6

Question 2: Relational Algebra

Part 1: Consider the following relational schema: **[/4 Marks]**

BOOK (DocId, Title, Publisher, Year)
STUDENT (StId, StName, Major, Age)
BORROWS (DocId, StId, Date)
AUTHOR (AName, Nationality)
HAS-WRITTEN (DocId, AName)

- **Write relational algebra expression for the following queries:**

$\frac{1}{2}$ 1. **List the year and title of each book.**
$\pi_{\text{year, title}}$ **(BOOK)**

1 2. **List all information about students whose major is CS.**
$\sigma_{\text{major='CS'}}$**(STUDENT)**

1 3. **List the name of those authors who are french.**
$\pi_{\text{Anam}}(\sigma_{\text{nationality='French'}}$**(AUTHOR))**

1 4. **List the authors of the books the student 'Smith' has borrowed.**
$\pi_{\text{AName}}(\sigma_{\text{StName}}=\text{'Smith'}(\text{has-written} \bowtie (\text{borrows} \bowtie \text{STUDENT}))$

$\frac{1}{2}$ 5. **Find the title of the oldest book; solve it without the use of aggregate functions (hint: Use Cartesian Product)**

$$\pi_{\text{Title}}(\text{BOOK}) - \pi_{\text{B1.Title}}(\sigma_{\text{B1.Year>B2.Year}}(\rho\text{B1}(\text{BOOK}) \times \rho\text{B2}(\text{BOOK})))$$

Part 2: Given the following set of tables: [/2 Marks]

R

A	B	C	D
1	2	3	J
3	2	1	K
2	1	2	L
5	4	7	M
6	9	6	N

S

E	A	D
8	7	J
16	3	K
3	2	J
9	5	M
8	1	L

- **For each of the following expressions, represent the resulting table:**

1. $\pi_D \; (\sigma_{A=C \text{ OR } A>B} \; (R))$

1/2

Solution:

D
K
L
M
N

2. $\pi_{A,B,D} \; (\sigma_{B<=4 \text{ AND } C=1} \; (R * S))$

1/2

Solution:

A	B	D
3	2	K

3. $(\pi_{A,B}(R)) \div (\pi_A \; (\sigma_{D='K' \text{ OR } D='L'} \; (S))$

1/2

Solution:

B
2

4. $\pi_{A,D,E} \; (R \; ⟕_{R.A=S.A \text{ AND } R.D=S.D} \; S)$

1/2

Solution:

A	D	E
1	J	Null
3	K	16
2	L	Null
5	M	9
6	N	Null

End of Exam Questions

Part 2: The Premiere Products database contains the following four tables about customers, orders, order line and part. Attributes belonging to the primary keys are underlined. [/7 Marks]

CUSTOMER

CUST_NUMB	CUST_FIRST	CUST_LAST	CUST_ADDR	CUST_CITY	CUST_STATE	CURRENT_BALANCE	CREDIT_LIMIT
124	SALLY	ADAMS	481 OAK	LANSING	MI	$418.75	$500.00
256	ANN	SAMUELS	215 PETE	GRANT	MI	$10.75	$800.00
311	DON	CHARLES	48 COLLEGE	IRA	MI	$200.10	$300.00
315	TOM	DANIELS	914 CHERRY	KENT	MI	$320.75	$300.00

ORDERS

ORDER_NUMB	ORDERDATE	CUST_NUMB
12489	9/2/1987	124
12491	9/2/1987	311
12494	9/4/1987	315
12498	9/5/1987	256

ORDERLINE

ORDER_NUMB	PART_NUMB	NUM_ORDERED
12489	AX12	11
12491	BT04	1
12491	BZ66	1
12494	CB03	4
12498	AZ52	2
12498	BA74	4

PART

PART_NUMB	PART_DESC	Available_Units	Warehouse_NUMB	UNIT_PRICE
AX12	IRON	104	3	$17.95
AZ52	SKATES	20	2	$24.95
BA74	BASEBALL	40	1	$4.95
BH22	TOASTER	95	3	$34.95
BT04	STOVE	11	2	$402.99
BZ66	WASHER	52	3	$311.95
CA14	SKILLET	2	3	$19.95
CB03	BIKE	44	1	$187.50

- **Write SQL queries that answer the following:**

1

1. **Find the names of all the customers who have a credit limit of at least $800.**

```
SELECT CUST_FIRST, CUST_LAST
        FROM CUSTOMER
        WHERE CREDIT_LIMIT >= 800;
```

1

2. List the number and first and last name of all customers whose balance is over the average balance of all customers.

```
SELECT CUST_NUMB, CUST_FIRST, CUST_LAST, CURRENT_BALANCE
        FROM CUSTOMER
        WHERE CURRENT_BALANANCE >
                (SELECT AVG(CURRENT_BALANCE)
                FROM CUSTOMER);
```

1

3. Find the number and date of those orders that do not contain part "BT04".

```
SELECT ORDER_NUMB, ORDERDATE
        FROM ORDERS
        WHERE NOT EXISTS
                (SELECT *
                        FROM ORDERLINE
                        WHERE ORDERS.ORDER_NUMB =
                                ORDERLINE.ORDER_NUMB
                        AND PART_NUMB = 'BT04');
```

or

Having not allow to use Agr. fun within them of sub select

or

<> BT04 !

1

4. Find all of the numbers and dates of those orders that include a part located in warehouse 3.

```
SELECT ORDERS.ORDER_NUMB, ORDERDATE
        FROM ORDERLINE, ORDERS, PART
        WHERE ORDERLINE.ORDER_NUMB = ORDERS.ORDER_NUMB
        AND ORDERLINE.PART_NUMB = PART.PART_NUMB
        AND Warehouse_NUMB= 3;
```

Or by using multiple nesting.

1

5. List the number and first name of all customers who have a credit limit of $800 but who do not currently have orders on file.

```
SELECT CUST_NUMB, CUST_FIRST
        FROM CUSTOMER
        WHERE CREDIT_LIMIT = 800;
MINUS
SELECT CUSTOMER.CUST_NUMB, CUST_FIRST
        FROM CUSTOMER, ORDERS
        WHERE CUSTOMER.CUST_NUMB = ORDERS.CUST_NUMB;
```

1

6. List each credit limit together with the number of customers who have this limit, but only list those credit limits held by more than one customer.

```
SELECT CREDIT_LIMIT, COUNT(CUST_NUMB)
        FROM CUSTOMER
        GROUP BY CREDIT_LIMIT
        HAVING COUNT(CUST_NUMB) > 1;
```

1

7. Change the address of customer number 124 to "111 Brookhollow".

```
UPDATE Customer
        SET CUST_ADDR = '111 Brookhollow'
        WHERE CUST_NUMB = 124;
```