LAB 3

1. Creating a database called LIBRARY and doing some queries:

Task #1 Creating database called LIBRARY

```
create database LIBRARY;
```

Task #2 Using Database in default

```
use LIBRARY;
```

Task #3 Solution:

```
create table books
(
	book_id int primary key,
	title varchar(100),
	author varchar(50),
	genre varchar(50),
	publication_year int,
	available_copies int
);
```

Task #4 Solution:

```
insert into books values(1, 'The Catcher in the Rye', 'J.D. Salinger', 'Fiction', 1951, 10);
insert into books values(2, 'To Kill a Mockingbird', 'Harper Lee', 'Classics', 1960, 15);
insert into books values(3, '1984', 'George Orwell', 'Dystopian', 1949, 8);
insert into books values(4, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Classics', 1925, 12);
insert into books values(5, 'Pride and Prejudice', 'Jane Austen', 'Romance', 1813, 20);
```

Task #5 Order by publication year in ascending order

```
select * from books
order by publication_year;
```

Order by available copies in descending order

```
select * from books
order by available_copies desc;
```

Task #6 Order by title in ascending order

```
select * from books
order by title;
```

Order by title in descending order

```
select * from books
order by title desc;
```

Task #7 Filter by book_id

```
select * from books
where book_id = 3;
```

Task #8 Filter by specific title

```
select * from books
where title = 'The Great Gatsby';
```

Task #9 Add ISBN attribute in table

```
alter table books
add column ISBN varchar(13) default NULL;
```

Task #10 Drop genre column from table

```
alter table books
drop column genre;
```

2. Creating a database called LIBRARY and doing some queries:

Task #1: Creating a database called SMART_BANK

```
create database SMART_BANK;
```

Task #2: Using the SMART_BANK database

```
use SMART_BANK;
```

Task #3: Creating a table 'accounts' with specific attributes

```
create table accounts
(
        account_number int primary key,
        account_holder varchar(50),
        account_type varchar(20),
        balance decimal(10, 2),
        branch varchar(50),
        last_transaction_date date
);
```

Task #4: Inserting data into the 'accounts' table

```
insert into accounts values(1001, 'John Doe', 'Savings', 5000.00, 'Main Branch',
'2023-01-15');
insert into accounts values(1002, 'Jane Smith', 'Checking', 8000.00, 'Downtown Branch',
'2023-02-01');
insert into accounts values(1003, 'Bob Johnson', 'Savings', 12000.00, 'West Branch',
'2022-12-20');
insert into accounts values(1004, 'Alice Williams', 'Checking', 10000.00, 'East Branch',
'2023-01-05');
insert into accounts values(1005, 'Charlie Brown', 'Savings', 7000.00, 'North Branch',
'2023-02-10');
```

Task #5: Ordering data by balance in ascending order

```
select * from accounts
order by balance;
```

Ordering data by last_transaction_date in descending order

```
select * from accounts
order by last_transaction_date desc;
```

Task #6: Ordering data by account_holder in ascending and descending order

```
select * from accounts
order by account_holder;

select * from accounts
order by account_holder desc;
```

Task #7: Filtering data by account_number

```
select * from accounts
where account_number = 1003;
```

Task #8: Filtering data by specific account_holder

```
select * from accounts
where account_holder = 'Alice Williams';
```

Task #9: Adding phone_number attribute to the 'accounts' table

```
alter table accounts
add column phone_number varchar(15) default NULL;
```

Task #10: Dropping the 'branch' column from the 'accounts' table

```
alter table accounts
drop column branch;
```

Discussion:

The SQL tasks performed in the lab showcased fundamental database management operations within the "LIBRARY" and "SMART_BANK" databases. Creating tables, inserting data, and executing queries to sort, filter, and alter table structures demonstrated essential skills for managing relational databases. The "LIBRARY" database tasks focused on book information, covering diverse operations, while the "SMART_BANK" database tasks dealt with account management, emphasizing practical aspects of data manipulation. The lab provides a practical understanding of SQL commands for users aiming to work effectively and offering hands-on experience in common tasks.

Conclusion:

The lab equips us with essential skills in SQL for effective database management. Through hands-on tasks, we gained proficiency in creating and manipulating tables, inserting and retrieving data, and modifying table structures. These skills are foundational for roles involving database administration, software development, and data analysis. The lab's structured approach ensures that users not only understand the syntax but also understands the practical application of SQL commands in real-world scenarios, laying a solid groundwork for further exploration and mastery in database management.