GROFERS

Department Table: Contains Information about Departments

Column Name	Data type	Constraints	Description
DEPT_ID	Number(10)	Unique id	Unique dept id
DEPT_CODE	Varchar2(50)	Not null PK	Enter the dept code

Ans:

```
CREATE TABLE department (
   dept_id   NUMBER(10) UNIQUE,
   dept_code VARCHAR2(50) NOT NULL PRIMARY KEY
);
```

Employee Table: Contains Information about Employee

Column Name	Data type	Constraints	Description
EMP_CODE	Number(10)	Unique id,PK	Unique EMP CODE
EMP_NAME	Varchar2(50)	Not null	Name of the employee
DEPT_CODE	Varchar2(50)	Not Null FK	Department code
GRADE	Varchar2(50)	Not null	Grade of employee
AGE	Number(10)	Not null	Age of employee
DATE_JOIN	DATE	Not null	Date of joining
SEX	Varchar2(50)	Not null	Sex of employee
SALARY	Number(6)	Not null	Salary of employee
REPORTS_TO	Number(6),fk	Not null	reports to whom

Ans:

```
CREATE TABLE employee (
   emp_code  NUMBER(10) PRIMARY KEY,
   emp_name  VARCHAR2(50) NOT NULL,
   dept_code VARCHAR2(50) NOT NULL
      REFERENCES department ( dept_code ),
   grade     VARCHAR2(50) NOT NULL,
   age       NUMBER(10) NOT NULL,
```

```
    date_join DATE NOT NULL,
    sex       VARCHAR(10) CHECK ( sex IN ( 'M', 'F', 'm', 'f' ) ),
    salary    NUMBER(6) NOT NULL,
    reports_to NUMBER(6)
       REFERENCES employee ( emp_code )
);
```

Product Table: Contains Information about product Details

Column Name	Data type	Constraints	Description
PROD_CODE	Number(10)	Unique id,PK	Enter the product code
PROD_NAME	Varchar2(50)	Not null	Name of the Product
PRODGR_CODE	Number(10)	Not Null,FK	Product group code
SALE_PRICE	Number(10)	Not null	Sale price of product
TARGET	Number(10)	Not null	Product targets
DIRECT_SALES	Number(10)	Unique id	Direct sale of product
INDIRECT_SALES	Number(20)	Not null	Indirect Sales of product
PROFIT_MARGIN	Number(10)	Not null	Profit margin of product
BRAND_NAME	Varchar2(50)	Not null	Brand Name of product

Ans:

```
CREATE TABLE products (
    prod_code      NUMBER(10) PRIMARY KEY,
    prod_name      VARCHAR2(50) NOT NULL ,
    prodgrp_code    NUMBER(10) NOT NULL
       REFERENCES productgroup ( prodgrp_code ),
    sale_price     NUMBER(10) NOT NULL,
    target         NUMBER(10) NOT NULL,
    direct_sales   NUMBER(10) UNIQUE,
    indirect_sales NUMBER(20) NOT NULL,
    profit_margin  NUMBER(10) NOT NULL,
    brand_name     VARCHAR2(50) NOT NULL
);
```

Product Group: Contains Information about product Group

Column Name	Data type	Constraints	Description
PRODGR_CODE	Number(10)	Not null PK	Enter the Product group code
PRODGR_NAME	Varchar2(50)	Not Null,Unique	Enter the Product groupname
DEPT_CODE	Varchar2(50)	Not null fk	Enter the dept code

Ans:

```
CREATE TABLE productgroup (
   prodgrp_code NUMBER(10) PRIMARY KEY,
   dept_code    VARCHAR2(20) NOT NULL
      REFERENCES department ( dept_code ),
   prodgrp_name VARCHAR2(50) NOT NULL UNIQUE
);
```

Customer: Contains Information about Customer

Column Name	Data type	Constraints	Description
CUST_CODE	Number(10)	PK	Unique customer code
CUST_NAME	Varchar2(50)	Not null, Unique id	Name of the customer
CITY_CODE	number(10)	Not null	City code of customer
STATE_NAME	Varchar2(50)	Not null	State name
CREDIT_RATING	Varchar2(50)	Not null	Credit rating of customer

Ans:

```
CREATE TABLE customer (
   cust_code     NUMBER(10) PRIMARY KEY,
   cust_name     VARCHAR2(50) NOT NULL UNIQUE,
   city_code     NUMBER(10) NOT NULL,
   state_name    VARCHAR2(10) NOT NULL,
```

```
    credit_rating VARCHAR2(50) NOT NULL CHECK ( credit_rating IN (
'good', 'very good', 'satisfactory', 'fair', 'unsatisfactory',
                                                'excellent' ) )
);
```

Orders: Contains Information about Orders of Product

Column Name	Data type	Constraints	Description
ORDER_NUMBER	Number(10)	PK, Not Null	Unique order id
ORDER_DATE	Date	Not null	Enter Date of Order
CUST_CODE	Number(5)	Not null,fk	Enter Customer code
PROD_CODE	Number(4)	fK,Not null	Unique Product Id
QUANTITY	Number(5)	Not null	Quantity of Products

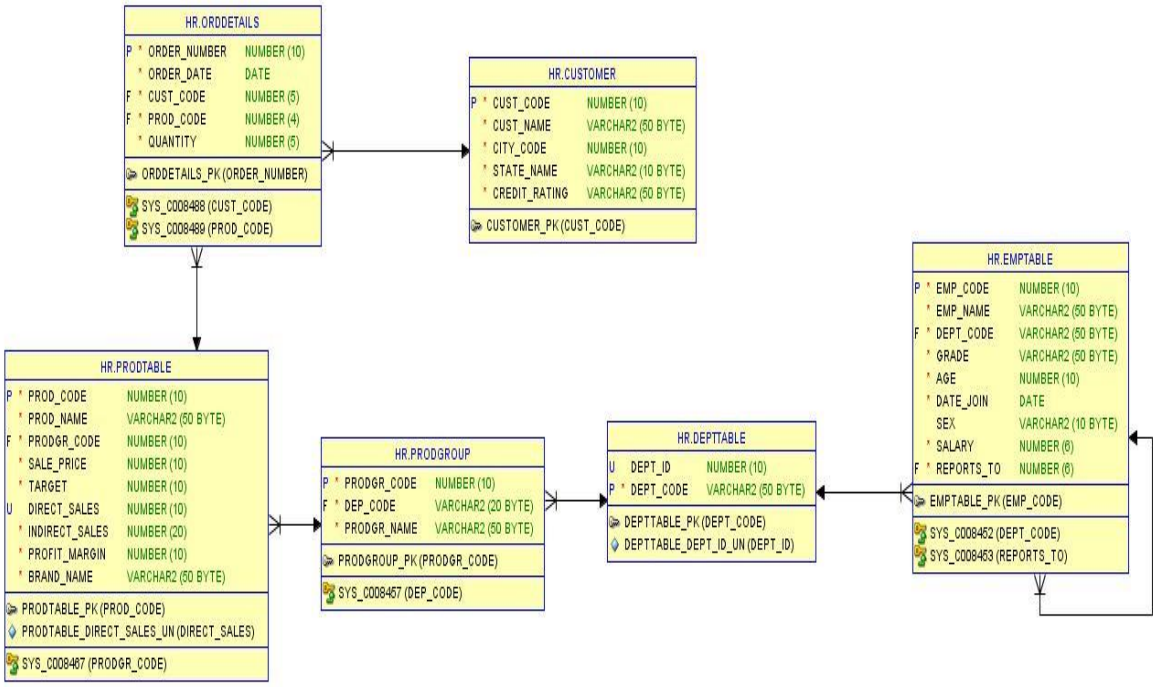
Ans:

```
CREATE TABLE orderdetails (
    order_number NUMBER(10) NOT NULL PRIMARY KEY,
    order_date   DATE NOT NULL,
    cust_code    NUMBER(5) NOT NULL
       REFERENCES customer ( cust_code ),
    prod_code    NUMBER(4)
       REFERENCES products ( prod_code )
    NOT NULL,
    quantity     NUMBER(5) NOT NULL
);
```

Step 3:

Show the relationships and constraints in the tables

Ans:



Step 4:

create sequences and views on the tables to be used in case study.

--sequence and views

sequences

create sequence customer_code start with 1

increment by 1

minvalue 1

maxvalue 99999999

cache 3

nocycle;

```
create sequence orderno start with 10100
increment by 10
minvalue 10100
maxvalue 9999999999999
cache 20
nocycle;

create sequence prodgrpcode start with 1
increment by 1
minvalue 1
maxvalue 999
cache 2;

create sequence deptid start with 10
increment by 10
minvalue 10
maxvalue 99999
cache 10;
```

Inserting The data

Step 5:

Data has to be inserted on the basis of constraints on tables. Use sequences for columns having unique and sequence values in it.

Ans:

Inserting values into employee

--dept it-D1001,accounting-D1002,finance-D1003,sales-D1004,marketing-D1005,

insert into employee values (101,'SMITH','D1001','A',21,'06-09-12','F',24000,null);

insert into employee values (102,'ALLEN','D1007','B',25,'07-05-14','M',19000,null);

insert into employee values (103,'WARD','D1004','C',25,'01-01-13','M',11000,null);

insert into employee values (104,'JONES','D1002','A',20,'26-06-10','F',23000,101);

insert into employee values (105,'BLAKE','D1003','D',28,'14-05-19','M',15000,102);

insert into employee values (106,'CLARK','D1004','B',29,'20-09-15','F',19000,104);

insert into employee values (107,'SCOTT','D1002','C',25,'18-04-14','M',14000,105);

insert into employee values (108,'KING','D1003','A',32,'25-01-17','F',15000,107);

insert into employee values (109,'TURNER','D1001','B',26,'15-08-13','M',17000,106);

insert into employee values (110,'ADAMS','D1003','D',30,'01-12-17','F',11000,105);

insert into employee values (111,'JAMES','D1005','B',27,'26-05-19','F',15000,107);

insert into employee values (112,'MILLER','D1004','A',29,'11-04-10','M',24000,111);

insert into employee values (113,'FORD','D1001','B',28,'30-10-14','M',19000,106);

insert into employee values (114,'MARTIN','D1002','C',25,'15-07-16','M',12000,109);

insert into employee values (115,'MARK','D1005','A',24,'28-02-13','F',23000,113);

insert into employee values (116,'John','D1006','A',31,'08-07-17','F',23000,106);

insert into employee values (117,'William','D1006','C',31,'28-02-10','M',13000,115);

insert into employee values (118,'Foster','D1007','B',28,'15-06-14','F',19000,116);

select * from employee;

	EMP_CODE	EMP_NAME	DEPT_CODE	GRADE	AGE	DATE_JOIN	SEX	SALARY	REPORTS_TO
1	101	SMITH	D1001	A	21	06-09-12	F	24000	(null)
2	102	ALLEN	D1007	B	26	07-05-14	M	19000	(null)
3	103	WARD	D1004	C	25	01-01-13	M	11000	(null)
4	104	JONES	D1002	A	20	26-06-10	F	23000	101
5	105	BLAKE	D1003	D	28	14-05-19	M	15000	102
6	106	CLARK	D1004	B	29	20-09-15	F	19000	104
7	107	SCOTT	D1002	C	25	18-04-14	M	14000	105
8	108	KING	D1003	A	32	25-01-17	F	15000	107
9	109	TURNER	D1001	B	26	15-08-13	M	17000	106
10	110	ADAMS	D1003	D	30	01-12-17	F	11000	105
11	111	JAMES	D1005	B	27	26-05-19	F	15000	107
12	112	MILLER	D1004	A	29	11-04-10	M	24000	111
13	113	FORD	D1001	B	28	30-10-14	M	19000	106
14	114	MARTIN	D1002	C	25	15-07-16	M	12000	109
15	115	MARK	D1005	A	24	28-02-13	F	23000	113
16	116	John	D1006	A	31	08-07-17	F	23000	106
17	117	William	D1006	C	31	28-02-10	M	13000	115
18	118	Foster	D1007	B	28	15-06-14	F	19000	116

Inserting into department table

insert into department values(deptid.nextval,'D1004');

insert into department values(deptid.nextval,'D1001');

insert into department values(deptid.nextval,'D1003');

insert into department values(deptid.nextval,'D1002');

insert into department values(deptid.nextval,'D1005');

insert into department values(deptid.nextval,'D1006');

insert into department values(deptid.nextval,'D1007');

insert into department values(deptid.nextval,'D1008');

insert into department values(deptid.nextval,'D1009');

select * from department;

select deptid.currval from dual;

	DEPT_ID	DEPT_CODE
1	10	D1004
2	20	D1001
3	30	D1003
4	40	D1002
5	50	D1005
6	60	D1006
7	70	D1007
8	80	D1008
9	90	D1009

Inserting into productgroup table

insert into productgroup values(prodgrpcode.nextval,'D1001','Accessories Supplies');

insert into productgroup values(prodgrpcode.nextval,'D1002','Camera Photo');

insert into productgroup values(prodgrpcode.nextval,'D1003','Wearable Technology');

insert into productgroup values(prodgrpcode.nextval,'D1004','Video Game Consoles');

insert into productgroup values(prodgrpcode.nextval,'D1005','eBook Readers');

insert into productgroup values(prodgrpcode.nextval,'D1006','GPS Navigation');

insert into productgroup values(prodgrpcode.nextval,'D1008','Television Video');

insert into productgroup values(prodgrpcode.nextval,'D1007','Computer accessories');

select * from productgroup;

select prodgrpcode.currval from dual;

	PRODGRP_CODE	DEPT_CODE	PRODGRP_NAME
1	1	D1001	Accessories Supplies
2	2	D1002	Camera Photo
3	3	D1003	Wearable Technology
4	4	D1004	Video Game Consoles
5	5	D1005	eBook Readers
6	6	D1006	GPS Navigation
7	7	D1008	Television Video
8	8	D1007	Computer accessories

Inserting into products table:

insert into products values(1001,'Phone Holder',1,50,100,10,50,15,'mivi');

insert into products values(1002,'Digital Picture Frame',1,80,50,20,30,10,'mivi');

insert into products values(1003,'USB Charging Cable',1,75,200,50,80,7,'mi');

insert into products values(1004,'Selfie Stick Tripod',2,150,10,13,20,13,'poco');

insert into products values(1005,'MicroSD Card',2,90,50,23,50,8,'mi');

insert into products values(1006,'HDMI Cable',2,100,30,45,30,6,'karbon');

insert into products values(1007,'35 mm Lense',2,120,30,8,20,16,'intel');

insert into products values(1008,'Light Stand',2,70,25,12,50,6,'poco');

insert into products values(1009,'Smartwatch',3,2000,85,56,70,15,'boat');

insert into products values(1010,'Bluetooth Earphones',3,2500,65,55,90,18,'boat');

insert into products values(1011,'Bluetooth Earphones',3,1500,85,58,60,17,'mivi');

insert into products values(1012,'Bluetooth Earphones',3,1900,45,52,40,20,'oneplus');

insert into products values(1013,'Overhead Headphones',3,3000,35,48,20,22,'readgear');

insert into products values(1014,'Remote Controller',4,2600,75,98,60,26,'xbox');

insert into products values(1015,'Game Console',4,8500,100,111,80,29,'xbox');

insert into products values(1016,'eBook Reader',5,1000,50,11,60,19,'kindle');

insert into products values(1017,'Reader Case',5,900,40,19,40,18,'spare');

insert into products values(1018,'GPS Navigator',6,1900,20,36,40,20,'max');

insert into products values(1019,'Key Finder',6,1700,30,7,25,22,'fax');

insert into products values(1020,'4K Led TV',7,29000,90,84,40,18,'mi');

insert into products values(1021,'Streaming Stick',7,3300,120,101,50,26,'oneplus');

insert into products values(1022,'External Hard Drive',8,1500,30,29,30,17,'intel');

insert into products values(1023,'Graphic Card',8,2500,70,49,40,18,'ryzen');

insert into products values(1024,'Laptop',8,25000,100,79,50,20,'intel');

insert into products values(1025,'Keyboard',8,1500,150,77,100,19,'intel');

insert into products values(1026,'Mouse',8,500,70,80,50,14,'ryzen');

insert into products values(1027,'Mouse',8,500,60,40,30,17,'hp');

insert into products values(1028,'16 GB Ram',8,250,30,16,30,11,'micro');

insert into products values(1029,'Webcam',8,2000,60,81,40,20,'acer');

insert into products values(1030,'Laptop',8,30000,70,60,85,25,'acer');

insert into products values(1031,'Laptop',8,75000,100,99,49,30,'apple');

insert into products values(1032,'Monitor',8,35000,75,66,10,25,'apple');

select * from products;

	PROD_CODE	PROD_NAME	PRODGRP_CODE	SALE_PRICE	TARGET	DIRECT_SALES	INDIRECT_SALES	PROFIT_MARGIN	BRAND_NAME
1	1001	Phone Holder	1	50	100	10	50	15	mivi
2	1002	Digital Picture Frame	1	80	50	20	30	10	mivi
3	1003	USB Charging Cable	1	75	200	50	80	7	mi
4	1004	Selfie Stick Tripod	2	150	10	13	20	13	poco
5	1005	MicroSD Card	2	90	50	23	50	8	mi
6	1006	HDMI Cable	2	100	30	45	30	6	karbon
7	1007	35 mm Lense	2	120	30	8	20	16	intel
8	1008	Light Stand	2	70	25	12	50	6	poco
9	1009	Smartwatch	3	2000	85	56	70	15	boat
10	1010	Bluetooth Earphones	3	2500	65	55	90	18	boat
11	1011	Bluetooth Earphones	3	1500	85	58	60	17	mivi
12	1012	Bluetooth Earphones	3	1900	45	52	40	20	oneplus
13	1013	Overhead Headphones	3	3000	35	48	20	22	readgear
14	1014	Remote Controller	4	2600	75	98	60	26	xbox
15	1015	Game Console	4	8500	100	111	80	29	xbox
16	1016	eBook Reader	5	1000	50	11	60	19	kindle
17	1017	Reader Case	5	900	40	19	40	18	spare
18	1018	GPS Navigator	6	1900	20	36	40	20	max
19	1019	Key Finder	6	1700	30	7	25	22	fax

Inserting data into customer

insert into customer values(customer_code.nextval,'Christian Sanders',10025,'Iraq','good');

```
insert into customer values(customer_code.nextval,'Piper
Richardson',10525,'Brazil','very good');

insert into customer values(customer_code.nextval,'Everly
Walker',10962,'Denmark','fair');

insert into customer values(customer_code.nextval,'Penelope
Guerrero',18505,'italy','unsatisfactory');

insert into customer values(customer_code.nextval,'William
Foster',85625,'hongkong','good');

insert into customer values(customer_code.nextval,'Jade
Rojas',10525,'Kenya','fair');

insert into customer values(customer_code.nextval,'David
Chu',18962,'france','excellent');

insert into customer values(customer_code.nextval,'Sofia
Cheng',18242,'germany','excellent');

insert into customer values(customer_code.nextval,'Audrey
Richardson',10632,'france','satisfactory');

insert into customer values(customer_code.nextval,'Leo
Herrera',18952,'germany','unsatisfactory');

insert into customer values(customer_code.nextval,'Robert
Wright',17654,'Denmark','fair');

insert into customer values(customer_code.nextval,'Scarlett
Kumar',13254,'canada','good');

insert into customer values(customer_code.nextval,'Lillian
Khan',18965,'Brasilia','excellent');

select * from customer;
```

	CUST_CODE	CUST_NAME	CITY_CODE	STATE_NAME	CREDIT_RATING
1	1	Christian Sanders	10025	Iraq	good
2	2	Piper Richardson	10525	Brazil	very good
3	3	Everly Walker	10962	Denmark	fair
4	4	Penelope Guerrero	18505	italy	unsatisfactory
5	5	William Foster	85625	hongkong	good
6	6	Jade Rojas	10525	Kenya	fair
7	7	David Chu	18962	france	excellent
8	8	Sofia Cheng	18242	germany	excellent
9	9	Audrey Richardson	10632	france	satisfactory
10	10	Leo Herrera	18952	germany	unsatisfactory
11	11	Robert Wright	17654	Denmark	fair
12	12	Scarlett Kumar	13254	canada	good
13	13	Lillian Khan	18965	Brasilia	excellent

Inserting data into orderdetalis

```
insert into orderdetails values(orderno.nextval,'04-06-18',1,1004,2);
insert into orderdetails values(orderno.nextval,'05-02-12',2,1006,1);
insert into orderdetails values(orderno.nextval,'13-02-13',13,1009,5);
insert into orderdetails values(orderno.nextval,'19-01-19',6,1026,7);
insert into orderdetails values(orderno.nextval,'21-08-16',9,1024,2);
insert into orderdetails values(orderno.nextval,'28-01-12',5,1016,5);
insert into orderdetails values(orderno.nextval,'10-06-19',2,1014,3);
insert into orderdetails values(orderno.nextval,'18-09-16',3,1016,2);
insert into orderdetails values(orderno.nextval,'06-10-18',4,1025,5);
insert into orderdetails values(orderno.nextval,'14-06-15',8,1020,1);
insert into orderdetails values(orderno.nextval,'29-11-10',9,1023,9);
insert into orderdetails values(orderno.nextval,'20-05-18',10,1001,2);
insert into orderdetails values(orderno.nextval,'19-02-14',11,1005,6);
insert into orderdetails values(orderno.nextval,'06-12-16',13,1028,1);
insert into orderdetails values(orderno.nextval,'12-04-10',12,1031,1);
insert into orderdetails values(orderno.nextval,'18-08-17',10,1003,10);
insert into orderdetails values(orderno.nextval,'08-07-15',11,1015,5);
select * from orderdetails;
```

	ORDER_NUMBER	ORDER_DATE	CUST_CODE	PROD_CODE	QUANTITY
1	10100	04-06-18	1	1004	2
2	10110	05-02-12	2	1006	1
3	10120	13-02-13	13	1009	5
4	10130	19-01-19	6	1026	7
5	10140	21-08-16	9	1024	2
6	10150	28-01-12	5	1016	5
7	10160	10-06-19	2	1014	3
8	10170	18-09-16	3	1016	2
9	10180	06-10-18	4	1025	5
10	10190	14-06-15	8	1020	1
11	10200	29-11-10	9	1023	9
12	10210	20-05-18	10	1001	2
13	10220	19-02-14	11	1005	6
14	10230	06-12-16	13	1028	1
15	10240	12-04-10	12	1031	1
16	10250	18-08-17	10	1003	10
17	10260	08-07-15	11	1015	5

views

1. View of employee Working in department D1006.

```
create or replace view v_employee
as
select emp_code,emp_name,dept_code from
employee where dept_code='D1006';


select * from v_employee;
```

2 View of Products having profit margin greater than 10.

```
create or replace view v_products
as
select * from products
where profit_margin>10;


select * from v_products;
```

3.View for order details with quantity is equal to 2.

```
create or replace view v_orderdetails
as
select prod_code,order_number from orderdetails
where quantity=2;

select * from v_orderdetails;
```

Step 6:

Show data on the basis of joining the tables and sub queries to be used

Ans:

```
--sub queries
```

1.Subquery to know the employees details greater than the average salary.

```
select emp_code,emp_name,date_join
from employee
where salary >
(select avg(salary) from employee)
order by salary;
```

2.Subquery to know the employees details whose name has J and A in between.

```
select emp_code,emp_name,grade
from employee
where dept_code in
(select dept_code from employee where emp_name like '%J%' and emp_name like '%A%');
```

3.Subquery to know the employees details whose department code is D1004

```
select emp_name,dept_code,grade
```

```
from employee
where dept_code in
(select dept_code from department WHERE dept_code='D1004');
```

4.find those employees who get second-highest salary

```
select *  from employee
where emp_code in
(select emp_code from employee where salary in
(select max(salary) from employee where salary < (select max(salary) from employee)));
```

5.Subquery to know the product details where product group is computer accessories.

```
select prod_code,sale_price,target
from products
where prodgrp_code in
(select prodgrp_code from productgroup where prodgrp_name = 'Computer accessories');
```

--joins

1.To get the employee details using Various joins.

```
select emp_name,salary,e.dept_code
from employee  e  join department d
on e.dept_code = d.dept_code;
```

```
-- natural join
select emp_name,salary,dept_code
from employee natural join department;
```

--using clause

```
select emp_name,salary,dept_code
from employee   join department
using(dept_code);
```

2.To get the order details using joins where quantity greater than 2.

```
select order_number,quantity,order_date from
orderdetails left outer join customer using(cust_code)
where quantity >2;
```

3.To get the details of products where profit margin is greater than 10.

```
select prodgrp_name,direct_sales,indirect_sales,target from
products right outer join productgroup
using(prodgrp_code)
where profit_margin>10;
```

Procedures

Step 7:

create Procedures on the basis of the case study which has to be used frequently and stored in database permanently

1.To add a new department

--ADD department

```
create or replace procedure add_dept(dcode department.dept_code%type)
as
deptcode exception;
```

```
pragma exception_init(deptcode,-00001);
dept exception;
begin
if (dcode is not null) then
insert into  department values(deptid.nextval,dcode);
dbms_output.put_line('----------processing----------');
dbms_output.put_line('added department '||' '||dcode);
elsif (dcode='') then
raise dept;
else
raise deptcode;
end if;
exception
when dept then
dbms_output.put_line('invalid department');
when deptcode then
dbms_output.put_line('department exists');
end;

exec add_dept('D1004');
```

2.To add a new product group into table

```
--ADD Group
create or replace procedure add_prodgrp(dcode varchar2,prdgpname varchar2)
as
grpexcep exception;
pragma exception_init(grpexcep,-00001);
```

```
begin
if(prdgpname is not null and dcode is not null) then
insert into  productgroup values(prodgrpcode.nextval,dcode,prdgpname);
dbms_output.put_line('---------------processing--------------');
dbms_output.put_line('added new product '||' '||'product name'||' '||prdgpname);
else
raise grpexcep;
end if;
exception
when grpexcep then
dbms_output.put_line('product group name exists');
end;

exec add_prodgrp('D1002','Wearable Technology');
```

3.To know the employee’s name and salary from emptable

```
--Employee_details
create or replace procedure query_emp(id in employee.emp_code%type,
name   out employee.emp_name%type,
salary out employee.salary%type)
is
begin
select   emp_name, salary into name, salary
from    employee
where   emp_code = id;
exception
when no_data_found then
```

```
dbms_output.put_line('Employee id does not exists');
end query_emp;

declare
emp_name employee.emp_name%type;
emp_sal  employee.salary%type;
begin
query_emp(142, emp_name, emp_sal);
dbms_output.put_line(emp_name||' '||emp_sal);
end;
```

4.To get details of specific customer order details

```
--Order details
create or replace procedure orddate(cutno orderdetails.cust_code%type,ord_date
out orderdetails.order_date%type,quant  out orderdetails.quantity%type)
is
begin
declare
cursor c1 is
select * from orderdetails where cust_code=cutno;
begin
for rec in c1 loop
dbms_output.put_line('customer code:'||rec.cust_code||', '||'order
date:'||rec.order_date||', '||'quantity:'||rec.quantity);
end loop;
end;
end;
```

```
declare
odate date;
quan number(10);
begin
orddate(11,odate,quan);
end;
```

Functions

Step 8:

Create Functions to make task easier for future transactions like calculation of orders

1.function to know the products greater the given sale price

```
create or replace function prod(price number)
return number
is
cursor c1 is select profit_margin,brand_name from products where sale_price>price;
margin products.profit_margin%type;
brand products.brand_name%type;
begin
open c1;
loop
fetch c1 into margin,brand;
dbms_output.put_line('profit margin '||margin||','||'brand_name '||brand);
exit when c1%notfound;
end loop;
close c1;
return margin;
```

```
end;

exec dbms_output.put_line(prod(2400));

2.function to know the orders placed by customer
--Totalorders
create or replace function totalorders(id number)
return number
is
o_count number(10);
begin
select count(order_number) into o_count from orderdetails
where cust_code=id;
return o_count;
end;
--method -1
variable ocount number
exec :ocount := totalorders(3)
print :ocount

--method -2
declare
ordcount number;
begin
ordcount := totalorders(9);
dbms_output.put_line(ordcount);
end;
```

```
--method - 3
exec dbms_output.put_line(totalorders(10));

--method -4
select totalorders(11) from dual;
3.function to know the actul sale price by removing profitmargin
create or replace function margin(id number)
return number
is
c_price number;
price products.sale_price%type;
profit products.profit_margin%type;
begin
select sale_price,profit_margin into price,profit from products
where prod_code=id;
c_price :=price-(price*profit/100);
return c_price;
exception
when no_data_found then
dbms_output.put_line('product code not listed');
return c_price;
end;

exec dbms_output.put_line(margin(1032));

select margin(1024) from dual;
```

```
select * from products;
```

4.function to know the total count of products present in the productgroup

```
create or replace function totalprod(id number,totprod out number)
return number
is
avgprice number;
begin
select count(prod_code) into tprod
from products
where prodgrp_code = id;
return tprod;
end;

variable totprod number
exec :totprod := totalprod(8,:totprod);
print :totprod
```

Packages:

Step 9:

Create packages of all the required cursor, exception, procedure, f unction etc. for easy access of data.

1. package to show the number of employees working in each dept_code

```
create or replace package num_emp
is
function user_valid_deptno(p_dno varchar2)
```

```
return boolean;
procedure show_strength(p_deptno employee.dept_code%type);
end num_emp;

create or replace package body num_emp
is
function user_valid_deptno(p_dno varchar2)
return boolean
as
did department.dept_code%type;
begin
select count(*) into did from department where dept_code = p_dno;
if did = 1 then
dbms_output.put_line('deptno exists');
return true;
else
dbms_output.put_line('deptno doesnot exists');
return false;
end if;
end user_valid_deptno;
procedure show_strength(p_deptno employee.dept_code%type)
as
ans number;
invalid_dept exception;
begin
if(user_valid_deptno(p_deptno)) then
select count(emp_code) into ans from employee where dept_code =p_deptno;
```

```
dbms_output.put_line('employees in dept '||p_deptno ||' is '|| ans);
else
raise invalid_dept;
end if;
exception
when invalid_dept then
dbms_output.put_line('invalid dept');
end show_strength;
end;

exec num_emp.show_strength('D1004');
```

2. creating the package to see the most purchased product

```
create or replace package find_product as
procedure display_Product;
end;
create or replace package body find_product as
function max_purchased
return number
as
prodid number:=0;
begin
select prod_code into prodid  from orderdetails where quantity=  (select
max(quantity) from orderdetails);
dbms_output.put_line('product selected by max customers is '||prodid);
return prodid;
end max_purchased;
procedure display_product
```

```
as
maxprodid number;
v_prodname products.prod_name%type;
v_pcode  products.prod_code%type;
v_pgrpcode productgroup.prodgrp_code%type;
v_pgrpname productgroup.prodgrp_name%type;
begin
maxprodid:=max_purchased;
select prod_name,prod_code into v_prodname,v_pcode from products where
prod_code=maxprodid;
dbms_output.put_line('Product code  : '||maxprodid||' Product Name :
'||v_prodname);
select prodgrp_code into v_pgrpcode from products where
prod_code=maxprodid;
select prodgrp_name into v_pgrpname from productgroup where
prodgrp_code=v_pgrpcode;
dbms_output.put_line('product group name : '||v_pgrpname);
end display_product;
end;

exec find_product.display_product;
```