

GROFERS

Team Members:

J. Chaitanya Venkata Sai Narayana

B. Vishnu Vardhan Reddy

Jagarlamudi Chandini

V.Dhana Lakshmi

Pasupuleti Chandana

Bandaru DhanaLakshmi

Tommandru Lavanya





ER- Diagram

constraints

Procedures

Functions

Packages

ER-DIAGRAM

EMP TABLE

Emp_code
Emp_name
Dept_code
Grade
Age
Date_join
Sex
Salary

DEPTTABLE

Dept_id
Dept_code
Dept_name

SALESMAN

Emp_code
Emp_name
Age
Date_join
Salary
Married
Reports_to

CITY

City_code
City_name
State_code

STATE

State_code
State_name

IMPORTPROD

Prod_code
Prod_name
Prodgr_code
Sale_price
Target
Direct_sales
Indirect_sales
Profit_margin

OLD PROD

Prod_code
Prod_name

PROD GROUP

Prodgr_code
Prodgr_name

PROD TABLE

Prod_code
Prod_name
Prodgr_code
Sale_price
Target
Direct_sales
Indirect_sales
Profit_margin
Brand_name

ORDDETAILS

Order_number
Prod_code
Quantity

ORDTABLE

Order_number
Order_date
Cust_code

CUSTOMER

Cust_code
Cust_name
city_code
Credit_rating



TABLE STRUCTURE

Desc depttable

Name	Null?	Type
DEPT_ID		NUMBER(10)
DEPT_CODE	NOT NULL	VARCHAR2(50)
DEPT_NAME	NOT NULL	VARCHAR2(50)

desc emptable

Name	Null?	Type
EMP_CODE	NOT NULL	NUMBER(10)
EMP_NAME	NOT NULL	VARCHAR2(50)
DEPT_CODE	NOT NULL	VARCHAR2(50)
GRADE	NOT NULL	VARCHAR2(50)
AGE	NOT NULL	NUMBER(10)
DATE_JOIN	NOT NULL	DATE
SEX		VARCHAR2(10)
SALARY	NOT NULL	NUMBER(6)



TABLE STRUCTURE

desc prodtable

Name	Null?	Type
PROD_CODE	NOTNULL	NUMBER(10)
PROD_NAME	NOTNULL	VARCHAR2(50)
PRODGR_CODE	NOT NULL	NUMBER(10)
SALE_PRICE	NOT NULL	NUMBER(10)
TARGET	NOT NULL	NUMBER(10)
DIRECT_SALES		NUMBER(10)
INDIRECT_SALES	NOT NULL	NUMBER(20)
PROFIT_MARGIN	NOT NULL	NUMBER(10)
BRAND_NAME	NOT NULL	VARCHAR2(50)

desc prodgroup

Name	Null?	Type
PRODGR_CODE	NOT NULL	NUMBER(10)
PRODGR_NAME	NOT NULL	VARCHAR2(50)



TABLE STRUCTURE

desc oldprod

Name	Null?	Type
PROD_CODE	NOT NULL	NUMBER(10)
PROD_NAME	NOT NULL	VARCHAR2(50)

desc importprod

Name	Null?	Type
PROD_CODE	NOT NULL	NUMBER(10)
PROD_NAME	NOT NULL	VARCHAR2(50)
PRODGR_CODE	NOT NULL	NUMBER(10)
SALE_PRICE	NOT NULL	NUMBER(10)
TARGET	NOT NULL	NUMBER(10)
DIRECT_SALES		NUMBER(10)
INDIRECT_SALES	NOT NULL	NUMBER(20)
PROFIT_MARGIN	NOT NULL	NUMBER(10)



TABLE STRUCTURE

desc state

Name	Null?	Type
STATE_CODE	NOT NULL	NUMBER(30)
STATE_NAME	NOT NULL	VARCHAR2(50)

desc city

Name	Null?	Type
CITY_CODE	NOT NULL	NUMBER(10)
CITY_NAME	NOT NULL	VARCHAR2(50)
STATE_CODE	NOT NULL	NUMBER(30)



TABLE STRUCTURE

desc salesman

Name	Null?	Type
EMP_CODE	NOT NULL	NUMBER(10)
EMP_NAME	NOT NULL	VARCHAR2(50)
AGE	NOT NULL	NUMBER(2)
DATE_JOIN	NOT NULL	DATE
SEX		VARCHAR2(10)
SALARY	NOT NULL	NUMBER(6)
MARRIED		VARCHAR2(10)
REPORTS_TO	NOT NULL	NUMBER(6)

desc customer

Name	Null?	Type
CUST_CODE	NOT NULL	NUMBER(10)
CUST_NAME	NOT NULL	VARCHAR2(50)
CITY_CODE	NOT NULL	NUMBER(10)
CREDIT_RATING	NOT NULL	VARCHAR2(50)



TABLE STRUCTURE

desc ordtable

Name	Null?	Type
ORDER_NUMBER	NOT NULL	NUMBER(10)
ORDER_DATE	NOT NULL	DATE
CUST_CODE	NOT NULL	NUMBER(5)

▪ desc orddetails

Name	Null?	Type
ORDER_NUMBER	NOT NULL	NUMBER(10)
PROD_CODE	NOT NULL	NUMBER(4)
QUANTITY	NOT NULL	NUMBER(5)



CONSTRAINTS

PARENT TABLE	COLUMNS	UNIQUE	PRIMARY KEY	FOREIGN KEY	REFERENCES
Dept Table	Dept_id	✓			
	Dept_code		✓		
Emp table	Emp_code		✓		
	Dept_code			✓	Dept Table
Product table	Prod_code		✓		
	Direct_sales	✓			
	Prodgr_code			✓	Product group Table
Product group	Prodgr_code		✓		
Import products	Prod_code		✓		
	Direct_sales	✓			
	Prodgr_code			✓	Product group Table
state	State_code		✓		




PARENT TABLE	COLUMNS	UNIQUE	PRIMARY KEY	FOREIGN KEY	REFERENCES
City Table	City_code		✓		
	state_code			✓	State Table
Salesman table	Emp_code	✓	✓		
	Reports_to			✓	Salesman Table
Customer table	Cust_code	✓	✓		
Orders table	Order_number		✓		
	Cust_code			✓	Customer Table
Order detail	Order_number		✓	✓	Orders Table
	Prod_code			✓	Product Table



PROCEDURES

- ❖ Is a type of subprogram that performs an action
- ❖ Can be stored in the database as a schema object
- ❖ Promotes reusability and maintainability

```
CREATE [OR REPLACE] PROCEDURE procedure_name  
[(parameter1 [mode] datatype1,  
parameter2 [mode] datatype2,...)]  
IS | AS  
[local_variable_declarations; ...]  
BEGIN  
-- actions;  
END [procedure_name];
```



PL/sql block



1.Procedure to change the employees Manager

- Create or replace function Procedure reports to(did varchar2)

2.Procedure to add department

- Create or replace function Procedure add_dept(dcode varchar2,dname varchar2)

3.Procedure to add new product group

- Create or replace function Procedure add_prodgrp(prdgpname varchar2)

4.Procedure to know employee and salary details

- Create or replace function Procedure query_emp(id in emptable.emp_code%type,name outemptable.emp_name%type,salary out emptable.salary%type)

5.Procedure to know imported products

- Create or replace function Procedure impprod(sale in importprod.sale_price%type,units out importprod.target%type,margin out importprod.profit_margin%type)



FUNCTIONS

- ❖ Is a named PL/SQL block that returns a value
- ❖ Can be stored in the database as a schema object for repeated execution
- ❖ Is called as part of an expression or is used to provide a parameter value
- ❖ The PL/SQL block must have at least one RETURN statement.

SYNTAX

- ❖ **CREATE [OR REPLACE] FUNCTION function_name**
[(parameter1 [model] datatype1, ...)]
RETURN datatype IS|AS
[local_variable_declarations; ...]
BEGIN
-- actions;
RETURN expression;
END [function_name];



1.Function to Know the products prices greater than the given price

- Create or replace function prod(price number)
- Return type number

2.Function to know the number of orders placed by customer

- Create or replace function total orders(id number)
- Return type number

3.Function to know the actual sale price by removing profit margin

- Create or replace function margin(id number)
- Return type number

4.Function to know the total count of products present in the product group

- Create or replace function totalprod (id number, tprod out number)
- Return Type number



PACKAGES

- ❖ PL/SQL packages:
 - ❖ Group logically related components:
- ❖ PL/SQL types
 - ❖ Variables, data structures, and exceptions
 - ❖ Subprograms: procedures and functions
- ❖ Consist of two parts:
 - ❖ A specification
 - ❖ A body
- ❖ SYNTAX:

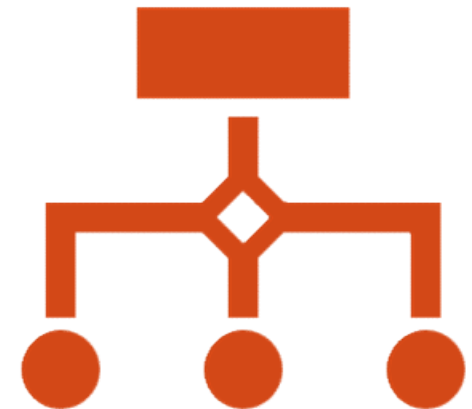
```
CREATE [OR REPLACE] PACKAGE <package_name>
IS
<sub_program and public element declaration>
.
END <package name>
```



1.Package to show the number of employees working in each department

- Package consists of one Function and one Procedure

```
Create or replace package no_emp
is
function user_valid_deptno(p_dno varchar2)
return boolean;
procedure show_strength(p_deptno
    emptable.dept_code%type);
end no_emp;
```



2. creating the package to see the most purchased product

- Package consists of one Procedure one function

create or replace package find_product

As

function max_purchased;

Return number

procedure display_Product;

End Find_product;

