Q1. what is stored procedure?

Ans. The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks.

- It is just like procedures in other programming languages.
- A procedure contains a header and a body.

→ Heading:

- The header contains the name of the procedure and the parameters of variables passed to the procedure.

→ Body:

- The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.

→ Procedures: Passing Parameters

1. IN parameters:
   - This parameter is used for giving input to the subprograms.
2. OUT parameters:
   - This parameter is used for getting output from the subprograms.
3. IN OUT parameters:
   - This parameter is used for giving input and for getting output from the subprograms.

Q2. Describe the use of, %ROWTYPE and %TYPE in SQL.

Ans. The %ROWTYPE attribute, used to declare PL/SQL variables of type record with fields that correspond to the columns of a table or view, is supported by the data server.

- The %TYPE attribute, used in PL/SQL variable and parameter declarations, is supported by the data server. The data type of this column or variable is assigned to the variable being declared.

eg:

```
DECLARE
    p_employee_record employees%ROWTYPE;
    p_employee_salary employees.emp_salary%TYPE;
BEGIN
    SELECT * INTO p_employee_record FROM employees
    WHERE emp_id = 18;
    p_employee_salary := p_employee_record.emp_salary;

    dbms_output.put_line ('Employee name: ' ||
                          p_employee_record.emp_name
                          || 'Employee salary: ' ||
                          p_employee_salary);
END;
/
```

Q3. Explain IN, OUT, IN-OUT mode in stored procedure.

Ans. 1. IN parameters:

- The IN parameter can be referenced by the procedure or function.
- This parameter is used for giving input to the subprograms.
- It is a read-only variable inside the subprograms, their values cannot be changed inside the subprogram.

2. OUT ~~proced~~ parameters:

- The OUT parameter cannot be referenced by the procedure or function.

This parameter is used for getting output from the subprograms.
It is a read-write variable inside the subprograms, their values can be changed inside the subprograms.

3. INOUT parameters:
The INOUT parameter can be referenced by the procedure or function.
This parameter is used for both giving input and for getting out put from the subprograms
It is a read-write variable inside the subprograms, their values can be changed inside the subprograms.

Q4 What is a stored Function?
Ans. The PL/SQL function is very similar to PL/SQL procedure.
The main difference between procedure and a function is, a function must always return a value, and on the other hand a procedure may or may not return a value.
Except this, all the other things of PL/SQL procedure are true for PL/SQL function too.
→ Syntax to create a function:

```
CREATE [OR REPLACE] FUNCTION function_name [parameters]
[(parameter_name [IN] [OUT] [IN OUT] type [,...])]
RETURN return_datatype
{IS | AS}
BEGIN
    <function_body>
END [function_name];
/
```

→ Syntax for removing your created function:
. DROP FUNCTION function_name;

Q5. What is the difference between stored functions and stored procedures?

Ans

	Stored Functions	Stored Procedure
Return value	Always returns a value	may or may not return a value
Parameters	can only have input parameters	can have either input or output parameters
Calling	can be called from procedures	can be called by name from an application
SELECT statements	Can be embedded in select statements	cannot be used in SELECT statements
Exceptions	Try-catch block cannot be used within functions	Exceptions can be handled in try-catch blocks within procedures
Transactions	Transactions are not permitted within a function	A stored procedure can contain transaction