INDEX AND ER DIAGRAM

Question 2

```
Select b.Emp_name , Sum(a.Salary) from Salary_table a join Employee_table b
on a.Employee_Id = b.Employee_Id group by a.Employee_Id
```

===

Question 3

```
Create as view sample as

Select a.mem_id , a.hotel_id, b.hotel_name( case when a.mem_id = 0 then
b.duration * b.guest_cost else b.duration * b.member_cost end ) bill from
booking a join hotel b on a.hotel_id = b.hotel_id;


Select hotel_id , hotel_name ,(case when sum(bill)> 14300 then ‘high’

                                 when sum(bill) < 1000 then ‘low’

else ‘average’ end ) status from sample  a group by hotel_id;
```

===

Question 5

```
Select a.stadium_name , a.city, count (*) as match_count from stadium a left
join Match_schedule b on a.stadium_id = b. stadium_id group by
a.stadium_name , a.city order by match_count DESC;
```

Question 6

```
Select a.name from player a join team_player b on a.player_id = b.player_id

Join team c on c.team_id = b.team_id where c.name = ‘Mumbai Indians’
```

===

Question 7

```
Select a.team_id , b.name,count(a.role) as count from team_player a join team
b on a.team_id = b.team_id where a.role =’All rounder’ group by a.team_id
having count > 4 order by count desc ;
```

Question 1

create a table called student with below column
sid int,
sname varchar(20),
phno char(10),
class varchar(10),
email_id varchar(50)
create a index for the above table and justify the answer

Query

```
CREATE INDEX idx_sid ON student (sid);
```

Justification

In this example, I created an index on the sid column. Creating an index on a column like sid allows the database to quickly find rows based on the sid values.

Indexes are especially useful for columns commonly used in search, join, and filtering operations. In this case, creating an index on sid can improve the speed of queries that involve searching or retrieving records based on the student ID.

Question 4

```
With CTE as(
select a.empid , a.emp_Fisrtname , cast(b.order_date as date) as Order_date ,
count(b.id) as orders_count
from employees a join orders b on a.id = b.emp_id
where a.empid = 201
group by cast(b.order_date as date)
ORDER BY cast(b.order_date as date) ASC
)
select * , sum(orders_count) over(partition by
year(b.Order_date),month(b.Order_date)
order by a.empid,year(b.Order_date),month(b.Order_date) ) as
cummulative_order_count
from CTE;
```