

SOLVE QUERY OF LIBRARY DATABASE EXERCISE

1. Write a query to find the title, author name, publisher name of all the books available in the library.

A:SELECT books.books_title,authors.authors_name,publisher.name as publisher_name
from books,authors,publisher where books.author_id=authors.author_id and
books.publisher_id=publisher.publisher_id;

2. SQL query to list the details of borrowers who do not borrow any books yet.

A:SELECT * from borrowers where borrower_id not in(SELECT DISTINCT
borrow_book.borrower_id from borrow_book);

3. Write a query to display the name of an author who have write a book having 'e' as the second character.

A:SELECT authors.author_name ,books.book_title FROM authors INNER JOIN books on
authors.author_id=books.author_id WHERE book_title like '_e%';

4. SQL query to list the details of borrowers who have borrowed more than five books.

A:SELECT * ,COUNT(borrowers.borrower_id) FROM borrowers INNER JOIN borrow_book
on borrowers.borrower_id=borrow_book.borrower_id GROUP BY
(borrow_book.borrower_id) HAVING COUNT(borrowers.borrower_id)>=5;

5. Show the details of the book which have been borrowed most.

A:SELECT * ,COUNT(borrow_book.book_id) FROM books INNER JOIN borrow_book on
books.book_id=borrow_book.book_id GROUP BY(borrow_book.book_id) limit 1;

6. Write a query to show the title and author of each book.

A:SELECT books.book_title , authors.author_name from books INNER join authors on
authors.author_id=books.author_id;

7. Create a view for the following problem:

Show the list of those borrower who have crossed the due date.

A:SELECT * FROM borrowers INNER join borrow_book on borrow_book.borrower_id=
borrowers.borrower_id and borrow_book.duedate>CURRENT_DATE;

8. Write a query to get the borrower name (firstanme and lastname), book name and borrowing month.

A:SELECT concat(borrowers.firstname,' ',borrowers.lastname) as borrower_name
,books.book_title,extract(month from borrow_book.borrowingdate) FROM borrowers INNER
join books inner join borrow_book on borrow_book.book_id=books.book_id AND
borrow_book.borrower_id=borrowers.borrower_id;

9. First execute the following command:

CREATE TABLE witters(

```
writer_id INT NOT NULL AUTO_INCREMENT,  
writer_name varchar(40),  
writer_address varchar(40),
```

```
PRIMARY KEY(writer_id)  
);
```

Now, Write a SQL statement insert rows to writers table from authors table.

```
A:INSERT into writers SELECT * FROM authors;  
10
```

.

Write a query to update the portion of the phone in the borrowers table, within the phone number the substring '337' will be replaced by '999'.

```
A:SELECT REPLACE(phone,'337','999') FROM borrowers;
```

SQL LABTEST2

a.SELECT company_name ,city FROM company_name WHERE city='Dhaka';

b.SELECT employee.id,employee.employee_name ,works.salary FROM employee
INNER join works on employee.id=works.employee_id ORDER by salary DESC limit
2;

c.SELECT comp_id,COUNT(employee_id) AS NUM_OF_EMPLOYEE FROM works
GROUP by comp_id;

d.SELECT employee.id,employee.employee_name,works.salary from employee
INNER join works on employee.id=works.employee_id WHERE salary>(SELECT
AVG(salary) FROM works);

e.UPDATE works

set salary=salary*1.20

WHERE salary<30000;