

1.  $\Pi \text{ status, branch}(\sigma(\text{barcode} = 1) \wedge (\text{status} = \text{available}) \text{branch} \bowtie \text{libraryitem})$

(It returns the status, library branch of a given library item with available status. It joins “libraryitem” table and “branch” table via branch name)

2

$\Pi \text{ name, dateout, returndate, fines}(\sigma(\text{barcode} = 1)(\text{libraryitem} \bowtie \text{libraryitem.barcode} = \text{borrow.borroweditem}..\text{borrow} \bowtie \text{borrow.memberid} = \text{patron.memberId}..\text{patron}))$

(It returns name of a patron, borrowed date, return date and overdue fine for a given library item. This relational algebra joins “borrow” and “libraryitem” tables via barcode and “borrow” and “patron” tables via memberId)

3.

$\Pi \text{ branch, row, section}(\sigma(\text{barcode} = 1)(\text{libraryitem} \bowtie \text{libraryitem.rackId} = \text{rack.rackId}..\text{rack}))$

(It returns which branch and rack (section and row) a given library item found, it inner joins “libraryitem” and “rack” table via “rackId”)

4.

$\Pi \text{ name, fines, status}(\sigma(\text{fines} > 50)(\text{patron} \bowtie \text{patron.memberId} = \text{borrows.memberId}..\text{borrows}))$

(This relational algebra selects a patron with over due fines greater than \$50, look for the name, fines and status (active/ inactive). It uses “patron” and “borrow” table for the inner join via “memberId” which is found as a primary key in “patron” table and as a foreign key in “borrow” table)

5.

$\Pi \text{ name, reservationstatus, creationdate}(\sigma(\text{barcode} = 1)(\text{patron} \bowtie \text{patron.memberId} = \text{reserves.memberId}..\text{reserves} \bowtie \text{reserves.reservedItem} = \text{libraryitem.barcode}..\text{libraryitem}))$

(It returns name of a patron, reservation status and creation date for a given library item reservation. It joins “patron” table and “libraryitem” table via “memberId” and “reserves” and “libraryitem” via “barcode”)

6.

$$\prod \text{barcode}, \text{callnum}, \text{MarclD}(\sigma(\text{barcode} = 1) \vee (\text{callnum} = 2)(\text{libraryitem} \bowtie \text{libraryitem.callnum} = \text{work.callnum}..\text{work}))$$

(This relational algebra returns catalog details of a given library item, it joins “libraryitem” and “work” tables via call number which is a primary key in “work” and foreign key in “libraryitem”)

7.

$$\prod \text{name}, \text{work}(\sigma(\text{ldreserve} = 1)(\text{branch} \bowtie \text{branch.name} = \text{reserves.pickup}..\text{work}))$$

(It returns library branch name and which item a given reservation for. It joins “reserves” and “branch” table using “name” which is a primary key for “branch” table and “pickup” which is a foreign key for “work” table )

$$\prod \text{name}, \text{status}, \text{email}, \text{phonenum}(\sigma(\text{cardnum} = 1)(\text{librarycard} \bowtie \text{librarycard.member} = \text{patron.memberId}..\text{patron}))$$

8.

(It returns name, email, phonenum of a patrons whose library card status is inactive. It joins “librarycard” and “patron” table via “member” which is a foreign key for “librarycard” table and “memberId” which is a primary key for “patron” table)

9.

$$\prod \text{work}, \text{reservationDate}, \text{marclD}(\sigma(\text{pickup} = \text{branch1}) \wedge (\text{status} = \text{reserved})(\text{work} \bowtie \text{work.callNum} = \text{reserves.callnum}..\text{reserves} \bowtie \text{reserves.pickup} = \text{branch.name}..\text{branch}))$$

(This relational algebra returns catalog details and reservation date of all reserved items from a given library branch. It joins “work” and “reserves” table via “callnum” and “branch” and “reserve” table via “pickup” which a foreign key in branch table and name which is primarykey in “branch” table)

10.

$$\prod \text{barcode}, \text{dateout}, \text{returndate}, \text{fines}(\sigma(\text{memeberId} = 1) \wedge (\text{libraryitem.status} = \text{borrowed})(\text{libraryitem} \bowtie \text{libraryitem.barcode} = \text{borrow.item}..\text{borrowborrow.memberId} = \text{patron.memeberIdpatron}))$$

(It return which items a given patron borrowed, its borrowed date, return date and fine for each borrowed library item. It joins "libraryitem" and "borrow" table via "barcode" and "borrow" and "patron" via "memberId")