**# Current DB fulfill 3NF (Normal Form) and support to medium level Organization.**

**# We can add some more table/column details for enterprise level Organization. We can convert this DB format to 4th, 5th NF for Big Data storage and data mining etc.**

**# For small level organization we can save whole data in single table and as a result redundancy will increase.**

**EMPLOYEE DATABASE:**

**Query:**

CREATE DATABASE employee

**EMPLOYEE TABLE:**

**Query:**

CREATE TABLE employee.employee (

id int(11) NOT NULL AUTO\_INCREMENT,

name varchar(250) NOT NULL,

dob date NOT NULL,

SSN varchar(9) NOT NULL COMMENT 'SSN (Social Security Number)',

status binary(1) NOT NULL DEFAULT 0 COMMENT '1=Yes , 0 = No',

PRIMARY KEY (id),

UNIQUE INDEX SSN (SSN)

)

ENGINE = INNODB

AUTO\_INCREMENT = 1

CHARACTER SET utf8

COLLATE utf8\_unicode\_ci

COMMENT = 'Employee General Details';

**PORTFOLIO TABLE**

**Query:**

CREATE TABLE employee.portfolio (

id int(11) NOT NULL,

employee\_id int(11) DEFAULT NULL,

lang\_id int(11) NOT NULL,

introduction varchar(500) NOT NULL,

education varchar(500) DEFAULT NULL,

work\_experience varchar(1000) DEFAULT NULL,

PRIMARY KEY (id),

CONSTRAINT FK\_portfolio\_employee\_id FOREIGN KEY (employee\_id)

REFERENCES employee.employee (id) ON DELETE RESTRICT ON UPDATE RESTRICT,

CONSTRAINT FK\_portfolio\_language\_id FOREIGN KEY (lang\_id)

REFERENCES employee.language (id) ON DELETE RESTRICT ON UPDATE RESTRICT

)

ENGINE = INNODB

CHARACTER SET utf8

COLLATE utf8\_unicode\_ci

COMMENT = 'Employee Portfolio';

**LANGUAGE TABLE**

**Query:**

CREATE TABLE employee.language (

id int(11) NOT NULL AUTO\_INCREMENT,

language varchar(50) NOT NULL,

PRIMARY KEY (id),

UNIQUE INDEX language (language)

)

ENGINE = INNODB

AUTO\_INCREMENT = 1

CHARACTER SET utf8

COLLATE utf8\_unicode\_ci;

**LOG TABLE**

**Query:**

CREATE TABLE employee.log (

id int(11) NOT NULL AUTO\_INCREMENT,

employee\_id int(11) NOT NULL,

action varchar(50) NOT NULL,

timestamp timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

CONSTRAINT FK\_log\_employee\_id FOREIGN KEY (employee\_id)

REFERENCES employee.employee (id) ON DELETE RESTRICT ON UPDATE RESTRICT

)

ENGINE = INNODB

AUTO\_INCREMENT = 1

CHARACTER SET utf8

COLLATE utf8\_unicode\_ci;

Task 2

<?php

/\*\*

\* @author Engr. Nayab Bukhari, Syed

\* @copyright 2017

\* outputs an <ul> list of all integers in the range of 1-100

\* in place of all numbers that are divisible by 3, outputs string “foo” instead

\* in place of all numbers that are divisible by 5, outputs string “bar” instead

\* for numbers that are divisible by both 3 and 5, outputs string “foobar” instead

\*

\*\*/

echo '<ul>';

foreach(range(1,100)as$n){

echo '<li>';

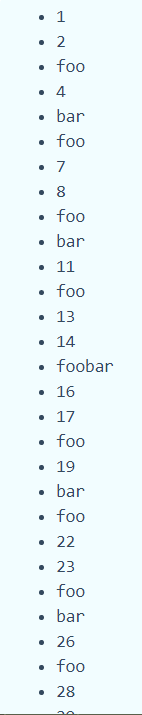
echo(''==($s=($n%3==0?"foo":"").($n%5==0?"bar":""))?$n:$s).'</li>';

}

echo '</ul>';

?>

**OUTPUT:**



Task 3:

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even){

background-color: #f2f2f2

}

th {

background-color: #4CAF50;

color: white;

}

</style>

<?php

// The start and end of nighttime hours

$SETTINGS\_nighttime\_start = '22:00';

$SETTINGS\_nighttime\_end = '07:00';

// Employees and their shifts

$EMPLOYEES = array(

'0' => array(

'name' => 'Bernice Lyons',

'shift\_start' => '15:15',

'shift\_end' => '23:45'

),

'1' => array(

'name' => 'Gregg Santos',

'shift\_start' => '10:00',

'shift\_end' => '22:00'

),

'2' => array(

'name' => 'Bennie Montgomery',

'shift\_start' => '22:30',

'shift\_end' => '08:00'

),

'3' => array(

'name' => 'Nelson Austin',

'shift\_start' => '20:00',

'shift\_end' => '10:00'

),

'4' => array(

'name' => 'Garrett Sims',

'shift\_start' => '09:00',

'shift\_end' => '17:00'

),

'5' => array(

'name' => 'Joanna Pratt',

'shift\_start' => '23:00',

'shift\_end' => '06:00'

),

'6' => array(

'name' => 'Bernice Lyons',

'shift\_start' => '15:15',

'shift\_end' => '23:40'

)

);

foreach($EMPLOYEES AS $emp){

echo '<table><tr><th>'.strtoupper('Name: '.$emp['name']).'</th></tr>

<tr><td><ul><li>Shift Start:'.$emp['shift\_start'].'</li>

<li>Shift End:'.$emp['shift\_end'].'</li>';

echo calc\_hour('am', $emp['shift\_start'], $emp['shift\_end']);

echo calc\_hour('pm', $emp['shift\_start'], $emp['shift\_end']);

echo '</ul></td></tr></table>';

}

/\*\*

\* @author Engr. Nayab Bukhari, Syed

\* @copyright 2017

\* @output:

\* how much of the shift is daytime hours

\* how much of the shift is nighttime hours

\* The results should be displayed with a 15 minute (0.25h) precision, both “x.25”

\* and “x:15” formats are allowed.

\* All times are in the 24 hour HH:MM format. Times are with 15 minute precision

\* (for example, 15:00, 16:15, 17:30 are allowed values, whereas 18:02 and 19:41 are not).

\*

\*\*/

function calc\_hour($shift, $shiftStart, $shiftEnd){

$shift\_start=explode(":", $shiftStart);

$shift\_end=explode(":", $shiftEnd);

if(($shift\_start[1]% 15 != 0) && ($shift\_start[1] != 00 )){

$result = '<li>Night Time Hours : Invalid shift start time </li>';

$shift = 'error';

}elseif(($shift\_end[1]% 15 != 0) && ($shift\_end[1] != 00 )){

$result = '<li>Night Time Hours : Invalid shift end time </li>';

$shift = 'error';

}

switch($shift){

case 'pm':

if(($shift\_start[0] > 12) && ($shift\_end[0] < 12)){

if($shift\_start[1] == 00){

$night\_hour = 24 - $shift\_start[0];

}else{

$night\_hour = 23 - $shift\_start[0];

}

$result = '<li>Night Time Hours:'.$night\_hour.':'. $shift\_start[1] \* 100 / 60 .'</li>';

break;

}

if(($shift\_start[0] > 12) && ($shift\_end[0] > 12)){

$night\_time = $shift\_start[0];

if($shift\_start[1] + $shift\_end[1] == 60 ){

$shift\_end[0]= $shift\_end[0]+1;

$shift\_end[1]= "00";

}

}else{

$night\_time = 12;

}

if($shift\_end[0] > 12){

$night\_hour = abs($shift\_end[0] - $night\_time);

$result = '<li>Night Time Hours:'.$night\_hour.':'.$shift\_end[1].'</li>';

}else{

$result = '<li>Night Time Hours: NILL</li>';

}

break;

case 'am':

if($shift\_end[0] - $shift\_start[0] < 0){

$shift\_start[0]= "00";

$shift\_start[1] ="00";

}

if($shift\_end[0] < 12){

$day\_time = $shift\_end[0];

if($shift\_start[1] + $shift\_end[1] \* 100 / 60 == 100 ){

$shift\_start[0]= $shift\_start[0]+1;

$shift\_start[1]= "00";

}

}else{

$day\_time = 12;

}

if($shift\_start[0] < 12){

$day\_hour = abs($shift\_start[0]-$day\_time);

$result = '<li>Day Time Hours:'.$day\_hour.':'.$shift\_start[1].'</li>';

}else{

$result = '<li>Day Time Hours: NILL</li>';

}

break;

default:

return $result;

}

return $result;

}

?>

**OUTPUT:**

