

Queries

1. Find the titles of all books by Pratchett that cost less than \$10

Relational algebra expression:

```
AUTHOR_PRATCHETT  $\sigma_{\text{Lname} = \text{'Pratchett'}}$  (AUTHOR)
BOOKS_BY_PRATCHETT  $\bowtie_{\text{authorID} = \text{authorID}}$  BOOK
BOOKS_UNDER_10  $\sigma_{\text{price} < 10}$  (BOOKS_BY_PRATCHETT)
RESULT  $\pi_{\text{title}}$  (BOOKS_UNDER_10)
```

SQL code:

```
SELECT B.title
FROM Book B, Author A
WHERE B.author_ID=A.author_ID AND A.Lname='Pratchett' AND B.Price<10
GROUP BY B.title;
```

2. Give all the titles and their dates of purchase made by A B

Relational algebra expression:

```
CUSTOMER_ORDERS  $\bowtie_{\text{customerID} = \text{customerID}}$  ORDER
ORDER_DETAILS  $\bowtie_{\text{bookID} = \text{bookID}}$  BOOK
RESULT  $\pi_{\text{title, date}}$  ( $\sigma_{\text{Fname} = \text{'John'} \text{ AND Lname} = \text{'Smith'}}$  (ORDER_DETAILS))
```

SQL code:

```
SELECT B.title, O.date
FROM Customer C, Orders O, Book B, Buys A
WHERE C.customer_ID = A.customer_ID AND A.order_ID = O.order_ID AND B.book_ID =
A.book_ID AND C.fname = 'A'
GROUP BY B.title;
```

3. Find the titles and ISBNs for all books with less than 5 copies in stock

Relational algebra expression:

```
STOCK_WAREHOUSE  $\bowtie_{\text{bookID} = \text{bookID}}$  BOOK
LOW_STOCK  $\sigma_{\text{amount} < 5}$  STOCK
RESULT  $\pi_{\text{title, ISBN}}$  (LOW_STOCK)
```

SQL code:

```
SELECT B.ISBN, B.title
FROM Book B, WAREHOUSE W
WHERE B.book_ID = W.book_ID
GROUP BY 1,2
HAVING count(*)<5;
```

4. Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased

Relational algebra expression:

$$\begin{aligned} & \text{Pratchett_BOOK_ORDER } (\sigma_{\text{Lname} = \text{'Pratchett'}} (\text{AUTHOR})) \bowtie_{\text{bookID}=\text{bookID}} \text{ORDER} \\ & \text{Pratchett_BOOK } (\sigma_{\text{Lname}=\text{'Pratchett'}} \text{AUTHOR}) \bowtie_{\text{authorID}=\text{authorID}} \text{BOOK} \\ & \text{CUSTOMER_Pratchett_BOOK } \text{Pratchett_BOOK_ORDER} \bowtie_{\text{customerID}=\text{customerID}} \text{CUSTOMER} \\ & \text{RESULT } \pi_{\text{Fname,Lname,title}} \text{CUSTOMER_Pratchett_BOOK} \end{aligned}$$

SQL code:

```
SELECT C.fname, C.lname, B.title
FROM Customer C, Book B, Orders O, Author A, Buys K
WHERE C.customer_ID=K.customer_ID AND O.order_ID=K.order_ID AND
B.author_ID=A.author_ID AND K.book_ID = B.book_ID AND A.lname='Pratchett'
ORDER BY C.fname;
```

5. Find the total number of books purchased by A B

Relational algebra expression:

$$\begin{aligned} & \text{SINGLE_CUSTOMER } \sigma_{\text{customerID} = \text{customerID}} (\text{CUSTOMER}) \\ & \text{CUSTOMER_ORDERS } \text{SINGLE_CUSTOMER} \bowtie_{\text{customerID} = \text{customerID}} (\text{ORDER}) \\ & \text{ORDER_DETAILS } \text{CUSTOMER_ORDERS} \bowtie_{\text{bookID} = \text{bookID}} (\text{BOOK}) \\ & \text{BMAX } \mathcal{F}_{\text{COUNT bookID}} (\text{ORDER_DETAILS}) \\ & \text{RESULT } \pi_{\text{total_book}} (\sigma_{\text{Fname} = \text{'John'} \text{ AND } \text{Lname} = \text{'Smith'}} (\text{BMAX})) \end{aligned}$$

SQL code:

```
SELECT C.Fname, C.Lname, count(A.book_ID)
FROM CUSTOMER C, BUYS A
WHERE C.Fname = 'A' AND C.Lname = 'B' AND A.customer_ID = C.customer_ID;
```

6. Find the customer who has purchased the most books and the total number of books they have purchased

Relational algebra expression:

$$\begin{aligned} & \text{C_BOOKS } \pi_{\text{customerID, total_book}} (\text{CUSTOMER}) \\ & \text{RESULT } \text{customerID } \mathcal{F}_{\text{MAX total_book}} (\text{C_BOOKS}) \end{aligned}$$

SQL code:

```
SELECT C.Fname, C.lname, max(y.num)
FROM Customer C, Buys A, (SELECT count(A.book_ID) as num
FROM BUYS A, Customer C
WHERE A.customer_ID = C.customer_ID
GROUP BY C.customer_ID) y
WHERE C.customer_ID = A.customer_ID;
```

7. Find the employee with the highest salary (aggregate MAX, and extra entity)

Relational algebra expression:

$$\begin{aligned} & \text{SALARIES } \pi_{\text{salary, employeeID}} (\text{EMPLOYEE}) \\ & \text{RESULT } \pi_{\text{employeeID}} (\text{employeeID } \mathcal{F}_{\text{MAX salary}} (\text{SALARIES})) \end{aligned}$$

SQL code:

```
SELECT E.employeeID, MAX(E.salary)
FROM Employee E;
```

8. Find all books published by Pearson

Relational algebra expression:

```
BOOK_DETAILS PUBLISHER * BOOK  
RESULT  $\sigma_{\text{publisher\_name} = \text{'pearson'}}$  (BOOK_DETAILS)
```

SQL code:

```
FROM PUBLISHER P, BOOK B  
WHERE P.publisher_name = 'Pearson'  
AND P.book_ID = B.book_ID;
```

9. Find how many books are in each warehouse

Relational algebra expression:

```
BOOKS_IN_WAREHOUSE BOOK * WAREHOUSE  
RESULT  $\pi_{\text{amount}}$  (BOOKS_IN_WAREHOUSE)
```

SQL code:

```
SELECT W.warehouse_ID, count(B.book_ID)  
FROM WAREHOUSE W, BOOK B  
WHERE W.book_ID = B.book_ID  
GROUP BY W.warehouse_ID;
```