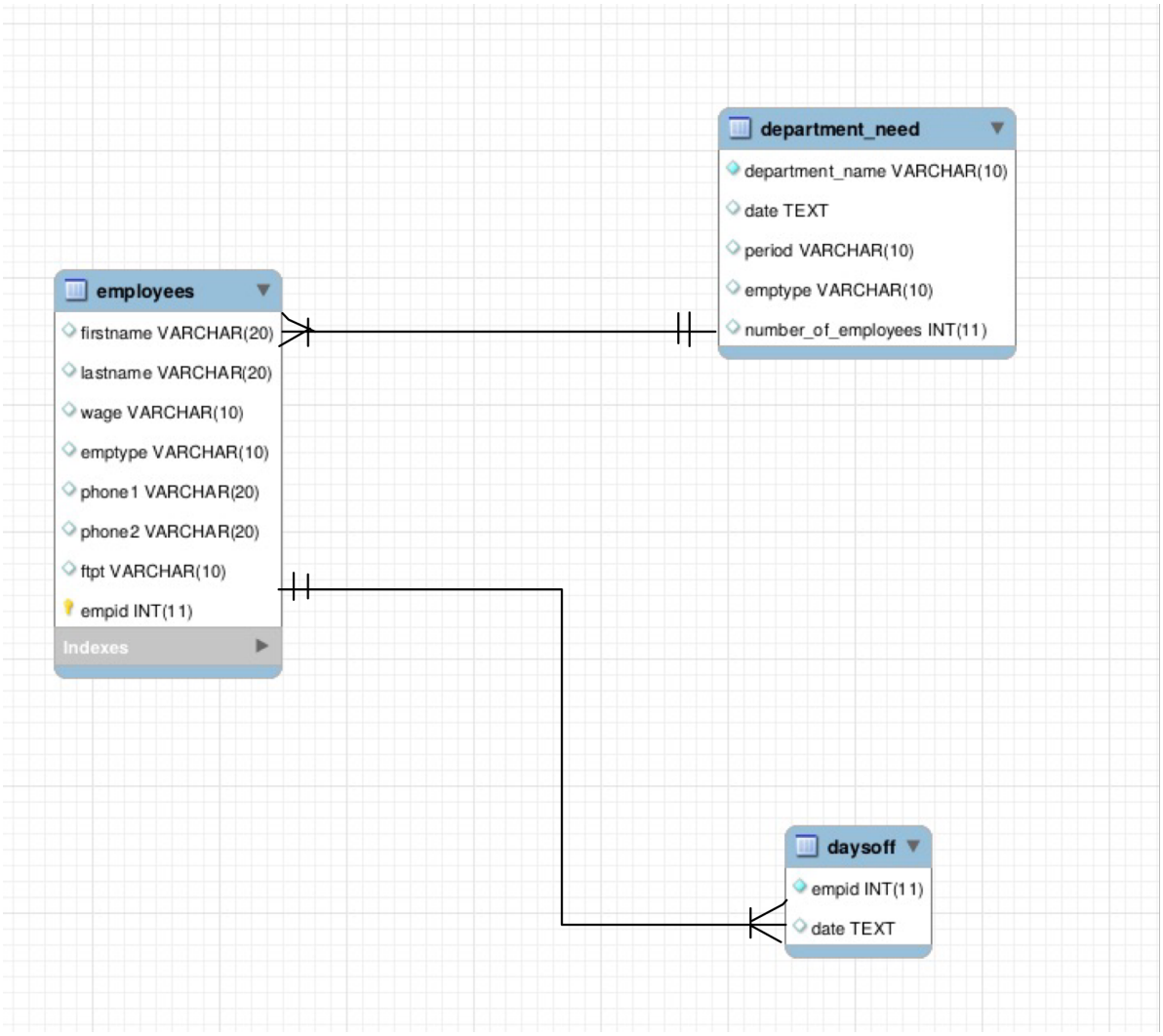
Assignment 11

Jiarong Ye

November 14, 2018

Q1

A ER diagram showing your data table design in 3NF.



Q2

Load this data into your 3NF tables structure.

In [4]:
```
CREATE TABLE department_need
(
  department_name  VARCHAR(10) NOT NULL,
  date             TEXT        NULL,
  period           VARCHAR(10) NULL,
  emptype          VARCHAR(10) NULL,
  number_of_employees  INT     NULL
);

CREATE TABLE employees
(
  firstname VARCHAR(20) NULL,
  lastname  VARCHAR(20) NULL,
  wage      VARCHAR(10) NULL,
  emptype   VARCHAR(10) NULL,
  phone1    VARCHAR(20) NULL,
  phone2    VARCHAR(20) NULL,
  ftpt      VARCHAR(10)  NULL
);


CREATE TABLE daysoff
(
  firstname VARCHAR(20) NOT NULL,
  lastname  VARCHAR(20) NOT NULL,
  date  TEXT NULL
);



LOAD DATA INFILE 'project_asgn_11_data/needs.csv' INTO TABLE department_need
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';


LOAD DATA INFILE 'project_asgn_11_data/employee2.csv' INTO TABLE employees
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';


LOAD DATA INFILE 'project_asgn_11_data/daysoffrequests.csv' INTO TABLE daysoff
FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';


UPDATE daysoff SET firstname = REPLACE(firstname, ' ', '');
UPDATE employees SET firstname = REPLACE(firstname, '"', '');
UPDATE employees SET lastname = REPLACE(lastname, '"', '');
```

```
UPDATE employees SET wage = REPLACE(wage, '"', '');
UPDATE employees SET wage = REPLACE(wage, '$', '');
UPDATE employees SET emptype = REPLACE(emptype, '"', '');
UPDATE employees SET phone1 = REPLACE(phone1, '"', '');
UPDATE employees SET phone2 = REPLACE(phone2, '"', '');
UPDATE employees SET ftpt = REPLACE(ftpt, '"', '');


ALTER TABLE employees ADD empid INT NOT NULL AUTO_INCREMENT PRIMARY KEY;

CREATE TABLE daysoff_tmp
(
  empid INT  NOT NULL,
  date  TEXT NULL
) SELECT e.empid, d.date
  FROM employees as e, daysoff as d
  WHERE (d.firstname, d.lastname) = (e.firstname, e.lastname);

DROP TABLE daysoff;
RENAME TABLE daysoff_tmp TO daysoff;

SHOW TABLES;
DESCRIBE department_need;
DESCRIBE employees;
DESCRIBE daysoff;
```

99 rows affected. (Query took 0.0469 seconds.)

```
UPDATE employees SET lastname = REPLACE(lastname, '"', '')
```

99 rows affected. (Query took 0.0471 seconds.)

```
UPDATE employees SET wage = REPLACE(wage, '"', '')
```

99 rows affected. (Query took 0.0689 seconds.)

```
UPDATE employees SET wage = REPLACE(wage, '$', '')
```

99 rows affected. (Query took 0.0508 seconds.)

```
UPDATE employees SET emptype = REPLACE(emptype, '"', '')
```

99 rows affected. (Query took 0.0690 seconds.)

```
UPDATE employees SET phone1 = REPLACE(phone1, '"', '')
```

99 rows affected. (Query took 0.2803 seconds.)

```
UPDATE employees SET phone2 = REPLACE(phone2, '"', '')
```

99 rows affected. (Query took 0.0879 seconds.)

```
UPDATE employees SET ftpt = REPLACE(ftpt, '"', '')
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.5754 seconds.)

```
ALTER TABLE employees ADD empid INT NOT NULL AUTO_INCREMENT PRIMARY KEY
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.4098 seconds.)

```
CREATE TABLE daysoff_tmp ( empid INT NOT NULL, date TEXT NULL ) SELECT e.empid, d.date FROM employees as e, daysoff as d WHERE (d.firstname, d.lastname) = (e.firstname, e.lastname)
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.1114 seconds.)

```
DROP TABLE daysoff
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.1610 seconds.)

```
RENAME TABLE daysoff_tmp TO daysoff
```

Your SQL query has been executed successfully.

```
SHOW TABLES
```

+ Options

Tables_in_assignment11
daysoff
department_need
employees

Your SQL query has been executed successfully.

```
DESCRIBE department_need
```

+ Options

Field	Type	Null	Key	Default	Extra
department_name	varchar(10)	NO		*NULL*	
date	text	YES		*NULL*	
period	varchar(10)	YES		*NULL*	
emptype	varchar(10)	YES		*NULL*	
number_of_employees	int(11)	YES		*NULL*	

Your SQL query has been executed successfully.

```
DESCRIBE employees
```

+ Options

Field	Type	Null	Key	Default	Extra
firstname	varchar(20)	YES		*NULL*	
lastname	varchar(20)	YES		*NULL*	
wage	varchar(10)	YES		*NULL*	
emptype	varchar(10)	YES		*NULL*	
phone1	varchar(20)	YES		*NULL*	
phone2	varchar(20)	YES		*NULL*	
ftpt	varchar(10)	YES		*NULL*	
empid	int(11)	NO	PRI	*NULL*	auto_increment

Your SQL query has been executed successfully.

```
DESCRIBE daysoff
```

+ Options

Field	Type	Null	Key	Default	Extra
empid	int(11)	NO		*NULL*	
date	text	YES		*NULL*	

Q3

Calculate the total needs of all departments, all days, all shifts by employee type. For example: RNs: 3214 hours, LPNs: 2735 hours, etc.

```
In [ ]: SELECT emptype, cast(SUM(shifthrs) AS INT) AS total_working_hrs
        FROM
          (SELECT emptype,
            8*number_of_employees AS shifthrs
           FROM department_need) as tmp
        GROUP BY emptype;
```

Showing rows 0 - 5 (6 total, Query took 0.0024 seconds.)

```
SELECT emptype, cast(SUM(shifthrs) AS INT) AS total_working_hrs FROM (SELECT emptype, 8*number_of_employees AS shifthrs FROM department_need) as tmp GROUP BY emptype
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

emptype	total_working_hrs
LPN	2296
NA	1480
PHLEB	768
RN	4816
ULTRA	504
XRAY	520

Q4

Calculate the total available hours per employee type. For example: RNs: 3000 hours, LPNs: 2800 hours. Note that a part-time person is limited to 24 hours per week. Also note that requested time off is not figured into this calculation.

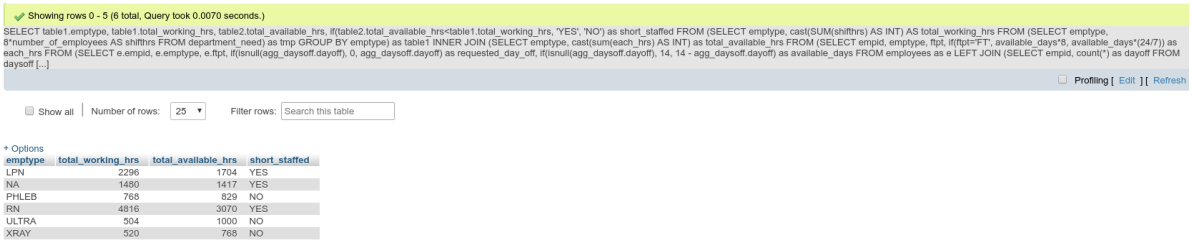
```
In [ ]: SELECT emptype, cast(sum(each_hrs) AS INT) as total_available_hrs
        FROM (SELECT empid, emptype, ftpt, if(ftpt='FT', available_days*8, available_days*(24/7))
        as each_hrs
               FROM (SELECT e.empid, e.emptype, e.ftpt,
                     if(isnull(agg_daysoff.dayoff), 0, agg_daysoff.dayoff) as requested_day_off,
                     if(isnull(agg_daysoff.dayoff), 14, 14 - agg_daysoff.dayoff) as available_days
                     FROM employees as e
                     LEFT JOIN (SELECT empid, count(*) as dayoff
                                FROM daysoff
                                GROUP BY empid) as agg_daysoff
                     ON e.empid=agg_daysoff.empid) as tmp1) as tmp2
        GROUP BY emptype;
```

Q5

List (via PHP/SQL code) which employee types are short-staffed (you don't have enough possible hours to fill the needs for that employee type).

```
In [ ]: SELECT table1.emptype,
               table1.total_working_hrs,
               table2.total_available_hrs,
               if(table2.total_available_hrs<table1.total_working_hrs, 'YES', 'NO') as short_staffed
        FROM (SELECT emptype, cast(SUM(shifthrs) AS INT)
        AS total_working_hrs
              FROM
                (SELECT emptype,
                   8*number_of_employees AS shifthrs
                 FROM department_need) as tmp
              GROUP BY emptype) as table1
        INNER JOIN
          (SELECT emptype, cast(sum(each_hrs) AS INT)
          as total_available_hrs
           FROM (SELECT empid, emptype, ftpt, if(ftpt='FT', available_days*8, available_days*(24/7))
                        as each_hrs
                 FROM (SELECT e.empid, e.emptype, e.ftpt,
                         if(isnull(agg_daysoff.dayoff), 0, agg_daysoff.dayoff) as requested_day_off,
                         if(isnull(agg_daysoff.dayoff), 14, 14 - agg_daysoff.dayoff) as available_days
```

```
                FROM employees as e
                  LEFT JOIN (SELECT empid, count(*) as dayoff
                             FROM daysoff
                             GROUP BY empid) as agg_daysoff
                    ON e.empid=agg_daysoff.empid) as tmp1) as tmp2
   GROUP BY emptype) as table2
ON table1.emptype=table2.emptype;
```



Showing rows 0 - 5 (6 total, Query took 0.0070 seconds.)

```
SELECT table1.emptype, table1.total_working_hrs, table2.total_available_hrs, if(table2.total_available_hrs<table1.total_working_hrs, 'YES', 'NO') as short_staffed FROM (SELECT emptype, cast(SUM(shifthrs) AS INT) AS total_working_hrs FROM (SELECT emptype,
8*number_of_employees AS shifthrs FROM department_need) as tmp GROUP BY emptype) as table1 INNER JOIN (SELECT emptype, cast(sum(each_hrs) AS INT) as total_available_hrs FROM (SELECT empid, emptype, ftpt, if(ftpt='FT', available_days*8, available_days*(24/7)) as
each_hrs FROM (SELECT e.empid, e.emptype, e.ftpt, if(isnull(agg_daysoff.dayoff), 0, agg_daysoff.dayoff) as requested_day_off, if(isnull(agg_daysoff.dayoff), 14, 14 - agg_daysoff.dayoff) as available_days FROM employees as e LEFT JOIN (SELECT empid, count(*) as dayoff FROM
daysoff [...]
```

Profiling [Edit] [Refresh

Show all | Number of rows: 25 Filter rows: Search this table

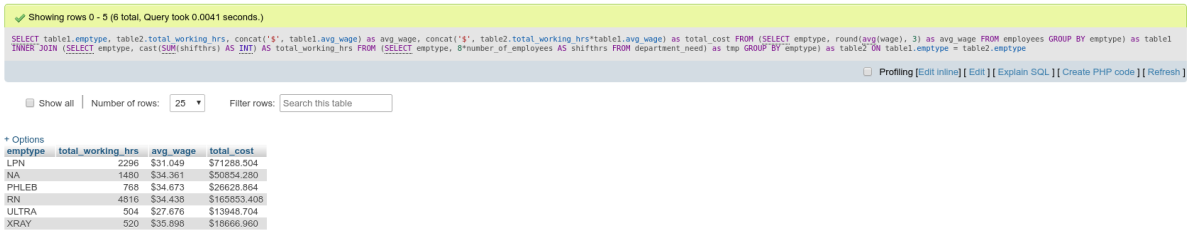
+ Options

emptype	total_working_hrs	total_available_hrs	short_staffed
LPN	2296	1704	YES
NA	1480	1417	YES
PHLEB	768	829	NO
RN	4816	3070	YES
ULTRA	504	1000	NO
XRAY	520	768	NO

Q6

Calculate the average cost per hour for each employee type, then use that number to estimate the total cost for each employee type for the entire schedule.

```
In [ ]: SELECT table1.emptype,
               table2.total_working_hrs,
               concat('$', table1.avg_wage) as avg_wage,
               concat('$', table2.total_working_hrs*table1.avg_wage) as total_cost
        FROM (SELECT emptype, round(avg(wage), 3) as avg_wage
              FROM employees
              GROUP BY emptype) as table1
        INNER JOIN
          (SELECT emptype, cast(SUM(shifthrs) AS INT) AS total_working_hrs
                    FROM
                      (SELECT emptype,
                         8*number_of_employees AS shifthrs
                       FROM department_need) as tmp
                    GROUP BY emptype) as table2
        ON table1.emptype = table2.emptype;
```



Showing rows 0 - 5 (6 total, Query took 0.0041 seconds.)

```
SELECT table1.emptype, table2.total_working_hrs, concat('$', table1.avg_wage) as avg_wage, concat('$', table2.total_working_hrs*table1.avg_wage) as total_cost FROM (SELECT emptype, round(avg(wage), 3) as avg_wage FROM employees GROUP BY emptype) as table1
INNER JOIN (SELECT emptype, cast(SUM(shifthrs) AS INT) AS total_working_hrs FROM (SELECT emptype, 8*number_of_employees AS shifthrs FROM department_need) as tmp GROUP BY emptype) as table2 ON table1.emptype = table2.emptype
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

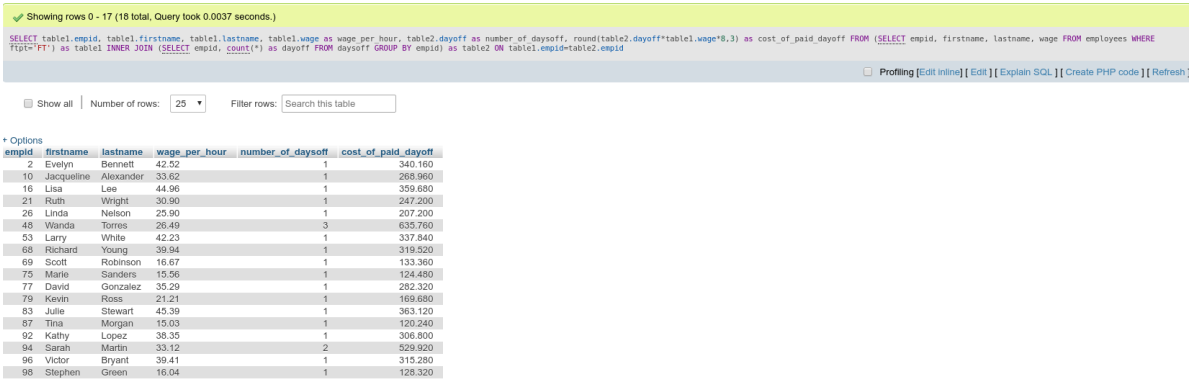
+ Options

emptype	total_working_hrs	avg_wage	total_cost
LPN	2296	$31.049	$71288.504
NA	1480	$34.361	$50854.280
PHLEB	768	$34.673	$26628.864
RN	4816	$34.438	$165853.408
ULTRA	504	$27.676	$13948.704
XRAY	520	$35.898	$18666.960

Q7

For all full time employees, calculate the total cost of giving them the day off. This assumes that they get paid time off, and that they will be paid for one, 8-hour shift.

```
In [ ]: SELECT table1.empid,
               table1.wage as wage_per_hour,
               table2.dayoff as number_of_daysoff,
               round(table2.dayoff*table1.wage*8,3) as cost_of_paid_dayoff
        FROM
          (SELECT empid, wage
           FROM employees
           WHERE ftpt='FT') as table1
        INNER JOIN
          (SELECT empid, count(*) as dayoff
           FROM daysoff
           GROUP BY empid) as table2
        ON table1.empid=table2.empid;
```



Showing rows 0 - 17 (18 total, Query took 0.0037 seconds.)

```
SELECT table1.empid, table1.firstname, table1.lastname, table1.wage as wage_per_hour, table2.dayoff as number_of_daysoff, round(table2.dayoff*table1.wage*8,3) as cost_of_paid_dayoff FROM (SELECT empid, firstname, lastname, wage FROM employees WHERE
ftpt='FT') as table1 INNER JOIN (SELECT empid, count(*) as dayoff FROM daysoff GROUP BY empid) as table2 ON table1.empid=table2.empid
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

empid	firstname	lastname	wage_per_hour	number_of_daysoff	cost_of_paid_dayoff
2	Evelyn	Bennett	42.52	1	340.160
10	Jacqueline	Alexander	33.62	1	268.960
16	Lisa	Lee	44.96	1	359.680
21	Ruth	Wright	30.90	1	247.200
26	Linda	Nelson	25.90	1	207.200
48	Wanda	Torres	26.49	3	635.760
53	Larry	White	42.23	1	337.840
68	Richard	Young	39.94	1	319.520
69	Scott	Robinson	16.67	1	133.360
75	Marie	Sanders	15.56	1	124.480
77	David	Gonzalez	35.29	1	282.320
79	Kevin	Ross	21.21	1	169.680
83	Julie	Stewart	45.39	1	363.120
87	Tina	Morgan	15.03	1	120.240
92	Kathy	Lopez	38.35	1	306.800
94	Sarah	Martin	33.12	2	529.920
96	Victor	Bryant	39.41	1	315.280
98	Stephen	Green	16.04	1	128.320