**Lab File**

**Database Management System**

**(CSE 201)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

A blue and yellow logo

Description automatically generated with low confidence

Submitted to: Submitted by:

Dr Richa Gupta Shaina Mehta

Ast. Professor A2305219268

CSE Department, ASET B.tech. C.S.E.

5CSE-4Y

Amity School Of Engineering and technology

Amity University Uttar Pradesh

Noida -201301

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exp No | Assignment  Category | Code | Name of Experiment | Date of Allotment | Date of Evaluation | Max  Marks | Marks  Obtained | Faculty  Sign |
| 1 | Mandatory  Experiment |  | Lab Assignment 1 | 29-07-2021 | 29-09-2021 |  |  |  |
| 2 |  | Lab Assignment 2 | 05-08-2021 | 29-09-2021 |  |  |  |
| 3 |  | Lab Assignment 3 | 12-08-2021 | 29-09-2021 |  |  |  |
| 4 |  | Lab Assignment 4 | 19-08-2021 | 29-09-2021 |  |  |  |
| 5 |  | Lab Assignment 5 | 26-08-2021 | 29-09-2021 |  |  |  |
| 6 |  | Lab Assignment 6 | 02-09-2021 | 29-09-2021 |  |  |  |
| 7 |  | Lab Assignment 7 | 09-09-2021 | 29-09-2021 |  |  |  |
| 8 |  |  | Lab Assignment 8 | 16-09-2021 | 29-09-2021 |  |  |  |
| 9 |  |  | Lab Assignment  9 | 06-10-2021 | 27-10-2021 |  |  |  |
| 10 |  |  | Lab Assignment  10 | 13-10-2021 | 27-10-2021 |  |  |  |
| 11 |  |  | Lab Assignment 11 | 20-10-2021 | 27-10-2021 |  |  |  |
| 12 |  |  | Lab Assignment 12 | 27-10-2021 | 10-11-2021 |  |  |  |
|  | Viva | Viva |  |  |  |  |  |  |

**Lab Assignment 1**

**Q1) Create the following tables:**

1. **client\_master**

**Table

Description automatically generated**

1. **Product\_master**

**Table

Description automatically generated with low confidence**

**Q2) Insert the following data into their respective tables:**

1. **Data for client\_master**

**Table

Description automatically generated**

1. **Data for Product\_master**

**Table

Description automatically generated**

**Q3) On the basis of above two tables answer the following Questionries:**

1. **Find out the names of all the clients.**
2. **Retrieve the list of names and cities of all the clients.**
3. **List the various products available from the product\_master table.**
4. **List all the clients who are located in Bombay.**
5. **Display the information for client no 0001 and 0002.**
6. **Find the products with description as ‘1.44 drive’ and ‘1.22 Drive’.**
7. **Find all the products whose sell price is greater then 5000.**
8. **Find the list of all clients who stay in in city ‘Bombay’ or city ‘Delhi’ or ‘Madras’.**
9. **Find the product whose selling price is greater than 2000 and less than or equal to 5000.**
10. **List the name, city and state of clients not in the state of ‘Maharashtra’.**

**Queries:**

CREATE TABLE client\_master(client\_no VARCHAR(6) PRIMARY KEY, name VARCHAR(20), address\_1 VARCHAR(30), address\_2 VARCHAR(30), city VARCHAR(15), pincode NUMERIC(6), state VARCHAR(15), bal\_due NUMERIC(10,2));

CREATE TABLE product\_master(product\_no VARCHAR(6) PRIMARY KEY, description VARCHAR(20), profit\_percent NUMERIC, unit\_measure VARCHAR(10), qty\_on\_hand NUMERIC, reorder\_lv1 NUMERIC, sell\_price NUMERIC, cost\_price NUMERIC);

insert into client\_master values('0001', 'Ivan', '', '', 'Bombay', 400054, 'Maharashtra', 1500);

insert into client\_master values('0002', 'Vandana', '', '', 'Madras', 780001, 'Tamilnadu', 0);

insert into client\_master values('0003', 'Pramada', '', '', 'Bombay', 400057, 'Maharashtra', 5000);

insert into client\_master values('0004', 'Basu', '', '', 'Bombay', 400056, 'Maharashtra', 0);

insert into client\_master values('0005', 'Ravi', '', '', 'Delhi', 100001, ' ' , 2000);

insert into client\_master values('0006', 'Rukmini', '', '', 'Bombay', 400050, 'Maharashtra', 0);

insert into product\_master values('P00001', '1.44floppies', 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', 20, 5, 1050, 500);

insert into product\_master values('P07865', '1.22 floppies', 5, 'piece', 100, 20, 525, 500);

insert into product\_master values('P07868', 'Keyboards', 2, 'piece', 10, 1500, 3150, 3050);

insert into product\_master values('P07885', 'CD Drive', 2.5, 'piece', 10, 3, 5250, 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);

select \* from client\_master;

select \* from product\_master;

select name from client\_master;

select name, city from client\_master;

select description from product\_master;

select name from client\_master where city='Bombay';

select \* from client\_master where client\_no='0001' OR client\_no='0002';

select \* from product\_master where description='1.44 Drive' OR description='1.22 Drive';

select \* from product\_master where sell\_price > 5000;

select \* from client\_master where city='Bombay' or city='Delhi' or city='Madras';

select \* from product\_master where sell\_price > 2000 and sell\_price <= 5000;

select name, city, state from client\_master where not state = 'Maharashtra';

**Output:**

Application, table, Excel

Description automatically generated

Chart

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated



Table

Description automatically generated

**Lab Assignment 2**

**Q2) Using the table client master and product master answer the following Questionnaires.**

1. **Change the selling price of ‘1.44 floppy drive to Rs.1150.00**
2. **Delete the record with client 0001 from the client master table.**
3. **Change the city of client\_no’0005’ to Bombay.**
4. **Change the bal\_due of client\_no ‘0001, to 1000.**
5. **Find the products whose selling price is more than 1500 and also find the new selling price as original selling price \*15.**
6. **Find out the clients who stay in a city whose second letter is a.**
7. **Find out the name of all clients having ‘a’ as the second letter in their names.**
8. **List the products in sorted order of their description.**
9. **Count the total number of orders**
10. **Calculate the average price of all the products.**
11. **Calculate the minimum price of products.**
12. **Determine the maximum and minimum prices . Rename the tittle as ‘max\_price’ and min\_price respectively.**
13. **Count the number of products having price greater than or equal to 1500.**

**Queries:**

update product\_master set sell\_price = 1150 where description = '1.44floppies';

select \* from product\_master;

delete from client\_master where client\_no = '0001';

select \* from client\_master;

update client\_master set city = 'Bombay' where client\_no = '0005';

select \* from client\_master;

insert into client\_master values('0001', 'Ivan', '', '', 'Bombay', 400054, 'Maharashtra', 1500);

select \* from client\_master order by client\_no;

update client\_master set bal\_due = 1000 where client\_no = '0001';

select \* from client\_master order by client\_no;

select \*, sell\_price\*15 as original\_price from product\_master where sell\_price > 1500;

select \* from client\_master where city LIKE '\_a%';

select name from client\_master where name LIKE '\_a%';

select \* from product\_master order by description;

select sum(qty\_on\_hand) from product\_master;

select avg(sell\_price), avg(cost\_price) from product\_master;

select min(sell\_price), min(cost\_price) from product\_master;

select min(sell\_price) as min\_price, max(sell\_price) from product\_master;

select count(\*) from product\_master where sell\_price > 1500;

**Output:**

Chart

Description automatically generated with low confidence

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Table, Excel

Description automatically generated

Table

Description automatically generated



Graphical user interface, table

Description automatically generated

Chart

Description automatically generated with low confidence











**Lab Assignment 3**

**Q1) Create the following tables:**

1. **Sales\_master**

**Table

Description automatically generated Text

Description automatically generated with medium confidence**

1. **Sales\_order**

**Table

Description automatically generated**

1. **Sales\_order\_details**

**Table

Description automatically generated**

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

1. **Data for sales\_man master table:**

**Table

Description automatically generated**

1. **Data for salesorder table:Table

   Description automatically generated**
2. **Data for sales\_order\_details table:**

**Table

Description automatically generated**

**Queries:**

create table Sales\_master(Salesman\_no VARCHAR(6) NOT NULL PRIMARY KEY CHECK(Salesman\_no LIKE 'S%'), Sal\_name VARCHAR(20) NOT NULL,

Address VARCHAR(30) NOT NULL,

City VARCHAR(20), Pincode NUMERIC(6), State VARCHAR(20),

Sal\_amt NUMERIC(8,2) NOT NULL CHECK(Sal\_amt > 0),

Tgt\_to\_get NUMERIC(6,2) NOT NULL CHECK(Tgt\_to\_get > 0),

Ytd\_sales NUMERIC(6,2) NOT NULL CHECK(Ytd\_sales > 0), Remarks Varchar(30));

Insert into Sales\_master values('S00001','Kiran','A/14 worli','Bombay',400002,'Mah',3000,100,50,'Good');

Insert into Sales\_master values('S00002','Manish','65,nariman','Bombay',400001,'Mah',3000,200,100,'Good');

Insert into Sales\_master values('S00003','Ravi','P-7 Bandra','Bombay',400032,'Mah',3000,200,100,'Good');

Insert into Sales\_master values('S00004','Ashish','A/5 Juhu','Bombay',400044,'Mah',3000,200,150,'Good');

SELECT \* FROM Sales\_master;

create table Sales\_order(S\_order\_no VARCHAR(6) NOT NULL PRIMARY KEY CHECK(S\_order\_no like 'O%'), S\_order\_date DATE,

Client\_no VARCHAR(6) REFERENCES client\_master(client\_no), Dely\_add VARCHAR(6),

salesman\_no VARCHAR(6) REFERENCES Sales\_master(Salesman\_no),

Dely\_type CHAR(1) DEFAULT 'f' CHECK(Dely\_type = 'f' OR Dely\_type = 'p'),

Billed\_yn CHAR(1), Dely\_date DATE,

Order\_status VARCHAR(10), CHECK(Order\_status IN ('in process','fulfilled','back order','canceled')),

CONSTRAINT Dely\_date CHECK(Dely\_date >= S\_order\_date));

insert into Sales\_order values('O19001','1996-01-12','0001','NULL','S00001','f','n','1996-01-20','in process');

insert into Sales\_order values('O19002','1996-01-25','0002','NULL','S00002','p','n','1996-01-27','canceled');

insert into Sales\_order values('O16865','1996-02-18','0003','NULL','S00003','f','y','1996-02-20','fulfilled');

insert into Sales\_order values('O19003','1996-04-03','0001','NULL','S00001','f','y','1996-04-07','fulfilled');

insert into Sales\_order values('O46866','1996-05-20','0004','NULL','S00002','p','n','1996-05-22','canceled');

insert into Sales\_order values('O10008','1996-05-24','0005','NULL','S00004','f','n','1996-05-26','in process');

select \* from Sales\_order;

create table Sales\_order\_details(s\_order\_no VARCHAR(6) REFERENCES Sales\_order(S\_order\_no),

Product\_no VARCHAR(6) REFERENCES product\_master(product\_no),

Qty\_order NUMERIC(8), Qty\_disp NUMERIC(8), Product\_rate NUMERIC(10,2), PRIMARY KEY(S\_order\_no,Product\_no));

INSERT into Sales\_order\_details values('O19001','P00001',4,4,525);

INSERT into Sales\_order\_details values('O19001','P07965',2,1,8400);

INSERT into Sales\_order\_details values('O19001','P07885',2,1,5250);

INSERT into Sales\_order\_details values('O19002','P00001',10,0,525);

INSERT into Sales\_order\_details values('O16865','P07868',3,3,3150);

INSERT into Sales\_order\_details values('O16865','P07885',10,10,5250);

INSERT into Sales\_order\_details values('O19003','P00001',4,4,1050);

INSERT into Sales\_order\_details values('O19003','P03453',2,2,1050);

INSERT into Sales\_order\_details values('O46866','P06734',1,1,12000);

INSERT into Sales\_order\_details values('O46866','P07965',1,0,8400);

INSERT into Sales\_order\_details values('O10008','P07975',1,0,1050);

INSERT into Sales\_order\_details values('O10008','P00001',10,5,525);

select \* from Sales\_order\_details;

**Output:**

Table

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

**Lab Assignment 4**

**Q1) Create the following tables:**

1. **Challan Header**

**Chart

Description automatically generated with low confidence**

1. **Table Name: Challan\_Details**

**Text

Description automatically generated**

**Q2) Insert the following values into the challan header and challan\_details tables:**

1. **Data for Challan Header table**

**Text

Description automatically generated**

1. **Data for challan\_details table**

**Text

Description automatically generated with medium confidence**

**Q3) Answer the following Questionries:**

1. **Make the primary key to client\_no in client\_master.**
2. **Add a new column phone\_no in the client\_master table.**
3. **Add the not null constraint in the product\_master table with the columns description, profit percent, sell price and cost price.**
4. **Change the size of client\_no field in the client\_master table.**
5. **Select product\_no, description where profit percent is between 20 and 30 both inclusive**

**Queries:**

create table Challan\_Header(Challan\_no varchar(6) primary key, s\_order\_no varchar(6) references Sales\_order(S\_order\_no),

challan\_date date not NULL, billed\_yn char(1) default 'N' check(billed\_yn = 'Y' or billed\_yn = 'N'));

insert into Challan\_Header values('CH9001','O19001','1995-12-12','Y');

insert into Challan\_Header values('CH6865','O46866','1995-11-12','Y');

insert into Challan\_Header values('CH3965','O10008','1995-10-12','Y');

select \* from Challan\_Header;

create table Challan\_Details(Challan\_no VARCHAR(6) references Challan\_Header(Challan\_no),

Product\_no varchar(6), constraint FK\_product\_cd\_prodmaster foreign key(Product\_no) references product\_master(product\_no),

Qty\_disp NUMERIC(4,2) not NULL);

insert into Challan\_Details values('CH9001','P00001',4);

insert into Challan\_Details values('CH9001','P07965',1);

insert into Challan\_Details values('CH9001','P07885',1);

insert into Challan\_Details values('CH6865','P07868',3);

insert into Challan\_Details values('CH6865','P03453',4);

insert into Challan\_Details values('CH6865','P00001',10);

insert into Challan\_Details values('CH3965','P00001',5);

insert into Challan\_Details values('CH3965','P07975',2);

Select \* from Challan\_Details;

--Creating The Database

create database DBMS\_Lab2;

-- To Changes the database context to the specified database or database snapshot in SQL Server

use DBMS\_Lab2;

--Creating The Tables And Inserting Values In It

--Creating Tables

CREATE TABLE client\_master(client\_no VARCHAR(6), name VARCHAR(20), address\_1 VARCHAR(30), address\_2 VARCHAR(30),

city VARCHAR(15), pincode NUMERIC(6), state VARCHAR(15), bal\_due NUMERIC(10,2));

CREATE TABLE product\_master(product\_no VARCHAR(6), description VARCHAR(20), profit\_percent NUMERIC, unit\_measure VARCHAR(10),

qty\_on\_hand NUMERIC, reorder\_lv1 NUMERIC, sell\_price NUMERIC, cost\_price NUMERIC);

--Inserting Values In The Table

insert into client\_master values('0001', 'Ivan', 'n/a', 'n/a', 'Bombay', 400054, 'Maharashtra', 1500);

insert into client\_master values('0002', 'Vandana', 'n/a', 'n/a', 'Madras', 780001, 'Tamilnadu', 0);

insert into client\_master values('0003', 'Pramada', 'n/a', 'n/a', 'Bombay', 400057, 'Maharashtra', 5000);

insert into client\_master values('0004', 'Basu', 'n/a', 'n/a', 'Bombay', 400056, 'Maharashtra', 0);

insert into client\_master values('0005', 'Ravi', 'n/a', 'n/a', 'Delhi', 100001, 'n/a' , 2000);

insert into client\_master values('0006', 'Rukmini', 'n/a', 'n/a', 'Bombay', 400050, 'Maharashtra', 0);

insert into product\_master values('P00001', '1.44floppies', 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', 20, 5, 1050, 500);

insert into product\_master values('P07865', '1.22 floppies', 5, 'piece', 100, 20, 525, 500);

insert into product\_master values('P07868', 'Keyboards', 2, 'piece', 10, 1500, 3150, 3050);

insert into product\_master values('P07885', 'CD Drive', 2.5, 'piece', 10, 3, 5250, 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);

--Displaying The Tables

select \* from client\_master;

select \* from product\_master;

--Query 1

--Step 1

ALTER TABLE client\_master ALTER COLUMN client\_no varchar(6) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN name VARCHAR(20) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN address\_1 VARCHAR(30) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN address\_2 VARCHAR(30) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN city VARCHAR(15) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN pincode NUMERIC(6) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN state VARCHAR(15) NOT NULL;

ALTER TABLE client\_master ALTER COLUMN bal\_due NUMERIC(10,2) NOT NULL;

--Step 2

alter table client\_master add constraint client\_master\_pk primary key (client\_no);

--Query 2

alter table client\_master add phone\_number numeric(10);

--Query 3

alter table product\_master alter column description VARCHAR(20) NOT NULL;

alter table product\_master alter column profit\_percent NUMERIC not null;

alter table product\_master alter column sell\_price NUMERIC not null;

alter table product\_master alter column cost\_price NUMERIC not null;

--Query 4

ALTER TABLE dbo.client\_master DROP CONSTRAINT client\_master\_pk;

ALTER TABLE client\_master alter column client\_no varchar(10) NOT NULL;

alter table client\_master add constraint client\_master\_pk primary key (client\_no);

--Query 5

select product\_no, description from product\_master where profit\_percent >= 20 and profit\_percent <= 30;

--Dropping The Tables

IF OBJECT\_ID ('dbo.client\_master','U') IS NOT NULL

DROP TABLE client\_master;

IF OBJECT\_ID ('dbo.product\_master','U') IS NOT NULL

DROP TABLE product\_master;

SET ansi\_warnings OFF

GO

--Dropping The Database

Drop DBMS\_Lab2;

**Output:**

Table

Description automatically generated

Table

Description automatically generated

A close-up of a graph

Description automatically generated with low confidence

Chart

Description automatically generated with low confidence



**Lab Assignment 5**

**Q1) Answer the following Queries:**

1. **Find out the product which has been sold to ‘Ivan Sayross.’**
2. **Find out the product and their quantities that will have do delivered.**
3. **Find the product\_no and description of moving products.**
4. **Find out the names of clients who have purchased ‘CD DRIVE’**
5. **List the product\_no and s\_order\_no of customers haaving qty ordered less than 5 from the order details table for the product “1.44 floppies”.**
6. **Find the products and their quantities for the orders placed by ‘Vandan Saitwal ’and “Ivan Bayross”.**
7. **Find the products and their quantities for the orders placed by client\_no “C00001” and “C00002”.**
8. **Find the order No, Client No and salesman No. where a client has been received by more than one salesman.**
9. **Display the s\_order\_date in the format “dd-mm-yy” e.g. “12- feb-96”**
10. **Find the date, 15 days after date.**

**Queries:**

--Displaying The Tables for the review befre starting the assignment

select \* from client\_master;

select \* from product\_master;

select \* from Sales\_master;

select \* from Sales\_order;

select \* from Sales\_order\_details;

select \* from Challan\_Header;

select \* from Challan\_Details;

--Queries

select product\_master.product\_no, product\_master.description from Sales\_order\_details, product\_master, client\_master, Sales\_order where product\_master.product\_no = Sales\_order\_details.product\_no and Sales\_order.S\_order\_no = Sales\_order\_details.S\_order\_no and client\_master.client\_no = Sales\_order.client\_no and client\_master.name = 'Ivan';

select product\_master.product\_no, Sales\_order\_details.Qty\_order from Sales\_order\_details, product\_master, Sales\_order where Sales\_order.Order\_status = 'fulfilled' AND product\_master.product\_no = Sales\_order\_details.product\_no and Sales\_order.S\_order\_no = Sales\_order\_details.S\_order\_no;

select product\_master.product\_no, product\_master.description from product\_master, Sales\_order\_details, Sales\_order where Sales\_order.Order\_status = 'in process' and product\_master.product\_no=Sales\_order\_details.product\_no and Sales\_order.S\_order\_no = Sales\_order\_details.S\_order\_no;

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.S\_order\_no = Sales\_order\_details.s\_order\_no and client\_master.client\_no = Sales\_order.client\_no and product\_master.description = 'CD Drive' order by Sales\_order.client\_no;

select Sales\_order\_details.product\_no, Sales\_order\_details.S\_order\_no from product\_master, Sales\_order\_details where product\_master.product\_no = Sales\_order\_details.product\_no and Sales\_order\_details.Qty\_order < 5 and product\_master.description = '1.44floppies';

select client\_master.name, product\_master.product\_no, product\_master.description, sum(Sales\_order\_details.Qty\_order) 'Quantities' from product\_master, client\_master, Sales\_order, Sales\_order\_details where product\_master.product\_no = Sales\_order\_details.product\_no and client\_master.client\_no = Sales\_order.client\_no and Sales\_order\_details.S\_order\_no = Sales\_order.S\_order\_no and (client\_master.name = 'Ivan' or client\_master.name = 'Vandana') group by client\_master.name, product\_master.product\_no, product\_master.description;

Select client\_master.client\_no, product\_master.product\_no, product\_master.description, sum(Sales\_order\_details.Qty\_order) 'Quantities' from product\_master, client\_master, Sales\_order, Sales\_order\_details where product\_master.product\_no = Sales\_order\_details.product\_no and client\_master.client\_no = Sales\_order.client\_no and Sales\_order\_details.S\_order\_no = Sales\_order.S\_order\_no and (client\_master.client\_no = '0001' or client\_master.client\_no = '0002') group by client\_master.client\_no, product\_master.product\_no, product\_master.description;

select client\_master.client\_no, Sales\_master.Salesman\_no, Sales\_order.S\_order\_no from client\_master, Sales\_master, Sales\_order where client\_master.client\_no = Sales\_order.client\_no and Sales\_master.Salesman\_no = Sales\_order.Salesman\_no order by client\_master.client\_no;

select S\_order\_no, client\_no, Salesman\_no from Sales\_order where 1 < (select count(\*) from Sales\_order where client\_no = Sales\_order.client\_no and 1 = (select count(Sales\_order.Salesman\_no) from Sales\_order));

select format(Sales\_order.S\_order\_date,'D','en-gb') as S\_order\_date from Sales\_order;

select S\_order\_date from Sales\_order where S\_order\_date >= dateadd(day,15,'1996-01-12');

**Output:**

Application, table, Excel

Description automatically generated

Table

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated



Table

Description automatically generated

Table

Description automatically generated

**Lab Assignment 6**

**Q1) Answer the following Queries:**

1. **Print the description and total quantity sold for each product.**
2. **Find the value of each product sold.**
3. **Calculate the average quantity sold for each client that has a maximum order value of 15000.**
4. **Find out the products which has been sold to Ivan.**
5. **Find the names of clients who have ‘CD Drive’.**
6. **Find the products and their quantities for the orders placed by ‘Vandana’ and ‘Ivan’.**
7. **Select product\_no, total qty\_ordered for each product.**
8. **Select product\_no, product description and qty ordered for each product.**
9. **Display the order number and day on which clients placed their order.**
10. **Display the month and Date when the order must be delivered.**

**Queries:**

--Displaying the tables

select \* from client\_master;

select \* from product\_master;

select \* from Sales\_master;

select \* from Sales\_order;

select \* from Sales\_order\_details;

select \* from Challan\_Header;

select \* from Challan\_Details;

--Executing the queries

select description, Sum(Qty\_disp) from product\_master, Sales\_order\_details where product\_master.product\_no = Sales\_order\_details.product\_no group by description;

select Sales\_order\_details.Product\_no, product\_master.description from Sales\_order\_details, product\_master where product\_master.product\_no =Sales\_order\_details.Product\_no group by Sales\_order\_details.Product\_no, product\_master.description, Sales\_order\_details.Product\_rate;

select client\_master.client\_no, avg(Sales\_order\_details.Qty\_disp) as AvgSales from Sales\_order\_details, client\_master, Sales\_order where client\_master.client\_no = Sales\_order.Client\_no and Sales\_order.S\_order\_no = Sales\_order\_details.s\_order\_no group by client\_master.client\_no having max(Sales\_order\_details.Qty\_order \*Sales\_order\_details.Product\_rate) > 1500;

select client\_master.client\_no, product\_master.description from client\_master, product\_master, Sales\_order\_details, Sales\_order where client\_master.client\_no = Sales\_order.Client\_no and Sales\_order.s\_order\_no = Sales\_order\_details.s\_order\_no and client\_master.name = 'Ivan' group by client\_master.client\_no, product\_master.description;

select client\_master.name, product\_master.description from Sales\_order\_details, client\_master, Sales\_order, product\_master where client\_master.client\_no = sales\_order.client\_no and Sales\_order.S\_order\_no = Sales\_order\_details.s\_order\_no and product\_master.description = 'CD Drive' group by client\_master.name, product\_master.description;

select client\_master.name, product\_master.description, sum(Sales\_order\_details.Qty\_order) as Units from Sales\_order\_details, Sales\_order, client\_master, product\_master where client\_master.client\_no = Sales\_order.Client\_no and Sales\_order.S\_order\_no = Sales\_order\_details.s\_order\_no and product\_master.product\_no = Sales\_order\_details.Product\_no group by client\_master.name, product\_master.description having (client\_master.name = 'Ivan' or client\_master.name = 'Vandana');

select product\_master.product\_no, sum(Sales\_order\_details.Qty\_order) as Units from Sales\_order\_details, client\_master, product\_master, Sales\_order where Sales\_order.S\_order\_no = Sales\_order\_details.s\_order\_no and product\_master.product\_no = Sales\_order\_details.Product\_no group by product\_master.product\_no;

select product\_master.product\_no, product\_master.description, sum(Sales\_order\_details.Qty\_order) as Units from Sales\_order\_details, client\_master, Sales\_order, product\_master where Sales\_order.S\_order\_no = Sales\_order\_details.s\_order\_no and product\_master.product\_no = Sales\_order\_details.Product\_no group by product\_master.product\_no, product\_master.description;

select S\_order\_no, datename(weekday,S\_order\_date) from Sales\_order;

select S\_order\_no, datepart(d,S\_order\_date) as 'Delivery Date', datename(month,S\_order\_date) as 'Delivery Month' from Sales\_order;

**Output:**

Application, table, Excel

Description automatically generated

Table

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

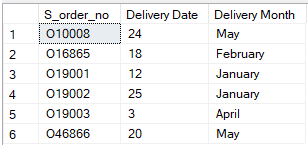
Description automatically generated

Table

Description automatically generated

Table

Description automatically generated



**Lab Assignment 7**

**Q1) Answer the following Queries:**

1. **Find the product\_no and description of non- moving products.**
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 “1.44 Drive” is ordered by only client and print the client\_no name to whom it was sold.**
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 all the clients and the salesman in the city of Bombay.**
8. **Select salesman name in “Bombay” who has atleast one client located at “Bombay”.**
9. **Select the product\_no, description, qty\_on-hand,cost\_price of non\_moving items in the product\_master table.**

**Queries:**

--Query 1

select product\_no, description from product\_master where product\_no in (select product\_no from Sales\_order\_details where S\_order\_no in( select S\_order\_no from Sales\_order where Order\_status = 'canceled'));

--Query 2

select client\_no, name, address\_1, address\_2, city, pincode from client\_master where client\_no in (select Client\_no from Sales\_order where S\_order\_no = 'O19001');

--Query 3

select name from client\_master where client\_no in (select Client\_no from Sales\_order where S\_order\_date < '1996-05-01');

--Query 4

select client\_no, name from client\_master where client\_no in (select client\_no from Sales\_order where S\_order\_no in (select S\_order\_no from Sales\_order\_details where Product\_no in (select Product\_no from product\_master where description = '1.44 Drive')));

--Query 5

select name from client\_master where client\_no in (select client\_no from Sales\_order where S\_order\_no in (select S\_order\_no from Sales\_order\_details where Product\_rate >= 10000.00));

--Query 6

select S\_order\_no, Client\_no from Sales\_order where Client\_no in (select client\_no from client\_master where name = 'Rahul Desai');

--Query 7 - Need some time to think till then I will execute Query 8, 9, and 10

--Query 8

select distinct name from client\_master where city = 'Bombay' union select distinct Sal\_name from Sales\_master where City = 'Bombay';

--Query 9

select Sal\_name from Sales\_master where Salesman\_no in (select salesman\_no from Sales\_order where Client\_no in (select client\_no from client\_master where city = 'Bombay'));

--Query 10

select product\_no, description, qty\_on\_hand, cost\_price from product\_master where product\_no not in (select product\_no from Sales\_order\_details);

**Output:**

Table

Description automatically generated



Table

Description automatically generated







Table

Description automatically generated

Graphical user interface, table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**Lab Assignment 8**

**Q1) Answer the following Questions:**

1. **Create an index on the table client\_master, field client\_no.**
2. **Create an index on the sales\_order, field s\_order\_no.**
3. **Create an composite index on the sales\_order\_details table for the columns s\_order\_no and product\_no.**
4. **Create an composite index ch\_index on challan\_header table for the columns challan no and s\_order\_no.**
5. **Create an uniQuestion index on the table salesman\_master, field salesman\_no.**
6. **Drop index ch\_index on table challan\_header.**
7. **Create view on salesman\_master whose sal\_amt is less than 3500.**
8. **Create a view client\_view on client\_master and rename the columns as name, add1, add2, city, pcode, state respectively.**
9. **Select the client names from client\_view who lives in city ‘Bombay’.**
10. **Drop the view client\_view.**

**Queries:**

**Creating the Database**

CREATE DATABASE DBMS\_Lab3;

use DBMS\_Lab3;

**Creating the Tables**

CREATE TABLE client\_master(client\_no VARCHAR(6) PRIMARY KEY, name VARCHAR(20), address\_1 VARCHAR(30), address\_2 VARCHAR(30), city VARCHAR(15), pincode NUMERIC(6), state VARCHAR(15), bal\_due NUMERIC(10,2));

CREATE TABLE product\_master(product\_no VARCHAR(6) PRIMARY KEY, description VARCHAR(20), profit\_percent NUMERIC, unit\_measure VARCHAR(10), qty\_on\_hand NUMERIC, reorder\_lv1 NUMERIC, sell\_price NUMERIC, cost\_price NUMERIC);

insert into client\_master values('0001', 'Ivan', '', '', 'Bombay', 400054, 'Maharashtra', 1500);

insert into client\_master values('0002', 'Vandana', '', '', 'Madras', 780001, 'Tamilnadu', 0);

insert into client\_master values('0003', 'Pramada', '', '', 'Bombay', 400057, 'Maharashtra', 5000);

insert into client\_master values('0004', 'Basu', '', '', 'Bombay', 400056, 'Maharashtra', 0);

insert into client\_master values('0005', 'Ravi', '', '', 'Delhi', 100001, ' ' , 2000);

insert into client\_master values('0006', 'Rukmini', '', '', 'Bombay', 400050, 'Maharashtra', 0);

insert into product\_master values('P00001', '1.44floppies', 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', 20, 5, 1050, 500);

insert into product\_master values('P07865', '1.22 floppies', 5, 'piece', 100, 20, 525, 500);

insert into product\_master values('P07868', 'Keyboards', 2, 'piece', 10, 1500, 3150, 3050);

insert into product\_master values('P07885', 'CD Drive', 2.5, 'piece', 10, 3, 5250, 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);

create table Sales\_master(Salesman\_no VARCHAR(6) NOT NULL PRIMARY KEY CHECK(Salesman\_no LIKE 'S%'), Sal\_name VARCHAR(20) NOT NULL,

Address VARCHAR(30) NOT NULL,

City VARCHAR(20), Pincode NUMERIC(6), State VARCHAR(20),

Sal\_amt NUMERIC(8,2) NOT NULL CHECK(Sal\_amt > 0),

Tgt\_to\_get NUMERIC(6,2) NOT NULL CHECK(Tgt\_to\_get > 0),

Ytd\_sales NUMERIC(6,2) NOT NULL CHECK(Ytd\_sales > 0), Remarks Varchar(30));

Insert into Sales\_master values('S00001','Kiran','A/14 worli','Bombay',400002,'Mah',3000,100,50,'Good');

Insert into Sales\_master values('S00002','Manish','65,nariman','Bombay',400001,'Mah',3000,200,100,'Good');

Insert into Sales\_master values('S00003','Ravi','P-7 Bandra','Bombay',400032,'Mah',3000,200,100,'Good');

Insert into Sales\_master values('S00004','Ashish','A/5 Juhu','Bombay',400044,'Mah',3000,200,150,'Good');

create table Sales\_order(S\_order\_no VARCHAR(6) NOT NULL PRIMARY KEY CHECK(S\_order\_no like 'O%'), S\_order\_date DATE,

Client\_no VARCHAR(6) REFERENCES client\_master(client\_no), Dely\_add VARCHAR(6),

salesman\_no VARCHAR(6) REFERENCES Sales\_master(Salesman\_no),

Dely\_type CHAR(1) DEFAULT 'f' CHECK(Dely\_type = 'f' OR Dely\_type = 'p'),

Billed\_yn CHAR(1), Dely\_date DATE,

Order\_status VARCHAR(10), CHECK(Order\_status IN ('in process','fulfilled','back order','canceled')),

CONSTRAINT Dely\_date CHECK(Dely\_date >= S\_order\_date));

insert into Sales\_order values('O19001','1996-01-12','0001','NULL','S00001','f','n','1996-01-20','in process');

insert into Sales\_order values('O19002','1996-01-25','0002','NULL','S00002','p','n','1996-01-27','canceled');

insert into Sales\_order values('O16865','1996-02-18','0003','NULL','S00003','f','y','1996-02-20','fulfilled');

insert into Sales\_order values('O19003','1996-04-03','0001','NULL','S00001','f','y','1996-04-07','fulfilled');

insert into Sales\_order values('O46866','1996-05-20','0004','NULL','S00002','p','n','1996-05-22','canceled');

insert into Sales\_order values('O10008','1996-05-24','0005','NULL','S00004','f','n','1996-05-26','in process');

create table Sales\_order\_details(s\_order\_no VARCHAR(6) REFERENCES Sales\_order(S\_order\_no),

Product\_no VARCHAR(6) REFERENCES product\_master(product\_no),

Qty\_order NUMERIC(8), Qty\_disp NUMERIC(8), Product\_rate NUMERIC(10,2), PRIMARY KEY(S\_order\_no,Product\_no));

INSERT into Sales\_order\_details values('O19001','P00001',4,4,525);

INSERT into Sales\_order\_details values('O19001','P07965',2,1,8400);

INSERT into Sales\_order\_details values('O19001','P07885',2,1,5250);

INSERT into Sales\_order\_details values('O19002','P00001',10,0,525);

INSERT into Sales\_order\_details values('O16865','P07868',3,3,3150);

INSERT into Sales\_order\_details values('O16865','P07885',10,10,5250);

INSERT into Sales\_order\_details values('O19003','P00001',4,4,1050);

INSERT into Sales\_order\_details values('O19003','P03453',2,2,1050);

INSERT into Sales\_order\_details values('O46866','P06734',1,1,12000);

INSERT into Sales\_order\_details values('O46866','P07965',1,0,8400);

INSERT into Sales\_order\_details values('O10008','P07975',1,0,1050);

INSERT into Sales\_order\_details values('O10008','P00001',10,5,525);

create table Challan\_Header(Challan\_no varchar(6) primary key, s\_order\_no varchar(6) references Sales\_order(S\_order\_no),

challan\_date date not NULL, billed\_yn char(1) default 'N' check(billed\_yn = 'Y' or billed\_yn = 'N'));

insert into Challan\_Header values('CH9001','O19001','1995-12-12','Y');

insert into Challan\_Header values('CH6865','O46866','1995-11-12','Y');

insert into Challan\_Header values('CH3965','O10008','1995-10-12','Y');

create table Challan\_Details(Challan\_no VARCHAR(6) references Challan\_Header(Challan\_no),

Product\_no varchar(6), constraint FK\_product\_cd\_prodmaster foreign key(Product\_no) references product\_master(product\_no),

Qty\_disp NUMERIC(4,2) not NULL);

insert into Challan\_Details values('CH9001','P00001',4);

insert into Challan\_Details values('CH9001','P07965',1);

insert into Challan\_Details values('CH9001','P07885',1);

insert into Challan\_Details values('CH6865','P07868',3);

insert into Challan\_Details values('CH6865','P03453',4);

insert into Challan\_Details values('CH6865','P00001',10);

insert into Challan\_Details values('CH3965','P00001',5);

insert into Challan\_Details values('CH3965','P07975',2);

--Displaying Tables

select \* from client\_master;

select \* from product\_master;

select \* from Sales\_master;

select \* from Sales\_order;

select \* from Sales\_order\_details;

select \* from Challan\_Header;

select \* from Challan\_Details;

**Executing Queries**

--Queries

--Query 1 to 6

--Query 1

drop index if exists [index\_cm] on [client\_master];

go

create index index\_cm on client\_master(client\_no);

EXEC sp\_helpindex 'dbo.client\_master';

GO

--Query 2

drop index if exists [index\_so] on [Sales\_order];

go

create index index\_so on Sales\_order(S\_order\_no);

EXEC sp\_helpindex 'dbo.Sales\_order';

GO

--Query 3

drop index if exists [index\_sod] on [Sales\_order\_details];

go

create index index\_sod on Sales\_order\_details(S\_order\_no,Product\_no);

EXEC sp\_helpindex 'dbo.Sales\_order\_details';

GO

--Query 4

drop index if exists [ch\_index] on [Challan\_Header];

go

create index ch\_index on Challan\_Header(Challan\_no,s\_order\_no);

EXEC sp\_helpindex 'dbo.Challan\_Header';

GO

--Query 5

drop index if exists [index\_sm] on [Sales\_master];

go

create unique index index\_sm on Sales\_master(Salesman\_no);

EXEC sp\_helpindex 'dbo.Sales\_master';

GO

--Query 6

drop index if exists [ch\_index] on [Challan\_Header];

go

EXEC sp\_helpindex 'dbo.Challan\_Header';

GO

--Query 7

--Query 7 Part 1

--DROP view if exists dbo.sales\_master\_view;

--go

create view Sales\_master\_view as select \* from Sales\_master where Sal\_amt < 3500;

--Query 7 Part 2

select \* from Sales\_master\_view;

--Query 8 to 10

--Query 8 Part 1

create view client\_view as select name as 'Name', address\_1 as 'Address\_1', address\_2 as'Address\_2', city as 'City', pincode as 'Pincode', state as 'State' from client\_master;

--Query 8 Part 2

select \* from client\_view;

--Query 9

select Name from client\_view where City = 'Bombay';

--Query 10

DROP view if exists dbo.client\_view;

Go

**Output**

**Queries 1 to 6**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated



**Query 7**

Table

Description automatically generated

**Table

Description automatically generated**

**Query 8 to 10**

A picture containing table

Description automatically generated

Table

Description automatically generated

Graphical user interface, table

Description automatically generated

Table

Description automatically generated

**Lab Assignment 9**

**Q) Perform the following Functions:**

1. SELECT Char(65), ASCII('A') AS A;

Graphical user interface, application

Description automatically generated

1. SELECT CHAR(122),ASCII('z') AS z;

Graphical user interface, application

Description automatically generated

1. SELECT "Testing the upper function." AS String,UPPER("Testing the upper function.") AS UpperCase;

Graphical user interface, application

Description automatically generated

1. SELECT "TESTING THE LOWER FUNCTION." AS String, LOWER("TESTING THE LOWER FUNCTION.") AS LowerCase;

Graphical user interface, application

Description automatically generated

1. SELECT "Testing the upper function." AS Initial,

REPLACE(

"Testing the upper function.",

"upper",

"replace"

) AS Final;

Graphical user interface, application

Description automatically generated

1. SELECT CURRENT\_DATE() AS DATE;

Graphical user interface, application

Description automatically generated

1. SELECT EXTRACT(DAY FROM "2021-09-29");

Graphical user interface, application

Description automatically generated

1. SELECT SYSDATE();

Graphical user interface, application

Description automatically generated

1. SELECT -32,ABS(-32) AS Absolute\_Value;

Graphical user interface, application

Description automatically generated with medium confidence

1. SELECT 2.43,FLOOR(2.43) AS Floor\_Value;

Graphical user interface, application

Description automatically generated

1. SELECT "1,2,3",GREATEST(1, 2, 3) AS Greatest\_Value;

Graphical user interface, application

Description automatically generated

1. SELECT "1,2,3",LEAST(1, 2, 3) AS Least\_Value;

Table

Description automatically generated

1. SELECT "This is a string",LENGTH("This is a string") AS Length;

Graphical user interface, application

Description automatically generated

1. SELECT "25" AS Number,SQRT(25) AS Square\_Root;

Table

Description automatically generated with low confidence

1. SELECT "5^4" AS Equation,POWER(5, 4) AS Result;

Graphical user interface, application

Description automatically generated

1. SELECT "4.56" AS Number,ROUND(4.56) AS RoundOff;

Graphical user interface, application

Description automatically generated

1. SELECT "4.46" AS Number,ROUND(4.46) AS RoundOff;

Graphical user interface, application

Description automatically generated

1. SELECT SIN(Pi() / 2);

Graphical user interface, application

Description automatically generated

1. SELECT COS(Pi());

Graphical user interface, application

Description automatically generated

1. SELECT TAN(Pi() / 4);

Graphical user interface, application

Description automatically generated

1. SELECT "-40" AS Number, SIGN(-40) AS Sign;

Graphical user interface, application

Description automatically generated with medium confidence

1. SELECT "40" AS Number, SIGN(40) AS Sign;

Graphical user interface, application

Description automatically generated

1. SELECT LN(10);

Graphical user interface, application

Description automatically generated

1. SELECT LOG(10, 10);

Graphical user interface, application

Description automatically generated

1. SELECT MOD(19, 7);

Graphical user interface, application

Description automatically generated

1. SELECT EXP(2);

Graphical user interface, application

Description automatically generated

1. SELECT "tttesting trimttt" AS String,

TRIM('t' FROM "tttesting trimttt") AS Trimmed

Graphical user interface, application

Description automatically generated

**Lab Assignment 10**

**Creating Tables:**

CREATE Table Students(

StudNr NUMERIC(5) PRIMARY KEY,

Name VARCHAR(20),

Semester NUMERIC

);

INSERT INTO Students VALUES (24002, 'Xenokrates', 18);

INSERT INTO Students VALUES (25403, 'Jonas', 12);

INSERT INTO Students VALUES (26120, 'Fichte', 10);

INSERT INTO Students VALUES (26830, 'Aristoxenos', 8);

INSERT INTO Students VALUES (27550, 'Schopenhauer', 6);

INSERT INTO Students VALUES (28106, 'Carnap', 3);

INSERT INTO Students VALUES (29120, 'Theophrastos', 2);

INSERT INTO Students VALUES (29555, 'Feuerbach', 2);

A picture containing table

Description automatically generated

CREATE Table Attend

(StudNr NUMERIC(5),

LectureNr NUMERIC(4));

INSERT INTO Attend VALUES (26120, 5001);

INSERT INTO Attend VALUES (27550, 5001);

INSERT INTO Attend VALUES (27550, 4052);

INSERT INTO Attend VALUES (28106, 5041);

INSERT INTO Attend VALUES (28106, 5052);

INSERT INTO Attend VALUES (28106, 5216);

INSERT INTO Attend VALUES (28106, 5259);

INSERT INTO Attend VALUES (29120, 5001);

INSERT INTO Attend VALUES (29120, 5041);

INSERT INTO Attend VALUES (29120, 5049);

INSERT INTO Attend VALUES (25403, 5022);

INSERT INTO Attend VALUES (29555, 5022);

INSERT INTO Attend VALUES (29555, 5001);

Table

Description automatically generatedA picture containing text, scoreboard

Description automatically generated

CREATE Table Lectures(

LectureNr NUMERIC(4) PRIMARY KEY,

Title VARCHAR(20)

);

INSERT INTO Lectures VALUES(5001, 'Grundzuge');

INSERT INTO Lectures VALUES(5041, 'Ethik');

INSERT INTO Lectures VALUES(5043, 'Erkenntnistheorie');

INSERT INTO Lectures VALUES(5049, 'Maeutik');

INSERT INTO Lectures VALUES(4052, 'Logik');

INSERT INTO Lectures VALUES(5052, 'Wissenschaftstheorie');

INSERT INTO Lectures VALUES(5216, 'Bioethik');

INSERT INTO Lectures VALUES(5259, 'Der Wiener Kreis');

INSERT INTO Lectures VALUES(5022, 'Glaube und Wissen');

INSERT INTO Lectures VALUES(4630, 'Die 3 Kritiken');

A screenshot of a computer

Description automatically generated with low confidence

**Queries:**

**All students with their name and number of lectures**

SELECT s.name, Count(a.LectureNr) AS 'Number of lectures' FROM Attend a, Students s

WHERE a.StudNr = s.StudNr GROUP BY s.name;

Table

Description automatically generated with medium confidence

**Average semester of all students**

SELECT AVG(Semester) As 'Average semester' FROM Students;

Graphical user interface, application

Description automatically generated

**Average number of lectures per student**

SELECT @total: = COUNT(LectureNr) FROM Attend;

SELECT s.name, Count(a.LectureNr) / @total AS 'Average Lectures'

FROM Attend a, Students s

WHERE a.StudNr = s.StudNr

GROUP BY s.name;

**Students who attend lectures which also Fiche attends**

SELECT s.name FROM Students s

WHERE StudNr IN (SELECT StudNr FROM Attend WHERE LectureNr IN

(SELECT LectureNr FROM Attend WHERE StudNr IN

(SELECT StudNr FROM Students WHERE name = 'Fichte')));

Graphical user interface

Description automatically generated

**Titles of lectures which are attended by students from the first 4 semesters**

SELECT Title FROM Lectures WHERE LectureNr IN

(SELECT LectureNr FROM Attend WHERE StudNr IN

(SELECT StudNr FROM Students WHERE Semester <= 4));

Graphical user interface, table

Description automatically generated

**Lab Assignment 11**

**Q1) Write a PL/SQL program to arrange the number of two variable in such a way that the small number will store in num\_small variable and large number will store in num\_large variable.**

**Code:**

DECLARE

num\_small NUMBER;

num\_large NUMBER;

num\_temp NUMBER;

BEGIN

num\_small := :num\_small;

num\_large := :num\_large;

IF num\_small > num\_large

THEN

num\_temp := num\_small;

num\_small := num\_large;

num\_large := num\_temp;

END IF;

DBMS\_OUTPUT.PUT\_LINE('num\_small = '||num\_small);

DBMS\_OUTPUT.PUT\_LINE('num\_large = '||num\_large);

END;

**Output:**

**Graphical user interface, application

Description automatically generated**

**Text

Description automatically generated**

**Q2) Write a PL/SQL procedure to calculate the incentive on a target achieved and display the message either the record updated or not.**

**Code:**

DECLARE

PROCEDURE test1 (

sal\_achieve NUMBER,

target\_qty NUMBER,

emp\_id NUMBER

)

IS

incentive NUMBER := 0;

updated VARCHAR2(3) := 'No';

BEGIN

IF sal\_achieve > (target\_qty + 200) THEN

incentive := (sal\_achieve - target\_qty)/4;

UPDATE emp

SET salary = salary + incentive

WHERE employee\_id = emp\_id;

updated := 'Yes';

END IF;

DBMS\_OUTPUT.PUT\_LINE (

'Table updated? ' || updated || ', ' ||

'incentive = ' || incentive || '.'

);

END test1;

BEGIN

test1(2300, 2000, 144);

test1(3600, 3000, 145);

END;

**Output:**

**Graphical user interface, text

Description automatically generated**

**Q3) Write a PL/SQL program to check whether a number is even or odd.**

**Code:**

DECLARE

num NUMBER;

BEGIN

num := :num;

IF (MOD(num,2) = 0)

THEN

DBMS\_OUTPUT.PUT\_LINE('even number');

ELSE

DBMS\_OUTPUT.PUT\_LINE('odd number');

END IF;

END;

**Output:**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**Q4) Write a PL/SQL procedure to calculate the incentive on a specific target otherwise a general incentive to be paid using IF-THEN-ELSE.**

**Code:**

DECLARE

PROCEDURE test1 (

sal\_achieve NUMBER,

target\_qty NUMBER,

emp\_id NUMBER

)

IS

incentive NUMBER := 0;

BEGIN

IF sal\_achieve > (target\_qty + 200) THEN

incentive := (sal\_achieve - target\_qty)/4;

ELSE

incentive :=75;

END IF;

DBMS\_OUTPUT.PUT\_LINE ('incentive = ' || incentive);

UPDATE emp

SET salary = salary + incentive

WHERE employee\_id = emp\_id;

END test1;

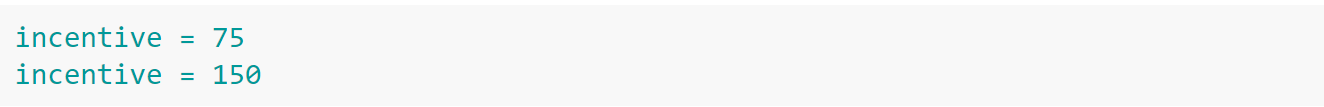
BEGIN

test1(2300, 2000, 144);

test1(3600, 3000, 145);

END;

**Output:**

****

**Q5) Write a PL/SQL program to check whether a date falls on weekend i.e. SATURDAY.**

**Code:**

DECLARE

dat DATE;

d CHAR(3);

BEGIN

dat := DATE '2021-10-24';

d := TO\_CHAR(dat, 'DY');

IF ( d = 'SAT' OR d = 'SUN')

THEN

DBMS\_OUTPUT.PUT\_LINE('weekend');

ELSE

DBMS\_OUTPUT.PUT\_LINE('weekday');

END IF;

END;

**Output:**

**Text

Description automatically generated with medium confidence**

**Q6) Write a PL/SQL procedure to calculate incentive achieved according to the specific sale limit.**

**Code:**

DECLARE

PROCEDURE test1 (sal\_achieve NUMBER)

IS

incentive NUMBER := 0;

BEGIN

IF sal\_achieve > 44000 THEN

incentive := 1800;

ELSIF sal\_achieve > 32000 THEN

incentive := 800;

ELSE

incentive := 500;

END IF;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE (

'Sale achieved : ' || sal\_achieve || ', incentive : ' || incentive || '.'

);

END test1;

BEGIN

test1(45000);

test1(36000);

test1(28000);

END;

**Output:**

**A picture containing background pattern

Description automatically generated**

**Q7) Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.**

**Code:**

SET SERVEROUTPUT ON

DECLARE

tot\_emp NUMBER;

BEGIN

SELECT Count(\*)

INTO tot\_emp

FROM employees e

join departments d

ON e.department\_id = d.department\_id

WHERE e.department\_id = 50;

dbms\_output.Put\_line ('The employees are in the department 50: '

||To\_char(tot\_emp));

IF tot\_emp >= 45 THEN

dbms\_output.Put\_line ('There are no vacancies in the department 50.');

ELSE

dbms\_output.Put\_line ('There are some vacancies in department 50.');

END IF;

END;

**Output:**

****

**Lab Assignment 12**

**Questions:**

**19. Write a program in PL/SQL to update the salary of a specific employee by 8% if the salary exceeds the mid range of the salary against this job and update up to mid range if the salary is less than the mid range of the salary, and display a suitable message.**

**20. Write a program in PL/SQL using nested loop with EXIT WHEN statement.**

**21. Write a program in PL/SQL using loop with CONTINUE statement.**

**22. Write a program in PL/SQL using loop with CONTINUE WHEN statement.**

**23. Write a program in PL/SQL to print 1st n numbers.**

**Code:**

**Q23)**

DECLARE

num NUMBER := :n;

itr NUMBER;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('The first '||num||' numbers are: ');

for itr in 1..num loop

DBMS\_OUTPUT.PUT(itr||' ');

END LOOP;

dbms\_output.new\_line;

END;

**Q21)**

DECLARE

num NUMBER := 0;

BEGIN

LOOP

DBMS\_OUTPUT.PUT\_LINE('Inside the loop: n = ' || TO\_CHAR(num));

num := num + 1;

IF(num < 5)

THEN

CONTINUE;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Inside the loop, after CONTINUE: n = ' || TO\_CHAR(num));

EXIT WHEN num = 7;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('When out from the loop: n = ' || TO\_CHAR(num));

END;

**Q20)**

DECLARE

m NUMBER := 0;

BEGIN

LOOP

m := m + 1;

DBMS\_OUTPUT.PUT\_LINE('The value of the m is: '||m);

EXIT WHEN m > 4;

END LOOP;

END;

**Q22)**

DECLARE

n NUMBER := 0;

BEGIN

LOOP -- After CONTINUE statement, control resumes here

DBMS\_OUTPUT.PUT\_LINE ('The value inside the loop: n = ' || TO\_CHAR(n));

n := n + 1;

CONTINUE WHEN n < 4;

DBMS\_OUTPUT.PUT\_LINE

('The value inside loop, after CONTINUE: n = ' || TO\_CHAR(n));

EXIT WHEN n = 6;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('The value after exit from the loop: n = ' || TO\_CHAR(n));

END;

**Q19)**

DECLARE

emp\_min\_salary NUMBER(6,0);

emp\_max\_salary NUMBER(6,0);

emp\_mid\_salary NUMBER(6,2);

tmp\_salary EMPLOYEES.SALARY%TYPE;

tmp\_emp\_id EMPLOYEES.EMPLOYEE\_ID%TYPE := 167;

tmp\_emp\_name EMPLOYEES.FIRST\_NAME%TYPE;

BEGIN

SELECT min\_salary,

max\_salary

INTO emp\_min\_salary,

emp\_max\_salary

FROM JOBS

WHERE JOB\_ID = (SELECT JOB\_ID

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = tmp\_emp\_id);

-- calculate mid-range

emp\_mid\_salary := (emp\_min\_salary + emp\_max\_salary) / 2;

-- get salary of the given employee

SELECT salary,first\_name

INTO tmp\_salary,tmp\_emp\_name

FROM employees

WHERE employee\_id = tmp\_emp\_id;

-- update salary

IF tmp\_salary < emp\_mid\_salary THEN

UPDATE employees

SET salary = emp\_mid\_salary

WHERE employee\_id = tmp\_emp\_id;

ELSE

UPDATE employees

SET salary = salary + salary \* 8 /100

WHERE employee\_id = tmp\_emp\_id;

END IF;

--display message

IF tmp\_salary > emp\_mid\_salary THEN

DBMS\_OUTPUT.PUT\_LINE('The employee '||tmp\_emp\_name||' ID ' || TO\_CHAR(tmp\_emp\_id) ||

' works in salary ' || TO\_CHAR(tmp\_salary) ||

' which is higher than mid-range of salary ' || TO\_CHAR(emp\_mid\_salary));

ELSIF tmp\_salary < emp\_mid\_salary THEN

DBMS\_OUTPUT.PUT\_LINE('The employee '||tmp\_emp\_name||' ID ' || TO\_CHAR(tmp\_emp\_id) ||

' works in salary ' || TO\_CHAR(tmp\_salary) ||

' which is lower than mid-range of salary ' || TO\_CHAR(emp\_mid\_salary));

ELSE

DBMS\_OUTPUT.PUT\_LINE('The employee '||tmp\_emp\_name||' ID ' || TO\_CHAR(tmp\_emp\_id) ||

' works in salary ' || TO\_CHAR(tmp\_salary) ||

' which is equal to the mid-range of salary ' || TO\_CHAR(emp\_mid\_salary));

END IF;

END;

**Output:**

Q23)

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Q20)

Text

Description automatically generated

Q21)

A picture containing calendar

Description automatically generated

Q22)

Text

Description automatically generated

Q19)

