Practical Assignment - 1

- Name: Hardik Arora
- Branch : Btech - CS
- Program: AIML
- University Roll No. : 2215500071
- Section: 2AC
- Class Roll No. : 28

Q. Create the following tables and fill the given data in them:-

Table-1

```
CREATE DATABASE kt;

USE kt;

Create Table IF NOT EXISTS College(
    cName VARCHAR(10),
    state VARCHAR(10),
    enrollment INT PRIMARY KEY NOT NULL
);

INSERT INTO College(cName, state, enrollment) VALUES('Stanford', 'CA', '15000');

INSERT INTO College(cName, state, enrollment) VALUES('Berkeley', 'CA', '36000');

INSERT INTO College(cName, state, enrollment) VALUES('MIT', 'MA', '10000');

INSERT INTO College(cName, state, enrollment) VALUES('Cornell', 'NY', '21000');

INSERT INTO College(cName, state, enrollment) VALUES('Harvard', 'MA', '50040');

SELECT * FROM College;
```

OUTPUT:

MySQL Local: SELECT * FROM Co...

cName	state	enrollment
Filter...	Filter...	Filter...
MIT	MA	10000
Stanford	CA	15000
Cornell	NY	21000
Berkeley	CA	36000
Harvard	MA	50040

Table -2

```
Create Table IF NOT EXISTS Student(
    sID INT PRIMARY KEY,
    sName VARCHAR(10),
    GPA FLOAT,
    sizeHS INT NOT NULL,
    DoB DATE
);


INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('123', 'Amy', '3.9',
'1000', '1996-06-26');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('234', 'Bob', '3.6',
'1500', '1995-04-07');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('345', 'Craig', '3.5',
'500', '1995-02-04');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('456', 'Doris', '3.9',
'1000', '1997-07-24');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('567', 'Edward', '2.9',
'2000', '1996-12-21');
```

```sql
INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('678', 'Fay', '3.8',
'200', '1996-08-27');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('789', 'Gary', '3.4',
'800', '1996-10-08');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('987', 'Helen', '3.7',
'800', '1997-03-27');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('876', 'Irene', '3.9',
'400', '1996-03-07');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('765', 'Jay', '2.9',
'1500', '1998-08-08');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('654', 'Amy', '3.9',
'1000', '1996-05-26');

INSERT INTO Student(sID, sName, GPA, sizeHS, DoB) VALUES('543', 'Craig', '3.4',
'2000', '1998-08-27');

SELECT * FROM Student;
```

OUTPUT :

MySQL Local: SELECT * FROM St...

sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
123	Amy	3.9000000953674316	1000	1996-06-26T00:00:00.000Z
234	Bob	3.5999999046325684	1500	1995-04-07T00:00:00.000Z
345	Craig	3.5000000000000004	500	1995-02-04T00:00:00.000Z
456	Doris	3.9000000953674316	1000	1997-07-24T00:00:00.000Z
543	Craig	3.4000000953674316	2000	1998-08-27T00:00:00.000Z
567	Edward	2.9000000953674316	2000	1996-12-21T00:00:00.000Z
654	Amy	3.9000000953674316	1000	1996-05-26T00:00:00.000Z
678	Fay	3.799999952316284	200	1996-08-27T00:00:00.000Z
765	Jay	2.9000000953674316	1500	1998-08-08T00:00:00.000Z
789	Gary	3.4000000953674316	800	1996-10-08T00:00:00.000Z
876	Irene	3.9000000953674316	400	1996-03-07T00:00:00.000Z
987	Helen	3.7000000476837163	800	1997-03-27T00:00:00.000Z

Table – 3

```
Create Table IF NOT EXISTS Apply(
    sID INT NOT NULL,
    cName VARCHAR(10),
    major VARCHAR(20),
    decision VARCHAR(1)
);


INSERT INTO Apply(sID, cName, major, decision) VALUES('123', 'Stanford', 'CS',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('123', 'Stanford', 'EE',
'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('123', 'Berkeley', 'CS',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('123', 'Cornell', 'EE',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('234', 'Berkeley',
'biology', 'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('345', 'MIT',
'bioengineering', 'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('345', 'Cornell',
'bioengineering', 'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('345', 'Cornell', 'CS',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('345', 'Cornell', 'EE',
'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('678', 'Stanford',
'history', 'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('987', 'Stanford', 'CS',
'Y');
```

```
INSERT INTO Apply(sID, cName, major, decision) VALUES('987', 'Berkeley', 'CS',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('876', 'Stanford', 'CS',
'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('876', 'MIT', 'biology',
'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('876', 'MIT', 'marine
biology', 'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('765', 'Stanford',
'history', 'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('765', 'Cornell',
'history', 'N');

INSERT INTO Apply(sID, cName, major, decision) VALUES('765', 'Cornell',
'psychology', 'Y');

INSERT INTO Apply(sID, cName, major, decision) VALUES('543', 'MIT', 'CS', 'N');

SELECT * FROM Apply;
```

OUTPUT:

sID	cName	major	decision
Filter...	Filter...	Filter...	Filter...
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

Q. State of SQL *PLUS Queries for each of the following:

1. List the student name, dob from student table

```
SELECT sName, DoB FROM student;
```

sName	DoB
Filter...	Filter...
Amy	1996-06-26T00:00:00.000Z
Bob	1995-04-07T00:00:00.000Z
Craig	1995-02-04T00:00:00.000Z
Doris	1997-07-24T00:00:00.000Z
Craig	1998-08-27T00:00:00.000Z
Edward	1996-12-21T00:00:00.000Z
Amy	1996-05-26T00:00:00.000Z
Fay	1996-08-27T00:00:00.000Z
Jay	1998-08-08T00:00:00.000Z
Gary	1996-10-08T00:00:00.000Z
Irene	1996-03-07T00:00:00.000Z
Helen	1997-03-27T00:00:00.000Z

2. List of the name of student scoring more than 3.7 in GPA.

```
SELECT sName FROM student WHERE GPA>3.7;
```

sName
Filter...
Amy
Doris
Amy
Fay
Irene
Helen

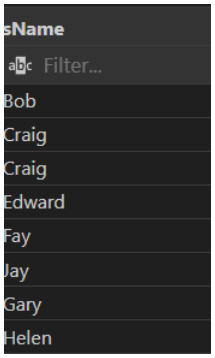
3. List the name of student whose High School size is atleast 1000 and born after 1996.

```
SELECT sName FROM student WHERE sizeHS >= 1000 AND DoB > '1996-12-31';
```

sName
Filter...
Doris
Craig
Jay

4. List the name of student who are scoring GPA between 2.9 and 3.9.

```
SELECT sName FROM student WHERE GPA >= 2.9 AND GPA <= 3.9;
```



sName
Filter...
Bob
Craig
Craig
Edward
Fay
Jay
Gary
Helen

5. List all the details of colleges who are situated in MA

```
SELECT * FROM college WHERE state = 'MA' ;
```

cName	state	enrollment
Filter...	Filter...	Filter...
MIT	MA	10000
Harvard	MA	50040

6. List the students who have scored more than 2.0 but less than 3.5.

```
SELECT sName FROM student WHERE GPA >= 2.0 AND GPA <= 3.5;
```

sName
Filter...
Craig
Craig
Edward
Jay
Gary

7. List the name of students who are born after 1 st Jul 1996 in the order of Date of Birth.

```
SELECT sName, DoB FROM student WHERE DoB > '1996-07-01' ORDER BY DoB ASC;
```

sName	DoB
Filter...	Filter...
Fay	1996-08-27T00:00:00.000Z
Gary	1996-10-08T00:00:00.000Z
Edward	1996-12-21T00:00:00.000Z
Helen	1997-03-27T00:00:00.000Z
Doris	1997-07-24T00:00:00.000Z
Jay	1998-08-08T00:00:00.000Z
Craig	1998-08-27T00:00:00.000Z

8. List the sID, cName, decision of applications that are accepted.

```
SELECT sID, cName, decision FROM apply WHERE decision = 'Y';
```

sID	cName	decision
Filter...	Filter...	Filter...
123	Stanford	Y
123	Berkeley	Y
123	Cornell	Y
345	MIT	Y
345	Cornell	Y
678	Stanford	Y
987	Stanford	Y
987	Berkeley	Y
876	MIT	Y
765	Stanford	Y
765	Cornell	Y

9. List the sID, cName of applications that are filed at Stanford.

```
SELECT sID,cName FROM apply WHERE cName = 'Stanford';
```

sID	cName
Filter...	Filter...
123	Stanford
123	Stanford
678	Stanford
987	Stanford
876	Stanford
765	Stanford

10. List all the colleges that has enrollment greater than 1000.

```
SELECT cName FROM college WHERE enrollment > '1000';
```

cName
Filter...
MIT
Stanford
Cornell
Berkeley
Harvard

11. List the colleges not in California.

```
SELECT cName FROM college WHERE state != 'CA';
```

cName
Filter...
MIT
Cornell
Harvard

12. List names of all students who came from high school having size greater than 1700 and scored GPA less than 3.8.

```
SELECT sName FROM student WHERE sizeHS > '1700' AND GPA < 3.8;
```

sName
Filter...
Craig
Edward

13. Display the description of student table.

```
DESCRIBE student;
```

Field	Type	Null	Key	Default	Extra
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
sID	int	NO	PRI	NULL	
sName	varchar(10)	YES		NULL	
GPA	float	YES		NULL	
sizeHS	int	NO		NULL	
DoB	date	YES		NULL	

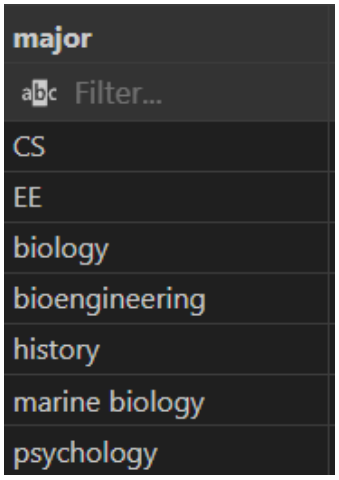
14. Display the details of all students.

```
SELECT * FROM student;
```

sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
123	Amy	3.9000000953674316	1000	1996-06-26T00:00:00.000Z
234	Bob	3.5999999046325684	1500	1995-04-07T00:00:00.000Z
345	Craig	3.5000000000000004	500	1995-02-04T00:00:00.000Z
456	Doris	3.9000000953674316	1000	1997-07-24T00:00:00.000Z
543	Craig	3.4000000953674316	2000	1998-08-27T00:00:00.000Z
567	Edward	2.9000000953674316	2000	1996-12-21T00:00:00.000Z
654	Amy	3.9000000953674316	1000	1996-05-26T00:00:00.000Z
678	Fay	3.799999952316284	200	1996-08-27T00:00:00.000Z
765	Jay	2.9000000953674316	1500	1998-08-08T00:00:00.000Z
789	Gary	3.4000000953674316	800	1996-10-08T00:00:00.000Z
876	Irene	3.9000000953674316	400	1996-03-07T00:00:00.000Z
987	Helen	3.7000000476837163	800	1997-03-27T00:00:00.000Z

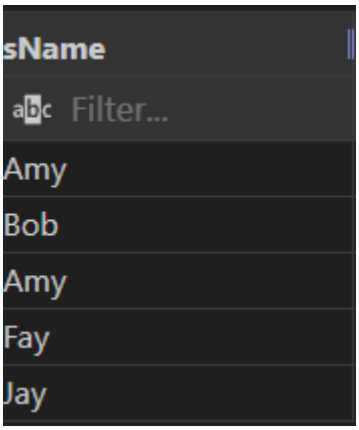
15. Display unique majors.

```
SELECT  DISTINCT major FROM apply;
```



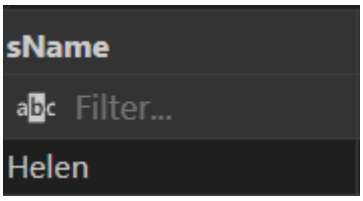
16. List the student names those are having three characters in their Names

```
SELECT sName FROM student WHERE CHAR_LENGTH(sName) = 3;
```



17. List the student names those are starting with 'H' and with five characters.

```
SELECT sName FROM student WHERE sName LIKE 'H%' AND CHAR_LENGTH(sName)=5;
```



18. List the student names that are having third character and fifth character must be 'e'.

```
SELECT sName FROM student WHERE CHAR_LENGTH(sName) >= 5 AND SUBSTRING(sName, 3,
1) = 'e' AND SUBSTRING(sName, 5, 1) = 'e';
```

sName
Irene

19. List the student names ending with ‘Y’

```
SELECT sName FROM student WHERE RIGHT(sName , 1) = 'Y';
```

sName
Amy
Amy
Fay
Jay
Gary

20. List the students in the order of their GPA.

```
SELECT sName,GPA FROM student ORDER BY GPA ASC;
```

sName	GPA
Edward	2.9
Jay	2.9
Craig	3.4
Gary	3.4
Craig	3.5
Bob	3.6
Helen	3.7
Fay	3.8
Amy	3.9
Doris	3.9
Amy	3.9
Irene	3.9

21. List the details of students in order of the ascending of GPA and descending of DoB.

```
SELECT * FROM student ORDER BY GPA ASC , DoB DESC;
```

sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
765	Jay	2.9	1500	1998-08-08
567	Edward	2.9	2000	1996-12-21
543	Craig	3.4	2000	1998-08-27
789	Gary	3.4	800	1996-10-08
345	Craig	3.5	500	1995-02-04
234	Bob	3.6	1500	1995-04-07
987	Helen	3.7	800	1997-03-27
678	Fay	3.8	200	1996-08-27
456	Doris	3.9	1000	1997-07-24
123	Amy	3.9	1000	1996-06-26
654	Amy	3.9	1000	1996-05-26
876	Irene	3.9	400	1996-03-07

22. List the sIDs of students who apply in either ‘Stanford’, ‘Cornell’ or ‘MIT’ college.

```
SELECT sID FROM apply WHERE cName IN ('Stanford','Cornell','MIT');
```

sID
Filter...
123
123
123
345
345
345
345
678
987
876
876
876
765
765
765
543

23.

23. Delete all applications filed at Stanford.

```
DELETE FROM apply WHERE cName = 'Stanford';
SELECT * FROM apply;
```

sID	cName	major	decision
Filter...	Filter...	Filter...	Filter...
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
987	Berkeley	CS	Y
876	MIT	biology	Y
876	MIT	marine biology	N
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

24. Modify the GPA of all students by giving 10% raise in GPA.

```
UPDATE student SET GPA = GPA + (0.10*GPA);
SELECT * FROM student;
```

sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
123	Amy	4.29	1000	1996-06-26
234	Bob	3.96	1500	1995-04-07
345	Craig	3.85	500	1995-02-04
456	Doris	4.29	1000	1997-07-24
543	Craig	3.74	2000	1998-08-27
567	Edward	3.19	2000	1996-12-21
654	Amy	4.29	1000	1996-05-26
678	Fay	4.18	200	1996-08-27
765	Jay	3.19	1500	1998-08-08
789	Gary	3.74	800	1996-10-08
876	Irene	4.29	400	1996-03-07
987	Helen	4.07	800	1997-03-27

25. Delete the college Stanford from the table.

```
DELETE FROM college WHERE cName = 'Stanford';
```

26. Increment the GPA of students by 1.5 whose GPA is less than 3.5 and whose High School Size is greater than 1500.

```
Update student SET GPA = GPA + 1.5 WHERE GPA < 3.5 AND sizeHS > 1500;
 SELECT * FROM student;
```

sID	sName	GPA	sizeHS	DoB
Filter...	Filter...	Filter...	Filter...	Filter...
123	Amy	4.29	1000	1996-06-26
234	Bob	3.96	1500	1995-04-07
345	Craig	3.85	500	1995-02-04
456	Doris	4.29	1000	1997-07-24
543	Craig	3.74	2000	1998-08-27
567	Edward	4.69	2000	1996-12-21
654	Amy	4.29	1000	1996-05-26
678	Fay	4.18	200	1996-08-27
765	Jay	3.19	1500	1998-08-08
789	Gary	3.74	800	1996-10-08
876	Irene	4.29	400	1996-03-07
987	Helen	4.07	800	1997-03-27

27. Delete the students who have scored less than 3.2 GPA.

```
DELETE * FROM student WHERE GPA < 3.2;
SELECT * FROM student;
```

Exercise:

Q. Determine the appropriate datatype:

- Deptno – INT
- Dname – VARCHAR(255)
- Loc – VARCHAR(255)

```
CREATE TABLE IF NOT EXISTS Dept(
    deptno INT PRIMARY KEY,
    dname VARCHAR(255),
    loc VARCHAR(255)
);

INSERT INTO Dept(deptno, dname, loc) VALUES('1', 'ACCOUNTING', 'ST LOUIS');

INSERT INTO Dept(deptno, dname, loc) VALUES('2', 'RESEARCH', 'NEW YORK');

INSERT INTO Dept(deptno, dname, loc) VALUES('3', 'SALES', 'ATLANTA');

INSERT INTO Dept(deptno, dname, loc) VALUES('4', 'OPERATIONS', 'SEATTLE');

SELECT * FROM Dept;
```

deptno	dname	loc
Filter...	Filter...	Filter...
1	ACCOUNTING	ST LOUIS
2	RESEARCH	NEW YORK
3	SALES	ATLANTA
4	OPERATIONS	SEATTLE

Q. Determine the appropriate datatype:

- empno – INT
- ename – VARCHAR(255)
- job – VARCHAR(255)
- mgr – VARCHAR(255)
- hiredate – DATE
- sal – INT
- comm – VARCHAAR(255)
- dept - INT

```
CREATE TABLE IF NOT EXISTS Employee(
    empno INT PRIMARY KEY,
    ename VARCHAR(255),
    job VARCHAR(255),
    mgr VARCHAR(255),
    hierdate DATE,
    sal INT NOT NULL,
    comm VARCHAR(255),
    dept INT NOT NULL
);

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('1', 'JOHNSON', 'ADMIN', '6', '1990-12-17', '18000', '(null)', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('2', 'HARDING', 'MANAGER', '9', '1998-02-02', '52000', '300', '3');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('3', 'TAFT', 'SALES 1', '2', '1996-01-02', '25000', '500', '3');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('4', 'HOOVER', 'SALES 1', '2', '1990-04-02', '27000', '(null)', '3');
```

```sql
INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('5', 'LINCOLN', 'TECH', '6', '1994-06-23', '22500', '1400', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('6', 'GARFIELD', 'MANAGER', '9', '1993-05-01', '54000', '(null)', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('7', 'POLK', 'TECH', '6', '1997-09-27', '25000', '(null)', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('8', 'GRANT', 'ENGINEER', '10', '1997-03-30', '32000', '(null)', '2');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('9', 'JACKSON', 'CEO', '(null)', '1990-01-01', '75000', '(null)', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('10', 'FILLMORE', 'MANAGER', '9', '1994-08-09', '56000', '(null)', '2');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('11', 'ADAMAS', 'ENGINEER', '10', '1996-03-15', '34000', '(null)', '2');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('12', 'WASHINGTON', 'ADMIN', '6', '1998-04-16', '18000', '(null)', '4');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('13', 'MONROE', 'ENGINEER', '10', '2000-12-03', '30000', '(null)', '2');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('14', 'ROOSEVELT', 'CPA', '9', '1995-10-12', '35000', '(null)', '1');

INSERT INTO Employee(empno, ename, job, mgr, hierdate, sal, comm, dept) VALUES
('15', 'HANCOCK', 'SALES 1', '2', '1990-03-02', '27500', '(null)', '3');

SELECT * FROM employee;
```

empno	ename	job	mgr	hierdate	sal	comm	dept
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	JOHNSON	ADMIN	6	1990-12-17	18000	(null)	4
2	HARDING	MANAGER	9	1998-02-02	52000	300	3
3	TAFT	SALES 1	2	1996-01-02	25000	500	3
4	HOOVER	SALES 1	2	1990-04-02	27000	(null)	3
5	LINCOLN	TECH	6	1994-06-23	22500	1400	4
6	GARFIELD	MANAGER	9	1993-05-01	54000	(null)	4
7	POLK	TECH	6	1997-09-27	25000	(null)	4
8	GRANT	ENGINEER	10	1997-03-30	32000	(null)	2
9	JACKSON	CEO	(null)	1990-01-01	75000	(null)	4
10	FILLMORE	MANAGER	9	1994-08-09	56000	(null)	2
11	ADAMAS	ENGINEER	10	1996-03-15	34000	(null)	2
12	WASHINGTON	ADMIN	6	1998-04-16	18000	(null)	4
13	MONROE	ENGINEER	10	2000-12-03	30000	(null)	2
14	ROOSEVELT	CPA	9	1995-10-12	35000	(null)	1
15	HANCOCK	SALES 1	2	1990-03-02	27500	(null)	3

Solve the following queries:

Q1. Employee Name and Hire Date sorted by Hire Date(Recent to Old).

```
SELECT ename,hierdate FROM employee ORDER BY hierdate DESC;
```

ename	hierdate
Filter...	Filter...
MONROE	2000-12-03
WASHINGTON	1998-04-16
HARDING	1998-02-02
POLK	1997-09-27
GRANT	1997-03-30
ADAMAS	1996-03-15
TAFT	1996-01-02
ROOSEVELT	1995-10-12
FILLMORE	1994-08-09
LINCOLN	1994-06-23
GARFIELD	1993-05-01
JOHNSON	1990-12-17
HOOVER	1990-04-02
HANCOCK	1990-03-02
JACKSON	1990-01-01

Q2. Employee Name and Job sorted by Job(Alphabetically).

```
SELECT ename,job FROM employee ORDER BY job ASC;
```

ename	job
Filter...	Filter...
JOHNSON	ADMIN
WASHINGTON	ADMIN
JACKSON	CEO
ROOSEVELT	CPA
GRANT	ENGINEER
ADAMAS	ENGINEER
MONROE	ENGINEER
HARDING	MANAGER
GARFIELD	MANAGER
FILLMORE	MANAGER
TAFT	SALES 1
HOOVER	SALES 1
HANCOCK	SALES 1
LINCOLN	TECH
POLK	TECH

Q3. Employee Name and Job for all Engineers, sorted by Employee Name Alphabetically.

```
SELECT ename,job FROM employee ORDER BY ename ASC;
```

ename	job
Filter...	Filter...
ADAMAS	ENGINEER
FILLMORE	MANAGER
GARFIELD	MANAGER
GRANT	ENGINEER
HANCOCK	SALES 1
HARDING	MANAGER
HOOVER	SALES 1
JACKSON	CEO
JOHNSON	ADMIN
LINCOLN	TECH
MONROE	ENGINEER
POLK	TECH
ROOSEVELT	CPA
TAFT	SALES 1
WASHINGTON	ADMIN

Q4. Job, Employee Name , Salary and Commission for employees with salary over 50000 sorted by Salary (Largest to Smallest).

```
SELECT job,ename,sal,comm FROM employee WHERE sal > '50000' ORDER BY sal DESC;
```

job	ename	sal	comm
Filter...	Filter...	Filter...	Filter...
CEO	JACKSON	75000	(null)
MANAGER	FILLMORE	56000	(null)
MANAGER	GARFIELD	54000	(null)
MANAGER	HARDING	52000	300

Q5. Job, Employee Name, Salary and Commission for employees with a Commission sorted by Salary (Largest to Smallest).

```
SELECT job,ename,sal,comm FROM employee WHERE comm != '(null)' ORDER BY sal DESC;
```

job	ename	sal	comm
Filter...	Filter...	Filter...	Filter...
MANAGER	HARDING	52000	300
SALES 1	TAFT	25000	500
TECH	LINCOLN	22500	1400

Q6. Job, Employee Name, Salary and Commission for employees whose name starts with the letter ‘H’.

```
SELECT job,ename,sal,comm FROM employee WHERE SUBSTRING(ename, 1, 1) = 'H';
```

job	ename	sal	comm
Filter...	Filter...	Filter...	Filter...
MANAGER	HARDING	52000	300
SALES 1	HOOVER	27000	(null)
SALES 1	HANCOCK	27500	(null)

Q7. Job, Employee Name, Salary and Commission for employees whose name starts with the letter 'H' and who do not get any Commission.

```
SELECT job,ename,sal,comm FROM employee WHERE SUBSTRING(ename, 1, 1) = 'H' AND
comm = '(null)';
```

job	ename	sal	comm
Filter...	Filter...	Filter...	Filter...
SALES 1	HOOVER	27000	(null)
SALES 1	HANCOCK	27500	(null)

Q8. Job, Employee Name for employees in Dept No. 3.

```
SELECT job,ename FROM employee WHERE dept = '3';
```

job	ename
Filter...	Filter...
MANAGER	HARDING
SALES 1	TAFT
SALES 1	HOOVER
SALES 1	HANCOCK

Q9. Dept Name and Loc for employees in Dept No. 3.

```
SELECT dname,loc FROM dept WHERE deptno = '3';
```

dname	loc
Filter...	Filter...
SALES	ATLANTA

Q10. Job, Employee Name, Dept, Salary sorted first by Dept(Smallest to Largest) and then Salary(Largest to Smallest).

```
SELECT job,ename,dept,sal FROM employee ORDER BY dept ASC,sal DESC;
```

job	ename	dept	sal
Filter...	Filter...	Filter...	Filter...
CPA	ROOSEVELT	1	35000
MANAGER	FILLMORE	2	56000
ENGINEER	ADAMAS	2	34000
ENGINEER	GRANT	2	32000
ENGINEER	MONROE	2	30000
MANAGER	HARDING	3	52000
SALES 1	HANCOCK	3	27500
SALES 1	HOOVER	3	27000
SALES 1	TAFT	3	25000
CEO	JACKSON	4	75000
MANAGER	GARFIELD	4	54000
TECH	POLK	4	25000
TECH	LINCOLN	4	22500
ADMIN	JOHNSON	4	18000
ADMIN	WASHINGTON	4	18000

- Submitted By : Hardik Arora
- Branch : Btech - CS
- Program: AIML
- University Roll No. : 2215500071
- Section: 2AC
- Class Roll No. : 28
- Submitted to: Ayushi Mam