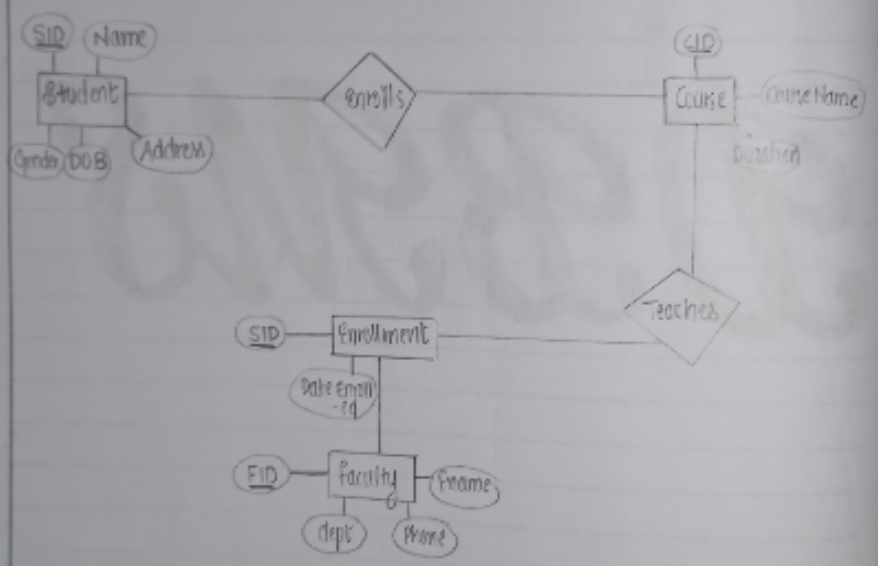
DBMS

SL No.	DATE	DESCRIPTION OF THE TITLE OF PROGRAM	PAGE NO.
I	3.07.2025	ER Diagrams	1 - 4
II	4.07.2025	DDL Commands	5 - 7
III	23.07.2025	DML Commands	8 - 13
IV	5.08.2025	Queries with constraints	14 -
V	7.08.2025	Queries on TCL	15 - 17
VI	12.08.2025	Aggregate Functions	18 -
VII	14.08.2025	Queries on GROUPBY, HAVING & ORDER BY	19 - 20
VIII	19.08.2025	Queries on Operators	21 - 23
IX	20.08.2025	Join operators	24 - 26
X	9.09.2025	Set theory operators	27 - 28
XI	18.09.2025	Queries on set theory operators	29 - 30
XII	20.09.2025	MongoDB Queries.	31 - 37

Output:



3 / 07 / 25

I ER - DIAGRAMS

1. Draw an ER Diagram of Registar Office

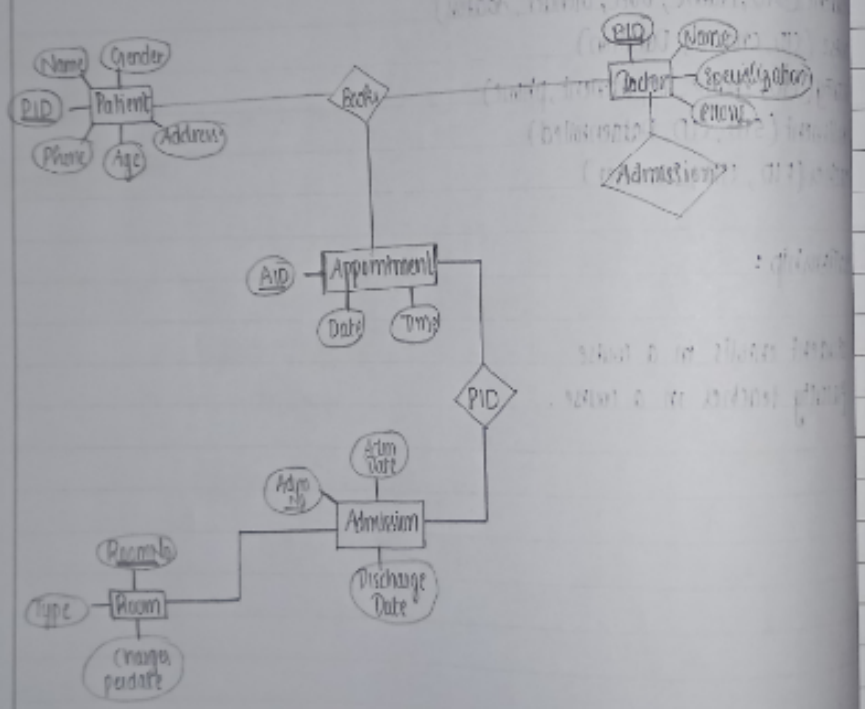
Entities and Attributes

Student (SID, Name, DOB, Gender, Address)
Course (CID, Cname, Duration)
Faculty (FID, Fname, Department, phone)
Enrollment (SID, CID, Datenrolled)
Teaches (FID, CID, Semester)

Relationship :

A student enrolls in a course
A faculty teaches in a course.

Output:



3 / 7 / 25

2. ER Diagram of Hospital Management System

Entities and Attributes

Patient (PID, Name, Age, Gender, Address, Phone)
Doctor (DID, Name, Specialization, Phone)
Appointment (AID, Date, Time, PID, DID)
Room (Room No, Type, Charges Per Day)
Admission (PID, Room No, Admission Date, Discharge Date)

Relationship:

A patient book an appointment with a doctor
A patient gets admitted to a room

3. Reduce the ER Diagram in Q1 [Registar Office] into Tables

create table Students (SID int(5) primarykey, Name varchar (100), DOB date, Gender char(1), Address Varchar (100));

Create table course (CID int(5) primarykey, cname varchar (100), Duration varchar (10));

Create table faculty (FID int(5) primarykey, Fname varchar (100), Department varchar (100), phno Numeric (10));

Create table enrollment (SID int(5) references student (SID), CID int (5) references course (CID), Date enrolled Date);

4 Reduction of Hospital management System ER diagram into tables

Create table patient (PID int(5) primarykey Name varchar(25), Age int(3), Gender char(1), Address varchar(50), phone numeric(10));

Create table Doctor (DID int(5) primarykey, Name varchar(25), Specialization varchar(50), phone numeric(10));

Create table appointment (AID int(5) primarykey, Date Date, time time, PID int(5) references patient(PID), DID int(5), references Doctor(DID);

Create table Room (Room int(5) primarykey, Type varchar(50), charges Per Day decimal(10,2));

Create table Admission (Admission No (5) primarykey PID int(5) references patient (PID), RoomNo int(5) references Room (Room No Admission Date date, discharge date));

II DDL Commands

Consider the following Schema

Supplier (SID, Sname, branch, city, phone)
Part (PID, Pname, color, price)
Supplies (SID, PID, qty, date_supplied)

1. Create the above tables

Query: create table supplier (SID numeric (10), sname varchar (10), city varchar (10), phone varchar (10));

Output: Query OK, 0 rows affected (0.01 sec)

Query: Create table part (PID numeric (10), pname varchar (10), color varchar (10), price numeric (8));

Output: Query OK, 0 rows affected (0.01 sec)

Query: Create table supplies (SID numeric (10), PID numeric (10), qty numeric (10), sdate date);

Output: Query OK, 0 rows affected (0.01 sec)

2. Add a new attribute state in supplier table.

Query: alter table supplier add state varchar (10);
Output: Query OK, 0 rows affected (0.01 sec)

3. Remove attribute city from supplier table

Query: alter table supplier drop column address;
Output: Query OK, 0 rows affected (0.01sec)

4. Modify the data type of phone attribute

Query: alter table supplier modify phone numeric (10);
Output: Query OK, 0 rows affected (0.01sec)

5. Change the name of attribute city to address.

Query: alter table supplier change city address varchar(15);
Output: Query OK, 0 rows affected (0.01sec)

6. Change a table's name, supplier to sup.

Query: alter table supplier rename to sup;
Output: Query OK, 0 rows affected (0.00 sec)

7. Use truncate to delete the contents of supplies table

Query: truncate table supplies;
Output: Query OK, 0 rows affected (0.00sec)

8. Remove the part table from database

Query : drop table part;
output : Query ok, 0 rows affected (0.00sec)

III DML Commands

1. Insert atleast 5 records into table supplier, part & supplier.

Query1: Insert into sup (SID, sname, branch, Address, phone, state) values (204001, 'Raj Traders', 'Grocery', 'Delhi', 9876547891, 'Delhi'), (204002, 'Meera stores', 'Sales', 'Delhi', 3291001923, 'Delhi'), (204003, 'Global Supplies', 'Stationary', 'UK', 3465225643, 'UK'), (204004, 'Vandana', 'Hardware', 'Gujarat', 321320652, 'Gujarat'), (204005, 'Fastools', 'Hardware', 'India', 5621000012, 'India');

Query2: insert into part (PID, pname, color, price) values (102100, 'Tshirt', 'Blue', 499), (102101, 'Laptop', 'Silver', 1000), (102102, 'Headphones', 'Black', 200), (102103, 'Bag', 'Green', 100), (102104, 'Headgear', 'Black', 250);

Query3: insert into supplies (SID, PID, qty, date supplied) values (204001, 102100, 2, 2024-01-02), (204002, 102101, 4, 2024-01-03), (204003, 102102, 6, 2024-01-04), (204004, 102103, 8, 2024-01-05), (204005, 102104, 9, 2024-01-07);

Output:

Query 1: Query OK, 5 row affected
Query 2: Query OK, 5 row affected
Query 3: Query OK, 5 row affected.

2. Show the content in table supplier, part, supplies

Select * from sup;

Output:

SID	Sname	Branch	Address	Phone no	State
204001	Raj Traders	Grocery	Delhi	9876547891	Delhi
204002	Meera Stores	Sales	Delhi	3291001923	Delhi
204003	Global Supplies	Stationary	UK	3463225643	UK
204004	Vandana	Hardware	Gujarat	9213206521	Gujarat
204005	Fastools	Hardware	India	5621000012	India

5 rows in set

Select * from part ;

Output:

PID	Pname	Color	Price
102100	Tshirt	Blue	499
102101	Laptop	Silver	1000
102102	Headphones	Black	200
102103	Bag	Green	100
102104	Headgear	Black	250

5 rows in set

Select * from supplies;

SID	PID	qty	Date Supplied
204001	102100	2	2024-01-02
204002	102101	4	2024-01-03
204003	102102	6	2024-01-04
204004	102103	8	2024-01-05
204005	102104	9	2024-01-07

5 rows in set

3. Find the name of city of all suppliers

Select sname, Address from sup;

Output:

Sname	Address
Raj Traders	Delhi
Meerastores	Delhi
Global Suppliers	Chennai
VR Industries	UK
Vandana	India

5 rows in set

4. Find the name and phone no of all suppliers whose stay in Delhi

Select sname, phone from sup where Address = 'Delhi';

Output:

sname	phone
Raj Traders	9876547891
Meera Stores	3291001923

5 Find all distinct branches of suppliers.

Select distinct Branch from sup;

Output:

Branch
Grocery
Sales
Stationary
Hardware
Hardware

5 rows in set

6. Delete the record of the supplier whose SID is 204001

Delete from sup where SID = 204001;

Output:
Query OK, 1 row affected

7. Delete all records of suppliers table
delete from sup;

Output:
Query OK, 5 rows affected

8. Delete all records of suppliers whose city start with A.

delete from sup where address like 'A%';

Output:
Query OK, 1 rows affected.

9. find the supplier name which have 'lk%' in any position

Select sname from sup where sname like '%lk%';

Output:

Sname
Alka Stores

1 row in set

10. find the supplier name where 'R' is in the second position.

Select sname from sup where sname like '- B%';

Output:

Sname
VB Industries

1 row in set

11. Change the city of all suppliers to Bombay

Update sup set Address = 'Bombay';

Output:
Query OK, 10 rows affected
Rows matched: 2 Changed: 2 Warnings: 0

12. Change the city of supplier Vandana to Goa

update subset Address = 'Goa' where sname = 'Vandana';

Output:
Query OK, 1 row affected
Rows matched: 1 Changed: 1 Warnings: 0

IV Queries with constraints

1. Create the supplier table with

a) primarykey constraint
b) check constraint
c) default constraint

create table supplier (SID numeric (8) primarykey, sname varchar(20) check (sname like '%A%'), branch varchar(15), city varchar(20) default 'Bombay', phone numeric (10));

output : Query OK, 0 rows affected

2. Create suppliestable with foreignkey constraint

create table supplies (SID numeric(8) references supplies, PID numeric (8), qty numeric (8), Date_supplied date);

output: Query OK, 0 rows affected.

3. Create a parttable with unique constraint

create table part (PID numeric (10) unique, pname varchar(15), color varchar (15), price numeric (8));

output: Query OK, 0 rows affected

V TCL Commands

1. Create Savepoint
2. Rollback to Savepoint
3. Use commit to save on

create table employee (emp num numeric (8), empname varchar (15));
Output: Query ok, 0 rows affected

alter table employee engine = innodb;
Output: Query OK, 0 rows affected
Records: 0 Duplicates: 0 Warnings: 0

Start transaction;
Output: Query ok, 0 rows affected

insert into employee (empnum, empname) values (102201, 'John'), (102202, 'James')
(102203, 'Andrew'), (102204, 'Clara')

Output: Query OK, 0 rows affected

Savepoint A;
Output: Query OK, 0 rows affected

Select * from employee;

output:

empnum	empname
102201	John
102202	James
102203	Andrew
102204	Clara

4 rows in set

delete from employee where empnum = 102202;
output:
Query ok, 1 row affected

select * from employee

output:

empnum	empname
102201	John
102203	Andrew
102204	Clara

3 rows in set

rollback to savepoint A;
Output:
Query ok, 0 rows affected

Select * from employee;

Output:

empnum	empname
102201	John
102202	James
102203	Andrew
102204	Clara

4 rows in set

Commit;

Output: Query ok, 0 rows affected

VI Aggrigate functions

1) find the minimum, maximum, average, sum of cost of parts

select min(price), max(price), avg(price), sum(price) from part;

Output:

min(price)	max(price)	avg(price)	sum(price)
350	600	434.00	2170

1 row in set (0.01 sec)

2) count the total number of parts present

Select count(*) from part;

output:

count(*)

5

1 row in set (0.01 sec)

3) Retrieve the average cost of all parts supplied by 'Mike'

Select avg(p.price) from supplies s joint part p on s.pid = p.pid join supplier sup on s.sid = sup.sid where sup.sname = 'Mike';

Output

avg(p.price)
3000.0000

1 row in set (0.02 sec)

VII Queries on GROUP BY, HAVING AND ORDER BY CLAUSES

1. Display total price of parts of each color

Select color, sum(price) as total_price from part group by color;

output:

color	total_price
blue	4000
green	3000
white	1000
yellow	4000

4 rows in set (0.01 sec)

2. Find the branch and the no of suppliers in that branches which have more than 2 supplier.

Select branch, count(*) as supplier_count from supplier group by branch having count(*) > 2;

output:

Branch	supplier_count
abc	3

3. Find all parts sorted by pname in Ascending order and cost in descending order

Select * from part order by pname asc, price desc;

Output:

PID	pname	color	price
1003	Akash	red	67
1000	Arun	blue	50
1001	Varun	green	90

4. find the branch and the no of suppliers in that branch

Select branch, count(*) as supplier_count from supplier group by branch;

output:

Branch	supplier_count
abc	3

VIII Queries on Operators

1. Find the pname, phoneno and cost of parts which have cost equal to or greater than 200 and less than or equal to 600

Select pname, phone, price from part where price >= 200 and price <= 600;

Output:

Pname	phone	price
Magnet	912350672	300
Nut	2135206723	350
Bolt	1234567891	250
Nail	1011295976 2	200

2. Find the sname, sid and branch of suppliers who are in 'local' branch or 'global' branch

Select sname, sid, branch from supplier where branch in ('local', 'global')

Output:

sname	sid	branch
Mike	101	local
Kichu	102	global
Leo	103	local
Don	104	global
Lia	105	local

3. find the pname, pno and cost of parts for which cost in between 200 and 600

select pname, pno, price from part where price between 200 and 600;

Output:

pname	phone	price
Magnet	9503201	300
Nut	1243567890	350
Bolt	1342562501	250
Nail	3509173321	200

4. find pname and color of parts which has the word 'NET' anywhere in in pname

Select pname, color from part where pname like '%NET%'

output

pname	color
MAGNET	red

5. find the pid and pname of parts with pname either 'NUT' or 'BOLT'

Select pid, pname from part where pname in ('NUT','BOLT')

Output:

pid	pname
202	NUT
203	BOLT

6. List the suppliers who supplied parts on '1st May 2000', '12 Jan 2021', '17 Dec 2020', '10 Jan 2021'.

```
Select * from supplies where date_supplied in
   ( TO_DATE ('01-May-2000', 'DD-MON-YYYY')
     TO_DATE ('12-Jan-2021', 'DD-MON-YYYY')
     TO_DATE ('17-Jan-2020', 'DD-MON-YYYY')
     TO_DATE ('10-Jan-2021', 'DD-MON-YYYY')
   );
```

Output:

SID	sname	pid	date_supplied
101	Mike	201	01-May-2000
102	Kichu	202	12-Jan-2021
103	Ted	203	17-Jan-2020
104	Don	204	10 Jan 2021

7. Find all the distinct costs of parts

```
Select distinct price from part;
```

Output:

price
300
350
250
200

IX Join operators

Create table customers and orders

a) create table customer (cid int primary key, name varchar (50));
Output: Query ok, 0 rows affected (0.01 sec)

b) create table order (oid int primary key, cid int, product varchar (50), foreign key(cid) references customer (cid));
Output: Query ok, 0 rows affected (0.01sec)

1. Perform innerjoin of 2 tables

Select customer.cid, customer.name, order.oid, order.product from customer inner join order on customer.cid = order.cid;
Output:

cid	name	oid	product
1	Alice	101	Pen
1	Alice	102	Notebook
2	Bob	103	Pencil

2. Perform Natural join on 2 tables

Select * from customer Natural join "order";

output :

cid	cname	oid	product
1	Alice	101	Pen
1	Alice	102	Notebook
2	Bob	103	Pencil

3. Perform leftouterjoin on table

Select customer.cid , customer.name , "order". oid , "order". product from customer left outerjoin "order" on customer . cid = "order". cid ;

Output :

cid	cname	oid	product
1	Alice	101	Pen
1	Alice	102	Notebook
2	Bob	103	Pencil
3	Charlie	NULL	NULL

4. Perform rightouter join on tables

Select customer . cid , customer . name , "order". oid , "order". product from customer right outer join " order " on customer . cid = "order". cid ;

output

cid	cname	oid	product
1	Alice	101	Pen
1	Alice	102	Notebook
2	Bob	103	Pencil

5. Perform full outerjoin on tables

```
Select customer.cid, customer.name, "order".oid, "order".product from
customer full outerjoin "order" on customer.cid = "order".cid;
```

output:

cid	cname	oid	product
1	Alice	101	Pen
1	Alice	102	Notebook
2	Bob	103	Pencil
3	Charlie	NULL	NULL

X Set theory operators

1. Show the use of union operator with union compatibility

Select sname from supplier union select pname from part;

Output:

sname
Raj Traders
Metro Stores
Global Supplies
Vandana
Fastools
Tshirt
laptop
Headphones
Bag
Headgear

2. Find the cartesian product of 2 tables

Select * from supplier, part;

Output :

SID	Sname	Branch	Address	Phno	State	PID	pname	color	price
204001	Raj Traders	Grocery	Delhi	9816647891	Delhi	102100	Tshirt	blue	499
204001	Raj Traders	Grocery	Delhi	9816647891	Delhi	102100	Laptop ~~Tshirt~~	black	4990
204001	Raj Traders	Grocery	Delhi	9816647891	Delhi	102100	Headphones ~~Tshirt~~	blue	4991
204002	Metro Stores	Sales	Delhi	2391061923	Delhi	102101	Tshirt	blue	499
204002	Metro Stores	Sales	Delhi	2391061923	Delhi	102101	Laptop	black	4990
204002	Metro Stores	Sales	Delhi	2391061923	Delhi	102101	Headphones	blue	4991
204003	Global S.	Stationery	UK	346323102	UK	102102	Tshirt	blue	499
204003	Global S.	Stationery	UK	346323102	UK	102102	Laptop	black	4990
204003	Global S.	Stationery	UK	346323102	UK	102102	Headphone	blue	4991

9 rows in set

XI Queries on Set theory operators

1. List all parts except 'shoes' and 'bag' in asc order of costs

Select * from part where pname NOT IN ('shoes','bag') order by price ASC;

Output:

PID	Pname	color	price
4444	Toys	yellow	1199
2222	Laptop	black	65000

2. Displays all parts that have been supplier so far

Select * from part where pid not in (select pid from supplies);

Output:

PID	pname	color	price
5555	Notebooks	White	666
3333	Tools	Grey	9999

2 rows in set

3. To display supplier names who have supplied 'green' part with cost 500/- and 'red' part with cost 400/- rupees.

Select distinct s.sname from suppliers s, supplies sup, part p where s.sid = sup.sid and sup.pid = p.pid and ((p.color = 'green' and p.price = 500) or (p.color = 'red' and p.price = 400));

output

sname
Raj Traders

4. To display the name of suppliers who have supplied all parts that are red in color.

Select sname from supplier s where not exists (select * from part P where color = 'Black' and not exists (select * from suppliers where supplier. PID = P. PID and suppliers. SID = S. SID));

output :

sname
Sharath

XII MongoDB Queries

1. create a collection and insert documents into it using insertOne() and insertMany()

```
db.supplier.insertOne({
  sid:"204001"
  sname: "Raj Traders",
  branch: "Grocery",
  address : "Delhi",
  phno : "9816647891",
  state : "delhi"
})
db.part.insertMany([
  {pid:"102100", pname:"Tshirt", color:"Blue", price: 499},
  {pid:"102101", pname:"Laptop", color:"silver", price:1000},
  {pid:"102102", pname:"Headphones", color:"Black", price:900},
  {pid:"102103", pname:"Bag", color:"Green", price:100},
  {pid:"102104", pname:"Headgear", color:"Black", price:850}
])
```

Output:

```
{
  acknowledged : true,
  insertedId : ObjectId("507f1f77bcf86cd799439011")
}
{
  acknowledged : true,
  insertIds : {
```

```
'0' : ObjectId ("5a7f191e810c19729de860ea"),
'1' : ObjectId ("5a7f191e810c19729de860eb"),
'2' : ObjectId ("5a7f191e810c19729de860ec"),
'3' : ObjectId ("5a7f191e810c19729de860ed"),
'4' : ObjectId ("5a7f191e810c19729de860ee")
}
}
```

2. Select all documents in collection

```
db.supplier.find()
Output :
[
 {
   _id : ObjectId("...."),
   sid : "204001",
   sname : "Raj Traders",
   branch : "Grocery",
   address : "Delhi",
   phno : 9816647891,
   state : "Delhi"
 },
 {
   _id : ObjectId("...."),
   sid : "204002",
   sname : "Metro Stores",
   branch : "Sales",
   address : "Delhi",
```

```
phno : "2391061923",
state : "Delhi"
},
{
 _id : ObjectId ("... ")
 sid : "204003",
 sname : "Global Supplies",
 branch : "Stationary",
 address : "UK",
 phno : "34632335643",
 state : "UK"
 }
]
```

3. Find the count of all suppliers

```
db.supplier.countDocuments()
```

Output :

5

4. Find all records that have city = 'Delhi'

```
db.supplier.find({address : "Delhi"})
```

Output :

```
[
 {
```

```
-id : ObjectId("..."),
sid : "204001",
sname : "Raj Traders",
branch : "Grocery",
address : "Delhi",
phno : "9816647891",
state : "Delhi"
},
{
-id : ObjectId("...."),
sid : "204002",
sname : "Metro Stores",
branch : "Sales",
address : "Delhi",
phno : "2391061923",
state : "Delhi"
}
]
```

5. Retrieve all document that have color equal to 'red' or 'green'

```
db.part.find({color : { $in : ["red", "green"]}})
```

output :

```
[
 {
  -id : ObjectId("...")
  pid : "102103"
  pname : "Bag"
```

```
color : "green"
price : 100
}
]
```

6. Retrieve all documents where part_name is 'FI' or price is less than 200

```
db.part.find({
 $or : [
    {pname : "FI"},
    {price : {$lt : 200}}
  ]
})
```

Output:

```
[
 {
  _id : ObjectId("...")
  pid : "102103"
  pname : "Bag"
  color : "green"
  price : 100
 }
]
```

7. Update the record of 'Geeta', set city = 'Bombay' and phno = "11223344"

```
db.supplier.updateOne(
  {sname : "Geeta},
  {$set : {address : "Bombay", phno: 1122334455}}
)
```

Output:

```
{
  acknowledged: true,
  matchcount: 1,
  modifiedcount: 1
}
```

8. Delete all records where price is greater than 5000

```
db.part.deleteMany({price : {$gt : 5000}})
```

Output:

```
{
 acknowledged : true
 deleteCount : 0
}
```

9. Display only the name and city of the supplier

```
db.supplier.find({}, {sname:1, Address:1, _id:0})
```

Output:

```
{"sname": "Raj Traders", "Address" : "Delhi"}
{"sname": "Meera Stores", "Address": "Delhi"}
```

{"sname": "Global Suppliers", "Address": "UK"}
{"sname": "Vandana", "Address": "Gujarat"}
{"sname": "Fastools", "Address": "India"}

10. Sort all suppliers on city and display only the first 2 records

db.suppliers.find().sort({Address:1}).limit(2)

Output:

```
{
"_id": ('...'),
"sname": "Raj Traders",
"Branch": "Grocery",
"Address": "Delhi",
"Phno": 9876547891
"State": "Delhi"
}
{
"_id": ("...")
"sname": "Meera Sares"
"Branch": "Saree"
"Address": "Delhi"
"Phno": 2391001925
"State": "Delhi"
}
```