1. Adding text attribute and full-text search

```
project1=> ALTER TABLE product ADD COLUMN using_review varchar(200);
ALTER TABLE
```

```
project1=> select * from product;
 product_id | category_id |            name             | price |       address        |          description           | using_review
------------+-------------+-----------------------------+-------+----------------------+--------------------------------+--------------
 88910009   | 130         | Samsung 32 Inc              |   109 | 106 Camelot Dr       | new 32 inches Samsung smart tv |
 89110024   | 130         | Hp omen gaming desktop      |   789 | 106 Camelot Dr       |  Intel i7-16GB 2060 512GB SSD  |
 66120457   | 110         | Nike Kyrie 4Ge              |    70 | 2275 Rainbow Road    |  Used, Us men size10           |
 98711112   | 112         | kitchen mixer and juice maker |   200 | 2479 Murry Street    | by itself 100' juice maker 50  |
 78810003   | 112         | Dining Table                |    95 | 1997 Duncan Avenue   | Good condition sell only table |
 78710065   | 127         | GIANT 770                   |   500 | 2275 Rainbow Road    |  unused Perfect condition      |
 19989002   | 118         | Hug Machine Book            |     8 | 61 Cubles Dr         |  Purchased for 8 selling for 5 |
 19910024   | 118         | $5 books                    |     5 | 2688 Pearl Street    | Paperbacks 5 Hardcovers 12     |
 99912765   | 125         | 2 dinosaurs                 |    45 | 2815 Saint Edwards Dr | Original 60 sell both40        |
 98846584   | 126         | Jay Chou Fantacy 2002       |   250 | 4661 Ferguson Street |  brand new, made in China.     |
(10 rows)
```

```
project1=> SELECT * FROM product WHERE using_review @@to_tsquery('english', 'quality');
```

Try: *SELECT * FROM product WHERE using_review @@to_tsquery('english', 'quality');*

---We add one text attribute "using_review" to the table "product" since it is a second-hand trading platform. Users could write reviews for using after they bought items from this platform. Moreover, to perform a full-text search on the "using_review" attribute, we can use the @@ operator along with the to_tsquery function. The query is aimed for searching reviews that contain the word "quality". It is important for platform managers to notice the details of talking about "quality", since second-hand items face more quality problems.

```
 product_id | category_id |      name      | price |      address       |          description
 |                                                  using_review

------------+-------------+----------------+-------+--------------------+------------------------------
-+-----------------------------------------------------------------------------------------------------
---------------------------------
 88910009   | 130         | Samsung 32 Inc |   109 | 106 Camelot Dr     | new 32 inches Samsung smart tv
 | It is a quite new television, and the screen is clear and bright. The quality of this product is good
 66120457   | 110         | Nike Kyrie 4Ge |    70 | 2275 Rainbow Road  |  Used, Us men size10
 | Although Kyrie is my favorite player in NBA, the quality of this shoes is poor. There are signs of da
mage to the upper and worn soles
 78810003   | 112         | Dining Table   |    95 | 1997 Duncan Avenue | Good condition sell only table
 | The quality of the table is quite good. My family could have dinner there, and the size is suitable f
or us
 19910024   | 118         | $5 books       |     5 | 2688 Pearl Street  | Paperbacks 5 Hardcovers 12
 | The book has beautiful hardcover, and articles are interesting. Moreover, the quality is better than
I thought
(4 rows)
```

2. Adding array attribute

```
project1=> ALTER TABLE delivery ADD COLUMN transac_price integer[];
ALTER TABLE
project1=> UPDATE delivery SET transac_price = ARRAY[5, 40, 29] WHERE track_number = '10762758';
UPDATE 1
project1=> UPDATE delivery SET transac_price = ARRAY[18, 50] WHERE track_number = '13225769';
UPDATE 1
project1=> UPDATE delivery SET transac_price = ARRAY[210, 19] WHERE track_number = '12536833';
UPDATE 1
project1=> UPDATE delivery SET transac_price = ARRAY[220] WHERE track_number = '15003271';
UPDATE 1
project1=> UPDATE delivery SET transac_price = ARRAY[70, 12, 4] WHERE track_number = '12546087';
UPDATE 1
project1=> UPDATE delivery SET transac_price = ARRAY[120, 39] WHERE track_number = '17552914';
```

```
project1=> select * from delivery;
 track_number |       address        | transac_price
--------------+----------------------+---------------
 10762758     | 2470 Fulton Street   | {5,40,29}
 13225769     | 4065 Murry Street    | {18,50}
 12536833     | 984 Greendale Avenue | {210,19}
 15003271     | 3468 Fraggle Drive   | {220}
 12546087     | 3196 Quilly Lane     | {70,12,4}
 17552914     | 1776 Nickel Road     | {120,39}
 16900325     | 1398 Ford Street     | {68}
 13229760     | 4939 Arrowood Drive  | {15,40}
 18302412     | 4542 Center Avenue   | {99,20,7}
 15882769     | 4555 Levy Court      | {100,17}
(10 rows)
```

We insert new rows of transaction price into the "delivery" table, with an integer array containing the values of each product's price in "transact_price" column. Because for

each track number, there could be more than one product delivered to the same address, we create an array attribute of price to see the number of prices for products that in the same package (same track number).

```
project1=> UPDATE delivery SET transac_price = array_append(transac_price, 39) WHERE track_number = '15003271';
UPDATE 1
```

Try: *UPDATE delivery SET transact_price = array_append(transact_price, 37) WHERE track_number = '13225769';*

---Although it seems unreliable, if the platform could support the function that users could purchase products and add them to the established track_number record within 2 hours after the initial purchase, we could append the price array and update the new attribute.

```
project1=> UPDATE delivery SET transac_price = array_remove(transac_price, 20) WHERE track_number = '18302412';
UPDATE 1
project1=> select * from delivery;
 track_number |       address        | transac_price
--------------+----------------------+---------------
 10762758     | 2470 Fulton Street   | {5,40,29}
 13225769     | 4065 Murry Street    | {18,50}
 12536833     | 984 Greendale Avenue | {210,19}
 12546087     | 3196 Quilly Lane     | {70,12,4}
 17552914     | 1776 Nickel Road     | {120,39}
 16900325     | 1398 Ford Street     | {68}
 13229760     | 4939 Arrowood Drive  | {15,40}
 15882769     | 4555 Levy Court      | {100,17}
 15003271     | 3468 Fraggle Drive   | {220,39}
 18302412     | 4542 Center Avenue   | {99,7}
(10 rows)
```

Try: *UPDATE delivery SET transact_price = array_remove(transact_price, 12) WHERE track_number = '12546087';*

---A more realistic design is that users would apply for refund when they don't want to buy second-hand items before delivering to their address (or set a restriction of refunding time). Under this condition, the price array for specific track number would change. We could use "array_remove" to adjust the transaction price of the array attribute.