Views

1) Customer History

With this view here we can see how many books customers have bought in their time with the store. This could be useful if the store comes up with a reward system for people who frequently buy books there.

Relational Algebra:

$PURCHASED\_BOOK\ BOOK \bowtie_{book\_ID = book\_ID} BUYS$

$COUNT\ customer\_ID\ \mathcal{F}_{COUNT\ book\_ID}(PURCHASED\_BOOK)$

$RESULT\ \pi_{customer\_ID,\ Count\_book\_ID}(COUNT)$

SQL Statements:

```
CREATE VIEW Customer_History
AS      SELECT A.customer_ID, count(A.book_ID)
        FROM BOOK B, BUYS A
        WHERE B.book_ID = A.book_ID
        GROUP BY A.customer_ID;
```

2) Total Books

This view allows us to see how many books are in each individual store. This is useful since it could tall managers when they need to order more books for that individual store.

Relational Algebra:

$AVAILABLE\_BOOK\ BOOK \bowtie_{book\_ID = book\_ID} AVAILABLE\_IN$

$COUNT\ store\_ID\ \mathcal{F}_{COUNT\ book\_ID}(AVAILABLE\_BOOK)$

$RESULT\ \pi_{store\_ID,\ Count\_book\_ID}(COUNT)$

SQL Statements:

```
CREATE VIEW Total_Books
AS      SELECT A.store_ID, count(A.book_ID)
        FROM BOOK B, AVAILABLE_IN A
        WHERE B.book_ID = A.book_ID
        GROUP BY A.store_ID;
```