BRANCH (Branchid, Branchname, HOD)

```
CREATE TABLE BRANCH (
   Branchid INT PRIMARY KEY,
   Branchname CHAR(15),
   HOD CHAR(15)
);

INSERT INTO BRANCH VALUES (1, 'MCA','Dr. RAJU');
INSERT INTO BRANCH VALUES (2, 'CS','Prof.PRAVEEN');
INSERT INTO BRANCH VALUES (3, 'EC','Dr. KRUPA R');
INSERT INTO BRANCH VALUES (4, 'MCA','Dr. KSN');
INSERT INTO BRANCH VALUES (5, 'CV','Prof. Raj');

SELECT * FROM BRANCH;
```

Branchid	Branchname	HOD
1	MCA	Dr. RAJU
2	CS	Prof.PRAVEEN
3	EC	Dr. KRUPA R
4	MCA	Dr. KSN
5	CV	Prof. Raj

STUDENT (USN, Name, Address, Branchid, sem)

```
CREATE TABLE STUDENT (
   USN VARCHAR(10) PRIMARY KEY,
   Name CHAR(15),
   Address VARCHAR(15),
   Branchid INT,
   SEM INT,
   FOREIGN KEY (Branchid) REFERENCES BRANCH (Branchid)
);
```

```
INSERT INTO STUDENT VALUES ('2JI20MC001', 'AKASH','KHANAPUR', 1, 2);
INSERT INTO STUDENT VALUES ('2JI20MC002', 'VINAY','BELAGAVI', 4, 2);
INSERT INTO STUDENT VALUES ('2JI20CS005', 'VIRAT','MUMBAI', 2, 6);
INSERT INTO STUDENT VALUES ('2JI20CV045', 'RAHUL','PUNE', 5, 7);
INSERT INTO STUDENT VALUES ('2JI19MC003','RAVI','HUBBALI', 2, 6);

SELECT * FROM STUDENT;
```

	USN	Name	Address	Branchid	sem
1	2JI19MC003	RAVI	HUBBALI	2	6
2	2JI20CS005	VIRAT	MUMBAI	2	6
3	2JI20CV045	RAHUL	PUNE	5	7
4	2JI20MC001	AKASH	KHANAPUR	1	2
5	2JI20MC002	VINAY	BELAGAVI	4	2

AUTHOR (Authorid, Authorname, Country, age)

```
CREATE TABLE AUTHOR (
   Authorid INT PRIMARY KEY,
   Authorname CHAR(15),
   Country CHAR(10),
   Age INT
);

INSERT INTO AUTHOR VALUES (444,'RAMKRISHNAN','INDIA', 56);
INSERT INTO AUTHOR VALUES (555,'ELMASRI','US', 75);
INSERT INTO AUTHOR VALUES (666,'HERBERT','CANADA', 60);
INSERT INTO AUTHOR VALUES (777,'KOTUR','INDIA', 58);
```

SELECT * FROM AUTHOR;

	Authorid	Authorname	Country	age
1	444	RAMKRISHNAN	INDIA	56
2	555	ELMASRI	US	75
3	666	HERBERT	CANADA	60
4	777	KOTUR	INDIA	58

BOOK (Bookid, Bookname, Authorid, Publisher, Branchid)

```
CREATE TABLE BOOK (
   Bookid VARCHAR(5) PRIMARY KEY,
   Bookname CHAR(15),
   Authorid INT,
   Publisher CHAR(15),
   Branchid INT,
   FOREIGN KEY (Authorid) REFERENCES AUTHOR (Authorid),
   FOREIGN KEY (Branchid) REFERENCES BRANCH (Branchid)
);
```

```
INSERT INTO BOOK VALUES ('B01','JAVA', 666,'TATA MCGRAW', 1);
INSERT INTO BOOK VALUES ('B02','DBMC', 555,'TATA MCGRAW', 1);
INSERT INTO BOOK VALUES ('B03','DBMS', 555,'TATA MCGRAW', 2);
INSERT INTO BOOK VALUES ('B04','C++', 777,'PH', 4);
INSERT INTO BOOK VALUES ('B05','CONSTRUCT', 444,'PH', 5);
```

SELECT * FROM BOOK;

	Bookid	Bookname	Authorid	Publisher	Branchid
1	B01	JAVA	666	TATA MCGRAW	1
2	B02	DBMC	555	TATA MCGRAW	1
3	B03	DBMS	555	TATA MCGRAW	2
4	B04	C++	777	PH	4
5	B05	CONSTRUCT	444	PH	5

BORROW (USN, Bookid, Borrowed_Date)

```
CREATE TABLE BORROW (
   USN VARCHAR(10),
   Bookid VARCHAR(5),
   Borrowdate DATE,
   FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (Bookid) REFERENCES BOOK (Bookid));
INSERT INTO BORROW VALUES ('2JI20MC001','B01','30-JAN-2021');
INSERT INTO BORROW VALUES ('2JI20MC001','B02','30-JAN-2021');
INSERT INTO BORROW VALUES ('2JI20MC002','B03','5-FEB-2020');
INSERT INTO BORROW VALUES ('2JI20CV045','B04','22-DEC-2019');
INSERT INTO BORROW VALUES ('2JI20CS005','B05','12-OCT-2019');
INSERT INTO BORROW VALUES ('2JI20MC001','B05','01-JUL-2020');

SELECT * FROM BORROW;
```

	USN	Bookid	Borrow_date
1	2JI20MC001	B01	2021-01-30
2	2JI20MC001	B02	2021-01-30
3	2JI20MC002	B03	2020-02-05
4	2JI20CV045	B04	2019-12-22
5	2JI20CS005	B05	2019-10-12
6	2JI20MC001	B05	2020-07-01

QUERY 1: List the details of Students who are all studying in 2nd sem MCA.

```
SELECT *
FROM STUDENT S, BRANCH B
WHERE S.Branchid = B.Branchid
AND S.sem = 2
AND B.Branchname = 'MCA';
```

	USN	Name	Address	Branchid	sem	Branchid	Branchname	HOD
1	2JI20MC001	AKASH	KHANAPUR	1	2	1	MCA	Dr. RAJU
2	2JI20MC002	VINAY	BELAGAVI	4	2	4	MCA	Dr. KSN

QUERY 2: List the students who are not borrowed any books

```
SELECT *
FROM STUDENT S
WHERE S.USN NOT IN (SELECT B.USN FROM BORROW B);
```

	USN	Name	Address	Branchid	sem
1	2JI19MC003	RAVI	HUBBALI	2	6

QUERY 3: Display the USN, Student name, Branch_name, Book_name, Author_name,

```
SELECT S.USN, S.NAME, BR.Branchname, BK.Bookname, A.Authorname, BW.Borrowdate
FROM STUDENT S, BRANCH BR, BOOK BK, AUTHOR A, BORROW BW
WHERE S.Branchid = BR.Branchid
AND S.Branchid = BK.Branchid
AND A.Authorid = BK.Authorid
AND BW.USN = S.USN
AND BK.Bookid = BW.Bookid
AND S.sem = 2
AND BR.Branchname = 'MCA';
```

	USN	NAME	Branchname	Bookname	Authorname	Borrow_date
1	2JI20MC001	AKASH	MCA	JAVA	HERBERT	2021-01-30
2	2JI20MC001	AKASH	MCA	DBMC	ELMASRI	2021-01-30

QUERY 4: Display the number of books written by each Author.

```
SELECT A.Authorid, A.Authorname, COUNT(DISTINCT BK.Bookid) AS No_of_Books
FROM AUTHOR A, BOOK BK
WHERE A.Authorid = BK.Authorid
GROUP BY A.Authorid, A.Authorname;
```

	Authorid	Authorname	No_of_Books
1	444	RAMKRISHNAN	1
2	555	ELMASRI	2
3	666	HERBERT	1
4	777	KOTUR	1

QUERY 5: Display the student details who borrowed more than two books.

```
SELECT S.USN, S.NAME
FROM STUDENT S, BORROW BW
WHERE S.USN = BW.USN
GROUP BY S.USN, S.NAME
HAVING COUNT(BW.BOOKID) > 2;
```

	USN	NAME
1	2JI20MC001	AKASH

QUERY 6: Display the student details who borrowed books of more than one Author.

```
SELECT S.USN, S.NAME, COUNT(BK.AUTHORID) AS NO_OF_AUTHOR
FROM STUDENT S, BORROW BW, BOOK BK
WHERE S.USN = BW.USN
AND BK.Bookid = BW.Bookid
GROUP BY S.USN, S.NAME
HAVING COUNT(BK.Authorid) > 1;
```

	USN	NAME	NO_OF_AUTHOR
1	2JI20MC001	AKASH	3

QUERY 7: Display the Book names in descending order of their names.

```
SELECT BOOKNAME FROM BOOK
ORDER BY Bookname DESC;
```

	BOOKNAME
1	JAVA
2	DBMS
3	DBMC
4	CONSTRUCT
5	C++

QUERY 8: List the details of students who borrowed the books which are all published by the same publisher.

```
SELECT S.USN, S.NAME, COUNT(BK.PUBLISHER) AS SAME_PUBLISHER
FROM STUDENT S, BOOK BK, BORROW BW
WHERE S.USN = BW.USN
AND BK.Bookid = BW.Bookid
GROUP BY S.USN, S.NAME
HAVING COUNT(BK.Publisher) > 1;
```

	USN	NAME	SAME_PUBLISHER
1	2JI20MC001	AKASH	3

Create the Student Table

```
CREATE TABLE STUDENT_LAB2 (
   USN VARCHAR(10) PRIMARY KEY,
   Name CHAR(15) NOT NULL,
   Date_of_birth DATE NULL,
   Branch CHAR(15) NOT NULL,
   Marks1 INTEGER NOT NULL,
   Marks2 INTEGER NOT NULL,
   Marks3 INTEGER NOT NULL,
   Total INTEGER,
   GPA REAL
);
```

```
INSERT INTO STUDENT_LAB2 VALUES ('2JI150MC01', 'GAURAV', '12-JUL-1999', 'MCA', 75, 86, 72, NULL, NULL);

INSERT INTO STUDENT_LAB2 VALUES ('2JI150CC10', 'JAI', '02-JAN-1998', 'COMM', 75, 86, 70, NULL, NULL);

INSERT INTO STUDENT_LAB2 VALUES ('2JI150SC09', 'DEEPAK', '12-DEC-1999', 'SCIE', 72, 86, 72, NULL, NULL);

INSERT INTO STUDENT_LAB2 VALUES ('2JI180MC01', 'ABHI', '27-FEB-1996', 'ARTS', 80, 86, 72, NULL, NULL);

INSERT INTO STUDENT_LAB2 VALUES ('2JI150MC45', 'SAI', '15-AUG-1999', 'MCA', 75, 86, 50, NULL, NULL);

INSERT INTO STUDENT_LAB2 VALUES ('1001', 'SAI', '15-AUG-1999', 'MCA', 75, 86, 50, NULL, NULL);
```

```
SELECT * FROM STUDENT_LAB2;
```

	USN	Name	Date_of_birth	Branch	Marks1	Marks2	Marks3	Total	GPA
1	1001	SAI	1999-08-15	MCA	75	86	50		
2	2JI150MC01	GAURAV	1999-07-12	MCA	75	86	72		
3	2JI150MC45	SAI	1999-08-15	MCA	75	86	50		
4	2JI150MC50	TENDULKAR	1970-04-25	MCA	75	86	70		
5	2JI150SC09	DEEPAK	1999-12-12	SCIENCE	72	86	72		
6	2JI180MC01	ABHI	1996-02-27	ARTS	80	86	72		

QUERY 1: Update the column total by adding the columns mark1, mark2, mark3.

```
UPDATE STUDENT_LAB2
SET Total = Marks1 + Marks2 + Marks3;
```

Output:

(6 row(s) affected)

```
SELECT * FROM STUDENT_LAB2;
```

	USN	Name	Date_of_birth	Branch	Marks1	Marks2	Marks3	Total	GPA
1	1001	SAI	1999-08-15	MCA	75	86	50	211	NULL
2	2JI150MC01	GAURAV	1999-07-12	MCA	75	86	72	233	
3	2JI150MC45	SAI	1999-08-15	MCA	75	86	50	211	
4	2JI150MC50	TENDULKAR	1970-04-25	MCA	75	86	70	231	
5	2JI150SC09	DEEPAK	1999-12-12	SCIENCE	72	86	72	230	
6	2JI180MC01	ABHI	1996-02-27	ARTS	80	86	72	238	

Query 2 :Find the GPA score of all the students.

```
UPDATE STUDENT_LAB2
SET GPA = (Total / 3.0);
```

Output:

(6 row(s) affected)

```
SELECT * FROM STUDENT_LAB2;
```

	USN	Name	Date_of_birth	Branch	Marks1	Marks2	Marks3	Total	GPA
1	1001	SAI	1999-08-15	MCA	75	86	50	211	70.33334
2	2JI150MC01	GAURAV	1999-07-12	MCA	75	86	72	233	77.66666
3	2JI150MC45	SAI	1999-08-15	MCA	75	86	50	211	70.33334
4	2JI150MC50	TENDULKAR	1970-04-25	MCA	75	86	70	231	77
5	2JI150SC09	DEEPAK	1999-12-12	SCIENCE	72	86	72	230	76.66666
6	2JI180MC01	ABHI	1996-02-27	ARTS	80	86	72	238	79.33334

QUERY 3: Find the students who born on a particular year of birth from the date_of_birth column.

```
SELECT USN, NAME, DATE_OF_BIRTH
FROM STUDENT_LAB2
WHERE Date_of_birth LIKE '1999%';
```

	USN	NAME	DATE_OF_BIRTH
1	1001	SAI	1999-08-15
2	2JI150MC01	GAURAV	1999-07-12
3	2JI150MC45	SAI	1999-08-15
4	2JI150SC09	DEEPAK	1999-12-12

Query 4: List the students who are studying in a particular branch of study.

```
SELECT USN, NAME, BRANCH
FROM STUDENT_LAB2
WHERE BRANCH = 'MCA';
```

	USN	NAME	BRANCH
1	1001	SAI	MCA
2	2JI150MC01	GAURAV	MCA
3	2JI150MC45	SAI	MCA
4	2JI150MC50	TENDULKAR	MCA

QUERY 5: Find the maximum GPA score of the student branch-wise.

```
SELECT USN, BRANCH, MAX(GPA) AS MAX_GPA
FROM STUDENT_LAB2
GROUP BY USN, BRANCH
ORDER BY MAX_GPA DESC;
```

	USN	BRANCH	MAX_GPA
1	2JI180MC01	ARTS	79.33334
2	2JI150MC01	MCA	77.66666
3	2JI150MC50	MCA	77
4	2JI150SC09	SCIENCE	76.66666
5	2JI150MC45	MCA	70.33334
6	1001	MCA	70.33334

QUERY 6: Find the students whose name starts with the alphabet “S”.

```
SELECT USN, NAME
FROM STUDENT_LAB2
WHERE Name LIKE 'S%';
```

	USN	NAME
1	1001	SAI
2	2JI150MC45	SAI

QUERY 7: Find the students whose name ends with the alphabets "AR".

```
SELECT USN, NAME
FROM STUDENT_LAB2
WHERE NAME LIKE '%AR';
```

	USN	NAME
1	2JI150MC50	TENDULKAR

QUERY 8: Delete the student details whose USN is given as 1001.

```
DELETE FROM STUDENT_LAB2
WHERE USN = '1001';
```

Output:

(1 row(s) affected)

	USN	Name	Date_of_birth	Branch	Marks1	Marks2	Marks3	Total	GPA
1	2JI150MC01	GAURAV	1999-07-12	MCA	75	86	72	233	77.66666
2	2JI150MC45	SAI	1999-08-15	MCA	75	86	50	211	70.33334
3	2JI150MC50	TENDULKAR	1970-04-25	MCA	75	86	70	231	77
4	2JI150SC09	DEEPAK	1999-12-12	SCIENCE	72	86	72	230	76.66666
5	2JI180MC01	ABHI	1996-02-27	ARTS	80	86	72	238	79.33334

TEAM (Teamid, Teamname, Coach, City)

```
CREATE TABLE TEAM (
   Teamid VARCHAR(4) PRIMARY KEY,
   Teamname CHAR(15) NOT NULL,
   Coach CHAR(15) NOT NULL,
   City CHAR(15)
);

INSERT INTO TEAM VALUES ('T01', 'RCB', 'RAY JENNINGS', 'BANGALORE');
INSERT INTO TEAM VALUES ('T02', 'MI', 'MAHELA J', 'MUMBAI');
INSERT INTO TEAM VALUES ('T03', 'KKR', 'McCULLUM', 'KOLKATA');
INSERT INTO TEAM VALUES ('T04', 'SRH', 'TOM MOODY', 'HYDERABAD');
INSERT INTO TEAM VALUES ('T05', 'CSK', 'STEPHEN F', 'CHENNAI');

SELECT * FROM TEAM;
```

	Teamid	Teamname	Coach	City
1	T01	RCB	RAY JENNINGS	BANGALORE
2	T02	MI	MAHELA J	MUMBAI
3	T03	KKR	McCULLUM	KOLKATA
4	T04	SRH	TOM MOODY	HYDERABAD
5	T05	CSK	STEPHEN F	CHENNAI

PLAYER (Playerid, Playername, Phone, Age, Teamid)

```
CREATE TABLE PLAYER (
   Playerid VARCHAR(4) PRIMARY KEY,
   Playername CHAR(15) NOT NULL,
   Phone INT NOT NULL,
   Age INT NOT NULL,
   Teamid VARCHAR(4),
   FOREIGN KEY (Teamid) REFERENCES TEAM (Teamid)
);

INSERT INTO PLAYER VALUES ('P01', 'VIRAT', 65412, 28, 'T01');

SELECT * FROM PLAYER;
```

	Playerid	Playername	Phone	Age	Teamid
1	P10	HARDIK	14522	22	T03
2	P11	BHUVI	45632	34	T04
3	P12	SEKHAR	32112	34	T04
4	P13	JADEJA	78964	32	T05
5	P14	BRAVO	74582	21	T05
6	PO1	VIRAT	65412	28	T01
7	PO2	MSD	12345	28	T05
8	PO3	DINESH	65212	36	T03
9	PO4	DAVID W	45412	32	T04
10	PO5	ROHIT	11412	32	T02
11	PO6	ROHAN	44412	20	T01
12	PO7	SAMARTH	11112	22	T01
13	PO8	SMITH	78412	26	T02
14	PO9	SUNIL	22212	17	T02

CAPTAIN (Captainid, CTeamid)

```
CREATE TABLE CAPTAIN (
   Captainid VARCHAR(4),
   CTeamid VARCHAR(4),
   FOREIGN KEY (Captainid) REFERENCES PLAYER (Playerid),
   FOREIGN KEY (CTeamid) REFERENCES TEAM (Teamid)
);

INSERT INTO CAPTAIN VALUES ('P01', 'T01');
```

	Captainid	CTeamid
1	PO1	T01
2	PO2	T05
3	PO3	T03
4	PO4	T04
5	PO5	T02

STADIUM (Stadiumid, Stadiumname, City, Area_name, Pincode)

```
CREATE TABLE STADIUM (
   Stadiumid VARCHAR(4) PRIMARY KEY,
   Stadiumname CHAR(15) NOT NULL,
   City CHAR(15) NOT NULL,
   Area_name CHAR(15) NOT NULL,
   Pincode INT NOT NULL
```

```
);

INSERT INTO STADIUM VALUES ('ST01', 'CHINNASWAMY', 'BANGLORE', 'MG ROAD',
'55123');
INSERT INTO STADIUM VALUES ('ST02', 'MODI', 'GUJARAT', 'GANDHI ROAD', '55555');
INSERT INTO STADIUM VALUES ('ST03', 'CRICKET_STD', 'BELAGAVI', 'KANBARGI', '590001');

SELECT * FROM STADIUM;
```

	Stadiumid	Stadiumname	City	Area_name	Pincode
1	ST01	CHINNASWAMY	BANGLORE	MGROAD	55123
2	ST02	MODI	GUJRAT	GANDHI ROAD	55555
3	ST03	CRICKET_STD	BELAGAVI	KANBARGI	590001

MATCH (Matchid, Teamid1, Teamid2, Scheduled_date, Stime, Stadiumid, Winningteamid, MOMPlayerid)

```
CREATE TABLE MATCH (
   Matchid VARCHAR(4) PRIMARY KEY,
   Teamid1 VARCHAR(4),
   Teamid2 VARCHAR(4),
   Scheduled_date DATE,
   Stime INT,
   Stadiumid VARCHAR(4),
   Winningteamid VARCHAR(4),
   MOMPlayerid VARCHAR(4),
   FOREIGN KEY (Teamid1) REFERENCES TEAM (Teamid),
   FOREIGN KEY (Teamid2) REFERENCES TEAM (Teamid),
   FOREIGN KEY (Stadiumid) REFERENCES STADIUM (Stadiumid),
   FOREIGN KEY (Winningteamid) REFERENCES TEAM (Teamid),
   FOREIGN KEY (MOMPlayerid) REFERENCES PLAYER (Playerid)
);

INSERT INTO MATCH VALUES ('MC01', 'T01', 'T02', '12-JAN-2021', 1, 'ST01', 'T01', 'P01');

SELECT * FROM MATCH;
```

	Matchid	Teamid1	Teamid2	Scheduled_date	Stime	Stadiumid	Winningteamid	MOMPlayerid
1	MC01	T01	T02	2021-01-12	1	ST01	T01	PO1
2	MC02	T01	T03	2021-01-15	1	ST01	T01	PO6
3	MC03	T01	T04	2021-01-17	1	ST01	T01	PO6
4	MC04	T02	T03	2021-01-18	12	ST02	T02	PO8
5	MC05	T04	T05	2021-01-20	12	ST02	T05	P13
6	MC06	T02	T05	2021-01-22	12	ST02	T05	P14
7	MC07	T03	T04	2021-01-24	12	ST03	T03	PO3

QUERY 1: Display the youngest player (in terms of age) Name, Team name, age in which he belongs of the tournament.

```
SELECT P.Playername, P.Playerid, MIN(P.Age) AS AGE, T.Teamname
FROM PLAYER P, TEAM T
WHERE P.Teamid = T.Teamid
GROUP BY P.Playername, P.Playerid, T.Teamname
ORDER BY AGE;
```

	Playername	Playerid	AGE
1	SUNIL	PO9	17
2	ROHAN	PO6	20
3	BRAVO	P14	21
4	HARDIK	P10	22
5	SAMARTH	PO7	22
6	SMITH	PO8	26
7	VIRAT	PO1	28
8	MSD	PO2	28
9	DAVID W	PO4	32
10	ROHIT	PO5	32
11	JADEJA	P13	32
12	BHUVI	P11	34
13	SEKHAR	P12	34
14	DINESH	PO3	36

QUERY 2: List the details of the stadium where the maximum number of matches were played.

```
SELECT M.Stadiumid, COUNT(M.Matchid) AS No_of_Matches, S.Stadiumname
FROM MATCH M, STADIUM S
WHERE M.Stadiumid = S.Stadiumid
GROUP BY M.Stadiumid, S.Stadiumname
```

```
ORDER BY No_of_Matches DESC
LIMIT 1;
```

	Stadiumid	No_of_Matches	Stadiumname
1	ST01	3	CHINNASWAMY
2	ST02	3	MODI
3	ST03	1	CRICKET_STD

QUERY 3: List the details of the player who is not a captain but got the man_of _match award at least in two matches.

```
SELECT P.Playerid, P.Playername
FROM PLAYER P
WHERE P.Playerid NOT IN (SELECT Captainid FROM CAPTAIN)
AND P.Playerid IN (SELECT MOMPlayerid FROM MATCH
        GROUP BY MOMPlayerid
        HAVING COUNT(MOMPlayerid) > 1);
```

	Playerid	Playername
1	PO6	ROHAN

QUERY 4: Display the Team details who won the maximum matches.

```
SELECT T.Teamid, T.Teamname, COUNT(M.Winningteamid) AS No_of_Matches_WON
FROM MATCH M, TEAM T
WHERE M.Winningteamid = T.Teamid
GROUP BY T.Teamid, T.Teamname
ORDER BY No_of_Matches_WON DESC
LIMIT 1;
```

	Winningteamid	No_of_Matches_WON
1	T02	1
2	T03	1
3	T05	2
4	T01	3

QUERY 5: Display the team name where all its won matches played in the same stadium.

```
SELECT M.Winningteamid, T.Teamname, S.Stadiumid, S.Stadiumname
FROM MATCH M, TEAM T, STADIUM S
WHERE M.Winningteamid = T.Teamid
AND M.Stadiumid = S.Stadiumid
GROUP BY M.Winningteamid, T.Teamname, S.Stadiumid, S.Stadiumname
HAVING COUNT(DISTINCT M.Stadiumid) = 1;
```

	Winningteamid	Teamname	Stadiumid	Stadiumname
1	T01	RCB	ST01	CHINNASWAMY
2	T02	MI	ST02	MODI
3	T05	CSK	ST02	MODI
4	T03	KKR	ST03	CRICKET_STD

Constituency (cons_id, csname, csstate, no_of_voters)

```
CREATE TABLE constituency (
  cons_id NUMBER(20) PRIMARY KEY,
  csname VARCHAR(20),
  csstate VARCHAR(20),
  no_of_voters NUMBER(10)
);

INSERT INTO constituency VALUES (111, 'Rajajinagar', 'Karnataka', 4);
INSERT INTO constituency VALUES (222, 'Ramnagar', 'Kerala', 1);
```

Party (pid, pname, psymbol)

```
CREATE TABLE party (
  pid NUMBER(20) PRIMARY KEY,
  pname VARCHAR(20),
  psymbol VARCHAR(10)
);

INSERT INTO party VALUES (876, 'BJP', 'Lotus');
INSERT INTO party VALUES (877, 'Congress', 'Hand');
```

Candidates (cand_id, phone_no, age, state, name, pid)

```
CREATE TABLE candidates (
  cand_id NUMBER(12) PRIMARY KEY,
  phone_no NUMBER(10),
  age NUMBER(2),
  state VARCHAR(20),
  name VARCHAR(20),
  pid INT REFERENCES party(pid)
);

INSERT INTO candidates VALUES (121, 9538904626, 23, 'Kerala', 'Raksha', 876);
INSERT INTO candidates VALUES (122, 9740777502, 24, 'Karnataka', 'Veena', 877);
```

Contest (cons_id, cand_id)

```
CREATE TABLE contest (
  cons_id NUMBER(20) REFERENCES constituency(cons_id),
  cand_id NUMBER(12) REFERENCES candidates(cand_id)
```

```
);

INSERT INTO contest VALUES (111, 122);
INSERT INTO contest VALUES (222, 121);
INSERT INTO contest VALUES (222, 122);
```

Voter (vid, vname, vage, vaddr, cons_id, cand_id)

```
CREATE TABLE voter (
   vid NUMBER(20) PRIMARY KEY,
   vname VARCHAR(20),
   vage NUMBER(5),
   vaddr VARCHAR(20),
   cons_id NUMBER(20) REFERENCES constituency(cons_id),
   cand_id NUMBER(12) REFERENCES candidates(cand_id)
);

INSERT INTO voter VALUES (345, 'Prashanth', 21, 'Kanakpura', 222, 122);
INSERT INTO voter VALUES (346, 'Prakash', 23, 'Ramnagar', 111, 121);
INSERT INTO voter VALUES (348, 'Nagesh', 30, 'Mandya', 111, 121);
INSERT INTO voter VALUES (349, 'Praveen', 30, 'Mandya', 111, 121);
```

Query 1: List the details of the candidates who are contesting from more than one constituencies which are belongs to different states.

```
SELECT * FROM candidates
WHERE cand_id IN (
   SELECT cand_id FROM contest
   JOIN constituency ON contest.cons_id = constituency.cons_id
   GROUP BY cand_id
   HAVING COUNT(DISTINCT csstate) > 1
);
```

CAND_ID	PHONE_NO	AGE	STATE	NAME	PID
122	9740777502	24	karnataka	veena	877

Query 2: Display the state name having maximum number of constituencies.

```
SELECT csstate FROM constituency
GROUP BY csstate
HAVING COUNT(csstate) = (
   SELECT MAX(COUNT(csstate)) FROM constituency GROUP BY csstate
);
```

```
CSSTATE
----------
Karnatak
```

Query 3: Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18 years old, then insert the tuple into the voter else display the "Not an eligible voter msg" .

```
create or replace procedure agechecking ( id in number,age in number) as
BEGIN
if age>18 then
insert into voter(vid,vage) values (id,age); else
dbms_output.put_line('age should be high'); end if;
end agechecking;
/

Procedure created.

SQL> set serveroutput on; SQL> exec
agechecking (25,21);
PL/SQL procedure successfully completed. // row inserted

SQL> exec agechecking (20,15);

age should be high //Message displayed as age is less than or equal to 18

PL/SQL procedure successfully completed.
```

Query 4: Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure.

```
create or replace procedure display_count( const_id number) as
vid constituency.cons_id % type; begin
select no_of_voters into vid from
constituency
where cons_id = const_id and rownum = 1; dbms_output.put_line ( 'total
voters are: ' || vid); end;
/

Procedure created.

SQL> select * from constituency;

CONS_ID          CSNAME          CSSTATE          NO_OF_VOTERS
---------- ---------- ---------- ----------
111              rajajinagar                karnataka          2
222              ramnagar                   kerala             1
SQL> exec display_count(111); total voters
are: 2
```

Query 5:

```
CREATE OR REPLACE TRIGGER count
AFTER INSERT ON voter
FOR EACH ROW
BEGIN
   UPDATE constituency
   SET no_of_voters = no_of_voters + 1
   WHERE cons_id = :NEW.cons_id;
END count;
/ Trigger created.

SQL> set serveroutput on;

SQL> select * from constituency;
```

CONS_ID	CSNAME	CSSTATE	NO_OF_VOTERS
111	rajajinagar	karnataka	2
222	ramnagar	kerala	1

```
SQL> insert into voter values (348,'nagesh',30,'mandya',111,121);
1 row created.
```

After insertion into voter table, the constituency table is automatically updated.

```
SQL> select * from constituency;
```

CONS_ID	CSNAME	CSSTATE	NO_OF_VOTERS
111	rajajinagar	karnataka	3
222	ramnagar	kerala	1

Tourist_Place (tpid, history, kilometers, state, tpname)

```
CREATE TABLE tourist_place (
   tpid NUMBER PRIMARY KEY,
   history VARCHAR(20),
   kilometers NUMBER(3),
   state VARCHAR(20),
   tpname VARCHAR(20)
);

INSERT INTO tourist_place VALUES (11, 'beauty', 160, 'Karnataka', 'Ooty');
INSERT INTO tourist_place VALUES (12, 'monuments', 270, 'Kerala', 'Beluru');
INSERT INTO tourist_place VALUES (13, 'beach', 360, 'Tamil Nadu', 'Marina');

SELECT * FROM tourist_place;
```

TPID	HISTORY	KILOMETERS	STATE	TPNAME
11	beauty	160	karnataka	ooty
12	monuments	270	kerala	beluru
13	beach	360	tamilnadu	marina

Tourist (tid, country, age, tname)

```
CREATE TABLE tourist (
   tid NUMBER PRIMARY KEY,
   country VARCHAR(20),
   age NUMBER,
   tname VARCHAR(20)
);

INSERT INTO tourist VALUES (22, 'India', 34, 'Prakash');
INSERT INTO tourist VALUES (23, 'Orissa', 28, 'Bhanu');
INSERT INTO tourist VALUES (24, 'India', 30, 'Nagesh');

SELECT * FROM tourist;
```

TID	COUNTRY	AGE	TNAME
22	India	34	prakash
23	orissa	28	bhanu
24	India	30	nagesh

Visits (tpid, tid, vdate)

```
CREATE TABLE visits (
   tpid NUMBER(3) REFERENCES tourist_place(tpid),
   tid NUMBER REFERENCES tourist(tid),
   vdate DATE
);

INSERT INTO visits VALUES (12, 23, '13-NOV-2014');
INSERT INTO visits VALUES (11, 24, '24-JUN-2013');
INSERT INTO visits VALUES (13, 22, '25-SEP-2011');
INSERT INTO visits VALUES (11, 23, '23-FEB-2010');
INSERT INTO visits VALUES (13, 23, '12-JAN-2010');
INSERT INTO visits VALUES (14, 24, '10-JAN-2017');

SELECT * FROM visits;
```

TPID	TID	VDATE
12	23	13-NOV-14
11	24	24-JUN-13
13	22	25-SEP-11
11	23	23-FEB-10
13	23	12-JAN-10
14	24	10-JAN-17

Email (tid, email)

```
CREATE TABLE email (
   tid NUMBER REFERENCES tourist(tid),
   email VARCHAR(20)
);

INSERT INTO email VALUES (23, 'bhanu12@gmail.com');
INSERT INTO email VALUES (22, 'prakash242@gmail.com');
INSERT INTO email VALUES (24, 'nageshh@gmail.com');
```

Query 1: List the state name which is having maximum number of tourist places.

```
SELECT state
FROM tourist_place
GROUP BY state
HAVING COUNT(state) = (
  SELECT MAX(COUNT(state))
  FROM tourist_place
  GROUP BY state
);
```

```
STATE
------------------------------

Karnataka
```

Query 2: List details of Tourist place where maximum number of tourists visited.

```
SELECT *
FROM tourist_place
WHERE tpid IN (
  SELECT tpid
  FROM visits
  GROUP BY tpid
  HAVING COUNT(tpid) = (
    SELECT MAX(COUNT(tpid))
    FROM visits
    GROUP BY tpid
  )
);
```

TPID	HISTORY	KILOMETERS	STATE	TPNAME
11	beauty	160	karnataka	ooty
13	beach	360	tamilnadu	marina

Query 3: List the details of tourists visited all tourist places of the state "KARNATAKA".

```
SELECT *
FROM tourist t
WHERE t.tid IN (
  SELECT tid
  FROM visits
  JOIN tourist_place ON visits.tpid = tourist_place.tpid
  WHERE state = 'Karnataka'
  GROUP BY tid
  HAVING COUNT(state) = (
    SELECT COUNT(state)
    FROM tourist_place
    WHERE state = 'Karnataka'
  )
);
```

TID	COUNTRY	AGE	TNAME
24	india	30	nagesh

Query 4: Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places.

```
SELECT *
FROM tourist t
WHERE t.tid IN (
  SELECT tid
  FROM visits
  JOIN tourist_place ON visits.tpid = tourist_place.tpid
  GROUP BY tid
  HAVING COUNT(DISTINCT state) = (
    SELECT COUNT(DISTINCT state)
    FROM tourist_place
  )
);
```

TID	COUNTRY	AGE	TNAME
23	orissa	28	bhanu

Query 5: Display the details of the tourist place visited by the tourists of all country.

```
SELECT *
FROM tourist_place
WHERE tpid IN (
  SELECT tpid
  FROM visits
  JOIN tourist ON visits.tid = tourist.tid
  GROUP BY tpid
  HAVING COUNT(DISTINCT country) = (
    SELECT COUNT(DISTINCT country)
    FROM tourist
  )
);
```

TPID	HISTORY	KILOMETERS	STATE	TPNAME
11	beauty	160	karnataka	ooty
13	beach	360	tamilnadu	marina

Lab 1: Create an XHTML page that provides information about your department. Your XHTML page must use the following tags: a) Text Formatting tags b) Horizontal rule

c) Meta element d) Links e) Images f) Tables

(Use of additional tags encouraged).

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="UTF-8">
  <title>Department Information</title>
</head>
<body>

  <h1>Welcome to the Department of Master of Computer Applications</h1>

  <h2>About Us</h2>
  <p>
    The Department of Master of Computer Applications was established in the year 2011.
    The Department imparts quality and knowledge-based technical skills to graduate students
    from various disciplines to be successful.
  </p>

  <h2>Faculty</h2>
  <ul>
    <li>Dr. Ovi Smith - Specialization in IoT</li>
    <li>Dr. Ira Mendke - Specialization in Data Analytics</li>
    <li>Dr. Mark Johnson - Specialization in Python</li>
  </ul>

  <h2>Research Areas</h2>
```

```
<ol>
    <li>Data Mining</li>
    <li>Artificial Intelligence</li>
    <li>Machine Learning</li>
    <li>Robotic Process Automation (RPA)</li>
</ol>
<hr>
<h2>Contact Information</h2>
<p>
    For more information, please contact our department office:<br>
    <strong>Department of MCA, Jain College of Engineering, Belagavi</strong><br>
    Address: Tippu Sultan Nagar, Hunchanatti Cross, Machhe<br>
    Phone: 123-456-7890<br>
    Email: <a href="mailto:info@jainbgm.in">info@jainbgm.in</a>
</p>

<h2>Useful Links</h2>
<ul>
    <li><a href="https://www.jce.ac.in/master-of-computer-application">MCA</a></li>
    <li><a href="https://www.jce.ac.in/student-engagement-cell">SEC</a></li>
    <li><a href="https://www.jce.ac.in/index.php/admissions20">Admission</a></li>
</ul>

<h2>Department Logo</h2>
<img src="download.png" alt="Department Logo" width="200">

<h2>Program Offerings</h2>
<table border="1">
    <thead>
        <tr>
            <th>Program</th>
```

```
            <th>Degree</th>
            <th>Duration</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>PG</td>
            <td>MCA</td>
            <td>2 years</td>
          </tr>
          <tr>
            <td>UG</td>
            <td>Computer Science</td>
            <td>4 years</td>
          </tr>
          <tr>
            <td>UG</td>
            <td>Mechanical</td>
            <td>4 years</td>
          </tr>
        </tbody>
    </table>
</body>
</html>
```

Output:

Welcome to the Department of Master of Computer Applications

About Us

The Department of Master of Computer Applications was established in the year 2011. The Department imparts quality and knowledge based technical skills to the graduate students from various disciplines to be successful.

Faculty

- Dr. Ovi Smith - Specialization in IOT
- Dr. Ira Mendke - Specialization in Data Analytics
- Dr. Mark Johnson - Specialization in Python

Research Areas

1. Data Mining
2. Artificial Intelligence
3. Machine Learning
4. RPA

Contact Information

For more information, please contact our department office:
Department of MCA, Jain College of Engineering, Belagavi
Address: Tippu Sultan Nagar,Hunchanatti Cross, Machhe
Phone: 123-456-7890
Email: info@jainbgm.in

Useful Links

- MCA
- SEC
- Admission

Department Logo

Program Offerings

Program	Degree	Duration
PG	MCA	2 years
UG	Computer Science	4 years
UG	Mechanical	4 years

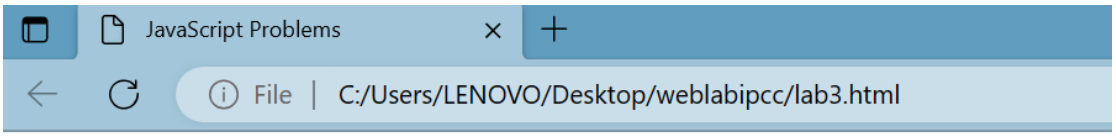
Lab 2 : Develop and demonstrate a XHTML file that includes Javascript script for the following problems: a) Input : A number n obtained using prompt Output : The first n Fibonacci numbers

b) Input : A number n obtained using prompt Output : A table of numbers from 1 to n and their squares using alert

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="UTF-8" />
  <title>JavaScript Problems</title>
  <script>
    // Problem A: Fibonacci Numbers
    function displayFibonacciNumbers() {
      var n = parseInt(prompt("Enter a number:"));
      var fibonacciNumbers = [0, 1]; // Initialize with the first two numbers

      // Generate Fibonacci numbers
      for (var i = 2; i < n; i++) {
        fibonacciNumbers.push(fibonacciNumbers[i - 1] + fibonacciNumbers[i - 2]);
      }

      alert(fibonacciNumbers);
    }

    // Problem B: Table of Numbers and Their Squares
    function displayNumberAndSquares() {
      var n = parseInt(prompt("Enter a number:"));
      var tableContent = "Number\tSquare\n";

      // Generate the table content
      for (var i = 1; i <= n; i++) {
```

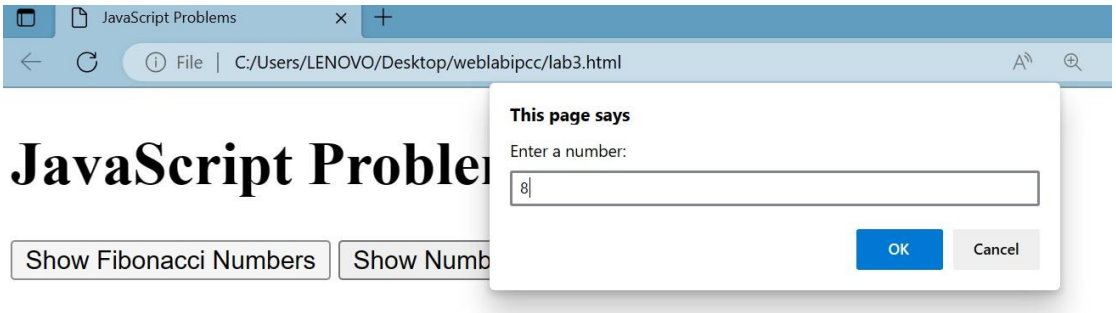
```
            tableContent += i + "\t" + (i * i) + "\n";
        }

        alert(tableContent);
    }
  </script>
</head>
<body>
  <h1>JavaScript Problems</h1>

  <button onclick="displayFibonacciNumbers()">Show Fibonacci Numbers</button>
  <button onclick="displayNumberAndSquares()">Show Number and Squares Table</button>
</body>
</html>
```

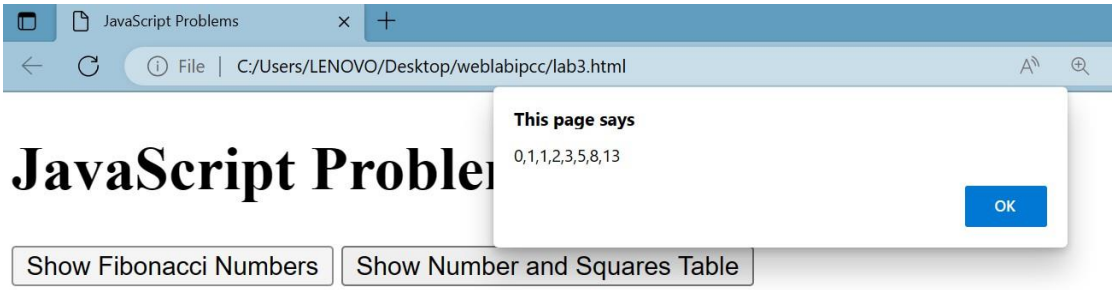
Output:

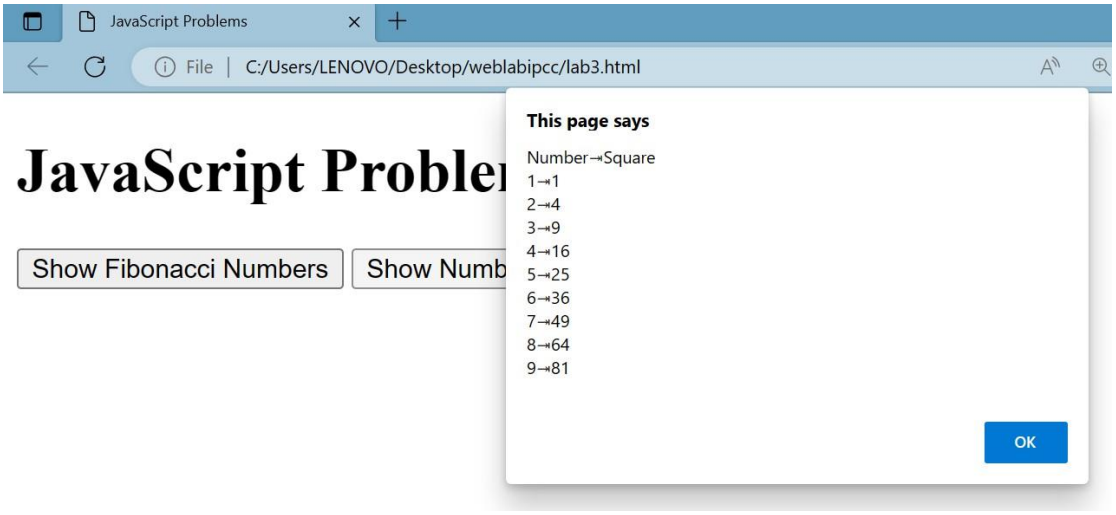


JavaScript Problems

Show Fibonacci Numbers | Show Number and Squares Table




JavaScript Problems
File | C:/Users/LENOVO/Desktop/weblabipcc/lab3.html
This page says
0,1,1,2,3,5,8,13
OK
JavaScript Proble
Show Fibonacci Numbers
Show Number and Squares Table


JavaScript Problems
File | C:/Users/LENOVO/Desktop/weblabipcc/lab3.html
This page says
Number→Square
1→1
2→4
3→9
4→16
5→25
6→36
7→49
8→64
9→81
OK
JavaScript Proble
Show Fibonacci Numbers
Show Numb

Lab 3. Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible. Modify the above document so that when a text is moved from the top stacking position, it returns to its original position rather than to the bottom.

```
<!DOCTYPE html>
<html>
<head>
  <title>Stack Ordering with MoveBack</title>
  <!-- Define styling properties -->
  <style type="text/css">
    #layer1 {
      border: solid thick black;
      background-color: brown;
      padding: 10px;
      width: 300px;
      height: 200px;
      position: absolute;
      top: 100px;
      left: 200px;
      z-index: 1;
    }

    #layer2 {
      border: solid thick black;
      background-color: gray;
      padding: 10px;
      width: 300px;
      height: 200px;
      position: absolute;
```

```
        top: 120px;
        left: 220px;
        z-index: 2;
      }

      #layer3 {
        border: solid thick black;
        background-color: white;
        padding: 10px;
        width: 300px;
        height: 200px;
        position: absolute;
        top: 140px;
        left: 240px;
        z-index: 3;
      }
    </style>
</head>
<body>

    <script type="text/javascript">
      var topLayer = "layer3";
      var origpos;

      // Function to place the chosen layer on top
      function mover(toTop, pos) {
        document.getElementById(toTop).style.zIndex = "4";
        topLayer = toTop;
        origpos = pos;
      }
```

```
    // Function to place the chosen layer back to its original place
    function moveBack() {
      document.getElementById(topLayer).style.zIndex = origpos;
    }
  </script>

  <p id=layer1 onMouseOver=mover('layer1', '1'); onMouseOut=moveBack();>This is the last layer</p>
  <p id=layer2 onMouseOver=mover('layer2', '2'); onMouseOut=moveBack();>This is the middle layer</p>
  <p id=layer3 onMouseOver=mover('layer3', '3'); onMouseOut=moveBack();>This is the first layer</p>

</body>
</html>
```

Output: