

SQL LAB – 7

QUALIFIERS, HAVING CLAUSE, ALTER TABLE, TRANSACTIONS

NAME: Keerthana K R

ID: AF0363623

QUESTIONS

1. Assume you are managing a database of student records, and you need to retrieve information about students born after June 16, 2009. What will be the SQL query for this?
2. Assume you have a database containing a "Student" table with information about students, including their first names. You want to retrieve records of students whose first names start with either 'A' or 'J'. To achieve this, what will be your SQL query?
3. Let's consider a scenario where you have a database with a "Student" table that contains information about students, including their first names and email addresses. You want to retrieve records of students whose first name is not 'Alice' and whose email addresses contain the domain '@example.com'. To achieve this, what will be your SQL query?
4. Create a table Person with PersonID int, FirstName varchar (255), LastName varchar (255) and age (int). Make PersonID PRIMARY KEY.
5. Create a table Employee with emp_id int, first_name varchar (255), last_name varchar (255) and age (int). Make emp_id PRIMARY KEY.
6. Insert data to Person table
7. Insert data to Employee table
8. Create Union of two tables

ChatGPT Exercise

Using ChatGPT generate SQL queries of the below problem.

Scenario 1: In a student grades database with tables for courses and grades, find the courses where the average grade is below a 'C' (consider 'C' as a passing grade). We have a "Course" table with the following columns: CourseId, CourseName, CreditHours and "Grade" table with the following a columns: StudentId (Foreign Key), CourseID((Foreign Key), Grade. You want to find courses where the average grade is below a "C". Generate the ChatGPT prompt for creating the queries for the above requirement.

1. Assume you are managing a database of student records, and you need to retrieve information about students born after June 16, 2009. What will be the SQL query for this?

Code:

```
-- Retrieving details of students who born after june 16, 2009 using where clause
SELECT * FROM student WHERE DateOfBirth > '2009-06-16';
```

Output:

StudentID	FirstName	LastName	DateOfBirth	Gender	Email	Phone
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Assume you have a database containing a "Student" table with information about students, including their first names. You want to retrieve records of students whose first names start with either 'A' or 'J'. To achieve this, what will be your SQL query?

Code:

```
-- Retrieving student details whose firstname start with A or J using like
SELECT * FROM student WHERE FirstName LIKE 'A%' or 'J%';
```

Output:

StudentID	FirstName	LastName	DateOfBirth	Gender	Email	Phone
1	Alice	Johnson	1995-03-18	Female	alice.johnson@gmail.com	1234567890
NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Let's consider a scenario where you have a database with a "Student" table that contains information about students, including their first names and email addresses. You want to retrieve records of students whose first name is not 'Alice' and whose email addresses contain the domain '@example.com'. To achieve this, what will be your SQL query?

Code:

```
-- Retrieving details of student whose firstname is not Alice and email contain '@example.com'
SELECT * FROM student WHERE FirstName != "Alice" and Email LIKE '%@example.com';
```

Output:

StudentID	FirstName	LastName	DateOfBirth	Gender	Email	Phone
7	Roger	White	2000-08-21	Male	rogerwhite@example.com	9876543210
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Create a table Person with PersonID int, FirstName varchar (255), LastName varchar (255) and age (int). Make PersonID PRIMARY KEY.

Code:

```
CREATE TABLE Person
(
    PersonID INT PRIMARY KEY,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Age INT
);

DESC person;
```

Output:

Field	Type	Null	Key	Default	Extra
PersonID	int	NO	PRI	NULL	
FirstName	varchar(255)	YES		NULL	
LastName	varchar(255)	YES		NULL	
Age	int	YES		NULL	

5. Create a table Employee with emp_id int, first_name varchar (255), last_name varchar (255) and age (int). Make emp_id PRIMARY KEY.

Code:

```
CREATE TABLE Employee
(
    EmpID INT PRIMARY KEY,
    First_Name VARCHAR(255),
    Last_Name VARCHAR(255),
    Age INT
);

DESC employee;
```

Output;

Field	Type	Null	Key	Default	Extra
EmpID	int	NO	PRI	NULL	
First_Name	varchar(255)	YES		NULL	
Last_Name	varchar(255)	YES		NULL	
Age	int	YES		NULL	

6. Insert data to Person table

Code:

```
INSERT INTO Person
VALUES
    (1, 'John', 'Doe', 30),
    (2, 'Jane', 'Smith', 25),
    (3, 'Emily', 'Clark', 22),
    (4, 'Michael', 'Lee', 28),
    (5, 'Sophia', 'Garcia', 24),
    (6, 'Daniel', 'Wang', 31),
    (7, 'Olivia', 'Kim', 27);

SELECT * FROM Person;
```

Output:

PersonID	FirstName	LastName	Age
1	John	Doe	30
2	Jane	Smith	25
3	Emily	Clark	22
4	Michael	Lee	28
5	Sophia	Garcia	24
6	Daniel	Wang	31
7	Olivia	Kim	27
NULL	NULL	NULL	NULL

7. Insert data to Employee table

Code;

```
INSERT INTO Employee
VALUES
    (101, 'Alice', 'Johnson', 28),
    (102, 'Bob', 'Brown', 35),
    (103, 'Ethan', 'Smith', 30),
    (104, 'Isabella', 'Johnson', 26),
    (105, 'Liam', 'Brown', 33),
    (106, 'Ava', 'Martinez', 29),
    (107, 'Noah', 'Chen', 25);

SELECT * FROM employee;
```

Output:

EmpID	First_Name	Last_Name	Age
101	Alice	Johnson	28
102	Bob	Brown	35
103	Ethan	Smith	30
104	Isabella	Johnson	26
105	Liam	Brown	33
106	Ava	Martinez	29
107	Noah	Chen	25
NULL	NULL	NULL	NULL

8. Create Union of two tables.

Code:

```
SELECT * FROM person
UNION
SELECT * FROM employee;
```

Output:

PersonID	FirstName	LastName	Age
1	John	Doe	30
2	Jane	Smith	25
3	Emily	Clark	22
4	Michael	Lee	28
5	Sophia	Garcia	24
6	Daniel	Wang	31
7	Olivia	Kim	27
101	Alice	Johnson	28
102	Bob	Brown	35
103	Ethan	Smith	30
104	Isabella	Johnson	26
105	Liam	Brown	33
106	Ava	Martinez	29
107	Noah	Chen	25

ChatGPT Exercise

Using ChatGPT generate SQL queries of the below problem.

Scenario 1: In a student grades database with tables for courses and grades, find the courses where the average grade is below a 'C' (consider 'C' as a passing grade). We have a "Course" table with the following columns: CourseId, CourseName, CreditHours and "Grade" table with the following a columns: StudentId (Foreign Key), CourseID((Foreign Key), Grade. You want to find courses where the average grade is below a "C". Generate the ChatGPT prompt for creating the queries for the above requirement.

```
sql Copy code

-- Assuming a grading system where grades are stored as letters in the Grade table
-- and we need to convert them to numeric values

-- Create a CTE (Common Table Expression) to convert letter grades to numeric values
WITH GradeConversion AS (
    SELECT
        StudentId,
        CourseID,
        CASE
            WHEN Grade = 'A' THEN 4.0
            WHEN Grade = 'B' THEN 3.0
            WHEN Grade = 'C' THEN 2.0
            WHEN Grade = 'D' THEN 1.0
            WHEN Grade = 'F' THEN 0.0
            ELSE NULL -- Handle any unexpected values
        END AS NumericGrade
    FROM Grade
)

↓
```

```
-- Calculate the average grade for each course
SELECT
    c.CourseId,
    c.CourseName,
    AVG(gc.NumericGrade) AS AverageGrade
FROM
    Course c
JOIN
    GradeConversion gc ON c.CourseId = gc.CourseID
GROUP BY
    c.CourseId, c.CourseName
HAVING
    AVG(gc.NumericGrade) < 2.0;
```