More SQL

After you finish your Thanksgiving dinner, you realize that you still need to buy gifts for all your loved ones over the holidays. However, you also want to spend as little money as possible (you're not cheap, just looking for a great sale!).

This question utilizes the following tables:

products

category	name	MSRP	rating
phone	uPhone	99.99	4.5
…	…	…	…
computer	kBook	99.99	3.8

inventory

store	item	price
Hallmart	uPhone	99.99
Targive	uPhone	100.99
RestBuy	uPhone	89.99
…	…	…
RestBuy	kBook	94.99

stores

store	address	Mbs
Hallmart	50 Lawton Way	25
Targive	2 Red Circle Way	40
RestBuy	1 Kiosk Ave	30

Q1: Price Check

Let's start off by surveying our options. Using the `products` table, write a query that creates a table `average_prices` that lists categories and the average price of items in the category (using MSRP as the price).

You should get the following output:

```
computer|109.0
games|350.0
phone|90.0
```

```
SELECT category as category, AVG(msrp) as average_price FROM products GROUP BY category;
    OR
SELECT category, SUM(msrp)/COUNT(*) FROM products GROUP BY category;
```

Q2: Status of Inventory

We want to see which products are listed in the inventory of any store. Create a table `inventory_by_store`, which lists every product's name and msrp. If the product is available in a store, also include the store name and the price it is sold for. Otherwise, if a product is not in an inventory, the store and price columns will have a NULL value.

Here's a partial table of the output:

```
uPhone|99.99|Hallmart|99.99
uPhone|99.99|Targive|100.99
kBook|99.99|RestBuy|94.99
OtherProduct|149.99|NULL|NULL
```

```
SELECT p.name, p.msrp, i.store, i.price FROM products p
LEFT JOIN inventory i ON p.name = i.item;
```

Q3: The Price is Right

Now, you want to figure out which stores sell each item in products for the lowest price. Write a SQL query that uses the `inventory` table to create a table `lowest_prices` that lists items, the stores that sells that item for the lowest price, and the price that the store sells that item for.

You should expect the following output:

```
Hallmart|GameStation|298.98
Targive|QBox|390.98
Targive|iBook|110.99
RestBuy|kBook|94.99
Hallmart|qPhone|85.99
Hallmart|rPhone|69.99
RestBuy|uPhone|89.99
RestBuy|wBook|114.29
```

```
SELECT store, item, MIN(price) FROM inventory GROUP BY item; OR
SELECT * FROM inventory GROUP BY item HAVING MIN(price);
```

Q4: Bang for your Buck

You want to make a shopping list by choosing the item that is the best deal possible for every category. For example, for the "phone" category, the uPhone is the best deal because the MSRP price of a uPhone divided by its ratings yields the lowest cost. That means that uPhones cost the lowest money per rating point out of all of the phones. Note that the item with the lowest MSRP price may not necessarily be the best deal.

Write a query to create a table `shopping_list` that lists the items that you want to buy from each category.

After you've figured out which item you want to buy for each category, add another column that lists the store that sells that item for the lowest price.

You should expect the following output:

```
GameStation|Hallmart
uPhone|RestBuy
wBook|RestBuy
```

```
SELECT name, store FROM products AS p, lowest_prices AS l
WHERE l.item = p.name
GROUP BY category HAVING MIN(MSRP/rating);
```

Q5: Driving the Cyber Highways

Using the Mbs (megabits) column from the `stores` table, write a query to calculate the total amount of bandwidth needed to get everything in your shopping list.

```
SELECT SUM(s.mbs) FROM stores AS s, shopping_list AS sl WHERE s.store = sl.store;
```

Document the Occasion

Please all fill out the attendance form (one submission per person per week).

Note: This worksheet is a problem bank—most TAs will not cover all the problems in discussion section.