

Sequence:

1. Create sequence student_id_sequence
Increment by 1
Start with 100
Minvalue 099
Maxvalue 200
No cycle;
2. Create sequence teacher_id_sequence
Increment by 1
Start with 1
Minvalue 0
Maxvalue 100
No cycle;

View:

1. Create or replace view student_information
As
Select student.s_id, student.s_name, student.blood_group, student.s_phone, location.city,
location.country from student, location
Where student.subcode_num=location.subcode_num;
2. Create or replace view teacher_information
As
Select teacher.t_id, teacher.t_name, teacher.designation, location.city, location.country from
teacher, location
Where teacher.course_id=course.course_id;
And course.code_num=location.code_num;

1. Create or replace function total_teacher(teacher teacher.t_id%type)
Return teacher.t_id%type
Is
t_id student.t_id%type
Begin
Select count(*) into t_id from teacher
Return t_id
End;
2. Create or replace function total_staff(staff admin.staffs_id%type)
Return admin.staffs_id%type
Is
Staff_id admin.staffs_id%type
Begin
Select count(*) into staffs_id from admin
Return staffs_id
End;
3. Create or replace procedure insert_student(s_id in varchar2, s_name in varchar2)
Is
Begin
Insert into insert_student values(s_id, s_name)
End;
4. Create or replace procedure insert_teacher(t_id in varchar2, t_name in varchar2)
Is
Begin
Insert into insert_teacher values(t_id, t_name)
End;

5. Create or replace staff_details
Is
Cursor c
Is
Select staffs_name,job from admin;
X admin.staffs_id%type
Begin
For x in c
Loop
Dbms_output.put_line(c.staffs_name||' '||c.job);
End loop;
End;

Trigger:

1. Create or replace trigger syllabus_change
Before insert or update or delete on course
For each row
Begin
If syllabus in ('Chapter 1-3', 'Chapter1-2', 'Chapter 1-4');
And class_name='Six'
Then
Raise_application_error(-8888,'invalid');
End if;
2. Create or replace trigger chairman_name_change
Before update or delete on admin
For each roe
Begin
If staffs_name='Ford' and job='Chairman'
Then
Raise_application_error(-8888,'Cannot be changed');
End if;

3. Create or replace trigger teacher_id_change
Before update on teacher
For each row
Begin
If t_id not between '01' and '100'
Then
Raise_application_error(-8888,'Cannot be changed');
End if;
4. Create or replace trigger student_id_change
Before update on student
Begin
If s_id not between '000' and '200'
Then
Raise_application_error(-8888,'Cannot be changed');
End if;

Package:

1. Create package student_info as procedure insert_student(s_id student.s-id%type, s_name student.s_name%type, loc location.city%type)
is
begin
insert into insert_student values (s_id,s_name,loc);
end insert_student(s_id,s_name,loc);
end student_info;
end;