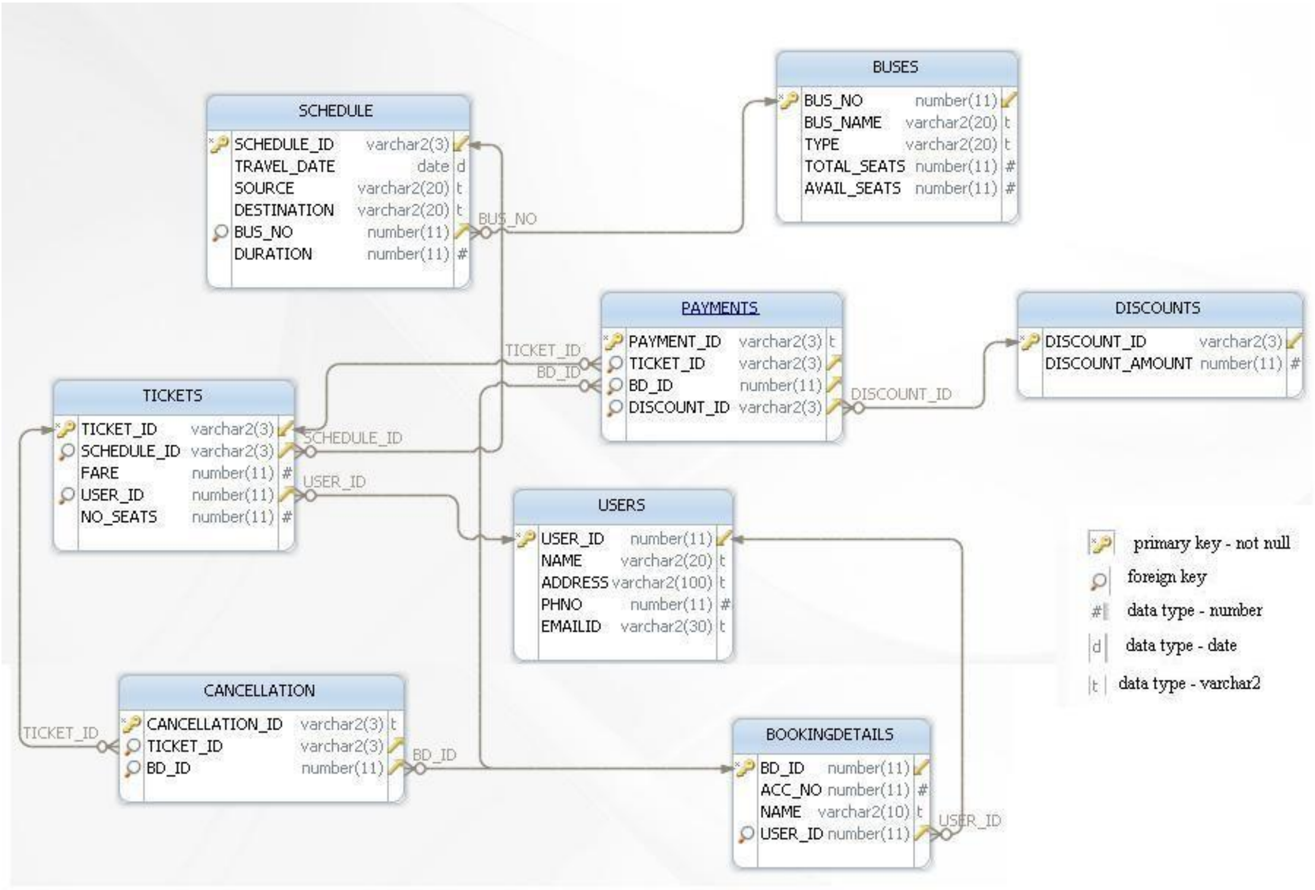
# Buses based on Source and Destination

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display list of bus number and names where source and destination of a bus is the destination and source of another bus. Display unique records in ascending order by bus_no.

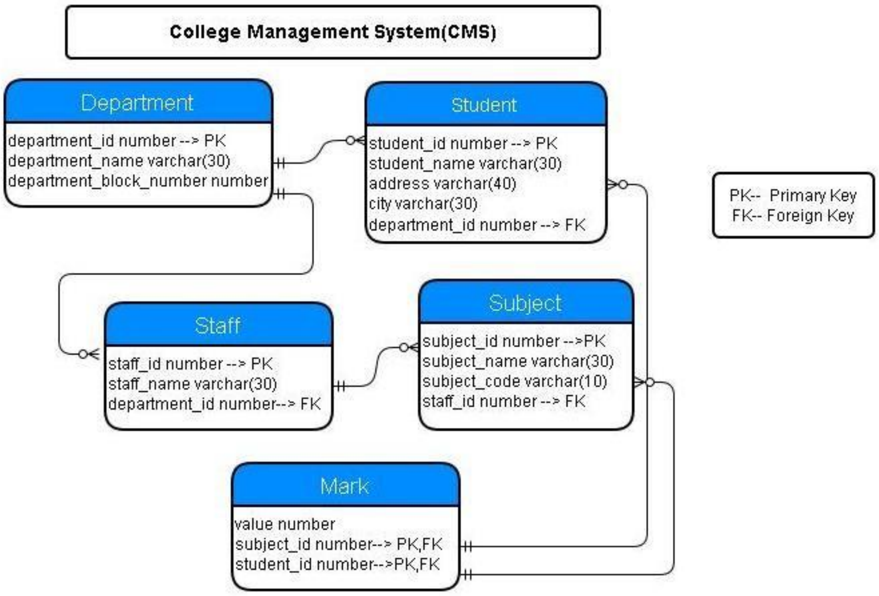


# Department Name with Maximum Student Count

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the name of the department that has the maximum student count.

Refer the below schema diagram:



Remaining time

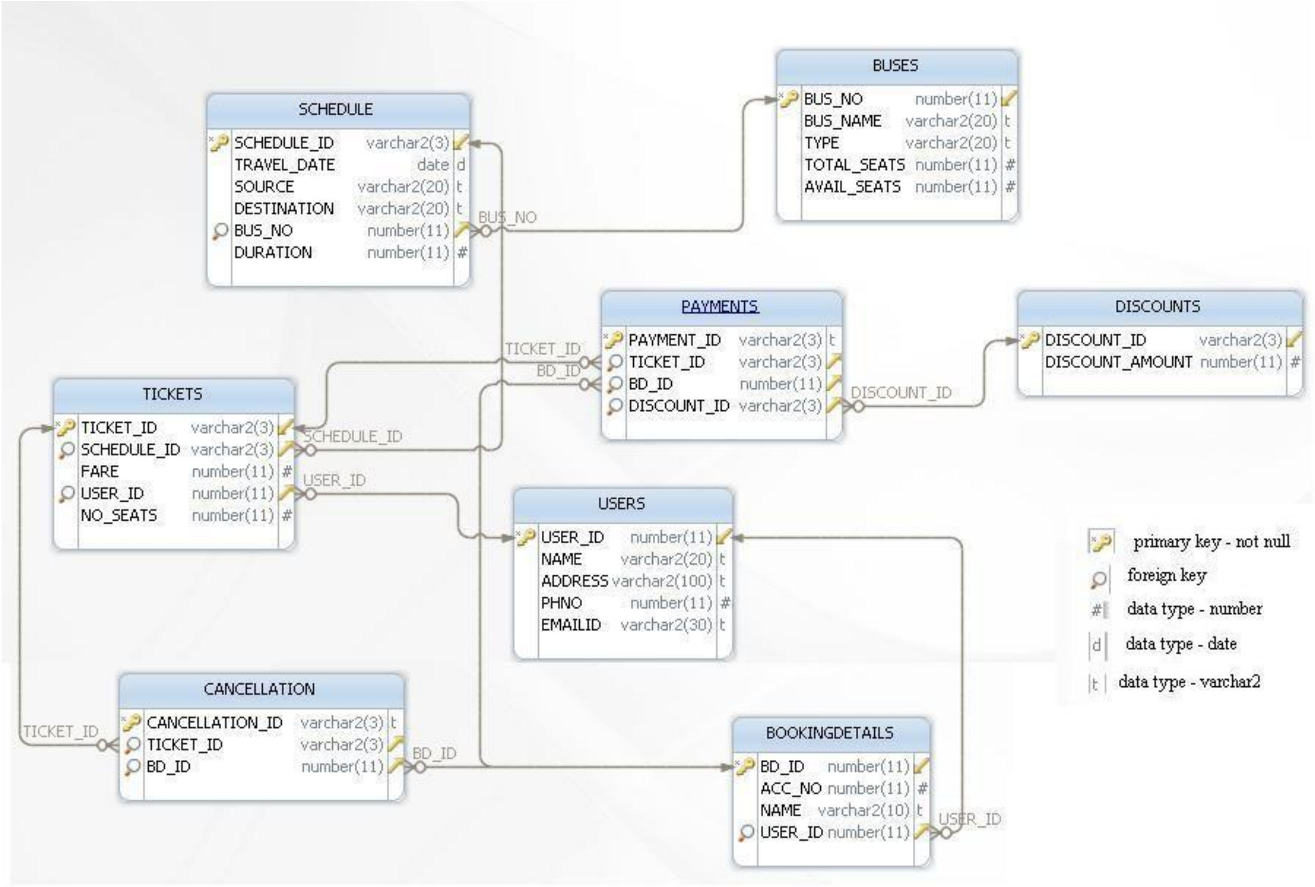
**Summary Details**

**Live Proctoring**

# Minimum Travel Time

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Display schedule_id, source, destination and travel_time which has minimum travel time. Sort the result based on schedule id

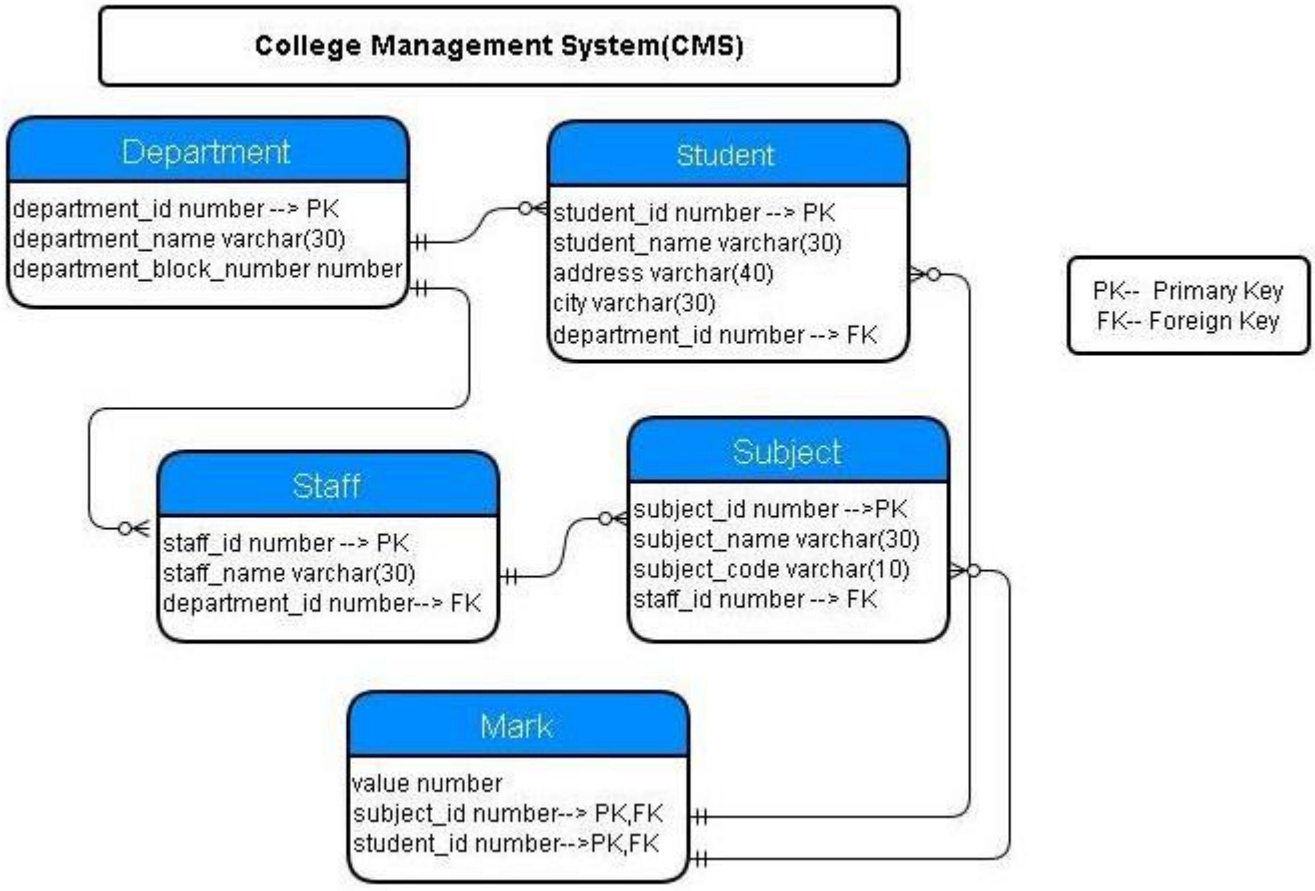


# Staff details based on subjects

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the names of the staff who are handling atleast 1 subject, ordered by name in ascending order.

Refer the below schema diagram:

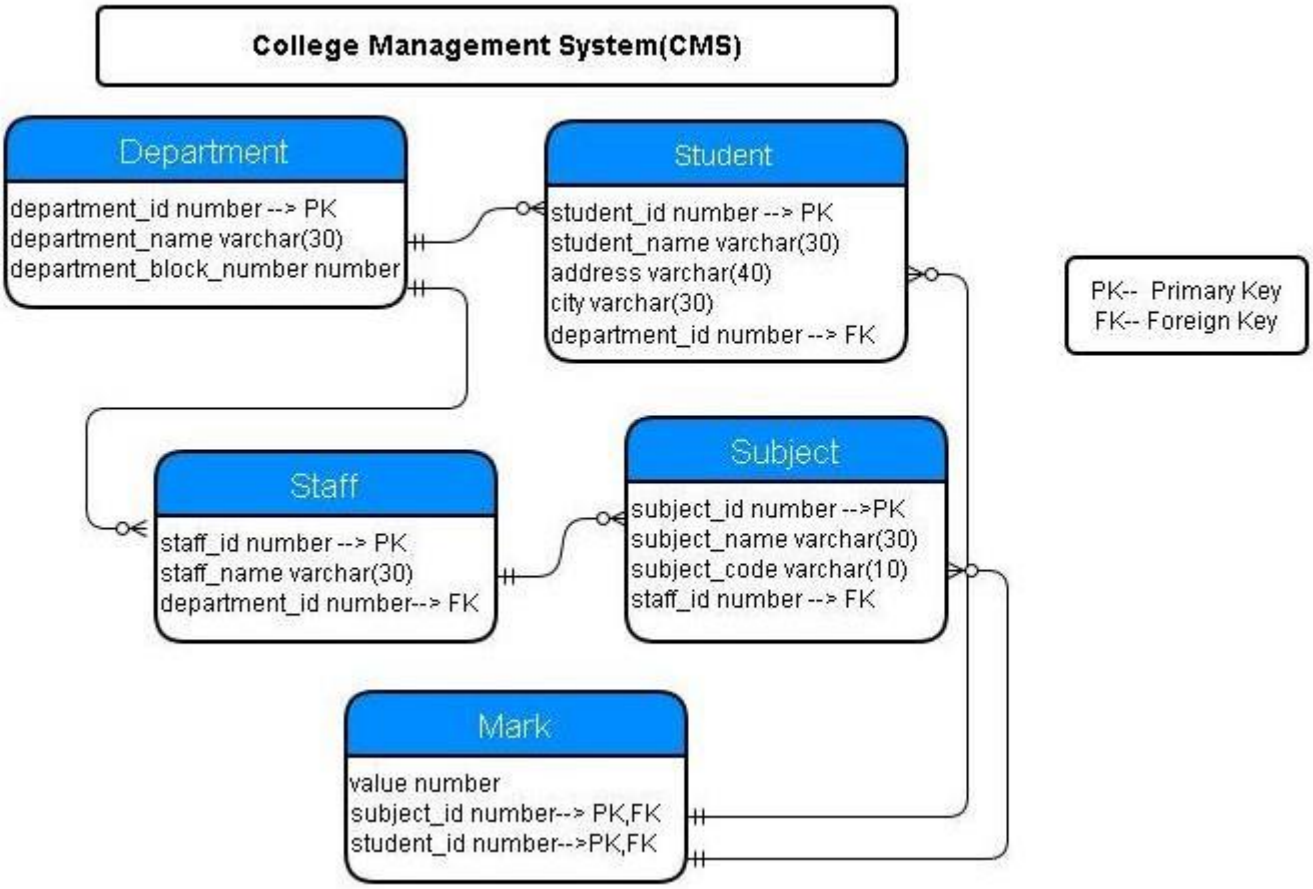


# Staff details without handling any subjects

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the names of the staff who are not handling any subject, ordered by name in ascending order.

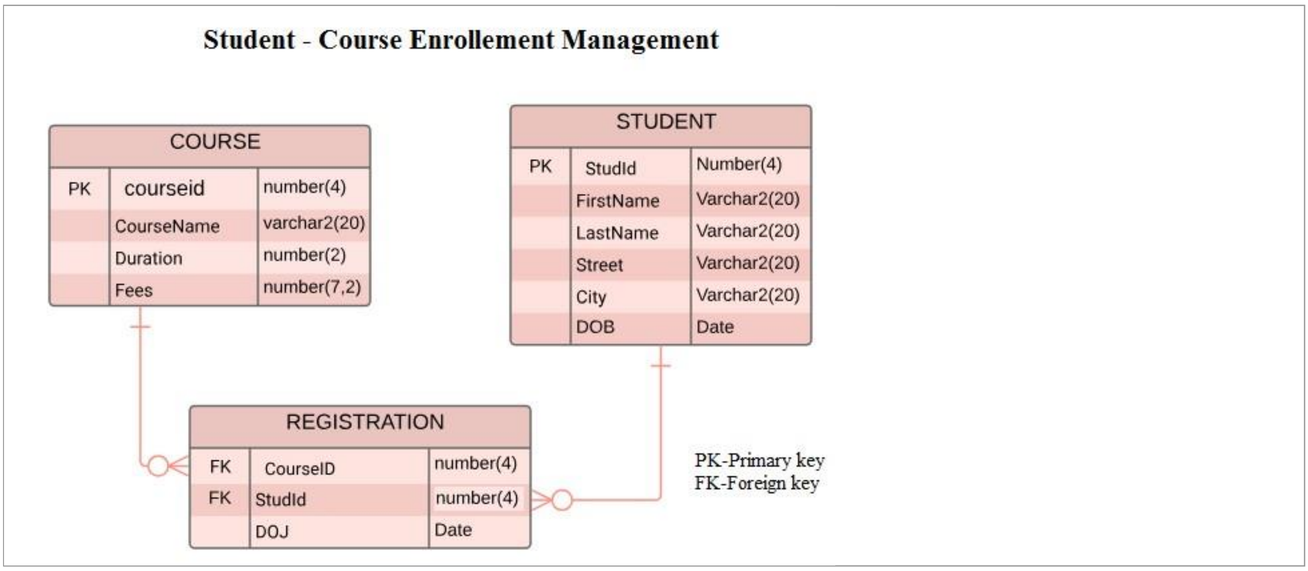
Refer the below schema diagram:



## Student ID with Total Fees

**Grade settings**: Maximum grade: 40
**Based on**: Meta - Select -Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the student's ID and the total fees paid by each student. Give alias name to total fees as TOTALFEES. Sort the result based on student ID.



Student - Course Enrollement Management

COURSE

| | | |
|---|---|---|
| PK | courseid | number(4) |
| | CourseName | varchar2(20) |
| | Duration | number(2) |
| | Fees | number(7,2) |

STUDENT

| | | |
|---|---|---|
| PK | StudId | Number(4) |
| | FirstName | Varchar2(20) |
| | LastName | Varchar2(20) |
| | Street | Varchar2(20) |
| | City | Varchar2(20) |
| | DOB | Date |

REGISTRATION

| | | |
|---|---|---|
| FK | CourseID | number(4) |
| FK | StudId | number(4) |
| | DOJ | Date |

PK-Primary key
FK-Foreign key

Remaining time

Summary Details

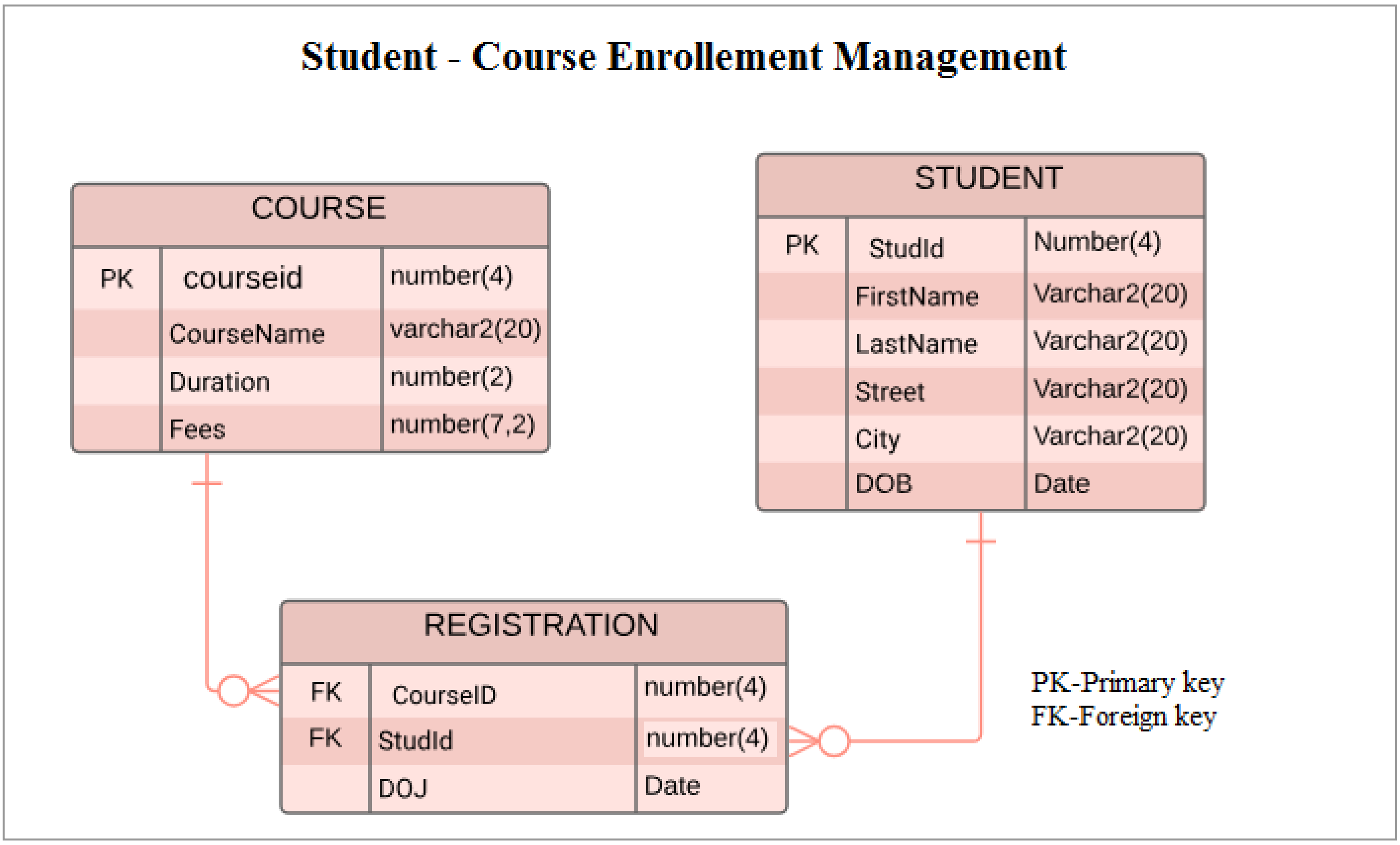
Live Proctoring

# Student Name alone with DOJ and DOC

**Grade settings**: Maximum grade: 40
**Based on**: Meta - Select -Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the student's first name, date of joining and completing date of each course. Give alias name for completion date as DOC. Sort the result based on student's first name , Date of joining and Date of Completion.

[Note: Completion Date can be calculated using DOJ and duration]



Student - Course Enrollement Management

COURSE

| | | |
|---|---|---|
| PK | courseid | number(4) |
| | CourseName | varchar2(20) |
| | Duration | number(2) |
| | Fees | number(7,2) |

STUDENT

| | | |
|---|---|---|
| PK | StudId | Number(4) |
| | FirstName | Varchar2(20) |
| | LastName | Varchar2(20) |
| | Street | Varchar2(20) |
| | City | Varchar2(20) |
| | DOB | Date |

REGISTRATION

| | | |
|---|---|---|
| FK | CourseID | number(4) |
| FK | StudId | number(4) |
| | DOJ | Date |

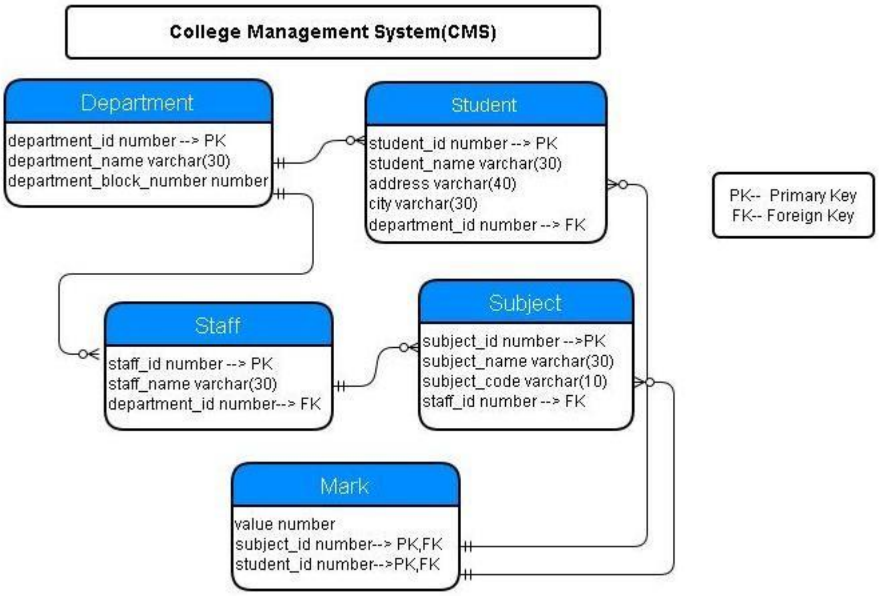
PK-Primary key
FK-Foreign key

## Student name with Computer Programming marks

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the students names and their marks in Computer Programming order by marks in ascending order and then ordered by name in descending order. Give an alias to marks as CP_MARKS.

Refer the below schema diagram:



Remaining time

Summary Details

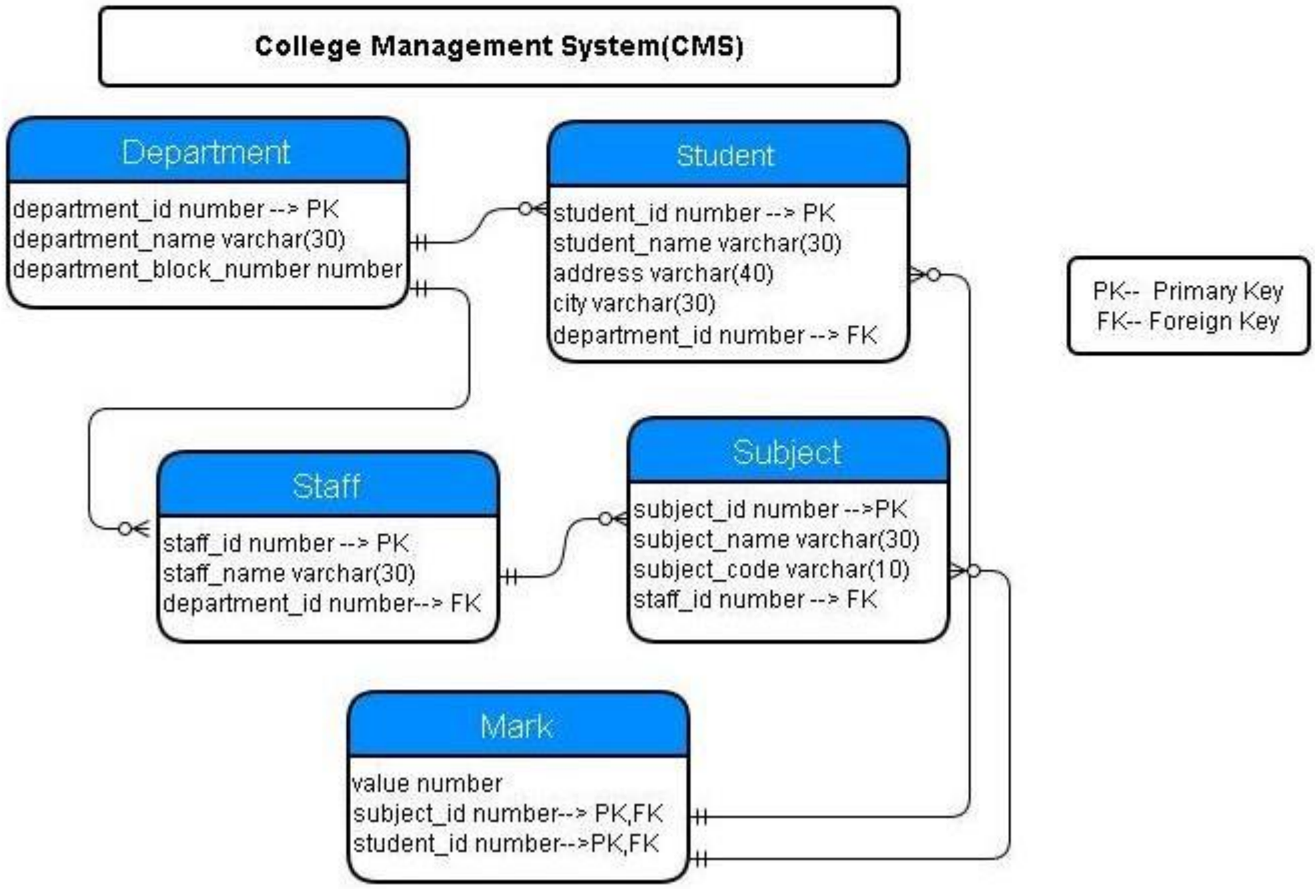
Live Proctoring

# Students details based mark scored

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write a query to display the names of all students who have secured more than 50 in all subjects that they have appeared in, ordered by student name in ascending order.

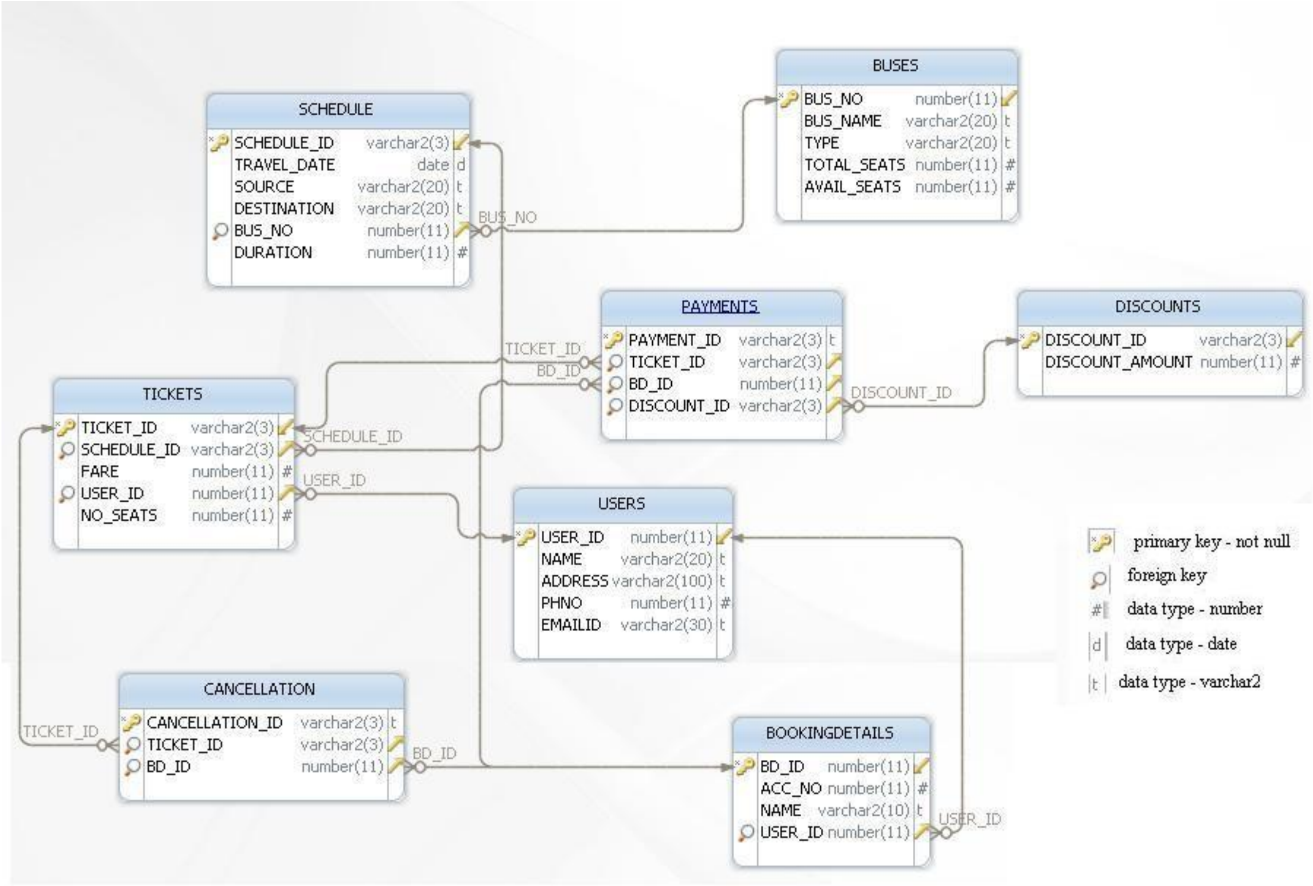
Refer the below schema diagram:



# User details based on Cancellation

**Grade settings**: Maximum grade: 40
**Based on**: DRL - Oracle
**Run**: Yes **Evaluate**: Yes
**Automatic grade**: Yes

Write query to display user_id and user name of those who have canceled their tickets and display records in ascending order by user_id. (Hint: Take user,tickets and cancellation tables)



SCHEDULE

| | | |
|---|---|---|
| SCHEDULE_ID | varchar2(3) | |
| TRAVEL_DATE | date | d |
| SOURCE | varchar2(20) | t |
| DESTINATION | varchar2(20) | t |
| BUS_NO | number(11) | |
| DURATION | number(11) | # |

BUSES

| | | |
|---|---|---|
| BUS_NO | number(11) | |
| BUS_NAME | varchar2(20) | t |
| TYPE | varchar2(20) | t |
| TOTAL_SEATS | number(11) | # |
| AVAIL_SEATS | number(11) | # |

PAYMENTS

| | | |
|---|---|---|
| PAYMENT_ID | varchar2(3) | t |
| TICKET_ID | varchar2(3) | |
| BD_ID | number(11) | |
| DISCOUNT_ID | varchar2(3) | |

DISCOUNTS

| | | |
|---|---|---|
| DISCOUNT_ID | varchar2(3) | |
| DISCOUNT_AMOUNT | number(11) | # |

TICKETS

| | | |
|---|---|---|
| TICKET_ID | varchar2(3) | |
| SCHEDULE_ID | varchar2(3) | |
| FARE | number(11) | # |
| USER_ID | number(11) | |
| NO_SEATS | number(11) | # |

USERS

| | | |
|---|---|---|
| USER_ID | number(11) | |
| NAME | varchar2(20) | t |
| ADDRESS | varchar2(100) | t |
| PHNO | number(11) | # |
| EMAILID | varchar2(30) | t |

CANCELLATION

| | | |
|---|---|---|
| CANCELLATION_ID | varchar2(3) | t |
| TICKET_ID | varchar2(3) | |
| BD_ID | number(11) | |

BOOKINGDETAILS

| | | |
|---|---|---|
| BD_ID | number(11) | |
| ACC_NO | number(11) | # |
| NAME | varchar2(10) | t |
| USER_ID | number(11) | |

## Buses based on Source and Destination

```
select b.bus_no, b.bus_name
from buses b, schedule s1,
schedule s2
where b.bus_no=s1.bus_no and
s1.source=s2.destination
and s2.source=s1.destination and
s1.bus_no!=s2.bus_no
order by 1;
```

## Department Name with Maximum Student Count

```
SELECT department_name FROM Department
NATURAL JOIN student GROUP BY department_name
HAVING COUNT(student_id)=(SELECT MAX(stu_count)
FROM (SELECT department_name, COUNT (student_id) AS stu_count
FROM Department
NATURAL JOIN student
GROUP BY department_name)
);
```

## Minimum Travel Time

```
select schedule_id,source,destination,duration from schedule
where duration=(select min(duration) from schedule)
order by 1;
```

## Staff details based on subjects

```
select staff_name
from Subject s join Staff t
on s.staff_id= t.staff_id
having count(subject_id)!=0
group by staff_name
order by staff_name;
```

## Staff details without handling any subjects

```
select staff_name from staff
where staff_id NOT IN(select staff_id from Subject)
order by staff_name;
```

## Student ID with Total Fees

```
select Studid, sum(fees) as TOTALFEES from REGISTRATION
join COURSE using (courseid)
group by Studid
order by Studid;
```

## Student Name alone with DOJ and DOC

```
select s.firstname, r.doj,add_months(r.doj,c.duration)
as DOC from Registration r,Course c,student s where
r.courseid=c.courseid and
r.studid=s.studid
order by s.firstname,r.doj,doc;
```

## Student name with Computer Programming marks

```
select st.student_name, m.value as CP_MARKS
from mark m
inner join student st on st.student_id=m.student_id
inner join subject su on su.subject_id=m.subject_id
where su.subject_name='Computer Programming'
order by m.value asc,st.student_name desc;
```

## Students details based mark scored

```
select student_name from student join mark using(student_id) group by student_name having min(value)>50 order by student_name;
```

# User details based on Cancellation

```
select u.USER_ID,u.NAME
from((users u inner join tickets
t
          on
u.USER_ID=t.USER_ID)inner
join  cancellation c
                 on
t.ticket_id=c.ticket_id)
where cancellation_ID is not null
group by u.USER_ID,u.NAME
order by 1;
```