SQL

Lalit Patil

SQL VIEW

- a view is a virtual table that is derived from the result of a query
- Views allow you to define complex queries and bound them into a single view.

```
CREATE VIEW view_name AS
SELECT RND, MKT FROM 50_startups
WHERE RND > 100000;
```

```
create view v1 as
select * from emp_2 inner join dept
on emp_2.did=dept.dept_id;
```

- Benefits of Creating View

➢ Simplifying complex queries and joining multiple tables.
➢ Creating virtual tables for reporting and analysis purposes.
➢ Enhancing data integrity and consistency by applying filters and validations
➢ Creating reusable query templates for commonly used data retrieval.

Scenario 1

Retrieve all employees' first names and last names:

```
CREATE VIEW EmployeeNames AS
SELECT first_name, last_name
FROM employee_data;
```

Scenario 2

List all employees earning a salary greater than $50,000:

```
CREATE VIEW HighSalaryEmployees AS
SELECT *
FROM employee_data
WHERE salary > 50000;
```

Scenario 3

Find the highest salary among employees

```
CREATE VIEW HighestSalary AS
SELECT MAX(salary) AS highest_salary
FROM employee_data;
```

Scenario 4

Count the number of employees in each location:

```
CREATE VIEW EmployeeCountsByLocation AS
SELECT location, COUNT(*) AS employee_count
FROM employee_data
GROUP BY location;
```

Scenario 5

Retrieve the names of managers

```
CREATE VIEW Managers AS
SELECT DISTINCT manager_id, first_name, last_name
FROM employee_data
WHERE manager_id IS NOT NULL;
```

Scenario 6

Calculate the average salary for each location

```
CREATE VIEW AverageSalaryByLocation AS
SELECT location, AVG(salary) AS avg_salary
FROM employee_data
GROUP BY location;
```

Scenario 7

Find employees who don't have a manager

```
CREATE VIEW EmployeesWithoutManagers AS
SELECT *
FROM employee_data
WHERE manager_id IS NULL;
```

Scenario 8

Retrieve employees whose first names start with 'A':

```
CREATE VIEW EmployeesWithAFirstName AS
SELECT *
FROM employee_data
WHERE first_name LIKE 'A%';
```

Scenario 9

Find the employee with the highest salary in each location

```
SELECT loc, fname, lname, salary FROM emp_data e
WHERE salary in
(SELECT MAX(salary) FROM emp_data WHERE loc = e.loc);
```

SHOW VIEW

SHOW CREATE VIEW v1; → *Shows the statement used to create the view*

DROP VIEW

DROP VIEW v1; → *Use to drop the view*

UNION

The UNION operator combines the result sets of multiple SELECT statements, removing duplicate rows. It returns distinct rows only, eliminating any duplicates that exist across the combined result sets.

```
select * from 50_startups where RND>50000
Union
 select * from 30_startups where mkt>50000;
```

UNION ALL

The UNION ALL operator also combines the result sets of multiple SELECT statements, but it includes all rows, including duplicates. It does not remove duplicate rows and simply appends the rows from each SELECT statement to the result set.

```
select dept_id from dept
union all
select did from emp_2;
```

SQL INDEX

- An index is a database structure that improves the speed of data retrieval operations on database tables.
- Indexes provide faster lookup and retrieval of data by allowing the database engine to navigate directly to the relevant rows based on the indexed column(s). This can significantly improve the performance of queries that involve filtering, sorting, or joining data.

```
CREATE INDEX i1
ON 50_startups(RND);
```

CREATE/SHOW/DROP

- CREATE INDEX i1 ON 50_startups(RND);
- SHOW INDEXES FROM 50_startups;
- drop index i1 on 50_startups;

Scenario 1

1. Find the average salary by department for departments with more than 2 employees.

```
SELECT dept_id, AVG(salary) AS avg_salary
FROM emp_data
GROUP BY dept_id
HAVING COUNT(empid) > 2;
```

Scenario 2

Get the count of employees by location for locations with salaries above 50,000.

```
SELECT loc, COUNT(empid) AS emp_count
FROM emp_data
WHERE salary > 50000
GROUP BY loc;
```

Scenario 3

Determine the maximum salary for male and female employees in each department.

```
SELECT dept_id, gender, MAX(salary) AS max_salary
FROM emp_data
GROUP BY dept_id, gender;
```

Scenario 4

Determine the average salary according to the locations and gender.

```
select loc, avg(salary) from emp_data
group by loc;
```

```
select gender,loc, avg(salary) from emp_data
group by loc,gender;
```

Thank You