PACKAGES:

> IT IS A COLLECTION OF VARIABLES, CURSORS, PROCEDURES & FUNCTIONS ARE STORED IN ONE LOCATION.

> EASY TO SHARE THE SUBPROGRAMS IN APPLICATION S/W TOOLS

> THEY IMPROVE PERFORMANCE WHILE ACCESSING SUBPROGRAMS FROM CLIENT LOCATION

> THEY ARE STORED IN "USER_SOURCE" SYSTEM TABLE

> THEY SUPPORTS FUNCTION OVERLOADING, ENCAPSULATION & DATABINDING

> IT HAS TWO PARTS

1. **PACKAGE SPECIFICATION:**

IT HOLDS THE DECLARATION OF VARIABLES, CURSORS & SUBPROGRAMS.

SYNTAX:

```
CREATE [OR REPLACE] PACKAGE <PACKAGE NAME> IS /
AS
<DECLARE VARIABLES, CURSORS, SUB BLOCKS>;
END;
/
```

2. **PACKAGE BODY:**

IT HOLDS THE BODY OF SUBPROGRAMS.

SYNTAX:

```
CREATE [OR REPLACE] PACKAGE BODY <PACKAGE NAME> IS /
AS
<IMPLEMENTING SUB BLOCKS>;
END;
/
```

EX1:

CREATE A PACKAGE TO ENCAPSULATE / BIND MULTIPLE

PORCEDURES?

```
SQL> CREATE OR REPLACE PACKAGE PK1
 IS
 PROCEDURE SP1;
 PROCEDURE SP2;
 END;
 /
PACKAGE CREATED.

SQL> CREATE OR REPLACE PACKAGE BODY PK1  IS
 PROCEDURE SP1
 AS
 BEGIN
 DBMS_OUTPUT.PUT_LINE ('MY NAME IS PROCEDURE1');  END SP1;
 PROCEDURE SP2
 AS
 BEGIN
 DBMS_OUTPUT.PUT_LINE ('MY NAME IS PROCEDURE2');  END
SP2;
 END;
 /
PACKAGE BODY CREATED.

SQL> EXECUTE PK1.SP1;
MY NAME IS PROCEDURE
PL/SQL PROCEDURE SUCCESSFULLY COMPLETED.
SQL> EXECUTE PK1.SP2;
MY NAME IS PROCEDURE2
PL/SQL PROCEDURE SUCCESSFULLY COMPLETED.
```

EX2:

CREATE A PACKAGE TO BIND VARIABLE, PROCEDURE & FUNCTION?

```
CREATE PACKAGE PK2
IS
A NUMBER (10): =2000;
PROCEDURE SP1;
FUNCTION SF1(X NUMBER) RETURN NUMBER; END;
/
PACKAGE CREATED.
CREATE PACKAGE BODY PK2
IS
PROCEDURE SP1
AS
X NUMBER (10);
BEGIN
X: =A/2;
DBMS_OUTPUT.PUT_LINE(X);
END SP1;
FUNCTION SF1(X NUMBER)
RETURN NUMBER
AS
BEGIN
RETURN X*A;
END SF1;
END;
/
PACKAGE BODY CREATED.
```

EX3:

PACKAGE SPECIFICATION:

```
CREATE OR REPLACE PACKAGE MY_PACK
```

```
 IS.
 RESULT VARCHAR2(50);
 PROCEDURE EMP_EXP (TEMPNO EMP.EMPNO%TYPE);
 FUNCTION EMP_NETSAL (TEMPNO EMP.EMPNO%TYPE) RETURN
VARCHAR2;
  END MY_PACK;
```

PACKAGE BODY:

```
CREATE OR REPLACE PACKAGE BODY MY_PACK
 IS
  PROCEDURE EMP_EXP (TEMPNO EMP.EMPNO%TYPE)  IS
  TDATE EMP.HIREDATE%TYPE; -- PRIVATE VARIABLES TEXP
  NUMBER;
  BEGIN
 SELECT HIREDATE INTO TDATE FROM EMP WHERE
EMPNO=TEMPNO;
 TEXP: =ROUND((SYSDATE-TDATE)/365);
 DBMS_OUTPUT.PUT_LINE (TEMPNO||' EMPLOYEE EXPERIENCE  IS
'||TEXP||'YEARS.');
  END EMP_EXP;
  FUNCTION EMP_NETSAL (TEMPNO EMP.EMPNO%TYPE)
RETURN VARCHAR2
 IS
  TSAL EMP.SAL%TYPE;
  TCOMM EMP.COMM%TYPE;
  BEGIN
 SELECT SAL+NVL(COMM,0) INTO RESULT FROM EMP
                                    WHERE EMPNO=TEMPNO;
 RETURN (TEMPNO||'EMPLOYEE NET SALARY RS.'||RESULT); END
  EMP_NETSAL;
END MY_PACK;
```

TO EXECUTE ABOVE PACKAGE:

```
EXEC MY_PACK.EMP_EXP (7788);
SELECT MY_PACK.EMP_NETSAL (7788) FROM DUAL;
```

FUNCTION OVERLOADING USING PACKAGE:

EX:

```
SQL> CREATE OR REPLACE PACKAGE PK3
 IS
   FUNCTION   SF1(X   NUMBER,   Y   NUMBER)   RETURN   NUMBER
FUNCTION SF1(A NUMBER, B NUMBER, C NUMBER) RETU  END;
 /
PACKAGE CREATED.

SQL> CREATE OR REPLACE PACKAGE BODY PK3  IS
 FUNCTION SF1(X NUMBER, Y NUMBER)
 RETURN NUMBER
 AS
 BEGIN
 RETURN (X*Y);
 END SF1;
 FUNCTION SF1(A NUMBER, B NUMBER, C NUMBER)
RETURN NUMBER
 AS
 BEGIN
 RETURN A+B+C;
 END SF1;
 END;
 /

PACKAGE BODY CREATED.
```

```
SQL> SELECT PK3.SF1 (10,20,30) FROM DUAL;


PK3.SF1 (10,20,30)
-----------------
 60


SQL> SELECT PK3.SF1 (10,20) FROM DUAL;


PK3.SF1 (10,20)
--------------
 200
```

EX:

```
CREATE OR REPLACE PACKAGE FO_PACK
 IS
FUNCTION ADDVAL (A NUMBER, B NUMBER) RETURN NUMBER;
FUNCTION ADDVAL (A NUMBER, B NUMBER, C NUMBER) RETURN NUMBER;
FUNCTION ADDVAL (STR1 VARCHAR2, STR2 VARCHAR2) RETURN VARCHAR2;
FUNCTION ADDVAL (STR1 VARCHAR2, STR2 VARCHAR2, STR3 VARCHAR2) RETURN VARCHAR2;
END FO_PACK;
```

PACKAGE BODY:

```
CREATE OR REPLACE PACKAGE BODY FO_PACK
 IS
FUNCTION ADDVAL (A NUMBER, B NUMBER) RETURN NUMBER  IS
BEGIN
 RETURN(A+B);
END ADDVAL;
```

```
FUNCTION ADDVAL (A NUMBER, B NUMBER, C NUMBER) RETURN
NUMBER
 IS
BEGIN
  RETURN(A+B+C);
END ADDVAL;

FUNCTION ADDVAL (STR1 VARCHAR2, STR2 VARCHAR2) RETURN
VARCHAR2
 IS
BEGIN
  RETURN(STR1||STR2);
END ADDVAL;

FUNCTION ADDVAL (STR1 VARCHAR2, STR2 VARCHAR2, STR3
VARCHAR2) RETURN VARCHAR2
 IS
BEGIN
  RETURN(STR1||STR2||STR3);
END ADDVAL;
END FO_PACK;
/
```

CALLING PACKAGE:

```
SELECT FO_PACK.ADDVAL(10,20) FROM DUAL;
SELECT FO_PACK.ADDVAL('RAMA ','KRISHNA ','RAJU') FROM  DUAL;
SELECT FO_PACK.ADDVAL(10,20,50) FROM DUAL;
```

NOTE:

> ALL PACKAGES BODIES ARE STORED IN 'USER_SOURCE'. > TO SEE THE PACKAGE BODY.

EX:

```
SQL> SELECT TEXT FROM USER_SOURCE WHERE
NAME='FO_PACK';
```

DROPPING PACKAGE BODY:

SYNTAX:

```
SQL> DROP PACKAGE BODY <PACKAGE NAME>;
```

EX:

```
SQL> DROP PACKAGE BODY MY_PACK;
```

DROPPING PACKAGES:

SYNTAX:

```
SQL> DROP PACKAGE <PACKAGE NAME>;
```

EX:

```
SQL> DROP PACKAGE MY_PACK;
```