CSE 4508 – RDBMS Programming Lab

Lab 9

Group 1A

Instructor: Shahriar Ivan

Materials: Prof. Dr. Abu Raihan Mostafa Kamal

Task 1:

IUT wants to create a database for all students. The student table consists of four rows: ID (varchar2), Date of Admission (date), Department (char), Program (char), Section (char). Department, Program and Section are 1 character strings with values ranging from '1' to '6'. Now create a function Gen_ID which generates an ID (a string) automatically in the following format:

YY00DPSXX

YY are the last two digits of the year, extracted from the Date of Admission. D is Dept number, P is Programme number and S is Section Number each of 1 character. *XX* is an automatically generated number (you can use a sequence, like 01 for first entry, then 02 for next and so on) that increments as you add new members. Once you have made Gen_ID, create a trigger such that, every time you insert into the table Student with only the Date of Admission, Department, Program, and Section, the ID is not given as input and rather automatically generated by calling Gen_ID inside the trigger and then inserted automatically along with the other components.

Task 2:

Accounts (ID, Name, AccCode, Balance, LastDateofInterest)

AccountProperties (ID, Name, interestRate, GP)

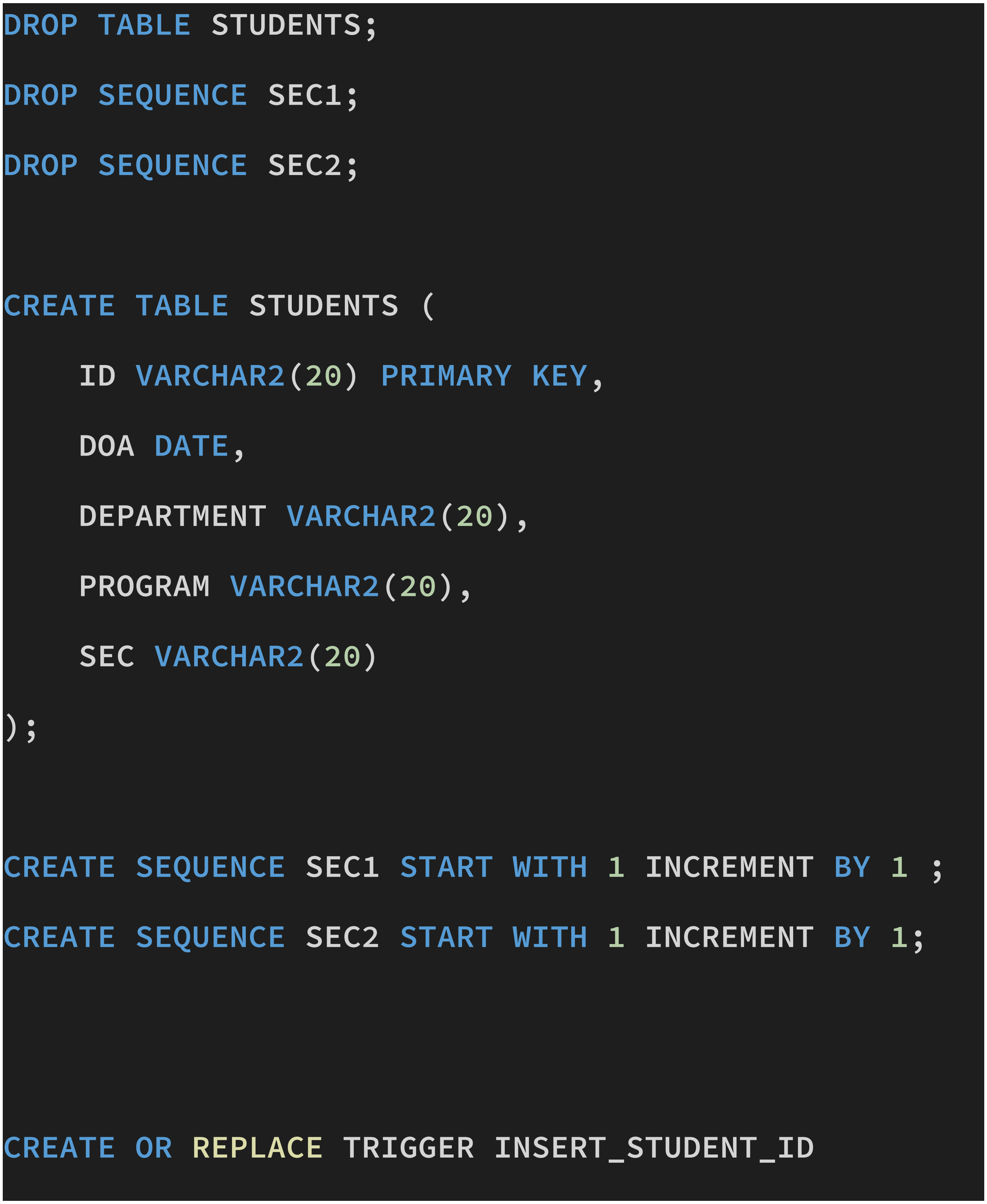
GP = 1 means daily

GP = 2 means monthly

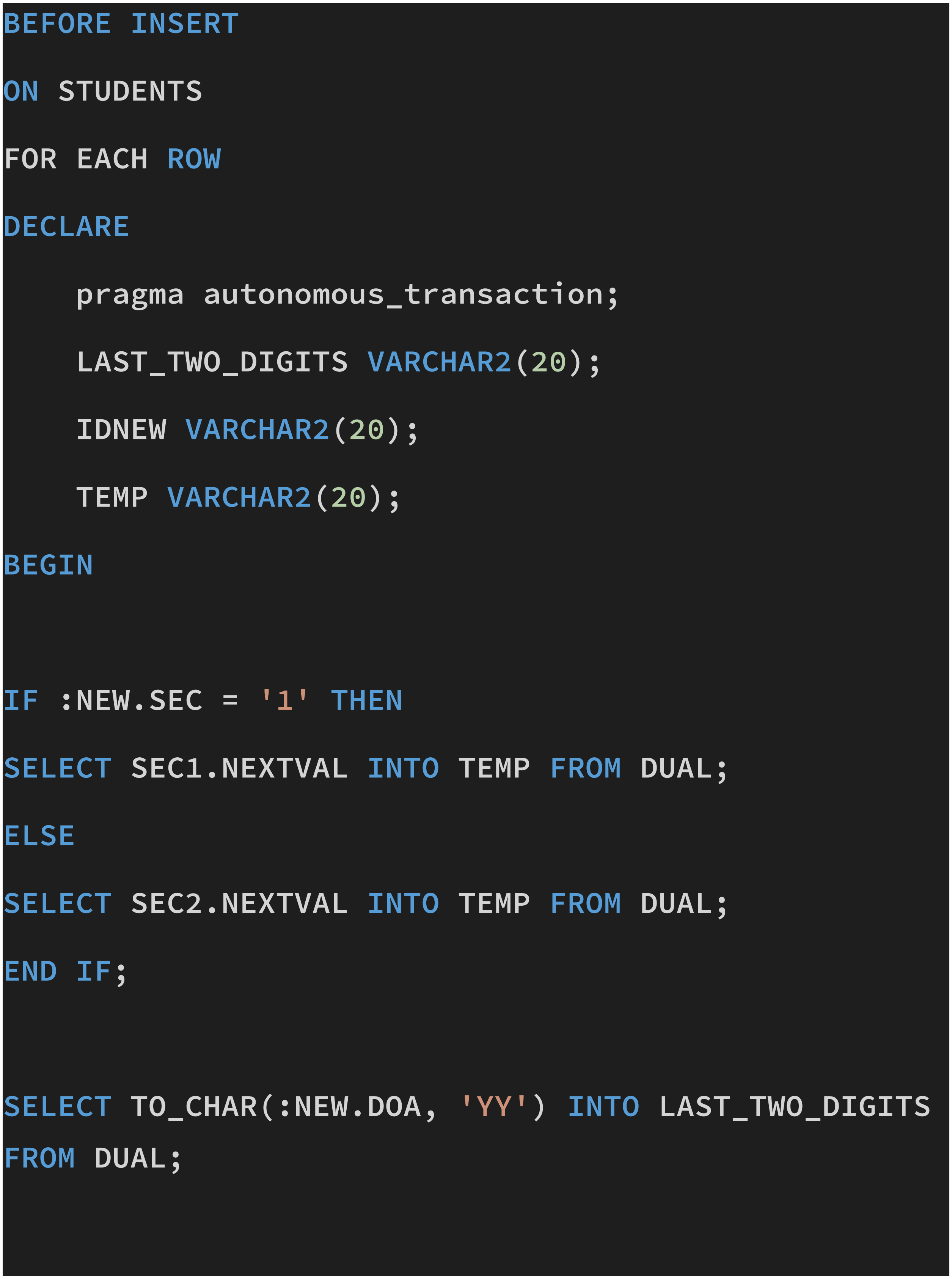
GP = 3 means yearly

Using an explicit cursor, write a procedure that updates all account current balances adding interest if it satisfies the condition.

Task 1

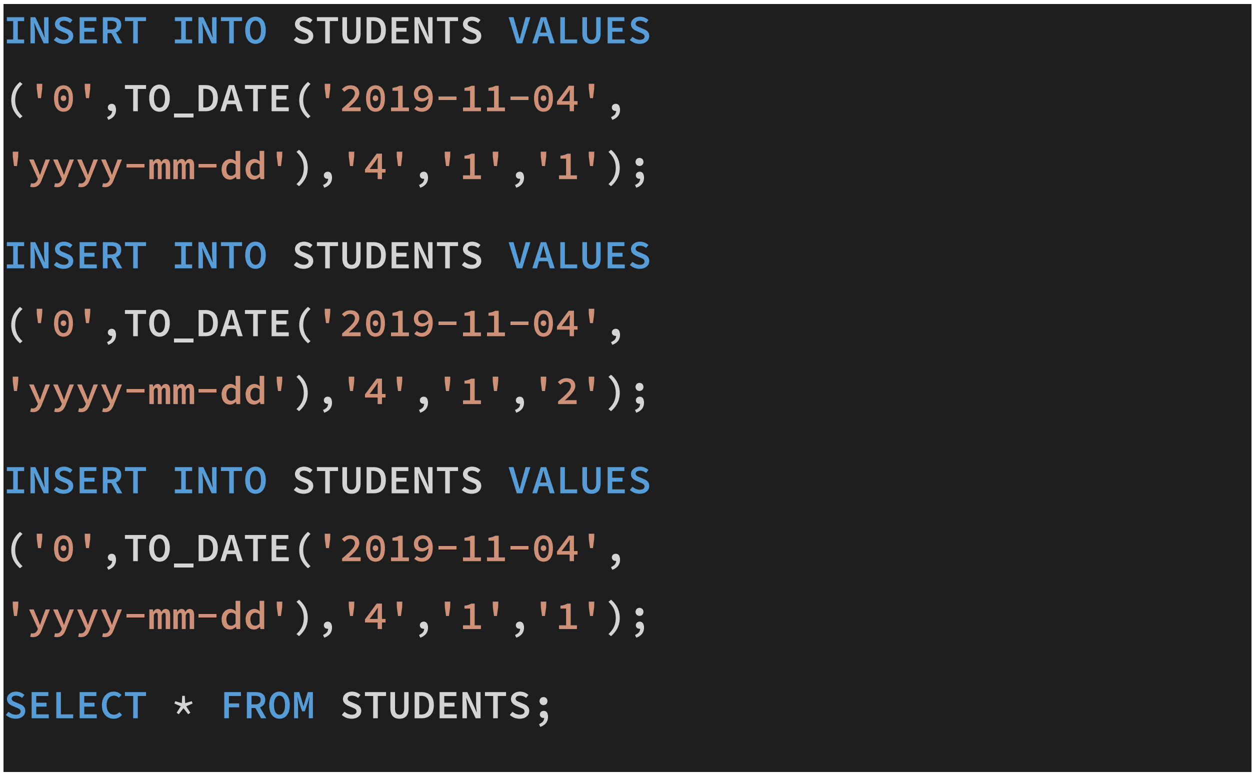


```
DROP TABLE STUDENTS;
DROP SEQUENCE SEC1;
DROP SEQUENCE SEC2;

CREATE TABLE STUDENTS (
    ID VARCHAR2(20) PRIMARY KEY,
    DOA DATE,
    DEPARTMENT VARCHAR2(20),
    PROGRAM VARCHAR2(20),
    SEC VARCHAR2(20)
);

CREATE SEQUENCE SEC1 START WITH 1 INCREMENT BY 1 ;
CREATE SEQUENCE SEC2 START WITH 1 INCREMENT BY 1;


CREATE OR REPLACE TRIGGER INSERT_STUDENT_ID
```

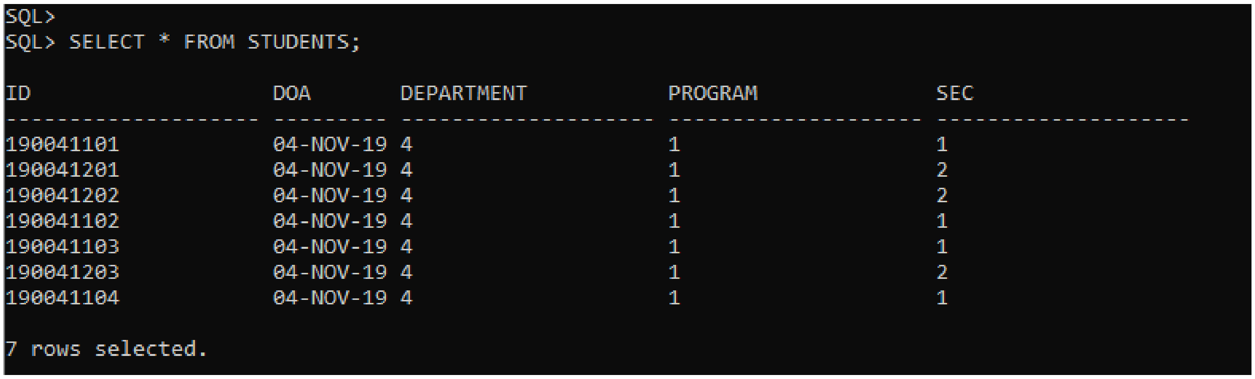


```
BEFORE INSERT
ON STUDENTS
FOR EACH ROW
DECLARE
    pragma autonomous_transaction;
    LAST_TWO_DIGITS VARCHAR2(20);
    IDNEW VARCHAR2(20);
    TEMP VARCHAR2(20);
BEGIN

IF :NEW.SEC = '1' THEN
SELECT SEC1.NEXTVAL INTO TEMP FROM DUAL;
ELSE
SELECT SEC2.NEXTVAL INTO TEMP FROM DUAL;
END IF;

SELECT TO_CHAR(:NEW.DOA, 'YY') INTO LAST_TWO_DIGITS FROM DUAL;
```

```
IDNEW := LAST_TWO_DIGITS || '00' || :NEW.DEPARTMENT
|| :NEW.PROGRAM || :NEW.SEC ||  '0' || TEMP;


:NEW.ID := IDNEW;


commit;
END;
/
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','1');
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','2');
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','2');
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','1');
```

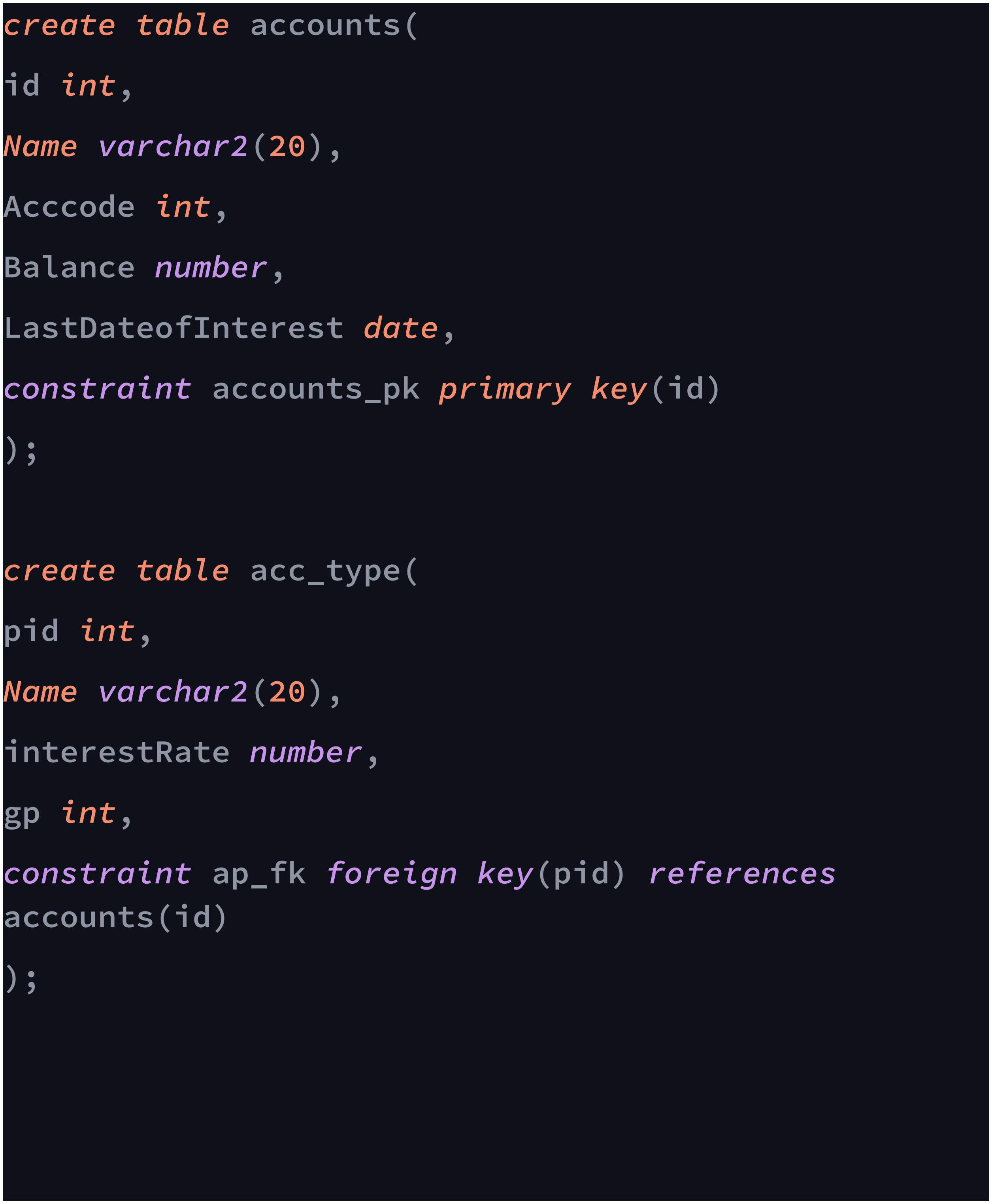


```sql
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','1');
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','2');
INSERT INTO STUDENTS VALUES
('0',TO_DATE('2019-11-04',
'yyyy-mm-dd'),'4','1','1');
SELECT * FROM STUDENTS;
```

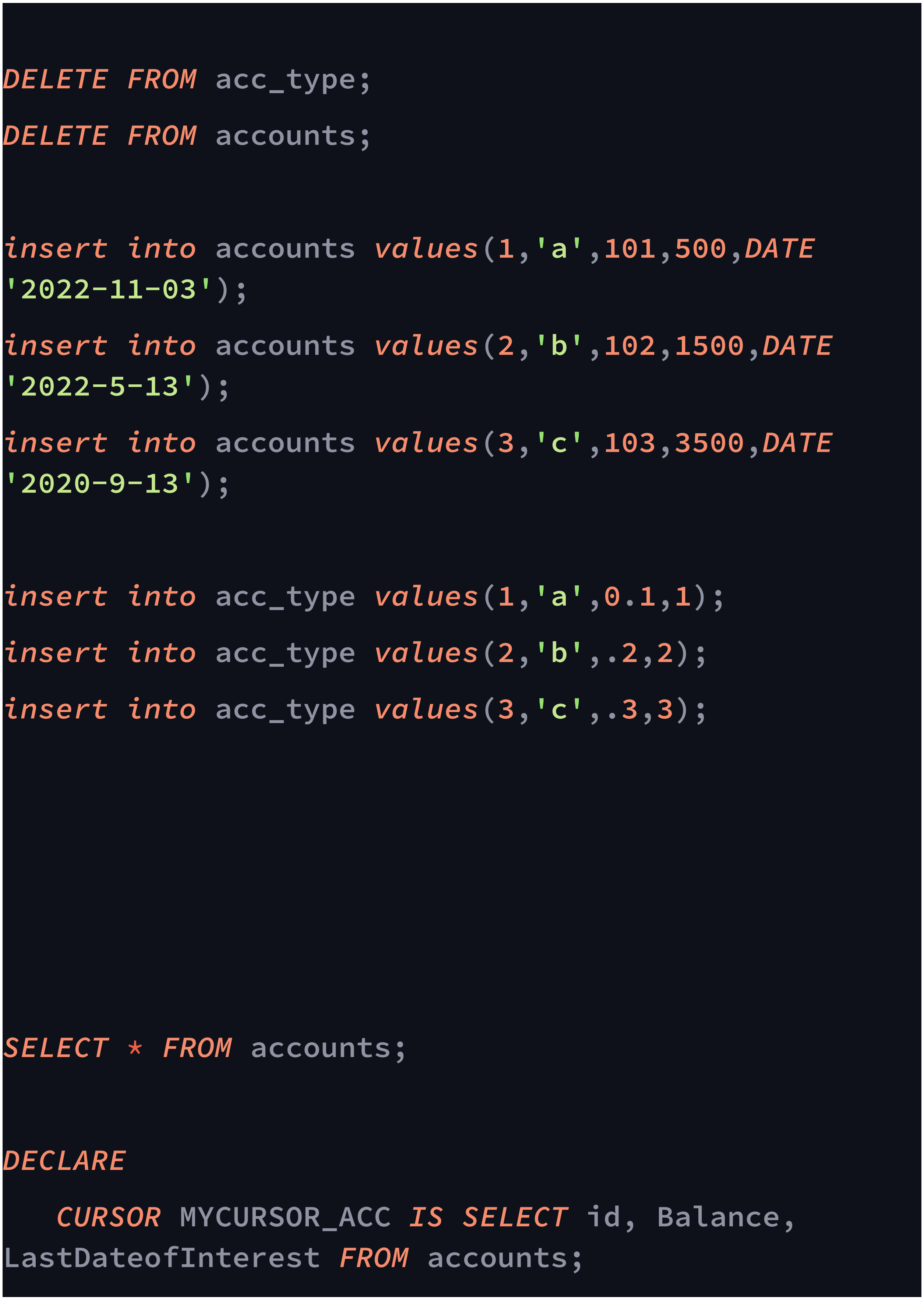


```
SQL>
SQL> SELECT * FROM STUDENTS;

ID                   DOA       DEPARTMENT           PROGRAM              SEC
-------------------- --------- -------------------- -------------------- --------------------
190041101            04-NOV-19 4                    1                    1
190041201            04-NOV-19 4                    1                    2
190041202            04-NOV-19 4                    1                    2
190041102            04-NOV-19 4                    1                    1
190041103            04-NOV-19 4                    1                    1
190041203            04-NOV-19 4                    1                    2
190041104            04-NOV-19 4                    1                    1

7 rows selected.
```

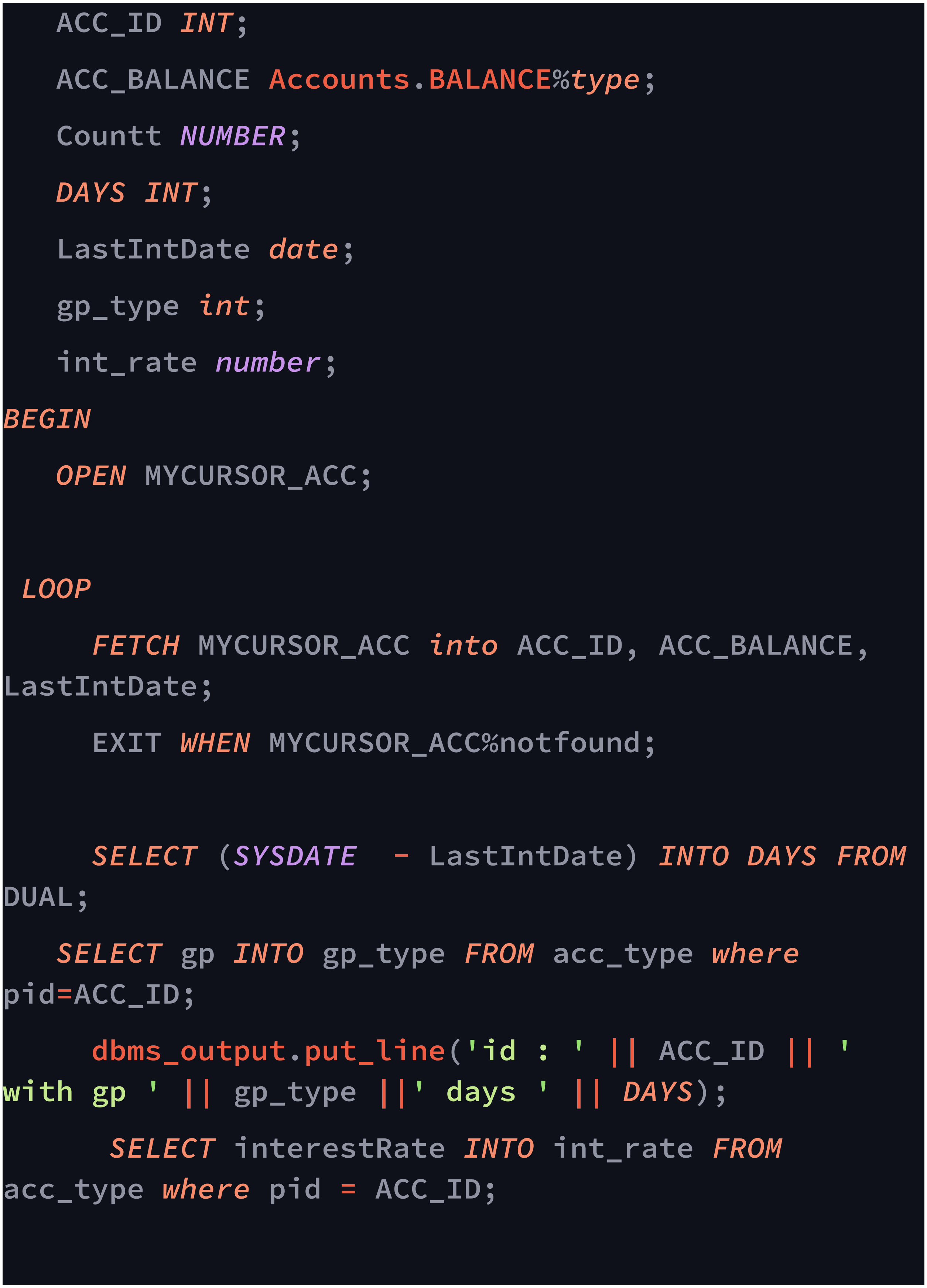
Task 2



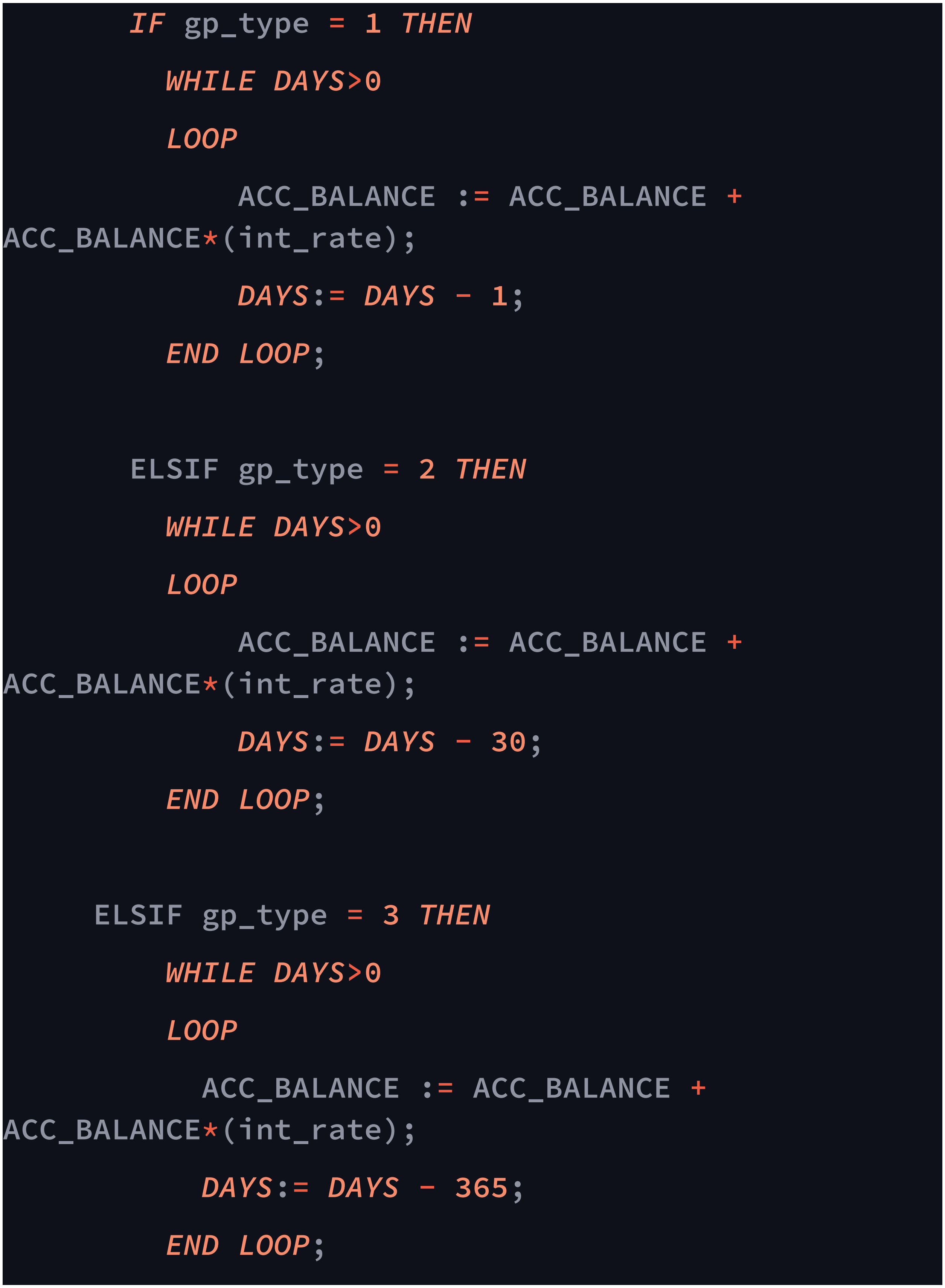
```
create table accounts(
id int,
Name varchar2(20),
Acccode int,
Balance number,
LastDateofInterest date,
constraint accounts_pk primary key(id)
);

create table acc_type(
pid int,
Name varchar2(20),
interestRate number,
gp int,
constraint ap_fk foreign key(pid) references
accounts(id)
);
```



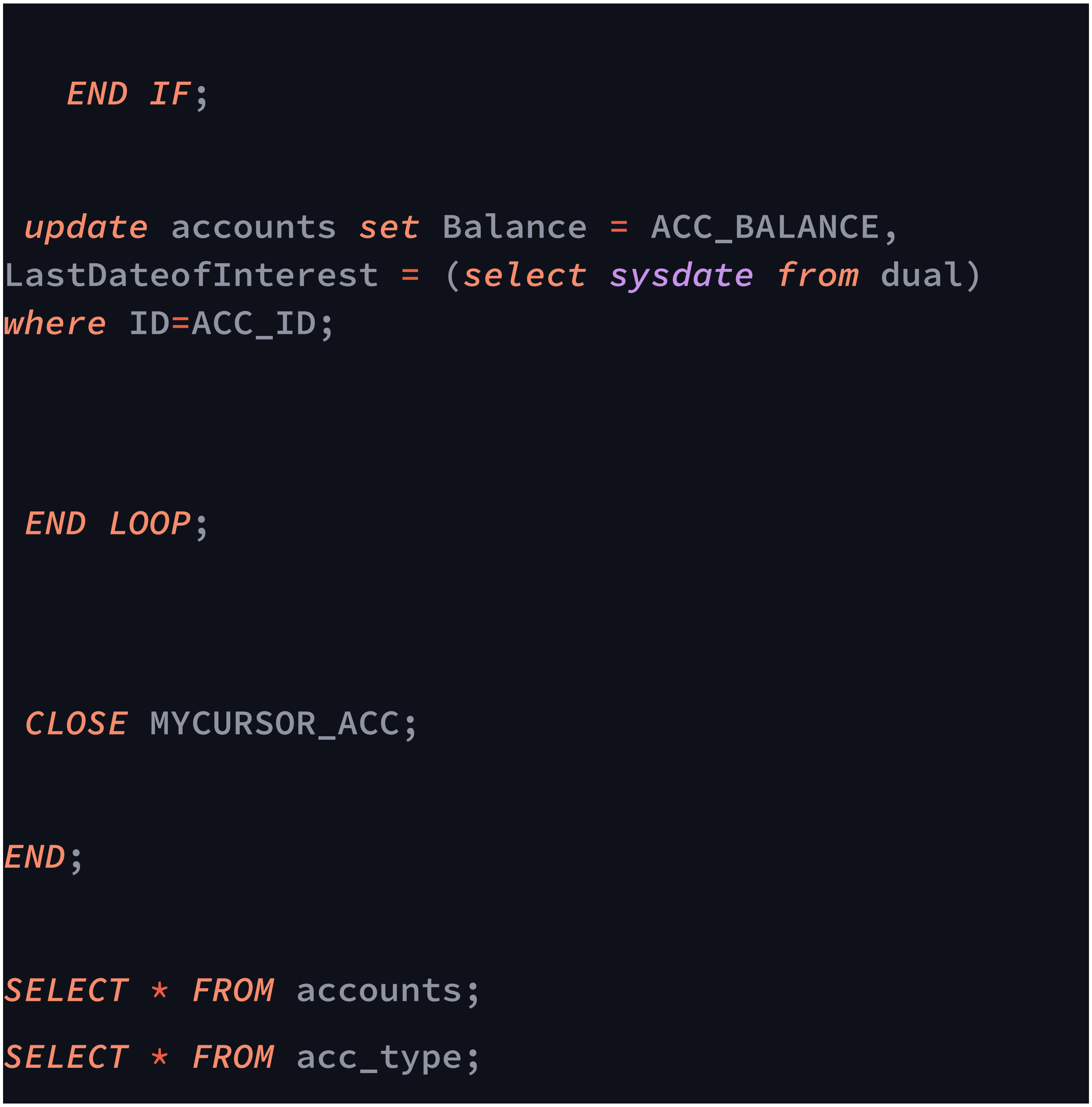
```
DELETE FROM acc_type;
DELETE FROM accounts;

insert into accounts values(1,'a',101,500,DATE
'2022-11-03');
insert into accounts values(2,'b',102,1500,DATE
'2022-5-13');
insert into accounts values(3,'c',103,3500,DATE
'2020-9-13');

insert into acc_type values(1,'a',0.1,1);
insert into acc_type values(2,'b',.2,2);
insert into acc_type values(3,'c',.3,3);






SELECT * FROM accounts;

DECLARE
   CURSOR MYCURSOR_ACC IS SELECT id, Balance,
LastDateofInterest FROM accounts;
```



```
   ACC_ID INT;
   ACC_BALANCE Accounts.BALANCE%type;
   Countt NUMBER;
   DAYS INT;
   LastIntDate date;
   gp_type int;
   int_rate number;
BEGIN
   OPEN MYCURSOR_ACC;

 LOOP
     FETCH MYCURSOR_ACC into ACC_ID, ACC_BALANCE,
LastIntDate;
     EXIT WHEN MYCURSOR_ACC%notfound;

     SELECT (SYSDATE  - LastIntDate) INTO DAYS FROM
DUAL;
   SELECT gp INTO gp_type FROM acc_type where
pid=ACC_ID;
     dbms_output.put_line('id : ' || ACC_ID || '
with gp ' || gp_type ||' days ' || DAYS);
      SELECT interestRate INTO int_rate FROM
acc_type where pid = ACC_ID;
```



```
       IF gp_type = 1 THEN
         WHILE DAYS>0
         LOOP
             ACC_BALANCE := ACC_BALANCE +
ACC_BALANCE*(int_rate);
             DAYS:= DAYS - 1;
         END LOOP;

       ELSIF gp_type = 2 THEN
         WHILE DAYS>0
         LOOP
             ACC_BALANCE := ACC_BALANCE +
ACC_BALANCE*(int_rate);
             DAYS:= DAYS - 30;
         END LOOP;

     ELSIF gp_type = 3 THEN
         WHILE DAYS>0
         LOOP
           ACC_BALANCE := ACC_BALANCE +
ACC_BALANCE*(int_rate);
           DAYS:= DAYS - 365;
         END LOOP;
```



```
   END IF;


 update accounts set Balance = ACC_BALANCE,
LastDateofInterest = (select sysdate from dual)
where ID=ACC_ID;



 END LOOP;



 CLOSE MYCURSOR_ACC;


END;


SELECT * FROM accounts;
SELECT * FROM acc_type;
```