**TEST-1 AZURE SQL**

1. Create the tables below with given instructions and load sample data after that

* Give appropriate data type to columns
* Implement primary key in both tables
* Establish primary key - foreign key relationship between the tables
* Identify a column in employee table to implement unique key and not null constraint

Table 1: Department

|  |
| --- |
| **DeptID** |
| **DeptName** |
| **DeptCode** |

Table 2: Employee

|  |
| --- |
| **EmpID** |
| **EmpName** |
| **EmailId** |
| **DeptID** |
| **Designation** |

**ANSWER -**

drop table department;

create table department(

    DeptId integer PRIMARY KEY,

    DeptName varchar(16),

    DeptCode numeric not null

);

drop table employee;

create table employee(

    EmpId integer PRIMARY KEY,

    EmpName varchar(25) not null,

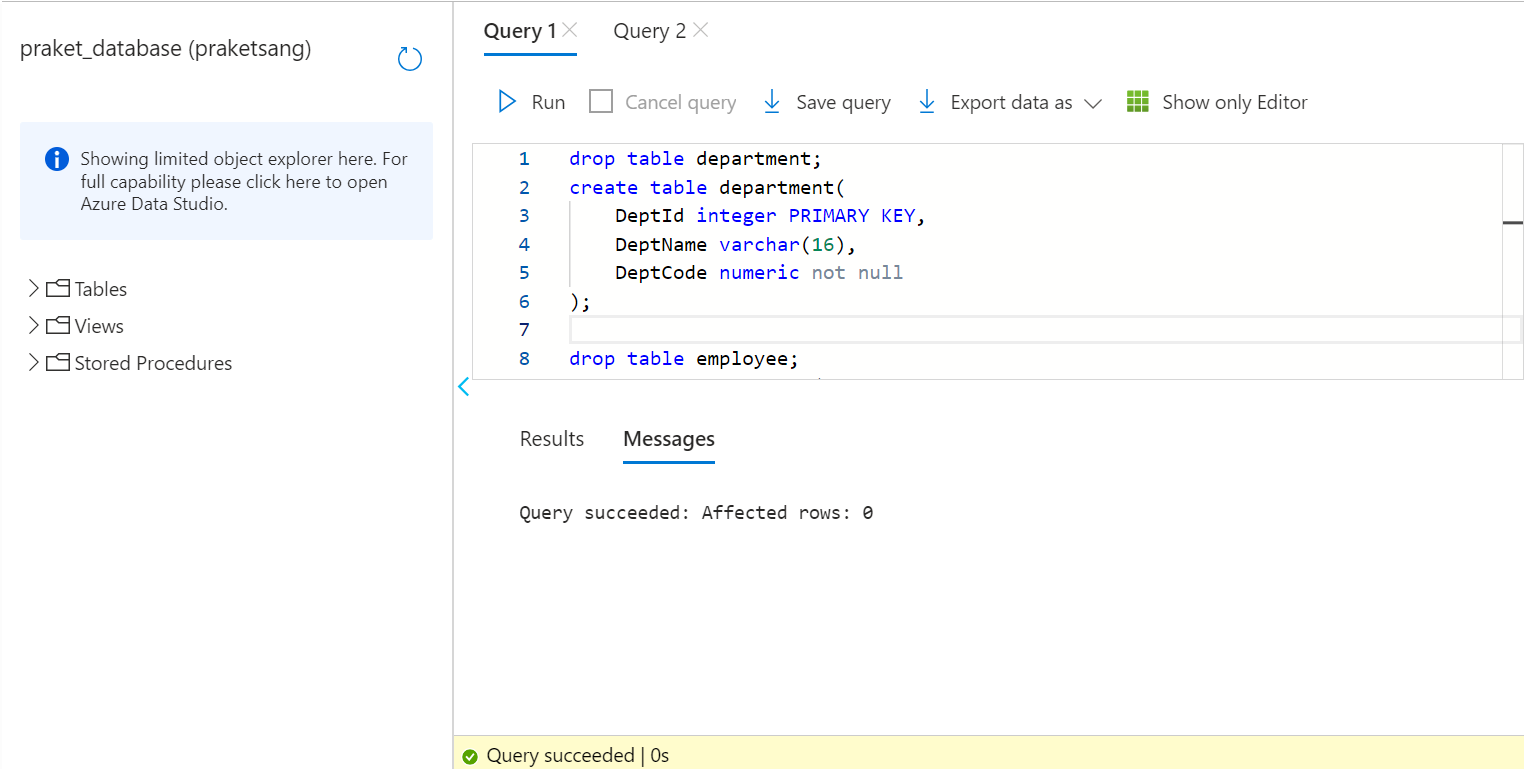
    EmailId varchar(50) unique,

    DeptId integer,

    Designation varchar(10),

    FOREIGN KEY (DeptId) REFERENCES department(DeptId)

);



1. Create the below table & Write queries based on below table

|  |  |
| --- | --- |
| Table Name: **students** |  |
|  |  |
| Column Name | Type |
| slno | Integer |
| id | integer |
| name | String |
| subject | String |
| marks | Integer |

Sample Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| slno | id | name | subject | marks |
| 1 | 5030 | Ashley | DWH | 50 |
| 2 | 5031 | Priya | DWH | 61 |
| 3 | 5032 | Julia | DWH | 90 |
| 4 | 5033 | Don | DWH | 50 |
| 5 | 5034 | Bob | DWH | 65 |
| 6 | 5030 | Ashley | SQL | 75 |
| 7 | 5031 | Priya | SQL | 80 |
| 8 | 5032 | Julia | SQL | 40 |
| 9 | 5033 | Don | SQL | 88 |
| 10 | 5034 | Bob | SQL | 92 |
| 11 | 5030 | Ashley | Azure | 95 |
| 12 | 5031 | Priya | Azure | 45 |
| 13 | 5032 | Julia | Azure | 85 |
| 14 | 5033 | Don | Azure | 57 |
| 15 | 5034 | Bob | Azure | 44 |

Write queries to display

1. the record of the students who scored highest score in each subject
2. the records of the students in the order of highest to lowest marks in each subject and subjects in the ascending order
3. the record of the students with marks less than 50
4. the records of students with highest to lowest marks in ’Azure’
5. display the total score (sum of the score in all subjects) of each student in a new column
6. display the rank of the students in ascending order based on total score
7. display the rank of the department in ascending order based on total score
8. display the record of the students with the second highest scorer in each subject
9. display the highest mark for each subject as a new column against corresponding subjects.

Eg given below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| slno | name | subject | marks | highest\_score |
| 1 | Ashley | DWH | 50 | 90 |
| 2 | Priya | DWH | 65 | 90 |
| 3 | Julia | DWH | 90 | 90 |

1. Display a new column email id combining name & id column. Format should be

first 3 characters of the name underscore last 2 characters of the id column @ta.com

**Note: Prepare data dictionary for the tables in Q1 and Q2.**

**ANSWER**

create table students(

    slno integer primary key,

    id integer not null,

    s\_name varchar(16) not null,

    s\_subject varchar(16) not null,

    s\_marks integer

);

INSERT INTO Students VALUES(1,5030,'Ashley','DWH',50);

INSERT INTO Students VALUES(2,5031,'Priya','DWH',61);

INSERT INTO Students VALUES(3,5032,'Julia','DWH',90);

INSERT INTO Students VALUES(4,5033,'Don','DWH',50);

INSERT INTO Students VALUES(5,5034,'Bob','DWH',65);

INSERT INTO Students VALUES(6,5030,'Ashley','SQL',75);

INSERT INTO Students VALUES(7,5031,'Priya','SQL',80);

INSERT INTO Students VALUES(8,5032,'Julia','SQL',40);

INSERT INTO Students VALUES(9,5033,'Don','SQL',88);

INSERT INTO Students VALUES(10,5034,'Bob','SQL',92);

INSERT INTO Students VALUES(11,5030,'Ashley','Azure',95);

INSERT INTO Students VALUES(12,5031,'Priya','Azure',45);

INSERT INTO Students VALUES(13,5032,'Julia','Azure',85);

INSERT INTO Students VALUES(14,5033,'Don','Azure',57);

INSERT INTO Students VALUES(15,5034,'Bob','Azure',44);

--- Query 1 ---

select s1.s\_name, s1.s\_subject, s1.s\_marks from students s1

join

(select s\_subject, max(s\_marks) as max\_marks from students group by s\_subject) s2

on (s1.s\_subject = s2.s\_subject and s1.s\_marks = max\_marks);

                        --- OR ---

with cte as (select dense\_rank() over (partition by s\_subject order by s\_marks desc) xyz, s\_subject, s\_marks, s\_name from students)

select \* from cte where xyz = 1;

--- Query 2 ---

select \* from students order by s\_subject asc, s\_marks desc;

--- Query 3 ---

select \* from students where s\_marks<50;

--- Query 4 ---

select \* from students where s\_subject = 'Azure' order by s\_marks desc;

--- Query 5 ---

select s\_name, sum(s\_marks) from students group by s\_name;

--- Query 6 ---

select s\_name, sum(s\_marks) from students group by s\_name order by sum(s\_marks);

--- Query 7 ---

select s\_subject, sum(s\_marks) from students group by s\_subject order by sum(s\_marks);

--- Query 8 ---

with cte as (select dense\_rank() over (partition by s\_subject order by s\_marks desc) xyz, s\_subject, s\_marks, s\_name from students)

select \* from cte where xyz = 2;

--- Query 9 ---

select s1.s\_name, s1.s\_subject, s1.s\_marks, s2.max\_marks from students s1

join

(select s\_subject, max(s\_marks) as max\_marks from students group by s\_subject) s2

on (s1.s\_subject = s2.s\_subject);

--- Query 10 ---

select \*, (left(s\_name,3) + right(id,2) + '@ta.com') as email from students;