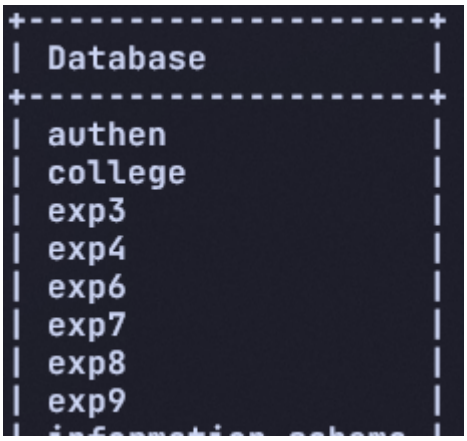
Assignment 9

```
Name : Ravish Ranjan
Course : MCA
Semester : 1st semester
```

Creating DB and Tables

Creating database

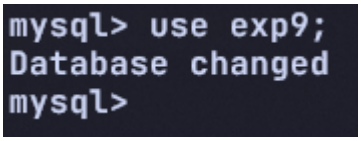
```
create database exp9;
```



```
+--------------------+
| Database           |
+--------------------+
| authen             |
| college            |
| exp3               |
| exp4               |
| exp6               |
| exp7               |
| exp8               |
| exp9               |
| information_schema |
```

using database

```
use exp9;
```



```
mysql> use exp9;
Database changed
mysql>
```

Creating Students Table

```
create table Students(
    StudentID int not null primary key,
    Name varchar(24) not null,
    Department varchar(12) not null,
    Year int not null,
    GPA Decimal(2,1)
);
```

Field	Type	Null	Key	Default	Extra
StudentID	int	NO	PRI	NULL	
Name	varchar(24)	NO		NULL	
Department	varchar(12)	NO		NULL	
Year	int	NO		NULL	
GPA	decimal(2,1)	YES		NULL	

Creating Courses Table

```
create table Courses(
    CourseID varchar(10) not null primary key,
    CourseName varchar(24) not null,
    Department varchar(12) not null,
    Credits int not null
);
```

Field	Type	Null	Key	Default	Extra
CourseID	varchar(10)	NO	PRI	NULL	
CourseName	varchar(24)	NO		NULL	
Department	varchar(12)	NO		NULL	
Credits	int	NO		NULL	

Creating Instructor Table

```
create table Instructors(
    InstructorID int not null primary key,
    Name varchar(24) not null,
    Department varchar(12) not null,
    Experience int not null
);
```

Field	Type	Null	Key	Default	Extra
InstructorID	int	NO	PRI	NULL	
Name	varchar(24)	NO		NULL	
Department	varchar(12)	NO		NULL	
Experience	int	NO		NULL	

Creating Enrollment Table

```sql
create table Enrollments(
    EnrollID int not null primary key,
    StudentID int not null,
    CourseID varchar(10) not null,
    Semester varchar(10) not null,
    Marks int not null,
    foreign key (StudentID) references Students(StudentID),
    foreign key (CourseID) references Courses(CourseID)
);
```

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| EnrollID  | int         | NO   | PRI | NULL    |       |
| StudentID | int         | NO   | MUL | NULL    |       |
| CourseID  | varchar(10) | NO   | MUL | NULL    |       |
| Semester  | varchar(10) | NO   |     | NULL    |       |
| Marks     | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Creating Course_Assignment Table

```sql
create table Course_Assignment(
    CourseID varchar(10) not null,
    InstructorID int not null,
    Year int not null,
    foreign key (CourseID) references Courses(CourseID),
    foreign key (InstructorID) references Instructors(InstructorID)
);
```

```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| CourseID     | varchar(10) | NO   | MUL | NULL    |       |
| InstructorID | int         | NO   | MUL | NULL    |       |
| Year         | int         | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

Inserting data in Students table

```sql
insert into Students (StudentID,Name,Department,Year,GPA) values
    (101, 'Aarav Sharma', 'CSE', 3, 8.5),
    (102, 'Riya Verma', 'CSE', 2, 9.1),
    (103, 'Kabir Singh', 'ECE', 4, 7.8),
    (104, 'Meera Nair', 'EEE', 3, 8.0),
    (105, 'Ananya Gupta', 'CSE', 4, 8.9);
```

StudentID	Name	Department	Year	GPA
101	Aarav Sharma	CSE	3	8.5
102	Riya Verma	CSE	2	9.1
103	Kabir Singh	ECE	4	7.8
104	Meera Nair	EEE	3	8.0
105	Ananya Gupta	CSE	4	8.9

Inserting data in Courses table

```
insert into Courses (CourseID,CourseName,Department,Credits) value
    ('C101', 'DBMS', 'CSE', 4),
    ('C102', 'Operating Systems', 'CSE', 3),
    ('C103', 'Digital Electronics', 'ECE', 3),
    ('C104', 'Power Systems', 'EEE', 3),
    ('C105', 'AI and ML', 'CSE', 4);
```

CourseID	CourseName	Department	Credits
C101	DBMS	CSE	4
C102	Operating Systems	CSE	3
C103	Digital Electronics	ECE	3
C104	Power Systems	EEE	3
C105	AI and ML	CSE	4

Inserting data in Instructors table

```
insert into Instructors (InstructorID,Name,Department,Experience) value
    (1, 'Dr. R.K. Rao', 'CSE', 15),
    (2, 'Dr. S. Patel', 'ECE', 10),
    (3, 'Dr. Neha Joshi', 'EEE', 8),
    (4, 'Prof. A. Mehta', 'CSE', 12);
```

InstructorID	Name	Department	Experience
1	Dr. R.K. Rao	CSE	15
2	Dr. S. Patel	ECE	10
3	Dr. Neha Joshi	EEE	8
4	Prof. A. Mehta	CSE	12

Inserting data in Enrollments table

```
insert into Enrollments (EnrollID,StudentID,CourseID,Semester,Marks) value
    (1, 101, 'C101', 'Sem5', 87),
    (2, 101, 'C102', 'Sem5', 78),
    (3, 102, 'C101', 'Sem3', 91),
    (4, 103, 'C103', 'Sem7', 67),
    (5, 105, 'C105', 'Sem8', 94),
    (6, 105, 'C101', 'Sem8', 89),
    (7, 102, 'C105', 'Sem4', 92);
```

EnrollID	StudentID	CourseID	Semester	Marks
1	101	C101	Sem5	87
2	101	C102	Sem5	78
3	102	C101	Sem3	91
4	103	C103	Sem7	67
5	105	C105	Sem8	94
6	105	C101	Sem8	89
7	102	C105	Sem4	92

Inserting data in Course_Assignment table

```
insert into Course_Assignment (CourseID,InstructorID,Year) value
    ('C101', 1, 2024),
    ('C102', 1, 2024),
    ('C103', 2, 2024),
    ('C104', 3, 2024),
    ('C105', 4, 2024);
```

CourseID	InstructorID	Year
C101	1	2024
C102	1	2024
C103	2	2024
C104	3	2024
C105	4	2024

Questions

1. Create a view `CSE_COURSE_DETAILS` that lists all CSE department courses with instructor name, experience, and number of students enrolled.

```
create view CSE_COURSE_DETAILS as
select
    C.CourseID,
```

```
    C.CourseName,
    C.Credits,
    I.Name as `Instructor Name`,
    I.Experience,
    Count(E.StudentID) as `StudentCount`
from Courses C
join Course_Assignment CA on CA.CourseID = C.CourseID
join Instructors I on I.InstructorID = CA.InstructorID
join Enrollments E on E.CourseID = C.CourseID
where C.Department = "CSE"
group by C.CourseID,C.CourseName,C.Credits,I.Name,I.Experience;
```

```
mysql> select * from CSE_COURSE_DETAILS;
+----------+--------------------+---------+-----------------+------------+--------------+
| CourseID | CourseName         | Credits | Instructor Name | Experience | StudentCount |
+----------+--------------------+---------+-----------------+------------+--------------+
| C101     | DBMS               |       4 | Dr. R.K. Rao    |         15 |            3 |
| C102     | Operating Systems  |       3 | Dr. R.K. Rao    |         15 |            1 |
| C105     | AI and ML          |       4 | Prof. A. Mehta  |         12 |            2 |
+----------+--------------------+---------+-----------------+------------+--------------+
3 rows in set (0.00 sec)
```

2. Create a view STUDENT_COURSE_PERFORMANCE showing each student's name, department, course name, and marks.

```
create view STUDENT_COURSE_PERFORMANCE as
select
    S.Name,
    S.Department,
    C.CourseName,
    E.Marks
from Enrollments E
join Courses C on E.CourseID = C.CourseID
join Students S on E.StudentID = S.StudentID;
```

```
mysql> select * from STUDENT_COURSE_PERFORMANCE;
+--------------+------------+----------------------+-------+
| Name         | Department | CourseName           | Marks |
+--------------+------------+----------------------+-------+
| Aarav Sharma | CSE        | DBMS                 |    87 |
| Riya Verma   | CSE        | DBMS                 |    91 |
| Ananya Gupta | CSE        | DBMS                 |    89 |
| Aarav Sharma | CSE        | Operating Systems    |    78 |
| Kabir Singh  | ECE        | Digital Electronics  |    67 |
| Ananya Gupta | CSE        | AI and ML            |    94 |
| Riya Verma   | CSE        | AI and ML            |    92 |
+--------------+------------+----------------------+-------+
7 rows in set (0.00 sec)

mysql>
```

3. Create a view TOP_PERFORMERS_VIEW displaying students whose marks are higher than the average marks of their respective course (use a subquery).

```
create view TOP_PERFORMERS_VIEW as
select
    S.StudentID,
    S.Name as `Student Name`,
    S.Department,
    S.Year,
    S.GPA,
    E.CourseID,
    C.CourseName
from Enrollments E
join Students S on S.StudentID = E.StudentID
join Courses C on C.CourseID = E.CourseID
where E.Marks > (
    select avg(Marks)
    from Enrollments
    where StudentID = S.StudentID
);
```

```
mysql> select * from TOP_PERFORMERS_VIEW;
+-----------+--------------+------------+------+------+----------+-------------+
| StudentID | Student Name | Department | Year | GPA  | CourseID | CourseName  |
+-----------+--------------+------------+------+------+----------+-------------+
|       101 | Aarav Sharma | CSE        |    3 |  8.5 | C101     | DBMS        |
|       105 | Ananya Gupta | CSE        |    4 |  8.9 | C105     | AI and ML   |
|       102 | Riya Verma   | CSE        |    2 |  9.1 | C105     | AI and ML   |
+-----------+--------------+------------+------+------+----------+-------------+
3 rows in set (0.00 sec)
```

4. Create a view DEPARTMENT_AVG_VIEW that lists each department along with average GPA of students and average marks in courses of that department.

```
create view DEPARTMENT_AVG_VIEW as
select
    S.Department,
    avg(S.GPA) as `Average GPA`,
    avg(E.Marks) as `Average Marks`
from Students S
left join Enrollments E on E.StudentID = S.StudentID
left join Courses C on C.CourseID = E.CourseID and C.Department = S.Department
group by S.Department;
```

```
mysql> select * from DEPARTMENT_AVG_VIEW;
+------------+-------------+---------------+
| Department | Average GPA | Average Marks |
+------------+-------------+---------------+
| CSE        |     8.83333 |       88.5000 |
| ECE        |     7.80000 |       67.0000 |
| EEE        |     8.00000 |          NULL |
+------------+-------------+---------------+
3 rows in set (0.00 sec)
```

5. Modify CSE_COURSE_DETAILS to include average marks for each course as an additional column.

```sql
create or replace view CSE_COURSE_DETAILS as
select
    C.CourseID,
    C.CourseName,
    C.Credits,
    I.Name as `Instructor Name`,
    I.Experience,
    Count(E.StudentID) as `Student Count`,
    avg(E.Marks) as `Average Marks`
from Courses C
join Course_Assignment CA on CA.CourseID = C.CourseID
join Instructors I on I.InstructorID = CA.InstructorID
join Enrollments E on E.CourseID = C.CourseID
where C.Department = "CSE"
group by C.CourseID,C.CourseName,C.Credits,I.Name,I.Experience;
```

```
mysql> select * from CSE_COURSE_DETAILS;
+----------+------------------+---------+-----------------+------------+---------------+---------------+
| CourseID | CourseName       | Credits | Instructor Name | Experience | Student Count | Average Marks |
+----------+------------------+---------+-----------------+------------+---------------+---------------+
| C101     | DBMS             |       4 | Dr. R.K. Rao    |         15 |             3 |       89.0000 |
| C102     | Operating Systems |       3 | Dr. R.K. Rao    |         15 |             1 |       78.0000 |
| C105     | AI and ML        |       4 | Prof. A. Mehta  |         12 |             2 |       93.0000 |
+----------+------------------+---------+-----------------+------------+---------------+---------------+
3 rows in set (0.00 sec)
```

6. Using STUDENT_COURSE_PERFORMANCE, write a query to find top 3 students per department based on average marks.

```sql
select *
from (
        select
            *,
            avg(Marks) as `Average Marks`,
            rank() over (partition by Department order by avg(Marks) desc) as
`Rank In Department`
        from STUDENT_COURSE_PERFORMANCE
        group by Department,Name,CourseName,Marks
    ) as ranked
where `Rank In Department` <= 3;
```

Name	Department	CourseName	Marks	Average Marks	Rank In Department
Ananya Gupta	CSE	AI and ML	94	94.0000	1
Riya Verma	CSE	AI and ML	92	92.0000	2
Riya Verma	CSE	DBMS	91	91.0000	3
Kabir Singh	ECE	Digital Electronics	67	67.0000	1