Database Management System: Assignment 2

Total Marks : 20

January 18, 2024

Question 1

Marks: 2 **MCQ**

Consider the following table `Collections` :

Collections		
cid	item	price
11	stickers	50
13	idol	1050
23	postcard	150
2	stamp	500
10	idol	2000
14	stickers	70

How many tuples will be returned by the following query?

```
SELECT item, price FROM Collections
WHERE price>(SELECT MIN(price) FROM Collections);
```

a) 2

b) 3

c) 4

d) 5

Answer: d)
Explanation: The SQL query returns all those tuples which are associated with price more than 50. Hence, option (d) is correct.

Question 2

Marks: 2 **MCQ**

Consider the following table `Collections` :

`Collections`		
cid	item	price
11	stickers	50
13	idol	1050
23	postcard	150
2	stamp	500
10	idol	2000
14	stickers	70

Which of the following options will NOT be present in the output produced by `SELECT MAX(cid) FROM COLLECTIONS GROUP BY item;`?

a) 11

b) 13

c) 23

d) 2

Answer: a)
Explanation: The SQL query returns the highest value of `cid` for each of the `item` groups. Hence, option (a) is correct.

Question 3

Marks: 2 **MCQ**

Consider the following table `Delivery` :

Delivery		
purchaseid	deliverydate	delay
193	12/09/2010	10
183	15/10/2011	0
200	02/09/2011	0
2	30/09/2011	2
60	5/09/2010	4

What will be the output of the following SQL query?

```
SELECT COUNT(purchaseid) FROM Delivery
WHERE deliverydate LIKE '%2011' AND deliverydate NOT LIKE '30/';?
```

a) 2

b) 3

c) 5

d) 0

Answer: b)
Explanation: The SQL query returns the count of tuples having `deliverydate`s ending with 2011. Hence, option (b) is correct.

Question 4

Marks: 2 **MCQ**

Consider the following instance of table `Employee` :

Employee			
id	**lastname**	**firstname**	**age**
19	Rai	Rajeev	24
19	Singh	Rajeev	24
20	Roy	Sayan	24
21	Roy	Sayan	29

Identify the correct, "`CREATE`" statement for this table.

a)
```
CREATE TABLE Employee (
      id int NOT NULL,
      lastname varchar(255) NOT NULL,
      firstname varchar(255),
      age int,
      PRIMARY KEY (ID));
```

b)
```
CREATE TABLE Employee (
      id int NOT NULL,
      lastname varchar(255) NOT NULL,
      firstname varchar(255),
      age int,
      PRIMARY KEY (ID,lastName));
```

c)
```
CREATE TABLE Employee (
      id int,
      lastname varchar(255) NOT NULL,
      firstname varchar(255),
      age int,
      PRIMARY KEY (lastName));
```

d)
```
CREATE TABLE Employee (
      id int NOT NULL,
      lastname varchar(255) NOT NULL,
      firstname varchar(255),
      age int,
      PRIMARY KEY (firstname, lastName));
```

Answer: b)
Explanation: It is clear from the above instance that only `id` or only `lastName` cannot be a key. Hence, the `(id,lastName)` pair must be the key.
Attributes in the PRIMARY KEY cannot be NULL, and `(firstname,lastName)` pair cannot be the PRIMARY KEY.
Hence, option (b) is correct.

Question 5

Marks: 2 **MCQ**

Consider the two instances:

STATIONARY	
SL	PNAME
1	PENCIL
2	ERASER
3	SHARPENER
4	PEN

BRAND	
SL	BNAME
1	NATARAJ
4	PIERRE CARDIN
4	SMOOTHLINK

Which of the following operations will generate the following output:

SL	BNAME	SL	PNAME
1	NATARAJ	1	PENCIL
4	PIERRE CARDIN	4	PEN
4	SMOOTHLINK	4	PEN

a) `BRAND INNER JOIN STATIONARY`

b) `STATIONARY NATURAL JOIN BRAND`

c) `BRAND NATURAL RIGHT OUTER JOIN STATIONARY`

d) `STATIONARY NATURAL LEFT OUTER JOIN BRAND`

Answer: a)
Explanation: Innerjoin is a join where the join returns only the rows that have equal values for the specified column(s) and the compared column(s) present twice.

Question 6

Marks: 2 **MCQ**

Consider the following relation instance:

Number_Tab	
Num1	Num2
4	5
5	7
6	7
7	8
3	4

Both attributes `Num1` and `Num2` are integers and do not have null values. `Num1` is the primary key of the table and `Num2` is the foreign key of the same table, `Number_Tab` and references with `on delete cascade` constraints. A tuple `(Num1, Num2)` will be in the table only if Num1 $\leq$ Num2. Which of the following is possible if the tuple `(5, 7)` is deleted from the table?

a) The deletion of `(5, 7)` will be prohibited.

b) Tuple `(4, 5)` and `(3, 4)` also will be deleted.

c) Tuple `(6, 7)` and `(7, 8)` also will be deleted.

d) Only tuple `(7, 8)` will be deleted.

Answer: b)
Explanation: In the `Number_Tab(Num1, Num2)`, where `Num1` is the primary key, and `Num2` is the foreign key which is referencing the primary key `Num1` of its own relation.
Now if we delete tuple (5,7) then tuple (4,5) should also be deleted (as 5 in the tuple (4,5) references to 5 in the tuple (5,7) which no longer exists; hence, the referencing tuple should also be deleted), and as (4,5) is deleted, tuple (3,4) should also be deleted for the same reason. Therefore, in total, 3 rows have to be deleted if the tuple (5,7) is deleted.
Hence, option b) is correct.

Question 7

Marks: 2 **MSQ**

Suppose a bank wants to make a view consisting of the names of customers having loan in 'MUMBAI' branch with the loan amount being more than equal to 50000 but less than equal to 70000.

- loan(loan_number, branch_name, amount)
- borrower(customer_name, loan_number)

Identify the correct query from the following. Primary keys are underlined in the schema.

a)
```
CREATE VIEW v1 AS
SELECT customer_name
FROM loan, borrower
WHERE branch_name = 'MUMBAI'
AND loan.loan_number = borrower.loan_number
AND amount >= 50000 AND amount <= 70000;
```

b)
```
CREATE VIEW v1 AS
SELECT customer_name
FROM loan
WHERE branch_name = 'MUMBAI'
AND amount >= 50000 AND amount <= 70000;
```

c)
```
CREATE VIEW v1 AS
SELECT customer_name
FROM loan, borrower
WHERE branch_name = 'MUMBAI'
AND loan.loan_number = borrower.loan_number
AND amount BETWEEN 50000 AND 70000;
```

d)
```
CREATE VIEW v1 AS
SELECT customer_name
FROM loan, borrower
WHERE branch_name = 'MUMBAI'
AND amount >= 50000, amount <= 70000;
```

Answer: a), c)

Explanation: The Syntax for creating a VIEW in SQL is:

```
CREATE VIEW view name AS
SELECT column1, column2...
FROM table name
WHERE condition;
```

Moreover, the joining will be used to avoid duplicate values in the view. So, from the above only option a) and c) are satisfying the requirements and rest are incorrect.
Option b) generates error,"CUSTOMER_NAME": invalid identifier since borrower table is missing in the FROM clause.
Option d) also generates error – SQL command not properly ended since the conditions are combined through (,) instead of AND operator.

Question 8

Marks: 2 **MCQ**

Consider the following instance of `StudentDetails(StudName, DeptName, Address, Age)` relation.

StudentDetails			
StudName	DeptName	Address	Age
Ayush	CSE	Kolkata	28
Priya	CSE	Hyderabad	26
Ankush	IT	Kolkata	30
Rumki	IT	Hyderabad	25
Sujit	ECE	Bangalore	24
Sayan	IEE	Mumbai	28

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

StudentDetails			
StudName	DeptName	Address	Age
Ayush	CSE	Kolkata	28
Ankush	IT	Kolkata	30
Rumki	IT	Hyderabad	25
Sayan	IEE	Mumbai	28

a)
```
SELECT * FROM StudentDetails
WHERE Age>=28;
```

b)
```
SELECT * FROM StudentDetails
WHERE DeptName='IT';
```

c)
```
SELECT * FROM StudentDetails
WHERE Age>=28 AND DeptName='IT';
```

d)
```
SELECT * FROM StudentDetails
WHERE Age>=28 OR DeptName='IT';
```

Answer: d)
Explanation: Output table containing tuples whose `Age` is greater than or equal to 28 or `DeptName='IT'`.
Hence, option d) is correct.

Question 9

Marks: 2 **MCQ**

Consider the following instance of `StudentDetails(StudName, DeptName, Address, Age)` relation.

StudentDetails			
StudName	DeptName	Address	Age
Ayush	CSE	Kolkata	28
Priya	CSE	Hyderabad	26
Ankush	IT	Kolkata	30
Rumki	IT	Hyderabad	25
Sujit	ECE	Bangalore	24
Sayan	IEE	Mumbai	28

Identify the correct SQL command to find the average age of students in the CSE department.

a) `SELECT avg(Age) from StudentDetails;`

b) `SELECT avg(Age) from StudentDetails where DeptName='CSE';`

c) `SELECT * from StudentDetails where DeptName='CSE' AND avg(Age);`

d) `SELECT * from StudentDetails where DeptName='CSE' OR avg(Age);`

Answer: b)
Explanation: As per SQL syntax, avg(Age) is used to find the average age, in which condition used DeptName='CSE' to find the students whose department name is 'CSE'.
Hence, option b) is correct.

Question 10

Marks: 2 **MCQ**

Consider the following instance of `StudentDetails(StudName, DeptName, Address, Age)` relation.

StudentDetails			
StudName	DeptName	Address	Age
Ayush	CSE	Kolkata	28
Priya	CSE	Hyderabad	26
Ankush	IT	Kolkata	30
Rumki	IT	Hyderabad	25
Sujit	ECE	Bangalore	24
Sayan	IEE	Mumbai	28

Identify the correct statement(s) to find the `StudName` and Address whose `Age` is greater than the `Age` of all students in the 'IT' department.

a)
```
SELECT StudName, Address from StudentDetails
        where Age >  (select Age
                        from StudentDetails
                        where DeptName = 'IT');
```

b)
```
SELECT StudName, Address from StudentDetails
        where Age > all (select Age
                        from StudentDetails
                        where DeptName = 'IT');
```

c)
```
SELECT StudName, Address from StudentDetails
        where Age >  (select max(Age)
                        from StudentDetails
                        where DeptName = 'IT');
```

d)
```
SELECT StudName, Address from StudentDetails
        where Age > for (select Age
                        from StudentDetails
                        where DeptName = 'IT');
```

Answer: b)
Explanation: The `all` operator returns TRUE if ALL of the subquery values meet the condition.
Hence, option b) is correct.