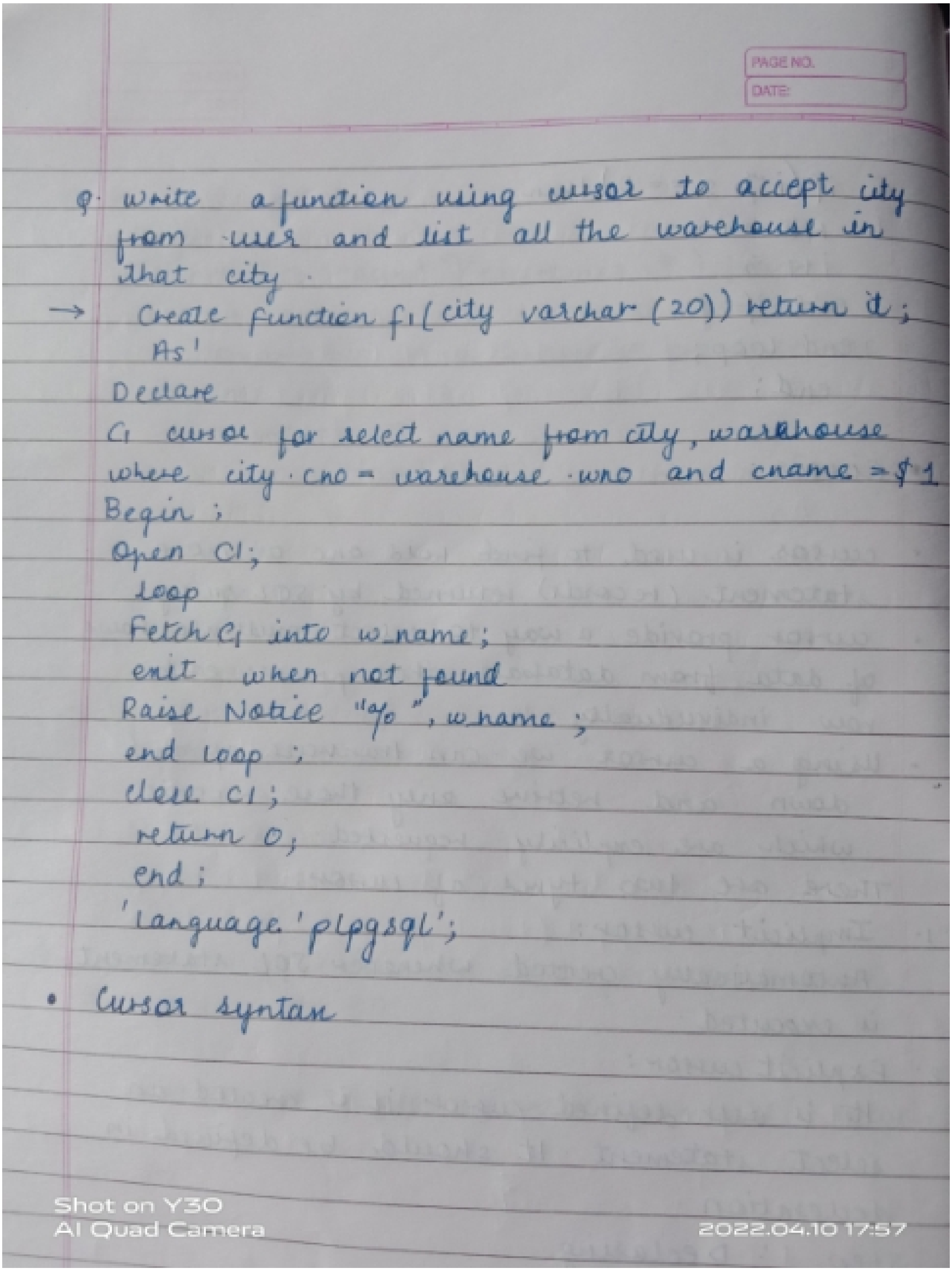


Q. write a function using cursor to accept city from user and list all the warehouse in that city.

```
→ Create function f1(city varchar(20)) return it;
   As'
Declare
C1 cursor for select name from city, warehouse
where city.cno = warehouse.wno and cname = $1
Begin;
Open C1;
 loop
 Fetch C1 into w_name;
 exit when not found
 Raise Notice "%", w_name;
 end loop;
 close C1;
 return 0;
 end;
 'Language 'plpgsql';
```

- Cursor syntax

PAGE NO.
DATE:

```
  if(i % 2 == 0) then
  return i;
  i++ ;
 end if;
 end loop;
  end;
```

- Cursor :

- cursor is used to ~~find~~ hold one or more statement (records) returned by SQL query
- cursor provide a way to select multiple rows of data from database then process each row indivedually.
- Using a cursor we can tranverse up and down and retrive only those records which are explicity requested.

There are two types of cursor:

1. Implicit cursor:
Automaticaly created whenever SQL statement is executed.

2. Explicit cursor:
It is user defined cursor it is created on select statement. It should be defined in declaration.

step 1: Declaring
2: Opening
3: Fetching
4: Closing.

```
        count int ;
    Begin
     select into count *(customer.*) from
     branch , customer where branch.bid =
     customer.bid and ~~branch~~ branch name =
     bname
      if count == 0
    Raise exception "Invalid branch name" ~~exit~~;
     exit ;
      end if ;
     return cnt;
      end;
```

- syntax for loop:

```
     while (condition)
      loop
      statements ;
     end loop;
```

Q. write a function to display all even numbers between 1 to 50

```
     Create or Replace Function numbers() return
      int
     As
    Declare
          i int = 1;
    Begin
      While (i< =50)
    ~~while~~ ~~condition~~ Loop.
```

PAGE NO.
DATE:

- Exception level calls transcation to be terminated

Q. write a function to increase loan amt approve by 20% if ~~essential~~ inital loan amt approve is less than 100000/- then print msg or notice as loan amt is less then 1 lakh.

Ans:
```
    Create Function F1() return int
    As
    Declare
        amt int;
    Begin
      select into amt Lamtapp from loan;
        if (amt <100000)
    Raise Notice "Amount is less than 100000";
      exit;
      end if;
      update loan set ~~loan amt~~ Lamtapp =
       lamtapp+ (Lamtapp*0.2) from Loan;
   Return 0;
   end.
```

Q. Write a function to print total number of customer of a particular branch ~~expect~~ accept branch name as input incase branch name is invalid raise exception for the same.
```
    Create Function Branch(bname varchar)
    return int
   As
   Declare
```

PAGE NO.
DATE:

Handling Errors and Exceptions in Function.

Raise statement used to send errors and expeptions during a SQL function execution raise statement conist of two things i.e

i Level ii) string (msg)

- Level is of three types ~~to~~ debug, notice, exception
- To insert variable in error msg we use percent sign as placeholder

a. Debug Level:
It sends specifed text as debug msg to database

b. Notice Level:
It will send specified text as notice.

c. Exception level:
It will send specified text as an error or exception.

- Syntax:
Raise Level " string ";

- Example:

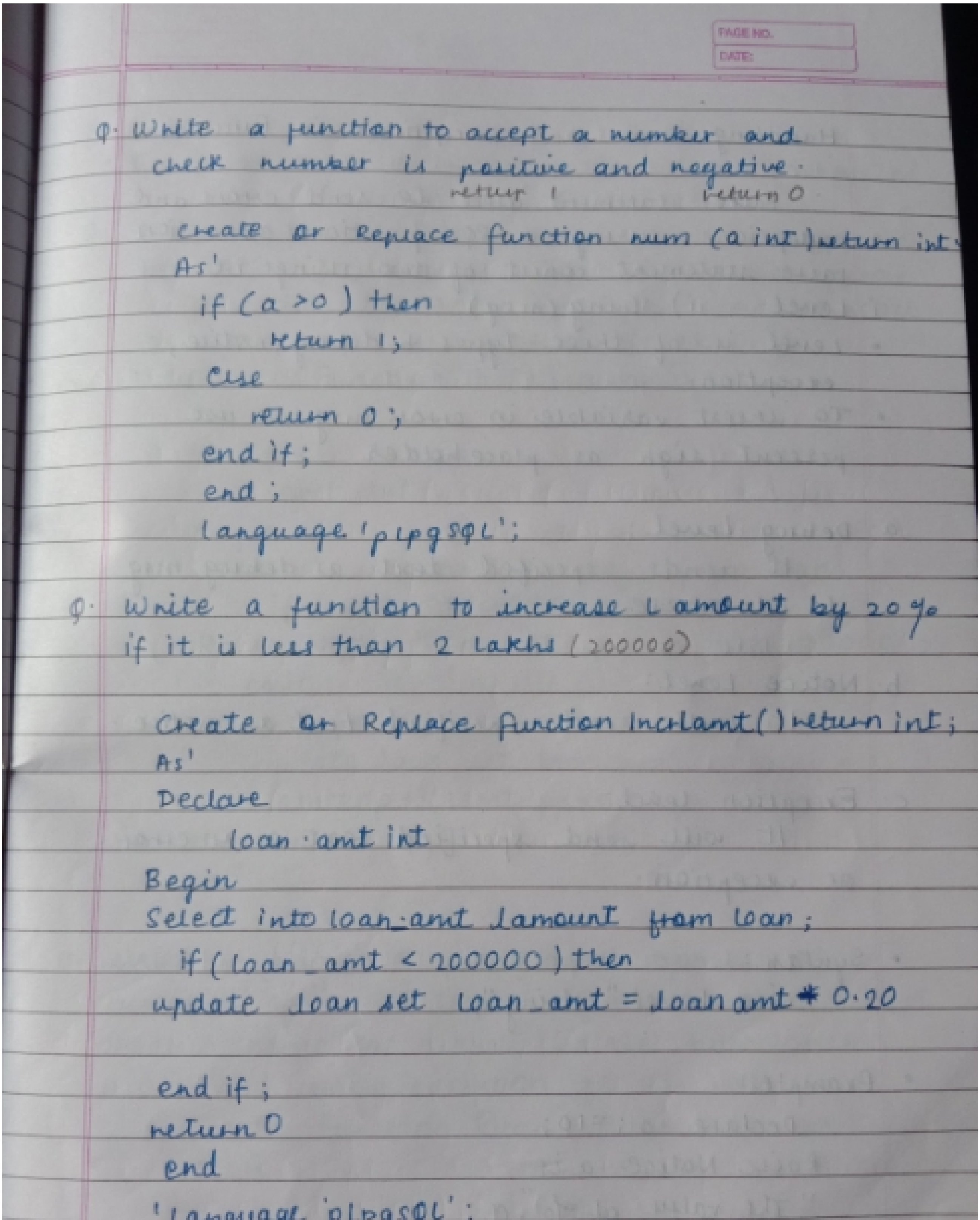
```
    Declare a;=10;
    Raise Notice a is
    "The value of %", a;
```

• Enc

Q. W

Ans.

Q.



Q. Write a function to accept a number and check number is positive and negative.
return 1 return 0

```
create or Replace function num (a int) return int;
As'
  if (a >0) then
     return 1;
  else
    return 0;
  end if;
  end;
  language 'plpgSQL';
```

Q. Write a function to increase L amount by 20% if it is less than 2 Lakhs (200000)

```
Create or Replace function IncrLamt() return int;
As'
Declare
    loan.amt int
Begin
Select into loan_amt Lamount from loan;
  if (loan_amt < 200000) then
update loan set loan_amt = loanamt * 0.20

end if;
return 0
end
'language 'plpgSQL';
```

```
Begin
  if (a >b ) then
  return a ;
 else
   return b;
 end if ;
  end ;
 'Language 'plpgsqL';
```

Q. Write a function to accept a number check wheather the number in even or odd. If no. is even return 1 if no. is odd return 0.

```
Create or Replace function num (a int , ~~b int~~)
  return int;
  As'
            a
  if ( num % 2 = = 0 ) then
   return 1;
 else
   return 0;
 end if;
 end ;
 'Language' plpgsqL';
```

Q. Write a function to accept city name as an input and display number of ware houses in that city (One city many ware house)

```
  Create  or Replace Function Display
  count (cityname varchar(20)) returns int
  As'
   Declare
       cnt int
   Begin
    select into cnt (count (warehouse.*) from
   city, warehouse where city.no = warehouse.cno
  and cname = city name;
   end;
  'Language 'plpgSQL';
```

- Conditional Statement.

```
Syntax: if else
          if (condition) then
            —
            —
            —
          else
         end if;
```

Q. Write a function to accept two numbers from a user and display maximum number

```
  Create or Replace Function Display maximum
  ~~no int~~ (a int, bint) return int :
  As'
```

Q. Write a Function to display number of customers of a particular branch. accept branch name as input relation is one branch many customers

ans count number of customers from shivaji nagar branch

select count (customer.*) from customer, Branch where
Branch. bid = customer. bid and bname = 'Shivaji Nagar'

- Create or Replace Function Display count (brname varchar(20)) returns int

select into
↓
result of
record or
a row in
a variable

```
As'
Declare
cnt int;
Begin
Select int cnt [count (customer*) from customer.
Branch where Branch. bid = customer. bid and bname =
brname;
return cnt;
end;
'Language 'PlpgSQL'.
```

PAGE NO.

DATE:

with argument

```
Create or replace function addnum() returns int
As'
 Declare
        result int;
        a int:= 2;
        b int:= 3;
 Begin
      result:= a+b;
        return result;
 End;
  'Language'plpgSQL';
```

Q. Table

Student (sno, sname, class)
Teacher (tno, tname, experience)
relation Many to many with descriptive attribute sname

Create a view to find teacher name teaching maximum no. of subjects.

PAGE NO.
DATE

a an object create function command is use to create stored function, drop function command is used to delete function

- Syntax: optional keyword use to edit or replace existing fun

```
Create 'or Replace' function fun_name (arguments)
                                 user define      optional
returns data type As'                          variable data
Declare                                                 type
variables [ ___;                                Attribute / col-name
            ___;        Add (int a; int b)
            ___;          Add ()
Begin ___ expression
          statement
code      [ ___;
block       ___;
            ___;
prog/logic end ___
   /
nested code   ;
              ;
              ;
'Language 'plpgsql';
function must return single code
```

If we want to return more than one value we have to user cursor within the function

Q. write a program to add two numbers

without arguement

```
Create or replace function Addnum() returns int As'
Declare
        resultint;
Begin
        result : = 2+3;
          return result;
end;
'Language 'plpgsql';
```

PAGE NO.

DATE:

5 List names of different subjects taught by each teacher.

→ Create view teacher_view As
select ~~subject~~ distinct, tname n subjed
from teacher
group by ~~teacher~~ tname;

- STORED FUNCTIONS:

Function is a block of statements to complete a particular task
eg: Add, subtract, main, printf, scanf, etc

- variable:

Variable is a container or a name given to memory location which holds some value. value can be of different types to identify or specify which type of value it is we use datatypes
eg: int, varchar, char, double, type, float etc
(declaration: roll int) (callen: assigned sepration)
(Intialization: rollno int :=)

i Variable must be declared before being used ~~to declare~~

ii To declare variable declare keyword is used

iii Variable may be intializzed with some value otherwise it gets intialize to null type

- STORED FUNCTION:

It is a user define function. It is named is stored function because their are stored in a DB

PAGE NO.
DATE:

1. Display total marks ~~for~~ of student for each class

→ Create view student_view As.

```
    select sum(marks)
    from student group by class;
```

2. Display total strength of each class

→ Create view student_view As

```
   select count (rno)
   from student group by class;
```

3. Display details of student learning subject DBMS by teacher Veena Gandhi

→ Create view student_view As

```
    select student . *
  from student , teacher
  where teacher.tno =student.tno and tname=
'Veena Gandhi';
```

4. Display total no. of students taught by each teacher.

→ Create ~~stu~~ view student_view As

```
   select student . * count (rno), tname
  from student, teacher
 where teacher.tno = student.tno
 group by tname;
```

3. Display details of all routes on which bus no 10 is running.

```
→ Create view third As
Select route * from route, bus ternary where route.rno
= ternary.rno and bus.bno = ternary.bno and bus.bno = 10;
```

4. Find out driver name working in both shift.

```
→ Create view fourth As
Select dname from driver, ternary where driver.dno
= ternary.dno and shift in(1.2);

    // and shift = 1 and shift = 2;
```

5. Display names of driver having maximum salary.

```
→ Create view fifth As
Select dname from driver where salary =
(select max (salary) from driver);
```

6. Display details of bus whos capacity is less than 30.

```
→ Create view Bus_view As
Select Bus *
From Bus
where Capacity < 30;
```

Q. Student (sno, name, marks, percentage, class)
teacher (tno, tname, subject)
Realtion one to many

where loan_app.lno = ternary.lno and,
branch, bid = ternary.bid and bname = 'shivaji nagar';

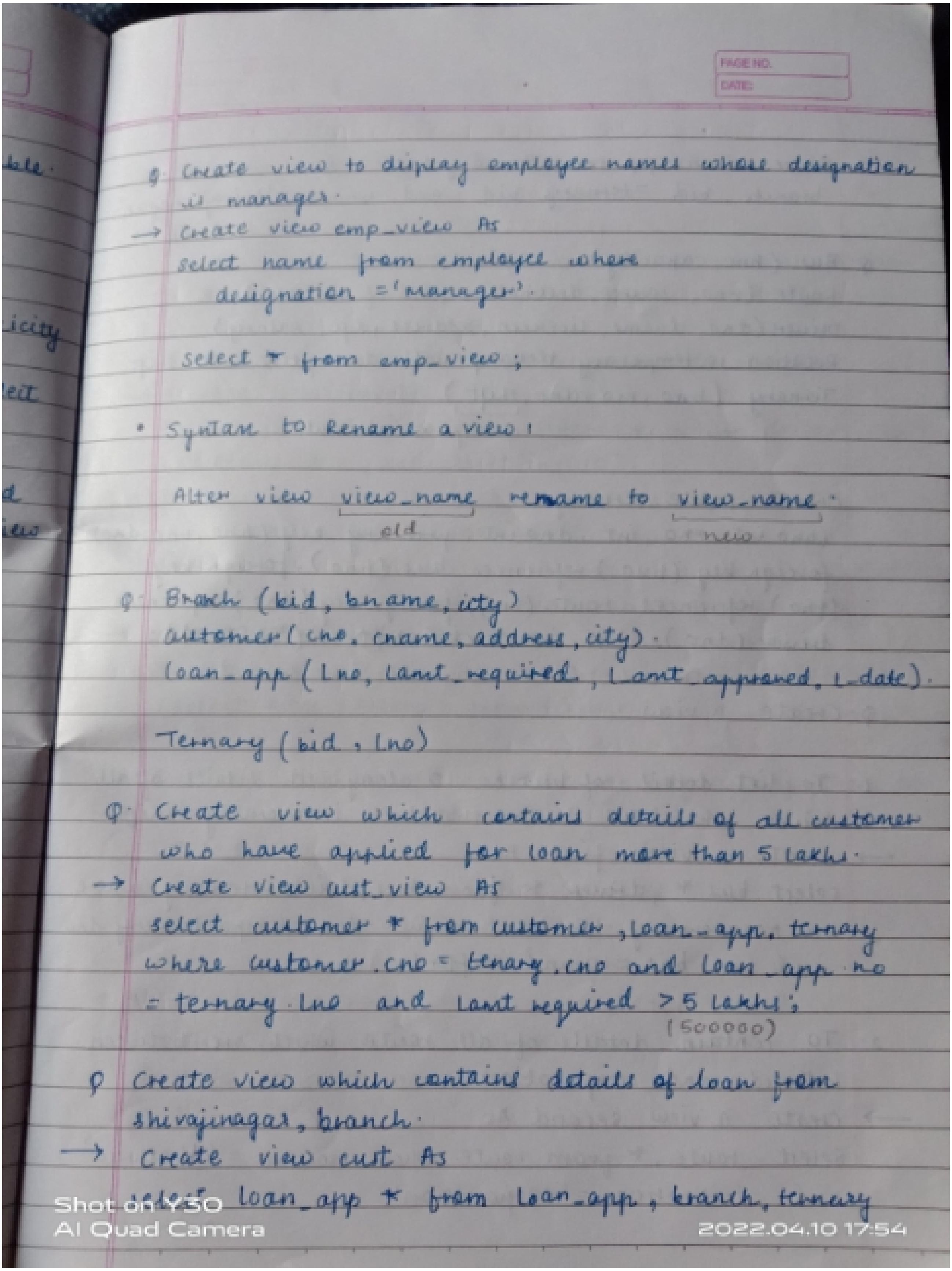
Q. Bus (bno, capacity)
Route (rno, source, destination, no of stations)
Driver (dno, dname, licensen, address, age, salary)
Relation is ~~temporary~~ ternary with descriptive alt shift
Ternary (bno, rno, dno, $\underbrace{\text{shift}}_{\text{descriptive attribute}}$)

Create table ternary
(bno int, rno, int, dno int, primary key (bno, rno, dno)
foreign key (bno) references bus (bno), foreign key
(rno) references route (rno), foreign key (dno) references
driver (dno). Shift int check (shift in ("1","2"));

Q. Create a view

1. To list detail of bus no. 10 along with details of all
driver who have driven that bus in morning shift
→ Create a view first As
select bus *, driver * from bus, driver, ternary where
bus.no = ternary.bno and driver.dno = ~~tempo~~ ternary.dno
and bus.bno = 10 and shift = 1;

2. To contain details of all route which are between
kothrud and corporation.
→ Create a view second As
Select route.* from route where source = 'Kothrud'
and destination = 'corporation'.



ble.

icity

ect

d

iew

PAGE NO.

DATE:

Q. Create view to display employee names whose designation is manager.

→ Create view emp_view As
select name from employee where
designation ='manager'.

select * from emp_view ;

- Syntax to Rename a view :

Alter view $\underbrace{\text{view\_name}}_{\text{old}}$ rename to $\underbrace{\text{view\_name}}_{\text{new}}$.

Q. Branch (bid, bname, icty)
customer (cno, cname, address, city).
Loan_app (Lno, Lamt_required, Lamt_approved, L_date).

Ternary (bid , Lno)

Q. Create view which contains details of all customer who have applied for loan more than 5 lakhs.

→ Create view cust_view As
select customer * from customer, Loan_app, ternary
where customer.cno = tenary.cno and loan_app no
= ternary.Lno and Lamt required > 5 Lakhs;
(500000)

Q Create view which contains details of loan from shivajinagar, branch.

→ Create view cust As
select loan_app * from Loan_app, branch, ternary

is a subset of real table but it is not a real table.

- uses of View:

i Restricting data access: view provide additional level of security by restricting access to original table.

ii Hiding data complexcity: A view can hide complexicity that exists in a multiple table join.

iii Simplify commands for user: View allows users to select information from multiple tables without performing join.

iv store complex query: Queries which are complex and required many numbers of time can be stored in a view

- Syntax:

```
CREATE VIEW: view_name As select
col1, col2, ..... from table(s) where condition,
```

view_name - user defined name.

As - ~~keyboard~~ key word

* Emp(id, name, designation, salary)

Q. Create a view to display id, name, salary of employee

→ Create view emp_view As
Select id, name, salary from employee,
Displaying created table
Select * from emp_view;

Advance Database Management System.

UNITS:

1. Relational database system [practical based]
2. Transaction concepts
3. Concurrency control
4. Crash recovery
5. Database security
6. Database system architenture

- UNIT I : Relational database system.

i Language structure
ii Controlling program flow : a> conditional statements
b> Loops
i*c> View
ii*d> Functions
e> Handling cursor and exception
iii f> Cursor
g> Trigger

- VIEW :

Views are virtual table or temporary table it does not store any data in memory.

You can create view from a table or more than one table.

A view can be created from another view. A view is