Package:

```
create or replace PACKAGE  YOUR_PACKAGE_NAME
AS
PROCEDURE  GET_DATA ( MID  VARCHAR2,FROMDATE DATE, MY_CURSOR  OUT  SYS_REFCURSOR );

END  YOUR_PACKAGE_NAME;

create or replace PACKAGE BODY  YOUR_PACKAGE_NAME
AS
PROCEDURE  GET_DATA ( MID  VARCHAR2,FROMDATE DATE,  MY_CURSOR OUT SYS_REFCURSOR )
IS
BEGIN

IF MID IS NULL THEN
OPEN  MY_CURSOR FOR

SELECT * FROM  TABLE_NAME1;

ELSE

OPEN  MY_CURSOR FOR
SELECT * FROM  TABLE_NAME2;

END IF;
EXCEPTION
WHEN OTHERS THEN
RAISE;

END GET_DATA;

END YOUR_PACKAGE_NAME;
```

Procedure:

1. cannot return any values through the RETURN statement.
2. CREATE PROCEDURE instructs the compiler to create new procedure. Keyword 'OR REPLACE' instructs the compile to replace the existing procedure (if any) with the current one.
3. Procedure name should be unique.
4. Keyword 'IS' will be used, when the procedure is nested into some other blocks. If the procedure is standalone then 'AS' will be used. Other than this coding standard, both have the same meaning.

EXAMPLE 1:

```
create or replace PACKAGE  YOUR_PACKAGE_NAME
AS
PROCEDURE  GET_DATA (MY_CURSOR  OUT  SYS_REFCURSOR );
PROCEDURE  GET_SINGLE_RESPONSE(P_RESPONSE OUT  VARCHAR2);

END  YOUR_PACKAGE_NAME;

--BODY
create or replace PACKAGE BODY  YOUR_PACKAGE_NAME
AS
PROCEDURE  GET_DATA(MY_CURSOR OUT SYS_REFCURSOR )
IS
BEGIN
OPEN  MY_CURSOR FOR
SELECT * FROM EPORTAL_TERMINAL;
EXCEPTION
WHEN OTHERS THEN
RAISE;
END GET_DATA;

PROCEDURE GET_SINGLE_RESPONSE(P_RESPONSE OUT  VARCHAR2)
IS
 ERR_NUM  NUMBER;
 ERR_MSG  VARCHAR2(100);
BEGIN
--ANY LOGIC CAN BE WRIITEN HERE
--SELECT * FROM EPORTAL_TERMINAL;
P_RESPONSE :='SUCCESS BHACCHI';
EXCEPTION
WHEN OTHERS THEN
 P_RESPONSE :='ERROR';
  RAISE;
END GET_SINGLE_RESPONSE;

END YOUR_PACKAGE_NAME;
```

What is Function?

Functions is a standalone PL/SQL subprogram. Like PL/SQL procedure, functions have a unique name by which it can be referred. These are stored as PL/SQL database objects. Below are some of the characteristics of functions.

- Functions are a standalone block that is mainly used for calculation purpose.
- Function use RETURN keyword to return the value, and the datatype of this is defined at the time of creation.
- A Function should either return a value or raise the exception, i.e. return is mandatory in functions.
- Function with no DML statements can be directly called in SELECT query whereas the function with DML operation can only be called from other PL/SQL blocks.
- It can have nested blocks, or it can be defined and nested inside the other blocks or packages.
- It contains declaration part (optional), execution part, exception handling part (optional).
- The values can be passed into the function or fetched from the procedure through the parameters.
- These parameters should be included in the calling statement.
- Function can also return the value through OUT parameters other than using RETURN.
- Since it will always return the value, in calling statement it always accompanies with assignment operator to populate the variables.

Procedure Vs. Function: Key Differences

Procedure	Function
• Used mainly to a execute certain process	• Used mainly to perform some calculation
• Cannot call in SELECT statement	• A Function that contains no DML statements can be called in SELECT statement
• Use OUT parameter to return the value	• Use RETURN to return the value
• It is not mandatory to return the value	• It is mandatory to return the value
• RETURN will simply exit the control from subprogram.	• RETURN will exit the control from subprogram and also returns the value
• Return datatype will not be specified at the time of creation	• Return datatype is mandatory at the time of creation

Built-in Functions in PL/SQL

PL/SQL contains various built-in functions to work with strings and date datatype. Here we are going to see the commonly used functions and their usage.

Conversion Functions

These built-in functions are used to convert one datatype to another datatype.

Function Name	Usage	EXAMPLE
TO_CHAR	Converts the other datatype to character datatype	TO_CHAR(123);
TO_DATE (string, format)	Converts the given string to date. The string should match with the format.	TO_DATE('2015-JAN-15', 'YYYY-MON-DD'); Output: 1/15/2015
TO_NUMBER (text, format)	Converts the text to number type of the given format. Informat '9' denotes the number of digits	Select TO_NUMBER('1234','9999') fro m dual; Output: 1234 Select TO_NUMBER('1,234.45','9,999. 99') from dual; Output: 1234

String Functions

These are the functions that are used on the character datatype.

Function Name	Usage	EXAMPLE
INSTR(text, string, start , occurance)	Gives the position of particular text in the given string. • text – Main string • string – text that need to be searched • start – starting position of the search (optional) • accordance – occurrence of the searched string (optional)	Select INSTR('AEROPLANE','E',2,1) from dual **Output**: 2 Select INSTR('AEROPLANE','E',2,2) from dual Output: 9 (2nd occurance of E)
SUBSTR (text, start, le ngth)	Gives the substring value of the main string. • text – main string • start – starting position • length – length to be sub stringed	select substr('aeroplane',1,7) from dual **Output**: aeropla
UPPER (text)	Returns the uppercase of the provided text	Select upper('guru99') from dual; **Output**: GURU99
LOWER (text)	Returns the lowercase of the provided text	Select lower ('AerOpLane') from dual; **Output**: aeroplane

Function Name	Usage	EXAMPLE
INITCAP (text)	Returns the given text with the starting letter in upper case.	Select ('guru99') from dual **Output**: Guru99 Select ('my story') from dual **Output**: My Story
LENGTH (text)	Returns the length of the given string	Select LENGTH ('guru99') from dual; **Output**: 6
LPAD (text, length, pad_char)	Pads the string in the left side for the given length (total string) with the given character	Select LPAD('guru99', 10, '$') from dual; **Output**: $$$$guru99
RPAD (text, length, pad_char)	Pads the string in the right side for the given length (total string) with the given character	Select RPAD('guru99',10,'-') from dual **Output**: guru99----
LTRIM (text)	Trims the leading white space from the text	Select LTRIM(' Guru99') from dual; **Output**: Guru99
RTRIM (text)	Trims the trailing white space from the text	Select RTRIM('Guru99 ') from dual; **Output**; Guru99

Date Functions

These are functions that are used for manipulating with dates.

Function Name	Usage	EXAMPLE
ADD_MONTHS (date, no.of months)	Adds the given months to the date	ADD_MONTH('2015-01-01',5); **Output**: 05/01/2015
SYSDATE	Returns the current date and time of the server	Select SYSDATE from dual; **Output**: 10/4/2015 2:11:43 PM
TRUNC	Round of the date variable to the lower possible value	select sysdate, TRUNC(sysdate) from dual; **Output**: 10/4/2015 2:12:39 PM 10/4/2015
ROUND	Rounds the date to the nearest limit either higher or lower	Select sysdate, ROUND(sysdate) from dual **Output**: 10/4/2015 2:14:34 PM 10/5/2015
MONTHS_BETWEEN	Returns the number of months between two dates	Select MONTHS_BETWEEN (sysdate+60, sysdate) from dual **Output**: 2

Q #3) How will you differentiate between VARCHAR & VARCHAR2?

VARCHAR can store characters up to 2000 bytes while VARCHAR2 can store up to 4000 bytes.

VARCHAR will hold the space for characters defined during declaration even if all of them are not used whereas VARCHAR2 will release the unused space.

Q #4) What is the difference between TRUNCATE & DELETE command?

Ans: Both the commands are used to remove data from a database.
The finer differences between the two include:

- TRUNCATE is a DDL operation while DELETE is a DML operation.
- The TRUNCATE command will free the object storage space while the DELETE command does not.

Q #7) What is the difference between SUBSTR & INSTR functions?

Ans: SUBSTR function returns the sub-part identified by numeric values from the provided string.
Example: [Select SUBSTR ('India is my country', 1, 4) from dual] will return "Indi".
INSTR will return the position number of the sub-string within the string.

Example: [SELECT INSTR ('India is my country', 'a') from dual] will return 5.

Q #8) How can we find out the duplicate values in an Oracle table?

```
SELECT EMP_NAME, COUNT (EMP_NAME)
FROM EMP
GROUP BY EMP_NAME
HAVING COUNT (EMP_NAME) > 1;
```

Q #10) What is a NVL function? How can it be used?

Ans: NVL is a function, which helps the user to substitute a value if null is encountered for an expression.
It can be used as the below syntax.
[NVL (Value_In, Replace_With)]

Q #11) What is the difference between a Primary Key & a Unique Key?

Ans: Primary key is used to identify each table row uniquely, while a Unique Key prevents duplicate values in a table column.
Given below are few differences:

- The primary key cannot hold null value at all while Unique key allows multiple null values.
- The primary key is a clustered index while a unique key is a non-clustered index.

Q #32) What is meant by an index?

Ans: An index is a schema object, which is created to search the data efficiently within the table. Indexes are usually created on certain columns of the table, which are accessed the most.
Indexes can be clustered or non-clustered.

Q #13) How can we find out the current date and time in Oracle?

```
SELECT SYSDATE into CURRENT_DATE from dual;
```

Q #17) What is the quickest way to fetch the data from a table?

Ans: The quickest way to fetch the data would be to use primary key column with index in the SQL Query.

Q #21) What is the use of Aggregate functions in Oracle?

Ans: Aggregate functions perform summary operations on a set of values to provide a single value. There are several aggregate functions that we use in our code to perform calculations.

Few of them are listed below:

- AVG
- MIN
- MAX
- COUNT
- SUM
- STDEV

Q #15) How will you write a query to get a 5th RANK student from a table STUDENT_REPORT?

Ans: The Query will be as follows:

```
SELECT TOP 1 RANK
FROM (SELECT TOP 5 RANK
FROM STUDENT_REPORT
ORDER BY RANK DESC) AS STUDENT
ORDER BY RANK ASC;
```

Q #22) What are the set operators UNION, UNION ALL, MINUS & INTERSECT meant to do?

Ans: Set operator facilitates the user to fetch the data from two or more than two tables at once if the columns and relative data types are same in the source tables.

- UNION operator returns all the rows from both the tables except the duplicate rows.
- UNION ALL returns all the rows from both the tables along with the duplicate rows.
- MINUS returns rows from the first table, which does not exist in the second table.
- INTERSECT returns only the common rows in both the tables.

Q #23) Can we convert a date to char in Oracle and if so, what would be the syntax?

Ans: We can use the TO_CHAR function to do the above conversion.

The syntax will be as follows:

[SELECT to_char (to_date ('30-01-2018′, 'DD-MM-YYYY'), 'YYYY-MM-DD') FROM dual;]

Q #24) What do you mean by a database transaction & what all TCL statements are available in Oracle?

Ans: Transaction occurs when a set of SQL statements are executed in one go. To control the execution of these statements, Oracle has introduced TCL i.e. Transaction Control Statements that use a set of statements.

The set of statements include:

- **COMMIT:** Used to make a transaction permanent.
- **ROLLBACK:** Used to roll back the state of DB to last the commit point.
- **SAVEPOINT:** Helps to specify a transaction point to which rollback can be done later.

Q #25) What do you understand by a database object? Can you list a few of them?

Ans: An object used to store the data or references of the data in a database is known as a Database object.

The database consists of various types of DB objects such as tables, views, indexes, constraints, stored procedures, triggers etc.

Q #27) Can we save images in a database and if yes, how?

Ans: BLOB stands for Binary Large Object, which is a datatype that is generally used to hold images, audio & video files or some binary executables.
This datatype has the capacity of holding data up to 4 GB.

Q #28) What do you understand by database schema and what does it hold?

Ans: Schema is a collection of database objects owned by a database user who can create or manipulate new objects within this schema.
The schema can contain any DB objects like table, view, indexes, clusters, stored procs, functions etc.

Q #30) What is a View and how is it different from a table?

Ans: A view is a user-defined database object that is used to store the results of a SQL query, which can be referenced later. Views do not store this data physically but as a virtual table, hence it can be referred as a logical table.
A table can hold data but not SQL Query results whereas View can save the query results, which can be used in another SQL Query as a whole.

The table can be updated or deleted while Views cannot be done so.

Q #31) What is meant by a deadlock situation?

Ans: Deadlock is a situation when two or more users are simultaneously waiting for the data, which is locked by each other and hence, results in all blocked user sessions.

Q #37) What are the parameters that we can pass through a stored procedure?

Ans: We can pass IN, OUT & INOUT parameters through a stored procedure and they should be defined while declaring the procedure itself.

Q #38) What is a trigger and what are its types?

Ans: A trigger is a stored program which is written in such a way that it gets executed automatically when some event occurs. This event can be any DML or a DDL operation.
PL/SQL supports two types of triggers:

- Row Level
- Statement Level

Q #39) How will you distinguish a global variable with a local variable in PL/SQL?

Ans: Global variable is the one, which is defined at the beginning of the program and survives until the end.
It can be accessed by any methods or procedures within the program, while the access to the local variable is limited to the procedure or method where it is declared.

Q #40) What are the packages in PL SQL?

Ans: A Package is a group of related database objects like stored procs, functions, types, triggers, cursors etc. that are stored in Oracle database. It is a kind of library of related objects which can be accessed by multiple applications if permitted.

PL/SQL Package structure consists of 2 parts: package specification & package body.