

[table definitions are given at the end of the document.]

Sql queries: [lab5]

[Referred tables: 'products', 'manufacturer']

i. Select the names of all the products in the store.

```
SELECT NAME
FROM PRODUCTS;
```

ii. Select the names and the prices of all the products in the store.

```
SELECT NAME, PRICE
FROM PRODUCTS;
```

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

```
SELECT NAME
FROM PRODUCTS
WHERE PRICE<=200;
```

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

```
SELECT *
FROM PRODUCTS
WHERE PRICE >=60 AND PRICE<=120;
```

v. Select the name and price in cents (i.e., the price must be multiplied by 100).

```
SELECT PRICE *100,NAME
FROM PRODUCTS;
```

vi. Compute the average price of all the products.

```
SELECT AVG(PRICE)
FROM PRODUCTS;
```

vii. Try the above task without using avg () function. [Use any other aggregate function!]

```
SELECT SUM(PRICE)/COUNT(PRICE)
FROM PRODUCTS;
```

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

```
SELECT AVG(PRICE)
FROM PRODUCTS
WHERE MANUFACTURER=2;
```

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

```
SELECT COUNT(NAME)
FROM PRODUCTS
WHERE PRICE>=180;
```

x. Select the name and price of all products with a price larger than or equal to \$180.

```
SELECT NAME, PRICE
FROM PRODUCTS
WHERE PRICE>=180;
```

xi. 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).

```
SELECT NAME, PRICE
FROM PRODUCTS
WHERE PRICE>=180
ORDER BY PRICE DESC;
```

xii. 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;
```

xiii. [Did u notice the difference in the output of X, XI, XII? Did u understand why it happened?]

xiv. Select the name and price of the cheapest product. [Hint: read the max() and min() function handbook]

```
SELECT PRICE, NAME
FROM PRODUCTS
WHERE PRICE=(SELECT MIN(PRICE) FROM PRODUCTS );
```

xv. Select the product name, price, and manufacturer name of all the products. [Hint: use joining concepts]

```
SELECT P.NAME, P.PRICE, M.NAME
FROM PRODUCTS P, MANUFACTURERS M
WHERE P.MANUFACTURER= M.CODE;
```

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

```
SELECT AVG(PRICE), MANUFACTURER
FROM PRODUCTS P
GROUP BY MANUFACTURER;
```

xvii. Select the average price of each manufacturer's products, showing the manufacturer's name. [Hint: joining and group by]

```
SELECT AVG(PRICE), M.NAME
FROM PRODUCTS P, MANUFACTURERS M
WHERE P.MANUFACTURER= M.CODE
GROUP BY M.NAME;
```

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

```
SELECT AVG(PRICE), M.NAME
FROM PRODUCTS P, MANUFACTURERS M
WHERE P.MANUFACTURER= M.CODE
GROUP BY M.NAME
HAVING AVG(P.PRICE)>=150;
```

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

```
SELECT M.NAME, MAX(P.PRICE)
FROM PRODUCTS P, MANUFACTURERS M
WHERE P.MANUFACTURER= M.CODE
GROUP BY M.NAME;
```

xx. Select the name of each manufacturer along with the price of its most expensive product. Then order it by the maximum price of each category

```
SELECT M.NAME, MAX(P.PRICE)
FROM PRODUCTS P, MANUFACTURERS M
WHERE P.MANUFACTURER= M.CODE
GROUP BY M.NAME
ORDER BY MAX(P.PRICE);
```

xxi. Update the name of product 8 to "Laser Printer".

```
UPDATE PRODUCTS
SET NAME='LASER PRINTER'
WHERE CODE=8;
```

xxii. What will be the price if we apply a 10% discount to all products.[show it using modification in the select query]

```
SELECT NAME, PRICE*0.9
FROM PRODUCTS;
```

xxiii. Use 'update' statement to modify the price of all product with 10% discount!

```
UPDATE PRODUCTS
SET PRICE = PRICE*0.9;
```

xxiv. 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;
```

[DONT CARE THE WIERD TABLE DEFINITIONS!]

[TABLE:: MANUFACTURERS]

```
CREATE TABLE Manufacturers (
Code INTEGER,
Name VARCHAR (255) NOT NULL,
PRIMARY KEY (Code)
);
```

```
INSERT INTO Manufacturers (Codename) VALUES (1,'Sony');
INSERT INTO Manufacturers (Codename) VALUES (2,'Creative Labs');
INSERT INTO Manufacturers (Codename) VALUES (3,'Hewlett-Packard');
INSERT INTO Manufacturers (Codename) VALUES (4,'Iomega');
INSERT INTO Manufacturers (Codename) VALUES (5,'Fujitsu');
INSERT INTO Manufacturers (Codename) VALUES (6,'Winchester');
```

[TABLE:: PRODUCTS]

```
CREATE TABLE Products (
Code INTEGER,
Name VARCHAR (255) NOT NULL,
Price DECIMAL NOT NULL,
Manufacturer INTEGER NOT NULL,
PRIMARY KEY (Code),
FOREIGN KEY (Manufacturer) REFERENCES Manufacturers (Code)
);
```

```
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES (1,'Hard
drive', 240, 5);
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES
(2,'Memory', 120, 6);
INSERT INTO Products (Code, Name, Price, Manufacturer) VALUES (3,'ZIP
drive',150,4);
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES (4,'Floppy
disk', 5, 6);
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES
(5,'Monitor', 240, 1);
INSERT INTO Products (Code, Name, Price, Manufacturer) VALUES (6,'DVD
drive', 180, 2);
INSERT INTO Products (Code, Name, Price, Manufacturer) VALUES (7,'CD drive',
90, 2);
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES
(8,'Printer', 270, 3);
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES (9,'Toner
```

dbms_lab5_solution.txt

```
cartridge', 66, 3);  
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES (10, 'DVD  
burner', 180, 2);  
INSERT INTO Products (Code, Name, Price, and Manufacturer) VALUES (11, 'Card  
Reader', 180, 2);
```