

Name :- Sakshi chowasiya

Year :- 2nd year

Subject :- DBMS

Branch :- CSE.

Semester :- 3rd Sem

Serial no.	Experiment Description	Experiment Date	Submission Date	Remarks / Signature..
1.	Create tables and Specify the given SQL			
2.	manipulate the operations on the table			
3.	implement the restrictions on table			
4.	implement the structure of the table			
5.	implement the concept of join's			
6.	implement the concept of grouping of Data			,
7.	Implement the concept of Sub queries			,
8.	Implement the concept of Indexes and views			,
9.	Implement the basics of PL/SQL			,
10.	Implement the concept of cursor and trigger			,

* Experiment :- 1 *

* Objective :- Create tables and specify the given Queries in SQL

Q.1 Create the following tables:-

(i) Client master

Column name	datatype	Size
client - no	varchar 2	6
name	varchar 2	20
address 1	varchar 2	30
address 2	varchar 2	30
city	varchar 2	15
state	varchar 2	15
pincode	number	6
bal-due	number	10,2

Q.2 Insert the following data into their respective tables

client no.	name	city	pincode	state	bal-due
0001	Ishu	Bombay	400054	maharashtra	15000
0002	Vandana	Madras	480001	Tamilnadu	0
0003	Purnima	Bombay	400057	maharashtra	5000
0004	Basel	Bombay	400056	maharashtra	0
0005	Ravi	Delhi	100001		2000
0006	Rukmini	Bombay	400050	maharashtra	0

iii) Product - master

Column name	datatype	Series
product-no	varchar 2	
Description	varchar 2	
Profit - percent	number	
Unit - measure	varchar 2	
Qty - on - hand	number	
Reorder - number	number	
Sell - price	number	
Cost - price	number	

* Data for product master :

Product no	Description	Profit % - Percent	Unit measured	Qty on hand	Recd
P00001	1.44 floppies	5	Piece	100	20
P03453	monitors	6	Piece	10	3
P06784	mouses	5	Piece	20	5
P07865	1.22 floppies	5	Piece	100	80
P07885	keyboards	2	7T	10	3
P07965	CD Drive	2.5	7T	10	3
P07975	500 HDD	4	7T	10	3
P08865	1.44 Drive	5	7T	10	3
P08865	1.22 Drive	5	7T	2	3

sell price cost price

595	8400	500	8000
12000	1050	11200	10000
1050	1050	500	10000
525		500	
3150		3050	
5280		5100	

TABLE : ch
Create table ch
client-no varchar
name name
address1 address1
address2 address2
city city
state state
pincode pincode
bal - bal -
};
*

TABLE : client-master

Create table client-master

```

client-no varchar(6) primary key,
name varchar(20),
address1 varchar(30),
address2 varchar(30),
City varchar(15),
State varchar(15),
pincode number(6),
bal-due number(10,2)
);

```

* Insert the following data into their respective tables:-

Ans :- insert into client-master values ('0001', 'Inah', "", "",
 "Bombay", 'maharashtra', 400054, 15000);

* insert into client-master values ('0002', 'vandana', "", "",
 'madras', 'Tamilnadu', 780001, 0);

* insert into client-master values ('0003', 'poramal', "", "",
 "", 'Bombay', 'maharashtra', 402057, 5000);

* insert into client-master values ('0004', 'Beza', "", "",
 'Bombay', 'maharashtra', 400056, 0);

* insert into client-master values ('0005', 'Reeti', "", "",
 'Delhi', "", 100001, 2000);

* insert into client-master values ('0006', 'Rukmini', "", "",
 "", "Bombay", 'maharashtra', 400050, 0);

ii) TABLE: Product-master
Create table product-master

(
product-no varchar(2) primary key,
description varchar(20),
profit - percent number(5,2),
unit - measure varchar(20),
q_ty - con - hand number(6),
reader - del number(6),
sell - price number(10);
cost - price number(10);
);

→ Data for product master
insert into product master values ('P00001', '1.44 Floppies', 5,
'piece', 100, 20, 525, 500);
insert into product - master values ('P03453', 'monitors', 6,
'piece', 10, 3, 12000, 11200);
insert into product - master values ('P06734', 'mouse', 5,
'piece', 10, 3, 1050, 500);
insert into product - master values ('P07865', '1.22
floppies', 5, 'piece', 100, 20, 525, 500);
insert into product - master value ('P07868', 'keyboards', 2,
'piece', 10, 3, 3150, 8050);
insert into product - master value ('P07895', 'CD Drive', 2.5,
'piece', 10, 3, 3250, 5100);
insert into product - master values ('P07965', '540 HDD',
'4', 'piece', 10, 3, 8400, 8000);
insert into product - master, values ('P07975', '1.44 Drive',
'5', 'piece', 10, 3, 1050, 1000);
insert into product - master values ('P08865', '1.22 Drive',
'5', 'piece', 2, 3, 1050, 1000);

Q.3 On the basis of above two tables answer the following questionies .

(i) find out the names of all the clients

Ans Select name from client - master ;

(ii) Retrieve the list of names and cities of all the clients

Ans Select client-no, name , city from client - master ;

(iii) List the various products available from the product master tables .

Ans Select product-no, description from product - master ;

(iv) List all the clients who are located in Bombay.

Ans Select client-no, name from client - master where city = 'Bombay' ;

(v) Display the information of client no. 001 and 002

Ans Select * from client - master where client - no in (0001, 0002) ;

(vi) Find the product with description as '1.44 Drive and

Ans Select * from product - master whose descriptions in ('1.44 Drive'), ('1.22 Drive') ;

(vii) find all the product whose sell price is greater than 5000.

Ans Select * from product - master where Sell - price > 5000;

(viii) find the list of all clients who stay in city 'Bombay' or city 'Delhi' or 'madras'

Ans Select * from client - master where city in ('Bombay', 'Delhi', 'madras');

(ix) find the product whose selling price is greater than 2000 and less than or equal to 5000.

Ans Select * from product - master where Sell - price between 2000 and 5000;

(x) List the name , city and state of client not in the state of 'maharashtra',

Ans Select name , city , State from client - master where state != 'maharashtra';

* Experiment - 2 *

Objective:- To manipulate the operations on the table.

Question:- Using the table client master and product master answer the following questionries.

- i) Change the selling price of '1.14 floppy disk' to RS. 1150.00
- ⇒ Update product - master Set sell - price = 1150 where description = '1.14 floppies';
- ii) Delete the record with client no=1 from the client master table
- ⇒ delete from client - master where client - no = '0001';
- iii) Change the city of client - no '0005' to Bombay
- ⇒ Add record into client table with this data : ('0001', 'Tata', " ", " ", 'Bombay', 'maharashtra', 400054, 15000)
- Change the bal - due of client - no '0001' to 1000;
- Insert into client - master values ('0001', 'Tata', " ", " ", 'Bombay', 'maharashtra', 400054, 1000);
- iv) Change the bal - due of client - no '0001' to 1000.
- One Update client - master Set bal - due = 1000 where client - no = '0001';

- v) find the products whose selling price is more than 1500
and also find the new selling price as original selling
price * 15.
- ⇒ Select product-no, description, sell-price * 15 as
new sell-price from product-master where
sell-price > 1500;
- vi) find out the clients who stay in a city whose second
letter is q
- ⇒ Select * from client-master where city like ' _ - q % ' ;
- vii) find out the name of all clients having 'q' as the
second letter in their names.
- ⇒ Select * from client-master where name like 'q % ' ;
- viii) the products in sorted order of their description.
- ⇒ Select * from product-master order by description.
- ix) Count the total number of orders
- ⇒ Select sum(Quantity) from product-master;
- x) Calculate the minimum price of products.
- ⇒ Select avg(cost-price) from product-master;
- xi) Calculate the average price of all the products.
- ⇒ Select avg(cost-price) from product-
master;

5, 'piece', 2, 3, 1050, 1000);

xii) Determine the maximum and minimum prices
Rename the title as 'max-price' and 'min-
price' respectively.

⇒ Select max(cost-price) max - price, min(cost-
price) min - price from product-master;

xiii) Count the number of products having price
greater than or equal to 1500.

⇒ Select count(product-no) from product-master
where sell - price >= 1500;

* Experiment :- 3 *

* Objective :- To implement the restrictions on the table

Question :- 2 Create the following tables:

Q) Sales-master

Column name	Datatype	Size	Attributes
Salesman-no	Varchar(2)	6	Primary key first letter must start
Sel-name	Varchar(2)	20	not null
Address	Varchar(2)		not null
City	Varchar(2)	20	
State	Varchar(2)	20	
Pincode	Varchar(2)	6	
Sel-amt	Varchar(2)	8.2	not null, constraint
Tgt-to-get	Varchar(2)	6.2	not null, constraint
Ytd-Sales	Number	6.2	not null, constraint
Remarks	Varchar(2)	30	

Ans:- Salesman - no Varchar(2)(6) Primary key,

Sel-name Varchar(2)(20) not null,

Address Varchar(2)(20) not null,

City Varchar(2)(20),

State Varchar(2)(20),

Pincode Number(6);

Sel-amt Number(8,2) not null,

Tgt-to-get Number(6,2) not null,

Ytd-Sales Number(6,2) not null,

Remarks Varchar(2)(30).

;

insert into Product - master values

(100, 1000);

ii) Sales - Order

Columnname	Datatype	Size	other info
S-order-no	varchar	6	Primary/first letter
S-order-date	Date	6	Primary key reference client no of client - master table.
client-no	varchar	25	
Delivery-add	varchar	6	
Salesman-no	varchar	6	foreign key references Salesman-no of Salesman - master table
Delv-type	char	1	Delivery part (P) / full (F) & cleantf
Billed-yn	char	1	
Delv-date	Date		cannot be less than S-order-date
Order-status	varchar2	10	values ('in process', 'fulfilled', 'back order', 'cancelled')

Ques: CREATE table SALES - ORDERC

S-ORDER - NO VARCHAR2 (6) check (S-ORDER
- NO LIKE '0%'),
S-ORDER - DATE DATE,

CLIENT - NO VARCHAR2 (25),

DELV - ADD VARCHAR2 (6),

SALESMAN - NO VARCHAR2 (6),

DELV - TYPE CHAR (1),

BILLED - YN CHAR (1),

DELV - DATE DATE check (DELV - DATE) > = S - ORDER
- DATE),

ORDER - STATUS VARCHAR₂(10),
 FOREIGN KEY (SALESMAN-NO) REFERENCES SALES-
 MASTER
 FOREIGN KEY (CLIENT-NO) REFERENCES CLIENT-
 MASTER

};

I Sales-order-details

Columns	Datatype	Size	Attributes
S-order-no	VARCHAR ₂	6	primary key foreign key Sales Order
Product-no	VARCHAR ₂	6	primary key foreign key references master
Oty-Datch	Number	8	
Oty-disp	Number	9	
product-mater.	Number	10,2	

Insert the following data into their respective tables
using insert statement:

Date for sales-man master table.

Create table sales_order_details (

l. Order_no varchar(5),

Product_no varchar(2(6)),

Qty - integer number(10),

Org_Cat_number(5)),

product - create number(10,2),

foreign key (l. Order_no) references sales_order,

foreign key (Product_no) references product_master

)

Salesman	Customer Address	City	Phonecode	State	Order No.	Order Date	Order Type	Order Status	Delivery Date	Deliveryman	Order No.	Order Date	Order Type	Order Status	Delivery Date	Deliveryman	Order No.	Order Date	Order Type	Order Status
Goldman	123 Main St	Bombay	40002	Maharashtra	00001	12-Jan-96	F	P	25-Jan-96	00002	015001	12-Jan-96	P	P	25-Jan-96	00003	015002	18-Feb-96	F	F
Kroll	456 London Rd	Bombay	40002	Maharashtra	00002	015002	P	P	015001	00001	015001	12-Jan-96	F	P	25-Jan-96	00004	015004	20-May-96	F	P
Manuel	657 London Rd	Bombay	40002	Maharashtra	00003	015003	F	P	015002	00002	015002	015002	F	F	08-Apr-96	00005	015005	20-Feb-96	F	F
Spencer	768 London Rd	Bombay	40002	Maharashtra	00004	015004	F	P	015003	00003	015003	015003	F	F	27-Jan-96	00006	015006	20-May-96	C	C
Subbarao	"	"	"	"	00005	015005	F	P	015004	00004	015004	015004	F	F	07-Apr-96	00007	015007	20-May-96	C	C
Gold	"	"	"	"	00006	015006	F	P	015005	00005	015005	015005	F	F	10-Jun-96	00008	015008	20-May-96	C	C
Gupta	"	"	"	"	00007	015007	F	P	015006	00006	015006	015006	F	F	10-Jun-96	00009	015009	20-May-96	C	C
Patel	"	"	"	"	00008	015008	F	P	015007	00007	015007	015007	F	F	10-Jun-96	00010	015010	20-May-96	C	C
Sharma	"	"	"	"	00009	015009	F	P	015008	00008	015008	015008	F	F	10-Jun-96	00011	015011	20-May-96	C	C
Verma	"	"	"	"	00010	015010	F	P	015009	00009	015009	015009	F	F	10-Jun-96	00012	015012	20-May-96	C	C
Wong	"	"	"	"	00011	015011	F	P	015010	00010	015010	015010	F	F	10-Jun-96	00013	015013	20-May-96	C	C

⑪ Data from Salesorder table :-

iii) Data for Sales - Order - details table :-

SalesOrder	Qtyordered	ProductNo	ProductNo	Qtyordered	Qtyordered	Qtyordered
019001	4		P00001	4		4
019001	2		P07965	2		1
019001	2		P07886	2		0
019002	10		P00001	10		3
046865	3		P07868	3		10
046865	10		P07885	10		4
019003	4		P00001	4		2
019003	2		P03453	2		1
046866	1		P06734	1		0
046866	1		P07965	1		0
010008	1		P07975	1		5
010008	10		P00001	10		

Insert all

into Sales-master (salesman-no, sal-name, address,
city, state, pincode, sal-amt, qt-to-get, ytd-
sales, remarks)

values ('500001', 'Kiran', 'A/14 woali', 'Bombay', 'Mah',
400002, 3000, 100, 50, 'Good')

into Sales-master

values ('500002', 'Manish', '65, navi man, 'Bombay',
'man', 400001, 3000, 200, 100, 'Good')

into Sales-master

values ('500003', 'Ravi', 'P-7 Bandra', 'Bombay', 'man',
400003, 3000, 200, 100, 'Good')

into Sales-master.

values ('500004', 'Ashish', 'Als yehu', 'Bombay',
'man', 400004, 3000, 200, 150, 'Good')

Select * from deal.

ii) insert all

into Sales - Order (5 - Order - no, 5 - Order - date, all
dely - add, Salesmen - no, dely - type, filled - yes
date, Order - status
values ('019001', 'Jan-12-96', '0001', '1', 'S')
'F', 'N', Jan-20-96', 'IP')

into Sales - Order values ('019002', 'Jan-25-96', '10002',
'S'00002', 'P', 'N', 'Jan-27-96', 'C')
into Sales - Order values ('016865', 'feb-18-96', '10003', '1'
'50003', 'F', 'Y', 'feb-20-96', 'F')

into Sales - Order values ('019003', 'apr-03-96', '10001',
'1', '50001', 'F', 'Y', 'apr-27-96', 'F')
into Sales - Order values ('046866', 'may-20-96', '1'
'0004', '1', '50002', 'P', 'N', 'may-22-96', 'C')

into Sales - Order values ('010008', 'may-24-96', '0005',
'1', '50004', 'F', 'N', 'may-26-96', 'IP')

Select * from deal.

Click insert all
into sales - order - details (s-order-no, product-no, Qty-order, Qty-disp, Product-rate)

values ('019001', 'P00001', 4, 4, 525)

into sales - order - details values ('019001', 'P07965', 2, 1, 8400)

into sales - order - details values ('019001', 'P07885', 2, 1, 5250)

into sales - order - details values ('019002', 'P00001', 10, 0, 525)

into sales - order - details values ('046865', 'P07868', 3, 3, 3150)

into sales - order - details values ('046865', 'P07885', 10, 10, 5250)

into sales - order - details values ('019003', 'P00001', 4, 4, 1050)

into sales - order - details values ('019003', 'P03453', 2, 2, 1050)

into sales - order - details values ('046866', 'P06734', 1, 1, 12000)

into sales - order - details values ('046866', 'P07965', 1, 0, 8400)

into sales - order - details values ('010008', 'P07975', 1, 0, 1050)

into sales - order - details values ('010008', 'P00001', 10, 5, 525)

Experiment :- 4

Objective :- To Implement the structure of the table

Question :- Create the following tables :

Challan Header

column name	datatype	Size	Attributes
challan-no	varchar2	6	Primary key
S-order-no	varchar2	6	foreign key references S- Order table

Challan - date

challan - date	date	not null
litterd - ya	char	values ('Y', 'N') Default 'N'

Table name : challan - Details

Column name	datatype	Size	Attributes
challan - no	varchar2	6	Primary key foreign key references product - no of product master
Qty-disp	number	4,2	not null.

```
create table challan - header (
    challan - no varchar 2(6) primary key ,
    S - Order - no varchar 2(6) ,
    challan - date DATE ,
    blocked - yn char (1) DEFAULT 'N' ;
foreign key (S - order - no) references sales - Order
);
```

Ans=> Create table challan - details (

```
challan - no varchar 2(6) primary key ,
product - no varchar 2(6) ,
Qty - disp number(4,2) not null ,
```

Q2 Insert the following values onto the challan header and challan - details tables.

(i)	Challan No	S-Order No	Challan Date	Billed
	CH 9001	019001	12-DEC-95	4
	CH 3965	046065	12-NOV-95	4
	CH 3965	010008	12-OCT-95	4

Data for challan - details table.

Challan no	Product no	Qty Disp.
CH 9001	P00001	4
CH 9001	P07965	1
CH 9001	P07865	1
CH 6865	P07868	3
CH 6865	P03453	4
CH 6865	P00001	10
CH 3965	P00001	5
CH 3965	P07975	2

insert
into challan - details (challan - no, product - no; qty - size)
value ('CH 3001', 'P00001', 4)

into challan - details values ('CH 3001', 'P07965', 1)

into challan - details values ('CH 3001', 'P07885', 1)

into challan - details values ('CH 6865', 'P07968', 2)

into challan - details values ('CH 6865', 'P00001', 10)

into challan - details values ('CH 6865', 'P00001', 10)

into challan - details values ('CH 6865', 'P00001', 5)

into challan - details values ('CH 3965', 'P07975', 2)

Select * from all

objective - Answer the following Question.

Q.1) make the primary key to client-no in client-master.

Ans alter table client-master

add primary key (client-no);

Q.2) Add a new column phone-no in the client-master table.

Ans alter table client-master

add (phone-no varchar(11));

Q.3) Add the not null constraint on the product-master table with the columns description, profit percent, sell price and cost price.

Ans 1) description

alter table product-master modify
(description varchar(20) not null)

2) profit

alter table product-master modify
(profit_percent number(5,1) not null)

3) sell-price

alter table product-master modify
(sell-price number(20,5) not null)

4) cost-price

alter table product-master modify
(cost-price number(5,0) not null)

Q.4 change the size of client - no field in the client - master table.

Ans : alter table client - master modify
(client - no varchar(10))

Q.5. select product - no , description where profit percent between 20 and 30 with inclusive .

Ans : Select product - no , description
from product - master
where profit - percent between 20 and 30

Experiment :- 5

Objective :- To implement the concept of Joins.

Answer the following question(s) :

Question(s)

1. find out the product which has been sold to Indian buyers.

Ans :- Select

product - master . product - no , product - master
description , sales - order - details . qty - order
from client - master , product - master , sales -
order - details , Sales - Order

where product - master . product - no = Sales - Order -
details . product - no and Sales - Order - details
. S - Order - no = Sales - Order . S - Order - no
and Sales - Order . Client - no = Client - master .
Client - no and Client - master . name like '%W%' ,

2. find out the product and their quantities that will have to delivered.

Ans :- Select product - master . product - no , Qty -
Order from product - master , Sales -
Order - details where product - master .
product - no = Sales - Order details product - no

3. find the product - no and description of memory products

Ans :- Select product - no, description

from product - master

where description like '% floppies %'

union

Select product - no, description

from product - master

where description like '% Drive %'

union

Select product - no, description

from product - master

where description like '% HDD %'.

4. find out the names of clients who have purchased 'CD DRIVE'

Ans :- Select sales - order - details . product - no, Sales - order - details . S - Order - no

from product - master, sales - order - details

where product - master . description = '1.44

'floppies' and sales - order - details . qty - order < 5 and product - master . product - no = sales - order - details . product - no

5. List the product-no and s-order-no of customers having qty ordered less than 5 from the order details table for the product "1.44 floppies"

Ans :- Select client-master-name
 from client-master, product-master, sales-order-details, sales-order
 where product-master.product-no = Sales-order-details.product-no and
 Sales-order-details.no = client-master-client-no and product-master.description,
 = 'CD DRIVE'

6. Find the products and their quantities for the orders placed by 'Vardhan Saitwai' and 'Ivan Bayress'.

Ans :- Select product-master.product-no, product-master.description, sales-order-details, qty-order-from.
 client-master.product, product-master, sales-order-details, sales-order
 where product-master.product-no = Sales-order-details.product-no and
 Sales-order-client-no = client-no master, client-no-and-client-master.name like '%Vardhan%',
 union
 select product-master.product-no, product-master.description, sales-order-details, qty-order-from

client - master, product - master, sales - order
details, sales - order,

where, product - master product - no = Sales - Order
details, product no and
Sales - order details. S-order - no = Sales - orders order
no and

Sale for client no - client master. Client - no and client
master name like 'Y. Inay'.

7. Find the product and their for the orders placed by
client no "5000" and "6000".

8. find the order no, client no and salesman no where
a client has than received by more
than one salesman

Ans:- Select S-order - no, Client - no, Salesman - no
from Sales - order

where Client - no in (Select Client - no)

from Solt Sales - order

group by Client - no
having count (Salesman - no) > 1)

9. Display the S-order - date in the format "dd - mm - yy"
e.g. "12 - feb - 96"

Ans:- select To char (S-order - date, 'DD - mon - yy') li

from sales-order

10. find the date 15 day after date.

ans select to char (s-order-data + 15 'DD-mon-yy)
from sales - order.

Experiment :- 6

Objective :- To implement the concept of grouping of Data.

Answer the following question :-

Q. 1:- Print the description and total quantity sold for each product.

Ans:- With Product Quantity (Product - No, Total - Qty) as
(Select Product - No, sum (Qty - disp) as Total - Qty
group by Product from Sales - Order - Details
group by Product - No)
select Product - Quantity, Product - No, Product -
master, description, quantity, Total - Qty from
Product - Quantity, Product - master where
Product Quantity Product - No = Product - master -
Product - No.

Q. 2 find the value of each product sold.

Ans:- With Product - Quantity (Product - no, Total - Qty)-
as (Select Product - No, sum (Qty - disp) as Total -
Qty from Sales - Order - Details group by Product -
No.)

Select

Product - quantity Product - No, Product - master -
description, Product - quantity, Total - Qty, Product

Quantity . Total - Qty Sales - Order - Details . Product
greater than sales values

from Product - Quantity , Product - master , Sales , and
Details where Product - quantity , Product . No = Product
Master Product - No and Product - quantity . Product
No = Sales - Order - Details . Product - No .

Q.3 calculate the average quantity sold for each client
that has a maximum order value of 15000.

Ans. with Product - Quantity (Product - No , Total - Qty
Ordered , Avg . Qty - Sold) as (Select Product - No - Sum
(Qty - Order .) as Total - Qty - Ordered , Avg (Qty - Disp)
as Avg - Qty - Sold .

from Sales - Order - Details
group by Product - No

Select

distinct (Product . Quantity . Product - No) , Product - master
description , Product - quantity . Total - Qty . Ordered , Product
Quantity . Avg . Qty - Sold , Product - quantity . Total - Qty .
Ordered . Sales - Order - Details Product - o rate as Sold -
values

from Product - quantity , Product - master , Sales - Order
- Details

where Product - quantity . Product - No = Product -
master . Product - No and .

product - quantity, product . no = sales - order - details
product - no and .

Product - quantity, total - qty. ordered. sales - order
details product rate < 15000.

Q4 find out the products which has ever sold to Iran.

Ans :- select product - master . product - no, product - master
description, sales - order details . qty - order from
client - master, product - master, sales - order
details, sales - order

where product - master . product - no = sales -
order - details, Product - no and
Sales - order - details - order - no = sales - orders .
order - no and .

sales - order client - no = client - master . client
no and . client - master . name = 'Iran'

Q5 find the names of client who have (CD Drive)

Ans :- select client - master . name .

from client - master , product - master , sales - order
details , Sales - order

where product - master . product - no = sales . order -
details product - no and .

Sales - order - details . g - order - no = sales - order . 3 -
order - no and .

Sales order. Client - no = client - master . Client - no and . Product - master . description = ' CD Drive '

Q.6 Find the products and their quantities for the orders placed by ' vandang ' and ' ivan '.

Ans : Select product - master . Product - no , Product - master . description , sales - order - details . Ity - order from client - master , product - master , sales - order - details . Sales - order .
Where product - master . Product - no = sales - order - details . Product - no and .

Sales - order - details . ? order - no = sales - orders .
order - no and .

Sales - order . Client - no = client - master . no and
Client - master . name like ' vandang '

union .

Select product - master . Product - no , Product - master . description , sales - order - details . Ity - order from client - master , product - master , sales - order - details . Sales - order .

Where product - master . Product - no = sales - order - details . Product - no and .

Sales - order - details . S - order - no = sales - order . S - order - no and .

sales-order-client → no. client master client
no and client master name like 'Tun'

Q.7 Select product no, total qty-ordered for each product.

Ans: select product no, sum (Qty-order) as total qty-ordered from sales-order-details
group by Product-no

Q.8 select product no, product description and total qty ordered for each product

Ans: with details as (select product-no, sum (Qty-order)
as total-qty-ordered
from sales-order-details
group by Product-no)

select details.product-no, product-master.
description, details.total-qty-ordered
from details, product-master
where details.product-no = product-master.product-no

Q.9 display the orden number and day on which client placed their order.

Ans - Select s-order-no, to-char(s-order-date,
mon-yy') from Sales-order.

Q10 Display the month and Date when the order
must be delivered.

Ans - Select s-order-no, to-char(day-date, 'mon') as
month, to-char(day-date, 'dd') as delay-date
from Sales-order.

Experiment :-

Objective :- To implement the concept of Subquery
functions.

Answer the following questions :-

1. find the product - no and description of non-moving products

Ans Select product - no, description

from product - master

where description like 'monitors'
union

Select product - no, description

from product - master

where description like 'mouse'
union

Select product - no, description

from product - master

where description like 'keyboards'

2. find the customer name, address, city and pincode
for the client who has placed order no "019001"

3. find the client names who have placed
order before the month of may 96 .

4. find out if product 67.44 Drive " is ordered
by only client and print the client - no
name to whom it was sold .

- PAGE NO. _____
DATE: _____
5. find the names of client who have placed orders worth Rs. 10000 or more.
 6. Select the orders placed by "Rahul Desai"
 7. Select the names of person who are in my predecessor's department and who have also worked on an inventory control system.
 8. Select all the clients and the salesman in the city of Bombay,
 9. Select salesman name in "Bombay" who has atleast one client located at "Bombay"
 10. Select the product-no, description, qty-on-hand, Cost - price of non-moving items in the product master table.

from client - master, client - no, client - master, name, client
 client city, client - master, pincode, client - master,
 date.

from client - master, sales - order

where client - master, client - no = sales - order, client - no
 and sales - order, s - order - no = '019001'

SAns with product as (select sales - order - details, product
 no, sales - order - details, s - order - no
 from product - master, sales - order - details

where product - master, description = '1.44 floppies'
 and sales - order - details, qty - order < 5 and product
 - master, product - no = sales - order - details, product - no)

Select client - master, client - no, client - master, name
 from client - master, product, sales - order.

where product, s - order - no = sales - order, s - order
 - no and sale order, client - no = client - master, client
 - no

SAns Select client - master, client - no, client - master, name
 from sales - order - details, sales - order, client - master
 where sales - order - details, qty - order * sales - order -
 details, product - rate > 10000 and

sales - orders, s - order - no = sales - order - details, s -
 order - no and

client - master, client - no = sales - order, client - no.

select sales-order-details · product-no, sales
order - details qty - order

(3)

from sales - order - details , sales - order , client - master
where client - master . name = 'Rahal Desai' and client
master . client - no = sales - order . client - no and
sales - order . order - no = sales - order - details -
order - no

8 Ans Select distinct (client - master . name) , sales - master .

Sal - name

from sales - master , client - master
where client - master , city = 'Bombay' and sales -
master . city = client - master . city .

9 Ans select sales - master . Sal - name , sales - master .
Salesman no , sales - order . client - no

from sales - master , sales - order , client - master
where sales - master . city = 'Bombay' and sales -
master .

Salesman - no = sales - order . Salesman - no and client
master . client - no = sales - order . client - no and client
master . city = 'Bombay'

10 Ans with non - moving as (select product - no ,

description from product - master)

where description like 'monitors'

union

Select product - no , description

from product - master

where description like 'mouse'

Union

Select product - no , description
from product - master
where description like 'keywords')

Select
product - master . Product - no , product - master
description ,
product - master . Qty - on - hand , product -
master . Cast price
from product - master , non - moving
where non - moving . description = product - master
description .

Experiment 6-8

Objective: To implement the of Indexes and views.

Answer the following questions

Q.1 Create an index on the table client-master, field client-no

Q.2 Create an index on the sales - order, field S - order no.

Q.3 Create an composite Index over the Sales - Order - details table for the columns S - Order - no and product - no.

Q.4 Create an composite Index ch - index on challan - header table for the columns challan no and S - order - no.

Q.5. Create an unique Index on the table salesman master, field Salesman - no.

Q.6. Drop Index ch - index on table challan - header.

Q.7. Create view on Sales man - master whose sal - amt is less than 3500.

- Q.8) Create a view client -view on client -master and renamed the columns as name , add 1, add 2, city , state respectively .
- Q.9) Select the client names from client -view who lives by city "Bombay"
- Q.10) Drop the view client -view .

an index on the table client - master, field client - no
CREATE INDEX client - ndxe
ON client - master (client - no);

Create a composite index on the sales - order - details
table for the columns s - order - no and product - no
CREATE INDEX sales - order - details - ndxe
ON sales - order - details (s - order - no, product - no);

2 Ans Create an index on the table sales - order , fields
-order no CREATE INDEX client - ndxe.
ON sales - order (field s - order - no);

4 Ans Create a composite index on the ch - index table
for the challan - header and s - order - no
CREATE INDEX ch - index - ndxe
ON challan - header (challan no, s - order - no);

5 Ans Create table sales - master ||
(salesman - no varchar 2(6) check (salesman - no LIKE 'S%'),
Sal - names varchar 2(20) NOT Null,
Address varchar(20) NOT Null,
City varchar(2(20)),
State varchar(2(20)),
Pincode varchar(2(6)),
Sal - amount number(8,2) Not null,
tgt - to - get number(6,2) not null,
Ytd - sales number(6,2) not null,

41
table cheiller - details
client - no varchar(6) PRIMARY KEY ,
product - no varchar(6),
Hy - disp number(4,2) NOT null)

Ans SQL > insert into Salesman master 2 values.
('50000 ' , ' arshesh ' , ' als ' , ' guh ' , ' bombay ' , ' 400044 ' ,
' maharashtra ')

BAns Create table client - master (clientno varchar(6),
name varchar(20),
Address1 varchar(30),
Address2 varchar(30),
City varchar(20),
Pincode int ,
State varchar(20)
Bal Due int);

gAns SELECT NAME FROM CLIENT - MASTER;

Output :-

NAME
Ivan Bayross
Vandend
Ponawich
Basel
Ram
Rech mani

10 Ans DROP VIEW Client - vka
DROP VIEW AnIdentifier

Experiment :- 9

Objective :- To implement the basics of PL/SQL

- 1) WAP in PL/SQL for addition of two numbers.
- 2) WAP in PL/SQL for addition of 1 to 100 numbers.
- 3) WAP in PL/SQL to check the given number is even or odd.
- 4) WAP in PL/SQL to inverse a number; e.g., number 5639 when inverted must be displayed output 9365.

PAGE NO.:
DATE:
value from n: 456
n number : = 2n;
n number : = 456 ;
is reverse of a number

SQL> declare
a number(2);
b number(2);
c number(2);
begin
a := 2a;
b := 2b;
c := a+b;
dbms_output.put_line ('all '+' '+ b || ' = ' || c);
end;
/
Enter value for a: 44
old a: a := 2a;
new a: a := 44;
Enter value for b: 48
old a: b = 2b;
new a: b := 48;
44 + 48 = 92

2 AND SQL> Declare
a number;
b number := 0;
Begin
for a from 1 to 100 loop
b := b + a;
end loop;
dbms_output.put_line ('TOTAL SUM IS = ' || b);
end;
/
TOTAL SUM IS = 5050

Ans SQL> declare
a number(4);

begin

a := 29;

if mod(a,2)=0 then

dbms_output.put_line('A IS an even NUMBER');

else

dbms_output.put_line('A IS an odd NUMBER'),

end if;

end;

/

Enter value for a: 32

old 4: a: 29;

new 4: a := 32;

A IS an even NUMBER

Ans SQL> declare

n number := 27;

s number := 0;

number;

begin

while (n > 0)

loop

i := n mod 10;

s := (s * 10) + i;

n := floor(n / 10);

end loop;

dbms_output.put_line ('SII: Is reverse of a number');

end;

/

Q. WAP in PLSQL for changing the price of product
'poroor' to 4000 if the price is less than 4000 in
product-mastery table. the change is recorded
in the old-price-table along with product
no and the date on which the price was
changed last.

, declare
number : = 8n ;
number1 : = 4030 ;
number2 ;

begin
while (n) 40000)

loop
q : = n mod '100000';
s : = (8 * 10000) + i;
n : = floor (n / 10000);

end loop;

dbms - output. productmaster ('the old - price table
of product - no);
end;

/
Enter value of n : 40000
old 2 : n number1 : = 8n ;
new 2 : n number1 : = 525 ;

The change is recorded in the old - price - table along
with product - no and the date on which the price
was changed.

Experiment :- 10

objective :- To implement the concept of cursor and trigger.

(Q1) Two tables are there.

Table name : O-Employee

Column name	Datatype.	Size	Attributes
Emp - Code	varchar(12)	10	Primary key
Ename	varchar(12)	20	The name of the candidate
Deptno	Number	5	The Department No
Job	varchar(12)	20	The Employee Job.
Salary	number	8,2	The current salary. of Em.

Table name :- Emp - raise

Column name	Datatype	Size	Attributes
EMP - Code	varchar(9)	10	Primary key
Raise - Date	date		The Date on which the
Raise - Amt	number	8,2	The raise given to the employee .

The HRD manager has decided to raise the salary for all the employees in the Dept no 20 by 5%. whenever only such raise is given to the employee , a record for the same is maintained in the emp - raise table .

It includes the Empno, the date when the raise was given and the actual raise. Write PL/SQL block to update the salary of each employee and the actual raise.

Write a PL/SQL block to update the salary of each employee and insert a record in the emp-raise table.

trigger being triggered
one INSERT ON Sales
EACH ROW

```

DECLARE num-new INTEGER;
DECLARE tot-new INTEGER;
SELECT COUNT(*)
    INTO tot-new
    FROM performance
WHERE employee-id = NEW.employee-id;
SELECT COUNT(*)
    INTO num-new
    FROM Performance
WHERE employee-id = NEW.employee-id;
IF num-new > 0 THEN
    UPDATE Performance
    SET total-sales = NEW-sales - amt + tot-new - sales,
        ave-sale = total-sales / (tot-new + 1)
    WHERE employee-id = NEW.employee-id;
ELSE
    INSERT INTO PERFORMANCE
    (employee-id, name, total-sales, ave-sale)
    VALUES (NEW.employee-id, NEW.name, NEW.sale
            - amt, new-sale - amt);
END IF;

```

Two tables are there

Table name :- client-master

Column name	Date type	size	Attributes
client-no	varchar	6	Primary key / first letter must start not null.
name	varchar	20	
city	varchar	10	
State	varchar	10	
Bal-due	varchar	10,2	

Table name :- audit-client

Column name	Date type	Size	Attributes
client-no	varchar	10	primary key
name	varchar	20	
Bal-Due	varchar	10,2	
operation	number	6	
date	Date		

Create a transparent audit system for a table client master. The system must keep track of the records that are being deleted or modified and when they have been deleted or modified.

CREATE TRIGGER audit-trail
AFTER UPDATE OR DELETE ON client
FOR EACH ROW
DECL ARE

oper varchar2(8);
client-no varchar2(6);
name varchar2(20);
Bal-due number(2);

BEGIN

IF updating then.

oper := 'update';
End if;

IF deleting then

oper := 'delete';

Endif;

client-no := :old-client-no;

name := :old-name;

Bal-due := old-client-due;

Insert into auditclient

values (client-no, name, Bal-due, oper, sysdate);

END;



PAPERS

SESSIONS

Database management System

Experiment :- 1

Aim:- Create tables and Specify the given queries in SQL.

Theory :-

Introduction about SQL :- SQL (Structured Questionsly language) is a nonprocedural language you specify what you want not how to get it. A block structured format of english key words is used in this questionsly language. It has the following components.

DDL (Data Definition Language) :- The SQL DDL provides command for defining relations schemas deleting relations and modifying relations schema.

DML (Data manipulation Language) :- It includes commands to insert tuples into delete tuples from and modify tuples in the database.

View definition :- The SQL DDL includes commands for defining views. Transaction Control - SQL includes for specifying the beginning and ending of transactions.

SESSIONAL PAPERS

Roll No.....

Scholar No..... 2

Embedded SQL and Dynamic SQL :- Embedded and Dynamic SQL define how SQL statements can be embedded with in general purpose programming languages such as C, C++, JAVA, COBOL, pascal and fortran.

Integrity :- The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must specify. updates that violate integrity constraints are allowed.

Authorization :- The SQL DDL includes commands for specifying access rights to relations and views.

Data Definition Language :- The SQL DDL allows specification of not only a set of relations but also information about each relation including

- * Schema for each relation

- * The domain of values associated with each attribute

- * The integrity constraints

- * The set of indices to be maintained for each relation.



Domain types in SQL :- The SQL Standard supports a variety of built-in domain types including

- * char(n) - A fixed length character length string with user specified length.
- * varchar(n) - A variable character length string with user specified maximum length n.
- * int - An Integer
- * smallint - A small Integer
- * numeric (P,d) - A floating point number with precision of at least n digits
- * float (n) - A floating point number with precision of at least n digits
- * Date - A calendar date containing a (four digit) year, month and day of the month.

DDL Statement for creating a table :-

Syntax :-

Create table tablename;

[columnname datatype (size) columnname datatype (size);

Creating a table from a table :-

Syntax .

CREATE TABLE TABLENAME

[(columnname, columnname)]

AS Select columnname, columnname ..., FROM tablename;

SESSIONAL PAPERS

Vishal
PRODUCTS
BHOPALI

Insertion of data into tables :-

Syntax :-

Insert into tablename

[(columnname, columnname, . . .)]
values (expression, expression);

Inserting data into a table from another table :-

Syntax :-

Insert into tablename

SELECT columnname, columnname, . . .
FROM tablename;

Selecting a data set from table data :-

Syntax :-

SELECT columnname, columnname,

FROM tablename

WHERE Searchcondition ;

Roll No..... Scholar No... 4

Barkatul

Vishal
PRODUCTS

Experiment :- 2

Aim :- To manipulate the Operations on the table.

Theory :- DML (Data manipulation Language) Data manipulation is

The retrieval of information stored in the database
The insertion of new information into the database
The deletion of information from the database
The modification in information stored by the appropriate data model. There are two types.

(i) Procedural DML

(ii) Non-procedural DML

Updating the content of a table :- In creation situation we may wish to change a value in table without changing all value tuple. For this purpose the update statement can be used.

Update table name

Set columnname = expression Columnname = expression....

Where columnname = expression;

Deletion Operation :- A delete request is expressed in much the same way as query we can delete whole tuple (rows) we can delete values in only particulars attributes.

SESSIONAL PAPERS

Roll No..... Scholar No. 6

Barkatullah

Deletion of all rows :-

Syntax :-

Delete from tablename;

Deletion of Specified number of rows.

Syntax :-

Delete from tablename

Where search condition;

Computation in expression lists used to select data :-

+ Addition

- Subtraction

* multiplication

* * exponentiation.

/ Division

() Enclosed operation

Renaming columns used with expression lists :- the default output column names can be renamed by the user if required.

Syntax :-

Select column name
columnname

From table name;

result - columnname,

result - columnname,

Logical Operators :- The logical operators that can be used in SQL sentence core.

relational

AND all of must be included
OR any of may be included
NOT none of could be included

Range Searching :- Between operation is used for range searching.

Pattern Searching :- The most commonly used operation on string is pattern matching using the operations like we describe patterns by using two special characters.

* Percent (%) ; the % character matches any Substring we consider the following examples.

* 'P%' matches any string beginning with Perry

* '%dge%' matches any string containing Edge as Substring

* '---' matches any string exactly three characters

* '---%' matches any string of at least of three characters .

Oracle functions :- Functions are used to manipulate data items and return functions the format of functions - name (argument 1, argument 2 ...)

Roll No..... Scholar No..... 8



An arrangement is user defined variable of context.
the structure of function is such that it accepts
zero or more arguments except one.

Avg return average value of n.

Syntax :-

Avg (distinct/all) n)

min return minimum value of except.

Syntax :-

min (distinct/all except)

Count Returns the no of rows where except is not null

Syntax :-

Count (distinct/all) except.

Count (*) Returns the no rows in the table including duplicates and those with nulls.

Max Return max value of except.

ERS

Ans - To implement



PAPERS

FINAL PAPERS

Experiment :- 3

Aim :- To implement the restrictions on the table.

Theory :- Data constraints Besides the cell name cell length and cell data type there are other parameters i.e. other data constraints that can be passed to the DBA at check creation time the constraints can either be placed at column level or at the table level.

(i) Column Level constraints :- If the constraints are defined along with the column definition it is called a column level constraint.

(ii) Table Level constraints :- If the data constraint attached to a specific cell in a table preference the contents of another cell in the table then the user will have to use table level constraints.

Null value concepts :- While creating tables of a grant - Jodis a data value for particular column that column was defined as not null when the table was created.

Syntax

create table tablename

(columnname datatype (size) not null....)

.

SESSIONAL PAPERS

Primary key :- Primary key is one or more columns is a table used to uniquely identify each row in the table.

Primary key values must not be null and must be unique across the column A multicolumn primary key is called composite primary key.

Syntax :- Primary key a column constraint

Create table tablename
(columnname datatype (size) Primary key ...)

Primary key as a table constraint

Create table tablename
(columnname datatype (size) columnname datatype (size)
primary key (columnname, columnname));

* The Setup relation



Uniqueness key concept :- A uniqueness is similar to a primary key except that the purpose of a uniqueness is to ensure that information in the column for each record is unique as with telephone licence numbers. A table may have many uniqueness keys.

Syntax :- Uniqueness as a column constraint :-

Create table tablename -

(columnname datatype (size) uniqueness);

Uniqueness as table constraint :-

Create table tablename

(columnname datatype (size), columnname datatype
(size)) .. . Uniqueness (columnname, columnname);

Default value concept :- At the time of cell creation a default value can be assigned to it when the user is adding a record with values and leaves this cell empty.

DBA will automatically load this cell with the default value specified.

NATIONAL PAPERS

The data type of the default value should match data type of the column.

Syntax :-

Create table tablename ;
(columnname datatype (size) default value : :)

Foreign key concept :- foreign key represents relationship between tables. A foreign key is column whose values are derived from the primary key of the same of some other table. The existence of foreign key implies that the table with foreign key is related to the primary key table from which the foreign key is derived.

Syntax :-

Create table table name .

(columnname datatype (size) references table name);

Experiment :- 4

Objective :- To Implement the structure of the table.

Theory :- Modifying the structure of tables - Alter table command is used to changing the structure of a table. Using alter table clause you cannot perform the following tasks.

- (i) Change the name of table
- (ii) Change the name of column
- (iii) drop a column
- (iv) decrease the size of a table if table data exists.

The following tasks you can perform through alter table command.

(i) Adding new columns :-

Syntax:-

ALTER TABLE tablename
ADD (newcolumnname newdatatype (size));

(ii) Modifying existing table

Syntax:-

ALTER TABLE tablename

SESSIONAL PAPERS

Vishal
PRODUCTS
SHOPAHL

Scholar No. 19
Roll No.
Banka

MODIFY (new columnname need datatype (size));

Note:- Oracle not allow constraints defined using the alter table if the data in the table violates.

Removing/Deleting Tables :- following command is used for removing or deleting a table.

Syntax :-

DROP TABLE tablename ;

Defining Integrity constraints in the ALTER TABLE Command you can also define integrity constraints using the constraint in the ALTER TABLE command.

(P) Add PRIMARY KEY.

Syntax :-

ALTER TABLE tablename
ADD primary key (columnname);

* The key is relation.

Vishal
PRODUCTS
SHOPAHL

ERS

SESSIONAL PAPERS

Barkatullah University Institute of Technology, Bhopal

Roll No..... Scholar No..... 15

(ii) Add Foreign key.

Syntax :-

ALTER TABLE tablename .

ADD constraintname constraintname .

Foreign key (columnname) REFERENCES .

Dropping Integrity constraints in the ALTER TABLE command .

Experiment :- 5

Objective :- To implement the concept of Joins

Theory :- Joint multiple table (Equi Join) Some times we require to treat more than one table as though manipulate data from all the tables as though the tables were not separate object but one single entity. To achieve this we have to join tables.

The tables that have to be joined are specified in the FROM clause the joining attributes in the where clause.

Algorithm for Join in SQL :-

- * Cartesian product of tables (Specified in the from clause)
- * Selection of rows that match (predicate in the where clause)
- * project column Specified in the Select clause.

(1) Cartesian product :-

Consider two table Student course
Select S.* P.*
FROM Student S, course P;

* The Set up relation.

SESSIONAL PAPERS

Scholar No.....

16



Barkatullah University Institute of Technology, Bhopal

Roll No..... Scholar No..... 17

(ii) Inner join :-

Cartesian product followed by selection.

Select B.*P.*

FROM Student B, Courses P

WHERE B.Course # = P.Course #;

(iii) Left outer join :- Left outer join = Cartesian product + selection but include rows from the left table which are unmatched put nulls for the values of attributes belonging to the second table.

(iv) Right outer join :-

RIGHT outer join = Cartesian product + selection but include rows right table which are unmatched.

(v) full outer join :-

F-join

Select B.*P.*

From Student B Full Joining Courses

B.Course # = P.Course #;

SESSIONAL PAPERS

Vishal
PRODUCTS

Vishal
PRODUCTS
Bhopal

Experiment :- 6

Objective :- To implement the concept of grouping of Data.

Theory :- Grouping Data from tables :- There are circumstances where we would like to apply the aggregate function not only to a single set of tuples but also to group of sets of tuples we specify this wish in SQL using the group by clause.

The attribute or attribute given in the group by clause are used to form group tuples with the same value on all attributes in the group by clause are placed in one group.

Syntax :-

```
SELECT columnname columnname  
FROM tablename  
GROUP BY columnname;  
HAVING Searchcondition;
```

At times it is useful to state a condition that applies to group rather than to tuples. For example we might be interested in only those branches where the average account balance is more than 2000.

* The query -

relation .

Roll No..... Scholar No..... 18

Barkatullah

To Prepr
Clause

SD

SESSIONAL PAPERS

Visha
PRODUCTS
Bhopal

Barkatullah University Institute of Technology, Bhopal

Roll No..... Scholar No..... 19

To express such query we use the having clause of SQL. SQL applies predicates on the having may be used.

Syntax :-

```
SELECT columnname, columnname  
FROM tablename  
GROUP BY columnname;  
HAVING somecondition;
```

SESSIONAL PAPERS

Vishal
PRODUCTS
Bhopal

Vishal
PRODUCTS
Bhopal

Experiments

Objective :- To implement the concept of Sub questionaries.

Theory :- Sub Questionaries A Questionries is a form of an SQL statement that appears inside another SQL statement.

It also termed as nested Questionry, the statement containing a sub Questionries called a parent statement the rows returned by the sub questionry are used by the following statements.

It can be used by the following Commands.

- To insert records in the target-table.
- To create tables and insert records in this table.
- To update records in the target-table.
- To create view.

Example:-

Creating client master table from oldclient_master table,

Create table client - master

AS Select * From oldclient - master.

relation.

Roll No.....

Scholar No.: 20

Barkar

Ding 7
Ding 10

S

Using the union, Intersect and minus clause:-
Union clause:-

The user can put together multiple questionnaires and combine their output using the union clause.

The union clause merges the output of two or more questionnaires into a single set of rows and column the final output of union clause will be.

Output = Records only for questionnaire one + records only in questionnaire two + A single set of records with is common in the both questionnaires.

Syntax:-

SELECT columnname , columnname
From tablename |

Union

Select columnname , columnname
From tablename 2

SESSIONAL PAPERS

Vishal
PRODUCTS
Bhopal

Vishal
PRODUCTS
Bhopal

Intersect clause:- The use can put together multiple questionnaires and their output using the interest due the final output of the Interest- clause will be output = a single of records which are common in both questionnaires.

Melaku..

Roll No. Scholar No. 22

Barkatul

objection

RS

Experiment :- 8

Objective :- To implement the concept of Indexes and views.

Indexes :- An index is an ordered list of content of a column or group of columns in a table. An index created on the single column of the table is called simple index.

When multiple table columns are included in the index it is called composite index.

Creating an index for a table

Syntax (Simple)

CREATE INDEX index-name
ON tablename (column name);

Composite Index :-

CREATE INDEX index-name
ON tablename (columnname, columnname);

Creating an Unique/Non Unique Index :-

CREATE UNIQUE/NON UNIQUE INDEX indexfilename
ON tablename (columnname);

Dropping Indexes :-

An index can be dropped by using DROP INDEX.

SESSIONAL PAPERS

SESSIONAL PAPERS

Vishal
PRODUCTS
BHOPAL

Syntax :-

DROP INDEX indexfilename;

Views :-

Logical data is how we want to see the current data in our database physical data is how this data is actually placed in our database.

- (i) The DBA stores the views as a definition only.
Hence there is no duplication of data.
- (ii) Simplifies Queries.
- (iii) Can be queried as a base table itself.
- (iv) Provides data security.
- (v) Avoids data redundancy.

Creation of views :-

Syntax :-

CREATE VIEW Viewname AS
SELECT Columnname Columnname
FROM tablename
WHERE Columnname = expression list;

stream

Roll No..... Scholar No..... 24

Bankatulla

Renaming the
symbol:

RS

Rename the columns of a view :-

Syntax:-

CREATE VIEW viewname AS

SELECT newcolumnname

FROM tablename

WHERE columnname = expression-list;

SESSIONAL PAPERS

Roll No.

Scholar No.

26

Experiment :- 9

Objective :- To implement the basics of PL/SQL.

Introduction :- PL/SQL bridges the gap between database technology and procedural programming languages. It can be thought of as a development tool that extends the facilities of Oracle's SQL database language.

In PL/SQL you can insert, delete, update and retrieve table data as well as use procedural techniques such as writing loops or branching to another block of code.

PL/SQL Block Structure :-

DECLARE

Declaration of memory variables used later

BEGIN

SQL executable statements for manipulating table data

Exceptions

SQL and/or PL-SQL code to handle errors.

END;

Barkatullah University Institute of Technology, Bhopal

Roll No..... Scholar No..... 27

Displaying user messages on the screen :- Any programming tool requires a method through which messages can be displayed to the user.

dbms output :- Is a package that includes a number of procedures and functions that accumulate information in buffer so that it can be retrieved later.

Put line :- Put a piece of information in the buffer followed by a end of line marker. It can also be used to display message to the user.

Conditional control in PL/SQL

Syntax :-

```
IF <condition> Then  
    <Action>  
ELSE IF <condition>  
    <Action>  
ELSE  
    <Action>  
END IF ;
```

The for Loop statement

Syntax :-

SESSIONAL PAPERS

Roll No.....

Scholar No..... 28

For variable (in [reverse]) Start — end
Loop
(Action)
END LOOP ;

The GOTO statement :- The goto statement allows you to change the flow of control within a PL/SQL Block.

Q by

ans

J.

SESSIONAL PAPERS

Scholar No.

28

Barkatullah University Institute of Technology, Bhopal

Roll No.....

Scholar No.

28

Experiment :- 10

Objective :- To implement the concept of cursor and trigger.

Cursor :- We have seen how Oracle executes and SQL statement Oracle DBA uses a work area for its internal processing.

This work area is private to SQL operation and is called a cursor the data that is stored in the cursor is called the Active Data Set.

Explicit cursor :- You can explicitly declare a cursor to process the rows individually a cursor declared by the user is called Explicit cursor.

Why use an Explicit cursor :- cursor can be used when the user wants to process data one row at a time.

Explicit cursor management :- The steps involved in declaring a cursor and manipulating data in the active data set are,

SESSIONAL PAPERS

* Open the cursor

* fetch the data from the cursor one row at a time

* Close the cursor

Explicit cursor attributes :- Oracle provides certain attributes cursor variables to control the execution of the cursor. (explicit to implicit) is opened and used Oracle creates a set of fixed system variables via which Oracle keeps track of the current status of the cursor(s).

How to declare the cursor :-

The general syntax to create any particular cursor is as follows,

Cursor & Cursorname) is SQL Statement;

How to open the cursor :-

The general syntax to open any particular cursor is as follows.

(Open Cursorname)

Fetching a record from the cursor :- The fetch statement retrieves the rows from the active set to the variables one at a time. Each time a fetch is executed the focus of the DBA cursor advances to that next row in the active set.

One can make use of any the loop statements (loop - End loop along with while for) to fetch the records from the cursor into variable one row at a time.

Closing a cursor

The general syntax to close the cursor is as follows.

Close < cursorname >;

Database triggers:-

Database triggers are procedures that are stored in the database and are implicitly executed (fire) when the contents of table are changed.

Roll No.....

Scholar No..... 32

How to Delete a trigger :-

The Syntax for Deleting the Trigger is as follows.

Drop trigger < (triggername) ;