1- Create table ->

LIVE SQL

2-CREATE TABLE FOR EACH CATEGORY

CREATE TABLE XXBCM\_SUPP

(

SUPP\_ID NUMBER,

SUPPLIER\_NAME VARCHAR2(50),

SUPP\_CONTACT\_NAME VARCHAR2(50),

SUPP\_ADDRESS VARCHAR2(200),

SUPP\_CONTACT\_NUMBER VARCHAR2(15),

SUPP\_EMAIL VARCHAR2(50),

CONSTRAINT SUPP\_ID\_PK PRIMARY KEY (SUPP\_ID)

);

CREATE TABLE XXBCM\_INVOICE

(

INVOICE\_ID NUMBER,

INVOICE\_REFERENCE NUMBER(15),

INVOICE\_DATE DATE,

INVOICE\_STATUS VARCHAR2(200),

INVOICE\_HOLD\_REASON VARCHAR2(200),

INVOICE\_AMOUNT NUMBER(10),

INVOICE\_DESCRIPTION VARCHAR2(200),

CONSTRAINT INVOICE\_ID\_PK PRIMARY KEY (INVOICE\_ID)

);

CREATE TABLE XXBCM\_ORDER

(

ORDER\_ID NUMBER,

ORDER\_REF VARCHAR2(15),

ORDER\_DATE DATE,

ORDER\_TOTAL\_AMOUNT NUMBER(15),

ORDER\_DESCRIPTION VARCHAR2(200),

ORDER\_STATUS VARCHAR2(200),

ORDER\_LINE\_AMOUNT NUMBER(15),

SUPP\_ID NUMBER NOT NULL,

INVOICE\_ID NUMBER NOT NULL,

CONSTRAINT ORDER\_ID\_PK PRIMARY KEY (ORDER\_ID),

CONSTRAINT FK\_SUPPLIER

FOREIGN KEY (SUPP\_ID)

REFERENCES XXBCM\_SUPP(SUPP\_ID),

CONSTRAINT FK\_INVOICE

FOREIGN KEY (INVOICE\_ID)

REFERENCES XXBCM\_INVOICE(INVOICE\_ID)

);

3- stored procedure->

CREATE OR REPLACE PROCEDURE XXBCM\_PROC

IS BEGIN

FOR

mgt\_data

IN

(SELECT \* FROM XXBCM\_ORDER\_MGT)

LOOP

BEGIN

INSERT INTO XXBCM\_SUPP

( SUPPLIER\_NAME, SUPP\_CONTACT\_NAME, SUPP\_ADDRESS, SUPP\_CONTACT\_NUMBER, SUPP\_EMAIL )

VALUES

( mgt\_data.SUPPLIER\_NAME, mgt\_data.SUPP\_CONTACT\_NAME, mgt\_data.SUPP\_ADDRESS, mgt\_data.SUPP\_CONTACT\_NUMBER, mgt\_data.SUPP\_EMAIL );

INSERT INTO XXBCM\_INVOICE

( INVOICE\_REFERENCE, INVOICE\_DATE, INVOICE\_STATUS, INVOICE\_HOLD\_REASON, INVOICE\_AMOUNT, INVOICE\_DESCRIPTION )

VALUES

( mgt\_data.INVOICE\_REFERENCE, mgt\_data.INVOICE\_DATE, mgt\_data.INVOICE\_STATUS, mgt\_data.INVOICE\_HOLD\_REASON, mgt\_data.INVOICE\_AMOUNT, mgt\_data.INVOICE\_DESCRIPTION );

INSERT INTO

XXBCM\_ORDER ( ORDER\_REF, ORDER\_DATE, ORDER\_TOTAL\_AMOUNT, ORDER\_DESCRIPTION, ORDER\_STATUS, ORDER\_LINE\_AMOUNT )

VALUES

( mgt\_data.ORDER\_REF, mgt\_data.ORDER\_DATE, mgt\_data.ORDER\_TOTAL\_AMOUNT, mgt\_data.ORDER\_DESCRIPTION, mgt\_data.ORDER\_STATUS, mgt\_data.ORDER\_LINE\_AMOUNT );

EXCEPTION WHEN OTHERS THEN

dbms\_output.put\_line('Exception occured on '||mgt\_data.ORDER\_REF);

END;

END LOOP;

END;

4- Extract report ->

**prefix P0**

SELECT SUBSTR(ORDER\_REF, 5 ) FROM XXBCM\_ORDER\_MGT;**Date**

SELECT TO\_DATE (ORDER\_DATE, 'MON-DD') FROM XXBCM\_ORDER\_MGT;

**Uppercase**

DECLARE

Test\_String string(40) := 'Hello, welcome to geeksforgeeks.';

BEGIN

dbms\_output.put\_line(INITCAP(Test\_String));

END;

**Number format for order amount**

SELECT TO\_CHAR(ORDER\_TOTAL\_AMOUNT, '99,999,990.00') FROM XXBCM\_ORDER\_MGT;

**Number format for invoice amount**

SELECT TO\_CHAR(INVOICE\_AMOUNT, '99,999,990.00') FROM XXBCM\_ORDER\_MGT;

**If invoice status**

**DECLARE**

**ORDER\_STATUS.XXBCM\_ORDER\_MGT = 'Paid';**

**BEGIN**

**case**

**when ORDER\_STATUS = 'Paid' then dbms\_output.put\_line('OK');**

**when INVOICE\_STATUS = 'Pending' then dbms\_output.put\_line('To follow up');**

**else dbms\_output.put\_line('To verify);**

**end case;**

SELECT INVOICE\_STATUS

FROM XXBCM\_ORDER\_MGT

WHERE INVOICE\_STATUS operator ALL  
(SELECT INVOICE\_STATUS FROM XXBCM\_ORDER\_MGT WHERE condition);

IF INVOICE\_STATUS = 'PAID' THEN

DBMS\_OUTPUT.put\_line ('Ok');

ELSE

SELECT INVOICE\_STATUS

FROM XXBCM\_ORDER\_MGT

WHERE INVOICE\_STATUS operator ANY  
(SELECT INVOICE\_STATUS FROM XXBCM\_ORDER\_MGT WHERE condition);

IF INVOICE\_STATUS = 'PENDING' THEN

DBMS\_OUTPUT.put\_line ('To follow up');

ELSE

DBMS\_OUTPUT.put\_line ('To verify');

5-Return detail highest order total amount

SELECT TOP 3 NOM DE COLONE

FROM TABLE

ORDER BY DESC