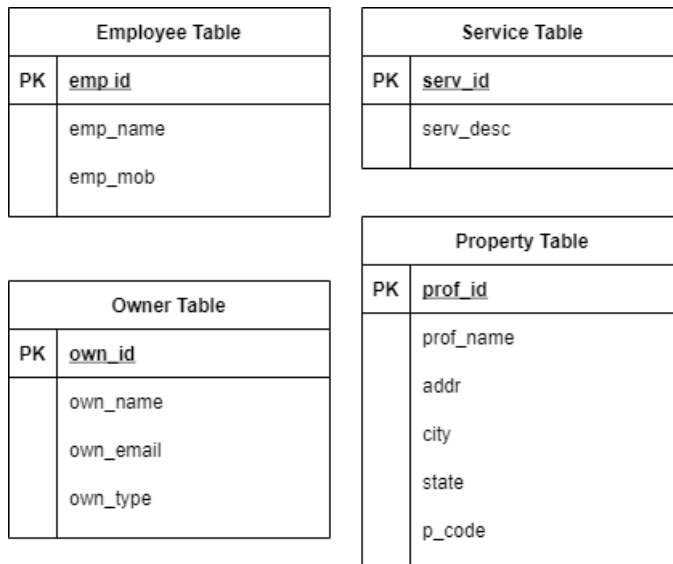


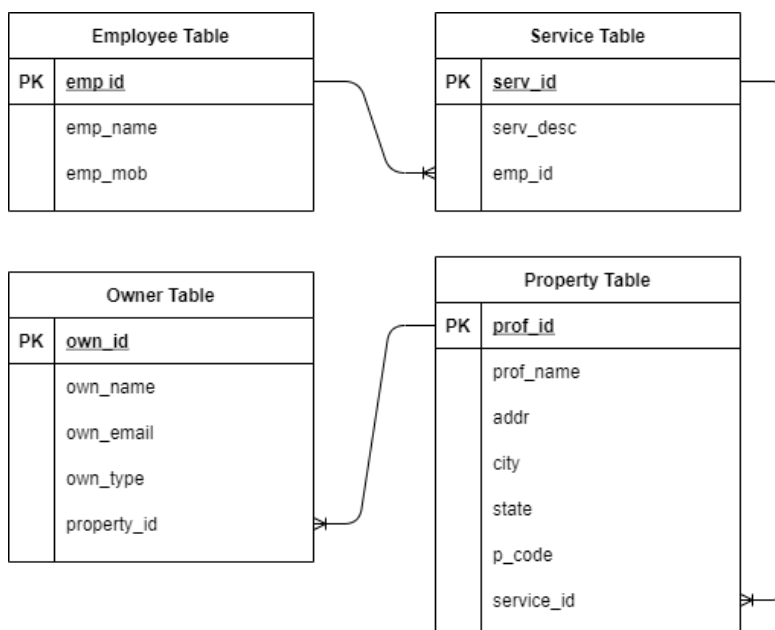
a. Using the table and data in Figure 1, state assumptions about dependencies among the columns of the table. Justify your assumptions on the basis of the sample data and also on the basis of what you know about service business.

in the figure 1 Property Column is totally depend upon owners' column, service of each property is depend upon the type of service column and whom is providing by the employee column. So will separate and create Property, Owner, Service and Employees Tables. Below is my example



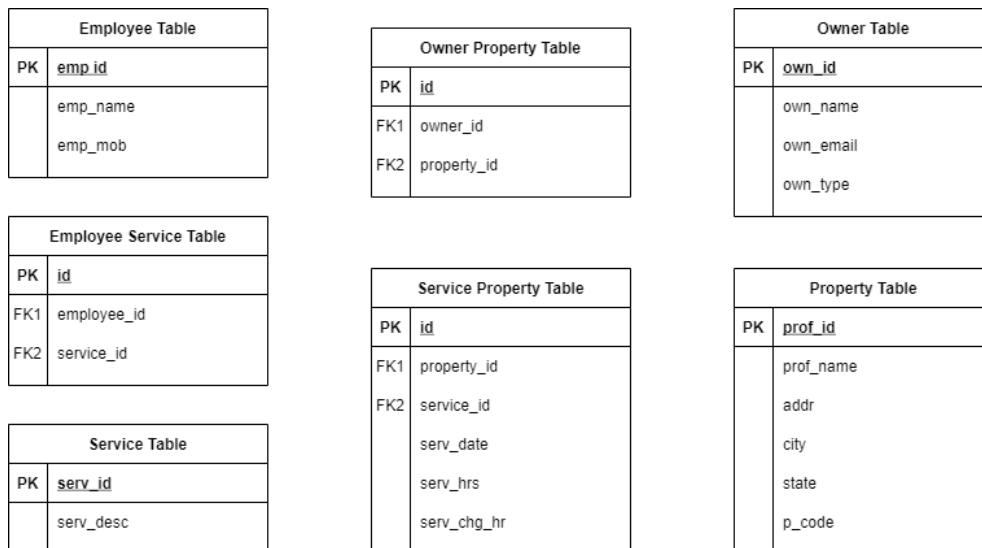
b. Employing the dependencies stated in in (a), write a relational schema and draw a dependency diagram for the table in 1NF. The dependency diagram must have proper labels for all functional, partial and/or transitive dependencies, if there are any.

Figure 1 is in already in 1NF because there are no multi-valued attributes. Following is the relational schema for the table:

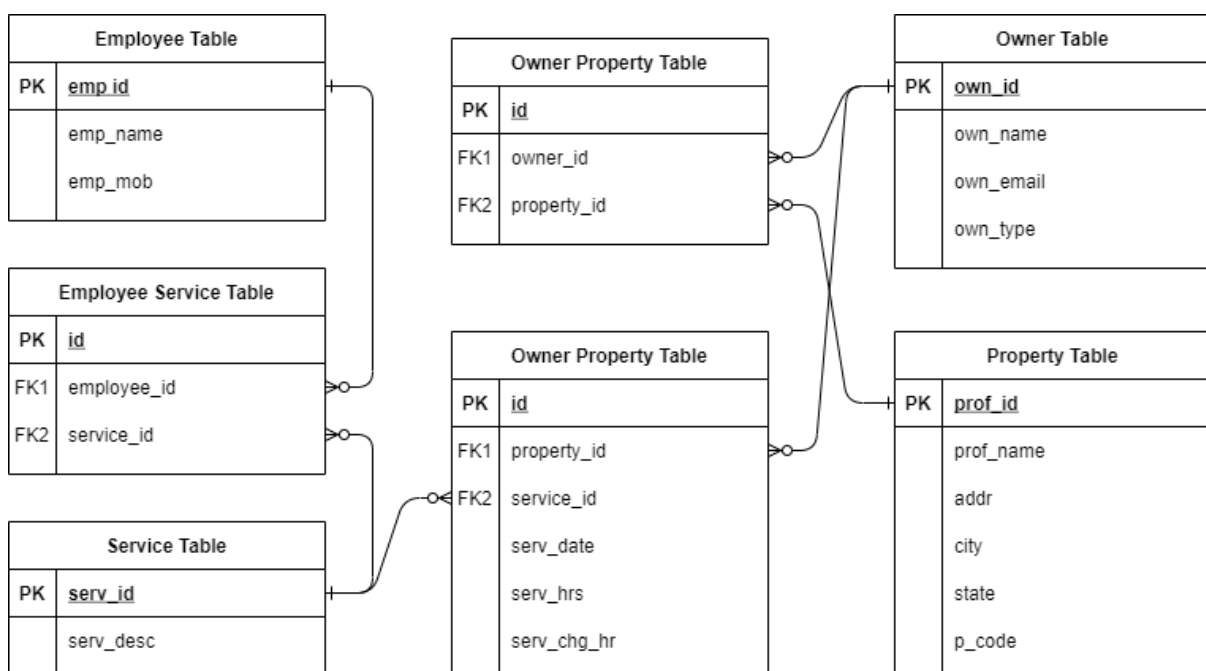


c. Break up the dependency diagram you drew in (b) to produce dependency diagrams that are in 3 NF and also write the relational schemas for the table in 3NF. Make sure the new dependency diagrams contain attributes that meet proper design criteria; that is, make sure that there are no multivalued attributes, that the naming conventions are met, and so on.)

From above diagram, I have created three more tables: Employee Service Table, Owner Property Table, Service Property Table. In above table, there were chances of redundancy to occur. By separating dependency tables furthermore, schema is in 3NF now, and chances of redundancy are quite less.



d. Develop an E-R diagram based on the task done in (c). Use crow-foot style and specify entities, attributes, relationships, and multiplicity. Justify the decisions you make regarding minimum and maximum cardinality. Describe how you would go about validating this model.



e. Using the E-R diagram you developed in(d), convert it to a relational design. Document your design as follows:

- Specify tables, primary keys, and foreign keys.

Sro No	Table Name	Primary Key	Foreign Key
1	employee_table	emp_id	-
2	service_table	serv_id	-
3	employee_service_table	id	employee_id, service_id
4	owner_table	own_id	-
5	propety_table	prof_id	-
6	owner_property_table	Id	owner_id, property_id
7	service_proper_table	Id	property_id, service_id

- Describe how you have represented weak entities, if there are any.
 - a. There is no week entities in all the tables. A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.
- Document relationship enforcement.

f. Create a database using MySQL, with primary keys, foreign keys, and other attributes mentioned for each table developed in (e) using proper constraints.

```
CREATE Database database_project;
```

```
USE database_project;
```

```
CREATE TABLE `employee_table` (
  `emp_id` int(11) NOT NULL PRIMARY KEY,
  `emp_name` varchar(150) DEFAULT NULL,
  `emp_mob` varchar(15) DEFAULT NULL
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `service_table` (
  `serv_id` int(11) NOT NULL PRIMARY KEY,
  `serv_desc` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `employee_service_table` (  
  `id` int(11) NOT NULL PRIMARY KEY,  
  `employee_id` int(11) DEFAULT NULL,  
  `service_id` int(11) DEFAULT NULL,  
  CONSTRAINT `fk_employee_id` FOREIGN KEY (`employee_id`) REFERENCES `employee_table`  
  (`emp_id`),  
  CONSTRAINT `fk_service_id` FOREIGN KEY (`service_id`) REFERENCES `service_table` (`serv_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `owner_table` (  
  `own_id` int(11) NOT NULL PRIMARY KEY,  
  `own_name` varchar(150) DEFAULT NULL,  
  `own_email` varchar(150) DEFAULT NULL,  
  `own_type` varchar(150) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `property_table` (  
  `prof_id` int(11) NOT NULL PRIMARY KEY,  
  `prof_name` varchar(150) DEFAULT NULL,  
  `addr` varchar(150) DEFAULT NULL,  
  `city` varchar(150) DEFAULT NULL,  
  `state` varchar(150) DEFAULT NULL,  
  `p_code` varchar(150) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `owner_property_table` (  
  `id` int(11) NOT NULL PRIMARY KEY,  
  `owner_id` int(11) DEFAULT NULL,  
  `property_id` int(11) DEFAULT NULL,  
  CONSTRAINT `fk_owner_id` FOREIGN KEY (`owner_id`) REFERENCES `owner_table` (`own_id`),  
  CONSTRAINT `fk_property_id` FOREIGN KEY (`property_id`) REFERENCES `property_table` (`prof_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```

CREATE TABLE `service_propert_table` (
  `id` int(11) NOT NULL PRIMARY KEY,
  `property_id` int(11) NOT NULL,
  `service_id` int(11) NOT NULL,
  `serv_date` date DEFAULT NULL,
  `serv_hrs` float DEFAULT NULL,
  `serv_chg_hr` float DEFAULT NULL,
  CONSTRAINT `fk_property_id` FOREIGN KEY (`property_id`) REFERENCES `propety_table`
  (`prof_id`),
  CONSTRAINT `fk_service_id` FOREIGN KEY (`service_id`) REFERENCES `service_table` (`serv_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

g. Create SQL statements to satisfy the following:

1. Write SQL statements to insert at least 7 rows of data into each of the table created in task (f). You may use the sample data provided in Figure 1 for this task

```

INSERT INTO `employee_table` (`emp_id`, `emp_name`, `emp_mob`) VALUES
(101, 'Robert McGraw', '401234567'),
(102, 'Mike Burrows', '414563453'),
(103, 'Ben Keen', '414563453');

```

```

INSERT INTO `owner_table` (`own_id`, `own_name`, `own_email`, `own_type`) VALUES
(1, 'Helmet Jones', 'helmet.jones@gmail.com', 'Individual'),
(2, 'Kenny Blackmore', 'Kenny.blackmore@gmail.com', 'Coroporation'),
(3, 'Barry Wilson', 'barry.wilson@gmail.com', 'Individual'),
(4, 'Craig Noon', 'carig.noon@gmail.com', 'Individual'),
(5, 'Wendy Sullivan', 'wendy.sullivan@gmail.com', 'Coroporation'),
(6, 'Jim Bruno', 'jim.gruno@brunoandosone.com', 'Individual'),
(7, 'Peter Emerson', 'peter.emerson@emersonflyingjet.com', 'Coroporation');

```

```

NSERT INTO `propety_table` (`prof_id`, `prof_name`, `addr`, `city`, `state`, `p_code`) VALUES
(10, 'Eastlake Building', '123 Eastlake', 'Maroona', 'VIC', '3210'),
(11, 'Earls Court', '235 East West', 'Portland', 'VIC', '3330'),
(12, 'Barry Wilson', '75 West Bound', 'Dundee', 'VIC', '3500'),

```

```
(13, 'Jack and Jill', '105 Young', 'Freshy', 'VIC', '3350'),  
(14, 'Cosey Here', '144 Sensible', 'Sunshine', 'VIC', '3456'),  
(15, 'Bruno & Son', '66/30 Palm Beach', 'Newland', 'VIC', '3333'),  
(16, 'Emerson Flying Jet', '707 Ardunino', 'Mega', 'VIC', '3256');
```

```
INSERT INTO `service_table` (`serv_id`, `serv_desc`) VALUES  
(1, 'Garden Service'),  
(2, 'Lawn Mow');
```

```
INSERT INTO `employee_service_table` (`id`, `employee_id`, `service_id`) VALUES  
(1, 101, 1),  
(2, 102, 2),  
(3, 101, 1),  
(4, 102, 2),  
(5, 103, 1),  
(6, 102, 2),  
(7, 103, 2),  
(8, 101, 2);
```

```
INSERT INTO `owner_property_table` (`id`, `owner_id`, `property_id`) VALUES  
(1, 1, 10),  
(2, 2, 11),  
(3, 3, 12),  
(4, 4, 13),  
(5, 5, 14),  
(6, 6, 15),  
(7, 7, 16);
```

```
INSERT INTO `service_proper_table` (`id`, `property_id`, `service_id`, `serv_date`, `serv_hrs`,  
`serv_chg_hr`) VALUES  
(1, 10, 1, '2020-05-05', 2.5, 75),  
(2, 11, 2, '2020-05-07', 2, 55),  
(3, 12, 1, '2020-05-07', 2.5, 85),
```

(4, 11, 2, '2020-05-12', 1.5, 50),
(5, 13, 1, '2020-05-19', 3, 85),
(6, 14, 2, '2020-05-17', 2, 90),
(7, 15, 2, '2020-05-14', 2, 55),
(8, 16, 2, '2020-05-10', 1, 50);

2. Write SQL statements to list all columns of all tables

SHOW COLUMNS FROM employee_table;

Your SQL query has been executed successfully.

```
SHOW COLUMNS FROM employee_table;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	NO	PRI	NULL	
emp_name	varchar(150)	YES		NULL	
emp_mob	varchar(15)	YES		NULL	

SHOW COLUMNS FROM owner_table;

Your SQL query has been executed successfully.

```
SHOW COLUMNS FROM owner_table;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
own_id	int(11)	NO	PRI	NULL	
own_name	varchar(150)	YES		NULL	
own_email	varchar(150)	YES		NULL	
own_type	varchar(150)	YES		NULL	

SHOW COLUMNS FROM property_table;

Your SQL query has been executed successfully.

SHOW COLUMNS FROM property_table;

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
prof_id	int(11)	NO	PRI	NULL	
prof_name	varchar(150)	YES		NULL	
addr	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
state	varchar(150)	YES		NULL	
p_code	varchar(150)	YES		NULL	

SHOW COLUMNS FROM service_table;

Your SQL query has been executed successfully.

SHOW COLUMNS FROM service_table;

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
serv_id	int(11)	NO	PRI	NULL	
serv_desc	varchar(255)	YES		NULL	

SHOW COLUMNS FROM employee_service_table;

Your SQL query has been executed successfully.

SHOW COLUMNS FROM employee_service_table;

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Re](#)

+ Options

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
employee_id	int(11)	YES	MUL	NULL	
service_id	int(11)	YES	MUL	NULL	

SHOW COLUMNS FROM owner_property_table;

Your SQL query has been executed successfully.

```
SHOW COLUMNS FROM owner_property_table;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
owner_id	int(11)	YES	MUL	NULL	
property_id	int(11)	YES	MUL	NULL	

SHOW COLUMNS FROM service_proPERT_table;

Your SQL query has been executed successfully.

```
SHOW COLUMNS FROM service_proPERT_table;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)] [[Refresh](#)]

+ Options

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
property_id	int(11)	NO	MUL	NULL	
service_id	int(11)	NO	MUL	NULL	
serv_date	date	YES		NULL	
serv_hrs	float	YES		NULL	
serv_chg_hr	float	YES		NULL	

3. Write SQL statements to list the name and mobile phone for all employees

SELECT emp_name,emp_mob FROM employee_table ORDER BY emp_name

Options

				emp_name	emp_mob
<input type="checkbox"/>	Edit	Copy	Delete	Ben Keen	414563453
<input type="checkbox"/>	Edit	Copy	Delete	Mike Burrows	414563453
<input type="checkbox"/>	Edit	Copy	Delete	Robert McGraw	401234567

4. Write SQL statements to list the name and email address for all owners.

```
SELECT own_name,own_email FROM owner_table ORDER BY own_name ASC
```

<input type="checkbox"/>				Barry Wilson	barry.wilson@gmail.com
<input type="checkbox"/>				Craig Noon	carig.noon@gmail.com
<input type="checkbox"/>				Helmet Jones	helmet.jones@gmail.com
<input type="checkbox"/>				Jim Bruno	jim.gruno@brunoandosone.com
<input type="checkbox"/>				Kenny Blackmore	Kenny.blackmore@gmail.com
<input type="checkbox"/>				Peter Emerson	peter.emerson@emersonflyingjet.com
<input type="checkbox"/>				Wendy Sullivan	wendy.sullivan@gmail.com

5. Write SQL statements to list the property name, address, state, and post code for all properties.

```
SELECT prof_name,addr,city,state,p_code FROM propety_table ORDER BY prof_name ASC
```

<input type="checkbox"/>				Barry Wilson	75 West Bound	Dundee	VIC	3500
<input type="checkbox"/>				Bruno & Son	66/30 Palm Beach	Newland	VIC	3333
<input type="checkbox"/>				Cosey Here	144 Sensible	Sunshine	VIC	3456
<input type="checkbox"/>				Earls Court	235 East West	Portland	VIC	3330
<input type="checkbox"/>				Eastlake Building	123 Eastlake	Maroona	VIC	3210
<input type="checkbox"/>				Emerson Flying Jet	707 Ardunino	Mega	VIC	3256
<input type="checkbox"/>				Jack and Jill	105 Young	Freshy	VIC	3350

☐ Check all With selected: Edit Copy Delete Export

6. Write SQL statements to list all owner names and their property owned.

```
SELECT ot.own_name,pt.prof_name
```

```
FROM owner_property_table as opt
```

```
INNER JOIN owner_table as ot ON opt.owner_id=ot.own_id
```

```
INNER JOIN propety_table as pt ON opt.property_id=pt.prof_id
```

```
ORDER BY ot.own_name ASC
```

own_name	prof_name
Barry Wilson	Barry Wilson
Craig Noon	Jack and Jill
Helmet Jones	Eastlake Building
Jim Bruno	Bruno & Son
Kenny Blackmore	Earls Court
Peter Emerson	Emerson Flying Jet
Wendy Sullivan	Cosey Here

7. Write SQL statements to determine how many times of 'Lawn Mow' have been done at 'Earls Courts'?

```
SELECT
COUNT(*) no_of_times
FROM service_propert_table as spt
INNER JOIN service_table as st ON st.serv_id=spt.service_id
INNER JOIN propety_table as pt ON pt.prof_id=spt.property_id
WHERE st.serv_desc='Lawn Mow' and pt.prof_name='Earls Court'
```

no_of_times
2

8. Write SQL statements to list name of employees who have provided 'Garden Service to a property owned by 'Individual'.

```
SELECT
e.emp_name
FROM service_propert_table as spt
INNER JOIN service_table as s ON s.serv_id=spt.service_id
INNER JOIN propety_table as p ON p.prof_id=spt.property_id
INNER JOIN owner_property_table as opt ON opt.property_id=spt.property_id
INNER JOIN owner_table as o ON o.own_id=opt.owner_id
INNER JOIN employee_service_table as est ON est.service_id=s.serv_id
INNER JOIN employee_table as e ON e.emp_id=est.employee_id
WHERE s.serv_desc='Garden Service' and o.own_type='Individual'
GROUP BY e.emp_id
```

emp_name
Robert McGraw
Ben Keen

9. Write SQL statements to list total service charge amounts for each property. Note that a service charge is calculated by multiplying the service hour by the service charged per hour.

SELECT

p.prof_name,(spt.serv_hrs*spt.serv_chg_hr) as service_charges

FROM service_propert_table as spt

INNER JOIN propety_table as p ON p.prof_id=spt.property_id

prof_name	service_charges
Eastlake Building	187.5
Earls Court	110
Barry Wilson	212.5
Earls Court	75
Jack and Jill	255
Cosey Here	180
Bruno & Son	110
Emerson Flying Jet	50