Lab RDB - Relational databases

Student: Xuan Wang (xuawa284)
Student: Lepeng Zhang (lepzh903)

1. Show all employees and related information from the table jbemployee

```
mysql> SELECT * FROM jbemployee;
+------+--------------------+--------+---------+-----------+-----------+
| id   | name               | salary | manager | birthyear | startyear |
+------+--------------------+--------+---------+-----------+-----------+
|   10 | Ross, Stanley      |  15908 |     199 |      1927 |      1945 |
|   11 | Ross, Stuart       |  12067 |    NULL |      1931 |      1932 |
|   13 | Edwards, Peter     |   9000 |     199 |      1928 |      1958 |
|   26 | Thompson, Bob      |  13000 |     199 |      1930 |      1970 |
|   32 | Smythe, Carol      |   9050 |     199 |      1929 |      1967 |
|   33 | Hayes, Evelyn      |  10100 |     199 |      1931 |      1963 |
|   35 | Evans, Michael     |   5000 |      32 |      1952 |      1974 |
|   37 | Raveen, Lemont     |  11985 |      26 |      1950 |      1974 |
|   55 | James, Mary        |  12000 |     199 |      1920 |      1969 |
|   98 | Williams, Judy     |   9000 |     199 |      1935 |      1969 |
|  129 | Thomas, Tom        |  10000 |     199 |      1941 |      1962 |
|  157 | Jones, Tim         |  12000 |     199 |      1940 |      1960 |
|  199 | Bullock, J.D.      |  27000 |    NULL |      1920 |      1920 |
|  215 | Collins, Joanne    |   7000 |      10 |      1950 |      1971 |
|  430 | Brunet, Paul C.    |  17674 |     129 |      1938 |      1959 |
|  843 | Schmidt, Herman    |  11204 |      26 |      1936 |      1956 |
|  994 | Iwano, Masahiro    |  15641 |     129 |      1944 |      1970 |
| 1110 | Smith, Paul        |   6000 |      33 |      1952 |      1973 |
| 1330 | Onstad, Richard    |   8779 |      13 |      1952 |      1971 |
| 1523 | Zugnoni, Arthur A. |  19868 |     129 |      1928 |      1949 |
| 1639 | Choy, Wanda        |  11160 |      55 |      1947 |      1970 |
| 2398 | Wallace, Maggie J. |   7880 |      26 |      1940 |      1959 |
| 4901 | Bailey, Chas M.    |   8377 |      32 |      1956 |      1975 |
| 5119 | Bono, Sonny        |  13621 |      55 |      1939 |      1963 |
| 5219 | Schwarz, Jason B.  |  13374 |      33 |      1944 |      1959 |
+------+--------------------+--------+---------+-----------+-----------+
25 rows in set (0.00 sec)

mysql>
```

2. List the all the distinct department name in alphabetical order from the table jbdept

```
mysql> SELECT DISTINCT name FROM jbdept ORDER BY name ASC;
```

```
+------------------+
| name             |
+------------------+
| Bargain          |
| Book             |
| Candy            |
| Children's       |
| Furniture        |
| Giftwrap         |
| Jewelry          |
| Junior Miss      |
| Junior's         |
| Linens           |
| Major Appliances |
| Men's            |
| Sportswear       |
| Stationary       |
| Toys             |
| Women's          |
+------------------+
16 rows in set (0.00 sec)
```

3. List the parts name that are not in store from the table jbparts

```
mysql> SELECT name FROM jbparts WHERE qoh=0;
+-------------------+
| name              |
+-------------------+
| card reader       |
| card punch        |
| paper tape reader |
| paper tape punch  |
+-------------------+
4 rows in set (0.00 sec)

mysql>
```

4. List the employees name who have a salary between 9000 (included) and 10000 (included) from jbemployee

```
mysql> SELECT name FROM jbemployee WHERE salary >= 9000 AND salary <= 10000;
+----------------+
| name           |
+----------------+
| Edwards, Peter |
| Smythe, Carol  |
| Williams, Judy |
| Thomas, Tom       |
+----------------+
4 rows in set (0.01 sec)

mysql>
```

5. List the age of each employee when they started working from table jbemployee

```
mysql> SELECT name, startyear - birthyear AS start_age FROM jbemployee;
+--------------------+-----------+
| name                        | start_age |
+--------------------+-----------+
| Ross, Stanley         |            18 |
| Ross, Stuart           |              1 |
| Edwards, Peter        |            30 |
| Thompson, Bob          |            40 |
| Smythe, Carol          |            38 |
| Hayes, Evelyn          |            32 |
| Evans, Michael        |            22 |
| Raveen, Lemont        |            24 |
| James, Mary            |            49 |
| Williams, Judy        |            34 |
| Thomas, Tom             |            21 |
| Jones, Tim               |            20 |
| Bullock, J.D.         |              0 |
| Collins, Joanne       |            21 |
| Brunet, Paul C.       |            21 |
| Schmidt, Herman       |            20 |
| Iwano, Masahiro       |            26 |
| Smith, Paul             |            21 |
| Onstad, Richard       |            19 |
| Zugnoni, Arthur A. |            21 |
| Choy, Wanda             |            23 |
| Wallace, Maggie J. |            19 |
| Bailey, Chas M.       |            19 |
| Bono, Sonny             |            24 |
| Schwarz, Jason B.   |            15 |
+--------------------+-----------+
25 rows in set (0.00 sec)

mysql>
```

6. List employees name who have a last name ending with “son” using where cause

from the table jbemployee.

```
mysql> SELECT name FROM jbemployee WHERE name LIKE '%son';
Empty set (0.01 sec)

mysql>
```

7. List the items name that have been delivered by a supplier called Fisher-Price from the table jbitem using a subquery in the where-clause .

```
mysql> SELECT    name    FROM jbitem WHERE supplier IN (SELECT id FROM jbsupplier
WHERE name = 'Fisher-Price');
+-----------------+
| name            |
+-----------------+
| Maze            |
| The 'Feel' Book |
| Squeeze Ball    |
+-----------------+
3 rows in set (0.01 sec)

mysql>
```

8. List the items name that have been delivered by a supplier called Fisher-Price by using the table jbitem left join the table jbsupplier with a where-cause.

```
mysql> SELECT    A.name    FROM jbitem A LEFT JOIN jbsupplier B ON A.supplier = B.id
WHERE B.name = 'Fisher-Price';
+-----------------+
| name            |
+-----------------+
| Maze            |
| The 'Feel' Book |
| Squeeze Ball    |
+-----------------+
3 rows in set (0.00 sec)

mysql>
```

9. Show all cities that have suppliers located in them using a subquery in the where-clause from the table jbcity.

```
mysql> SELECT name FROM jbcity WHERE id IN (SELECT city FROM jbsupplier);
+----------------+
| name           |
+----------------+
| Amherst        |
| Boston           |
| New York         |
| White Plains     |
| Hickville        |
| Atlanta          |
| Madison          |
| Paxton           |
| Dallas           |
| Denver           |
| Salt Lake City |
| Los Angeles      |
| San Diego        |
| San Francisco    |
| Seattle          |
+----------------+
15 rows in set (0.01 sec)

mysql>
```

10. Show the name and color of the parts from the table jbparts and filter those are heavier than a card reader by using a subquery in the where-cause.

```
mysql> SELECT name, color FROM jbparts WHERE weight > (SELECT weight FROM jbparts WHERE name = "card reader");
+--------------+--------+
| name         | color  |
+--------------+--------+
| disk drive   | black  |
| tape drive   | black  |
| line printer | yellow |
| card punch   | gray   |
+--------------+--------+
4 rows in set (0.00 sec)

mysql>
```

11. Show the name and color of the parts from the table jbparts and filter those are heavier than a card reader by left join itself with a where-cause.

```
mysql> SELECT A.name, A.color FROM jbparts A LEFT JOIN jbparts B ON A.weight > B.weight WHERE B.name = "card reader";
+--------------+--------+
| name         | color  |
+--------------+--------+
| disk drive   | black  |
| tape drive   | black  |
| line printer | yellow |
| card punch   | gray   |
+--------------+--------+
4 rows in set (0.00 sec)

mysql>
```

12. List the average weight of black parts by using avg() function.

```
mysql> SELECT AVG(weight) as average_weight FROM jbparts WHERE color = "black";
+----------------+
| average_weight |
+----------------+
|       347.2500 |
+----------------+
1 row in set (0.00 sec)

mysql>
```

13. To get the name and the total weight of all parts that each supplier in Massachusetts (“Mass”) has delivered, firstly, using the table jbsupplier left join the table jbsupply to get the parts and quantity delivered, then second left join the table jbparts to retrieve the weight of each parts, then filtering the supplier in Mass using a where-cause subquery, finally calculating the total weight by using group by supplier name.

```
mysql> SELECT jbsupplier.name, SUM(jbparts.weight * jbsupply.quan) AS total_weight FROM jbsupplier
    -> LEFT JOIN jbsupply ON jbsupplier.id = jbsupply.supplier
    -> LEFT JOIN jbparts ON jbsupply.part = jbparts.id
    -> WHERE jbsupplier.city IN (SELECT id FROM jbcity WHERE state = "Mass")
    -> GROUP BY jbsupplier.name;
+--------------+--------------+
| name         | total_weight |
+--------------+--------------+
| DEC          |         3120 |
| Fisher-Price |      1135000 |
+--------------+--------------+
2 rows in set (0.00 sec)

mysql>
```

14. Firstly create a new table called new_jbitem, which has the same attributes as the table items, then add foregin keys for the tabble new_jbitem, finally fill the table with all items that cost less than the average price for items by using a subquery in the where-cause.

```
mysql> DROP TABLE IF EXISTS new_jbitem CASCADE;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
mysql> CREATE TABLE new_jbitem (
    -> id INT,
    -> name VARCHAR(20),
    -> dept INT NOT NULL,
    -> price INT,
    -> qoh INT UNSIGNED,
    -> supplier INT NOT NULL,
    -> CONSTRAINT pk_item PRIMARY KEY(id)) ENGINE=InnoDB;
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql>
mysql> ALTER TABLE new_jbitem ADD CONSTRAINT fk_new_item_dept FOREIGN KEY
(dept) REFERENCES jbdept(id);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE new_jbitem ADD CONSTRAINT fk_new_item_supplier FOREIGN
KEY (supplier) REFERENCES jbsupplier(id);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO new_jbitem
    -> SELECT * FROM jbitem WHERE price < (SELECT AVG(price) FROM jbitem);
Query OK, 14 rows affected (0.01 sec)
Records: 14  Duplicates: 0  Warnings: 0

mysql>

mysql> SELECT * FROM new_jbitem;
+------+-----------------+------+-------+------+----------+
| id   | name            | dept | price | qoh  | supplier |
+------+-----------------+------+-------+------+----------+
|   11 | Wash Cloth      |    1 |    75 |  575 |      213 |
|   19 | Bellbottoms     |   43 |   450 |  600 |       33 |
|   21 | ABC Blocks      |    1 |   198 |  405 |      125 |
|   23 | 1 lb Box        |   10 |   215 |  100 |       42 |
|   25 | 2 lb Box, Mix   |   10 |   450 |   75 |       42 |
|   26 | Earrings        |   14 |  1000 |   20 |      199 |
|   43 | Maze                    |     49 |     325 |    200 |            89 |
| 106 | Clock Book          |     49 |     198 |   150 |           125 |
| 107 | The 'Feel' Book |     35 |     225 |   225 |            89 |
| 118 | Towels, Bath       |     26 |     250 | 1000 |          213 |
| 119 | Squeeze Ball       |     49 |     250 |   400 |            89 |
| 120 | Twin Sheet          |     26 |     800 |   750 |          213 |
| 165 | Jean                    |     65 |     825 |   500 |            33 |
| 258 | Shirt                   |     58 |     650 | 1200 |            33 |
+-----+-----------------+------+-------+------+----------+
14 rows in set (0.00 sec)
```