Naman Mittal
A2305219718
5CSE-12X

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

DATABASE MANAGEMENT SYSTEMS

Submitted by:
Naman Mittal
A2305219718
B.Tech 5CSE-12X

Submitted to:
Dr. Rashmi Bhel

Naman Mittal
A2305219718
5CSE 12X

INDEX

S. No.	Lab Activity Name	Lab Date	Evaluation Date	Remarks
1.	AIM: Introduction to SQL and DDL Commands.	21.07.2020		
2.	Exercise on Data Creation	28.07.2021		
3.	Simple Data Retrieval Queries	04.08.2021		
4.	Simple Data Retrieval Queries	11.08.2021		
5.	Queries on Natural Join	18.08.2021		
6.	Simple Data Retrieval Queries	25.08.2021		
7.	Queries on View	01.09.2021		
8.	Practice Question	08.09.2021		
9.	Queries on Triggers	22.09.2021		
10.	Index Creation	06.10.2021		
11.	PL/SQL Queries	20.10.2021		

Naman Mittal
A2305219718
5CSE 12X

Experiment-1

Introduction to SQL and DDL Commands.

SQL

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

MySQL is a very popular open-source relational database management system (RDBMS).

SQL is the standard language for dealing with Relational Databases. SQL is used to insert, search, update, and delete database records.

DDL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.

DDL- DDL stands for Data Definition Language**.** DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc. All the command of DDL are auto committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL-

a. CREATE: It is used to create a new table in the database.
b. DROP: It is used to delete both the structure and record stored in the table.
c. ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.
d. TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.
e. COMMENT: add comments to the data dictionary
f. RENAME: rename an object

EXPERIMENT -2

Exercise on Data Creation

The table given in the book by " Ivon Bayross"

CLIENT TABLES

Client Master Table

CREATE :

create table client_master(client_no varchar(10), name varchar(30), address1 varchar(40), address2 varchar(40), city varchar(40), pincode int, state varchar(40), bal_due float);

INSERT:

insert into client_master values("C00001",'Basu Navindgi','shop_no_1','Badi Gali','Bombay',400054,'Maharashtra',15000);

insert into client_master values("C00002",'Vandana Saitwal','shop_no_2','Cafe Road','Madras',780001,'Tamil Nadu',0);

insert into client_master values("C00003"," Pramada Jaguste,'shop_no_34','Bullock Road','Bombay',400057,'Maharashtra',5000);

insert into client_master values("C00004"," Basu Navindgi,'shop_no_33','Old Road','Bombay',400056,'Maharashtra',0);

insert into client_master values("C00005"," Rukmini,'shop_no_28','café Road','Bombay',400058,'Maharashtra',0);

insert into client_master values("C00006", Ravi Sreedharan','shop_no_23','Old Road','Janakpuri',110058,'Delhi',2000);

insert into client_master values("C00007"," Satyendra Jain','shop_no_288','Mandir Road','Madras',780001,'Tamil Nadu',0);

insert into client_master values("C00008"," Pramod Kumar','shop_no_2888','Mandir Wali Road','Bombay',400057,'Maharashtra',5000);

insert into client_master values("C00009",' Basu Sharma','shop_no_287','Nai Road','Bombay',400056,'Maharashtra',0);

insert into client_master values("C00010",' Ravi Mittal','shop_no_296','Nai Road','Tilak Nagar',110058,'Delhi',2000);

SHOW:

Select * from client_master;

client_no	name	address1	address2	city	pincode	state	bal_due
C00001	Ivan Bayross	shop_no_1	Badi Gali	Bombay	400054	Maharashtra	15000.00
C00002	Vandana Saitwal	shop_no_2	Cafe Road	Madras	780001	Tamil Nadu	0.00
C00003	Pramada Jaguste	shop_no_34	Bullock Road	Bombay	400057	Maharashtra	5000.00
C00004	Basu Navindgi	shop_no_33	Old Road	Bombay	400056	Maharashhtra	0.00
C00005	Rukmini	shop_no_28	Cafe Road	Bombay	400058	Maharashtra	0.00
C00006	Ravi Sreedharan	shop_no_23	Old Road	Janakpuri	110058	Delhi	2000.00
C00007	Satyendra Jain	shop_no_288	Mandir Road	Madras	780001	Tamil Nadu	0.00
C00008	Pramod Kumar	shop_no_2888	Mandir Wali Road	Bombay	400057	Maharashtra	5000.00
C00009	Basu Sharma	shop_no_287	Nai Road	Bombay	400056	Maharashhtra	0.00
C00010	Ravi Mittal	shop_no_296	Nai Road	Tilak Nagar	110058	Delhi	2000.00

Product Master Table

CREATE:

create table Product_master (product_no varchar(10),description varchar(60), profit_percent float, unitmeasure int, qtyonhand int, reorderlvl int ,sellprice float,costprice float);

INSERT :

insert into product_master values('P00001','1.44 Floppies',5,'Piece',100,20,525,500);

insert into product_master values('P03453','Monitors',6,'Piece',10,3,12000,11280);

insert into product_master values('P06734','Mouse',5,'Piece',20,5,1050,1000);

insert into product_master values('P07865','1.22 Floppies',5,'Piece',100,20,525,500);

insert into product_master values('P07868','Keyboards',2,'Piece',10,3,3150,3050);

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);

SHOW :

select * from product_master;

product_no	description	profit_percent	unitmeasure	qtyonhand	reorderlvl	sellprice	costprice
P00001	1.44 Floppies	5.00	Piece	100	20	525.00	500.00
P03453	Monitors	6.00	Piece	10	3	12000.00	11280.00
P06734	Mouse	5.00	Piece	20	5	1050.00	1000.00
P07865	1.22 Floppies	5.00	Piece	100	20	525.00	500.00
P07868	Keyboards	2.00	Piece	10	3	3150.00	3050.00
P07885	CD Drive	2.50	Piece	10	3	5250.00	5100.00
P07965	540 HDD	4.00	Piece	10	3	8400.00	8000.00
P07975	1.44 Drive	5.00	Piece	10	3	1050.00	1000.00
P08865	1.22 Drive	5.00	Piece	2	3	1050.00	1000.00

Salesman Master Table

CREATE :

```
create table salesman_master (salesman_no int,salesman_name varchar(30), address1 varchar(50), address2 varchar(50), city (40), pincode int, state varchar(30), salamt int,tgttoget int , ytdsales  varchar(30),remarks varchar(100));
```

INSERT:

```
insert into salesman_master values('S00001','Kiran','A/14','Worli','Bombay',400002,'MAH',3000,100,50,'Good');

insert into salesman_master values('S00002','Manish','65','Nariman','Bombay',400001,'MAH',3000,200,100,'Good');

insert into salesman_master values('S00003','Ravi','P-7','Bandra','Bombay',400032,'MAH',3000,200,100,'Good');

insert into salesman_master values('S00004','Ashish','A/5','Juhu','Bombay',400044,'MAH',3000,200,150,'Good');

insert into salesman_master values('S00005','Aman','A/6','Panvel','Bombay',400044,'MAH',3000,400,150,'Good');

insert into salesman_master values('S00006','Vaibhav','C/5','Tilak Nagar','Delhi',110058,'DEL',5000,700,250,'Good');

insert into salesman_master values('S00007','Harman','H1/55','New Road','Bengaluru',700048,'KAR',8000,700,180,'Good');

insert into salesman_master values('S00008','Geeta','T/512','Juhu','Bombay',400044,'MAH',3000,200,150,'Good');
```

insert into salesman_master
values('S00009','Harsh','A/52','Vishakaptnam','Bombay',400026,'MAH',3000,200,150,'Good');

insert into salesman_master
values('S00010','Manan','A/12','Juhu','Bombay',400044,'MAH',3000,200,150,'Good');

SHOW :

select * from salesman_master;

salesman_no	salesman_name	address1	address2	city	pincode	state	salamt	tgttoget	ytdsales	remarks
S00001	Kiran	A/14	Worli	Bombay	400002	MAH	3000.00	100.00	50.00	Good
S00002	Manish	65	Nariman	Bombay	400001	MAH	3000.00	200.00	100.00	Good
S00003	Ravi	P-7	Bandra	Bombay	400032	MAH	3000.00	200.00	100.00	Good
S00004	Ashish	A/5	Juhu	Bombay	400044	MAH	3000.00	200.00	150.00	Good
S00005	Aman	A/6	Panvel	Bombay	400044	MAH	3000.00	400.00	150.00	Good
S00006	Vaibhav	C/5	Tilak Nagar	Delhi	110058	DEL	5000.00	700.00	250.00	Good
S00007	Harman	H1/55	New Road	Bengaluru	700048	KAR	8000.00	700.00	180.00	Good
S00008	Geeta	T/512	Juhu	Bombay	400044	MAH	3000.00	200.00	150.00	Good
S00009	Harsh	A/52	Vishakaptnam	Bombay	400026	MAH	3000.00	200.00	150.00	Good
S00010	Manan	A/12	Juhu	Bombay	400044	MAH	3000.00	200.00	150.00	Good

Sales Order Table

CREATE :

```
create table Sales_order (order_no varchar(6),order_date varchar(10),client_no varchar(6),dely_addr varchar(30), salesman_no varchar(6),dely_type varchar(10), billed_yn varchar(1), dely_date varchar(10), order_status varchar(10));
```

INSERT :

```
insert into Sales_order values('A101','12/6/2021','B101','A/30 Laxmi Nagar N.D','Q101','Rapid','Y','14/6/2021','Delivered');

select * from Sales_order;

insert into Sales_order values('A102','13/6/2021','B102','H/30 Tilak Nagar N.D','Q102','Rapid','Y','19/6/2021','Delivered');

insert into Sales_order values('A103','14/6/2021','B102','H/37 Tilak Nagar N.D','Q103','Rapid','Y','19/6/2021','Delivered');

insert into Sales_order values('A104','16/6/2021','B103','G1/657 Uttam Nagar N.D','Q104','Rapid','Y','19/6/2021','Delivered');

insert into Sales_order values('A105','17/6/2021','B104','U/336 Kalkaji N.D','Q105','Rapid','Y','19/6/2021','Delivered');

insert into Sales_order values('A106','18/6/2021','B105','N-778 Chirag Delhi N.D','Q106','Rapid','N','24/6/2021','Delivered');

insert into Sales_order values('A107','19/6/2021','B104','U/985 Kalkaji N.D','Q107','Rapid','Y','24/6/2021','Delivered');

insert into Sales_order values('A108','19/6/2021','B105','N-1002 Chirag Delhi N.D','Q108','Rapid','N','24/6/2021','Delivered');

insert into Sales_order values('A109','20/6/2021','B109','S no 2 Vikaspuri N.D','Q109','Rapid','Y','29/6/2021','Delivered');

insert into Sales_order values('A110','21/6/2021','B108','Wz-23 Shakti Nagar N.D','Q112','Rapid','N','30/6/2021','Delivered');
```

SHOW :

select * from Sales_order;

order_no	order_date	client_no	dely_addr	salesman_no	dely_type	billed_yn	dely_date	order_status
A101	12/6/2021	B101	A/30 Laxmi Nagar N.D	Q101	Rapid	Y	14/6/2021	Delivered
A102	13/6/2021	B102	H/30 Tilak Nagar N.D	Q102	Rapid	Y	19/6/2021	Delivered
A103	14/6/2021	B102	H/37 Tilak Nagar N.D	Q103	Rapid	Y	19/6/2021	Delivered
A104	16/6/2021	B103	G1/657 Uttam Nagar N.D	Q104	Rapid	Y	19/6/2021	Delivered
A105	17/6/2021	B104	U/336 Kalkaji N.D	Q105	Rapid	Y	19/6/2021	Delivered
A106	18/6/2021	B105	N-778 Chirag Delhi N.D	Q106	Rapid	N	24/6/2021	Delivered
A107	19/6/2021	B104	U/985 Kalkaji N.D	Q107	Rapid	Y	24/6/2021	Delivered
A108	19/6/2021	B105	N-1002 Chirag Delhi N.D	Q108	Rapid	N	24/6/2021	Delivered
A109	20/6/2021	B109	S no 2 Vikaspuri N.D	Q109	Rapid	Y	29/6/2021	Delivered
A110	21/6/2021	B108	Wz-23 Shakti Nagar N.D	Q112	Rapid	N	30/6/2021	Delivered

Sales Order Details Table

CREATE :

```
create table sales_order_details(s_order_no varchar(6) references
sales_order(s_order_no), product_no varchar(6),qty_ordered numeric(8),qty_disp
numeric(8),product_rate numeric(10,2));
```

INSERT :

```
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('O46865','P07868',3,3,3150);

insert into sales_order_details values('O46865','P07885',3,1,5250);

insert into sales_order_details values('O46865','P00001',10,10,525);

insert into sales_order_details values('O46865','P03453',4,4,1050);

insert into sales_order_details values('O19003','P03453',2,2,1050);

insert into sales_order_details values('O19003','P06734',1,1,12000);

insert into sales_order_details values('O46866','P07965',1,0,8400);

insert into sales_order_details values('O46866','P07975',1,0,1050);

insert into sales_order_details values('O10008','P00001',10,5,525);

insert into sales_order_details values('O10008','P07975',5,3,1050);
```

SHOW :

```
select * from sales_order_details;
```

s_order_no	product_no	qty_ordered	qty_disp	product_rate
O19001	P00001	4	4	525.00
O19001	P07965	2	1	8400.00
O19001	P07885	2	1	5250.00
O19002	P00001	10	0	525.00
O46865	P07868	3	3	3150.00
O46865	P07885	3	1	5250.00
O46865	P00001	10	10	525.00
O46865	P03453	4	4	1050.00
O19003	P03453	2	2	1050.00
O19003	P06734	1	1	12000.00
O46866	P07965	1	0	8400.00
O46866	P07975	1	0	1050.00
O10008	P00001	10	5	525.00
O10008	P07975	5	3	1050.00

LIBRARY TABLES

IT BOOKS TABLE

CREATE :

```
create table IT_book (IT_book_code varchar(10), IT_book_title varchar(30),
IT_book_author varchar(30),  IT_book_publ varchar(30),IT_book_price
numeric(4), IT_book_eddi numeric(4));
```

INSERT :

```
insert into IT_book values('B1001','C++ Fundamentals','John Shaikh','Registered
Publishers',250,2009);

insert into IT_book values('B1023','Java Fundamentals','John Shaikh','Registered
Publishers',800,2018);

insert into IT_book values('B1085','Python Fundamentals','John
Shaikh','Registered Publishers',2500,2017);

insert into IT_book values('B1014','ML Basics to Pro','Aman Dhattarwal','Aman
Publishers',856,2020);

insert into IT_book values('B1096','GO Fundamentals','Daksh Sharma','Saraswati
Publishers',900,2021);

insert into IT_book values('B1025','Internet all about','Karan Singh','Expert
Publishers',450,2015);

insert into IT_book values('B1125','CSS and JavaScript','Ben Stokes','Ben And Ben
Publishers',1500,2017);

insert into IT_book values('B1085','History of Internet','Chandler Hallow','MrBeast
Publishers',7780,2004);

insert into IT_book values('B1963','C++ Zero to hero','Karl Pound','Top
Publishers',985,2000);

insert into IT_book values('B1029','Django Basics to Expert','Atul Bhatia','Amethi
Publishers',634,2003);
```

SHOW :

select * from IT_book;

IT_book_code	IT_book_title	IT_book_author	IT_book_publ	IT_book_price	IT_book_eddi
B1001	C++ Fundamentals	John Shaikh	Registered Publishers	250	2009
B1023	Java Fundamentals	John Shaikh	Registered Publishers	800	2018
B1085	Python Fundamentals	John Shaikh	Registered Publishers	2500	2017
B1014	ML Basics to Pro	Aman Dhattarwal	Aman Publishers	856	2020
B1096	GO Fundamentals	Daksh Sharma	Saraswati Publishers	900	2021
B1025	Internet all about	Karan Singh	Expert Publishers	450	2015
B1125	CSS and JavaScript	Ben Stokes	Ben And Ben Publishers	1500	2017
B1085	History of Internet	Chandler Hallow	MrBeast Publishers	7780	2004
B1963	C++ Zero to hero	Karl Pound	Top Publishers	985	2000
B1029	Django Basics to Expert	Atul Bhatia	Amethi Publishers	634	2003

IT STUD TABLE

CREATE:

create table IT_stud (IT_stud_code varchar(10), IT_stud_name varchar(30), IT_stud_branch varchar(10), IT_stud_sem numeric(2), IT_stud_card_no numeric(10), IT_stud_card_type varchar(10));

INSERT :

insert into IT_stud values('A2301','Naman','CSE',5,2301,'Regular');

insert into IT_stud values('A2322','Vaibhav','ECE',5,2302,'Regular');

insert into IT_stud values('A2373','Ajay','IT',5,2303,'Regular');

insert into IT_stud values('A2394','Pranjul','AI',5,2304,'Regular');

insert into IT_stud values('A2335','Seema','CSE',5,2305,'Regular');

insert into IT_stud values('A2706','Naresh','AI',5,2306,'Regular');

insert into IT_stud values('A2907','Manju','ME',5,2307,'Regular');

insert into IT_stud values('A2508','Ankita','ECE',5,2308,'Regular');

insert into IT_stud values('A2809','Ridhi','IT',5,2309,'Regular');

insert into IT_stud values('A2710','Ritika','CSE',5,2310,'Regular');

SHOW :

select * from IT_stud;

IT_stud_code	IT_stud_name	IT_stud_branch	IT_stud_sem	IT_stud_card_no	IT_stud_card_type
A2301	Naman	CSE	5	2301	Regular
A2322	Vaibhav	ECE	5	2302	Regular
A2373	Ajay	IT	5	2303	Regular
A2394	Pranjul	AI	5	2304	Regular
A2335	Seema	CSE	5	2305	Regular
A2706	Naresh	AI	5	2306	Regular
A2907	Manju	ME	5	2307	Regular
A2508	Ankita	ECE	5	2308	Regular
A2809	Ridhi	IT	5	2309	Regular
A2710	Ritika	CSE	5	2310	Regular

IT ISSUE RETURN TABLE

CREATE:

create table IT_issu_retu (IT_issu_doi varchar(10), IT_issu_a_dor varchar(10), IT_issu_fine numeric(4), IT_book_code varchar(10),IT_stud_code varchar(10));

INSERT :

insert into IT_issu_retu values('01/06/2021','01/07/2021',100,'B2525','A5263');

insert into IT_issu_retu values('04/06/2021','07/07/2021',140,'B2874','A5274');

insert into IT_issu_retu values('08/06/2021','08/07/2021',140,'B9658','A5296');

insert into IT_issu_retu values('19/06/2021','01/07/2021',80,'B0202','A5020');

insert into IT_issu_retu values('12/06/2021','12/07/2021',100,'B0002','A5000');

insert into IT_issu_retu values('10/06/2021','10/07/2021',100,'B1052','A7855');

insert into IT_issu_retu values('28/06/2021','21/07/2021',90,'B2855','A9856');

insert into IT_issu_retu values('30/06/2021','19/07/2021',60,'B2585','A8578');

insert into IT_issu_retu values('26/06/2021','10/07/2021',40,'B2725','A8745');

insert into IT_issu_retu values('01/07/2021','10/07/2021',0,'B3225','A9632');

SHOW :

select * from IT_issu_retu;

IT_issu_doi	IT_issu_a_dor	IT_issu_fine	IT_book_code	IT_stud_code
01/06/2021	01/07/2021	100	B2525	A5263
04/06/2021	07/07/2021	140	B2874	A5274
08/06/2021	08/07/2021	140	B9658	A5296
19/06/2021	01/07/2021	80	B0202	A5020
12/06/2021	12/07/2021	100	B0002	A5000
10/06/2021	10/07/2021	100	B1052	A7855
28/06/2021	21/07/2021	90	B2855	A9856
30/06/2021	19/07/2021	60	B2585	A8578
26/06/2021	10/07/2021	40	B2725	A8745
01/07/2021	10/07/2021	0	B3225	A9632

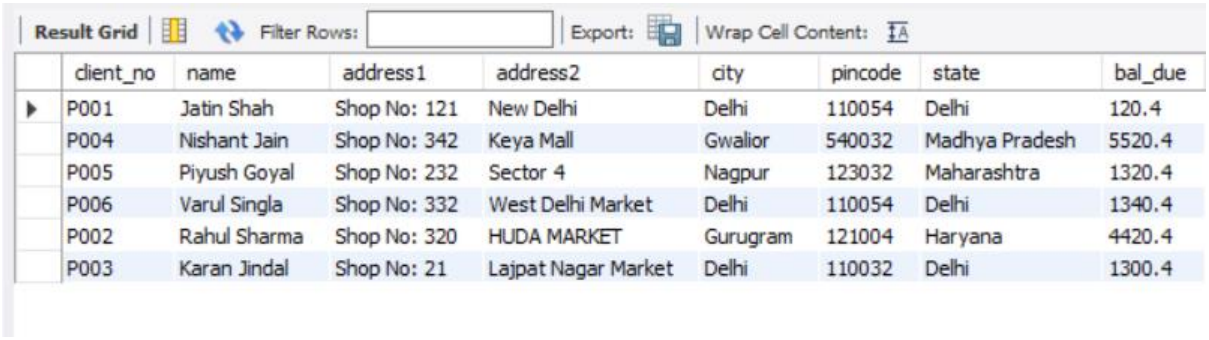
Experiment-3

Simple Data Retrieval

Queries on tables given in book by IVON BAYROSS

a) Display all the fields from "client_master"

select * from sales_order;

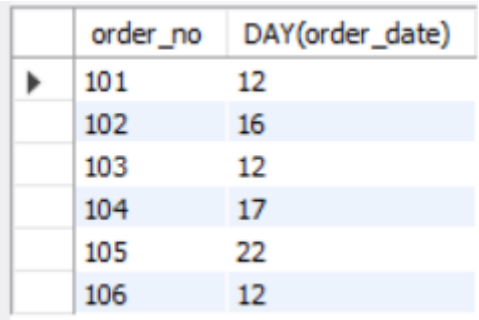


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

client_no	name	address1	address2	city	pincode	state	bal_due
P001	Jatin Shah	Shop No: 121	New Delhi	Delhi	110054	Delhi	120.4
P004	Nishant Jain	Shop No: 342	Keya Mall	Gwalior	540032	Madhya Pradesh	5520.4
P005	Piyush Goyal	Shop No: 232	Sector 4	Nagpur	123032	Maharashtra	1320.4
P006	Varul Singla	Shop No: 332	West Delhi Market	Delhi	110054	Delhi	1340.4
P002	Rahul Sharma	Shop No: 320	HUDA MARKET	Gurugram	121004	Haryana	4420.4
P003	Karan Jindal	Shop No: 21	Lajpat Nagar Market	Delhi	110032	Delhi	1300.4

b) Display the order_no and day on which the client placed their orders

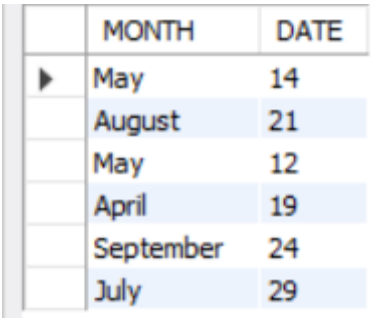
select order_no, DAY(order_date) from sales_order;



order_no	DAY(order_date)
101	12
102	16
103	12
104	17
105	22
106	12

c) Display the month and date when the order must be delivered

select DATE_FORMAT(dely_date,'%M') as MONTH, DAY(dely_date) as DATE from sales_order;



MONTH	DATE
May	14
August	21
May	12
April	19
September	24
July	29

d) Display the order date in format of DD-MM-YY

select DATE_FORMAT(dely_date,'%d-%m-%y') as DATE from sales_order;

DATE
14-05-21
21-08-20
12-05-20
19-04-20
24-09-20
29-07-21

e) Find the number of days after today's date

select datediff(sysdate(),order_date)as date_diffrence from sales_order;

date_diffrence
84
353
449
474
316
23

f) Find the no. of days elapsed between today's date and the delivery date of the order placed by the clients

select datediff(sysdate(),dely_date)as del_date_diffrence from sales_order;

del_date_diffrence
82
348
449
472
314
6

QUERIES ON LIBRARY TABLE

a) Display all the fields from the table "book"

select * from it_book;

IT_book_code	IT_book_title	IT_book_author	IT_book_publ	IT_book_price	IT_book_eddi
B101	the tales of XYZ	XYZ	ABC House	400	1
B102	A new start	CBD	ABC House	550	2
B103	Annexure by hkx	HKX	ABC House	600	1
B104	Resumed in life	ADC	SQL House	450	3
B105	Hello to my old friend	PRD	WAS House	300	5
B106	The death of hercules	FAR	ASW House	250	2
B107	The dream catcher	GFH	ABC House	340	3
B108	The eye of the tiger	DSE	SQL House	550	4

b) Display the name of the students and date on which they have issued the books;

select it_stud.it_stud_name as Name, it_issu_doi from it_issu_retu inner join it_stud on it_issu_retu.it_stud_code = it_stud.it_stud_code;

Name	it_issu_doi
Rohan	2020-09-23
Utkarsh	2020-08-12
Josh	2020-07-03
Rohit	2020-06-13
Yaman	2020-11-01
Josh	2020-09-13
Rohan	2020-10-09
Utkarsh	2020-09-03
Rohan	2020-09-23
Rohan	2020-09-23

c) Display the book title,author and publisher in the order of book title.

select it_book_title as "BOOK TITLE", it_book_author as "AUTHOR", it_book_pub as "PUBLISHER" from it_book order by it_book_title;

BOOK TITLE	AUTHOR	PUBLISHER
A new start	CBD	ABC House
Annexure by hkx	HKX	ABC House
Hello to my old friend	PRD	WAS House
Resumed in life	ADC	SQL House
The death of hercules	FAR	ASW House
The dream catcher	GFH	ABC House
The eye of the tiger	DSE	SQL House
the tales of XYZ	XYZ	ABC House

d) Display the date of issue and the actual date of return in the format “DD-MM-YY.

select date_format(it_issu_doi,'%d-%m-%y') as "ISSUE DATE",date_format(it_issu_a_dor,'%d-%m-%y') as "RETURN DATE" from it_issu_retu;

	ISSUE DATE	RETURN DATE
▶	23-09-20	10-10-20
	12-08-20	20-08-20
	03-07-20	20-07-20
	13-06-20	20-06-20
	01-11-20	10-11-20
	13-09-20	29-09-20
	09-10-20	22-10-20
	03-09-20	10-09-20
	23-09-20	10-10-20
	23-09-20	10-10-20

e) Display student name and fine respectively.

select it_stud.it_stud_name as NAME, it_issu_fine as FINE from it_issu_retu inner join it_stud on it_stud.it_stud_code=it_issu_retu.it_stud_code;

	NAME	FINE
▶	Rohan	40
	Utkarsh	0
	Josh	10
	Rohit	0
	Yaman	20
	Josh	30
	Rohan	50
	Utkarsh	0
	Rohan	40
	Rohan	40

f) Display the name of the students who have issued book having title “operating system” by any author.

select it_stud.it_stud_name as NAME, it_book.it_book_title as TITLE, it_issu_doi as ISSUE_DATE from it_issu_retu join it_stud on it_stud.it_stud_code=it_issu_retu.it_stud_code join it_book on it_book.it_book_code=it_issu_retu.it_book_code where it_book_title="Operating System";

	NAME	TITLE	ISSUE_DATE
▶	Utkarsh	Operating System	2021-06-22

Experiment-4

Data Retrieval with Constraints

1. Queries on the tables given in the book by Ivon Bayross

a) Display all information about suppliers whose name begins with the letter 'ja'.

```
SELECT * FROM
    salesman_master

WHERE

    salesman_name LIKE 'ja%';
```

salesman_no	salesman_name	address1	address2	city	pincode	state	sal_amt	tgt_to_getytd_sales	remarks
11	Jayant	Flat No 201	City Palace	Gurugram	121004	Haryana	19000	10000	Good Work
12	Jassi	Flat No 205	City Palace	Gurugram	121004	Haryana	17000	10000	As Expected !

b) Display specified client information for the clients who are not in 'bombay' or 'delhi'

```
SELECT * FROM
    client_master

WHERE

    city IN ('Delhi' , 'Bombay');
```

client_no	name	address1	address2	city	pincode	state	bal_due
P001	Jatin Shah	Shop No: 121	New Delhi	Delhi	110054	Delhi	120.4
P003	Karan Jindal	Shop No: 21	Lajpat Nagar Market	Delhi	110032	Delhi	1300.4
P006	Varul Singla	Shop No: 332	West Delhi Market	Delhi	110054	Delhi	1340.4
P007	Ivan Bayross	Shop No: 202	Tilak Nagar	Delhi	110059	Delhi	241.4
P008	Ajay Seema	Shop No: 100	Shipra Mall	Bombay	534020	Maharashtra	250.4

c) Display the product number and the total quantity ordered for the products 101 from sales_order_details table.

```
SELECT * FROM
    sales_order_details

WHERE

    order_no = 101;
```

order_no	product_no	qty_ordered	qty_disp	product_rate
101	R007	5	9	10

d) Display the information like order_no., client_no., order_date for all the orders placed by clients in the ascending order of date. The order_date should be displayed in DD-MM-YY format.

```
SELECT

    order_no,

    client_no,

    DATE_FORMAT(order_date, '%d-%m-%Y') AS order_date FROM

    sales_order ORDER

BY order_date;
```

order_no	client_no	order_date
103	P004	12-05-2020
101	P003	12-05-2021
106	P002	12-07-2021
102	P001	16-08-2020
104	P005	17-04-2020
105	P006	22-09-2020
107	P007	26-07-2021

e) Retrieve the product_no and the total quantity ordered for products 101 , 104 from sales_order_details table

```
SELECT
    product_no, qty_disp AS 'Total Quantity' FROM
    sales_order_details
WHERE
    order_no IN (101 , 104);
```

product_no	Total Quantity
R007	9
R002	5

2. Queries on Library Table

a) Display all the student's details in the order of branch and library membership code.

```
SELECT * FROM
    it_stud
ORDER BY it_stud_card_no, it_stud_branch;
```

IT_stud_code	IT_stud_name	IT_stud_branch	IT_stud_sem	IT_stud_card_no	IT_stud_card_type
S009	Rohit	CIVIL	V	A23123	Day-scl
S005	Yaman	MECH	II	A23211	Day-scl
S001	Josh	CSE	IV	A23212	Hosteller
S002	Utkarsh	CSE	IV	A23232	Day-scl
S003	Rohan	IT	IV	A23234	Day-scl
S006	Rohit	CIVIL	IV	A23321	Hosteller
S004	Tanya	IT	I	A23432	Hosteller
S008	Jatin	IT	VI	A23455	Day-scl
S007	Rahul	CSE	III	A23521	Day-scl

b) Display publication,author and title in the descending order of publication , author, and ascending order of their title.

```
SELECT
    it_book_pub as Publication,
    it_book_author as Author, it_book_title
    as Title
FROM
    it_book
ORDER BY
    it_book_pub desc,
    it_book_author desc,
    it_book_title asc;
```

Publication	Author	Title
WAS House	PRD	Hello to my old friend
SQL House	DSE	The eye of the tiger
SQL House	ADC	Resumed in life
ASW House	FAR	The death of hercules
ABC House	XYZ	the tales of XYZ
ABC House	HKX	Annexure by hkx
ABC House	GFH	The dream catcher
ABC House	CBD	A new start

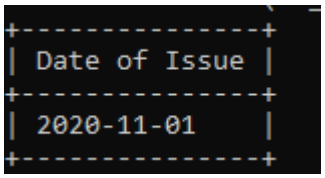
c) Display the book title of those books whose price is not given.

```
SELECT
    it_book_title AS title FROM
    it_book
WHERE
    it_book_price IS NULL;
```

title
New Bat

d) Display the date of issue of those books that are not returned for last 6 months.

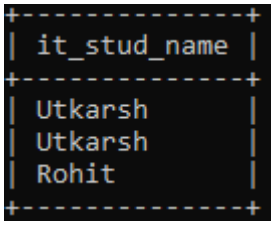
```
SELECT
    it_issu_doi AS 'Date of Issue' FROM
    it_issu_retu
WHERE
    DATEDIFF(it_issu_a_dor, it_issu_doi) > 182;
```



```
+---------------+
| Date of Issue |
+---------------+
| 2020-11-01    |
+---------------+
```

e) Display name of those students who have always returned books on time.

```
SELECT
    it_stud.it_stud_name FROM
    it_stud
        JOIN
    it_issu_retu ON it_issu_retu.it_stud_code = it_stud.it_stud_code WHERE
    it_issu_retu.it_issu_fine = 0;
```



```
+--------------+
| it_stud_name |
+--------------+
| Utkarsh      |
| Utkarsh      |
| Rohit        |
+--------------+
```

f) Display the books from ‘Tannenbaum’ which are published by ‘TMH’.

```
SELECT
    it_book_title AS title,
    it_book_author AS author,
    it_book_pub AS publisher
FROM
    it_book
WHERE
    it_book_pub = 'TMH';
```

title	author	publisher
The eye of the tiger	Tannenbaum	TMH

g) Display the details of those books which are moderately priced(if price between 1000 and 4000)……//300-700

```
SELECT
    *
FROM
    it_book
WHERE
    it_book_price BETWEEN 300 AND 600;
```

IT_book_code	IT_book_title	IT_book_author	IT_book_publ	IT_book_price	IT_book_eddi
B101	the tales of XYZ	XYZ	ABC House	400	1
B102	A new start	CBD	ABC House	550	2
B103	Annexure by hkx	HKX	ABC House	600	1
B104	Resumed in life	ADC	SQL House	450	3
B105	Hello to my old friend	PRD	WAS House	300	5
B107	The dream catcher	GFH	ABC House	340	3
B108	The eye of the tiger	Tannenbaum	TMH	550	4

Experiment - 5

Queries on Natural Join

1. Find out the products which have been sold to IVAN BAYROSS.

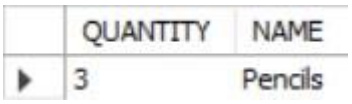
```
SELECT
    product_master.description AS name, client_master.name FROM
    product_master,
    sales_order,
    sales_order_details,
    client_master
WHERE
    product_master.product_no = sales_order_details.product_no AND
        sales_order.order_no = sales_order_details.order_no AND
        client_master.client_no=sales_order.client_no
        AND client_master.name = 'IVON BAYROSS';
```

	name	Name
▶	Highlighters	IVON BAYROSS

2. Find out the products and their quantity that will have to be delivered in the current month.

```
SELECT
    sales_order_details.qty_ordered AS QUANTITY,
    product_master.description AS NAME
FROM
    sales_order,
    sales_order_details,
    product_master
WHERE
    sales_order.order_no = sales_order_details.order_no
        AND sales_order_details.product_no = product_master.product_no
```

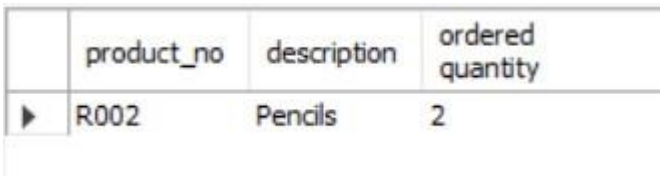
```
AND MONTH(dely_date) = MONTH(SYSDATE());
```



	QUANTITY	NAME
▶	3	Pencils

3. Find out the product_no. and description of constantly sold i.e. rapidly moving product.

```
SELECT
    sales_order_details.product_no,
    product_master.description AS description,
    COUNT(sales_order_details.product_no) AS 'ordered quantity' FROM
    sales_order_details,
    product_master
WHERE
    product_master.product_no = sales_order_details.product_no GROUP BY
sales_order_details.product_no
HAVING COUNT(sales_order_details.product_no) > 1;
```

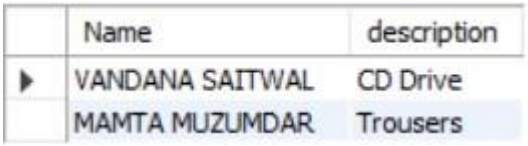


	product_no	description	ordered quantity
▶	R002	Pencils	2

4. Find the name of clients who have purchased CD Drive……//Trousers

```
SELECT
    client_master.Name, product_master.description FROM
    product_master,
    sales_order,
    sales_order_details,
    client_master
WHERE
    product_master.product_no = sales_order_details.product_no AND
        sales_order.order_no = sales_order_details.order_no AND
        client_master.client_no=sales_order.client_no
```

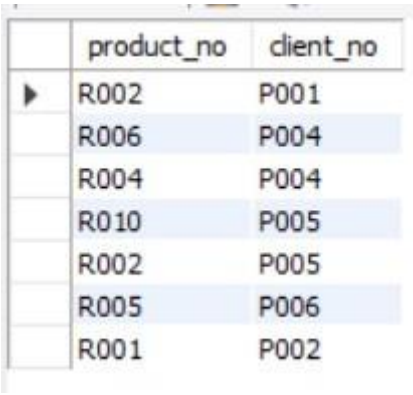
```
AND product_master.description IN ('CD Drive' , 'Trousers');
```



	Name	description
▶	VANDANA SAITWAL	CD Drive
	MAMTA MUZUMDAR	Trousers

5. List the product_no.,Order_no. of customers having qty_ordered less than 5 from sales_order_details table from the product “CD Drives” …..// “pull Overs”

```
SELECT
    sales_order_details.product_no, sales_order.client_no FROM
    sales_order_details,
    sales_order
WHERE
    sales_order_details.order_no = sales_order.order_no AND
        sales_order_details.qty_ordered < 5;
```



	product_no	client_no
▶	R002	P001
	R006	P004
	R004	P004
	R010	P005
	R002	P005
	R005	P006
	R001	P002

6. Find the product and their quantity for the orders placed by ‘IVAN BAYROSS” and ‘VANDANA SAITWAL”…..//Mamta Muzumdar

```
SELECT
    product_master.description, sales_order_details.qty_ordered FROM
    sales_order_details,
    sales_order,
    product_master,
```

```
        client_master
WHERE
        sales_order_details.order_no = sales_order.order_no
                AND sales_order_details.product_no = product_master.product_no AND
                sales_order.client_no = client_master.client_no
                AND client_master.name IN ('Ivon Bayross' , 'Vandana Saitwal', 'Mamta
Muzumdar');
```

	description	qty_ordered
▶	Markers	1
	Pencils	2
	Highlighters	5
	Trousers	1

7. Find the product and quantities for the orders placed by client_no. ‘P001 and ‘P002’

```
SELECT
        product_master.description, sales_order_details.qty_ordered FROM
        sales_order_details,
        sales_order,
        product_master,
        client_master
WHERE
        sales_order_details.order_no = sales_order.order_no
                AND sales_order_details.product_no = product_master.product_no AND
                sales_order.client_no = client_master.client_no
                AND client_master.client_no IN ('P001' , 'P002');
```

	description	qty_ordered
▶	Markers	1
	Pencils	3

Queries on “Library” table

a) Find the book title issued to ‘Gaurav’.

```
SELECT
    it_book.it_book_title, it_stud.it_stud_name FROM
    it_book, it_stud,
    it_issu_retu
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code
        AND it_stud.it_stud_code = it_issu_retu.it_stud_code AND
        it_stud.it_stud_name = 'Gaurav';
```

	it_book_title	it_stud_name
▶	Annexure by hkx	Gaurav

b) Find out the book title of those books which are having atleast 20 copies and not a single issued.

```
SELECT
    it_book.it_book_title AS title FROM
    it_book,
    it_issu_retu
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code
        AND it_book.it_book_code NOT IN (select it_book_code from it_issu_retu)
        AND it_book.it_book_copy > 20;
```

Empty Set .

c) Find the details of the book that is issued maximum times.

```
SELECT
    *
FROM
    it_book,
    it_issu_retu
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code GROUP BY
it_issu_retu.it_book_code
HAVING COUNT(it_issu_retu.it_book_code) > 1;
```

IT_book_code	IT_book_title	IT_book_author	it_book_pub	IT_book_price	IT_book_eddi	it_book_copy	it_issu_doi	it_issu_dor	IT_issu_fine	IT_book_code	IT_stud_coc
B101	the tales of XYZ	XYZ	ABC House	400	1	10	2020-09-23	2020-10-10	40	B101	S003
B103	Annexure by hkx	HKX	ABC House	600	1	14	2020-07-03	2020-07-20	10	B103	S001
B102	A new start	CBD	ABC House	550	2	20	2020-06-13	2020-06-20	0	B102	S006

d) Find the details of students who have taken books that are published by ‘TMH’.

```
SELECT
    it_stud.*
FROM
    it_book,
    it_issu_retu,
    it_stud
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code
        AND it_stud.it_stud_code= it_issu_retu.it_stud_code AND
        it_book.it_book_pub="TMH";
```

	IT_stud_code	IT_stud_name	IT_stud_branch	IT_stud_sem	IT_stud_card_no	IT_stud_card_type
▶	S001	Josh	CSE	IV	A23212	Hosteller
	S004	Tanya	IT	I	A23432	Hosteller
	S005	Gaurav	MECH	II	A23211	Day-scl

e) Find name of the students and book title that are issued last month.

```
SELECT
    it_stud.it_stud_name, it_book.it_book_title FROM
    it_book,
    it_issu_retu,
    it_stud
```

```
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code
        AND it_stud.it_stud_code = it_issu_retu.it_stud_code AND
        DATEDIFF(CURDATE(), it_issu_doi) < 31;
```

	it_stud_name	it_book_title
▶	Tanya	Resumed in life

f) Find the title and price of those books that are scheduled for deposit this week.

```
SELECT
    it_book.it_book_title, it_book.it_book_price FROM
    it_book,
    it_issu_retu,
    it_stud
WHERE
    it_book.it_book_code = it_issu_retu.it_book_code
        AND it_stud.it_stud_code = it_issu_retu.it_stud_code AND
        DATEDIFF(CURDATE(), it_issu_dor) < 7;
```

	it_book_title	it_book_price
▶	Resumed in life	450
	A new life in London	NULL

g) Find publisher name for the book with the same title and different publications.

```
SELECT
    it_book_pub AS publisher
FROM
    it_book
HAVING COUNT(it_book_title) >= 2;
```

	publisher
▶	ABC House

Experiment-6

Simple Data Retrieval

1) Queries on the table given in the book by IVON BAYROSS

a) Find the product no. and description of non-moving products i.e. not being sold.

select Description, Product_No from Product_master WHERE Product_No NOT IN (select Product_No from Sales_order_details);

Description	Product_No
Ball Point Pen	R003
Maths Notebook	R006
Erasers	R008
Blue Gel Pen	R009

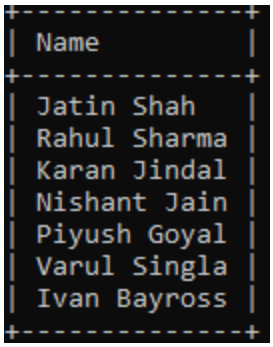
b) Find customer name, address1, address2, city and pincode for the client who has placed order no. '119'.

select Name, ADDRESS1, ADDRESS2, CITY, PINCODE from client_master WHERE Client_No IN (select Client_No from Sales_order WHERE Order_No=119);

Name	ADDRESS1	ADDRESS2	CITY	PINCODE
Rahul Sharma	Shop No: 320	HUDA MARKET	Gurugram	121004

c) Find the client name that has placed orders before the month of august 31’ 2021.

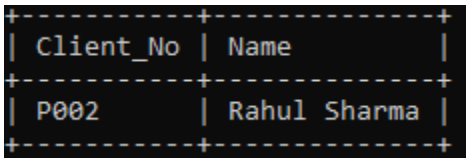
select Name from client_master WHERE Client_No IN (select Client_No from Sales_order WHERE ORDER_DATE<"2021-08-31");



Name
Jatin Shah
Rahul Sharma
Karan Jindal
Nishant Jain
Piyush Goyal
Varul Singla
Ivan Bayross

d) Find out if the product 'bracelet' has been ordered by any client and print the client no., name to whom it was sold.

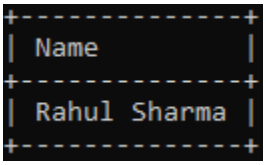
select Client_No, Name from client_master WHERE Client_No IN (select Client_No from Sales_order WHERE Order_no IN (select Order_No from Sales_order_details WHERE Product_No IN (select Product_no from Product_master WHERE Description="bracelet")));



Client_No	Name
P002	Rahul Sharma

e) Find the name of the client who has placed order worth Rs.1000 or more.

select Name from client_master WHERE Client_No IN (select Client_No from Sales_order WHERE Order_No IN (select Order_No from Sales_order_details WHERE QTY_ORDERED*Product_Rate>=1000));



Name
Rahul Sharma

2) Queries on “Library “table

a) Find the book title and author whose price is above average price.

SELECT Book_Title, Book_author, Book_price FROM IT_book WHERE Book_price > (SELECT AVG(Book_price) FROM IT_book);

IT_book_title	IT_book_author	IT_book_price
A new start	CBD	550
Annexure by hkx	HKX	600
Resumed in life	ADC	450
The eye of the tiger	Tannenbaum	550

b) Find the name of the student who has taken the book whose price is highest.

SELECT Book_Title, Book_price FROM IT_book WHERE Book_price=(SELECT MAX(Book_price) FROM IT_book);

IT_book_Title	IT_book_price
Annexure by hkx	600

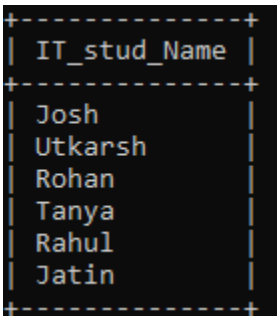
c) Find the name of the students of 'CSE' who has issued a book whose price is highest among the books issued by EC students.

select Stud_Name from IT_Stud WHERE Stud_branch="CSE" and Stud_code IN (select Stud_code from IT_issu_retu WHERE Book_code IN (select Book_code from IT_book WHERE Book_price=(select MAX(Book_price) from IT_book)));

IT_stud_Name
Josh

d) Find name of all students who have the costliest book issued in every branch.

select Stud_Name from IT_Stud WHERE Stud_branch="CSE" or Stud_branch="IT" or Stud_branch="AI" and Stud_code IN (select Stud_code from IT_issu_retu WHERE Book_code IN (select Book_code from IT_book WHERE Book_price=(select MAX(Book_price) from IT_book)));



IT_stud_Name
Josh
Utkarsh
Rohan
Tanya
Rahul
Jatin

EXPERIMENT 7

Aim : Queries on View

a) Create view of students containing student name, registration no., date of issue and fine.

CREATE VIEW Student_details AS SELECT IT_Stud.Stud_Name AS Name, IT_Stud.Stud_Card_No AS Registration_no, IT_issu_retu.IT_issu_doi AS 'Date of Issue', IT_issu_fine AS 'Fine' FROM IT_Stud, IT_issu_retu WHERE IT_Stud.Stud_code = IT_issu_retu.Stud_code;

select * from Student_details;

```
mysql> select * from Student_details;
+-----------+-----------------+---------------+------+
| Name      | Registration_no | Date of Issue | Fine |
+-----------+-----------------+---------------+------+
| Caroline  |            1106 | 2021-07-27    |    0 |
| Elena     |            1101 | 2021-07-18    |    0 |
| Bonnie    |            1105 | 2021-07-17    |   10 |
| Josie     |            1109 | 2021-07-15    |   20 |
| Katherine |            1108 | 2021-07-14    |   40 |
+-----------+-----------------+---------------+------+
5 rows in set (0.03 sec)
```

b) Create view a_detail contains all the details of IT students.

CREATE VIEW a_details AS SELECT IT_Stud.Stud_Name AS NAME, IT_Stud.Stud_branch AS Branch, IT_Stud.Stud_Sem AS SEM, IT_Stud.Stud_Card_No AS 'Registration', IT_Stud.Stud_Card_Type AS 'Type', IT_issu_retu.Stud_code AS 'Student_Code', IT_issu_retu.IT_issu_doi AS 'Date_of_issue', IT_issu_retu.IT_issu_dor AS 'Date_of_return', IT_issu_retu.IT_issu_fine AS 'Fine' FROM IT_Stud, IT_issu_retu WHERE IT_issu_retu.Stud_code = IT_Stud.Stud_code and Stud_branch = 'IT';

Select * from a_details;

```
mysql> select * from a_details;
+----------+--------+------+--------------+-------+--------------+---------------+----------------+------+
| NAME     | Branch | SEM  | Registration | Type  | Student_Code | Date_of_issue | Date_of_return | Fine |
+----------+--------+------+--------------+-------+--------------+---------------+----------------+------+
| Caroline | IT     |    3 |         1106 | basic |          105 | 2021-07-27    | 2021-07-10     |    0 |
+----------+--------+------+--------------+-------+--------------+---------------+----------------+------+
1 row in set (0.04 sec)
```

c) Display the total fine for eacd IT student from view a_detail.

select Fine AS 'Total Fine' from a_details;

```
mysql> select Fine AS 'Total Fine' from a_details;
+------------+
| Total Fine |
+------------+
|          0 |
+------------+
1 row in set (0.00 sec)
```

d) Insert five records (2 of IT and 3 of CS in the view) :

CREATE VIEW b_details AS select * from IT_Stud;

insert into b_details (Stud_code, Stud_Name, Stud_branch, Stud_Sem, Stud_Card_No, Stud_Card_Type) values(121,"Penny","IT",4,1114,"basic");

insert into b_details (Stud_code, Stud_Name, Stud_branch, Stud_Sem, Stud_Card_No, Stud_Card_Type) values(127,"Bernadett","IT",3,1117,"premium");

insert into b_details (Stud_code, Stud_Name, Stud_branch, Stud_Sem, Stud_Card_No, Stud_Card_Type) values(123,"Amy","CSE",5,1111,"premium");

insert into b_details (Stud_code, Stud_Name, Stud_branch, Stud_Sem, Stud_Card_No, Stud_Card_Type) values(125,"Sheldon","CSE",5,1115,"premium");

insert into b_details (Stud_code, Stud_Name, Stud_branch, Stud_Sem, Stud_Card_No, Stud_Card_Type)
values(129,"Leonard","CSE",3,1112,"basic");

select * from b_details;

```
mysql> select * from b_details;
+-----------+-----------+-------------+----------+--------------+----------------+
| Stud_code | Stud_Name | Stud_branch | Stud_Sem | Stud_Card_No | Stud_Card_Type |
+-----------+-----------+-------------+----------+--------------+----------------+
|       101 | Elena     | CSE         |        5 |         1101 | premium        |
|       102 | Bonnie    | CSE         |        1 |         1105 | basic          |
|       105 | Caroline  | IT          |        3 |         1106 | basic          |
|       107 | Katherine | AI          |        5 |         1108 | premium        |
|       108 | Josie     | CSE         |        7 |         1109 | premium        |
|       121 | Penny     | IT          |        4 |         1114 | basic          |
|       127 | Bernadett | IT          |        3 |         1117 | premium        |
|       123 | Amy       | CSE         |        5 |         1111 | premium        |
|       125 | Sheldon   | CSE         |        5 |         1115 | premium        |
|       129 | Leonard   | CSE         |        3 |         1112 | basic          |
+-----------+-----------+-------------+----------+--------------+----------------+
10 rows in set (0.00 sec)
```

e) Display the details of CS students which we have inserted today.

Above is the screenshot

f) Change the branch of the students who has got maximum fine from IT to MA in the view a_detail.

UPDATE a_details SET Branch="Mech"

WHERE Fine=(select MAX(Fine)) and Branch="IT";

```
mysql> UPDATE a_details
    -> SET Branch="Mech"
    -> WHERE Fine=(select MAX(Fine)) and Branch="IT";
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

g) Create 2 views cs_detail and ai_detail containing details of cs and ai students.

CREATE VIEW cs_details AS select * from IT_Stud WHERE Stud_branch="CSE"; select * from cs_details;

```
mysql> select * from cs_details;
+-----------+-----------+-------------+----------+--------------+----------------+
| Stud_code | Stud_Name | Stud_branch | Stud_Sem | Stud_Card_No | Stud_Card_Type |
+-----------+-----------+-------------+----------+--------------+----------------+
|       101 | Elena     | CSE         |        5 |         1101 | premium        |
|       102 | Bonnie    | CSE         |        1 |         1105 | basic          |
|       108 | Josie     | CSE         |        7 |         1109 | premium        |
|       123 | Amy       | CSE         |        5 |         1111 | premium        |
|       125 | Sheldon   | CSE         |        5 |         1115 | premium        |
|       129 | Leonard   | CSE         |        3 |         1112 | basic          |
+-----------+-----------+-------------+----------+--------------+----------------+
6 rows in set (0.04 sec)
```

CREATE VIEW ai_details AS select * from IT_Stud WHERE Stud_branch="AI"; select * from ai_details;

```
mysql> select * from ai_details;
+-----------+-----------+-------------+----------+--------------+----------------+
| Stud_code | Stud_Name | Stud_branch | Stud_Sem | Stud_Card_No | Stud_Card_Type |
+-----------+-----------+-------------+----------+--------------+----------------+
|       107 | Katherine | AI          |        5 |         1108 | premium        |
+-----------+-----------+-------------+----------+--------------+----------------+
1 row in set (0.00 sec)
```

h) Create a view CS_AI containing the details of CS and AI students whose name is ‘Gaurav’.

CREATE VIEW CS_AI AS select * from IT_Stud WHERE Stud_Name="Gaurav";

Select * from CS_AI;

```
mysql> select * from CS_AI;
+-----------+-----------+-------------+----------+--------------+----------------+
| Stud_code | Stud_Name | Stud_branch | Stud_Sem | Stud_Card_No | Stud_Card_Type |
+-----------+-----------+-------------+----------+--------------+----------------+
|       128 | Gaurav    | CSE         |        3 |         1112 | basic          |
|       124 | Gaurav    | AI          |        1 |         1116 | premium        |
+-----------+-----------+-------------+----------+--------------+----------------+
2 rows in set (0.00 sec)
```

Naman Mittal
5CSE 12X
A2305219718

DBMS

PRACTICE EXERCISE 1

QUESTION :

Consider the following relational schema for the Office of the Controller of Examinations Application. Write SQL queries for creating, adding constraints in the schema

Student (Rollno, Name, Dob, Gender, Doa, Bcode); Gender can be M of F, Date of Admission

Branch (Bcode, Bname, Dno); Department (Dno, Dname);

Course (Ccode, Cname, Credits, Dno); Branch_Course (Bcode, Ccode, Semester); Enrolls (Rollno, Ccode, Sess, Grade);

Add following Constraints: SESS can take values 'APRIL 2013', 'NOV 2013'

Implement a check constraint for grade Value Set ('S', 'A', 'B', 'C', 'D', 'E', 'U');

Students are admitted to Branches and they are offered by Departments. A branch is offered by only one department.

Each branch has a set of Courses (Subjects). Each student must enroll during a semester. Courses are offered by Departments. A course is offered only by one department. If a student is unsuccessful in a course he/she must enroll for the course during next session. A student has successfully completed a course if the grade obtained by is from the list (A, B, C, D, and E).

Question (A) Develop a SQL query to list details of Departments that offer more than 3 branches.

SELECT * FROM DEPARTMENT WHERE DEPARTMENT.DNO IN (SELECT BRANCH.DNO FROM BRANCH GROUP BY BRANCH.DNO HAVING COUNT(BRANCH.DNO) > 3);

```
mysql> SELECT * FROM DEPARTMENT WHERE DEPARTMENT.DNO IN
    -> (SELECT BRANCH.DNO FROM BRANCH GROUP BY BRANCH.DNO
    -> HAVING COUNT(BRANCH.DNO) > 3);
+-----+-------------+
| DNO | DNAME       |
+-----+-------------+
|  11 | Engineering |
+-----+-------------+
```

Question (B) Develop a SQL query to list the details of Departments that offer more than 6 courses.

SELECT * FROM DEPARTMENT WHERE DEPARTMENT.DNO IN (SELECT COURSE.DNO FROM COURSE GROUP BY COURSE.DNO HAVING COUNT(COURSE.CCODE) > 6);

Naman Mittal
5CSE 12X
A2305219718

```
mysql> SELECT * FROM DEPARTMENT WHERE DEPARTMENT.DNO IN
    -> (SELECT COURSE.DNO FROM COURSE GROUP BY COURSE.DNO
    -> HAVING COUNT(COURSE.CCODE) > 6);
+-----+-------------+
| DNO | DNAME       |
+-----+-------------+
|   1 | Mathematics |
+-----+-------------+
1 row in set (0.00 sec)
```

Question (C) Develop a SQL query to list the details of courses that are common for more than 3 branches.

SELECT * FROM COURSE WHERE COURSE.CCODE IN (SELECT BRANCH_COURSE.CCODE FROM BRANCH_COURSE GROUP BY BRANCH_COURSE.CCODE HAVING COUNT(BRANCH_COURSE.BCODE) > 3);

```
mysql> SELECT * FROM COURSE WHERE COURSE.CCODE IN (SELECT
    -> BRANCH_COURSE.CCODE FROM BRANCH_COURSE GROUP BY
    -> BRANCH_COURSE.CCODE HAVING
    -> COUNT(BRANCH_COURSE.BCODE) > 3);
Empty set (0.01 sec)
```

Question (D) Develop a SQL query to list students who got ‘E’ in more than 2 courses during single enrollment.

SELECT * FROM STUDENT WHERE Student.ROLLNO IN (SELECT ENROLLS.ROLLNO FROM ENROLLS WHERE ENROLLS.GRADE = 'E' GROUP BY ENROLLS.ROLLNO HAVING COUNT(ENROLLS.GRADE) > 2);

```
mysql> SELECT * FROM STUDENT WHERE Student.ROLLNO IN (SELECT
    -> ENROLLS.ROLLNO FROM ENROLLS WHERE ENROLLS.GRADE = 'E'
    -> GROUP BY ENROLLS.ROLLNO HAVING COUNT(ENROLLS.GRADE) >
    -> 2);
+--------+-------------+------------+--------+------------+-------+
| ROLLNO | NAME        | DOB        | GENDER | DOA        | BCODE |
+--------+-------------+------------+--------+------------+-------+
|  12007 | Garvit Rana | 1998-06-15 | M      | 2015-04-19 |   106 |
+--------+-------------+------------+--------+------------+-------+
1 row in set (0.00 sec)
```

Question (E) Create a view that will keep track of the roll number, name and number of courses, a student has completed successfully.

Naman Mittal
5CSE 12X
A2305219718

CREATE VIEW STUD_DATA AS SELECT ENROLLS.ROLLNO, STUDENT.NAME, COUNT(ENROLLS.CCODE) AS Course_Count FROM STUDENT , ENROLLS WHERE ENROLLS.ROLLNO = STUDENT.ROLLNO AND ENROLLS.GRADE!='U' GROUP BY ENROLLS.ROLLNO, STUDENT.NAME;

```
mysql> CREATE VIEW STUD_DATA AS SELECT ENROLLS.ROLLNO,
    -> STUDENT.NAME, COUNT(ENROLLS.CCODE) AS Course_Count FROM
    -> STUDENT , ENROLLS WHERE ENROLLS.ROLLNO =
    -> STUDENT.ROLLNO AND ENROLLS.GRADE!='U' GROUP BY
    -> ENROLLS.ROLLNO, STUDENT.NAME;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from STUD_DATA;
+--------+------------------+--------------+
| ROLLNO | NAME             | Course_Count |
+--------+------------------+--------------+
|  12001 | Ajay Rana        |            1 |
|  12002 | Vaibhav Kumar    |            2 |
|  12003 | Prima Kumar      |            2 |
|  12004 | Megha Pandit     |            1 |
|  12005 | Seema Abrol      |            1 |
|  12006 | Hitanshi Chauhan |            1 |
|  12007 | Garvit Rana      |            3 |
|  12008 | Khagesh Nagpal   |            2 |
|  12009 | Dolly Bindra     |            1 |
|  12010 | Priya Munjal     |            1 |
|  12011 | Kanchana Munjal  |            1 |
+--------+------------------+--------------+
11 rows in set (0.00 sec)
```

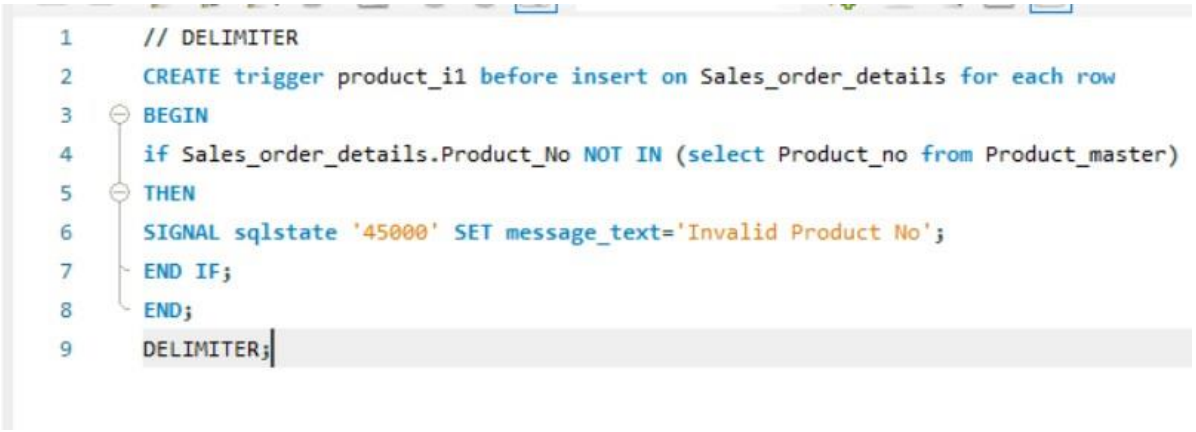
Experiment 9

AIM: Queries of Triggers

1) Queries on the table given in the book by IVON BAYROSS

a) Create a trigger product_i which should ensure that the product_no given at the time of insertion, in sales_order_master should be present in product_master.

```
// DELIMITER

CREATE trigger product_i1 before insert on Sales_order_details for each row

BEGIN

if Sales_order_details.Product_No NOT IN (select Product_no from
Product_master)

THEN

SIGNAL sqlstate '45000' SET message_text='Invalid Product No';

END IF;

END;

DELIMITER;
```



b) Create a trigger sales_u which should ensure that if client_no or order_no is modified in the sales_order table it should do the corresponding changes in the other table.

```
DELIMITER //

CREATE trigger Sales_u

AFTER

      insert on Sales_order for each row

    BEGIN

            if Sales_order.Client_no, Sales_order.Order_No MODIFIED

        THEN

            UPDATE Client_no IN (select Client_no from client_master) AND
UPDATE Order_No IN (select Order_No from Sales_order_details);
```

```
            END IF;
        END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger Sales_u
AFTER
    insert on Sales_order for each row
    BEGIN
        if Sales_order.Client_no, Sales_order.Order_No MODIFIED
        THEN
        UPDATE Client_no IN (select Client_no from client_master) AND UPDATE Order_No IN (select Order_No from Sales_order_details);
            END IF;
            END;
        DELIMITER;
```

c) Create a trigger sales_u which should ensure that if order_no is modified in sales_order_details table then it should not allow it and display an error message.

```
DELIMITER //
CREATE trigger Sales_u BEFORE insert on Sales_order_details for each row
BEGIN if Sales_order_deatils.Order_No MODIFIED THEN
SIGNAL sqlstate '45000' SET MESSAGE_TEXT="ERROR";
END IF;
END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger Sales_u BEFORE insert on Sales_order_details for each row
BEGIN if Sales_order_deatils.Order_No MODIFIED THEN
SIGNAL sqlstate '45000' SET MESSAGE_TEXT="ERROR";
END IF;
END;
DELIMITER;
```

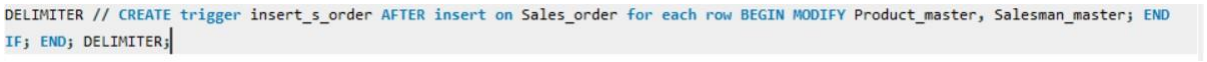
d) Create a trigger product_d which should ensure that if a product is deleted from product master table all corresponding entries should be removed from the issued table.

```
DELIMITER //
CREATE trigger product_d
      AFTER delete on Product_master for each row
BEGIN delete from Sales_order_details
      WHERE Product_master.Product_no=Sales_order_details.Product_no;
END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger product_d
    AFTER delete on Product_master for each row
BEGIN delete from Sales_order_details
    WHERE Product_master.Product_no=Sales_order_details.Product_no;
END;
DELIMITER;
```

e) Create a trigger insert _s_order which should do the corresponding changes in product_master and salesman_master table at the time of insertion in sales_order table.

```
DELIMITER //
CREATE trigger insert_s_order
      AFTER insert on Sales_order for each row
    BEGIN
            MODIFY product_master, Salesman_master;
    END;
DELIMITER;
```



f) Create a trigger client _u which should update the bal_due in client_master immediately when any bill of that client is cleared.

```
DELIMITER //
CREATE trigger client_u
      AFTER update of Sales_order.BILLED_YN on Sales_order for each row
    BEGIN
            if Sales_order.BILLED_YN="yes" then
                  UPDATE client_master SET client_master.BAL_DUE=0
                        WHERE client_master.Client_no=Sales_order.Client_no
;
            END IF;
      END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger client_u
    AFTER update of Sales_order.BILLED_YN on Sales_order for each row
    BEGIN
        if Sales_order.BILLED_YN="yes" then
            UPDATE client_master SET client_master.BAL_DUE=0
                WHERE client_master.Client_no=Sales_order.Client_no ;
        END IF;
    END;
DELIMITER;
```

2) Queries on “Library “table

a) Create a trigger book_p which should ensure that the book_code present in the issued table should be present in book table.

```
DELIMITER //
CREATE trigger book_p BEFORE insert on IT_issu_retu for each row
      BEGIN
DECLARE x INT;
            select COUNT(*) INTO x from IT_issu_retu
                  WHERE IT_Book.Book_code=IT_issu_retu.Book_code;
                        if x=0
                              then SIGNAL sqlstate '45000' SET
MESSAGE_TEXT="ERROR! This book doesn't exist in the table" ;
                        END IF;
      END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger book_p BEFORE insert on IT_issu_retu for each row
    BEGIN
DECLARE x INT;
       select COUNT(*) INTO x from IT_issu_retu
           WHERE IT_Book.Book_code=IT_issu_retu.Book_code;
               if x=0
                   then SIGNAL sqlstate '45000' SET MESSAGE_TEXT="ERROR! This book doesn't exist in the table" ;
               END IF;
    END;
DELIMITER;
```

b) Create a trigger book_u which should ensure that the book_code or student_code is modified in the issue table it should do the corresponding changes in the other tables

```
DELIMITER //
CREATE trigger book_u AFTER update on Book_code, Stud_code on IT_issu_retu
for each row
      BEGIN
            update IT_Book SET IT_Book.Book_code=IT_issu_retu.Book_code;
        update IT_Stud SET IT_Stud.Stud_code=IT_issu_retu.Stud_code;
      END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger book_u AFTER update on Book_code, Stud_code on IT_issu_retu for each row
    BEGIN
        update IT_Book SET IT_Book.Book_code=IT_issu_retu.Book_code;
        update IT_Stud SET IT_Stud.Stud_code=IT_issu_retu.Stud_code;
    END;
DELIMITER;
```

c) Create a trigger student_u which should ensure that the student code is modified in the student table then it should not allow it and display an error message.

```
DELIMITER //
CREATE trigger student_u
      BEFORE update of Stud_code on IT_Stud for each row
            BEGIN SIGNAL sqlstate '45000' SET MESSAGE_TEXT="Error! Student
      code cannot be modified" ;
            END;
DELIMITER;
```

```
DELIMITER //
CREATE trigger student_u
BEFORE update of Stud_code on IT_Stud for each row
BEGIN SIGNAL sqlstate '45000' SET MESSAGE_TEXT="Error! Student code cannot be modified" ;
END;
DELIMITER;
```

d) Create a trigger book_d which should ensure that if a book is deleted from book table all corresponding entries should be removed from issued table.

```
DELIMITER //

CREATE trigger book_d

      AFTER delete on IT_Book for each row

            BEGIN DELETE from IT_issu_retu

                  WHERE IT_issu_retu.Book_code=IT_Book.Book_code ;

        END;

DELIMITER;
```

```
DELIMITER //
CREATE trigger book_d
    AFTER delete on IT_Book for each row
        BEGIN DELETE from IT_issu_retu
            WHERE IT_issu_retu.Book_code=IT_Book.Book_code ;
        END;
DELIMITER;
```

Experiment 10

AIM: Index Creation

Single column Index

```
create INDEX index_1 ON Product_master (REORDER_LVL);
```

```
create INDEX index_1 ON Product_master (REORDER_LVL);
```

Multiple column Index

```
create INDEX index_2 ON Product_master (QTY_ON_HAND, Profit_Percent);
```

```
show index from Product_master from sales;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
product_master	1	id1	1	product_no	A	10	NULL	NULL	YES	BTREE			YES
product_master	1	id3	1	description	A	10	NULL	NULL	YES	BTREE			YES
product_master	1	index_1	1	reorder_lvl	A	8	NULL	NULL	YES	BTREE			YES

Unique Index

```
create UNIQUE INDEX index_3 ON client_master (Client_No);
```

```
show index from client_master;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expres

Experiment 11

Commands of PL/SQL.

1. **What are the components of a PL/SQL code block?**

   PL/SQL block has three parts:
   i. Declarative part
   ii. Executable part
   iii. Exception-handling part.

   The order of the parts is logical. First comes the declarative part, in which items can be declared. Once declared, items can be manipulated in the executable part. Exceptions raised during execution can be dealt with in the exception-handling part.

2. **What is the maximum value that can be stored in a variable bound to a table column?**

Data type	Length
Binary	8000 bytes
Var binary	8000 bytes
Image	2,147,483,647 bytes
Float	-1.79E + 308 to 1.79E + 308
Real	-3.40e + 38 to 3.40E + 38
Char	8000 characters
Varchar	8000 characters
Text	2,147,483,647 characters

3. **Write a PL/SQL code block that will accept a number from the user and debit an account of Rs.2000 from all the account has a minimum balance of 500 after the amount is debited. The process is to find account table.**

Account_id	Name	Bal
AC001	Anuj	5000
AC002	Robert	10000
AC003	Mita	5000
AC004	Sunita	15000
AC005	Melba	10000

```
Declare
  acct_balance number(7,2);
  acct_no varchar2(6);
  debit_amt number(7,2):=2000.00;
  min_bal constant number(7,2):=500.00;
  BEGIN
     acct_no:=&acct_no;
     SELECT bal INTO acct_balance
     FROM accounts3
     WHERE account_id=acct_no;
     acct_balance:=acct_balance-debit_amt;
     IF acct_balance>=min_bal THEN
         UPDATE accounts3 SET bal=bal-debit_amt
         WHERE account_id=acct_no;
     END IF;
  END;
```

Account_id	Name	Bal
AC001	Anuj	3000
AC002	Robert	8000
AC003	Mita	3000
AC004	Sunita	13000
AC005	Melba	8000

4. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in table.

```
Declare
  pi constant number(4,2):=3.14;
  radius number(5);
  area number(14,2);
  BEGIN
      radius :=3;
      while radius <=10
      loop
      area := pi*power(radius,2);
      INSERT INTO areas VALUES(radius,area);
      radius := radius+1;
  END loop;
  END;
```

Radius	Area
3	28.26
4	50.24
5	78.5
6	113.04
7	153.86

5. **Writ a PL/SQL block of code for investing a number ‘102345’ and a string ‘AMITY’.**

```
Declare
 num VARCHAR(6):='102345';
 len number(2);
 rev VARCHAR(6);
 BEGIN
     len:=length(num);
     FOR cntr IN reverse 1..len
       loop
       rev:=rev||substr(num,cntr,1);
     END loop;
     dbms_output.put_line('The Given Number is'||num);
     dbms_output.put_line('The Inverted Number is'||rev);
 END;
```

```
1  Declare
2    given_str VARCHAR(5):='AMITY';
3    str_length number(2);
4    inv_str VARCHAR(5);
5    BEGIN
6       str_length := length(given_str);
7       FOR cntr IN REVERSE 1..str_length
8          loop
9          inv_str := inv_str || substr(given_str,cntr,1);
10         END loop;
11      dbms_output.put_line('The given String is' || given_str);
12      dbms_output.put_line('The inverted String is ' || inv_str);
13   END;
```

6. **Write a PL/SQL block of code that first inserts record in an ‘emp’ table. Update the salary of Blake and Clark by Rs. 2500 and Rs. 3000.Then check to see that the total salary doesn’t exceed 25000.If the total salary is greater than 25000 then undo the update made to the salaries of Blake and Clark.**

emp_no	emp_name	salary
E001	ANUJ	5000
E002	HARMY	10000
E003	BLAKE	5000
E004	JACK	15000
E005	CLARK	10000

```
Declare
  sal number(8);
  BEGIN
      SELECT salary INTO sal
      FROM employee
      WHERE emp_name='BLAKE';
      sal:=sal+2500;
      IF sal <= 25000 THEN
        UPDATE employee SET salary=salary+2500
        WHERE emp_name='BLAKE';
      END IF;
      SELECT salary INTO sal
      FROM employee
      WHERE emp_name='CLARK';
      sal:=sal+3000;
      IF sal <= 25000 THEN
        UPDATE employee SET salary=salary+3000
        WHERE emp_name='CLARK';
      END IF;
  END;
```

emp_no	emp_name	salary
E001	ANUJ	5000
E002	HARMY	10000
E003	BLAKE	7500
E004	JACK	15000
E005	CLARK	13000