Database Management System: Assignment 2

Total Marks : 20

July 1, 2024

Question 1

Marks: 2 **MCQ**

In a particular messenger application, the instance of `ChatDetails` is as follows:

ChatDetails		
SenderID	ReceiverID	Total_Text
R001	S001	1000
R002	S003	1200
R001	S002	500
R003	S004	700
R004	S004	1400

For the instance, the `Total_Text` values need to be updated to increase by 500 for those entries whose current values are less than 1000. What is the correct SQL Query for updating the current instance?

a) `MODIFY ChatDetails Total_Text=Total_Text+500 where Total_Text<1000;`

b) `UPDATE ChatDetails set Total_Text=Total_Text+500 where Total_Text<1000;`

c) `UPDATE ChatDetails Total_Text=Total_Text+500 where Total_Text<1000;`

d) `ALTER ChatDetails set Total_Text=Total_Text+500 where Total_Text<1000;`

Answer: b)
Explanation: For updating a particular field, the `UPDATE` clause is used in SQL The correct command is given below:

```
UPDATE Tablename
SET column1 = value1, column2 = value2, column3 = value3,....
WHERE condition;
```

Hence, option (b) is the correct answer.

Question 2

Marks: 2 **MCQ**

In a particular messenger application, the instances of `ChatDetails` and `UserDetails` are as follows:

ChatDetails		
SenderID	ReceiverID	Total_Text
R001	S001	1000
R002	S003	1200
R001	S002	500
R003	S004	700

UserDetails	
SenderID	Address
R001	Kolkata
R002	Delhi
R003	Kolkata

What is the output of the following SQL Query?

```
SELECT COUNT(Address) FROM ChatDetails, UserDetails GROUP BY Address;
```

a) 4
   3

b) 4
   8

c) 4

d) 3

Answer: b)
Explanation: The cartesian product (SELECT * FROM ChatDetails, UserDetails;) between the 2 instances produce the following result:

SenderID	ReceiverID	Total_Text	SenderID	Address
R001	S001	1000	R001	Kolkata
R001	S001	1000	R002	Delhi
R001	S001	1000	R003	Kolkata
R002	S003	1200	R001	Kolkata
R002	S003	1200	R002	Delhi
R002	S003	1200	R003	Kolkata
R001	S002	500	R001	Kolkata
R001	S002	500	R002	Delhi
R001	S002	500	R003	Kolkata
R003	S004	700	R001	Kolkata
R003	S004	700	R002	Delhi
R003	S004	700	R003	Kolkata

`COUNT()` on `Address` with `GROUP BY Address` will return 4, 8 for the respective counts of `‘‘Delhi"` and `‘‘Kolkata"`.
Hence, option (b) is correct.

Question 3

Marks: 2 **MCQ**

In a particular messenger application, the instance of `UserDetails` is as follows:

UserDetails	
SenderID	Address
R001	Kolkata
R002	Delhi
R003	Kolkata
R004	Kerala
R005	Agartala
R006	Mumbai

Which of the options will not be present in the output generated by the SQL query:

```
SELECT Address FROM UserDetails WHERE Address LIKE ‘%a’ OR Address LIKE ‘M%’ ;
```

a) Agartala

b) Kolkata

c) Mumbai

d) Delhi

Answer: d)

Explanation: The operator like uses patterns that are described using two special characters: percent (%) and The underscore (_). The % character matches any substring. The _ character matches any character.
The clause WHERE Address LIKE ‘%a’ matches address ‘Kolkata’, ‘Kerala’ , ‘Agartala’. The clause Address LIKE ‘M%’ matches with address ‘Mumbai’.
Hence, the option (d) is the correct answer.

Question 4

Marks: 2 **MSQ**

Which of the following statements is `incorrect`?

a) `ALTER` command is used to add\remove\modify rows to a relation.

b) `ALTER` command is used to add\remove\modify attributes to a relation.

c) `DROP` command is used to delete all data from a relation.

d) `DROP` command is used to delete a relation.

Answer: a), c)
Explanation: `ALTER` command allows addition or deletion or modification of attributes in a relation.
`DROP` command allows deletion of relations.
`DELETE` command allows deletion of data.
Hence, options (a) and (c) are the correct answer.

Question 5

Marks: 2 **MCQ**

Consider the two instances:

ChatDetails		
SenderID	ReceiverID	Total_Text
R001	S001	1000
R002	S003	1200
R001	S002	500
R003	S004	700

UserDetails	
SenderID	Address
R001	Kolkata
R002	Delhi
R003	Kolkata

Which of the following operations will generate the output given below:

SenderID	ReceiverID	Total_Text	SenderID	Address
R001	S001	1000	R001	Kolkata
R002	S003	1200	R002	Delhi
R001	S002	500	R001	Kolkata
R003	S004	700	R003	Kolkata

a) `ChatDetails NATURAL JOIN UserDetails`

b) `ChatDetails NATURAL LEFT OUTER JOIN UserDetails`

c) `ChatDetails NATURAL RIGHT OUTER JOIN UserDetails`

d) `ChatDetails EQUI JOIN UserDetails ON ChatDetails.SenderID=UserDetails.SenderID`

Answer: d)
Explanation: An `EQUI JOIN` is a join where the join condition contains an equality operator. An `EQUI JOIN` returns only the rows that have equal values for the specified column(s) and the compared column(s) twice.
The `NATURAL JOIN` automatically detects that both tables have a column named SenderID. It joins the tables on the SenderID column and eliminates one of the duplicate SenderID columns from the output.
Hence, option (d) is the correct answer.

Question 6

Marks: 2 **MCQ**

Consider the following instance of `MountainDetails(MountainName,Altitude,StateName)` relation.

MountainDetails		
MountainName	Altitude	StateName
Kangchenjunga	8586	Sikkim
Kabru	7338	Sikkim
Pandim	6888	Sikkim
Nanda Devi	7816	Uttarakhand
Trisul	7120	Uttarakhand
Kamet	7756	Uttarakhand
Sandakfu	3636	West Bengal

What will be the output of the following query?

```
SELECT MountainName, Altitude
FROM MountainDetails md1
WHERE Altitude = (
    SELECT MAX(Altitude)
    FROM MountainDetails md2
    WHERE md1.StateName = md2.StateName);
```

a)

MountainName	Altitude
Kangchenjunga	8586
Nanda Devi	7816
Sandakfu	3636

b)

MountainName	Altitude
Pandim	6888
Kamet	7756
Sandakfu	3636

c)

MountainName	Altitude
Kangchenjunga	8586

d)

MountainName	Altitude
Sandakfu	3636

Answer: a)

Explanation: The SQL query selects the MountainName and Altitude from the MountainDetails table where the Altitude is equal to the maximum Altitude for each StateName.
Hence, option a) is correct.

Question 7

Marks: 2 **MCQ**

Consider the following instance of `MountainDetails(MountainName,Altitude,StateName)` relation.

MountainDetails		
MountainName	Altitude	StateName
Kangchenjunga	8586	Sikkim
Kabru	7338	Sikkim
Pandim	6888	Sikkim
Nanda Devi	7816	Uttarakhand
Trisul	7120	Uttarakhand
Kamet	7756	Uttarakhand
Sandakfu	3636	West Bengal

What will be the output of the following query?

```
SELECT MountainName, Altitude
FROM MountainDetails
WHERE Altitude >  (
    SELECT Altitude
    FROM MountainDetails
    WHERE StateName = "Uttarakhand");
```

a)

MountainName	Altitude
Kangchenjunga	8586
Nanda Devi	7816

b)

MountainName	Altitude
Kangchenjunga	8586
Nanda Devi	7816
Kamet	7756

c)

MountainName	Altitude
Kangchenjunga	8586

d)

MountainName	Altitude
Nanda Devi	7816

Answer: c)
Explanation: The SQL query selects the MountainName and Altitude from the MountainDetails table whose Altitude is greater than the maximum Altitude found among mountains in the state of “Uttarakhand”.
Hence, option c) is the correct answer.

Question 8

Marks: 2 **MCQ**

Consider the following instance `UserDetails` of a messenger application:

UserDetails			
SenderID	ReceiverID	Total_Text	Address
R001	S001	1000	Kolkata
R002	S003	1200	Delhi
R001	S002	500	Kolkata
R003	S004	700	Kolkata
R004	S004	1700	Mumbai

Identify the correct statement(s) to get the following output:

UserDetails			
SenderID	ReceiverID	Total_Text	Address
R002	S003	1200	Delhi
R004	S004	1700	Mumbai

a)
```
SELECT * FROM UserDetails
WHERE Address AS (‘Delhi’,‘Mumbai’);
```

b)
```
SELECT * FROM UserDetails
WHERE Address IN (‘Delhi’,‘Mumbai’);
```

c)
```
SELECT * FROM UserDetails
WHERE Address FOR (‘Delhi’,‘Mumbai’);
```

d)
```
SELECT * FROM UserDetails
WHERE Address TO (‘Delhi’,‘Mumbai’);
```

Answer: b)
Explanation: Output table containing tuples whose `Address` is either Delhi or Mumbai. The IN operator allows to specify multiple values in a WHERE clause.
Hence, option b) is correct.

Question 9

Marks: 2 **MCQ**

Consider the following instance `UserDetails` of a messenger application:

UserDetails			
SenderID	ReceiverID	Total_Text	Address
R001	S001	1000	Kolkata
R002	S003	1200	Delhi
R001	S002	500	Kolkata
R003	S004	700	Kolkata
R004	S004	1700	Mumbai

Identify the correct statement to create an `index` on `SenderID` and `Address` of `UserDetails` relation named as '`View_UserDetails`'

a)
```
Create View_UserDetails
AS UserDetails(SenderID, Address);
```

b)
```
Create index View_UserDetails
AS UserDetails(SenderID, Address);
```

c)
```
Create index View_UserDetails
ON UserDetails(SenderID, Address);
```

d)
```
Create index View_UserDetails
TO UserDetails(SenderID, Address);
```

Answer: c)
Explanation: The general syntax to create any `INDEX` is

```
CREATE INDEX indexname
ON tablename (column1, column2, ...);
```

So, option (c) is correct.

Question 10

Marks: 2 **MCQ**

Consider the following instance `UserDetails` of a messenger application:

UserDetails			
SenderID	ReceiverID	Total_Text	Address
R001	S001	1000	Kolkata
R002	S003	1200	Delhi
R001	S002	500	Kolkata
R003	S004	700	Kolkata
R004	S004	1700	Mumbai

Identify the correct statement to find the `SenderID`, `ReceiverID`, and `Address` of `UserDetails` table whose `Total_Text` is in between 700 and 1200.

a)
```
SELECT SenderID, ReceiverID, Address
FROM UserDetails
WHERE Total_Text AS (700,  1200);
```

b)
```
SELECT SenderID, ReceiverID, Address
FROM UserDetails
WHERE Total_Text IN (700,  1200);
```

c)
```
SELECT SenderID, ReceiverID, Address
FROM UserDetails
WHERE Total_Text BETWEEN (700,  1200);
```

d)
```
SELECT SenderID, ReceiverID, Address
FROM UserDetails
WHERE Total_Text BETWEEN 700 AND 1200;
```

Answer: d)
Explanation: The `BETWEEN` operator selects values within a given range. The values can be numbers, text, or dates and begin and end values are included.
The correct syntax to use BETWEEN opeartor is:

```
SELECT columnname(s)
FROM tablename
WHERE columnname BETWEEN value1 AND value2;
```

The IN operator allows you to specify multiple values in a WHERE clause. The IN operator is a shorthand for multiple OR conditions.
Hence, option d) is correct.