Ahsanullah University of Science and Technology

Distributed Database System Lab

CSE 4126

Book Store Management

Submitted by:
Md. Nahidul Islam
15.01.04.127

Group Members:
Md. Rasheeq Zaman
15.01.04.117
Afrina Zahan Mithila
15.01.04.115

Submitted to:
Mohammad Imrul Jubair
Safrun Nesa Saira

October 11, 2018

1 Introduction

Book Store Management System uses managing the books of the store efficiently. It helps the shop owner to keep tracking all most every thing inside the shop. It maintains all most all incoming and outgoing finance from the shop. it helps to remember which books are available now in the shop. It reduces paper works and provide fast service to the customers.

1.1 Features

1. Information about any books price, quantity
2. Customers transaction report
3. Income report(Daily, Weekly, Monthly)
4. Ranking of The book selling
5. Rank of the most authors by selling
6. Customers Full Information and transaction history.

2 Platform

- Oracle 10g.
- PL/SQL

3 Fragmentation

We fragmented our database based on location.As we consider we have two book shop in two different cities, e.g. Dhaka and Khulna.We fragmented our database allocated them in two different site (Dhaka and Khulna).We also have have one host pc which have all the data of Khulna and Dhaka site.

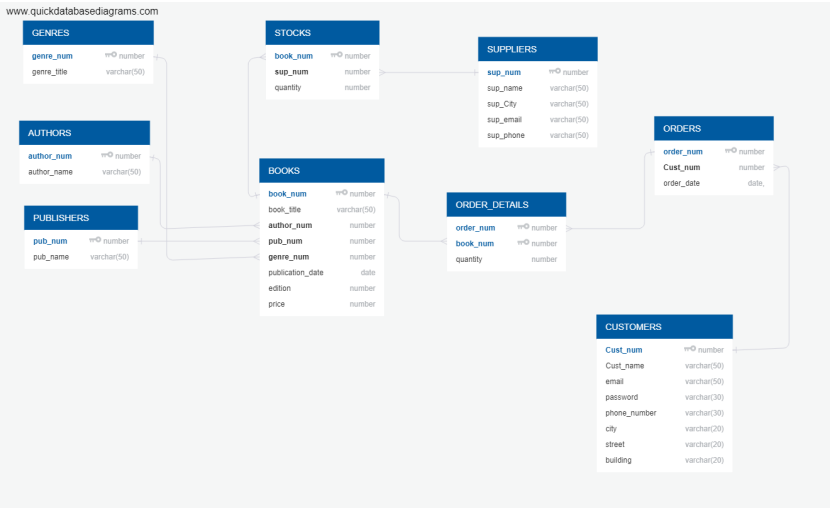


Figure 1: Database Schema.

4 Database Schema

5 Implemented Features

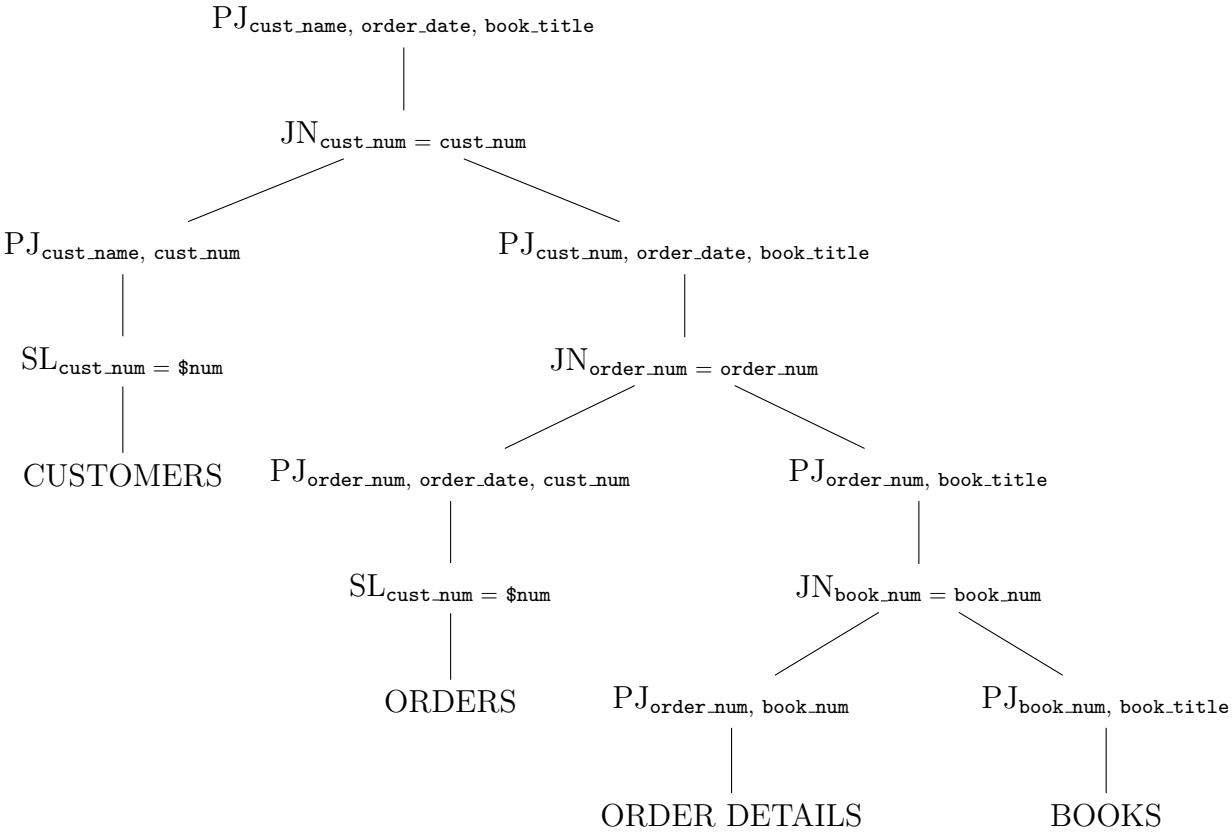
5.1 Transparency

We are able to simulate level - 3 distribution transparency. SQL code of our project using update steps are shown below:

```
update customers@site_link_dhaka
set cust_name = var_custName, email = var_email,
password = var_password, phone_number = var_phoneNumber,
city = var_city, building = var_building
where cust_num = var_custNum;
```

5.2 Operator Tree

Some of the operator trees for our project are simulated below:



5.3 Effect Of Update Query

An example of Effect Of Update Query As given Below:
Step 1:

```
Update Orders
set cust_num = 4 where
order_num = 3;
```

Table 1: Orders1

Cust_num	Order_num	Order_date
4	4	12-07-18

Step 2:

Table 2: Order2

Cust_num	Order_num	Order_date
2	3	11-07-18

```
Delete cust_name = 5 from Orders1;
```

Table 3: Orders1

Cust_num	Order_num	Order_date
4	4	12-07-18

Step 3:

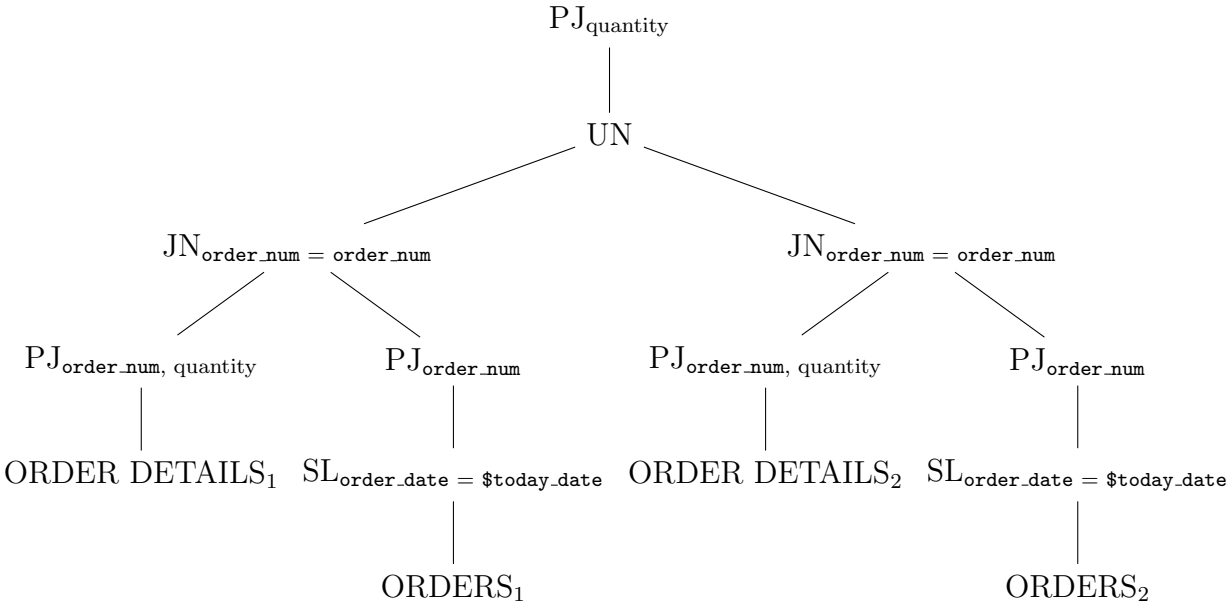
```
Insert into Orders2 values(5, 3, 11-07-18)
```

Table 4: Order2

Cust_num	Order_num	Order_date
2	3	11-07-18

5.4 Canonical expression

Some of the operator trees with canonical expression of our project are simulated below:


PJquantity
UN
JNorder_num = order_num
JNorder_num = order_num
PJorder_num, quantity
PJorder_num
PJorder_num, quantity
PJorder_num
ORDER DETAILS1
SLorder_date = $today_date
ORDER DETAILS2
SLorder_date = $today_date
ORDERS1
ORDERS2

5.5 Semi Join

We are able to simulate semi-join programs for join queries. SQL code of our project using update steps are shown below:

```
select cust_name, order_date, book_title
from (select order_num, order_date, cust_name
from orders@site_link_dhaka union
select order_num, order_date, cust_name
from orders@site_link_khulna) ord inner join
(select order_num, book_num from
order_details@site_link_dhaka union
select order_num, book_num from
order_details@site_link_khulna) ordd on
ord.order_num = ordd.order_num inner join
(select cust_num, cust_name from
customers@site_link_dhaka union
select cust_num, cust_name from
customers@site_link_khulna where
cust_num = var_custNum) cus on
cus.cust_num = ord.cust_num inner join
(select book_num, book_title from
books@site_link_dhaka union
select book_num, book_title from
books@site_link_khulna) bo on
bo.book_num = ordd.book_num;
```

5.6 Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur.Some Triggers from our projects given below:

```
set serveroutput on;
create or replace trigger order_trigger
after insert or update or delete on ORDERS
for each row
begin
dbms_output.put_line('Orders have been Changed');
end;
```

/

Figure 2: Result After using the above trigger

5.7 Estimating profiles of results of algebraic operations

5.7.1 Selection

$$card(order\_details) = 22; val(A[R] = 9; \rho = \frac{1}{val(A[R]} = \frac{1}{9}$$

```
Select * from order_details where order_num=10;
```

$$card(S) = \rho \times card(order\_details); = \frac{1}{9} \times 22 = 2.44 \approx 2$$

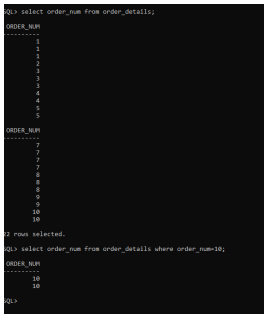


Figure 3: Estimating profiles with algebraic operations

5.7.2 Cartesian Product

$$card(orders) = 12; card(order\_details) = 22;$$

$$card(ordersCPPrder\_details) = card(orders) \times card(order\_details) = 12 \times 22 = 264$$

```
Select * from orders,order_details;
```

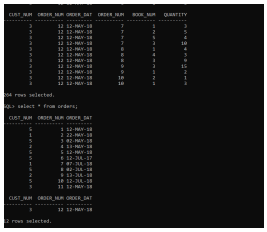


Figure 4: Estimating profiles with algebraic operations

5.8 Proof Of Algebraic Operation

5.8.1 proof of Commutative Law

Orders JN Customers = Customers JN Orders

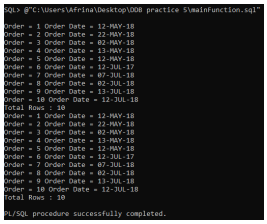


Figure 5: Commutative Proof

6 Future Plan

1. Employee Payroll
2. Printing receipt
3. Discount System by studying Customer History
4. implementing KNN Classifier
5. Supplying Cost

References

[1] Mcgraw Hill Osborne. [*Oracle Database 10G - The Complete Reference*].