

1. create or replace procedure new_transaction(transaction_date DATE, transaction_amount number, transaction_receiver varchar2, transaction_remarks varchar2, transaction_info out varchar2)
as
BEGIN
 Insert into transaction(transaction_id, transaction_date, transaction_receiver, transaction_amount, transaction_remarks) values (TRANSACTION_SEQ.nextval, transaction_date, transaction_receiver, transaction_amount, transaction_remarks);
 transaction_info:='Transaction_inserted';

EXCEPTION

 When others then

 transaction_info:'Error due to || SQLERRM;

end;

/

Variable info_insert varchar2;

Execute new_transaction('12Feb2024', 7688, 'annapoorna', 'Family', :info_insert);

Print info_insert;

2. create or replace procedure delete_transaction(transaction_id number, transaction_info out varchar2)
as
BEGIN
 Delete from transaction where transaction_id=transaction_id;
 Trans_info:='Transaction deleted';
EXCEPTION
 When no_data_found then
 Trans_info:='No such Transaction available';
 When others then
 Trans_info:='Error due to' ||SQLERRM;

end;

/

3. create or replace procedure select_transaction(transaction_remarks varchar2, transaction_receiver out varchar2, transaction_amount out varchar2, transaction_info out varchar2)
as
BEGIN
 Select transaction_receiver, transaction_amount into transaction_receiver, transaction_amount from transaction where transaction_remarks=transaction_remarks;

EXCEPTION

 When no_data_found then

```
Transaction_info:='No such Transaction available';

When others then

Transaction_info:='error due to' || SQLERRM;

End;

/

Variable found_receiver varchar2;

Variable found_amount varchar2;

variable info_error varchar2;

execute select_transaction('Education',:found_receiver,:found_amount,:info_error);

print found_num;

print found_amount;

print info_error;
```