Anbang Xu(UCI)

1. Create a database scheme with either MySQL or Microsoft SQL Server (information regarding how the CSV data should be interpreted is provided below)

I use MySQL in linux environment,

a. create a database

```
mysql> create database CODAZEN;
```

b. use this database

```
mysql> use CODAZEN;
```

c. create a table named ‘Department’

```
mysql> create table Department
    -> (id INT NOT NULL,
    -> name VARCHAR(255),
    -> PRIMARY KEY ( id )
    -> );
```

d. create a table named ‘Employee’

```
mysql> create table Employee
    -> (
    -> type VARCHAR(255),
    -> id VARCHAR(255) NOT NULL,
    -> first_name VARCHAR(255),
    -> last_name VARCHAR(255),
    -> gender ENUM('F','M') NOT NULL,
    -> hire_date VARCHAR(255),
    -> termination_date VARCHAR(255),
    -> department_ids VARCHAR(255),
    -> primary key ( id )
    -> );
```

e. create a table named ‘Manager’

```
mysql> create table Manager
    -> (
    -> type VARCHAR(255),
    -> id VARCHAR(255) NOT NULL,
    -> first_name VARCHAR(255),
    -> last_name VARCHAR(255),
    -> gender ENUM('F','M') NOT NULL,
    -> hire_date VARCHAR(255),
    -> termination_date VARCHAR(255),
    -> department_ids VARCHAR(255),
    -> annual_bonus INT NOT NULL,
    -> manager_start_date VARCHAR(255),
    -> primary key ( id )
    -> );
```

f. create a table named ‘Salary’

```
mysql> create table Salary
    -> (
    -> id VARCHAR(255) NOT NULL,
    -> start_date VARCHAR(255),
    -> end_date VARCHAR(255),
    -> salary INT NOT NULL
    -> );
```

Snapshot:

a. database:

```
mysql> show tables;
+--------------------+
| Tables_in_CODAZEN |
+--------------------+
| Department         |
| Employee           |
| Manager            |
| Salary             |
+--------------------+
4 rows in set (0.00 sec)
```

b. Department table:

```
mysql> describe Department;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(255) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

c. Employ table:

```
mysql> describe Employee;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| type             | varchar(255)  | YES  |     | NULL    |       |
| id               | varchar(255)  | NO   | PRI | NULL    |       |
| first_name       | varchar(255)  | YES  |     | NULL    |       |
| last_name        | varchar(255)  | YES  |     | NULL    |       |
| gender           | enum('F','M') | NO   |     | NULL    |       |
| hire_date        | varchar(255)  | YES  |     | NULL    |       |
| termination_date | varchar(255)  | YES  |     | NULL    |       |
| department_ids   | varchar(255)  | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

d. Manager table:

```
mysql> describe Manager;
+--------------------+---------------+------+-----+---------+-------+
| Field              | Type          | Null | Key | Default | Extra |
+--------------------+---------------+------+-----+---------+-------+
| type               | varchar(255)  | YES  |     | NULL    |       |
| id                 | varchar(255)  | NO   | PRI | NULL    |       |
| first_name         | varchar(255)  | YES  |     | NULL    |       |
| last_name          | varchar(255)  | YES  |     | NULL    |       |
| gender             | enum('F','M') | NO   |     | NULL    |       |
| hire_date          | varchar(255)  | YES  |     | NULL    |       |
| termination_date   | varchar(255)  | YES  |     | NULL    |       |
| department_ids     | varchar(255)  | YES  |     | NULL    |       |
| annual_bonus       | int(11)       | NO   |     | NULL    |       |
| manager_start_date | varchar(255)  | YES  |     | NULL    |       |
+--------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

e. Salary table:

```
mysql> describe Salary;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| id         | varchar(255) | NO   |     | NULL    |       |
| start_date | varchar(255) | YES  |     | NULL    |       |
| end_date   | varchar(255) | YES  |     | NULL    |       |
| salary     | int(11)      | NO   |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

2. Write a parser in Java or C# CSV and generate insert statements to insert the data into the tables designed in step 1

See Java src file. Remember to import 'mysql-connector-java-5.1.6-bin.jar' which is under lib folder. You also need to provide "username" and "password" for your mysql server from java input arguments.

3. Write queries to get the following information:
You can also check the file 'queries.sql'.

a. Get the names and departments of all managers
mysql> select first_name, last_name, department_ids from Manager;

```
mysql> select first_name, last_name, department_ids from Manager;
+------------+-----------+----------------+
| first_name | last_name | department_ids |
+------------+-----------+----------------+
| ARIEL      | LEWIS     | 0              |
| JOSE       | COLEMAN   | 3              |
| BRENDA     | GILBERT   | 1              |
| JULIANA    | GRANT     | 2              |
+------------+-----------+----------------+
4 rows in set (0.00 sec)
```

c. Get the name of the department that does not have manager
mysql> select * from Department d INNER JOIN Manager m where find_in_set(d.id, m.department_ids) < 0;

```
mysql> select * from Department d INNER JOIN Manager m where find_in_set(d.id, m.department_ids) < 0;
Empty set (0.00 sec)
```

d. Get the salary, first name, last name, and gender of the employee(s) that is/are currently hired with the lowest salary
mysql> select s.salary, e.first_name, e.last_name, e.gender from Salary s INNER JOIN Employee e on s.id = e.id where salary = (select min(salary) from Salary);

```
mysql> select s.salary, e.first_name, e.last_name, e.gender from Salary s INNER JOIN Employee e on s.id = e.id where salary = (select min(salary) from Salary);
+--------+------------+-----------+--------+
| salary | first_name | last_name | gender |
+--------+------------+-----------+--------+
|  30000 | ANGELICA   | ALLEN     | F      |
|  30000 | LANDON     | MORENO    | M      |
|  30000 | JARED      | COLLINS   | M      |
|  30000 | JARED      | GRAHAM    | M      |
|  30000 | CAROLINE   | STONE     | F      |
|  30000 | LUCAS      | STEVENS   | M      |
|  30000 | MARTIN     | RUSSELL   | M      |
|  30000 | KYLA       | HOWELL    | F      |
|  30000 | STELLA     | POWELL    | F      |
|  30000 | CARLOS     | PERKINS   | M      |
|  30000 | AMANDA     | PALMER    | F      |
|  30000 | REBECCA    | ROBERTSON | F      |
|  30000 | CASSIDY    | OLIVER    | F      |
+--------+------------+-----------+--------+
13 rows in set (0.00 sec)
```

e. Get the name and salary of the highest paid employee in “ENGINEERING” that is not a manager
mysql> select e.first_name,e.last_name,s.salary from Employee e INNER JOIN Salary s on e.id = s.id where s.salary = (select max(salary) from Employee e INNER JOIN Salary s on e.id = s.id INNER JOIN Department d where find_in_set(d.id, e.department_ids) AND d.name = 'ENGINEERING');

```
mysql> select e.first_name,e.last_name,s.salary from Employee e INNER JOIN Salary s on e.id = s.id where s.salary = (select max(salary) from Employee e INNER JOIN Salar
y s on e.id = s.id INNER JOIN Department d where find_in_set(d.id, e.department_ids) AND d.name = 'ENGINEERING');
+------------+------------+--------+
| first_name | last_name  | salary |
+------------+------------+--------+
| AUBREY     | GRIFFIN    |  84000 |
| ZOEY       | CUNNINGHAM |  84000 |
| LEAH       | MURRAY     |  84000 |
+------------+------------+--------+
3 rows in set (0.00 sec)
```

b. Get all employees who are “MALE”
mysql> select * from Employee where gender = 'M';

```
mysql> select * from Employee where gender = 'M';
+----------+------------+------------+------------+--------+------------+------------------+----------------+
| type     | id         | first_name | last_name  | gender | hire_date  | termination_date | department_ids |
+----------+------------+------------+------------+--------+------------+------------------+----------------+
| EMPLOYEE | 094ZE8E7NU | JESSE      | BLACK      | M      | 12/10/2002 | 03/22/2009       | 2              |
| EMPLOYEE | 0OAKXSD064 | OMAR       | MEDINA     | M      | 11/17/2010 | null             | 2              |
| EMPLOYEE | 1C9E7KAF8F | LANDON     | AUSTIN     | M      | 09/03/2008 | null             | 1;0            |
| EMPLOYEE | 1P46KXGF5V | MARTIN     | RUSSELL    | M      | 07/29/2013 | null             | 1;3            |
| EMPLOYEE | 30478X99C3 | CARLOS     | PERKINS    | M      | 08/09/2007 | null             | 2;3            |
| EMPLOYEE | 32X4432K5X | ALEXANDER  | MONTGOMERY | M      | 06/19/2000 | 04/02/2001       | 2              |
| EMPLOYEE | 3GUCG3AOXQ | BRYSON     | GRIFFIN    | M      | 09/26/2004 | 11/19/2010       | 3              |
| EMPLOYEE | 4LR6PT1GL8 | LUCAS      | STEVENS    | M      | 08/28/2012 | null             | 2              |
| EMPLOYEE | 5OW6BM03Z7 | MIGUEL     | GIBSON     | M      | 11/18/2003 | null             | 2;1            |
| EMPLOYEE | 5T9GY9HLQ5 | BRIAN      | GILBERT    | M      | 10/14/2012 | null             | 0              |
| EMPLOYEE | 5W738CG157 | COREY      | HUDSON     | M      | 07/26/2009 | 05/06/2013       | 1              |
| EMPLOYEE | 6I23YCC838 | TRISTAN    | TURNER     | M      | 06/25/2000 | 05/09/2011       | 3;0            |
| EMPLOYEE | 7S54IMHDGM | SHAWN      | TAYLOR     | M      | 07/11/2008 | null             | 0;1            |
| EMPLOYEE | 7X7DD047X8 | JESUS      | GRAY       | M      | 11/15/2005 | null             | 2;1            |
| EMPLOYEE | 848115U60M | ZANE       | GREEN      | M      | 09/23/2013 | null             | 2              |
| EMPLOYEE | 8B3OQ4FN7I | RAYMOND    | PORTER     | M      | 08/26/2005 | 03/25/2009       | 2;3            |
| EMPLOYEE | 8OKZ32OSK4 | JAIDEN     | GEORGE     | M      | 05/03/2009 | null             | 2;0            |
| EMPLOYEE | 9OYS5QWOM4 | GIOVANNI   | CASTRO     | M      | 03/06/2011 | null             | 2              |
| EMPLOYEE | 94OOF4N7E2 | TRISTAN    | HUDSON     | M      | 06/06/2007 | 01/02/2009       | 2              |
| EMPLOYEE | 985YN8YO7M | DRAKE      | LOPEZ      | M      | 07/24/2013 | null             | 2              |
| EMPLOYEE | 9F68NSNE80 | SPENCER    | GUTIERREZ  | M      | 07/18/2011 | null             | 0              |
| EMPLOYEE | 9TS92K652E | PETER      | COOK       | M      | 08/20/2005 | 10/25/2013       | 1              |
| EMPLOYEE | ANB2O4858Z | JOSHUA     | PORTER     | M      | 05/22/2009 | null             | 3              |
| EMPLOYEE | BSO93OO952 | DAVID      | GEORGE     | M      | 04/08/2001 | null             | 2              |
| EMPLOYEE | EMF2OP182R | ETHAN      | MEDINA     | M      | 03/13/2001 | 10/03/2010       | 0              |
| EMPLOYEE | GT8K3GCOD2 | BLAKE      | MOORE      | M      | 05/15/2011 | null             | 0;2            |
| EMPLOYEE | H64GR3WFGC | JACE       | PERKINS    | M      | 09/19/2013 | null             | 3              |
| EMPLOYEE | IZ35M8YX27 | JARED      | GRAHAM     | M      | 12/21/2007 | null             | 2              |
| EMPLOYEE | K5823AP20D | LANDON     | MORENO     | M      | 06/24/2010 | null             | 0              |
| EMPLOYEE | L4I2HO9S68 | ISAAC      | ADAMS      | M      | 06/17/2008 | 12/03/2012       | 1              |
| EMPLOYEE | MO62JT3T59 | DREW       | WEAVER     | M      | 09/24/2005 | null             | 3              |
| EMPLOYEE | N51ZONUZRZ | JESSE      | GREENE     | M      | 11/30/2011 | null             | 2;0            |
| EMPLOYEE | N5QO5U8I47 | JACOB      | REYNOLDS   | M      | 01/11/2003 | null             | 3              |
| EMPLOYEE | NZ9AA8C3D9 | MANUEL     | WATSON     | M      | 03/28/2000 | 08/17/2012       | 2;0            |
| EMPLOYEE | PI9RXHMR81 | PEYTON     | ALEXANDER  | M      | 04/27/2004 | 01/07/2012       | 3              |
| EMPLOYEE | PO6628QZZ6 | JAKE       | STEVENS    | M      | 12/21/2006 | 03/29/2008       | 1              |
| EMPLOYEE | TC175B9C4K | JAKE       | AGUILAR    | M      | 07/07/2006 | 12/24/2013       | 3              |
| EMPLOYEE | U4UYKS61LE | JARED      | COLLINS    | M      | 02/14/2004 | 02/26/2009       | 3              |
| EMPLOYEE | U5X217YP53 | JOEL       | WEBB       | M      | 12/12/2010 | null             | 1              |
| EMPLOYEE | U9NP5QF746 | MANUEL     | CARR       | M      | 07/21/2011 | null             | 1;2            |
| EMPLOYEE | XO0ATE2UPO | TRAVIS     | ALLEN      | M      | 01/04/2003 | 12/25/2008       | 0              |
| EMPLOYEE | Y3BO69N161 | KEVIN      | DIXON      | M      | 06/15/2009 | 09/10/2012       | 3;2            |
| EMPLOYEE | YK9IH4SO7W | ERIC       | PERKINS    | M      | 11/30/2000 | 08/06/2008       | 1              |
| EMPLOYEE | Z8O4T8MPO8 | JONAH      | THOMPSON   | M      | 01/13/2009 | null             | 2              |
+----------+------------+------------+------------+--------+------------+------------------+----------------+
44 rows in set (0.00 sec)
```

4. Further optimization

Because time is limited, there are lots of optimizations I think.
a. Regarding the stored data types in MySQL,
Some columns recording Date such as hire_date can be considered as Date type in MySQL;
“Salary” column can be considered as a ‘bigger” type than int;
Employee.“Type” column can be considered as a “enum” type;

b. Regarding create database and tables,
It can be generated automatically by Java. The workflow I think is:
input csv -> automatically parse -> create database and tables -> insert records line by line
User doesn’t need to deal with MySQL any more.

c. Regarding some format checking,
Some restrict input format regulars should be checked when parsing the input and provide users some hints or exception.

d. Regarding queries,
We should make our own query wrapper functions for some popular query statements, rather than using Naive MySQL query statement each time.