

Assignment 2

Refer to the assignment #1, Problem #1, As we know Meeseeks Interior Design Farm (MIDF) used to keep their records in paper documents. Given below is a portion of their table of data.

Employee Information Table

Employee_id	Name	Hourly Salary	Sex	Date of birth	Supervisor id	Branch name	Branch Address	Branch manager id	Managing since
124	Rick Astley	18	M	02/06/1966	64	Illinois	3398 Braxton Street	2	NULL
131	Dorothy Golden	18	F	30/08/1969	64	Illinois	3398 Braxton Street	2	NULL
137	Morty Smith	18	M	24/05/1976	11	Seattle	3862 Mutton Town Road	1	NULL
64	Cathrine Gonzalez	20	F	24/11/1959	11	Illinois	3398 Braxton Street	2	NULL
76	Satoshi Nakamoto	18	M	25/08/1975	11	Illinois	3398 Braxton Street	2	3/9/2012
77	John Lewis	17	M	05/03/1998	64	Seattle	3862 Mutton Town Road	1	NULL
65	Russell Bueno	17	M	25/04/1966	64	Seattle	3862 Mutton Town Road	1	NULL
122	Kristie Clark	30	F	09/05/1975	2	California	93401 San Luis Obispo	123	17/5/2010
123	Lynda Brumbaugh	24	F	02/05/1962	15	Seattle	3862 Mutton Town Road	1	NULL
1	Mr Meeseeks	33	M	11/10/1998	1	Seattle	3862 Mutton Town Road	1	05/01/2007
2	Rick Sanchez	32	M	19/08/1977	1	Illinois	3398 Braxton Street	2	05/01/2008
55	Jimmie Bassham	23	F	04/01/1969	2	California	93401 San Luis Obispo	123	NULL
11	Cheryl Brya	25	F	11/04/1988	123	California	93401 San Luis Obispo	123	NULL
101	Frank Whitney	15	M	27/12/1971	55	California	93401 San Luis Obispo	123	NULL
144	Glenda McCall	18	F	27/06/1995	55	California	93401 San Luis Obispo	123	NULL
105	Hugh Alt	15	M	05/04/1999	11	California	93401 San Luis Obispo	123	NULL
97	Roger Benson	18	M	27/04/1982	15	Illinois	3398 Braxton Street	2	NULL
66	Marcus Myers	15	M	16/02/1997	64	Seattle	3862 Mutton Town Road	1	NULL
5	Tracy Hilton	18	F	17/09/1978	2	Seattle	3862 Mutton Town Road	1	NULL
15	Margaret Fuller	23	F	22/07/1958	15	Seattle	3862 Mutton Town Road	1	NULL
35	Paul Staggs	19	M	14/08/1979	1	Seattle	3863 Mutton Town Road	2	NULL
98	Anthony Waston	19	M	06/11/1992	123	Illinois	3398 Braxton Street	2	NULL
89	James Goree	17	M	13/11/1989	55	Illinois	3398 Braxton Street	2	NULL
111	Viviana Jones	18	F	10/09/1975	11	California	93401 San Luis Obispo	123	NULL

Work log

Employee_id	Name	Customer_id	Customer Name	Address	Phone number	Handler branch	Date	Project Duration (hours)
124	Rick Astley	4	Walter Brown	803 Arroyo Lane	(95) 918-8942	Illinois	09/01/2010	12
137, 124	Rick Astley, Morty Smith	5	Martin K Pry	23 Harry Place	(86) 843-6110	Seattle	02/04/2010	12
137	Morty Smith	4	Walter Brown	803 Arroyo Lane	(87) 411-3197	Seattle	20/02/2010	56
131, 64	Dorothy Golden, Cathrine Gonzalez	9	Suzanne Davis	3366 Chardonnay Drive	(73) 847-3446	Illinois	16/04/2010	45
144	Glenda McCall	1	Estele Petruk	3 Raven Street	(37) 161-8336	California	16/04/2010	40
101,11	Frank Whitney, Cheryl Brya	2	Bidget Dobrovsky	174 Ramsey Circle	(55) 183-9994	California	07/05/2011	17
65	Russell Bueno	3	Corette Lashmore	69 Menomonie Crossing	(63) 615-3412	Seattle	09/10/2014	56
77, 137,1	John Lewis, Morty Smith, Mr Meeseeks	10	Alecia Banisch	56573 Hintze Lane	(29) 883-2203	Illinois	11/09/2015	32
66	Marcus Myers	7	Tanitansy Utteridge	6 Talmadge Hill	(25) 010-1897	Seattle	15/10/2016	50
89,98	Anthony Waston, James Goree	6	Thebault Balasin	6715 Shasta Lane	(54) 467-6802	Illinois	10/03/2017	41

Note: Project Duration column means, all employees involved in that project worked that many hours *individually*.

1. Explain how Insert, Delete and Update Anomaly can occur in this database.

- Insert Anomaly: We need information to distinguish tuples in a unique way. For example, in the employee table, for a certain branch, the branch manager id and the address should be known as well.
- Update Anomaly: If a branch name changes, then all the branch addresses must be updated.
- Delete Anomaly: If all tuples of a branch are deleted then the manager_id data is also lost.

2. Find the minimal cover for functional dependencies in both tables

- Minimal cover for employee table:

- Here is the table heading with relations:

Employee id	Name	Hourly Salary	Sex	Date of birth	Supervisor id	Branch name	Branch Address	Branch manager id	Managing since

- Let's assume we have the following abbreviations for each column name:

- Employee_id : E
 - Name: N
 - Hourly salary: H
 - Sex: G
 - Date of birth: D
 - Supervisor ID: S
 - Branch name: C
 - Branch address: I
 - Branch Manager ID: B
 - Managing Since: F

- Minimal cover **G** = {E → N, E → H, E → G, E → D, E → S, E → C, E → I, E → B, E → F, C → I, C → B, C → F, B → C, I → B}

- **NB: Minimal cover is not necessarily unique. So, for example, one can have I → C instead of I → B since through Transitivity we get I → B (I → C, C → B so I → B)**

- Minimal cover for the work_log table:

- Here is the table heading with relations:

Employee id	Name	Customer id	Customer Name	Address	Phone number	Handler branch	Project (hours)	Duration	Date

- Let's assume we have the following abbreviations for each column name:

- Employee_id : E
 - Name: N
 - Customer ID: K
 - Customer Name: L
 - Address: M
 - Phone number: Y
 - Handler Branch: H
 - Project Duration: P
 - Date: D

- Minimal cover **G** = {E → N, K → L, K → M, KD → Y, KD → H, EKD → P}

3. Normalize both tables to BCNF form if it was not possible, convert to 3NF.

- Employee table is not in BCNF nor is it in 3NF because a non-key attribute defines another non-key attribute i.e. branch_name → branch_address. This means that we need to split the tables (see page 42 of the lecture notes). The results are seen below:

Employees:

Employee id	Name	Hourly Salary	Sex	Date of birth	Supervisor id	Branch name
-------------	------	---------------	-----	---------------	---------------	-------------

Branches:

Branch name	Branch Address	Branch manager id	Managing since
-------------	----------------	-------------------	----------------

- Work_log table is not even in 1NF so it is not in the rest. So, we can first convert it to 1NF and then look how things are. Then, we need to split this table further.

Customers:

Customer id	Customer Name	Address
-------------	---------------	---------

Projects:

Employee id	Customer id	Phone number	Handler branch	Project Duration (hours)	Date
-------------	-------------	--------------	----------------	--------------------------	------

o **NB: The “employee name” can be recovered from the employee table so it was omitted.**

4. Using relational algebra, find out the name and id of all the customers which have been aided by more than one employee.

$\rho(P1 \text{ (Employee id} \rightarrow \text{Eid1)}), \pi_{\text{Employee id}, \text{Customer id}}(\text{Projects})$

$\rho(P2 \text{ (Employee id} \rightarrow \text{Eid2)}), \pi_{\text{Employee id}, \text{Customer id}}(\text{Projects})$

$\rho(P3, P1 \bowtie_{\text{Customer id}} P2)$

$\rho(P4, \pi_{\text{Customer id}}(\sigma_{\text{Eid1} \neq \text{Eid2}}(P3)))$

$\rho(P5, \pi_{\text{Customer id}, \text{Customer Name}} P4 \bowtie_{\text{Customer id}} \text{Customers})$

5. Find the names and ids of all the employees who are supervised by Cathrine Gonzalez.

$\rho(S1, \sigma_{\text{Name}=\text{Cathrine Gonzalez}}(\text{Employees}))$

$\rho(S2(\text{Employee id} \rightarrow \text{id}), \pi_{\text{Employee id}}(S1))$

$\rho(S3, \pi_{\text{Employee id}, \text{Name}}(\sigma_{\text{Supervisor id}=\text{id}}(S2)))$

Run SQL query to find out the answer for the following, one query for each question:

6. Create a working SQL database with normalized tables.

First, we create the database.

CREATE DATABASE DB;

```
CREATE TABLE `employee` (
  `employee_id` INT NOT NULL,
  `name` VARCHAR(50) NOT NULL,
  `hourly_salary` INT NOT NULL,
  `sex` VARCHAR(20) NOT NULL,
  `date_of_birth` DATE NOT NULL,
  `supervisor_id` INT NOT NULL,
  `branch_name` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`employee_id`)
);
```

```

CREATE TABLE `branches` (
  `branch_name` VARCHAR(50) NOT NULL,
  `branch_manager_id` INT NOT NULL,
  `branch_Address` VARCHAR(400) NOT NULL,
  `managing_since` DATE NOT NULL,
  PRIMARY KEY (`branch_name`)
);

CREATE TABLE `customers` (
  `customer_id` INT NOT NULL,
  `customer_name` VARCHAR(100) NOT NULL,
  `address` VARCHAR(400) NOT NULL,
  PRIMARY KEY (`customer_id`)
);

CREATE TABLE `projects` (
  `employee_id` INT NOT NULL,
  `customer_id` INT NOT NULL,
  `phone_number` VARCHAR(20) NOT NULL,
  `handler_branch` VARCHAR(50) NOT NULL,
  `project_duration` INT NOT NULL,
  `date` DATE NOT NULL,
  PRIMARY KEY (`employee_id`, `customer_id`, `date`)
);

```

7. Insert the same data in the newly created Database.

```

INSERT INTO `employee`(
`employee_id`, `name`, `hourly_salary`, `sex`, `date_of_birth`, `supervisor_id`, `branch_name`)
VALUES
(124,"Rick Astley",18,'M',"1966-06-02",64,'Illinois'),
(131,"Dorothy Golden",18,'F',"1969-08-30",64,'Illinois'),
(137,"Morty Smith", 18, 'M', "1976-05-24", 11, 'Seattle'),
(64,"Cathrine Gonzalez",20,'F',"1959-11-24",11,'Illinois'),
(76,"Satoshi Nakamoto",18,'M',"1975-08-25",11,'Illinois'),
(77, "John Lewis", 17, 'M', "1998-03-05", 64, 'Seattle'),
(65,"Russell Bueno",17,'M',"1966-04-25",64,'Seattle'),
(123,"Kristie Clark",30,'F',"1975-05-09",2,'California'),
(122,"Lynda Brumbaugh",24,'F',"1962-02-05",15,'Seattle'),
(1, "Mr Meeseeks", 33, 'M', "1998-10-11", 1, 'Seattle'),
(2, "Rick Sanchez", 32, 'M', "1977-08-19", 1, 'Illinois'),
(55,"Jimmie Bassham",23,'F',"1969-01-04",2,'California'),
(11, "Cheryl Brya", 25, 'F', "1988-04-11", 123, 'California'),
(101,"Frank Whitney",15,'M',"1971-12-27",55,'California'),
(144,"Glenda McCall",18,'F',"1995-06-27",55,'California'),
(105, "Hugh Alt", 15, 'M', "1999-04-05", 11, 'California'),
(97, "Roger Benson", 18, 'M', "1982-04-27", 15, 'Illinois'),
(66, "Marcus Myers", 15, 'M', "1997-02-16", 64, 'Seattle'),
(5, "Tracy Hilton", 18, 'F', "1978-09-17", 2, 'Seattle'),
(15,"Margaret Fuller",23,'F',"1958-07-22",15,'Seattle'),
(35, "Paul Staggs", 19, 'M', "1979-08-14", 1, 'Seattle'),
(98,"Anthony Waston",19,'M',"1992-11-06",123,'Illinois'),
(89, "James Goree", 17, 'M', "1989-11-13", 55, 'Illinois'),
(111,"Viviana Jones",18,'F',"1975-09-10",11,'California');

```

```

INSERT INTO `customers`(`customer_id`,`customer_name`,`address`)
VALUES
(4,"Walter Brown","803 Arroyo Lane"),
(5,"Martin K Pry","23 Harry Place"),
(9,"Suzanne Davis ","3366 Chardonnay Drive"),
(1,"Estele Petruk","3 Raven Street"),
(2,"Bidget Dobrovsky","174 Ramsey Circle"),
(3,"Corette Lashmore","69 Menomonie Crossing"),
(10,"Alecia Banisch ","56573 Hintze Lane"),
(7,"Tanitansy Utteridge","6 Talmadge Hill ");

```

```

(6,"Thebault Balasin","6715 Shasta Lane");

INSERT INTO `projects`
(`employee_id`,`customer_id`,`phone_number`,`handler_branch`,`date`,`project_duration`)
VALUES
(124,4,'(95) 918-8942','Illinois','2010-01-09',12),
(124,5,'(86) 843-6110','Seattle','2010-04-02',12),
(137,5,'(86) 843-6110','Seattle','2010-04-02',12),
(137,4,'(87) 411-3197','Seattle','2010-02-20',56),
(131,9,'(73) 847-3446','Illinois','2010-04-16',45),
(144,1,'(37) 161-8336','California','2010-04-16',40),
(101,2,'(55) 183-9994','California','2011-05-07',17),
(11,2,'(55) 183-9994','California','2011-05-07',17),
(65,3,'(63) 615-3412','Seattle','2014-10-14',56),
(77,10,'(29) 883-2203','Illinois','2015-09-11',32),
(137,10,'(29) 883-2203','Illinois','2015-09-11',32),
(1,10,'(29) 883-2203','Illinois','2015-09-11',32),
(66,7,'(25) 010-1897','Seattle','2016-10-15',50),
(89,6,'(54) 467-6802','Illinois','2017-03-10',41),
(98,6,'(54) 467-6802','Illinois','2017-03-10',41);

```

- 8. Insert a new entry for customer 4, handled by Illinois branch, Service provided by Employee 64 and 131, project duration 45.**

```

INSERT INTO projects
VALUES
('64','4','(95) 918-8942','Illinois','45','2018-03-10'),
('131','4','(95) 918-8942','Illinois','45','2018-03-10');

```

- 9. Make a table of past managers with employee id, management start date, branch name.**

```

CREATE TABLE `past_managers` (
`employee_id` INT NOT NULL,
`managing_since` DATE NOT NULL,
`branch_name` VARCHAR(20) NULL,
PRIMARY KEY (`employee_id`)
);

```

- 10. Add a trigger so when a manager is updated, the trigger should add update the table made in question 9 with the current manager.**

```

DROP TRIGGER IF EXISTS past_managers_trigger;
DELIMITER $$ 
CREATE TRIGGER past_managers_trigger AFTER UPDATE ON `branches` 
FOR EACH ROW BEGIN
IF (NEW.branch_manager_id <> OLD.branch_manager_id) THEN
INSERT INTO `past_managers` 
VALUES
(
old.branch_manager_id,
old.managing_since,
old.branch_name
);
END IF;
END$$
DELIMITER ;

```

- 11. Make employee 64 to be the manager of Illinois branch since 3/14/2015, Make Employee 76 the supervisor of employee 124 and 131.**

```

UPDATE `branches` 
SET branch_manager_id='64', managing_since = '2015-3-14'
WHERE branch_name = 'Illinois';

```

12. Remove employee 124 from the project that was done for customer 5.

```
DELETE FROM `projects`  
WHERE employee_id='124' AND customer_id = '5'
```

13. Name of the employee who made Illinois branch the most amount of money.

```
SELECT E.employee_id, E.name, SUM(P.project_duration) AS work_log  
FROM employee E, projects P  
WHERE E.employee_id=P.employee_id  
AND P.handler_branch = 'Illinois'  
GROUP BY employee_id  
ORDER BY work_log DESC  
LIMIT 1
```

14. Phone Number of the longest service taking customer.

```
SELECT DISTINCT(P.phone_number)  
FROM projects P,  
(SELECT customer_id, SUM(project_duration) as service_hours  
FROM projects  
GROUP BY customer_id  
ORDER BY service_hours DESC  
LIMIT 1  
) AS loyalists  
WHERE P.customer_id = loyalists.customer_id
```

15. Supervisor's name of the employees who work in a branch managed by employee 64.

```
SELECT em.name  
FROM employee em  
WHERE em.employee_id IN (SELECT em1.supervisor_id  
FROM employee em1, branches B  
WHERE em1.branch_name = B.branch_name AND B.branch_manager_id= 64 );
```