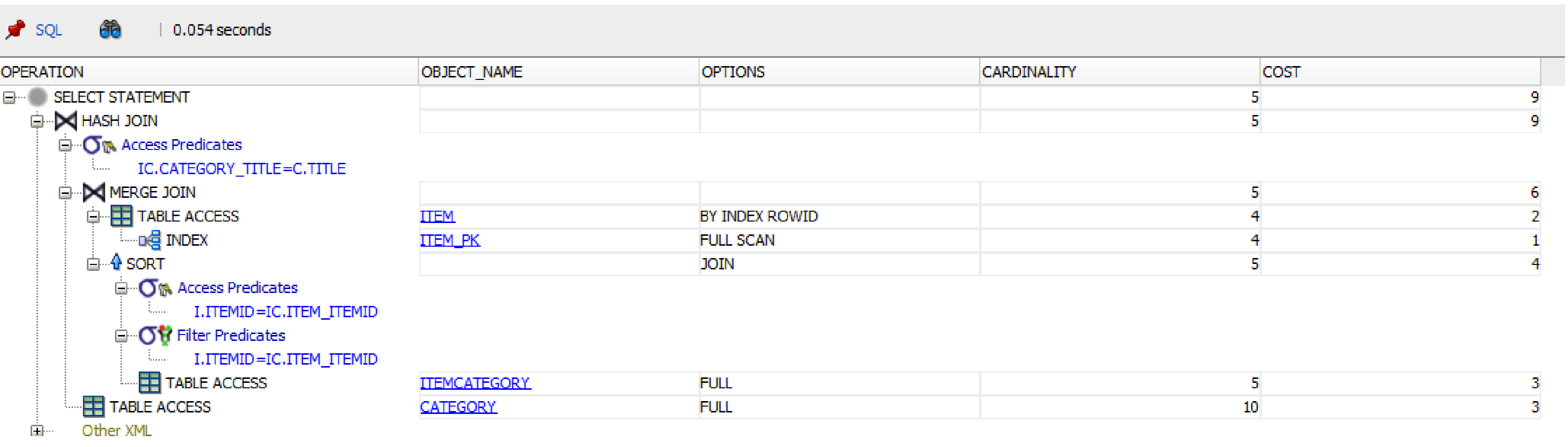
Assignment 4

Common Query 1

Index

The item category index with both item id and category title columns was used. An inner join was used to make use of the index. The significant changes are the decrease in cost using an index instead of the table.

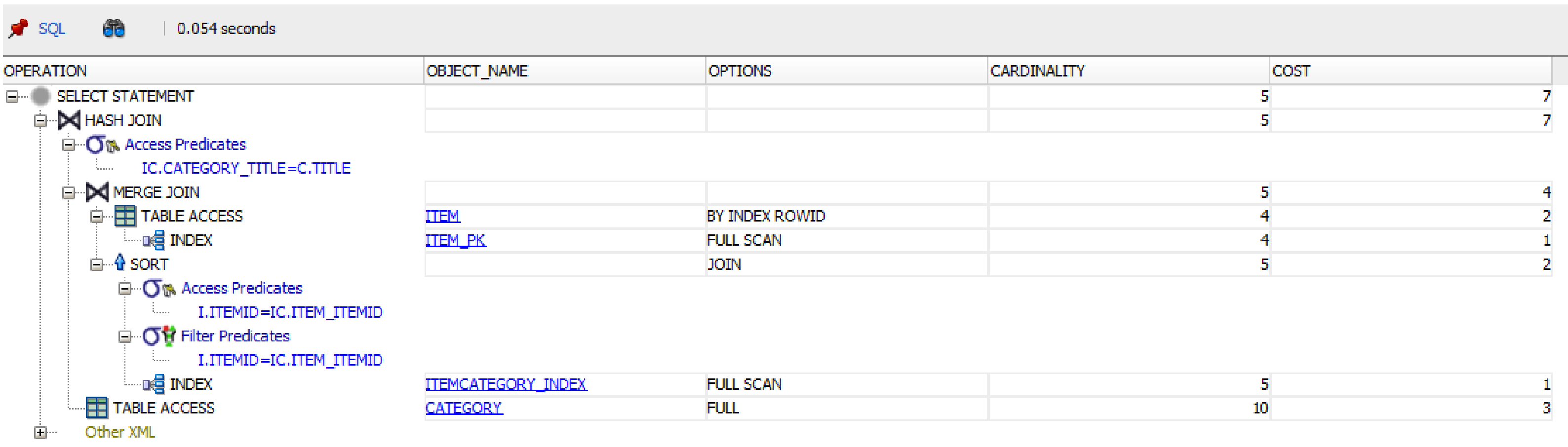
Before Index



```
SQL   0.054 seconds

OPERATION                               OBJECT_NAME         OPTIONS          CARDINALITY  COST
SELECT STATEMENT                                                                       5     9
  HASH JOIN                                                                            5     9
    Access Predicates
      IC.CATEGORY_TITLE=C.TITLE
    MERGE JOIN                                                                         5     6
      TABLE ACCESS                      ITEM                BY INDEX ROWID             4     2
        INDEX                           ITEM_PK             FULL SCAN                  4     1
      SORT                                                  JOIN                       5     4
        Access Predicates
          I.ITEMID=IC.ITEM_ITEMID
        Filter Predicates
          I.ITEMID=IC.ITEM_ITEMID
        TABLE ACCESS                    ITEMCATEGORY        FULL                       5     3
    TABLE ACCESS                        CATEGORY            FULL                      10     3
  Other XML
```

After Index



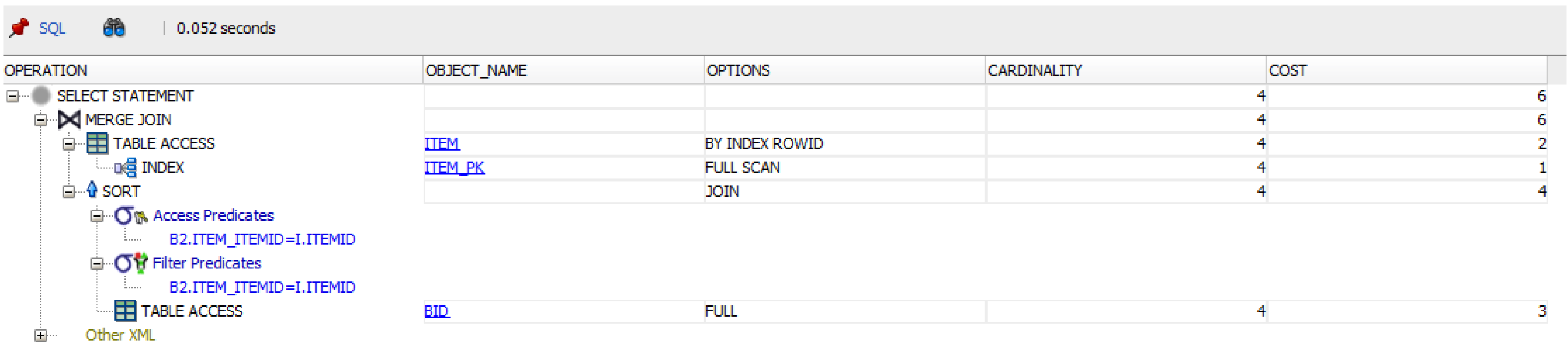
```
SQL   0.054 seconds

OPERATION                               OBJECT_NAME         OPTIONS          CARDINALITY  COST
SELECT STATEMENT                                                                       5     7
  HASH JOIN                                                                            5     7
    Access Predicates
      IC.CATEGORY_TITLE=C.TITLE
    MERGE JOIN                                                                         5     4
      TABLE ACCESS                      ITEM                BY INDEX ROWID             4     2
        INDEX                           ITEM_PK             FULL SCAN                  4     1
      SORT                                                  JOIN                       5     2
        Access Predicates
          I.ITEMID=IC.ITEM_ITEMID
        Filter Predicates
          I.ITEMID=IC.ITEM_ITEMID
        INDEX                           ITEMCATEGORY_INDEX  FULL SCAN                  5     1
    TABLE ACCESS                        CATEGORY            FULL                      10     3
  Other XML
```

Common Query 2

The index for bid and the amount column would help with the 2nd index. This allows to find the certain number of the bid amount without going through a huge range of numbers. However, the index is not used because it does not need to check the how much in an amount. Adding a 'WHERE' condition makes use of the range scan for the amount.

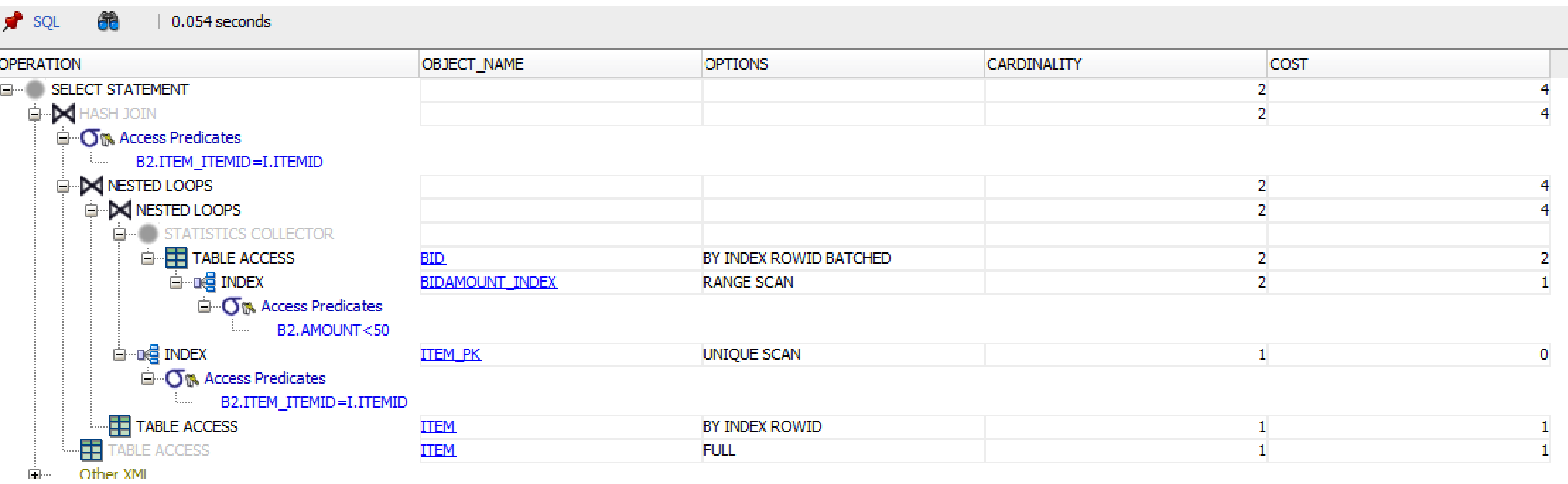
Before Index



SQL | 0.052 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4	6
MERGE JOIN			4	6
TABLE ACCESS	ITEM	BY INDEX ROWID	4	2
INDEX	ITEM_PK	FULL SCAN	4	1
SORT		JOIN	4	4
Access Predicates				
B2.ITEM_ITEMID=I.ITEMID				
Filter Predicates				
B2.ITEM_ITEMID=I.ITEMID				
TABLE ACCESS	BID	FULL	4	3
Other XML				

After Index



SQL | 0.054 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	4
HASH JOIN			2	4
Access Predicates				
B2.ITEM_ITEMID=I.ITEMID				
NESTED LOOPS			2	4
NESTED LOOPS			2	4
STATISTICS COLLECTOR				
TABLE ACCESS	BID	BY INDEX ROWID BATCHED	2	2
INDEX	BIDAMOUNT_INDEX	RANGE SCAN	2	1
Access Predicates				
B2.AMOUNT<50				
INDEX	ITEM_PK	UNIQUE SCAN	1	0
Access Predicates				
B2.ITEM_ITEMID=I.ITEMID				
TABLE ACCESS	ITEM	BY INDEX ROWID	1	1
TABLE ACCESS	ITEM	FULL	1	1
Other XML				

Partitioning Strategies

Listing partition on the category table and subcategory column is recommended for partitioning. It is helpful for specifying the specific category an item is under because there are many types of items in the subcategory. The performance may improve through partition pruning because only one main category of items will be accessed instead of all items in the database.

Targeted Advertising for Users

MongoDB database structure can be used for identifying target advertising and recommendations for users based on their past searches and view history. Keywords from recent searches can be used to query recommended items for users. If a word partially matches a description of an item, it queries the items and show it on the website for the user. Moreover, items that have been bid by the user will query other items in the same category for the 'users also bought' section.

For the view history, descriptions and the category of the items would be recorded for each user whenever an item is viewed. A section that would query similar keywords to the description and same category will show items that fit the user's browsing history on the items.

MongoDB Query

Find a description of an item based on the last keyword searched and category the item is in.

```
db.item.find(

  {description: RegExp('new', i),'category.title': 'Toys','category.sub_title': 'Diecast'}

)
```