0801IT211064 : Pranjal Jatav

LAB ASSIGNMENT – 04

Table Name: Manufacturer

Attri	bute	Data Type	Primary	Foreign	Constraint
Coe	de	Int	Y		Not Null
Nar	ne	Char			Not Null

Table Name: Products

Attribute	Data Type	Primary	Foreign	Constraint
Code	INT	Y		Not Null
Name	Char			Not Null
Price	Decimal			Not Null
Manufacturer	Int		Manufacturer(Code)	Not Null

Insert following data into Manufacturer Table.

Code	Name
1	Sony
2	Creative Labs
3	Hewlett-Packard
4	Iomega
5	Fujitsu
6	Winchester

Insert following data into Products Table.

Code	Name	Price	Manufacturer
1	Hard Drive	240	5
2	Memory	120	6
3	Zip Drive	150	4
4	Floppy Disk	5	6
5	Monitor	240	1
6	DVD Drive	180	2
7	CD Drive	90	2
8	Printer	270	3
9	Tone Cartridge	66	3
10	DVD Burner	180	2

mysql> use sonali it86;

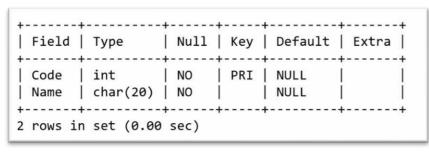
Database changed

mysql> CREATE TABLE Manufacturer(

- -> Code INT NOT NULL,
 - -> Name CHAR(20) NOT NULL,
 - -> PRIMARY KEY (Code)
 - ->);

Query OK, 0 rows affected (0.04 sec)

mysql> DESCRIBE Manufacturer;



mysql> CREATE TABLE Products(

- -> Code INT NOT NULL,
 - -> Name CHAR(20) NOT NULL,
 - -> Price DECIMAL(3,0) NOT NULL,
 - -> Manufacturers INT NOT NULL,
 - -> PRIMARY KEY (Code),
 - -> FOREIGN KEY (Manufacturer) REFERENCES Manufacturer(Code)
 - ->);

Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE Products;

Field	Type +	•		Default +	•
Code	int	NO NO	PRI	NULL	
Name	char(20)	NO		NULL	
Price	decimal(3,0)	NO		NULL	
Manufacturer	int	NO	MUL	NULL	
	+	+	+	+	+

mysql> INSERT INTO Manufacturer

- -> VALUES
- -> (1, "Sony"),
- -> (2,"Creative Labs"),
- -> (3,"Hewlett-Packard"),
- -> (4,"Iomega"),
- -> (5,"Fujitsu"),
- -> (6,"Winchester");

Query OK, 6 rows affected (0.02 sec) Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM Manufacturer;

mysql> INSERT INTO Products

- -> VALUES
 - -> (1, "Hard Drive", 240, 5),
 - -> (2, "Memory", 120, 6),
 - -> (3, "Zip Drive", 150, 4),
 - -> (4, "Floppy Disk", 5, 6),
 - -> (5, "Monitor", 240, 1),

```
-> (6, "DVD Drive", 180, 2),
```

- -> (7, "CD Drive", 90, 2),
- -> (8, "Printer", 270, 3),
- -> (9, "Tone Cartridge", 66, 3),
- -> (10, "DVD Burner", 180, 2);

```
Query OK, 10 rows affected (0.04 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

mysql> SELECT * FROM Products;

Code	Name	Price	Manufacturer
		+	+
1	Hard Drive	240	5
2	Memory	120	6
3	Zip Drive	150	4
4	Floppy Disk	5	6
5	Monitor	240	1
6	DVD Drive	180	2
7	CD Drive	90	2
8	Printer	270	3
9	Tone Cartridge	66	3
10	DVD Burner	180	2
		+	

Q.1. Select the names of all product in the store.

mysql> select DISTINCT Name

-> from Products;



Q.2. Select the names and the prices of all the products in the store.

mysql> select Name, Price

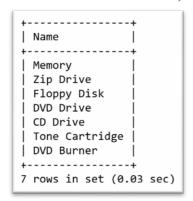
-> from Products;

+	++
Name	Price
+	++
Hard Drive	240
Memory	120
Zip Drive	150
Floppy Disk	5
Monitor	240
DVD Drive	180
CD Drive	90
Printer	270
Tone Cartridge	66
DVD Burner	180
+	++
10 rows in set (0.	.00 sec)

Q.3. Select the name of products with price less than or equal to \$200.

mysql> select Name

- -> from Products
- -> WHERE Price <= 200;



Q.4. Select all the products with a price between \$60 and \$120.

mysql> select *

- -> from Products
- -> WHERE Price BETWEEN 60 AND 120;

+	Name	+ Price	++ Manufacturer
7	Memory CD Drive Tone Cartridge	+ 120 90 66	6 2 3
3 rows	in set (0.01 sec)		

Q.5. Select name and price in cents(i.e. the price multiply by 100).

mysql> select Name, Price*100 AS "Price in Cents"

-> from Products;

Name	Price in Cents
Hard Drive	24000
Memory	12000
Zip Drive	15000
Floppy Disk	500
Monitor	24000
DVD Drive	18000
CD Drive	9000
Printer	27000
Tone Cartridge	6600
DVD Burner	18000
++	+

Q.6. Compute the average price of all the product.

mysql> select AVG(Price)

-> from Products;

```
+-----+
| AVG(Price) |
+-----+
| 154.1000 |
+-----+
1 row in set (0.04 sec)
```

Q.7. Compute average price of all products with manufacturer code equal to 2.

mysql> select AVG(Price)

- -> from Products
- -> WHERE Manufacturer = 2;

```
+----+
| AVG(Price) |
+-----+
| 150.0000 |
+-----+
1 row in set (0.01 sec)
```

Q.8. Compute the number of products with a price larger then or equal to \$180.

mysql> select COUNT(*)

-> from Products

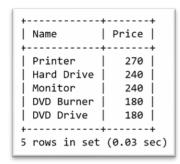
-> WHERE Price >= 180;

```
+-----+
| COUNT(*) |
+------+
| 5 |
+-----+
1 row in set (0.03 sec)
```

Q.9. Select the name and price of all products with a price larger then or equal to \$180 and, sort first by price (in descending order), and then by name(in ascending order).

mysql> select Name, Price

- -> from Products
 - -> WHERE Price >= 180
 - -> ORDER BY Price DESC, Name ASC;



Q.10. Select all the data from products, including all the data for each products manufacturer.

mysql> select Products.*, Manufacturer.Name

- -> from Products
- -> LEFT JOIN Manufacturer ON
- -> Products.Manufacturer = Manufacturer.Code;

Code	Name	Price	Manufacturer	Name
1	Hard Drive	+ 240	5	Fujitsu
2	Memory	120	6	Winchester
3	Zip Drive	150	4	Iomega
4	Floppy Disk	5	6	Winchester
5	Monitor	240	1	Sony
6	DVD Drive	180	2	Creative Labs
7	CD Drive	90	2	Creative Labs
8	Printer	270	3	Hewlett-Packard
9	Tone Cartridge	66	3	Hewlett-Packard
10	DVD Burner	180	2	Creative Labs

Q. 11. Select the product name, Price, and manufacturer names of all products.

mysql> select Products.Name, Products.Price, Manufacturer.Name AS "Manufacturer Name"

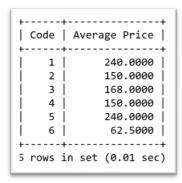
- -> from Products
- -> LEFT JOIN Manufacturer ON
- -> Products.Manufacturer = Manufacturer.Code;

Name	Price	Manufacturer Name
Hard Drive	240	Fujitsu
Memory	120	Winchester
Zip Drive	150	Iomega
Floppy Disk	5	Winchester
Monitor	240	Sony
DVD Drive	180	Creative Labs
CD Drive	90	Creative Labs
Printer	270	Hewlett-Packard
Tone Cartridge	66	Hewlett-Packard
DVD Burner	180	Creative Labs
	+	+

Q. 12. Select the average price of each manufacturers products, showing only the manufacturer code.

mysql> select Manufacturer.Code, AVG(Products.Price) AS "Average Price"

- -> from Products
- -> JOIN Manufacturer ON
- -> Products.Manufacturer = Manufacturer.Code
- -> GROUP BY Manufacturer.Code;



Q. 13. Select the average price of each manufacturers products, Showing the manufacturers name.

mysql> select Manufacturer.Name, AVG(Products.Price)

- -> from Manufacturer
- -> JOIN Products ON

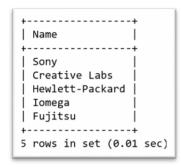
- -> Products.Manufacturer = Manufacturer.Code
- -> GROUP BY Manufacturer.Name;

Name	AVG(Products.Price)
Sony	240.0000
Creative Labs	150.0000
Hewlett-Packard	168.0000
Iomega	150.0000
Fujitsu	240.0000
Winchester	62.5000
+	

Q. 14. Select the name of manufacturers whose product have an average price larger than or equal to \$150.

mysql> select Manufacturer.Name

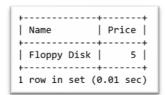
- -> from Products
- -> JOIN Manufacturer ON
- -> Products.Manufacturer = Manufacturer.Code
- -> GROUP BY Manufacturer.Name
- -> HAVING AVG(Products.Price) >= 150;



Q. 15. Select the name and price of the cheapest product.

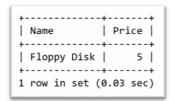
```
mysql> select Name, Price
```

- -> from Products
 - -> WHERE Price = (
 - -> select MIN(Price)
 - -> from Products
 - ->);



mysql> select Name, Price

- -> from Products
 - -> ORDER BY Price
 - -> LIMIT 1;



Q. 16. Select the name of each manufacturer along with the name and price of its most expensive product.

mysql> select M.Name, P.Name, P.Price

- -> from Products P
- -> JOIN Manufacturer M ON
- -> P.Manufacturer = M.Code
- -> JOIN(
- -> select Manufacturer, MAX(Price) AS Max_Price
- -> from Products
- -> GROUP BY Manufacturer
- ->) AS Max ON P.Manufacturer = Max.Manufacturer AND P.Price =

Max.Max Price;

Q. Add a

Name	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
+	+	++

product: Loudspeakers, \$70, Manufacturer 2.

mysql> INSERT INTO Products

- -> VALUES
- -> (11, "Loudspeakers", 70, 2);

Query OK, 1 row affected (0.03 sec)

17. new mysql> select *

-> from Products;

Code	Name	Price	Manufacturer
1	+ Hard Drive	+ 240	
2	Memory	120	6
3	Zip Drive	150	4
4	Floppy Disk	5	6
5	Monitor	240	1
6	DVD Drive	180	2
7	CD Drive	90	2
8	Printer	270	3
9	Tone Cartridge	66	3
10	DVD Burner	180	2
11	Loudspeakers	70	2
+	+	+	++
11 rows	in set (0.00 sec)	

Q. 18. Update the name of product 8 to "Laser printer".

```
mysql> UPDATE Products -> SET Name = "Laser printer" -> WHERE Code = 8;
```

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

mysql> select *

-> from Products;

Code	Name	Price	Manufacturer
	+	+	+
1	Hard Drive	240	5
2	Memory	120	6
3	Zip Drive	150	4
4	Floppy Disk	5	6
5	Monitor	240	1
6	DVD Drive	180	2
7	CD Drive	90	2
8	Laser printer	270	3
9	Tone Cartridge	66	3
10	DVD Burner	180	2
11	Loudspeakers	70	2
	+	+	·

Q. 19. Apply a 10% discount to all products.

mysql> select Code, Name, (9*Price)/10 AS "10% discount", Manufacturer -> from Products;

Code	Name	10% discount	Manufacturer
+		+	+
1	Hard Drive	216.0000	5
2	Memory	108.0000	6
3	Zip Drive	135.0000	4
4	Floppy Disk	4.5000	6
5	Monitor	216.0000	1
6	DVD Drive	162.0000	2
7	CD Drive	81.0000	2
8	Laser printer	243.0000	3
9	Tone Cartridge	59.4000	3
10	DVD Burner	162.0000	2
11	Loudspeakers	63.0000	2

Q. 20. Apply a 10% discount to all products with a price larger than or equal to \$120.

mysql> select Code, Name, (9*Price)/10, Manufacturer

- -> from Products
- -> WHERE Price >= 120;

Code	Name	(9*Price)/10	Manufacturer
1	Hard Drive	216.0000	5
2	Memory	108.0000	6
3	Zip Drive	135.0000	4
5	Monitor	216.0000	1
6	DVD Drive	162.0000	2
8	Laser printer	243.0000	3
10	DVD Burner	162.0000	2
+		+	