## Prior Instructions

---

- **Please do read all the questions before performing any operations in the database**
- **Once you have fully gone through the questions then likewise decide the contents and table columns and follow the below instructions**

---

1. Create Student Database
2. Create the following table under the Student Database:
    a. StudentBasicInformation
        i. Columns
            1. StudentName
            2. StudentSurname
            3. StudentRollNo
            4. StudentAddress
            5. Add more three basic columns of the name of your own
    b. StudentAdmissionPaymentDetails
        i. Columns
            1. StudentRollNo
            2. AmountPaid
            3. AmountBalance
            4. Add more four basic columns of the name of your own
    c. StudentSubjectInformation
        i. Columns
            1. SubjectOpted
            2. StudentRollNo
            3. SubjectTotalMarks
            4. SubjectObtainedMarks
            5. StudentMarksPercentage
            6. Add more one columns of the name of your own
    d. SubjectScholarshipInformation
        i. Columns
            1. StudentRollNo
            2. ScholarshipName
            3. ScholarshipDescription
            4. ScholarshipAmount
            5. ScholarshipCategory
            6. Add more two columns of the name of your own
3. Insert more than 10 records in each and every table created
4. Snap of the all the tables once the insertion is completed

Worksheet | Query Builder

```
1  select * from studentbasicinformation;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.11 seconds

| | STUDENTNAME | STUDENTSURNAME | STUDENTROLLNO | STUDENTADDRESS | AGE | PRN | PHNO |
|---|---|---|---|---|---|---|---|
| 1 | a | a | 1 | address1 | 10 | aa1 | 12345 |
| 2 | b | b | 2 | address2 | 2 | aa2 | 123456 |
| 3 | c | c | 3 | address3 | 3 | cc3 | 1234567 |
| 4 | d | d | 4 | address4 | 4 | dd4 | 12345678 |
| 5 | e | e | 5 | address5 | 5 | ee5 | 22345678 |
| 6 | f | f | 6 | abcd xyz6 | 6 | abc6 | 12345 |
| 7 | g | g | 7 | abcd xyz7 | 7 | abc7 | 23345 |
| 8 | h | h | 8 | abcd xyz8 | 8 | abc8 | 122122 |
| 9 | i | i | 9 | abcd xyz9 | 9 | abc9 | 564753 |
| 10 | j | j | 10 | abcd xyz10 | 10 | abc10 | 657654 |

```
1  select * from studentadmissionpaymentdetails;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.004 seconds

| | STUDENTROLLNO | AMOUNTPAID | AMOUNTBALANCE | ADMISSIONYEAR | GRADUATIONYEAR | CURRENTYEAR | COLLEGE |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 8000 | 2000 | 2019 | 2023 | 3 | dypcoe |
| 2 | 2 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 3 | 3 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 4 | 4 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 5 | 5 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 6 | 6 | 2000 | 8000 | 2018 | 2022 | 4 | dypcoe |
| 7 | 7 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 8 | 8 | 0 | 10000 | 2022 | 2026 | 1 | dypcoe |
| 9 | 9 | 4000 | 6000 | 2021 | 2025 | 2 | dypcoe |
| 10 | 10 | 7000 | 3000 | 2018 | 2022 | 4 | dypcoe |

Worksheet | Query Builder

```
1  select * from studentsubjectinformation;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.004 seconds

| | SUBJECTOPTED | STUDENTROLLNO | SUBJECTTOTALMARKS | SUBJECTOBTAINEDMARKS | STUDENTMARKSPERCENTAGE |
|---|---|---|---|---|---|
| 1 | science | 1 | 100 | 100 | 100 |
| 2 | science | 2 | 100 | 100 | 100 |
| 3 | commerce | 3 | 100 | 90 | 90 |
| 4 | arts | 4 | 100 | 80 | 80 |
| 5 | science | 5 | 100 | 60 | 60 |
| 6 | arts | 6 | 100 | 70 | 70 |
| 7 | arts | 7 | 100 | 70 | 70 |
| 8 | science | 8 | 100 | 80 | 80 |
| 9 | commerce | 9 | 100 | 90 | 90 |
| 10 | commerce | 10 | 100 | 95 | 95 |

```
select * from subjectscholarshipinformation;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.003 seconds

| | STUDENTROLLNO | SCOLARSHIPNAME | SCOLARSHIPDESCRIPTION | SCOLARSHIPAMOUNT | SCOLARSHIPCATEGORY |
|---|---|---|---|---|---|
| 1 | 1 | yojna | - | 10000 | finance |
| 2 | 2 | yojna | - | 10000 | finance |
| 3 | 3 | yojna | - | 10000 | finance |
| 4 | 4 | yojna | - | 10000 | finance |
| 5 | 5 | yojna | - | 10000 | finance |
| 6 | 6 | yojna | - | 10000 | finance |
| 7 | 7 | yojna | - | 10000 | finance |
| 8 | 8 | yojna | - | 10000 | finance |
| 9 | 9 | yojna | - | 10000 | finance |
| 10 | 10 | yojna | - | 10000 | finance |

5. Update any 5 records of your choice in any table like update the StudentAddress with some other address content and likewise so on with any records of any table of your choice
6. Snap of the all the tables post updation

```
select * from studentbasicinformation;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.004 seconds

| | STUDENTNAME | STUDENTSURNAME | STUDENTROLLNO | STUDENTADDRESS | AGE | PRN | PHNO |
|---|---|---|---|---|---|---|---|
| 1 | a | a | 1 | address1 | 10 | aa1 | 12345 |
| 2 | b | b | 2 | address2 | 2 | aa2 | 123456 |
| 3 | c | c | 3 | address3 | 3 | cc3 | 1234567 |
| 4 | d | d | 4 | address4 | 4 | dd4 | 12345678 |
| 5 | e | e | 5 | address5 | 5 | ee5 | 22345678 |
| 6 | f | f | 6 | abcd xyz6 | 6 | ff6 | 12345 |
| 7 | q | q | 7 | abcd xyz7 | 7 | qq7 | 23345 |
| 8 | h | h | 8 | abcd xyz8 | 8 | abc8 | 122122 |
| 9 | i | i | 9 | abcd xyz9 | 9 | abc9 | 564753 |
| 10 | j | j | 10 | abcd xyz10 | 10 | abc10 | 657654 |

Worksheet | Query Builder

```
1  select * from studentadmissionpaymentdetails;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.003 seconds

| | STUDENTROLLNO | AMOUNTPAID | AMOUNTBALANCE | ADMISSIONYEAR | GRADUATIONYEAR | CURRENTYEAR | COLLEGE |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 8000 | 2000 | 2019 | 2023 | 3 | dypcoe |
| 2 | 2 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 3 | 3 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 4 | 4 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 5 | 5 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 6 | 6 | 2000 | 8000 | 2018 | 2022 | 4 | dypcoe |
| 7 | 7 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 8 | 8 | 0 | 10000 | 2022 | 2026 | 1 | dypcoe |
| 9 | 9 | 4000 | 6000 | 2021 | 2025 | 2 | dypcoe |
| 10 | 10 | 7000 | 3000 | 2018 | 2022 | 4 | collleqe1 |

Worksheet | Query Builder

```
1  select * from studentsubjectinformation;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.003 seconds

| | SUBJECTOPTED | STUDENTROLLNO | SUBJECTTOTALMARKS | SUBJECTOBTAINEDMARKS | STUDENTMARKSPERCENTAGE |
|---|---|---|---|---|---|
| 1 | arts | 1 | 100 | 100 | 100 |
| 2 | science | 2 | 100 | 100 | 100 |
| 3 | commerce | 3 | 100 | 90 | 90 |
| 4 | arts | 4 | 100 | 80 | 80 |
| 5 | science | 5 | 100 | 60 | 60 |
| 6 | arts | 6 | 100 | 70 | 70 |
| 7 | arts | 7 | 100 | 70 | 70 |
| 8 | science | 8 | 100 | 80 | 80 |
| 9 | commerce | 9 | 100 | 90 | 90 |
| 10 | commerce | 10 | 100 | 95 | 95 |

```
1  select * from subjectscholarshipinformation;
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 × | Query Result 3 ×

SQL | All Rows Fetched: 10 in 0.003 seconds

| | STUDENTROLLNO | SCOLARSHIPNAME | SCOLARSHIPDESCRIPTION | SCOLARSHIPAMOUNT | SCOLARSHIPCATEGORY | HAS_GOT |
|---|---|---|---|---|---|---|
| 1 | 1 | yojna | - | 10000 | finance | yes |
| 2 | 2 | yojna | college lvl | 10000 | finance | yes |
| 3 | 3 | yojna | - | 10000 | finance | no |
| 4 | 4 | yojna | - | 10000 | finance | no |
| 5 | 5 | yojna | - | 10000 | finance | no |
| 6 | 6 | yojna | - | 10000 | finance | yes |
| 7 | 7 | yojna | - | 10000 | finance | no |
| 8 | 8 | yojna | - | 10000 | finance | yes |
| 9 | 9 | yojna | - | 10000 | finance | yes |
| 10 | 10 | yojna | - | 10000 | finance | yes |

7. Select the student details records who has received the scholarship more than 5000Rs/-

```
1  select * from subjectscholarshipinformation where scolarshipamount>5000;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 10 in 0.004 seconds

| | STUDENTROLLNO | SCOLARSHIPNAME | SCOLARSHIPDESCRIPTION | SCOLARSHIPAMOUNT | SCOLARSHIPCATEGORY |
|---|---|---|---|---|---|
| 1 | 1 | yojna | - | 10000 | finance |
| 2 | 2 | yojna | college lvl | 10000 | finance |
| 3 | 3 | yojna | - | 10000 | finance |
| 4 | 4 | yojna | - | 10000 | finance |
| 5 | 5 | yojna | - | 10000 | finance |
| 6 | 6 | yojna | - | 10000 | finance |
| 7 | 7 | yojna | - | 10000 | finance |
| 8 | 8 | yojna | - | 10000 | finance |
| 9 | 9 | yojna | - | 10000 | finance |
| 10 | 10 | yojna | - | 10000 | finance |

8. Select the students who opted for scholarship but has not got the scholarship

```
select * from studentbasicinformation where studentrollno in (
select studentrollno from subjectscholarshipinformation where has_got='no'
)
;
```

Query Result | Query Result 1

SQL | All Rows Fetched: 4 in 0.005 seconds

| | STUDENTNAME | STUDENTSURNAME | STUDENTROLLNO | STUDENTADDRESS | AGE | PRN | PHNO |
|---|---|---|---|---|---|---|---|
| 1 | c | c | 3 | address3 | 3 | cc3 | 1234567 |
| 2 | d | d | 4 | address4 | 4 | dd4 | 12345678 |
| 3 | e | e | 5 | address5 | 5 | ee5 | 22345678 |
| 4 | q | q | 7 | abcd xyz7 | 7 | qq7 | 23345 |

9. Fill in data for the percentage column i.e. StudentMarksPercentage in the table StudentSubjectInformation by creating and using the stored procedure created

Worksheet | Query Builder

```
set SERVEROUT on
create OR REPLACE PROCEDURE update_percentage
as
BEGIN
  update studentsubjectinformation
  set studentmarkspercentage=(subjectobtainedmarks/subjecttotalmarks)*100;
  commit;
END;
/

call update_percentage()
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.002 seconds

| | SUBJECTOPTED | STUDENTROLLNO | SUBJECTTOTALMARKS | SUBJECTOBTAINEDMARKS | STUDENTMARKSPERCENTAGE |
|---|---|---|---|---|---|
| 1 | arts | 1 | 100 | 100 | 100 |
| 2 | science | 2 | 100 | 100 | 100 |
| 3 | commerce | 3 | 100 | 90 | 90 |
| 4 | arts | 4 | 100 | 80 | 80 |
| 5 | science | 5 | 100 | 60 | 60 |
| 6 | arts | 6 | 100 | 70 | 70 |
| 7 | arts | 7 | 100 | 70 | 70 |
| 8 | science | 8 | 100 | 80 | 80 |
| 9 | commerce | 9 | 100 | 90 | 90 |
| 10 | commerce | 10 | 100 | 95 | 95 |

10. Decide the category of the scholarship depending upon the marks/percentage obtained by the student and likewise update the ScholarshipCategory column, create a stored procedure in order to handle this operation

11. Create the View which shows balance amount to be paid by the student along with the student detailed information (use join)

```
select sa.studentrollno,sb.studentname,sb.studentsurname,sb.prn,sa.amountbalance from studentadmissionpaymentdetails sa
join studentbasicinformation sb
on sa.studentrollno=sb.studentrollno ;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.004 seconds

| | STUDENTROLLNO | STUDENTNAME | STUDENTSURNAME | PRN | AMOUNTBALANCE |
|---|---|---|---|---|---|
| 1 | 1 | a | a | aa1 | 2000 |
| 2 | 2 | b | b | aa2 | 9000 |
| 3 | 3 | c | c | cc3 | 9000 |
| 4 | 4 | d | d | dd4 | 9000 |
| 5 | 5 | e | e | ee5 | 9000 |
| 6 | 6 | f | f | ff6 | 8000 |
| 7 | 7 | g | g | gg7 | 9000 |
| 8 | 8 | h | h | abc8 | 10000 |
| 9 | 9 | i | i | abc9 | 6000 |
| 10 | 10 | j | j | abc10 | 3000 |

12. Get the details of the students who haven't got any scholarship (use joins/subqueries)

```
select studentrollno,studentname,studentsurname,prn from studentbasicinformation where studentrollno in
(select studentrollno from subjectscholarshipinformation where has_got='no')
;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 4 in 0.005 seconds

| | STUDENTROLLNO | STUDENTNAME | STUDENTSURNAME | PRN |
|---|---|---|---|---|
| 1 | 3 | c | c | cc3 |
| 2 | 4 | d | d | dd4 |
| 3 | 5 | e | e | ee5 |
| 4 | 7 | g | g | gg7 |

13. Create Stored Procedure which will be return the amount balance to be paid by the student as per the student roll number passed through the stored procedure as the input

```
set SERVEROUTPUT ON

create or REPLACE PROCEDURE to_be_paid (roll_no int)
is
    amount int:=0;
begin
    select AMOUNTBALANCE into amount from studentadmissionpaymentdetails where studentrollno=roll_no;
    dbms_output.put_line('rno is '||roll_no||' amount balance '||amount);
end;
/

call to_be_paid(2)
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

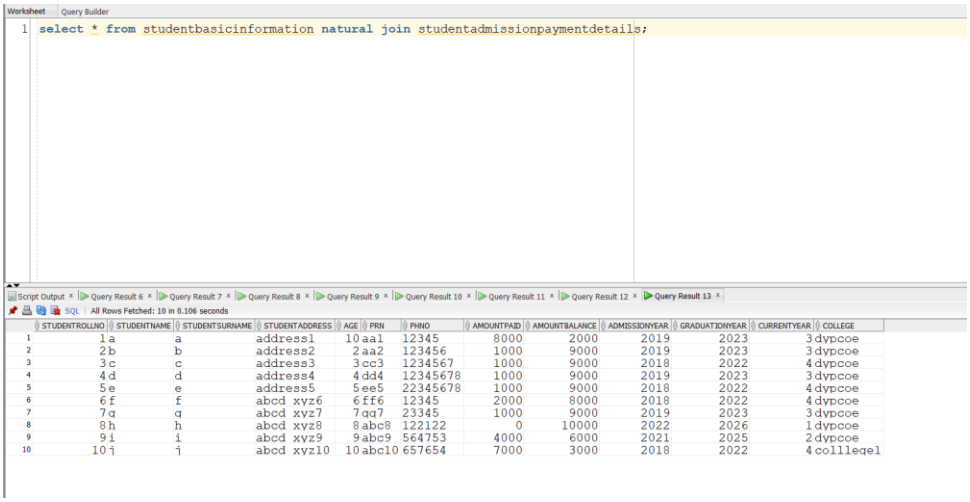
Task completed in 0.059 seconds

```
Procedure TO_BE_PAID compiled

rno is 2 amount balance 9000


Call completed.
```

14. Retrieve the top five student details as per the StudentMarksPercentage values (use subqueries)
15. Try to use all the three types of join learned today in a relevant way, and explain the same why you thought of using that particular join for your selected scenarios (try to cover relevant and real time scenarios for all the three studied joins)



```
Worksheet  Query Builder
1  select * from studentbasicinformation natural join studentadmissionpaymentdetails;
```

Script Output | Query Result 6 | Query Result 7 | Query Result 8 | Query Result 9 | Query Result 10 | Query Result 11 | Query Result 12 | Query Result 13

All Rows Fetched: 10 in 0.106 seconds

| | STUDENTROLLNO | STUDENTNAME | STUDENTSURNAME | STUDENTADDRESS | AGE | PRN | PHNO | AMOUNTPAID | AMOUNTBALANCE | ADMISSIONYEAR | GRADUATIONYEAR | CURRENTYEAR | COLLEGE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | a | a | address1 | 10 | aa1 | 12345 | 8000 | 2000 | 2019 | 2023 | 3 | dypcoe |
| 2 | 2 | b | b | address2 | 2 | aa2 | 123456 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 3 | 3 | c | c | address3 | 3 | cc3 | 1234567 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 4 | 4 | d | d | address4 | 4 | dd4 | 12345678 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 5 | 5 | e | e | address5 | 5 | ee5 | 22345678 | 1000 | 9000 | 2018 | 2022 | 4 | dypcoe |
| 6 | 6 | f | f | abcd xyz6 | 6 | ff6 | 12345 | 2000 | 8000 | 2018 | 2022 | 4 | dypcoe |
| 7 | 7 | g | g | abcd xyz7 | 7 | gg7 | 23345 | 1000 | 9000 | 2019 | 2023 | 3 | dypcoe |
| 8 | 8 | h | h | abcd xyz8 | 8 | abc8 | 122122 | 0 | 10000 | 2022 | 2026 | 1 | dypcoe |
| 9 | 9 | i | i | abcd xyz9 | 9 | abc9 | 564753 | 4000 | 6000 | 2021 | 2025 | 2 | dypcoe |
| 10 | 10 | j | j | abcd xyz10 | 10 | abc10 | 657654 | 7000 | 3000 | 2018 | 2022 | 4 | collleqe1 |

16. Mention the differences between the delete, drop and truncate commands
    Ans:

    **Delete**: this command can be used with where clause, if not will delete all tuples from table. Also, this command is slow. It is DML command
    **Drop**: this command is used to delete table along with its structure and data. It is DDL command. We can't use rollback to revert changes after this command
    **Truncate**: this is DDL command. It is used to delete entire data of table but doesn't delete table's structure. We cannot use rollback after this command.

17. Get the count of the Scholarship category which is highly been availed by the students, i.e. get the count of the total number of students corresponding to the each scholarships category

```
select count(*),scolarshipcategory from subjectscholarshipinformation group by scolarshipcategory;
```

| | COUNT(*) | SCOLARSHIPCATEGORY |
|---|---|---|
| 1 | 7 | finance |
| 2 | 3 | edu |

18. Along with the assignment no. 17 try to retrieve the maximum used scholarship category

```
select * from(
    select count(studentrollno),scolarshipcategory
    from subjectscholarshipinformation
    group by scolarshipcategory order by count(studentrollno) desc
) where ROWNUM<=1;
```

| | COUNT(STUDENTROLLNO) | SCOLARSHIPCATEGORY |
|---|---|---|
| 1 | 7 | finance |

19. Retrieve the percentage of the students along with students detailed information who has scored the highest percentage along with availing the maximum scholarship amount


20. Difference between the Triggers, Stored Procedures, Views and Functions
    Ans:

    **Triggers**: trigger gets auto-executed after certain event(create,update,delete)
    **Stored Procedure**: stored procedure can execute multiple statements. It may/may not return value.
    **Function**: same as procedure but returns value.

---

**Thank you. All The Best. Enjoy The Assignment.**