but **P. Nuttall** only requires cleaning services on a Wednesday from 10am until 1pm.

Whenever a new client is taken on, it is determined whether any special equipment is required and when. For example, three industrial floor cleaners may be needed on two out of five occasions for one client.

Therefore, the following information will be stored for **each equipment**, in addition to the equipment identifier: description, usage, and cost.

For **each employee** the following data will be stored: staff number (uniquely identifies an employee), first and last name, address, salary, and telephone number. For **each client** , the following data will be stored: client number (uniquely identifies a client), first and last name, address, and telephone number.

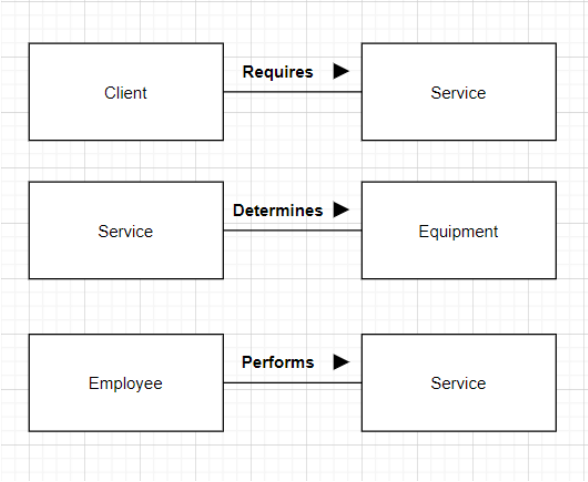
Conceptual Model

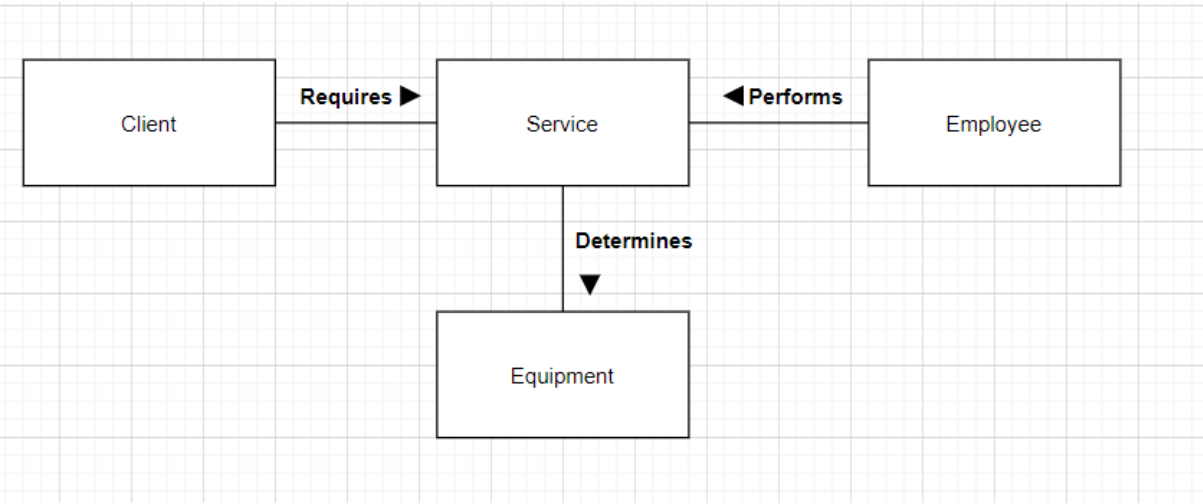
Relations

- Client (**client number**, first name, last name, address, and telephone number)
- Employee (**staff number**, first name, last name, address, salary, and telephone number)
- Equipment (**equipment identifier**, description, usage, cost)
- Service (**serviceNo**, start day, end day, start time, end time)

Relationship Types

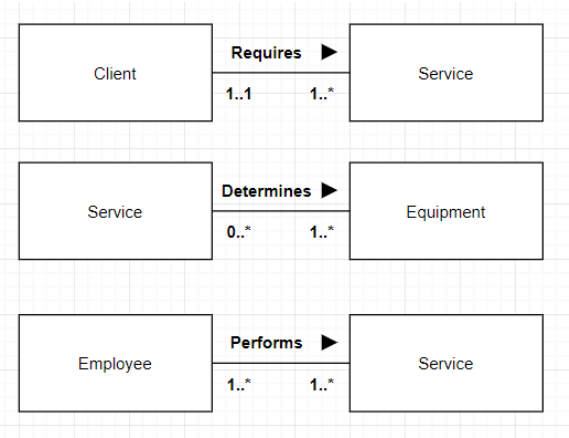
- Client requires Service
- Service determines Equipment
- Employees performs a Service

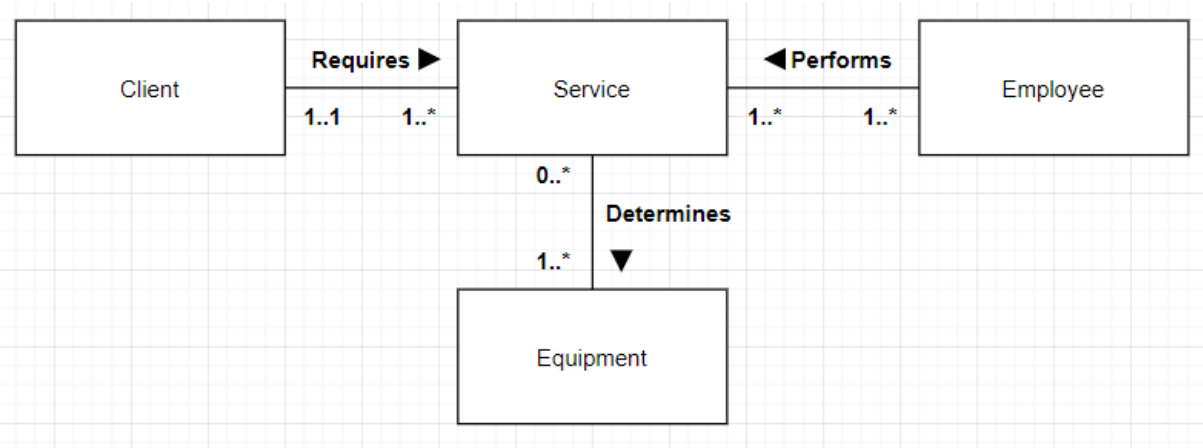




Cardinality Constraints

- **Client requires Service**
  - A client requires a services (1..*)
  - A service is required by one client (1..1)
- **Service determines Equipment**
  - A service determines the amount of equiment used (1..*)
  - If needed, a special equipment is required for services (0..*)
- **Employees performs a Service**
  - An Employee perform multiple services (1..*)
  - A service is performed by mutliple employees (1..*)





Attributes

Implementation

- Client requires Service
  - Client (**client number**, first name, last name, address, telephone number)
  - Service (**client identifier**, client name, start day, end day, start time, end time)
- Service determines Equipment
  - Service (**serviceNo**, start day, end day, start time, end time)
  - Equipment (**equipment identifier**, description, usage, cost)
- Employees performs Service
  - Employee (**staff number**, first name, last name, address, salary, and telephone number)
  - Service (**serviceNo**, start day, end day, start time, end time)

Assumptions

- Equipments are special and may or may not be determined for a given service
- Services do not require Equipment hence any attempt to use the logic of Service requires Equipment or vice versa will render the multiplicities innaccurate

Candidate and Primary Keys

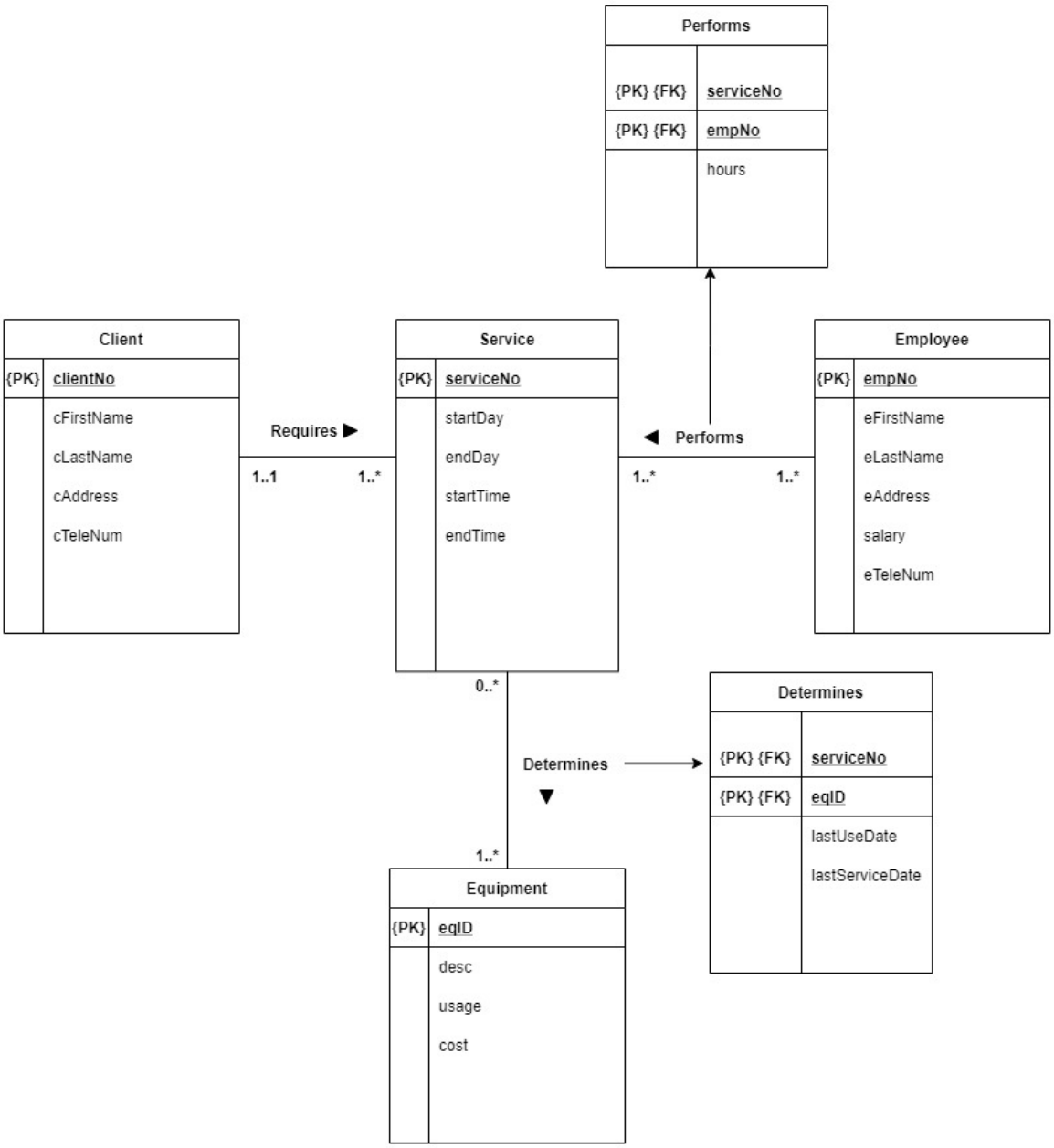
- Client (**client number**, first name, last name, address, telephone number)
  - Primary key: client number
  - Candidate key: client number, telephone number
- Employee (**staff number**, first name, last name, address, salary, and telephone number)
  - Primary key: staff number
  - Candidate key: staff number, telephone number
- Equipment (**equipment identifier**, description, usage, cost)
  - Primary key: equipment identifier
  - Candidate key: equipment identifier, description

  (Equipment are assumed to perform unique roles for efficiency)
- Service (**serviceNo**, start day, end day, start time, end time)
  - Primary key: serviceNo
  - Candidate key: serviceNo

Logical Data Model

Derived relations from the conceptual model.

We give each attribute a unique name to prevent data from being misplaced. Additionally, for adjacent one-to-many relationships or zero-to-many relationships we will draw tables for the relations involved. These relations will include the parent keys from adjacent entities



Normalization

- **1NF (Flattening the UNF Table)**

  Since tables were determined in the conceptual model, we can represent them in 1NF form

  - Clients(**clientNo**, cFirstName, cLastName, cAddress, cTeleNum)
  - Employees(**empNo ,**eFirstName, eLastName, eAddress, salary, eTeleNum)
  - Equipment(**eqID** ,desc, usage, cost)

- Service (**serviceNo**,startDay, endDay, startTime, endTime)
- Perfoms (**serviceNo, empNo**, hours)
- Determines (**serviceNo, eqID**, lastUseDate, lastServiceDate)

Functional Dependencies

clientNo > cFirstName, cLastName, cAddress, cTeleNum

empNo> eFirstName, eLastName, eAddress, salary, eTeleNum

eqID > desc, usage, cost

serviceNo > startDay, endDay, numHours

serviceNo, empNo > hours

serviceNo, eqID > lastUseDate, lastServiceDate

- **2NF (Make new tables based on the partial dependencies identified)**

  No partial dependencies were identified so no new tables are made.

  - Clients(**clientNo**, cFirstName, cLastName, cAddress, cTeleNum)
  - Employees(**empNo ,**eFirstName, eLastName, eAddress, salary, eTeleNum)
  - Equipment(**eqID** ,desc, usage, cost)
  - Service (**serviceNo**,startDay, endDay, startTime, endTime)
  - Perfoms (**serviceNo, empNo**, hours)
  - Determines (**serviceNo, eqID**, lastUseDate, lastServiceDate)
- 3NF (Removal of Transitive Dependencies, Putting those dependencies in individual table)

  No transitive dependencies were identified so no new tables are made.

  - Clients(**clientNo**, cFirstName, cLastName, cAddress, cTeleNum)
  - Employees(**empNo ,**eFirstName, eLastName, eAddress, salary, eTeleNum)
  - Equipment(**eqID** ,desc, usage, cost)
  - Service (**serviceNo**,startDay, endDay, startTime, endTime)
  - Perfoms (**serviceNo, empNo**, hours)
  - Determines (**serviceNo, eqID**, lastUseDate, lastServiceDate)

Since Services table has a one to many relationship with Client it will have one foreign key

- Service (**serviceNo**,startDay, endDay, startTime, endTime, clientNo)

The relationship tables help to manage entity relationship since they contain the primary keys of their adjacent tables

From the above, we can conlude that our table is already in 3NF form hence we can proceed to the implementation phase of the project

Validating Against User Transactions

We validate the Cardboard Box Company and P.Nutall through our system as the clients given in our question:

Client

clientNo	cFirstName	cLastName	cAddress	cTeleNum
1	Patrick	Nutall	1230 Hemingway Avenue	3067894328
2	The Cardboard Box	Company	114 Stanford Heights	2134874680

We pass sample employees through Employee table

Employee

empNo	eFirstName	eLastName	eAddress	eTeleNum
1	Chadwick	Pollet	1230 Hemingway Avenue	3067894328
2	George	Company	114 Stanford Heights	2134874680

We pass sample services through Services table

Service

serviceNo	startDay	endDay	startTime	endTime
1	M	F	2020-12-25	2020-12-22
2	T	TH	2020-12-25	2020-12-21
3	F	S	2020-12-25	2020-12-25

We pass elements into equipment table

Equipment

Aa serviceNo	desc	usage	cost
1	Mop	2	10
2	Rug	3	12
3	Paint	1	12

Since our base table passed, the test, we can proceed to finding constraints and implementation.

Constraints

Integrity constraints

i. **Primary key constraints.**

- Primary keys for each entity cannot be null and must be unique for each value

ii. **Referential integrity/Foreign key constraints.**

- If a foreign key contains a value, that value must refer to an existing tuple in the parent relation

iii. **Alternate key constraints (if any).**

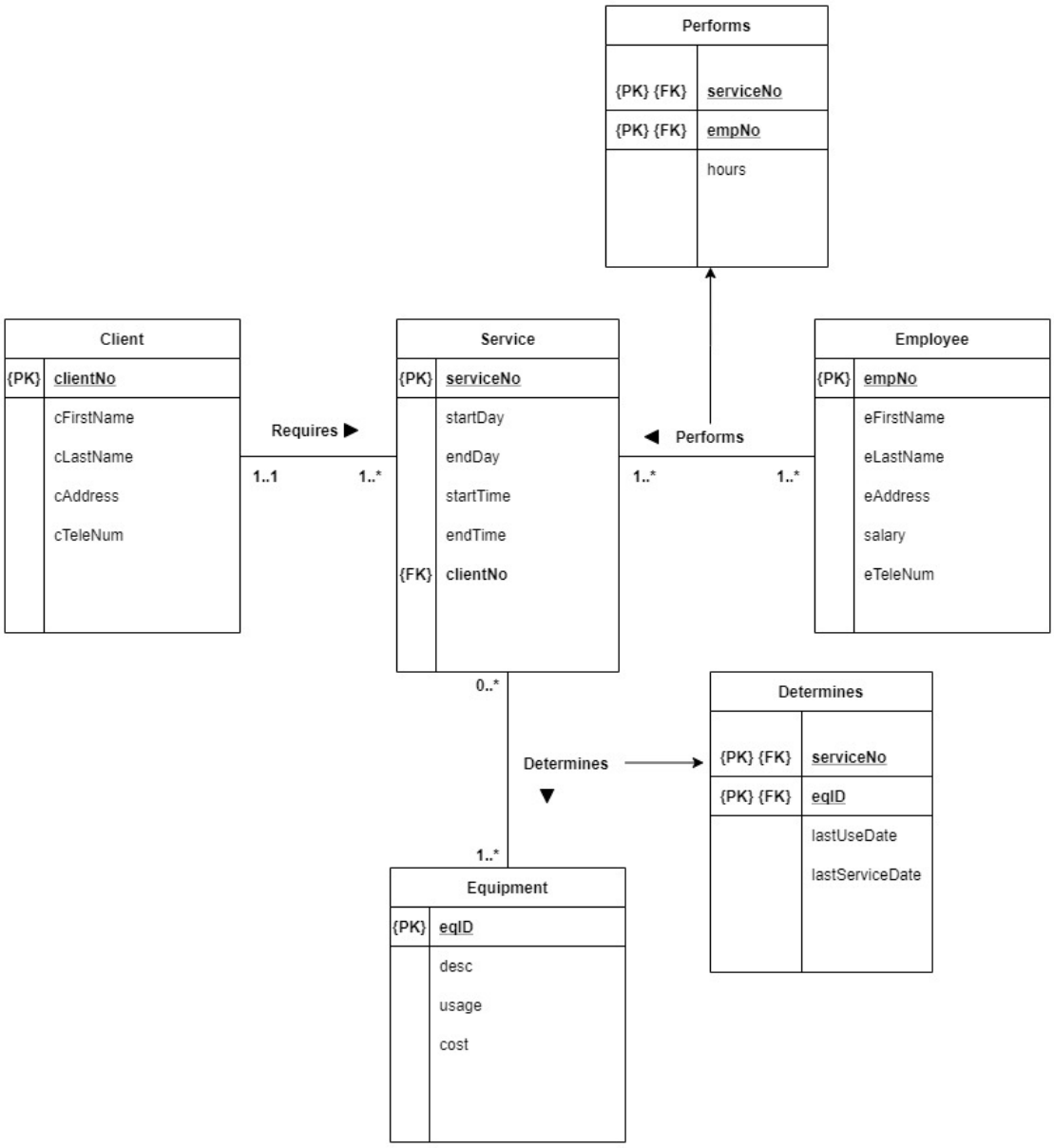
- No alternate keys were determined as of this implementation

iv. **General constraints (if any).**

- Services must start and end in the future, not in the past
- A service can only be stated by one client
- The end day must be greater than start day for Services.
- The end time must be greater than the start time
- Equipment description and usage are unique values for the purpose of efficiency
- Company names will occupy the First Name column and the word "Company" will occupy the last in situations where a company is the known client

- Telephone number must be less than 16 digits long
- Cleaning services are not offerred on weekends
- Days of the week are represented as letters

Final Implementation



Oracle DBMS Implementation

`BusyBee.sql`

```
--Uncomment Drop tables if tables already exist
--DROP TABLE DETERMINES;
--DROP TABLE PERFORMS;
--DROP TABLE EQUIPMENT;
--DROP TABLE SERVICE;
--DROP TABLE EMPLOYEE;
--DROP TABLE CLIENT;

--Client Table
CREATE TABLE Client
 (
  clientNo INT,
  cFirstName VARCHAR(100) ,
  cLastName VARCHAR(100),
  cAddress VARCHAR(100),
  cTeleNum INT CHECK (LENGTH(cTeleNum) < 16),
  PRIMARY KEY(clientNo)
);

-- Employee Table
CREATE TABLE Employee
 (
  empNo INT,
  eFirstName VARCHAR(100) ,
  eLastName VARCHAR(100),
  eAddress VARCHAR(100),
  salary INT ,
  eTeleNum VARCHAR(100) CHECK (LENGTH(eTeleNum) < 16),
  PRIMARY KEY(empNo)
);

--Service Table
CREATE TABLE Service
 (
  serviceNo INT,
  startDay VARCHAR(100) CHECK(startDay IN ('M', 'T', 'W','TH', 'F')),
  endDay VARCHAR(100) CHECK(endDay IN ('M', 'T', 'W','TH', 'F')),
  startTime DATE,
  endTime DATE,
  clientNo INT,
  empNo INT,
  PRIMARY KEY(serviceNo),
  CONSTRAINT SVAL CHECK (endTime > startTime),
  FOREIGN KEY (clientNo) REFERENCES Client(clientNo) ON DELETE SET NULL
);

-- Equipment Table
CREATE TABLE Equipment
 (
  eqID INT,
  descp VARCHAR(300),
  usage INT,
  cost INT,
  serviceNo INT,
PRIMARY KEY(eqID)
);
```

```
-- Performs Table
CREATE TABLE Performs
(
serviceNo INT,
empNo INT,
hours INT,
PRIMARY KEY(serviceNo, empNo),
FOREIGN KEY (serviceNo) REFERENCES Service(serviceNo) ON DELETE SET NULL,
FOREIGN KEY (empNo) REFERENCES Employee(empNo) ON DELETE SET NULL
);

-- Determines Table
CREATE TABLE Determines
(
serviceNo INT,
eqID INT,
lastUseDate DATE,
lastServiceDate DATE,
PRIMARY KEY(serviceNo, eqID),
FOREIGN KEY (serviceNo) REFERENCES Service(serviceNo) ON DELETE SET NULL,
FOREIGN KEY (eqID) REFERENCES Equipment(eqID) ON DELETE SET NULL
);
```

```
INSERT INTO CLIENT VALUES(1,'Patrick','Nutall','1230 Hemingway Avenue',306789432
8);
INSERT INTO CLIENT VALUES(2,'The Cardboard Box','Company','114 Stanford Heights',
2134874680);
INSERT INTO CLIENT VALUES(3,'Jerry','Jones','698 South Parkway',2067204328);
INSERT INTO CLIENT VALUES(4,'Dantey','Tope','1621 Dawn Terrace',3094328234);
INSERT INTO CLIENT VALUES(5,'Pogba','Paul','1230 Hemingway Avenue',3054986903);
INSERT INTO CLIENT VALUES(6,'Terry','Nero','1615 Schlimgen Crossing',4302099078);
----------------------------
INSERT INTO EMPLOYEE VALUES(1,'Chadwick','Pollett','8696 Victoria Court',286.67,9
167046280);
INSERT INTO EMPLOYEE VALUES(2,'Alexandros','Antunes','9 Alpine Crossing',418.07,8
745093029);
INSERT INTO EMPLOYEE VALUES(3,'Averill','Tomblett','34 Shopko Park',48.05,2269440
178);
INSERT INTO EMPLOYEE VALUES(4,'Bernard','Pretswell','647 Sauthoff Court',5359.80,
6142641343);
INSERT INTO EMPLOYEE VALUES(5,'Claudetta','Caherny','1 Carberry Court',9113.12,95
55627427);
INSERT INTO EMPLOYEE VALUES(6,'Richmond','Molesworth','1294 Pawling Place',277.3
8,4493208964);
----------------------------
INSERT INTO SERVICE VALUES(1,'M','F',TO_DATE('2020-12-07','YYYY-MM-DD'),TO_DATE
('2020-12-11','YYYY-MM-DD'),1,1);
--INSERT INTO SERVICE VALUES(2,'T','Th',TO_DATE('2020-12-08','YYYY-MM-DD'),TO_DAT
E('2020-12-10','YYYY-MM-DD'),1,2);
INSERT INTO SERVICE VALUES(3,'W','F',TO_DATE('2020-12-09','YYYY-MM-DD'),TO_DATE
('2020-12-11','YYYY-MM-DD'),2,2);
INSERT INTO SERVICE VALUES(4,'TH','M',TO_DATE('2020-12-10','YYYY-MM-DD'),TO_DATE
('2020-12-13','YYYY-MM-DD'),3,3);
INSERT INTO SERVICE VALUES(5,'F','T',TO_DATE('2020-12-11','YYYY-MM-DD'),TO_DATE
```

```
('2020-12-15','YYYY-MM-DD'),4,4);
--INSERT INTO SERVICE VALUES(6,'W','Th',TO_DATE('2020-12-16','YYYY-MM-DD'),TO_DAT
E('2020-12-18','YYYY-MM-DD'),1,2);
INSERT INTO SERVICE VALUES(7,'T','F',TO_DATE('2020-12-22','YYYY-MM-DD'),TO_DATE
('2020-12-25','YYYY-MM-DD'),1,1);
--INSERT INTO SERVICE VALUES(8,'F','S',TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE
('2020-12-26','YYYY-MM-DD'),1,5);
----------------------------
INSERT INTO EQUIPMENT VALUES(1,'Carpet Cleaner',3,20,1);
INSERT INTO EQUIPMENT VALUES(2,'Mop',2,40,6);
INSERT INTO EQUIPMENT VALUES(3,'Stone Waxer',1,79,6);
INSERT INTO EQUIPMENT VALUES(4,'Buffer',2,89,2);
INSERT INTO EQUIPMENT VALUES(5,'Sulphuric Acid',3,24,2);
INSERT INTO EQUIPMENT VALUES(6,'Oil',3,90,3);
INSERT INTO EQUIPMENT VALUES(7,'Polish',2,89,2);
----------------------------
INSERT INTO PERFORMS VALUES(1,1,4);
--INSERT INTO PERFORMS VALUES(2,2,4);
INSERT INTO PERFORMS VALUES(3,3,4);
INSERT INTO PERFORMS VALUES(1,2,5);
INSERT INTO PERFORMS VALUES(4,4,3);
INSERT INTO PERFORMS VALUES(5,5,3);
----------------------------
INSERT INTO DETERMINES VALUES(1,1,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('202
0-12-26','YYYY-MM-DD'));
--INSERT INTO DETERMINES VALUES(2,2,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('2
020-12-26','YYYY-MM-DD'));
INSERT INTO DETERMINES VALUES(3,3,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('202
0-12-26','YYYY-MM-DD'));
INSERT INTO DETERMINES VALUES(5,5,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('202
0-12-26','YYYY-MM-DD'));
INSERT INTO DETERMINES VALUES(1,3,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('202
0-12-26','YYYY-MM-DD'));
INSERT INTO DETERMINES VALUES(1,2,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('202
0-12-26','YYYY-MM-DD'));
--INSERT INTO DETERMINES VALUES(1,2,TO_DATE('2020-12-25','YYYY-MM-DD'),TO_DATE('2
020-12-26','YYYY-MM-DD'));
```

Output

Client

CLIENT...	CFIRSTNAME	CLASTNAME	CADDRESS	CTELENUM
1	Patrick	Nutall	1230 Hemingway Avenue	3067894328
2	The Cardboard Box	Company	114 Stanford Heights	2134874680
3	Jerry	Jones	698 South Parkway	2067204328
4	Dantey	Tope	1621 Dawn Terrace	3094328234
5	Pogba	Paul	1230 Hemingway Avenue	3054986903
6	Terry	Nero	1615 Schlimgen Crossing	4302099078

Employee

EMPNO	EFIRSTNAME	ELASTNAME	EADDRESS	SALARY	ETELENUM
1	Chadwick	Pollett	8696 Victoria Court	287	9167046280
2	Alexandros	Antunes	9 Alpine Crossing	418	8745093029
3	Averill	Tomblett	34 Shopko Park	48	2269440178
4	Bernard	Pretswell	647 Sauthoff Court	5360	6142641343
5	Claudetta	Caherny	1 Carberry Court	9113	9555627427
6	Richmond	Molesworth	1294 Pawling Place	277	4493208964

Service

	SERVICENO	STARTDAY	ENDDAY	STARTTIME	ENDTIME	CLIENTNO	EMPNO
1	1	M	F	07-DEC-20	11-DEC-20	1	1
2	3	W	F	09-DEC-20	11-DEC-20	2	2
3	4	TH	M	10-DEC-20	13-DEC-20	3	3
4	5	F	T	11-DEC-20	15-DEC-20	4	4
5	7	T	F	22-DEC-20	25-DEC-20	1	1

Equipment

EQID	DESCP	USAGE	COST	SERVICENO
1	Carpet Cleaner	3	20	1
2	Mop	2	40	6
3	Stone Waxer	1	79	6
4	Buffer	2	89	2
5	Sulphuric Acid	3	24	2
6	Oil	3	90	3
7	Polish	2	89	2

Performs

SERVICENO	EMPNO	HOURS
1	1	4
3	3	4
1	2	5
4	4	3
5	5	3

Determines

SERVICENO	EQID	LASTUSEDATE	LASTSERVICEDATE
1	1	25-DEC-20	26-DEC-20
3	3	25-DEC-20	26-DEC-20
5	5	25-DEC-20	26-DEC-20
1	3	25-DEC-20	26-DEC-20
1	2	25-DEC-20	26-DEC-20

Embedded SQL

`connect_oracle.py`

These queries were implemented in Python

```
-- Print all Employees
SELECT * FROM EMPLOYEE;

--Print all Clients
SELECT * FROM CLIENT;

--Print all Equipments
SELECT * FROM EQUIPMENT;

-- Print all Services
SELECT * FROM SERVICE;

-- Print Master Table
SELECT *
FROM CLIENT c, SERVICE s, EQUIPMENT e, Determines d
WHERE(c.clientNo = s.clientNo AND s.serviceNo = d.serviceNo AND d.eqID = e.eqID);
```

Run `connect_oracle.py` to enjoy the experience

```
Welcome to the Busy Bee Cleaning Company! We are all set for the annual audit. Kindly let me know if you have any questions
What would you like to see? Respond with the options below to view requisite info!
Type 1 to Print all Employees
Type 2 to Print all Clients
Type 3 to Print all Equipments
Type 4 to Print all Services
Type 5 to Print Master Table
Type 6 to Quit
All other values will be rejeted!
Enter your value:
```

`Sample Test`

```
Enter your value: 1
   EMPNO  EFIRSTNAME   ELASTNAME             EADDRESS  SALARY    ETELENUM
0      1    Chadwick     Pollett  8696 Victoria Court     287  9167046280
1      2  Alexandros     Antunes    9 Alpine Crossing     418  8745093029
2      3     Averill    Tomblett       34 Shopko Park      48  2269440178
3      4     Bernard   Pretswell   647 Sauthoff Court    5360  6142641343
4      5   Claudetta     Caherny     1 Carberry Court    9113  9555627427
5      6    Richmond  Molesworth   1294 Pawling Place     277  4493208964
Index(['EMPNO', 'EFIRSTNAME', 'ELASTNAME', 'EADDRESS', 'SALARY', 'ETELENUM'], dtype='object')
Enter another value: 2
   CLIENTNO         CFIRSTNAME CLASTNAME                CADDRESS    CTELENUM
0         1            Patrick    Nutall    1230 Hemingway Avenue  3067894328
1         2  The Cardboard Box   Company     114 Stanford Heights  2134874680
2         3              Jerry     Jones        698 South Parkway  2067204328
3         4             Dantey      Tope        1621 Dawn Terrace  3094328234
4         5              Pogba      Paul    1230 Hemingway Avenue  3054986903
5         6              Terry      Nero  1615 Schlimgen Crossing  4302099078
Index(['CLIENTNO', 'CFIRSTNAME', 'CLASTNAME', 'CADDRESS', 'CTELENUM'], dtype='object')
Enter another value: 3
   EQID           DESCP  USAGE  COST  SERVICENO
0     1  Carpet Cleaner      3    20          1
1     2             Mop      2    40          6
2     3     Stone Waxer      1    79          6
3     4          Buffer      2    89          2
4     5  Sulphuric Acid      3    24          2
5     6             Oil      3    90          3
6     7          Polish      2    89          2
Index(['EQID', 'DESCP', 'USAGE', 'COST', 'SERVICENO'], dtype='object')
Enter another value: 4
   SERVICENO STARTDAY ENDDAY  STARTTIME    ENDTIME  CLIENTNO  EMPNO
0          1        M      F 2020-12-07 2020-12-11         1      1
1          3        W      F 2020-12-09 2020-12-11         2      2
2          4       TH      M 2020-12-10 2020-12-13         3      3
3          5        F      T 2020-12-11 2020-12-15         4      4
4          7        T      F 2020-12-22 2020-12-25         1      1
Index(['SERVICENO', 'STARTDAY', 'ENDDAY', 'STARTTIME', 'ENDTIME', 'CLIENTNO',
       'EMPNO'],
      dtype='object')
```

Acknowledgements

The above code was implemented and designed by Isaac Kofi Attuah.