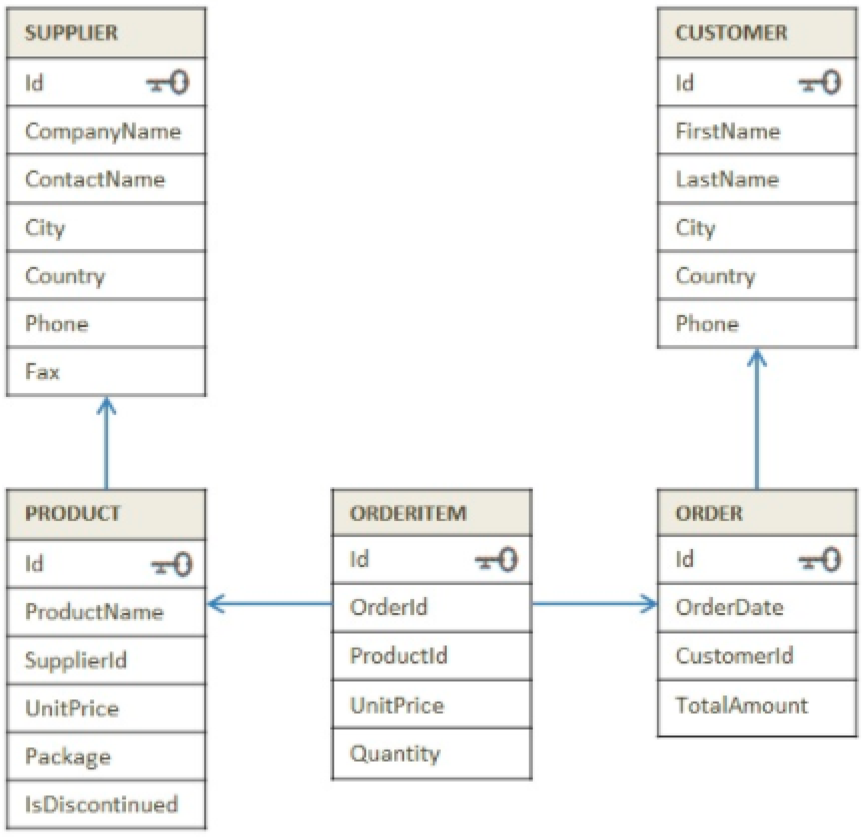
CSE-2007 LAB Assignment

Academic year: 2020-2021 **Slot :** L51-L52

Faculty Name: Ms. P Dhanavanthini mam **Date:** 26/3/2021

Student name: M.Taran **Reg. no.:** 19BCE7346

Data base used :



Database ERD

Syntax :

```
create table Customer (
   Id                     int                    identity,
   FirstName              nvarchar(40)           not null,
   LastName               nvarchar(40)           not null,
   City                 nvarchar(40)         null,
   Country              nvarchar(40)         null,
   Phone                nvarchar(20)         null,
   constraint PK_CUSTOMER primary key (Id)
);

create table "Order" (
   Id                   int                  identity,
   OrderDate            datetime             not null default
getdate(),
   OrderNumber          nvarchar(10)         null,
   CustomerId           int                  not null,
   TotalAmount          decimal(12,2)        null default 0,
   constraint PK_ORDER primary key (Id)
);

create table OrderItem (
   Id                   int                  identity,
   OrderId              int                  not null,
   ProductId            int                  not null,
   UnitPrice            decimal(12,2)        not null default 0,
   Quantity             int                  not null default 1,
   constraint PK_ORDERITEM primary key (Id)
);


create table Product (
   Id                   int                  identity,
   ProductName          nvarchar(50)         not null,
   SupplierId           int                  not null,
   UnitPrice            decimal(12,2)        null default 0,
   Package              nvarchar(30)         null,
   IsDiscontinued       bit                  not null default 0,
   constraint PK_PRODUCT primary key (Id)
);

create table Supplier (
   Id                   int                  identity,
   CompanyName          nvarchar(40)         not null,
   ContactName          nvarchar(50)         null,
   ContactTitle         nvarchar(40)         null,
   City                    nvarchar(40)         null,
   Country                 nvarchar(40)         null,
   Phone                   nvarchar(30)         null,
   Fax                     nvarchar(30)         null,
   constraint PK_SUPPLIER primary key (Id)
);


alter table "Order"
   add constraint FK_ORDER_REFERENCE_CUSTOMER foreign key
(CustomerId)
      references Customer (Id)
go

alter table OrderItem
   add constraint FK_ORDERITE_REFERENCE_ORDER foreign key
(OrderId)
      references "Order" (Id)
go

alter table OrderItem
   add constraint FK_ORDERITE_REFERENCE_PRODUCT foreign key
(ProductId)
      references Product (Id)
go

alter table Product
   add constraint FK_PRODUCT_REFERENCE_SUPPLIER foreign key
(SupplierId)
      references Supplier (Id)
go
```

```
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(62,'L
úcia','Carvalho','San Paulo','Brazil','(11) 555-1189');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(63,'H
orst','Kloss','Cunewalde','Germany','0372-035188');
INSERT INTO [Customer]
```

```
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(64,'S
ergio','Gutiérrez','Buenos Aires','Argentina','5123-8555');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(65,'P
aula','Wilson','Albuquerque','USA','(505) 555-5939');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(66,'M
aurizio','Moroni','Reggio Emilia','Italy','0522-556721');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(67,'J
anete','Limeira','Rio de Janeiro','Brazil','(21) 555-3412');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(68,'M
ichael','Holz','Genève','Switzerland','0897-034214');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(69,'A
lejandra','Camino','Madrid','Spain','(91) 745 6200');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(70,'J
onas','Bergulfsen','Stavern','Norway','07-98 92 35');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(71,'J
ose','Pavarotti','Boise','USA','(208) 555-8097');
INSERT INTO [Customer]
([Id],[FirstName],[LastName],[City],[Country],[Phone])VALUES(72,'H
ari','Kumar','London','UK','(171) 555-1717');


//

INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(11,'Heli Süßwaren GmbH & Co. KG','Petra
Winkler','Berlin','Germany','(010) 9984510',NULL);
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(12,'Plutzer Lebensmittelgroßmärkte AG','Martin
Bein','Frankfurt','Germany','(069) 992755',NULL);
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(13,'Nord-Ost-Fisch Handelsgesellschaft mbH','Sven
Petersen','Cuxhaven','Germany','(04721) 8713','(04721) 8714');
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(14,'Formaggi Fortini s.r.l.','Elio
Rossi','Ravenna','Italy','(0544) 60323','(0544) 60603');
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(15,'Norske Meierier','Beate
Vileid','Sandvika','Norway','(0)2-953010',NULL);
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(16,'Bigfoot Breweries','Cheryl Saylor','Bend','USA','(503)
555-9931',NULL);
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(17,'Svensk Sjöföda AB','Michael
Björn','Stockholm','Sweden','08-123 45 67',NULL);
INSERT INTO [Supplier] ([Id],[Company Name],[Contact
Name],[City],[Country],[Phone],[Fax])VALUES(18,'Aux joyeux
ecclésiastiques','Guylène Nodier','Paris','France','(1)
03.83.00.68','(1) 03.83.00.62');
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(19,'New England Seafood Cannery','Robb
Merchant','Boston','USA','(617) 555-3267','(617) 555-3389');
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(20,'Leka Trading','Chandra
Leka','Singapore','Singapore','555-8787',NULL);
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(21,'Lyngby Sold','Niels
Petersen','Lyngby','Denmark','43844108','43844115');
INSERT INTO [Supplier]
([Id],[CompanyName],[ContactName],[City],[Country],[Phone],[Fax])V
ALUES(22,'Zaanse Snoepfabriek','Dirk
Luchte','Zaandam','Netherlands','(12345) 1212','(12345) 1210');

//
```

```
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(7,'Uncle Bob''s Organic Dried Pears',3,30.00,'12 - 1
lb pkgs.',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(8,'Northwoods Cranberry Sauce',3,40.00,'12 - 12 oz
jars',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(9,'Mishi Kobe Niku',4,97.00,'18 - 500 g pkgs.',1);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(10,'Ikura',4,31.00,'12 - 200 ml jars',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(11,'Queso Cabrales',5,21.00,'1 kg pkg.',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(12,'Queso Manchego La Pastora',5,38.00,'10 - 500 g
pkgs.',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(13,'Konbu',6,6.00,'2 kg box',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(14,'Tofu',6,23.25,'40 - 100 g pkgs.',0);
INSERT INTO [Product]
([Id],[ProductName],[SupplierId],[UnitPrice],[Package],[IsDisconti
nued])VALUES(15,'Genen Shouyu',6,15.50,'24 - 250 ml bottles',0);

//

INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
11,'Jul 17 2012 12:00:00:000AM',20,2018.60,'542388');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
12,'Jul 18 2012 12:00:00:000AM',13,100.80,'542389');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
13,'Jul 19 2012 12:00:00:000AM',56,1746.20,'542390');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
14,'Jul 19 2012 12:00:00:000AM',61,448.00,'542391');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
15,'Jul 22 2012 12:00:00:000AM',65,624.80,'542392');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
16,'Jul 23 2012 12:00:00:000AM',20,2464.80,'542393');
INSERT INTO [Order]
([Id],[OrderDate],[CustomerId],[TotalAmount],[OrderNumber])VALUES(
17,'Jul 24 2012 12:00:00:000AM',24,724.50,'542394')


//

INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(46,16,30
,20.70,60);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(47,16,74
,8.00,36);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(48,17,2,
15.20,35);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(49,17,41
,7.70,25);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(50,18,17
,31.20,30);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(51,18,70
,12.00,20);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(52,19,12
,30.40,12);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2145,830
,39,18.00,2);
```

```
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2146,830
,41,9.65,3);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2147,830
,46,12.00,3);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2148,830
,52,7.00,2);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2149,830
,55,24.00,2);
INSERT INTO [OrderItem]
([Id],[OrderId],[ProductId],[UnitPrice],[Quantity])VALUES(2150,830
,60,34.00,2);
```

//2. Retrieve all the customers' information for a particular Country

```
SELECT * FROM Customer WHERE Country="Germany";
```

//3. Get Customer name along with his Phone number and Country.

```
SELECT FirstName, LastName, Phone, Country FROM Customer;
```

//4.Retrieve the Order whose OrderDate is '19-JUL-2012'.

```
SELECT Id, customerId, OrderNumber, TotalAmount FROM Order WHERE
OrderDate='19-JUL-2012';
```

//5. Get ProductName of the Products without duplications.

```
SELECT DISTINCT SALARY, FIRST_NAME, LAST_NAME FROM EMPLOYEE;
```

//6. Retrieve the Id, Country, ContactName of the Supplier of 'Tokyo Traders' CompanyName.

```
SELECT Id, Country,ContactName FROM Supplier WHERE
CompanyName='Tokyo Traders';
```

//7. Change the UnitPrice of an Product having ProductName as 'Pavlova' to 20

```
UPDATE Product SET UnitPrice=3 WHERE ProductName='Pavlova';
```

//8. Alter Table Customer add column PinCode of NUMBER data type and insert values into this column only.

```
ALTER TABLE Customer ADD PinCode NUMBER(10);
UPDATE Customer SET PinCode=12342346 WHERE Country='France';
UPDATE Customer SET PinCode=11346817 WHERE Country='Austria';
UPDATE Customer SET PinCode=19342380 WHERE Country='Spain';
UPDATE Customer SET PinCode=11234592 WHERE Country='Sweden';
UPDATE Customer SET PinCode=12123456 WHERE Customer='Italy';
```

//9. Change table Customer by modifying the size of field Phone.

```
ALTER TABLE Customer MODIFY Phone NUMBER(11);
```

//10. Modify the field name Phone of Customer table to PhoneNo.

```
ALTER TABLE Customer RENAME COLUMN Phone TO PhoneNo.
```

//11. Change name of Table Product to ProductDetails

```
RENAME Product TO ProductDetails;
```

//12. Alter Table Customer by removing column PinCode.

```
ALTER TABLE Customer DROP COLUMN PinCode;
```

//13. Create a table COPYOFOrderItem as a copy of the table OrderItem.

```
CREATE TABLE COPYOF_OrderItem AS
   SELECT *
    FROM OrderItem;
```

//14. Remove the rows from COPYOF OrderItem table with department number as 5.

```
DELETE FROM COPYOF_OrderItem WHERE ProductId=764;
```

//15. Remove COPYOF DEPT table.

```
DROP TABLE COPYOF_OrderItem;
```

//16. Delete all the rows from COPYOF OrderItem table.

```
DELETE FROM COPYOF_OrderItem WHERE ProductId=764;OrderItem;
```

//17. Remove COPYOF OrderItem table.

```
DROP TABLE COPYOF_OrderItem;
```

//18. Add Foreign Keys using Alter Table

```
ALTER TABLE
    ADD [address] FOREIGN KEY
    [index_name] (col_name, ...)
    REFERENCES tbl_name (col_name,...)
    [ON DELETE reference_option]
    [ON UPDATE reference_option]
```

//19. Drop Foreign key defined on SuperSSN and add it using Alter table command.

```
ALTER TABLE tbl_name DROP FOREIGN KEY fk_symbol;
```

//20. Find the ProductName having UnitPrice greater than Rs.25.00.

```
ALTER TABLE tbl_name DROP FOREIGN KEY fk_symbol;M Product
```

```
WHERE UnitPrice>25.00;
```

//21. Find the ProductName whose UnitPrice lies in the range between 80 and 90.

```
SELECT Id, ProductName, UnitPrice
    FROM Product
    WHERE UnitPrice BETWEEN 80 AND 90;
```

//22. Find the CompanyName who have no Fax.

```
SELECT Id, CompanyName
    FROM Supplier
    WHERE Fax IS NULL;
```

//23. Display the birthday of all employees in the format "DDthMonthYYYY'

```
SELECT FIRST_NAME, LAST_NAME, oderdate
    FROM order;
```

//24. Display the employee names whose OrderDate is on or before 1978.

```
SELECT CustomerId, TotalAmount, OrderDate
    FROM Order
    WHERE OrderDate<'01-12-2012';
```

//25. Display the ContactName having 'Melbourne' in their City.

```
 SELECT CompanyName, ContactName, City
    FROM Supplier
    WHERE City LIKE '%Melbourne%';
```

//26. Display the CompanyName that starts with 'M'.

```
SELECT CompanyName
```

```
    FROM Supplier
    WHERE CompanyName LIKE 'M____';
```

//27. Display the ContactName that ends with 'E'.

```
SELECT CompanyName, ContactName
    FROM Supplier
    WHERE ContactName LIKE '%e';
```

//28. Display the names of all the employees having Phone number with any of the following Phone number (0241-039123, 7675-3425).

```
SELECT FirstName, Phone
    FROM EMPLOYEE
    WHERE Phone IN (0241-039123, 7675-3425);
```

//29. Display all the CompanyName in upper case and lower case

```
SELECT UPPER(CompanyName)
    FROM Supplier;

SELECT LOWER(CompanyName)
    FROM Supplier;
```

//30. Display the first four characters and last four of the department names using the substring function.

```
SELECT SUBSTR('department',1,4)
    FROM DUAL;
SELECT SUBSTR('department',2,4)
    FROM DUAL;
```

//31. Display the substring of the CompanyName (starting from 5th position to 11 th position) of all employees.

```
SELECT SUBSTR(CompanyName, 5, 7)
    FROM Supplier;
```

//32. Display the OrderDate on adding three months to it.

```
SELECT OrderDate, Order+91
    FROM Order;

SELECT ADD_MONTHS(OrderDate, 3)
    FROM Order;
```

//33. Display the due of all the OrderDate rounded to two digits.

```
SELECT CustomerId, ROUND((MONTHS_BETWEEN(SYSDATE, OrderDate)/12),
2)
    FROM Order;
```

//34. Find the last day and next day of the month in which each manager has joined.

```
SELECT SUB_MONTHS(OrderDate, 1), ADD_MONTHS(OrderDate,1)
    FROM Order;
```

//35. Print a substring from the string 'Harini'.

```
SELECT SUBSTR(Harini, 5, 7)
    FROM DUAL;
```

//36. Replace the string 'ni' from 'Harini' by 'sh'

```
SELECT REPLACE('Harini', 'ni', 'sh'
    )
    FROM DUAL;
```

//37. Print the length of all the ProductNames.

```
SELECT ProductName, LENGTH(ProductName)
    FROM Product;
```

//38. Print the system date in the format 25 th May 2007.

```
SELECT TO_CHAR (SYSDATE, 'DDth "OF" fmmonth yyyy')
    from dual;
```

//39. Display the date after 10 months from the current date.

```
SELECT ADD_MONTHS(SYSDATE, 10)
    FROM DUAL;
```

//40. Display the next occurrence of Friday in this month.

```
SELECT NEXT_DAY(SYSDATE, 'Friday')
    FROM DUAL;
```

//41. How many different ProductNames are there in the Product table

```
select count(distinct ProductName) from Product;
```

//42. For each Product display the minimum and maximum Products UnitPrice

```
select min(UnitPrice),max(UnitPrice) from Product;
```

//43. Print the average annual TotalAmount from orders.

```
select avg(TotalAmount*12) from Order;
```

//44. Count the number of orders from 5 years.

```
select count(OrderNumber) from Order where (abs(extract(year from
sysdate)-(extract(year from OrderDate)))>5);
```

//45. Print the ProductName and average UnitPrice of each Product.

```
select ProductName, avg(UnitPrice) from Product group by
SupplierId;
```

//46. Create a view to display the Supplier details who is working from 'germany' Country

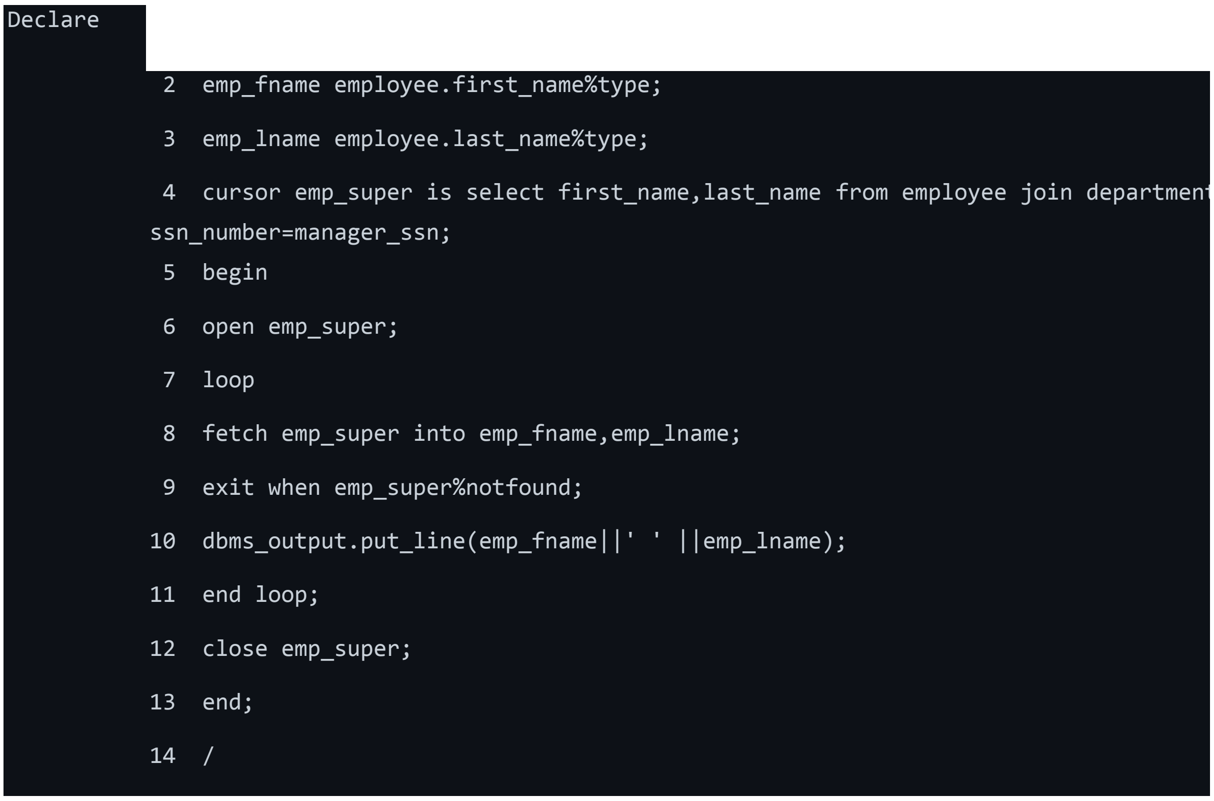
```
select Id, ProductName,SupplierId,UnitPrice from Supplier where
SupplierId =(select Id from department where Country='germany');
```

//47. Create a logical table to store employee details who are getting salaries more than 10000.

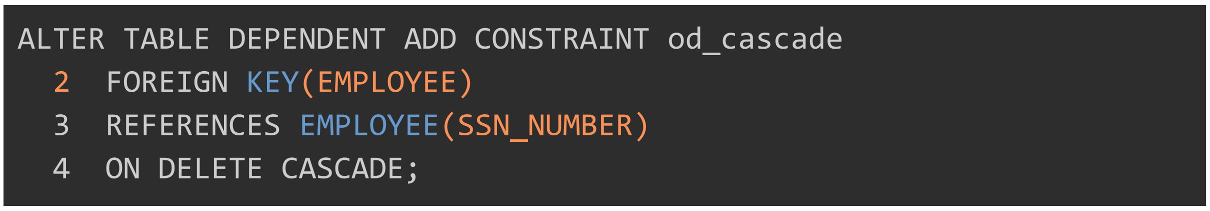
```
declare
n number;
result number;
begin
n :=&n;
result:=facto(n);
dbms_output.put_line('Factorial of '||n||' is '||result);
end;
/
```

//48. Create a table to store the employees details based on the department no

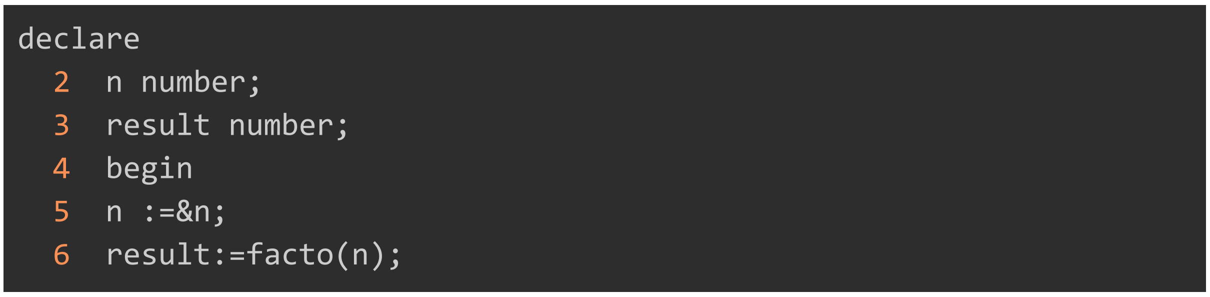
//49. List the names of all managers who have no dependents.


```
Declare
  2  emp_fname employee.first_name%type;
  3  emp_lname employee.last_name%type;
  4  cursor emp_super is select first_name,last_name from employee join departmen
ssn_number=manager_ssn;
  5  begin
  6  open emp_super;
  7  loop
  8  fetch emp_super into emp_fname,emp_lname;
  9  exit when emp_super%notfound;
 10  dbms_output.put_line(emp_fname||' ' ||emp_lname);
 11  end loop;
 12  close emp_super;
 13  end;
 14  /
```

//50. List the employee's names and the department names if they happen to manage a department.


```
ALTER TABLE DEPENDENT ADD CONSTRAINT od_cascade
  2  FOREIGN KEY(EMPLOYEE)
  3  REFERENCES EMPLOYEE(SSN_NUMBER)
  4  ON DELETE CASCADE;
```

//60. Write a PL/SQL program to retrieve the employees working in DNO-5 and increase their salary by 10%


```
declare
  2  n number;
  3  result number;
  4  begin
  5  n :=&n;
  6  result:=facto(n);
```

```
dbms_output.put_line('Factorial of '||n||' is '||result);
end;
/
```

//61. Write a PL/SQL cursor

PL/SQL Basic LOOPIn this loop structure, sequence of statements is enclosed between the LOOP and the END LOOP statements. At each iteration, the sequence of statements is executed and then control resumes at the top of the loop.

//62. Study of PL/SQL Conditional Statements

IF - THEN statementThe IF statement associates a condition with a sequence of statements enclosed by the keywords THEN and END IF. If the condition is true, the statements get executed and if the condition is false or NULL then the IF statement does nothing.

IF-THEN-ELSE statementIF statement adds the keyword ELSE followed by an alternative sequence of statements. If the condition is false or NULL, then only the alternative sequence of statements gets executed. It ensures that either of the sequence of statements is executed.

IF-THEN-ELSEIF statementIt allows you to choose between several alternatives.

Case statementLike the IF statement, the CASE statement selects one sequence of statements to execute.

However, to select the sequence, the CASE statement uses a selector rather than multiple Boolean expressions. A selector is an expression whose value is used to select one of several alternatives.

Searched CASE statementThe searched CASE statement has no selector, and it's WHEN clauses contain search conditions that yield Boolean values.

nested IF-THEN-ELSEYou can use one IF-THEN or IF-THEN-ELSE IF statement inside another IF-THEN or IF-THEN-ELSIF statement(s).

//63. Study of PL/SQL Loops

PL/SQL Basic LOOPIn this loop structure, sequence of statements is enclosed between the LOOP and the END LOOP statements. At each iteration, the sequence of statements is executed and then control resumes at the top of the loop.

PL/SQL WHILE LOOPRepeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

PL/SQL FOR LOOPExecute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

Nested loops in PL/SQLYou can use one or more loops inside any other basic loop, while, or for loop.

//a. Study of PL/SQL procedures and functions

PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms –

Functions – These subprograms return a single value; mainly used to compute and return a value.

Procedures – These subprograms do not return a value directly; mainly used to perform an action.
> A standalone procedure can be called in two ways –
Using the EXECUTE keyword
> Calling the name of the procedure from a PL/SQL block

A function is the same as a procedure except that it returns a value.

//b. Study of PL/SQL cursors

CURSORS Oracle uses ork areas called private SQL areas to execute SQL Statements PL/SQL allows naming the private work areas and accessing the stored information.
The PL/SQL construct to identify each and every work unit area is called cursors. There are user defined cursors, defined in the DECLARE section of the PL/SQL block

here are two types of cursors 1 .Explicit Cursors(user defined Cursors) 2.1mplicit Cursors AND.Explicit Cursors(user defined Cursors) Queries that returns more than one row is called Explicit Cursors

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it.

//c. Write a program using PL/SQL to raise Triggers

```
CREATE OR REPLACE TRIGGER t
  BEFORE
    INSERT OR
    UPDATE OF salary, department_id OR
    DELETE
  ON employees
BEGIN
  CASE
    WHEN INSERTING THEN
      DBMS_OUTPUT.PUT_LINE('Inserting');
    WHEN UPDATING('salary') THEN
      DBMS_OUTPUT.PUT_LINE('Updating salary');
    WHEN UPDATING('department_id') THEN
      DBMS_OUTPUT.PUT_LINE('Updating department ID');
    WHEN DELETING THEN
      DBMS_OUTPUT.PUT_LINE('Deleting');
  END CASE;
END;
/
```

//d. Write a PL/SQL program to handle Exceptions

Exception handling In PL/SQL warning or error conditions is called exception. exceptions either be internally defined by run time systems or can be user defined When an error occurs,an exception is raised,the normal execution stops and the control Transfers to the exception handling part of the PL/SQL.Block. Internal exceptions are raised automatically User defined exceptions must be raised explicitly using the RAISE Statements The RAISE statements can also be used to raise internal exceptions explicitly,