**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 ?**

**CODE:**

use sales\_db;

create table student(

sid int,

sname varchar(20),

phno char(10),

class varchar(10),

email\_id varchar(50)

);

--- Inserting 10 sample rows with "student" table

INSERT INTO student (sid, sname, phno, class, email\_id)

VALUES

(1, 'Rahul Sharma', '1234567890', '10A', 'rahul.sharma@example.com'),

(2, 'Priya', '9876543210', '9B', 'priya.patel@example.com'),

(3, 'Amit Verma', '5551237890', '11C', 'amit.verma@example.com'),

(4, 'Kavita Gupta', '9998887777', '8A', 'kavita.gupta@example.com'),

(5, 'Vivek Singh', '1112223333', '12B', 'vivek.singh@example.com'),

(6, 'Neha Reddy', '7773331111', '7C', 'neha.reddy@example.com'),

(7, 'Rajesh Kumar', '4445556666', '9A', 'rajesh.kumar@example.com'),

(8, 'Anjali Mishra', '8887776666', '11B', 'anjali.mishra@example.com'),

(9, 'Sandeep Joshi', '3334445555', '10C', 'sandeep.joshi@example.com'),

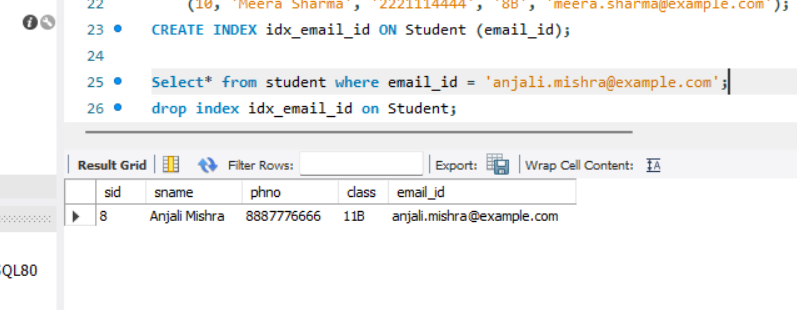
(10, 'Meera Sharma', '2221114444', '8B', 'meera.sharma@example.com');

CREATE INDEX idx\_email\_id ON Student (email\_id);

Select\* from student where email\_id = 'anjali.mishra@example.com';

drop index idx\_email\_id on Student;

**OUTPUT:**

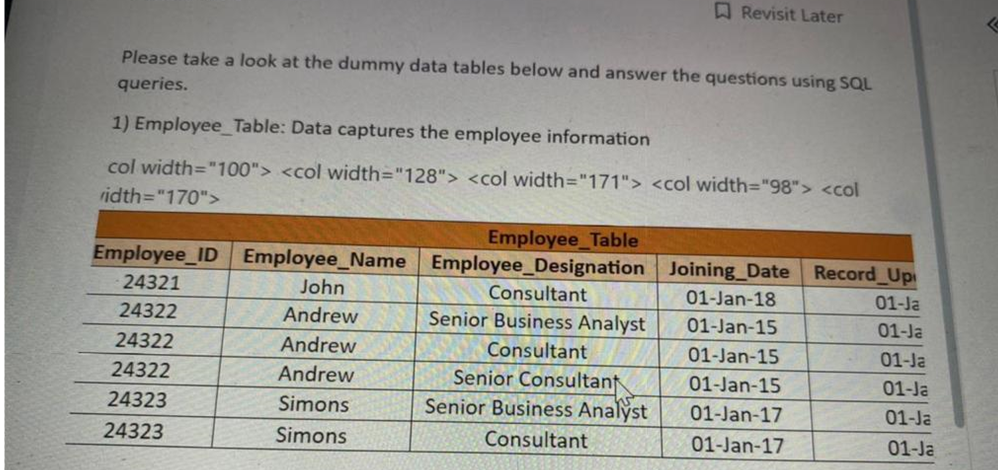


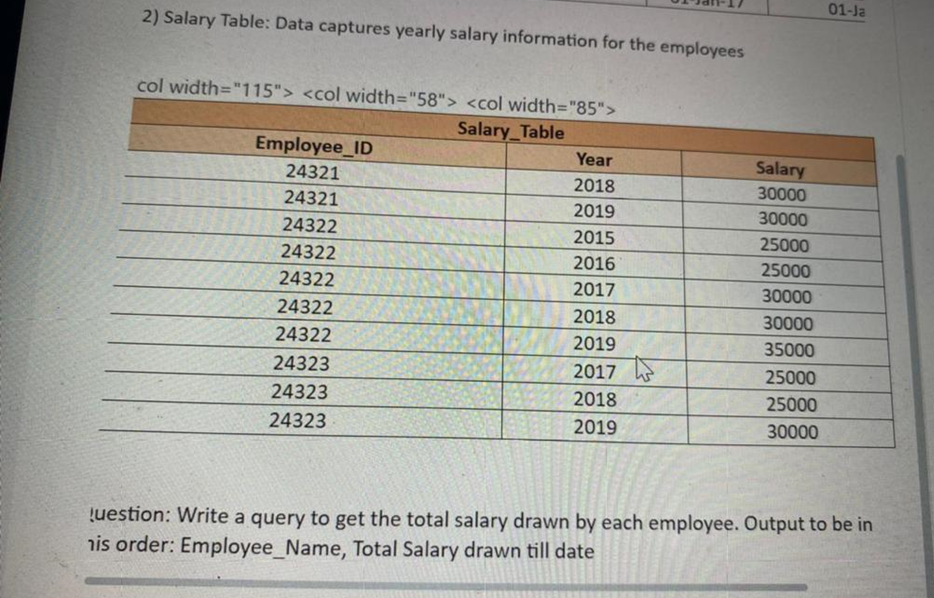
**Advantage of creating Index :**

* **When we create the index to the table, it will directly fetch the data from that particular index of that column instead of searching / Scanning entire table values to find the particular select query of a column.**
* **Here this index (**CREATE INDEX idx\_email\_id ON Student (email\_id)) **function takes very lesser time than normal select query does. It will directly go and fetch the values from the particular column when writing select query after creating index.**

**EX -** Select\* from student where email\_id = 'anjali.mishra@example.com';

**Question2**





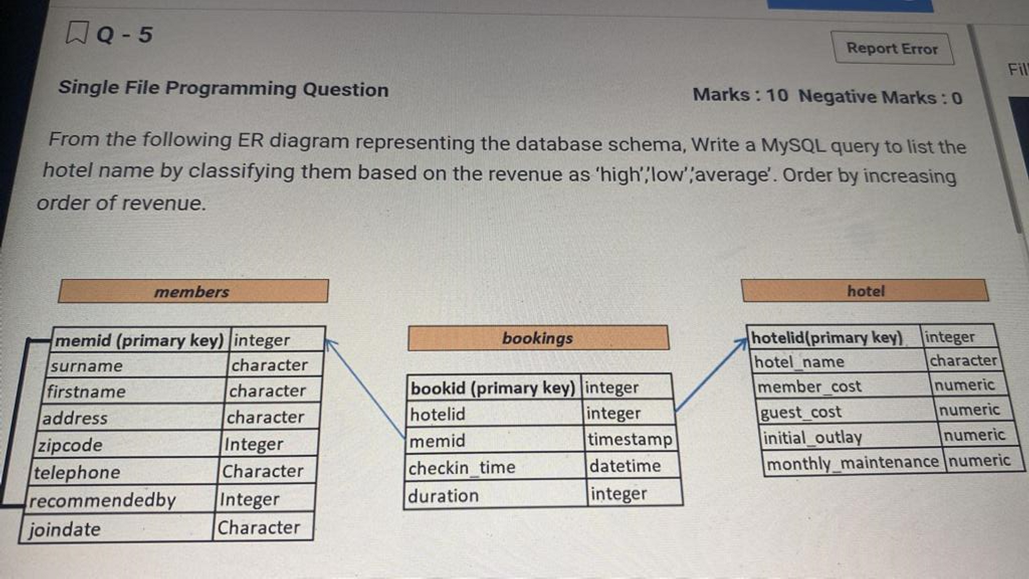
Query :

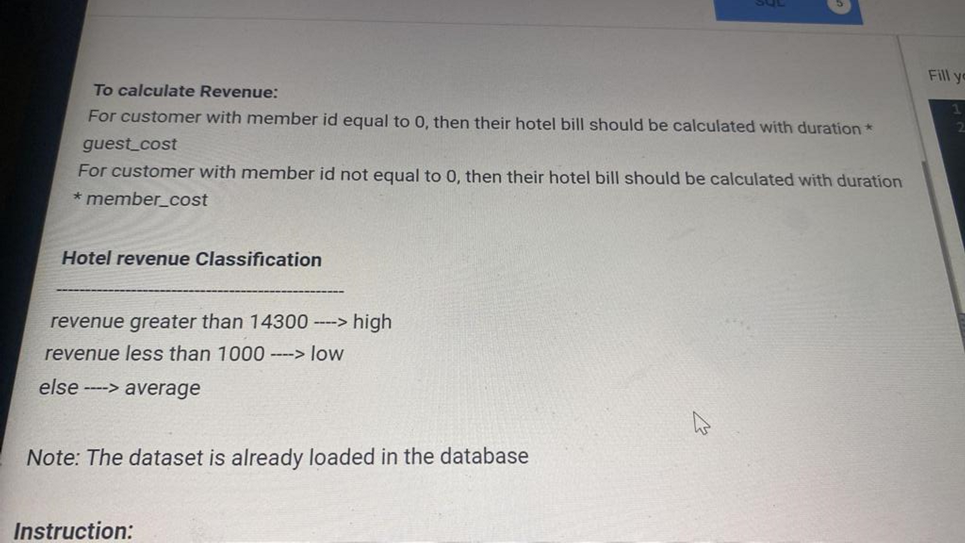
**SELECT** E.Employee\_Name, Sum(S.Salary) Employee E

**JOIN** Salary S  **ON** E.Emplyee\_ID = S.Employee\_ID

**GROUP BY** S.Employee\_ID;

**Question 3**





select hotel\_id, hotel\_name,

(case when sum (bill) > 14300 **THEN** 'High'

**WHEN** sum (bill) < 1000 **THEN** 'Low'

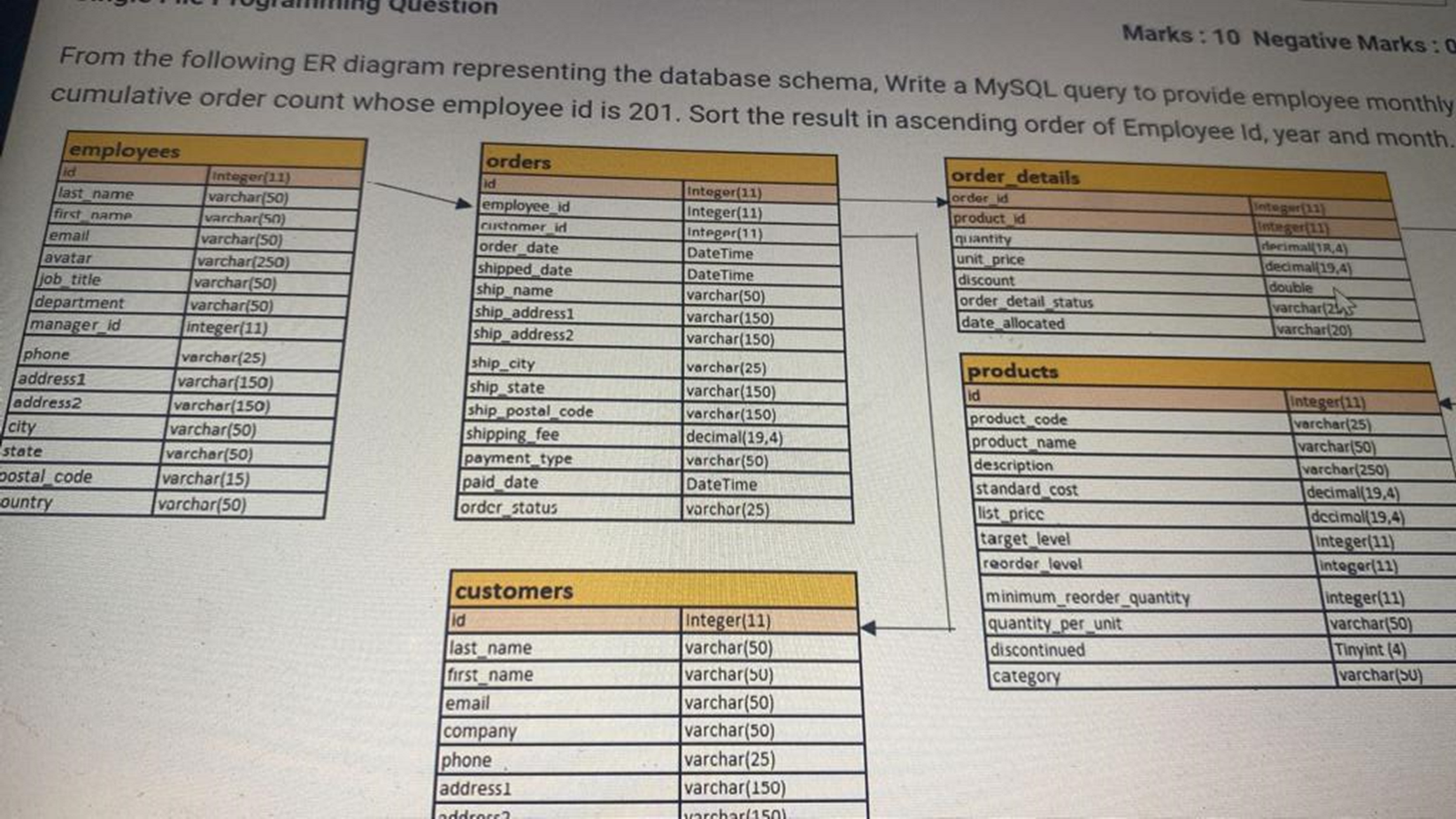
else 'Average' ) **END AS** status **FROM** (

**SELECT** memid, hotelid, hotel\_name

(case when memid = 0 **THEN** duration \*quest cost

Else duration\*member cost end) bill

**FROM** booking **JOIN** hotel **ON** booking.hotelid-hotel.hotelid) a **GROUP BY** hotelid;

**Question 4** 

**SELECT** employee\_id,

YEAR (order\_date) AS year,

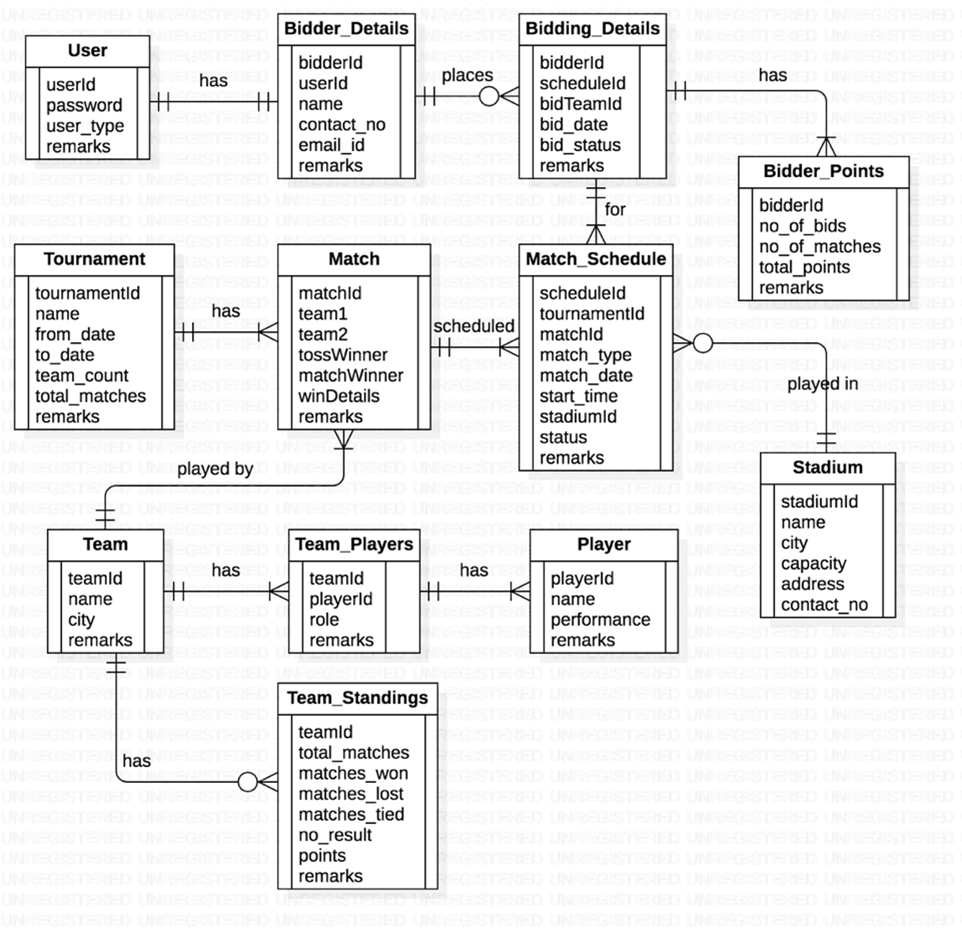
MONTH (order\_date) AS month,

COUNT (\*) **OVER** (PARTITION BY employee\_id, year, month **ORDER BY** order\_date) **AS** cumulative\_order\_count

**FROM** orders WHERE employee\_id = 201

**ORDER BY** employee\_id, year, month;

**Question 5**



Display the number of matches conducted at each stadium with stadium name, city from the database.

SELECT

stadium.name AS stadium\_name,

stadium.city,

COUNT(match\_schedule.stadium\_id) AS number\_of\_matches

FROM match\_schedule

JOIN stadium ON match\_schedule.stadium\_id = stadium.stadium\_id

GROUP BY stadium.name, stadium.city

ORDER BY stadium.name;

Question 6

Display the bowlers for Mumbai Indians team

SELECT PLAYER\_NAME FROM TEAM\_PLAYER

JOIN TEAM ON TEAM\_PLAYER.TEAM\_ID = TEAM.TEAM\_ID

WHERE TEAM.TEAM\_NAME = 'Mumbai Indians'

AND TEAM\_PLAYER.ROLE = 'bowler';

Question 7

How many all-rounders are there in each team, Display the teams with more than 4

all-rounder in descending order.

SELECT

Team.name AS team\_name,

COUNT(team\_player.player\_id) AS number\_of\_all\_rounders

FROM team\_player

JOIN team ON team\_player.team\_id = team.team\_id

WHERE team\_player.role = 'all-rounder'

GROUP BY team.name HAVING COUNT(team\_player.player\_id) > 4

ORDER BY number\_of\_all\_rounders DESC;