DBT - MCQ Test - 1

Feb19/ DBT/ 001

Database Technologies

Diploma in Advance Computing

February 2019

1. **ABC institute want to store all employee details like (employeeID, fullName, DOB, and his salary) in the employee table. Which of the following statement he may issue? Note: Every employeeID for all employees must be unique and not null.**
2. create table employee ( employeeID int primary key, fullName varchar(45), dob datetime, salary int);
3. create table employee ( employeeID int, fullName varchar(45), dob datetime, salary int, constraint pk\_employee primary key (employeeID));
4. create table employee ( employeeID int unique not null, fullName varchar(45), dob datetime, salary int);
5. **All of the above.**
6. **Consider the above employee table. The data entry operator wants to insert employee details for employee. Which of the following statement he may issue?**
7. insert into employee values(1, 'saleel', '1970-05-12', 4500.456);
8. insert into employee(employeeid, fullName, dob, salary) values(2,'ruhan', '1987-05-12', 5500.756);
9. insert into employee (employeeid, fullName, dob, salary) select 3, 'sharmin', '1999-11-18', 6100.568;
10. **All of the above.**
11. **The operator has inserted 3 record in the employee table. The records are as follows.**

|  |  |  |  |
| --- | --- | --- | --- |
| employeeID | fullName | DOB | salary |
| 1 | saleel | 1970-05-12 | 4500.456 |
| 2 | ruhan | 1987-05-12 | 5500.756 |
| 3 | sharmin | 1999-11-18 | 6100.568 |

**What will the output, if we issue the following select statement?**

**SELECT \* from employee;**

|  |  |  |  |
| --- | --- | --- | --- |
| employeeID | fullName | DOB | salary |
| 1 | saleel | 1970-05-12 | 4500.456 |
| 2 | ruhan | 1987-05-12 | 5500.756 |
| 3 | sharmin | 1999-11-18 | 6100.568 |

a)

b)

|  |  |  |  |
| --- | --- | --- | --- |
| employeeID | fullName | DOB | salary |
| 1 | saleel | 1970-05-12 | 4500.45 |
| 2 | ruhan | 1987-05-12 | 5500.75 |
| 3 | sharmin | 1999-11-18 | 6100.56 |

**c)**

|  |  |  |  |
| --- | --- | --- | --- |
| **employeeID** | **fullName** | **DOB** | **salary** |
| **1** | **saleel** | **1970-05-12** | **4500** |
| **2** | **ruhan** | **1987-05-12** | **5501** |
| **3** | **sharmin** | **1999-11-18** | **6101** |

1. All of the above
2. **Consider the above employee table with data. What value will be return by below query?**

**Select count(\*) from employee;**

1. 1
2. 2
3. **3**
4. none of the above
5. **Which standard SQL clause is used to restrict the rows returned by a query?**
6. AND
7. **WHERE**
8. HAVING
9. FROM
10. **What SQL statement will you issue to get the employee detail whose employeeID is 1. Assume the employeeID is not 0 and not negative.**
11. Select \* from employee where employeeID = 1;
12. Select \* from e where employeeid in (1);
13. Select \* from e where employeeid order by employeeID limit 1;
14. **All of the above**
15. **Examine the structure of the EMP and DEPT table.**

**EMP**

**empno int primary key**

**ename varchar (45)**

**mgr int**

**sal int**

**deptno int**

**DEPT**

**deptno int**

**dname varchar (45)**

**Which SQL statement produces the ename, dname of all the employees who earn more than 10000?**

1. **select ename, dname from emp inner join dept using(deptno) where sal > 10000;**
2. select ename, dname from emp, dept join on emp.deptno = dept.deptno where sal> 10000;
3. select ename, dname from emp, dept where sal > 10000;
4. select ename, dname from emp inner join dept where sal > 10000;
5. **Examine the structure of the EMP and NewEMP table.**

**EMP**

**ID int primary key**

**firstName varchar(45)**

**lastName varchar(45)**

**NewEMP**

**ID int primary key**

**fullName varchar(1005)**

**EMP table has following record.**

|  |  |  |
| --- | --- | --- |
| **ID** | **firstName** | **lastName** |
| **1** | **Saleel** | **Bagde** |
| **2** | **Ruhan** | **Bagde** |
| **3** | **Sharmin** | **Bagde** |

**Which of the following statement inserts the record in NewEMP table from EMP whose employee ID is 1 and after concatenating the firstName and lastName.**

1. Insert into NewEMP value (select id, concat(firstName, ' ', lastName) from EMP where id=1);
2. Insert into NewEMP values (select id, concat(firstName, ' ', lastName) from EMP where id=1);
3. **Insert into NewEMP (select id, concat(firstName, ' ', lastName) from EMP where id=1);**
4. Both A or B
5. **Examine the structure of the EMP table.**

**EMP**

**empno int primary key**

**ename varchar(45)**

**job varchar(45)**

**mgr int**

**salary int**

**deptno int**

**Which statement shows the job name, minimum salary, maximum salary paid to the employee who is working as ‘MANAGER’?**

1. Select job, min(sal), max(sal) from EMP where job='manager';
2. Select job, min(sal), max(sal) from EMP where job in ('manager');
3. Select job, min(sal), max(sal) from EMP where job like 'manager';
4. **All of the above**
5. **Which statement will delete the record from the EMP table whose employeeID is maximum?**
6. delete from emp where employeeID = (select max(employeeID) from emp);
7. delete from emp where employeeID = max(employeeID);
8. delete from emp where employeeID = max(employeeID) group by employeeID;
9. **None of the above statement is valid.**
10. **Examine the structure of the EMP table with data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| employeeID | fullName | DOB | salary | comm |
| 1 | saleel | 1970-05-12 | 4500.456 | 100 |
| 2 | ruhan | 1987-05-12 | 5500.756 | Null |
| 3 | sharmin | 1999-11-18 | 6100.568 | Null |

**Which statement shows all the employee details who are not getting commission.**

1. Select \* from EMP where comm <=> null;
2. Select \* from EMP where comm is null;
3. **Both A and B**
4. None of the above.
5. **Examine the structure of the EMP table.**

**EMP**

**empno int primary key**

**ename varchar(45) not null**

**job varchar(45)**

**mgr int**

**salary int**

**deptno int**

**You need to create a view call Insert\_Employee\_View than allows the user to insert rows through the view. Which SQL statement allows the user to insert rows?**

1. Create view Insert\_Employee\_View as SELECT empno, job, sal, deptno from EMP;
2. Create view Insert\_Employee\_View as SELECT ename, job, sal, deptno from EMP;
3. **Create view Insert\_Employee\_View as SELECT empno, ename, job, sal from EMP;**
4. None of the above.
5. **Examine the following SQL statement given by the developer.**

**Create table Person (id int primary key, name varchar (45) NOT NULL, mgr int references Person);**

**Create or replace view personView as SELECT \* from Person;**

**Which or the following SQL statements can be issued through the PersonView?**

1. Insert into personView values(1,’Saleel’, 1);
2. Update personView set name = ‘sharmin’ where id=1;
3. Delete from personView where id=1;
4. **All of the above**
5. **Examine the structure of the EMP and DEPT tables.**

**EMP**

**empno int primary key**

**ename varchar(45) not null**

**job varchar(45)**

**mgr int**

**salary int**

**deptno int**

**DEPT**

**deptno int primary key**

**dname varchar(45)**

**Which two SQL statements produce the employee name, department name of all the employees who earn more than 3000? (Chose two)**

1. **Select ename, dname from EMP e join DEPT D using(deptno) where sal > 3000;**
2. Select ename, dname from EMP, DEPT where sal > 3000;
3. Select emp.ename, dname from emp e join dept d using(e.deptno) where sal > 3000;
4. **Select ename, dname from EMP e, DEPT d where e.deptno = d.deptno and sal > 3000;**
5. **The STUDENT GRADES table has these columns:**

**StudentID int**

**Semister\_end date**

**GPA int**

**The register has asked for a report on the average(GPA) for all students enrolled during semester that end in the year 1981. Which statement accomplishes this?**

1. Select avg(GPA) from STUDENT GRADES where Semister\_end > '1981-01-01' and Semister\_end < '1981-12-31';
2. **Select avg(GPA) from STUDENT GRADES where year(Semister\_end) = 1981;**
3. **Select avg(GPA) from STUDENT GRADES where Semister\_end between '1981-01-01' and '1981-12-31';**
4. None of the above.
5. **Examine the structure of the EMP and newEMP tables:**

**EMP**

**ID int primary key**

**firstName varchar(45)**

**lastName varchar(45)**

**phone varchar(10)**

**emailID varchar(128)**

**newEMP**

**ID int primary key**

**firstName varchar(45)**

**lastName varchar(45)**

**phone varchar(10)**

**emailID varchar(128)**

**The EMP table has 10000 records and newEMP table is empty. The developer has been told to transfer the records from EMP table to newEMP table. Which of the following statements he will execute?**

1. Insert into newEMP values (select \* from EMP);
2. Insert into newEMP as select \* from EMP;
3. **Insert into newEMP select \* from EMP;**
4. None of the above.
5. **Which is true about aggregate functions?**
6. You can use aggregate function in any clause of a SELECT statement.
7. You can use aggregate function only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.
8. **You can pass column name, expressions, constants as a parameters to an aggregate function.**
9. None of the above.
10. **EMP table has 20 records. What will the output, if we issue the following SQL statement?**

**SELECT sum(1000) from EMP;**

1. 20
2. 1000
3. **20000**
4. Error
5. **EMP table has 20 records. What will the output, if we issue the following SQL statement?**

**SELECT count(1) from EMP;**

1. 1
2. 10
3. **20**
4. Error
5. **EMP table has 20 records. What will the output, if we issue the following SQL statement?**

**SELECT count(NULL) from EMP;**

1. 1
2. 2
3. **0**
4. NULL
5. **Examine the structure of the EMP and TAX tables:**

**EMP**

**ID int primary key**

**firstName varchar(45)**

**lastName varchar(45)**

**salary int**

**TAX**

**min\_Salary int**

**max\_salary int**

**tax\_percentage int**

**You need to find the percentage tax applicable for each employee. Which SQL statement would you use?**

1. **Select ID, salary, tax\_percentage from EMP, TAX where salary between min\_salary and max\_salary;**
2. Select ID, salary, tax\_percentage from EMP, TAX where salary > min\_salary, max\_salary;
3. Select ID, salary, tax\_percentage from EMP, TAX where min(salary) = min\_salary and max(salary) = max\_salary;
4. You cannot find the information because there is no common column between the two tables.
5. **Which two statements are true regarding the ORDER BY clause?(Choose two)**
6. **The sort is in ascending order by default**
7. The sort is in descending order by default
8. The ORDER BY clause must precede the WHERE clause.
9. **The ORDER BY clause comes last in the SELECT statement.**
10. **The customers table has these columns:**

**customer\_ID int primary key**

**customer\_name varchar(45)**

**city varchar(44)**

**Which two statements find the number of customers? (Chose two)**

1. Select TOTAL(\*) from customers;
2. **Select COUNT(\*) from customers;**
3. **Select COUNT(customer\_ID) from customers;**
4. Select COUNT(customers) from customers;
5. **Examine the structure of the STUDENT and STUDENT\_PHONE tables:**

**STUDENT**

**ID INT primary key**

**nameFirst varchar(45)**

**nameLast varchar(45)**

**STUDENT\_PHONE**

**ID int primary key**

**studentID int**

**number int**

**Which of the following statement will produce the result that displays student ID, nameFirst, nameLast and his phone number?**

1. **select student.ID, nameFirst, nameLast, number from student, student\_phone where student.ID = student\_phone.studentID;**
2. select student.ID, nameFirst, nameLast, number from student, student\_phone where student.ID in student\_phone.studentID;
3. select student.ID, nameFirst, nameLast, number from student, student\_phone where student.ID = = student\_phone.studentID;
4. None of the above.
5. **Examine the structure of the STUDENT and STUDENT\_ADDRESS tables:**

**STUDENT**

**ID INT primary key**

**nameFirst varchar(45)**

**nameLast varchar(45)**

**STUDENT\_ADDRESS**

**ID int primary key**

**studentID int**

**address varchar(128)**

**Which of the following statement will produce the result that displays student ID, nameFirst, nameLast and his address?**

1. select S.ID, nameFirst, nameLast, address from STUDENT S INNER JOIN STUDENT\_ADDRESS A ON S.ID = A.studentID;
2. select S.ID, nameFirst, nameLast, address from STUDENT S, STUDENT\_ADDRESS A where S.ID = A.studentID;
3. select S.ID, nameFirst, nameLast, address from STUDENT S JOIN STUDENT\_ADDRESS A ON S.ID = A.studentID;
4. **All of the above.**
5. **Examine the structure of the Modules table:**

**Modules**

**ID(PK)**

**name**

**duration**

**Which of the following statement will produce the result that displays all module names?**

1. SELECT name from MODULES;
2. SELECT name from (SELECT \* from MODULES);
3. SELECT name from (SELECT \* from MODULES) M;
4. **Both A and C.**
5. **You want to display the details of all employees whose last name is of 4 characters. Which statement will list all the employees whose last is of 4 characters? (Choose two)**
6. SELECT firstName, lastName from EMP where lastName = ASCII(4);
7. **SELECT firstName, lastName from EMP where length(lastName) = 4;**
8. **SELECT firstName, lastName from EMP where lastName like '\_\_\_\_';**
9. SELECT firstName, lastName from EMP where length(lastName) is 4;
10. **Based on the EMP table below, rename the EMP table to EMPLOYEE.**

**CREATE TABLE EMP**

**(id number(10) not null,**

**name varchar(50) not null,**

**CONSTRAINT emp\_pk PRIMARY KEY(id));**

1. RENAME table EMP to EMPLOYEE;
2. ALTER table EMP RENAME to EMPLOYEE;
3. **Both A and B**
4. None of the above
5. **Based on the employees table below, add a column called salary that is a number(6) datatype.**

**CREATE TABLE employees**

**( employee\_number int(10) not null,**

**employee\_number varchar(50) not null,**

**department\_id int(10),**

**CONSTRAINT employees\_pk PRIMARY KEY(employee\_number)**

**);**

1. ALTER TABLE employees ADD salary int(6);
2. ALTER TABLE employees ADD column salary int(6);
3. **Both A and B.**
4. None of the above.
5. **Which of the following queries displays the sum of all employee salaries for those employees not getting commission, for each job, including only those sums greater than 2500?**
6. select job, sum(sal) from emp where sum(sal) > 2500 and comm is null;
7. **select job, sum(sal) from emp where comm is null group by job having sum(sal) > 2500;**
8. select job, sum(sal) from emp where sum(sal) > 2500 and comm is null group by job;
9. select job, sum(sal) from emp group by job having sum(sal) > 2500 and comm is not null;
10. **You need to design a student campus database that contains several table storing student information and student\_phone.**

**Student table**

**ID int primary key**

**nameFirst varchar(45)**

**nameLast varchar(45)**

**Create a table student\_phone (Id int (PK), studentID int, number varchar (10)) and you need to create foreign key on studentID. Which statement will create a student\_phone table with studentID as foreign key? (Choose two)**

1. create table student\_phone(id int primary key, studentID int foreign key, number varchar(10));
2. **create table student\_phone(id int primary key, studentID int, number varchar(10), constraint fk\_studentID foreign key(studentID) references student(id));**
3. **create table student\_phone(id int primary key, studentID int, number varchar(10), foreign key(studentID) references student(id));**
4. None of the above.
5. **You need to design a student campus database that contains several table storing student information and student\_address. Which statement will create 1:1 relation between student and student\_address tables?**

**Student table**

**ID int primary key**

**nameFirst varchar(45)**

**nameLast varchar(45)**

**student\_address table contains (Id int (PK), studentID int, address varchar(128)) and you need to create 1:1 relation between student and student\_address table using studentID from student\_address table.**

1. **Create table student\_address (id int primary key, studentID int unique not null, address varchar (45), foreign key(studentID) references student(id));**
2. Create table student\_address (id int primary key, studentID int not null, address varchar(45), foreign key(studentID) references student(id));
3. Both A and B
4. None of the above.
5. **You need to design a student campus database that contains several table storing student information. The management wants to know that, who are the student who are born in the year 1982. Which statement should be issued to get the details? (Choose all applicable)**
6. select \* from student where year(dob) = 1982;
7. select \* from student where extract(year from dob) = 1982;
8. select \* from student where dob like '1982%';
9. None of the above.
10. **You need to design a student campus database that contains several table storing student information. The management wants to know that, who are the student whose nameFirst starts with the letter 'R'. Which statement should be issued to get the details? (Choose all applicable)**
11. **select \* from student where namefirst like 'R%';**
12. **select \* from student where substr(namefirst,1,1) = 'R';**
13. **select \* from student where mid(namefirst,1,1) = 'R';**
14. None of the above.
15. **Examine the structure of the Book and Author tables:**

**Book table**

**book\_id int primary key,**

**title varchar(45)**

**Author table**

**author\_id int primary key,**

**name varchar(45)**

**What SQL statement will you issue if you want to create M:M relation between Book table and Author table? (Choose all applicable)**

1. **create table book\_author (BID int, AID int, foreign key(BID) references book(book\_id), foreign key(AID) references author(author\_id));**
2. **create table book\_author (book\_ID int, author\_ID int, foreign key(book\_ID) references book(book\_id), foreign key(author\_ID) references author(author\_id));**
3. create table book\_author (book\_ID int, author\_ID int, foreign key(book\_ID), foreign key(author\_ID));
4. None of the above.
5. **Examine the structure of the Book, Author, and Book\_author tables:**

**Book Table**

|  |  |
| --- | --- |
| **Book\_id** | **title** |
| **1** | **MongoDB** |
| **2** | **Redis** |

**Author Table**

|  |  |
| --- | --- |
| **Author\_id** | **Name** |
| **1** | **Raju** |
| **2** | **Jones** |
| **3** | **Smith** |
| **4** | **Rahul** |

**Book\_Author table**

|  |  |
| --- | --- |
| **Book\_id** | **Author\_id** |
| **1** | **1** |
| **1** | **2** |
| **2** | **1** |
| **2** | **3** |

**What SQL statement will you issue to find the author details who have not written any book? (Choose all applicable)**

1. **select \* from author where not exists (select \* from book\_author where author.author\_id = book\_author.author\_id);**
2. **select \* from author where author.author\_id not in (select author\_id from book\_author);**
3. select \* from author where author.author\_id in (select author\_id from book\_author);
4. None of the above.
5. **Examine the structure of the CITY table:**

**City table**

**ID**

**Name**

**Countrycode**

**District**

**Population**

**Query the names of all American cities in CITY with populations larger than 120000. The CountryCode for America is USA.**

1. **select name from city where countrycode = 'usa' and population > 120000;**
2. select name from city where countrycode = 'USA' and population >= 120000;
3. select name from city where countrycode = 'usa' and population >in 120000;
4. None of the above.
5. **Find the difference between the total number of CITY entries in the table and the number of distinct CITY entries in the table. (Choose two)**

**Weather table**

|  |  |  |
| --- | --- | --- |
| **ID** | **City** | **State** |
| **1** | **Pune** | **MH** |
| **2** | **Mumbai** | **MH** |
| **3** | **Pune** | **MH** |
| **4** | **Nagpur** | **MH** |
| **5** | **Nasik** | **MH** |
| **6** | **Pune** | **MH** |
| **7** | **Nasik** | **MH** |
| **8** | **Nasik** | **MH** |

1. **SELECT count(city) - count(distinct city) FROM weather;**
2. SELECT count(city) - distinct count(city) FROM weather;
3. **SELECT count(city) - count(distinctrow city) FROM weather;**
4. SELECT Total(city) - count(distinct city) FROM weather;
5. **Examine the structure of the CITY table:**

**City table**

**ID**

**Name**

**Countrycode**

**District**

**Population**

**Which of the following statement will display the structure of the CITY table?**

1. DESCRIBT CITY
2. SHOW COLUMNS FROM CITY
3. EXPLAIN CITY
4. **All of the above.**
5. **The EMPLOYEES table contains these columns:**

**EMPLOYEE\_ID int(4) LAST\_NAME VARCHAR (25) JOB\_ID VARCHAR(10)**

**You want to search for strings that contain 'SA\_' in the JOB\_ID column. Which SQL statement do you use?**

1. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id = '%SA\_';
2. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\_';
3. SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA\_' ESCAPE "/";
4. **SELECT employee\_id, last\_name, job\_id FROM employees WHERE job\_id LIKE '%SA/\_%' ESCAPE '/';**
5. **You own a table called STUDENT with this table structure:**

**STUDENT\_ID INT Primary Key, FIRST\_NAME VARCHAR (25), LAST\_NAME VARCHAR(25), DOB DATE**

**What happens when you execute this DELETE statement?**

1. You get an error because of a primary key violation.
2. The data and structure of the STUDENT table are deleted.
3. You get an error because the statement is not syntactically correct.
4. **The data in the STUDENT table is deleted but not the structure.**
5. **You need to create a table named ORDERS that contains four columns:**

**1. an ORDER\_ID column of number data type**

**2. a CUSTOMER\_ID column of number data type**

**3. an ORDER\_STATUS column that contains a character data type**

**4. a DATE\_ORDERED column to contain the date the order was placed**

**When a row is inserted into the table, if no value is provided for the status of the order, the value ‘PENDING’ should be used instead. Which statement accomplishes this?**

1. CREATE TABLE orders (order\_id INT(10), customer\_id INT(8), order\_status VARCHAR(10) DEFAULT 'PENDING', date\_ordered VARCHAR);
2. **CREATE TABLE orders (order\_id INT(10), customer\_id INT(8), order\_status VARCHAR(10) DEFAULT 'PENDING', date\_ordered DATE);**
3. CREATE OR REPLACE TABLE orders (order\_id INT(10),customer\_id INT(8),order\_status VARCHAR(10) DEFAULT 'PENDING', date\_ordered DATE);
4. CREATE TABLE orders (order\_id INT(10),customer\_id INT(8),order\_status NUMBER(10) DEFAULT 'PENDING',date\_ordered DATE);
5. **Evaluate this SQL statement:**

**SELECT e.EMPLOYEE\_ID, e.LAST\_NAME, e.DEPARTMENT\_ID, d.DEPARTMENT\_NAME FROM EMPLOYEES e, DEPARTMENTS d WHERE e.DEPARTMENT\_ID = d.DEPARTMENT\_ID;**

**In the statement, which capabilities of a SELECT statement are performed?**

1. **selection, projection, join**
2. selection, intersection, join
3. difference, projection, product
4. difference, projection, join
5. **Which SQL statement generates the alias Date of Birth for the column DOB?**
6. SELECT nameFirst, nameLast, DOB 'Date of Birth' FROM student;
7. SELECT nameFirst, nameLast, DOB Date of Birth FROM student;
8. SELECT nameFirst, nameLast, DOB "Date of Birth" FROM student;
9. **Both A and C**
10. **Examine the structure of the STUDENT table:**

**STUDENT\_ID INT Primary Key, nameFirst VARCHAR(25), nameLast VARCHAR(25)**

**Which SQL statements insert a row into the table?**

1. INSERT INTO STUDENT (nameFirst, nameLast) VALUES( 'John', 'Smith');
2. INSERT INTO STUDENT VALUES ( NULL, 'John', 'Smith');
3. **INSERT INTO STUDENT (student\_id , nameFirst, nameLast) VALUES ( 1000, 'John', ' ');**
4. INSERT INTO STUDENT (nameFirst, nameLast, student\_id) VALUES ( 1000, 'John', 'Smith');
5. **Which SELECT statement should you use to extract the year from the DOB column in student table?**
6. select year(DOB)
7. select extract(year from DOB)
8. select date\_format(DOB, '%Y');
9. **All of the above**
10. **In which two cases would you use an outer join?**
11. Only when the tables have a primary key-foreign key relationship.
12. The tables being joined have only matched data.
13. **The tables being joined have both matched and unmatched data.**
14. None of the above.
15. **Evaluate the SQL statement:**

**SELECT TRUNCATE(45.936, 2);**

**Which values are displayed?**

1. 45
2. **45.93**
3. 45.94
4. 46.00
5. **Evaluate the SQL statement:**

**SELECT ROUND(156.936, -2);**

**Which values are displayed?**

1. 156.00
2. 156.93
3. **200**
4. Error
5. **You need to design a library management database that contains several table storing BOOK\_ISSUE details.**

**BOOK\_ISSUE table**

**BookID int primary key,**

**CustomerID int**

**Issue\_Date date,**

**Return\_date date**

**Today, if the book was issued to the customer, the return date will 7 days after the issue date. Find out what will be the return date. (Choose two)**

1. **select Issue\_Date, Issue\_Date + interval 7 day as "Return date" from book\_Issue;**
2. select Issue\_Date, Issue\_Date + interval 7 days as "Return date" from book\_Issue;
3. **select Issue\_Date, addDate(Issue\_Date, interval 7 day) as "Return date" from book\_Issue;**
4. select Issue\_Date, addDate(Issue\_Date, interval 7 days) as "Return date" from book\_Issue;
5. **You need to design a Weather management that contains several table storing WEATHER details**

**Weather table:**

**id int primary key,**

**daytime datetime,**

**temp int**

**Print date of max(temp), min(temp)**

**Print time of max(temp), min(temp)**

**Month wise hottest day**

**Week wise hottest day.**

**Year wise hottest day**

**Yearly avg temp**

**Monthly avg temp**

1. **Q**
2. **Q**
3. **Q**
4. **Q**
5. **Q**
6. **Q**
7. **Q**
8. **Q**
9. **Q**
10. **Q**
11. **Q**
12. **Q**
13. **Q**