```
--- SQL Class 2 Assignment

--- Q1. Create Database as SQL practice and use it for further questions.
--- Ans1
create database Class_2;
use Class_2;

/* Q2. Create a table named "Students" with the following columns: StudentID (int),
FirstName (varchar), LastName (varchar), and Age (int). Insert at least three records into
the table. */
--- Ans2
create table Students
(
        StudentID int,
   FirstName varchar (20),
   Age int
   );
insert into Students values
(1,'Ram',20),
(2,'Shyam',21),
(3,'Arjun',22);

/*Q3. Update the age of the student with StudentID 1 to 21. Delete the student with
StudentID 3 from the "Students" table. */
--- Ans3
update Students set Age=21 where StudentID=1;
delete from Students where StudentID=3;

--- Q4. Retrieve the first names and ages of all students who are older than 20.
--- Ans4
select FirstName,Age from Students where age>20;
```

```
--- Q5. Delete records from the same table where age<18.
--- Ans5
delete from Students where age<18;

/* Q6. Create a table named "Customers" with the following columns and constraints:
CustomerID (int) as the primary key.
FirstName (varchar) not null.
LastName (varchar) not null.
Email (varchar) unique.
Age (int) check constraint to ensure age is greater than 18. */
--- Ans6
create table Customers
(
        CustomerID int primary key,
   FirstName varchar (20) not null,
   LastName varchar (20) not null,
   Email varchar (100) unique,
   Age int check (Age > 18)
   );

/* Q7. You have a table named "Orders" with columns: OrderID (int), CustomerID (int),
OrderDate (date), and TotalAmount (decimal). Create a foreign key constraint on the
"CustomerID" column referencing the "Customers" table. */
--- Ans7
create table Orders
(
        OrderID int,
   CustomerID int,
   OrderDate date,
   TotalAmount decimal
```

```
   );
alter table Orders add Constraint CustID_fk FOREIGN key (CustomerID) references Customers
(CustomerID);


/* Q8. Create a table named "Employees" with columns:
EmployeeID (int) as the primary key.
FirstName (varchar) not null.
LastName (varchar) not null.
Salary (decimal) check constraint to ensure salary is between 20000 and 100000. */
--- Ans8
create table if not exists Employees
(
        EmployeeID int,
   FirstName varchar(20) not null,
   LastName varchar(20) not null,
   Salary decimal,
   Constraint EmpID_pk primary key (EmployeeID),
   Constraint Salary_chk check (Salary between 20000 and 100000)
   );


/* Q9. Create a table named "Books" with columns:
BookID (int) as the primary key.
Title (varchar) not null.
ISBN (varchar) unique. */
--- Ans9
create table if not exists Books
(
        BookID int,
   Title varchar (50) not null,
   ISBN varchar (100),
   Constraint B_ID_pk primary key (BookID),
```

```
  Constraint isbn_unq Unique (ISBN)
  );
```

/* Q10. Consider a table named "Employees" with columns: EmployeeID, FirstName, LastName, and Age. Write an SQL query to retrieve the first name and last name of employees who are older than 30. */

--- Ans10

```
select FirstName,LastName from Employees where Age > 30;
```

/* Q11. Using the same "Employees" table, write an SQL query to retrieve the first name, last name, and age of employees whose age is between 20 and 30. */

--- Ans11

```
select FirstName,LastName,Age from Employees where Age between 20 and 30;
```

/* Q12. Given a table named "Products" with columns: ProductID, ProductName, Price, and InStock (0- for out of stock, 1- for in stock). Write an SQL query to retrieve the product names and prices of products that are either priced above $100 or are out of stock. */

--- Ans12

```
select ProductName , Price from Products where Price>100 or InStock = 0;
```

/* Q13. Using the "Products" table, write an SQL query to retrieve the product names and prices of products that are in stock and priced between 50 and 150. */

--- Ans13

```
select ProductName , Price from Products where InStock = 1 and Price between 50 and 150;
```

/* Q14. Consider a table named "Orders" with columns: OrderID, OrderDate, TotalAmount, and CustomerID. Write an SQL query to retrieve the order IDs and total amounts of orders placed by customer ID 1001 after January 1, 2023, or orders with a total amount exceeding $500. */

--- Ans14

```
select OrderID , TotalAmount from Orders where CustomerID=1001 and OrderDate > '2023-01-01' or TotalAmount>500;

/* Q15. Retrieve the ProductName of products from the "Products" table that have a price
between $50 and $100. */
--- Ans15
select ProductName from Products where Price between 50 and 150;

/* Q16. Retrieve the names of employees from the "Employees" table who are both from the
"Sales" department and have an age greater than 25, or they are from the "Marketing"
department. */
--- Ans16
select Name from Employees where Department = 'Sales' and Age>25 or Department = 'Marketing';

/* Q17. Retrieve the names of customers from the "Customers" table who are not from the
city 'New York' or 'Los Angeles'. */
--- Ans17
select Name from Customers where City <> 'New York' or City <> 'Los Angeles';

/* Q18. Retrieve the names of employees from the "Employees" table who are either from the
"HR" department and have an age less than 30, or they are from the "Finance"
department and have an age greater than or equal to 35. */
--- Ans18
select Name from Employees where Department = 'HR' and Age<30 or Department = 'Finance' and Age>=35;

/* Q19. Retrieve the names of customers from the "Customers" table who are not from the
city 'London' and either have a postal code starting with '1' or their country is not 'USA'. */
--- Ans19
select Name from Customers where City <> 'London' and Postal_Code like '1%' or Country <> 'USA';
```