

PES UNIVERSITY
ELECTIVE 1: DATABASE TECHNOLOGIES (UE18CS315)
ASSIGNMENT 3

NAME: SHAAZIN SHEIKH SHUKOOR

SRN: PES1201801754

SEMESTER: 5

SECTION: J

Write a simple Map-Reduce program

Map-Reduce is a programming framework that allows us to perform distributed and parallel processing on large data sets in a distributed environment. Map-Reduce consists of two distinct tasks – Map and Reduce.

- As the name Map-Reduce suggests, the reducer phase takes place after the mapper phase has been completed.
- So, the first is the map job, where a block of data is read and processed to produce key-value pairs as intermediate outputs.
- The output of a Mapper or map job (key-value pairs) is input to the Reducer.
- The reducer receives the key-value pair from multiple map jobs.
- Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.

Problem statement:

- i. To print the count of number of books in the library which were published before 2000.
- ii. To print the name of the latest publisher.

Input: A library dataset of csv format containing ISBN, Book-Title, Book-Author, Year-Of Publication, Publisher, Image-URL-S, Image-URL-M, and Image-URL-L as its fields.

Output: 2 lines; 1st line is the count of the number of books published before 2000 and 2nd line is the name of the latest publisher.

Mapper code:

```
#!/usr/bin/python3.8

import sys

first=1

for line1 in sys.stdin:
    for line2 in line1.splitlines():
        list1 = [x.strip(' ') for x in line2.split(';')]
        if(first==1):
            idx1=list1.index('Year-Of-Publication')
            idx2=list1.index('Publisher')
            first=0
        else:
            print(list1[idx1], '\t', list1[idx2])
```

Reducer code:

```
#!/usr/bin/python3.8

import sys

count=0

for line1 in sys.stdin:
    line1=line1.strip().split('\t',1)
    if(int(line1[0]) < 2000):
        count=count+1
    name=line1[1].strip()
print(count)
print(name)
```

Running the map-reduce program in hadoop:

The commands for running the mapper and reducer and to print the contents of the output file are include in the bash script (start.sh).

(The bash script is ran using the command ./start.sh)

```
#!/bin/sh
bin/hdfs namenode -format
sbin/start-dfs.sh
sbin/start-yarn.sh
jps
bin/hdfs dfs -rm -r output
bin/hdfs dfs -rm -r input
bin/hdfs dfs -mkdir /user
bin/hdfs dfs -mkdir /user/shaaz
bin/hdfs dfs -mkdir input
bin/hdfs dfs -ls
bin/hdfs dfs -put '/home/shaaz/dbt3/input.csv' input
bin/hadoop jar '/home/shaaz/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.2.1.jar' -mapper '/home/shaaz/dbt3/mapper.py'
-reducer '/home/shaaz/dbt3/reducer.py' -input input/ -output output
bin/hdfs dfs -cat output/part-00000
```

```
Activities Terminal Sep 30 20:53
shaaz@PES1201801754:~/hadoop$ ./start.sh
2020-09-30 20:46:09,041 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = shaaz/127.0.1.1
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 3.2.1
STARTUP_MSG: classpath = /home/shaaz/hadoop/etc/hadoop:/home/shaaz/hadoop/share/hadoop/common/lib/hadoop-annotations-3.2.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-simplekdc-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-logging-1.1.3.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jsr305-3.0.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jackson-annotations-2.9.8.jar:/home/shaaz/hadoop/share/hadoop/common/lib/snappy-java-1.0.5.jar:/home/shaaz/hadoop/share/hadoop/common/lib/dnsjava-2.1.7.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-core-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/curator-recipes-2.13.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jackson-databind-2.9.8.jar:/home/shaaz/hadoop/share/hadoop/common/lib/httpcore-4.4.10.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-lang3-3.7.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jcip-annotations-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jsch-0.1.54.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-server-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-identity-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/httpclient-4.5.6.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-configuration2-2.1.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/stax2-api-3.1.4.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jsr311-api-1.1.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jackson-core-2.9.8.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jetty-webapp-9.3.24.v20180605.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-compress-1.10.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-admin-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/failureaccess-1.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/asn-5.0.4.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerby-config-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerby-uttl-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/slf4j-api-1.7.25.jar:/home/shaaz/hadoop/share/hadoop/common/lib/json-smart-2.3.jar:/home/shaaz/hadoop/share/hadoop/common/lib/nimbus-jose-jwt-4.41.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/paranamer-2.3.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-codec-1.11.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jackson-xc-1.9.13.jar:/home/shaaz/hadoop/share/hadoop/common/lib/metrics-core-3.2.4.jar:/home/shaaz/hadoop/share/hadoop/common/lib/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar:/home/shaaz/hadoop/share/hadoop/common/lib/curator-framework-2.13.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/hadoop-auth-3.2.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-net-3.6.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jersey-servlet-1.19.jar:/home/shaaz/hadoop/share/hadoop/common/lib/audience-annotations-0.5.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jboss-j2ee-1.1.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/guava-27.0-jre.jar:/home/shaaz/hadoop/share/hadoop/common/lib/htrace-core-4.4.1.0-incubating.jar:/home/shaaz/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/home/shaaz/hadoop/share/hadoop/common/lib/token-provider-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/avro-1.7.7.jar:/home/shaaz/hadoop/share/hadoop/common/lib/jersey-json-1.19.jar:/home/shaaz/hadoop/share/hadoop/common/lib/log4j-1.2.17.jar:/home/shaaz/hadoop/share/hadoop/common/lib/kerb-common-1.0.1.jar:/home/shaaz/hadoop/share/hadoop/common/lib/curator-client-2.13.0.jar:/home/shaaz/hadoop/share/hadoop/common/lib/woodstox-core-5.0.3.jar:/home/shaaz/hadoop/share/hadoop
```

```
Activities Terminal Sep 30 20:53
/*****
SHUTDOWN_MSG: Shutting down NameNode at shaaz/127.0.1.1
*****/
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [shaaz]
Starting resource manager
Starting nodemanagers
3235 NodeManager
2851 SecondaryNameNode
2518 NameNode
2647 DataNode
3432 Jps
3085 ResourceManager
rm: 'output': No such file or directory
rm: 'input': No such file or directory
Found 1 items
drwxr-xr-x - shaaz supergroup 0 2020-09-30 20:49 input
2020-09-30 20:49:24,751 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2020-09-30 20:49:36,908 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2020-09-30 20:49:37,389 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2020-09-30 20:49:37,389 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2020-09-30 20:49:37,552 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2020-09-30 20:49:38,538 INFO mapred.FileInputFormat: Total input files to process : 1
2020-09-30 20:49:38,626 INFO mapreduce.JobSubmitter: number of splits:1
2020-09-30 20:49:39,381 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1243119240_0001
2020-09-30 20:49:39,386 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-09-30 20:49:39,837 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2020-09-30 20:49:39,866 INFO mapreduce.Job: Running job: job_local1243119240_0001
2020-09-30 20:49:39,868 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2020-09-30 20:49:39,875 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2020-09-30 20:49:39,955 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2020-09-30 20:49:39,956 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup_temporary folders under output directory:false, ign
```

```
Activities Terminal Sep 30 20:53
Map output records=23
Map output bytes=553
Map output materialized bytes=605
Input split bytes=100
Combine input records=0
Combine output records=0
Reduce input groups=12
Reduce shuffle bytes=605
Reduce input records=23
Reduce output records=2
Spilled Records=46
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=113
Total committed heap usage (bytes)=245194752
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=6887
File Output Format Counters
Bytes Written=27
2020-09-30 20:49:55,019 INFO streaming.StreamJob: Output directory: output
2020-09-30 20:50:08,219 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
15
House of Anansi Press
shaaz@PES1201801754:~/hadoop$
```

The cat command at the end prints the output of the map-reduce program in the terminal (output of the reducer):

```
15
House of Anansi Press
```

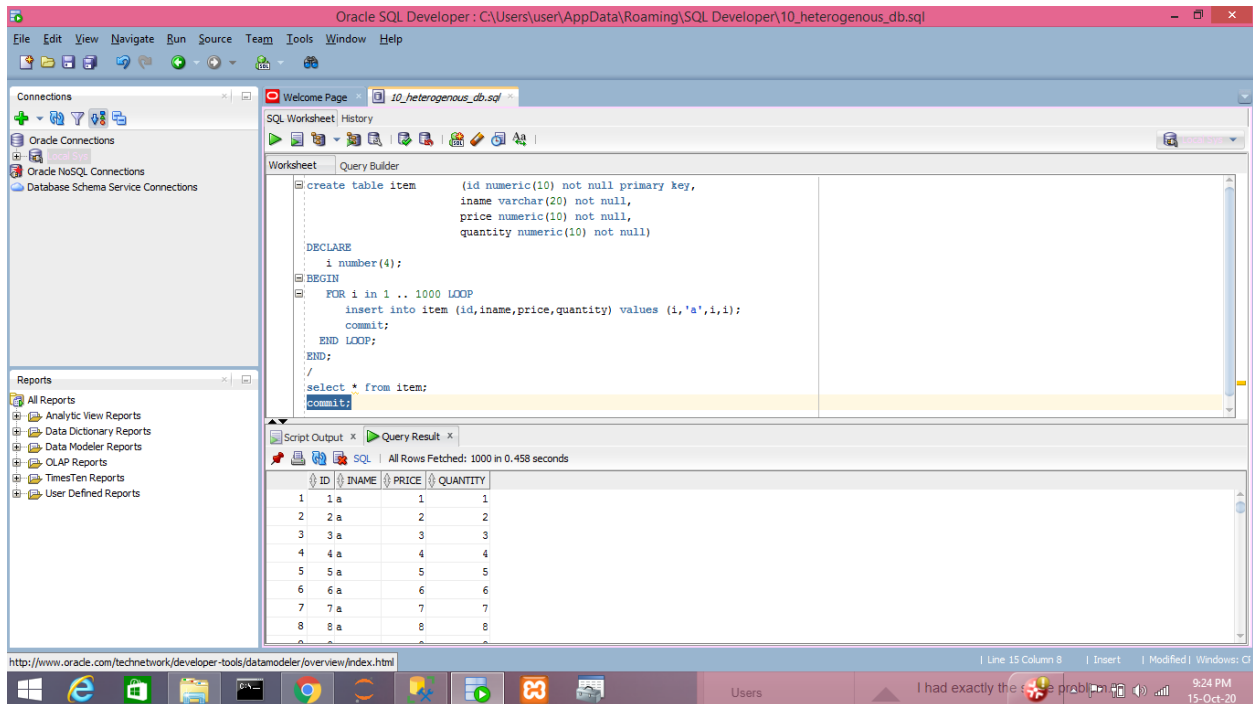
The intermediate key-value pair of the mapper (sorts it internally):

```
shaaz@PES1201801754:~/dbt3$cat output_mapper
1988 Emblem Editions
1991 HarperPerennial
1991 Putnam Pub Group
1993 Audioworks
1994 Plume
1994 River City Pub
1996 Random House
1998 Brilliance Audio - Trade
1998 Health Communications
1999 Dell
1999 Farrar Straus Giroux
1999 Kensington Publishing Corp.
1999 Mira Books
1999 Three Rivers Press
1999 W. W. Norton &
2000 Berkley Publishing Group
2000 Citadel Press
2001 HarperFlamingo Canada
2001 Ryland Peters &
2002 Oxford University Press
2002 Scribner
2003 Cypress House
2004 House of Anansi Press
shaaz@PES1201801754:~/dbt3$
```

Demonstrate a heterogeneous distributed query.

A distributed database is a set of interconnected databases that is distributed over the computer network or internet. A Distributed Database Management System (DDBMS) manages the distributed database and provides mechanisms so as to make the databases transparent to the users.

Item table located in oracle:



The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Worksheet' tab. The script creates a table named 'item' with columns 'id', 'iname', 'price', and 'quantity'. It then declares a variable 'i' of type 'number(4)', starts a loop from 1 to 1000, and inserts data into the 'item' table. Finally, it selects all data from the 'item' table and commits the transaction.

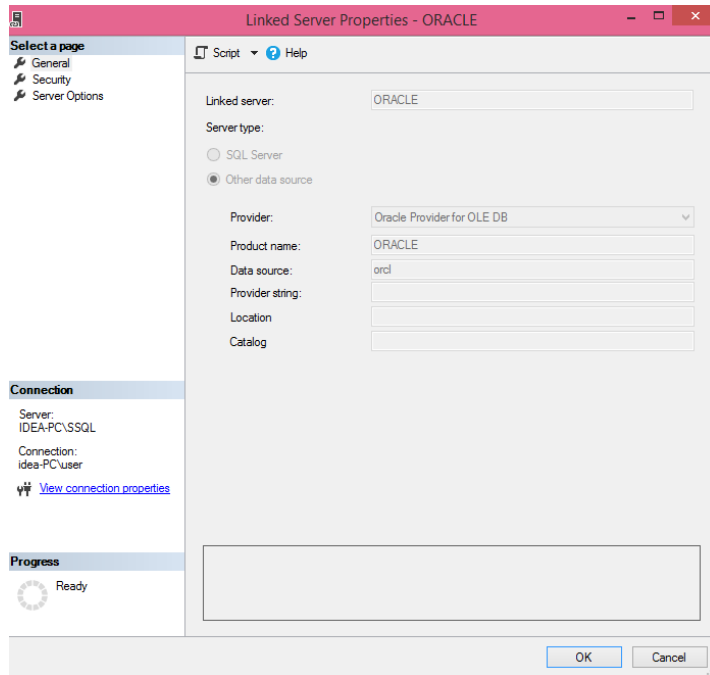
```
create table item (id numeric(10) not null primary key, iname varchar(20) not null, price numeric(10) not null, quantity numeric(10) not null)

DECLARE
  i number(4);
BEGIN
  FOR i in 1 .. 1000 LOOP
    insert into item (id,iname,price,quantity) values (i,'a',i,i);
    commit;
  END LOOP;
END;
/
select * from item;
commit;
```

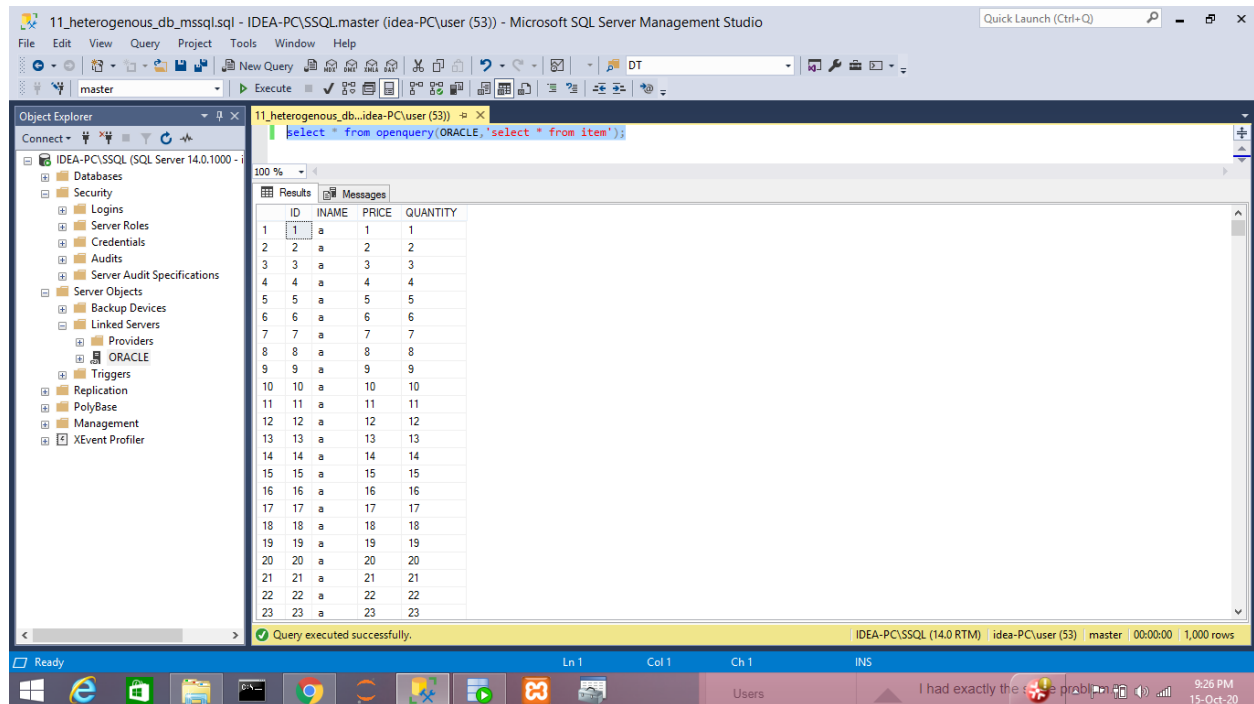
The 'Query Result' tab shows the output of the query, displaying 1000 rows of data. The columns are 'ID', 'INAME', 'PRICE', and 'QUANTITY'. The data shows a sequence of values from 1 to 1000, with 'INAME' always being 'a'.

ID	INAME	PRICE	QUANTITY
1	a	1	1
2	a	2	2
3	a	3	3
4	a	4	4
5	a	5	5
6	a	6	6
7	a	7	7
8	a	8	8

A linked server is created in Microsoft SQL Server Management Studio by using the provider 'Oracle Provider for OLE DB'



The contents of table 'item' can be viewed using openquery:



Queries using the item table:

```
1_heterogenous_db...idea-PC\user (53)) - X
commit transaction T6

--query
SET STATISTICS IO ON
SET STATISTICS TIME ON

SELECT i.id,i.iname
FROM (select * from openquery(ORACLE,'select * from item')) i
WHERE i.id IN(SELECT item_id
FROM ORDER_DETAIL
WHERE quantity>700
);

SELECT e.fname, SUM(od.quantity*i.price) as sum
FROM EMPLOYEE e, ORDER_DETAIL od, ORDERS o, (select * from openquery(ORACLE,'select * from item')) i
WHERE e.id=o.emp_id AND o.id=od.order_id AND od.item_id=i.id
GROUP BY e.fname
ORDER BY SUM(od.quantity*i.price) DESC

SELECT top 3 i.iname,SUM(od.quantity*i.price) as sum
FROM ORDER_DETAIL AS od LEFT OUTER JOIN (select * from openquery(ORACLE,'select * from item')) AS i ON od.item_id=i.id
GROUP BY i.iname
ORDER BY SUM(od.quantity*i.price) ASC

SET STATISTICS IO OFF
SET STATISTICS TIME OFF
```

Output of the above queries:

The screenshot displays the Microsoft SQL Server Management Studio interface. The query window shows three queries being executed sequentially. The first query returns a list of item IDs and names. The second query returns a list of employee names and their total order sums. The third query returns a list of item names and their total order sums.

Query 1 Results:

id	iname
701	a
702	a
703	a
704	a
705	a
706	a
707	a
708	a
709	a

Query 2 Results:

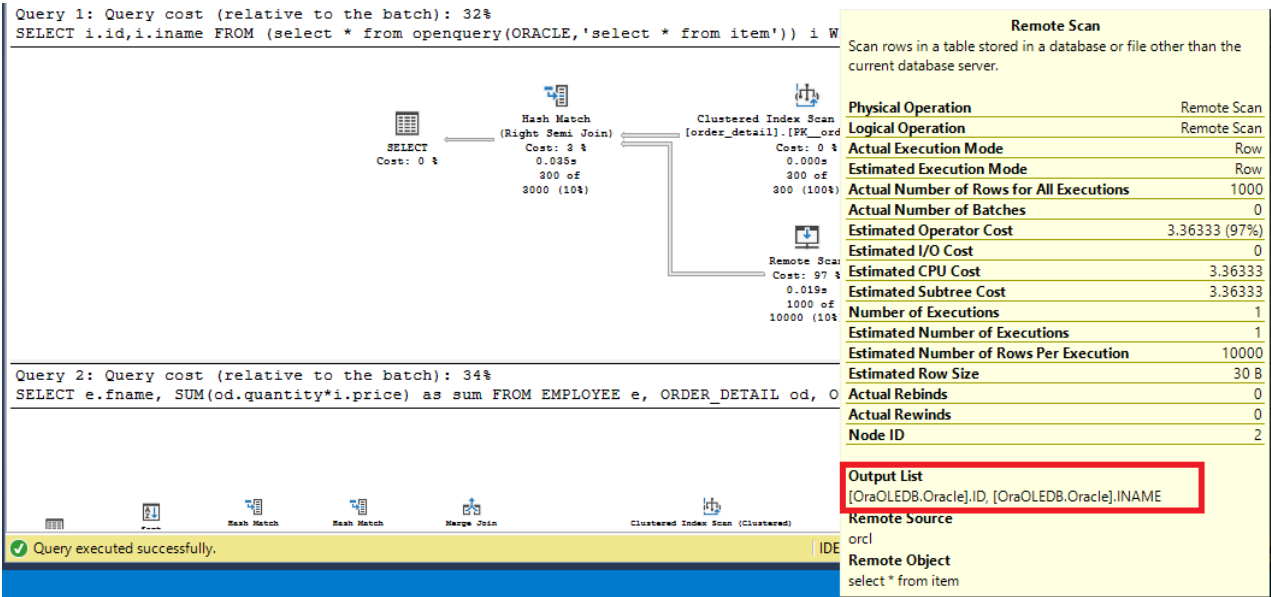
fname	sum
1000ABC	1000000
999ABC	998001
998ABC	996004
997ABC	994009
996ABC	992016
995ABC	990025
994ABC	988036
993ABC	986049
992ABC	984064

Query 3 Results:

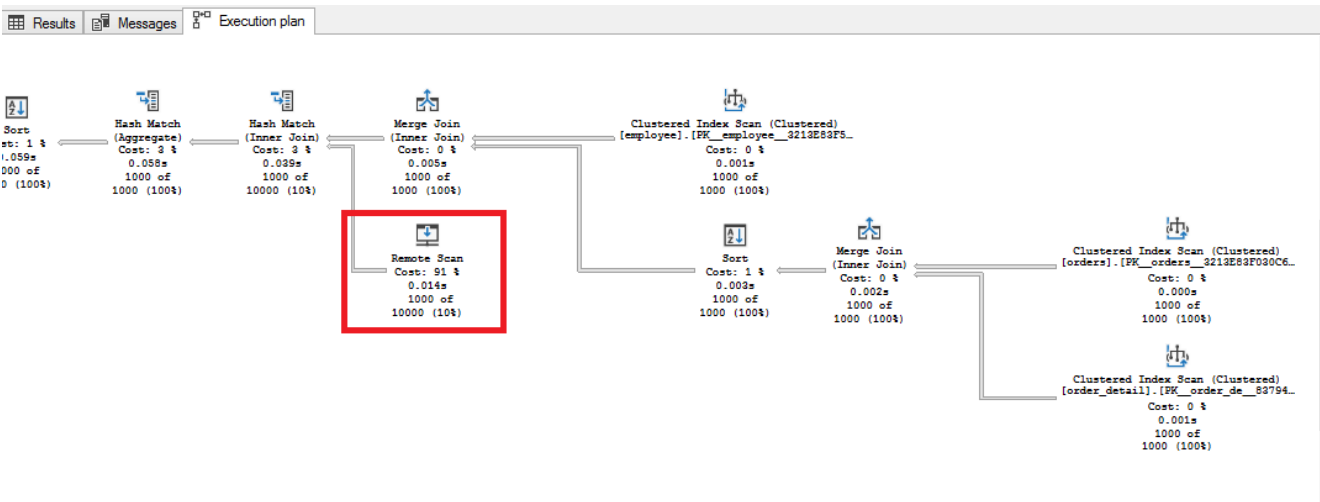
iname	sum
a	333833500

The status bar at the bottom indicates that the query was executed successfully, returning 1,301 rows.

We see that in the first query, only those rows are retrieved from the 'item' table that needs to be projected in the output:



One of the disadvantages of distributed database is that the query optimizer is not so efficient. Below is an example showing the order of join of tables. We see that the 'item' table is joined at the end:



Use of transaction and commit:

A record is inserted using 'insert openquery'. We see that the changes get reflection using 'commit'.

The screenshot shows a SQL Developer window with the following SQL script:

```
SET XACT_ABORT ON

begin transaction T1
INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
values (1001,'b',1001,1001);
commit transaction T1
select * from openquery(ORACLE,'select * from item');
```

The Results tab displays a table with 1,001 rows. The last row, ID 1001, has INAME 'b', PRICE 1001, and QUANTITY 1001. The status bar at the bottom indicates 'Query executed successfully.' and '1,001 rows'.

ID	INAME	PRICE	QUANTITY
988	a	988	988
989	a	989	989
990	a	990	990
991	a	991	991
992	a	992	992
993	a	993	993
994	a	994	994
995	a	995	995
996	a	996	996
997	a	997	997
998	a	998	998
999	a	999	999
1000	a	1000	1000
1001	b	1001	1001

Use of transaction and rollback:

A record is deleted using 'delete openquery'. We see that the change of the transaction does not get reflected using 'rollback'.

The screenshot shows a SQL Developer window with the following SQL script:

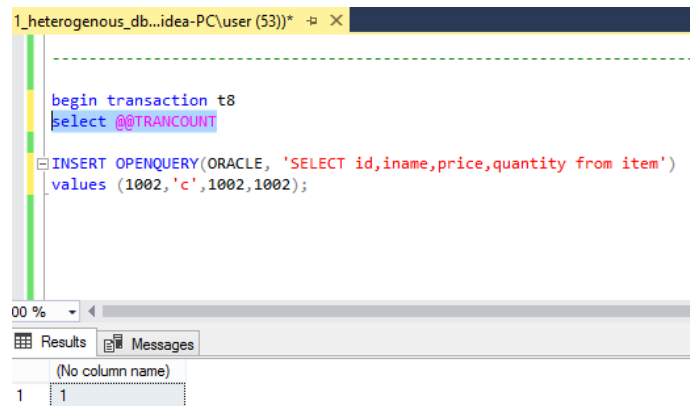
```
begin transaction T7
DELETE OPENQUERY(ORACLE, 'SELECT * from item where id=1001')
rollback transaction T7
select * from openquery(ORACLE,'select * from item');
```

The Results tab displays a table with 1,001 rows. The last row, ID 1001, has INAME 'b', PRICE 1001, and QUANTITY 1001. The status bar at the bottom indicates 'Query executed successfully.' and '1,001 rows'.

ID	INAME	PRICE	QUANTITY
988	a	988	988
989	a	989	989
990	a	990	990
991	a	991	991
992	a	992	992
993	a	993	993
994	a	994	994
995	a	995	995
996	a	996	996
997	a	997	997
998	a	998	998
999	a	999	999
1000	a	1000	1000
1001	b	1001	1001

Nested transactions:

Begin the outer transaction 't8':



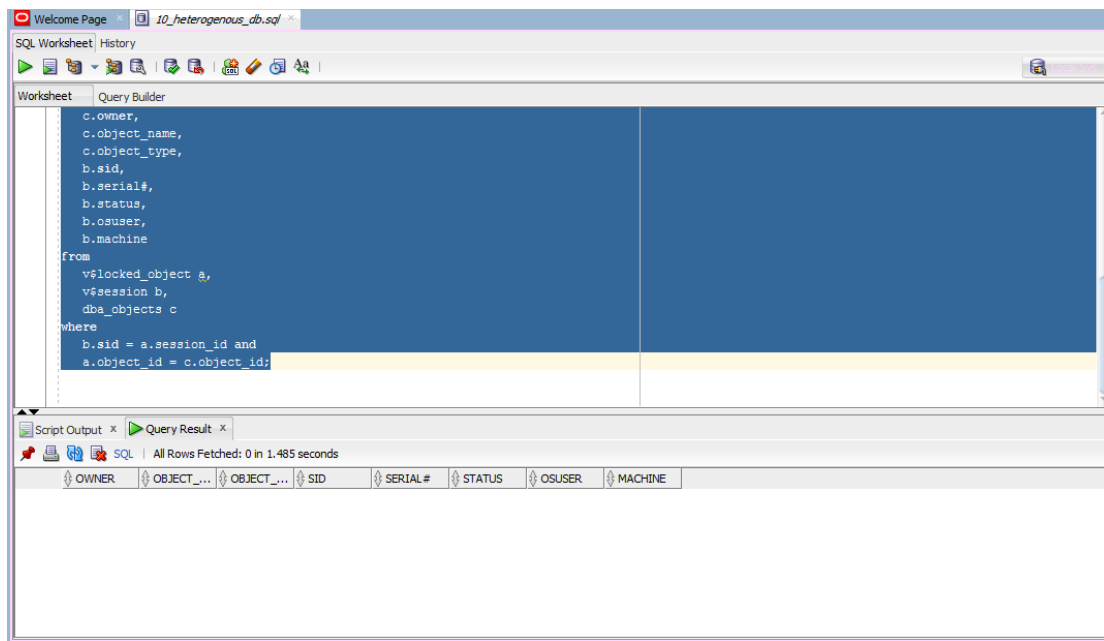
The screenshot shows a SQL Developer window with a query editor. The query is as follows:

```
begin transaction t8
select @@TRANSCOUNT
INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
values (1002, 'c', 1002, 1002);
```

Below the query editor, the 'Results' tab is active, showing a single row with the value '1' in the first column.

(No column name)
1

Currently there are no locks in oracle:



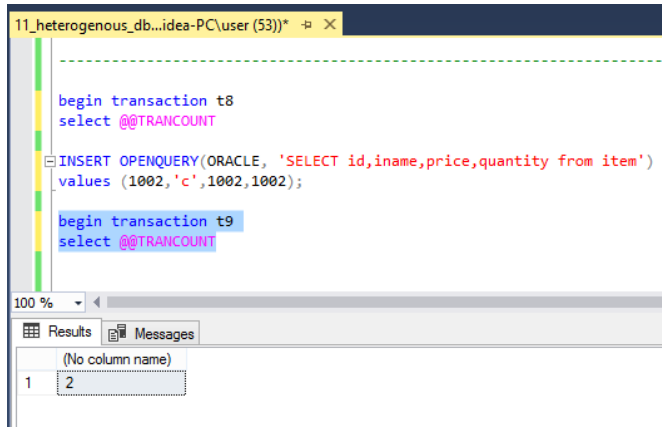
The screenshot shows a SQL Developer window with a query editor. The query is as follows:

```
c.owner,
c.object_name,
c.object_type,
b.sid,
b.serial#,
b.status,
b.osuser,
b.machine
from
v$locked_object a,
v$session b,
dba_objects c
where
b.sid = a.session_id and
a.object_id = c.object_id;
```

Below the query editor, the 'Query Result' tab is active, showing a table with the following columns: OWNER, OBJECT_NAME, OBJECT_TYPE, SID, SERIAL#, STATUS, OSUSER, MACHINE. The table is currently empty.

OWNER	OBJECT_NAME	OBJECT_TYPE	SID	SERIAL#	STATUS	OSUSER	MACHINE
-------	-------------	-------------	-----	---------	--------	--------	---------

Begin the inner transaction 't9':



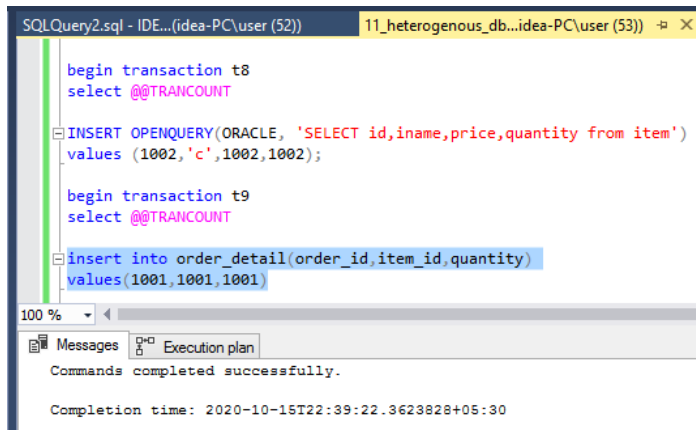
```
begin transaction t8
select @@TRANSCOUNT

INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
values (1002,'c',1002,1002);

begin transaction t9
select @@TRANSCOUNT
```

Results

(No column name)
1 2



```
begin transaction t8
select @@TRANSCOUNT

INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
values (1002,'c',1002,1002);

begin transaction t9
select @@TRANSCOUNT

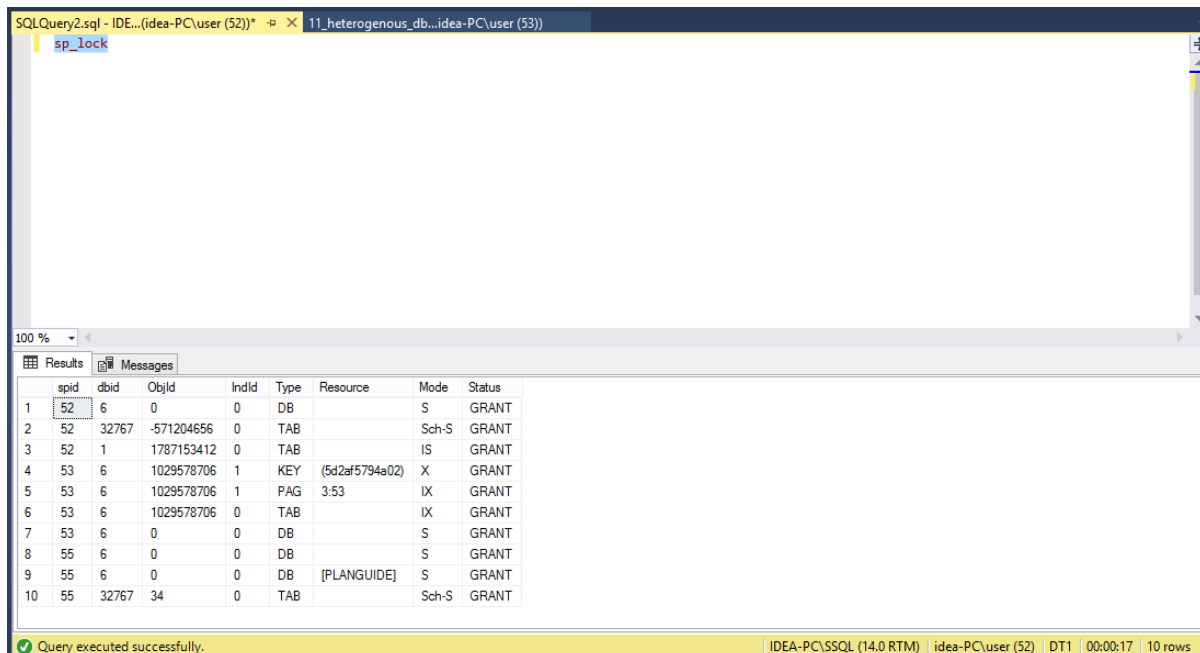
insert into order_detail(order_id,item_id,quantity)
values(1001,1001,1001)
```

Messages

Commands completed successfully.

Completion time: 2020-10-15T22:39:22.3623828+05:30

We see a lock on (53) in MS SQL:



```
sp_lock
```

Results

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	52	6	0	0	DB		S	GRANT
2	52	32767	-571204656	0	TAB		Sch-S	GRANT
3	52	1	1787153412	0	TAB		IS	GRANT
4	53	6	1029578706	1	KEY	(5d2af5794a02)	X	GRANT
5	53	6	1029578706	1	PAG	3:53	IX	GRANT
6	53	6	1029578706	0	TAB		IX	GRANT
7	53	6	0	0	DB		S	GRANT
8	55	6	0	0	DB		S	GRANT
9	55	6	0	0	DB	[PLANGUIDE]	S	GRANT
10	55	32767	34	0	TAB		Sch-S	GRANT

Query executed successfully.

IDEA-PC\SSQL (14.0 RTM) | idea-PC\user (52) | DT1 | 00:00:17 | 10 rows

We also see a lock in oracle:

The screenshot shows the SQL Developer interface with a query window titled '10_heterogenous_db.sql'. The query is as follows:

```
select
  c.owner,
  c.object_name,
  c.object_type,
  b.sid,
  b.serial#,
  b.status,
  b.osuser,
  b.machine
from
  v$locked_object a,
  v$session b,
  dba_objects c
where
  b.sid = a.session_id and
  a.object_id = c.object_id;
```

The query result is displayed in the 'Query Result' pane, showing one row of data:

	OWNER	OBJECT_NAME	OBJECT_TYPE	SID	SERIAL#	STATUS	OSUSER	MACHINE
1	SYSTEM	ITEM	TABLE	134	1105	INACTIVE	idea-PC\user	WORKGROUP\IDEA-PC

We commit the inner transaction 't9':

The screenshot shows the SQL Developer interface with a script window titled 'SQLQuery2.sql - IDE... (idea-PC\user (52))' and a connection to '11_heterogenous_db...idea-PC\user (53)'. The script contains the following SQL commands:

```
INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
values (1002,'c',1002,1002);

begin transaction t9
select @@TRANCOUNT

insert into order_detail(order_id,item_id,quantity)
values(1001,1001,1001)

commit transaction t9
```

The script is executed, and the 'Messages' pane at the bottom shows the following output:

```
Commands completed successfully.

Completion time: 2020-10-15T22:42:02.8828047+05:30
```

We still find the locks on (53) in MS SQL:

SQLQuery2.sql - IDE... (idea-PC\user (52))* 11_heterogenous_db...idea-PC\user (53))*

sp_lock

	spid	dbid	ObjId	IndId	Type	Resource	Mode	Status
1	52	6	0	0	DB		S	GRANT
2	52	32767	-571204656	0	TAB		Sch-S	GRANT
3	52	1	1787153412	0	TAB		IS	GRANT
4	53	6	1029578706	1	KEY	(5d2af5794a02)	X	GRANT
5	53	6	1029578706	1	PAG	3:53	IX	GRANT
6	53	6	0	0	DB		S	GRANT
7	53	6	1029578706	0	TAB		IX	GRANT
8	55	6	34	1	KEY	(5c7a9cea0b7e)	S	GRANT
9	55	32767	60	0	TAB		Sch-S	GRANT
10	55	6	34	1	KEY	(8d25e7d36bf1)	S	GRANT
11	55	6	0	0	DB		S	GRANT

Query executed successfully. IDEA-PC\SSQL (14.0 RTM) idea-PC\user (52) DT1 00:00:00 46 rows

We commit the outer transaction 't8':

11_heterogenous_db_mssql.sql - IDEA-PC\SSQL.DT1 (idea-PC\user (53))* - Microsoft SQL Server Management Studio

SQLQuery2.sql - IDE... (idea-PC\user (52))* 11_heterogenous_db...idea-PC\user (53))*

```
INSERT OPENQUERY(ORACLE, 'SELECT id,iname,price,quantity from item')
VALUES (1002,'c',1002,1002);

begin transaction t9
select @@TRANCOUNT

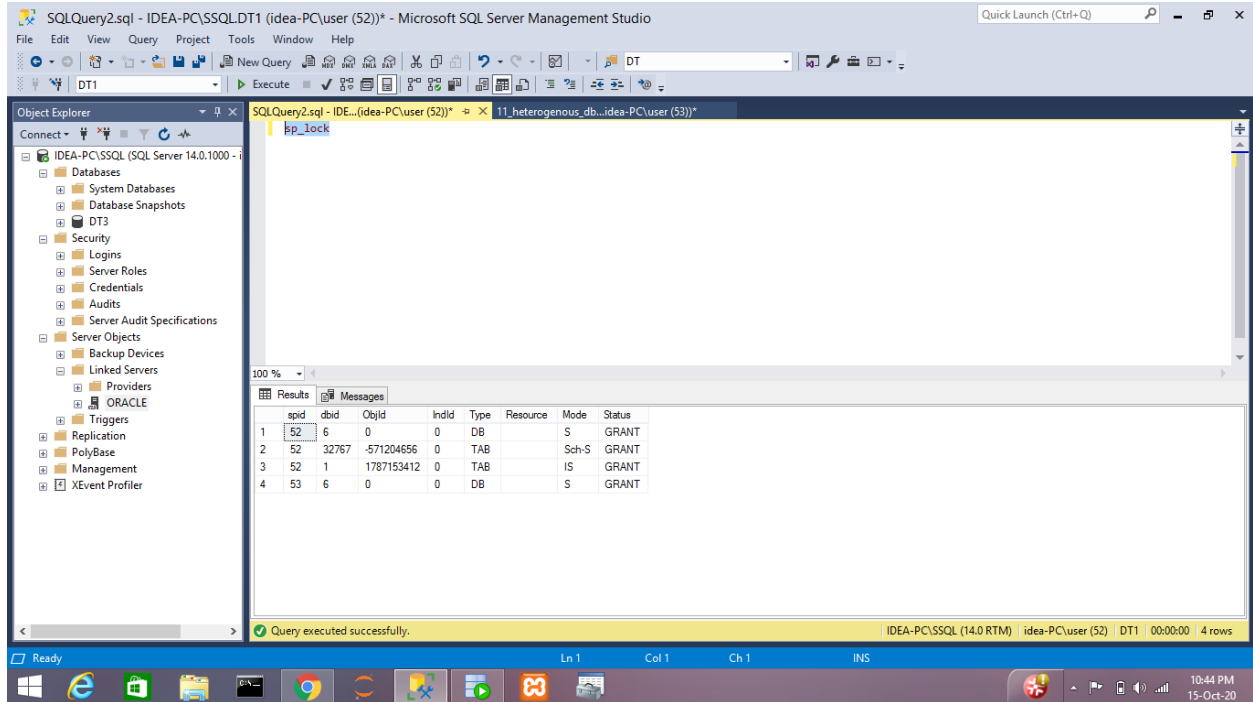
insert into order_detail(order_id,item_id,quantity)
values(1001,1001,1001)

commit transaction t9
commit transaction t8
select @@TRANCOUNT
```

	Results	Messages
1	(No column name)	
1	0	

Query executed successfully. IDEA-PC\SSQL (14.0 RTM) idea-PC\user (53) DT1 00:00:00 1 rows

The locks are removed:



The locks are also removed in oracle:

