

### Problem 1

6.4.6 a)

```
SELECT AVG(speed)
FROM PC;
```

AVG(speed)
2.4846

6.4.6 b)

```
SELECT AVG(speed)
FROM Laptop
WHERE price > 1000;
```

AVG(speed)
1.9983

6.4.6 c)

```
SELECT AVG(price)
FROM PC, Product
WHERE PC.model = Product.model AND Product.maker = 'A';
```

AVG(Price)
1195.6666

6.4.6 d)

```
SELECT AVG(systems.price)
FROM
((SELECT pc.price FROM pc, product WHERE pc.model = product.model and product.maker =
'D')
UNION
(SELECT laptop.price FROM laptop, product WHERE laptop.model = product.model and
product.maker = 'D')) systems;
```

AVG(Price)
730.00

6.4.6 e)

```
SELECT speed, AVG(price)
FROM PC
GROUP BY speed;
```

Speed	Average Price
2.66	2114.00
2.10	995.00
1.42	478.00
2.80	689.33
3.20	839.50
2.20	640.00
2.00	650.00
1.86	959.00
3.06	529.00

6.4.6 f)

```
SELECT maker, avg(screen)
FROM laptop, product
WHERE product.model = laptop.model
GROUP BY product.maker;
```

Maker	Average Screen Size
A	15.23
B	13.3
E	17.5
F	14.75
G	15.4

6.4.6 g)

```
SELECT maker  
FROM product  
WHERE type = 'pc'  
GROUP BY maker  
HAVING count(model) >= 3;
```

Maker
A
B
D
E

6.4.6 h)

```
SELECT maker, MAX(price)  
FROM Product,PC  
WHERE PC.model = Product.model  
GROUP BY maker;
```

Maker	MAX(price)
A	2114
B	1049
C	510
D	770
E	959

6.4.6 i)

```
SELECT speed, avg(price)
FROM PC
WHERE speed > 2.0
GROUP BY speed;
```

Speed	Average Price
2.66	2114
2.10	995
2.80	689.33
3.20	839.50
2.20	640
3.06	529

## Problem 2

6.5.1 a)

```
insert into product
values ('C', '1100', 'pc');
insert into pc
values ('1100', 3.2, 1024, 180, 2499);
```

**Effect:** The first insert statement adds a row into the product table with value 'C' in the maker and 1100 in the model and 'pc' as type. The second insert statement adds a row to the pc table with 1100 as a model, 3.2 as speed, 1024 as ram, 180 as hd, and 2499 as price.

6.5.1 b)

```
insert into Laptop(SELECT model + 1100 AS model, speed, ram, hd, '17', price + 500 AS price
FROM pc);
```

```
insert into Product(SELECT maker, model + 1100, 'laptop'
FROM pc, product
WHERE pc.model = product.model AND type = 'pc');
```

**Effect:** The first statement selects each row from the pc table and inserts one corresponding row into the laptop table. For each inserted row the model number of the laptop will be model number of pc + 1100 and price of each row inserted to laptop will be price of pc row + 500 and remaining fields speed, ram, and hd values will be taken as is from pc table. Screen size for each row inserted to the laptop will have a fixed value 17.

6.5.1 c)

```
DELETE  
FROM pc  
WHERE hd < 100;
```

**Effect:** All the rows from the pc table with hd less than 100 are deleted. There are two rows in the pc table that have hd less than 100 and these two rows will be deleted.

6.5.1 d)

```
DELETE  
FROM laptop  
JOIN Product ON laptop.model = product.model  
WHERE product.maker NOT IN (SELECT maker FROM Printer, Product WHERE printer.model  
= product.model AND type = 'printer')
```

**Effect:** Laptop table is joined with the product table to get the makers of the laptops and the subquery that joins printer and product tables returns the makers of the printer not in clause excludes the makers of laptops that match with the makers of printers and delete remaining makers laptops rows from laptop table.

6.5.1 e)

```
UPDATE product  
SET maker = 'A'  
WHERE maker = 'B';
```

**Effect:** All the rows in the product table that have maker as B will be updated to maker A. After the update there will not be any rows in the product table with maker B.

6.5.1 f)

```
UPDATE PC  
SET ram = ram * 2, hd = hd + 60;
```

**Effect:** Each and every row in the PC table will be updated and for each row ram value will be doubled and each row's hd value will be increased by 60.

6.5.1 g)

```
UPDATE laptop  
SET screen = screen + 1, price = price - 100  
WHERE model IN (SELECT model FROM product WHERE maker = 'B');
```

**Effect:** All laptops made by manufacturer B are updated to have a screen size increased by one and price reduced by 100 in the laptop table.

