Laboratory work 7

1. How can we store large-object types?
   Many database applications need to store attributes whose domain consists of large data items such as a photo, a high-resolution medical image, or a video. SQL, therefore, provides large-object data types for character data (clob) and binary data (blob). The letters “lob” in these data types stand for “Large OBject.”
   For result tuples containing large objects (multiple megabytes to gigabytes), it is inefficient or impractical to retrieve an entire large object into memory. Instead, an application would usually use an SQL query to retrieve a “locator” for a large object and then use the locator to manipulate the object from the host language in which the application itself is written.
2. What is the difference between privilege, role and user?
   A role is performed by one or more users. A user is an individual person who is included in the role We may assign a user several forms of authorizations on parts of the database. Authorizations on data include:
   • Authorization to read data.
   • Authorization to insert new data.
   • Authorization to update data.
   • Authorization to delete data.
   Each of these types of authorizations is called a privilege. A user privilege is a right to execute a particular type of SQL statement, or a right to access another user's object.
   A role groups several privileges and roles, so that they can be granted to and revoked from users simultaneously. A role must be enabled for a user before it can be used by the user. Privileges control the ability to run SQL statements. A role is a group of privileges. Granting a role to a user gives them the privileges contained in the role.
   - create accountant, administrator, support roles and grant appropriate privileges

```
create role accountant;
create role administrator;
create role support;
grant select on accounts to accountant;
grant select, update on transactions to accountant;
grant all privileges on accounts, transactions, customers to administrator;
grant select on accounts, transactions, customers to support;
```

   - create some users and assign them roles,

```
create user user1;
create user user2;
create user user3;
grant accountant to user1;
grant administrator to user2;
grant support to user3;
```

   - give to some of them permission to grant roles to other users

```
create role grant_roles createrole;
grant grant_roles to user2;
```

   - revoke some privilege from particular user

```
revoke select on accounts from user1;
revoke select on accounts from user3;
```

3. Create indexes:
    - index so that each customer can only have one account of one currency

```
create unique index a_index on accounts(account_id, customer_id, currency);
```

    - index for searching transactions by currency and balance

```
create index search_tr on accounts(currency, balance);
```

4. Write a SQL transaction that illustrates money transaction from one account to another:
    - create transaction with “init” status

```
insert into transactions values (4, '2021-07-05 18:09:35.000000', 'DB05053', 'DB03039', 100,
'init');
```

    - increase balance for destination account and decrease for source account

```
update accounts set balance=balance+10 where account_id='DB03039';
update accounts set balance=balance-10 where account_id='DB05053';
```

    - if in source account balance becomes below limit, then make rollback
    - update transaction with appropriate status(commit or rollback)