IT252
Database Management System

Assignment IV

NIRAJ NANDISH

191IT234 | IT252 | 10/02/2021

1. Manufacturers and Products schema

i. Compute the average price of all products with manufacturer code equal to 2.

```
> SELECT AVG(Price) FROM Products WHERE Manufacturer = 2;
```

```
mysql> select avg(Price) from Products where Manufacturer = 2;
+-------------+
| avg(Price) |
+-------------+
|   150.0000 |
+-------------+
1 row in set (0.01 sec)
```

ii. Compute the number of products with a price larger than or equal to $180.

```
> SELECT COUNT(*) AS “Product Count” FROM Products WHERE Price >= 180;
```

```
mysql> select count(*) as "Product Count" from Products where Price >= 180;
+---------------+
| Product Count |
+---------------+
|             5 |
+---------------+
1 row in set (0.00 sec)
```

iii. Select the name and price of all products with a price larger than or equal to $180, and sort first by price (in descending order), and then by name (in ascending order).

```
> SELECT Name, Price FROM Products WHERE Price >= 180 ORDER BY Price DESC, Name ASC;
```

```
mysql> select Name, Price from Products where Price >= 180 order by Price desc, Name asc;
+------------+-------+
| Name       | Price |
+------------+-------+
| Printer    |   270 |
| Hard drive |   240 |
| Monitor    |   240 |
| DVD burner |   180 |
| DVD drive  |   180 |
+------------+-------+
5 rows in set (0.00 sec)
```

iv. Select all the data from the products, including all the data for each product's manufacturer.

```
> SELECT * FROM Products INNER JOIN Manufacturers ON
Products.Manufacturer = Manufacturers.Code;
```

```
mysql> select * from Products inner join Manufacturers on Products.Manufacturer = Manufacturers.Code;
+------+-----------------+-------+--------------+------+-----------------+
| Code | Name            | Price | Manufacturer | Code | Name            |
+------+-----------------+-------+--------------+------+-----------------+
|    5 | Monitor         |   240 |            1 |    1 | Sony            |
|    6 | DVD drive       |   180 |            2 |    2 | Creative Labs   |
|    7 | CD drive        |    90 |            2 |    2 | Creative Labs   |
|   10 | DVD burner      |   180 |            2 |    2 | Creative Labs   |
|    8 | Printer         |   270 |            3 |    3 | Hewlett-Packard |
|    9 | Toner cartridge |    66 |            3 |    3 | Hewlett-Packard |
|    3 | ZIP drive       |   150 |            4 |    4 | Iomega          |
|    1 | Hard drive      |   240 |            5 |    5 | Fujitsu         |
|    2 | Memory          |   120 |            6 |    6 | Winchester      |
|    4 | Floppy disk     |     5 |            6 |    6 | Winchester      |
+------+-----------------+-------+--------------+------+-----------------+
10 rows in set (0.00 sec)
```

v. Select the average price of each manufacturer's products, showing only the manufacturer's code.

```
> SELECT Manufacturer, AVG(Price) FROM Products GROUP BY
Manufacturer;
```

```
mysql> select Manufacturer, avg(Price) from Products group by Manufacturer;
+--------------+------------+
| Manufacturer | avg(Price) |
+--------------+------------+
|            1 |   240.0000 |
|            2 |   150.0000 |
|            3 |   168.0000 |
|            4 |   150.0000 |
|            5 |   240.0000 |
|            6 |    62.5000 |
+--------------+------------+
6 rows in set (0.00 sec)
```

vi. Select the product name, price, and manufacturer name of all the products.

```
> SELECT Products.Name, Products.Price, Manufacturers.Name FROM
Products INNER JOIN Manufacturers ON Products.Manufacturer =
Manufacturers.Code;
```

```
mysql> select Products.name, Products.Price, Manufacturers.Name from Products inner join Manufacturers on Products.Manufacturer = Manufacturers.Code;
+-----------------+-------+-----------------+
| name            | Price | Name            |
+-----------------+-------+-----------------+
| Monitor         |   240 | Sony            |
| DVD drive       |   180 | Creative Labs   |
| CD drive        |    90 | Creative Labs   |
| DVD burner      |   180 | Creative Labs   |
| Printer         |   270 | Hewlett-Packard |
| Toner cartridge |    66 | Hewlett-Packard |
| ZIP drive       |   150 | Iomega          |
| Hard drive      |   240 | Fujitsu         |
| Memory          |   120 | Winchester      |
| Floppy disk     |     5 | Winchester      |
+-----------------+-------+-----------------+
10 rows in set (0.00 sec)
```

vii. Select the names of manufacturer whose products have an average price larger than or equal to $150.

```
> SELECT Manufacturers.Name FROM Products INNER JOIN Manufacturers ON
Products.Manufacturer = Manufacturers.Code GROUP BY
Manufacturers.Code HAVING AVG(Products.Price) >= 150;
```

```
mysql> select Manufacturers.Name from Products inner join Manufacturers on Products.Manufacturer = Manufacturers.Code group by Manufacturers.Code having avg
(Products.Price) >= 150;
+-----------------+
| Name            |
+-----------------+
| Sony            |
| Creative Labs   |
| Hewlett-Packard |
| Iomega          |
| Fujitsu         |
+-----------------+
5 rows in set (0.00 sec)
```

viii. Select the name and price of the cheapest product.

```
> SELECT Name, Price FROM Products ORDER BY Price ASC LIMIT 1;
```

```
mysql> select Name, Price from Products order by Price asc limit 1;
+-------------+-------+
| Name        | Price |
+-------------+-------+
| Floppy disk |     5 |
+-------------+-------+
1 row in set (0.00 sec)
```

ix. Select the name of each manufacturer along with the name and price of its most expensive product.

```
> SELECT M.Name AS “Manufacturer Name”, P.Name AS “Product Name”, P.Price FROM Products P INNER JOIN Manufacturers M ON M.Code = P.Manufacturer AND P.Price = (SELECT MAX(P.Price) FROM Products P WHERE P.Manufacturer = M.Code);
```

```
mysql> select M.Name as "Manufacturer Name", P.Name as "Product Name", P.Price from Products P inner join Manufacturers M on M.Code = P.Manufacturer and P.Price = ( select max(P.Price) from Products P where P.Manufacturer = M.Code);
+------------------+--------------+-------+
| Manufacturer Name | Product Name | Price |
+------------------+--------------+-------+
| Sony             | Monitor      |   240 |
| Creative Labs    | DVD drive    |   180 |
| Creative Labs    | DVD burner   |   180 |
| Hewlett-Packard  | Printer      |   270 |
| Iomega           | ZIP drive    |   150 |
| Fujitsu          | Hard drive   |   240 |
| Winchester       | Memory       |   120 |
+------------------+--------------+-------+
7 rows in set (0.00 sec)
```

x. Apply a 10% discount to all products with a price larger than or equal to $120.

```
> UPDATE Products SET Price = Price * 0.9 WHERE Price >= 120;
```

```
mysql> update Products set Price = Price * 0.9 where Price >= 120;
Query OK, 7 rows affected (0.07 sec)
Rows matched: 7  Changed: 7  Warnings: 0

mysql> select * from Products;
+------+-----------------+-------+--------------+
| Code | Name            | Price | Manufacturer |
+------+-----------------+-------+--------------+
|    1 | Hard drive      |   216 |            5 |
|    2 | Memory          |   108 |            6 |
|    3 | ZIP drive       |   135 |            4 |
|    4 | Floppy disk     |     5 |            6 |
|    5 | Monitor         |   216 |            1 |
|    6 | DVD drive       |   162 |            2 |
|    7 | CD drive        |    90 |            2 |
|    8 | Printer         |   243 |            3 |
|    9 | Toner cartridge |    66 |            3 |
|   10 | DVD burner      |   162 |            2 |
+------+-----------------+-------+--------------+
10 rows in set (0.00 sec)
```

2. Departments and Employees schema

i. Select the number of employees in each department (you only need to show the department code and the number of employees).

```
> SELECT Department, COUNT(*) AS “Count of Employees” FROM Employees
GROUP BY Department;
```

```
mysql> select Department, count(*) as "Count of Employees" from Employees group by Department;
+------------+--------------------+
| Department | Count of Employees |
+------------+--------------------+
|         14 |                  5 |
|         37 |                  2 |
|         59 |                  3 |
|         77 |                  2 |
+------------+--------------------+
4 rows in set (0.00 sec)
```

ii. Select the name and last name of each employee, along with the name and budget of the employee's department.

```
> SELECT E.Name, E.LastName, D.Name, D.Budget FROM Employees E INNER
JOIN Departments D on E.Department = D.Code;
```

```
mysql> select E.Name, E.LastName, D.Name, D.Budget from Employees E inner join Departments D on E.Department = D.Code;
+-----------+-----------+-----------------+--------+
| Name      | LastName  | Name            | Budget |
+-----------+-----------+-----------------+--------+
| Michael   | Rogers    | IT              |  65000 |
| Anand     | Manikutty | IT              |  65000 |
| MaryAnne  | Foster    | IT              |  65000 |
| Elizabeth | Doe       | IT              |  65000 |
| Kumar     | Swamy     | IT              |  65000 |
| Carol     | Smith     | Accounting      |  15000 |
| Joe       | Stevens   | Accounting      |  15000 |
| John      | Doe       | Human Resources | 240000 |
| Zacary    | Efron     | Human Resources | 240000 |
| Eric      | Goldsmith | Human Resources | 240000 |
| George    | ODonnell  | Research        |  55000 |
| David     | Smith     | Research        |  55000 |
+-----------+-----------+-----------------+--------+
12 rows in set (0.00 sec)
```

iii. Select the name and last name of employees working for departments with a budget greater than $60,000.

```
> SELECT E.Name, E.LastName FROM Employees E INNER JOIN Departments D
ON E.Department = D.Code WHERE D.Budget > 60000;
```

```
mysql> select E.Name, E.LastName from Employees E inner join Departments D on E.Department = D.Code where D.Budget > 60000;
+-----------+-----------+
| Name      | LastName  |
+-----------+-----------+
| Michael   | Rogers    |
| Anand     | Manikutty |
| MaryAnne  | Foster    |
| Elizabeth | Doe       |
| Kumar     | Swamy     |
| John      | Doe       |
| Zacary    | Efron     |
| Eric      | Goldsmith |
+-----------+-----------+
8 rows in set (0.00 sec)
```

iv. Select the departments with a budget larger than the average budget of all the departments.

```
> SELECT * FROM Departments WHERE Budget > (SELECT AVG(Budget) FROM
Departments);
```

```
mysql> select * from Departments where Budget > (select avg(Budget) from Departments);
+------+-----------------+--------+
| Code | Name            | Budget |
+------+-----------------+--------+
|   59 | Human Resources | 240000 |
+------+-----------------+--------+
1 row in set (0.00 sec)
```

v. Select the names of departments with more than two employees.

```
> SELECT Name FROM Departments WHERE Code in (SELECT Department FROM
Employees GROUP BY Department HAVING COUNT(*) > 2);
```

```
mysql> select Name from Departments where Code in (select Department from Employees group by Department having count(*) > 2);
+-----------------+
| Name            |
+-----------------+
| IT              |
| Human Resources |
+-----------------+
2 rows in set (0.01 sec)
```

vi. Select the name and last name of employees working for departments with second lowest budget.

```
> SELECT Name, LastName FROM Employees WHERE Department = (SELECT
Code FROM Departments ORDER BY Budget ASC LIMIT 1,1);
```

```
mysql> select Name, LastName from Employees where Department = (select Code from Departments order by Budget asc limit 1,1);
+--------+----------+
| Name   | LastName |
+--------+----------+
| George | ODonnell |
| David  | Smith    |
+--------+----------+
2 rows in set (0.00 sec)
```

vii. Reduce the budget of all departments by 10%.

```
> UPDATE Departments SET Budget = Budget * 0.9;
```

```
mysql> update Departments set Budget = Budget * 0.9;
Query OK, 4 rows affected (0.03 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> select * from Departments;
+------+-----------------+--------+
| Code | Name            | Budget |
+------+-----------------+--------+
|   14 | IT              |  58500 |
|   37 | Accounting      |  13500 |
|   59 | Human Resources | 216000 |
|   77 | Research        |  49500 |
+------+-----------------+--------+
4 rows in set (0.01 sec)
```

viii. Reassign all employees from the Research department (code 77) to the IT department (code 14).

```
> UPDATE Employees SET Department = 14 WHERE Department = 77;
```

```
mysql> update Employees set Department = 14 where Department = 77;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from Employees;
+-----------+-----------+-----------+------------+
| SSN       | Name      | LastName  | Department |
+-----------+-----------+-----------+------------+
| 123234877 | Michael   | Rogers    |         14 |
| 152934485 | Anand     | Manikutty |         14 |
| 222364883 | Carol     | Smith     |         37 |
| 326587417 | Joe       | Stevens   |         37 |
| 332154719 | MaryAnne  | Foster    |         14 |
| 332569843 | George    | ODonnell  |         14 |
| 546523478 | John      | Doe       |         59 |
| 631231482 | David     | Smith     |         14 |
| 654873219 | Zacary    | Efron     |         59 |
| 745685214 | Eric      | Goldsmith |         59 |
| 845657245 | Elizabeth | Doe       |         14 |
| 845657246 | Kumar     | Swamy     |         14 |
+-----------+-----------+-----------+------------+
12 rows in set (0.00 sec)
```

ix. Add a new department called "Quality Assurance", with a budget of $40,000 and departmental code 11. And Add an employee called "Mary Moore" in that department, with SSN 847-21-9811.

```
> INSERT INTO Departments VALUES (11, “Quality Assurance”, 40000);
> INSERT INTO Employees VALUES (847219811, “Mary”, “Moore”, 11);
```

```
mysql> insert into Departments values (11, "Quality Assurance", 40000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Employees values (847219811, "Mary", "Moore", 11);
Query OK, 1 row affected (0.01 sec)

mysql> select * from Departments;
+------+-------------------+--------+
| Code | Name              | Budget |
+------+-------------------+--------+
|   11 | Quality Assurance |  40000 |
|   14 | IT                |  58500 |
|   37 | Accounting        |  13500 |
|   59 | Human Resources   | 216000 |
|   77 | Research          |  49500 |
+------+-------------------+--------+
5 rows in set (0.00 sec)

mysql> select * from Employees;
+-----------+-----------+-----------+------------+
| SSN       | Name      | LastName  | Department |
+-----------+-----------+-----------+------------+
| 123234877 | Michael   | Rogers    |         14 |
| 152934485 | Anand     | Manikutty |         14 |
| 222364883 | Carol     | Smith     |         37 |
| 326587417 | Joe       | Stevens   |         37 |
| 332154719 | MaryAnne  | Foster    |         14 |
| 332569843 | George    | ODonnell  |         14 |
| 546523478 | John      | Doe       |         59 |
| 631231482 | David     | Smith     |         14 |
| 654873219 | Zacary    | Efron     |         59 |
| 745685214 | Eric      | Goldsmith |         59 |
| 845657245 | Elizabeth | Doe       |         14 |
| 845657246 | Kumar     | Swamy     |         14 |
| 847219811 | Mary      | Moore     |         11 |
+-----------+-----------+-----------+------------+
13 rows in set (0.00 sec)
```

x. Delete from the table all employees who work in departments with a budget greater than or equal to $60,000.

```
> DELETE FROM Employees WHERE Department IN (SELECT Code FROM
Departments WHERE Budget >= 60000);
```

```
mysql> delete from Employees where Department in (select Code from Departments where Budget >= 60000);
Query OK, 3 rows affected (0.01 sec)

mysql> select * from Employees;
+-----------+-----------+-----------+------------+
| SSN       | Name      | LastName  | Department |
+-----------+-----------+-----------+------------+
| 123234877 | Michael   | Rogers    |         14 |
| 152934485 | Anand     | Manikutty |         14 |
| 222364883 | Carol     | Smith     |         37 |
| 326587417 | Joe       | Stevens   |         37 |
| 332154719 | MaryAnne  | Foster    |         14 |
| 332569843 | George    | ODonnell  |         14 |
| 631231482 | David     | Smith     |         14 |
| 845657245 | Elizabeth | Doe       |         14 |
| 845657246 | Kumar     | Swamy     |         14 |
| 847219811 | Mary      | Moore     |         11 |
+-----------+-----------+-----------+------------+
10 rows in set (0.00 sec)
```