Stored procedures

MySQL Examples:

1. Creating a Stored Procedure:

```sql
-- MySQL Stored Procedure Example: Create a simple procedure
DELIMITER //
CREATE PROCEDURE GetEmployeeCount()
BEGIN
  SELECT COUNT(*) FROM employees;
END //
DELIMITER ;
```

2. Stored Procedure with Input and Output Parameters:

```sql
Sample.sql
sample1.sql
-- MySQL Stored Procedure Example: Calculate total salary for a department
DELIMITER //
CREATE PROCEDURE CalculateTotalSalary(IN department_id INT, OUT total_salary
DECIMAL(10, 2))
BEGIN
  SELECT SUM(salary) INTO total_salary FROM employees WHERE dept_id = department_id;
END //
DELIMITER ;
```

3. Executing Stored Procedures:

```sql
-- MySQL Example: Call the GetEmployeeCount procedure
CALL GetEmployeeCount();

-- MySQL Example: Call the CalculateTotalSalary procedure
SET @total_salary_result = 0;
CALL CalculateTotalSalary(1, @total_salary_result);
SELECT @total_salary_result;
```

SQL Server Examples:

1. Creating a Stored Procedure:

```sql
-- SQL Server Stored Procedure Example: Create a simple procedure
CREATE PROCEDURE GetEmployeeCount
AS
BEGIN
  SELECT COUNT(*) FROM employees;
END;
```

2. Stored Procedure with Input and Output Parameters:

```sql
-- SQL Server Stored Procedure Example: Calculate total salary for a department
CREATE PROCEDURE CalculateTotalSalary
  @department_id INT,
  @total_salary DECIMAL(10, 2) OUTPUT
AS
BEGIN
  SELECT @total_salary = SUM(salary) FROM employees WHERE dept_id = @department_id;
END;
```

3. Executing Stored Procedures:

```sql
-- SQL Server Example: Execute the GetEmployeeCount procedure
EXEC GetEmployeeCount;

-- SQL Server Example: Execute the CalculateTotalSalary procedure
DECLARE @total_salary_result DECIMAL(10, 2);
EXEC CalculateTotalSalary @department_id = 1, @total_salary = @total_salary_result
OUTPUT;
SELECT @total_salary_result;
```

Certainly! Let's continue with examples for each aspect of stored procedures:

PostgreSQL Examples:

1. Creating a Stored Procedure:

```sql
-- PostgreSQL Stored Procedure Example: Create a simple procedure
CREATE OR REPLACE PROCEDURE GetEmployeeCount()
AS $$
BEGIN
  SELECT COUNT(*) FROM employees;
END;
$$ LANGUAGE plpgsql;
```

2. Stored Procedure with Input and Output Parameters:

```sql
-- PostgreSQL Stored Procedure Example: Calculate total salary for a department
CREATE OR REPLACE PROCEDURE CalculateTotalSalary(IN department_id INT, OUT
total_salary DECIMAL)
AS $$
BEGIN
  SELECT SUM(salary) INTO total_salary FROM employees WHERE dept_id = department_id;
END;
$$ LANGUAGE plpgsql;
```

3. Executing Stored Procedures:

```sql
-- PostgreSQL Example: Call the GetEmployeeCount procedure
CALL GetEmployeeCount();

-- PostgreSQL Example: Call the CalculateTotalSalary procedure
DO $$
DECLARE
  total_salary_result DECIMAL;
BEGIN
  PERFORM CalculateTotalSalary(1, total_salary_result);
  RAISE NOTICE 'Total Salary: %', total_salary_result;
END $$;
```

Oracle Database Examples:

1. Creating a Stored Procedure:

```sql
```

```sql
-- Oracle Stored Procedure Example: Create a simple procedure
CREATE OR REPLACE PROCEDURE GetEmployeeCount
AS
  total_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO total_count FROM employees;
  DBMS_OUTPUT.PUT_LINE('Employee Count: ' || total_count);
END;
/
```

2. Stored Procedure with Input and Output Parameters:

```sql
-- Oracle Stored Procedure Example: Calculate total salary for a department
CREATE OR REPLACE PROCEDURE CalculateTotalSalary(department_id IN NUMBER,
total_salary OUT NUMBER)
AS
BEGIN
  SELECT SUM(salary) INTO total_salary FROM employees WHERE dept_id = department_id;
END;
/
```

3. Executing Stored Procedures:

```sql
-- Oracle Example: Execute the GetEmployeeCount procedure
EXEC GetEmployeeCount;

-- Oracle Example: Execute the CalculateTotalSalary procedure
DECLARE
  total_salary_result NUMBER;
BEGIN
  CalculateTotalSalary(1, total_salary_result);
  DBMS_OUTPUT.PUT_LINE('Total Salary: ' || total_salary_result);
END;
/
```

These examples cover creating stored procedures, defining input and output parameters, and executing the procedures in PostgreSQL and Oracle Database. Adjust the syntax as needed for your specific database system.