

# Assignment Day 4

## \*\*1. Constraints:\*\*

### \*\*Question:\*\*

1. Define a table named "Employees" with the following columns:

- EmployeeID (integer) as the primary key.
- FirstName (string) with a maximum length of 50 characters, not allowing NULL values.
- LastName (string) with a maximum length of 50 characters, allowing NULL values.
- Department (string) with a maximum length of 50 characters, not allowing NULL values.
- Position (string) with a maximum length of 50 characters, allowing NULL values.
- Salary (decimal) with precision 10 and scale 2, not allowing NULL values.

The screenshot shows the Snowflake SQL Editor interface. On the left, the 'Databases' sidebar shows the 'ASSIGNMENT3' database, 'ASSIGN\_DAY\_7' schema, and the 'EMPLOYEES' table. The main editor displays the following SQL code:

```
1 -- CREATE TABLE Employees (  
2 --   EmployeeID INTEGER PRIMARY KEY,  
3 --   FirstName VARCHAR(50) NOT NULL,  
4 --   LastName VARCHAR(50),  
5 --   Department VARCHAR(50) NOT NULL,  
6 --   Position VARCHAR(50),  
7 --   Salary DECIMAL(10, 2) NOT NULL  
8 -- );  
9 select * from employees
```

The 'Results' tab shows a message: "Query produced no results". The 'Query Details' panel on the right indicates a query duration of 39ms and a query ID of 01b53e22-0000-ab24-...

### \*\*Question:\*\*

2. Alter the "Employees" table to add a constraint ensuring that the Salary column cannot be negative.

The screenshot shows the Programiz Online SQL Editor. The 'Input' tab contains the following SQL code:

```
1 FirstName VARCHAR(50) NOT NULL,  
2 LastName VARCHAR(50),  
3 Department VARCHAR(50) NOT NULL,  
4 Position VARCHAR(50),  
5 Salary DECIMAL(10, 2) NOT NULL CHECK (Salary >= 0)  
6 );  
7  
8 INSERT INTO Employees_New SELECT * FROM Employees;  
9  
10 DROP TABLE Employees;  
11  
12 ALTER TABLE Employees_New RENAME TO Employees;  
13  
14 select * from Employees *  
15 PRAGMA TABLE_INFO(Employees);
```

The 'Output' tab shows the result of the PRAGMA query:

cid	name	type	notnull	dfit_value	pk
0	EmployeeID	INTEGER	0		1
1	FirstName	VARCHAR(50)	1		0
2	LastName	VARCHAR(50)	0		0
3	Department	VARCHAR(50)	1		0
4	Position	VARCHAR(50)	0		0

The 'Available Tables' panel on the right shows the 'Customers' and 'Employees' tables. The 'Employees' table has the following data:

EmployeeID	FirstName	LastName	Department	Position
1	John	Doe	HR	Manager
2	Jane	Smith	Finance	
3	Michael		IT	Developer
4	Emily	Davis	Marketing	Marketing Specialist

**\*\*Question:\*\***

3. Define a table named "Students" with the following columns:

- StudentID (integer) as the primary key.
- Name (string) with a maximum length of 100 characters, not allowing NULL values.
- Email (string) with a maximum length of 100 characters, ensuring it is unique for each student.

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing the 'STUDENTS' database under the 'INFORMATION\_SCHEMA' schema. The main panel displays the SQL editor with the following code:

```
5 -- Department VARCHAR(50) NOT NULL,
6 -- Position VARCHAR(50),
7 -- Salary DECIMAL(10, 2) NOT NULL
8 -- );
9
10 -- CREATE TABLE Students (
11 -- StudentID INTEGER PRIMARY KEY,
12 -- Name VARCHAR(100) NOT NULL,
13 -- Email VARCHAR(100) UNIQUE NOT NULL
14 -- );
15
16 | select * from Students;
```

The 'Results' tab shows a table with columns 'STUDENTID', 'NAME', and 'EMAIL'. Below the table, it states 'Query produced no results'. The 'Query Details' panel on the right shows a query duration of 17ms and 0 rows.

**\*\*Question:\*\***

4. Create a table named "Orders" with the following columns:

- OrderID (integer) as the primary key.
- OrderDate (date) not allowing NULL values.
- TotalAmount (decimal) with precision 10 and scale 2, ensuring it is greater than zero.

The screenshot shows the Programiz Online SQL Editor. The 'Input' tab contains the following SQL code:

```
-- Use the editor to create new tables, insert data and all other SQL operations.
-- CREATE TABLE Orders1 (
-- OrderID INT PRIMARY KEY,
-- OrderDate DATE NOT NULL,
-- TotalAmount DECIMAL(10, 2) CHECK (TotalAmount > 0)
--);
--INSERT INTO Orders (OrderID, OrderDate, TotalAmount)
--VALUES
-- (1, '2024-06-01', 150.25),
-- (2, '2024-06-02', 200.50),
-- (3, '2024-06-03', 350.75);

select * from Orders1
```

The 'Output' tab shows the result of the query, which is a table with columns 'OrderID', 'OrderDate', and 'TotalAmount'.

OrderID	OrderDate	TotalAmount
1	2024-06-01	150.25
2	2024-06-02	200.5
3	2024-06-03	350.75

The 'Available Tables' panel on the right shows the 'Customers' table with columns 'customer\_id', 'first\_name', 'last\_name', 'age', and 'country'.

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

## **\*\*2. Subqueries:\*\***

### **\*\*Question:\*\***

Write a SQL query to retrieve the names of employees who have salaries greater than the average salary of their respective departments.

### **\*\*Data Source Info:\*\***

You can provide a sample "Employees" table containing columns like EmployeeID, Name, DepartmentID, Salary, etc.

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing the 'EMPLOY' table under the 'INFORMATION\_SCHEMA' database. The main panel displays a SQL query:

```
33 -- SELECT AVG(Salary)
34 -- FROM Employ
35 -- WHERE DepartmentID = e.DepartmentID
36 -- Create a table with the necessary columns
37 -- select * from EMPLOY
38 SELECT Name
39 FROM Employ e
40 WHERE Salary > (
41     SELECT AVG(Salary)
42     FROM Employ
43     WHERE DepartmentID = e.DepartmentID
44 );
45
46
```

Below the query, the 'Results' tab shows a table with 3 rows:

NAME
1 Bob
2 David
3 Eve

On the right, the 'Query Details' panel shows a query duration of 130ms and 3 rows.

## **\*\*3. Views:\*\***

### **\*\*Question:\*\***

Create a view named "HighValueCustomers" that displays the details of customers who have made purchases totaling more than \$1000.

### **\*\*Data Source Info:\*\***

Provide a sample "Orders" table containing columns like OrderID, CustomerID, TotalAmount, etc.

The screenshot shows the Snowflake web interface. On the left, the 'Databases' sidebar is open, showing the 'ORDERS' table under the 'INFORMATION\_SCHEMA' database. The main panel displays a SQL query:

```
47 -- OrderID INT PRIMARY KEY,
48 -- CustomerID INT,
49 -- TotalAmount DECIMAL(10, 2)
50 -- );
51
52 -- CREATE VIEW HighValueCustomers AS
53 -- SELECT CustomerID, SUM(TotalAmount) AS TotalPurchases
54 -- FROM Orders
55 -- GROUP BY CustomerID
56 -- HAVING SUM(TotalAmount) > 1000;
57
58 | select * from HighValueCustomers;
59
60
```

Below the query, the 'Results' tab shows a table with 2 columns: 'CUSTOMERID' and 'TOTALPURCHASES'. The message 'Query produced no results' is displayed.

On the right, the 'Query Details' panel shows a query duration of 54ms and 0 rows.

**\*\*Question:\*\***

You are tasked with creating a view for a human resources database to display employee information. Design a view named "EmployeeView" that includes the following columns: EmployeeID, FullName, Department, Position, and Salary.

**\*\*Data Source Info:\*\***

Here is a simplified structure of the "Employees" table:

```
```sql
```

```
CREATE TABLE Employees (  
  EmployeeID INT PRIMARY KEY,  
  FirstName VARCHAR(50),  
  LastName VARCHAR(50),  
  Department VARCHAR(50),  
  Position VARCHAR(50),  
  Salary DECIMAL(10, 2)  
);
```

The screenshot displays the Snowflake web interface. On the left, a sidebar shows a database schema with tables EMPLOY, EMPLOYEES, ORDERS, and STUDENTS. The main area shows a SQL query being executed. The query includes comments and an INSERT statement, followed by a SELECT statement that queries the EmployeeView. Below the query, the 'Results' tab is active, showing a table with 4 rows and 5 columns: EMPLOYEEID, FULLNAME, DEPARTMENT, POSITION, and SALARY. The results are as follows:

EMPLOYEEID	FULLNAME	DEPARTMENT	POSITION	SALARY
1	John Doe	HR	Manager	80000.00
2	Jane Smith	Finance	Accountant	60000.00
3	Michael Johnson	IT	Developer	70000.00
4	Emily Davis	Marketing	Marketing Specialist	65000.00

Query Details on the right show a duration of 129ms and a query ID.

**\*\*Question:\*\***

In a bookstore database, create a view named "BestSellers" to display information about the top-selling books. Include the following columns: BookID, Title, Author, Genre, and SalesQuantity.

**\*\*Data Source Info:\*\***

Here is a simplified structure of the "Books" table:

```
```sql
```

```
CREATE TABLE Books (  
  BookID INT PRIMARY KEY,  
  Title VARCHAR(100),  
  Author VARCHAR(100),  
  Genre VARCHAR(50),  
  SalesQuantity INT  
);
```

The screenshot displays the Snowflake web interface. At the top, there's a navigation bar with tabs for 'Databases' and 'Worksheets'. Below this, a search bar and a list of tables are visible. The 'ORDERS' table is selected. The main area shows a SQL query: `SELECT * FROM Books ORDER BY SalesQuantity DESC;`. The query results are displayed in a table with 5 rows. The table has columns: BOOKID, TITLE, AUTHOR, GENRE, and SALESQUANTITY. The results are as follows:

BOOKID	TITLE	AUTHOR	GENRE	SALESQUANTITY
2	To Kill a Mockingbird	Harper Lee	Fiction	120
4	Pride and Prejudice	Jane Austen	Romance	110
1	The Great Gatsby	F. Scott Fitzgerald	Fiction	100
5	The Catcher in the Rye	J.D. Salinger	Fiction	95
3	1984	George Orwell	Science Fiction	90

On the right side, there's a 'Query Details' panel showing 'Query duration' as 228ms and 'Rows' as 5. The bottom status bar shows the user's name 'AK' and various system icons.