

PROBLEM STATEMENT 1. Create the table student and branch as following.

OBJECTIVE: Creating the table student and branch and inserting values into the tables

Query-

```
create table students(Roll_no int(2),Student_name varchar(20),Age int(2),Branch_id int(3));
```

```
insert into students values(1,'Andrew',18,10);
```

```
insert into students values(2,'Angle',19,10);
```

```
insert into students values(3,'Priya',20,10);
```

```
insert into students values(4,'Analisa',21,11);
```

```
create table branch(Branch_id int(3),Branch_name varchar(20),HOD varchar(10));
```

```
insert into branch values(10,'CSE','Mr.abc');
```

```
insert into branch values(11,'EC','Dr.xyz');
```

```
insert into branch values(12,'EX','Dr.pqr');
```

OUTPUT-

	Roll_no	Student_name	Age	Branch_id
►	1	Andrew	18	10
	2	Angle	19	10
	3	Priya	20	10
	4	Analisa	21	11
	5	Anna	21	12

	Branch_id	Branch_name	HOD
►	10	CSE	Mr.abc
	11	EC	Dr.xyz
	12	EX	Dr.pqr

PROBLEM STATEMENT 5- Select those students where their name ending with alphabet 'a'

OBJECTIVE – selecting students having ending with alphabet 'a'.

QUERY-

```
select * from students where Student_name LIKE "%a";
```

OUTPUT-

	Roll_no	Student_name	Age	Branch_id
▶	3	Priya	20	10
	4	Analisa	21	11
	5	Anna	21	12

PROBLEM STATEMEN 7 - Apply following Constraints on the above tables.

a> Roll no Should be Not Null.

b>Roll no Should be primary key in student table and Branch_id should be primary key in table branch.

c>Make Branch_id as a foreign key in table Branch which refers table student.

d> Apply **Check** constraint with condition age>=18.

OBJECTIVE – Applying the constrains in the tables as per the problem statement.

QUERY-

```
alter table students add constraint primary key (Roll_no);
```

```
alter table students add constraint check(age>=18);
```

	Field	Type	Null	Key	Default	Extra
►	Roll_no	int	NO	PRI	NULL	
	Student_name	varchar(20)	YES		NULL	
	Age	int	YES		NULL	
	Branch_id	int	YES		NULL	

PROBLEM STATEMENT -8. Use Alter command to add,delete and modify columns on table Student and show the results.

OBJECTIVE – using alter command as per the problem statemnet.

QUERY-

alter table students add column branch_name varchar(10);

	Roll_no	Student_name	Age	Branch_id	branch_name
▶	1	Andrew	18	10	NULL
	2	Angle	19	10	NULL
	3	Priya	20	10	NULL
	4	Analisa	21	11	NULL
	5	Anna	21	12	NULL
•	NULL	NULL	NULL	NULL	NULL

alter table students drop column branch_name;

	Roll_no	Student_name	Age	Branch_id
▶	1	Andrew	18	10
	2	Angle	19	10
	3	Priya	20	10
	4	Analisa	21	11
	5	Anna	21	12

PROBLEM STATEMENT 12 - Sort the Table Student in ascending order on the basis of age and for same age branch_id should be in descending order.

OBJECTIVE – applying sorting in the table as per problem statement.

QUERY-

```
select * from students order by Age,Branch_id desc;
```

OUTPUT-

	Roll_no	Student_name	Age	Branch_id
▶	1	Andrew	18	10
	2	Angle	19	10
	3	Priya	20	10
	5	Anna	21	12
	4	Analisa	21	11
•	NULL	NULL	NULL	NULL

PROBLEM STATEMENT 14-.Apply Aggregate functions as given below.

a> Find the sum of column age in table student.

b> Find the Maximum age in table student.

c> Find the Minimum age in table student.

d> Find the Average age in table student.

e> Count number of students where age is between 18 and 20.

OBJECTIVE – Applying aggregate function in the table as per the problem statement.

QUERY-

select sum(Age) from students;

	sum(Age)
▶	99

select max(Age) from students;

	max(Age)
▶	21

select min(Age) from students;

	min(Age)
▶	18

select avg(Age) from students;

	avg(Age)
▶	19.8000

select count(*) from students where Age between 18 and 20;

	count(*)
▶	3

PROBLEM STATEMET-13. Apply IN, ANY and ALL operators for above tables and show the results.(Sub queries)

OBJECTIVE – Applying operators in the tables,

QUERY -

select Student_name, Branch_id from students where Branch_id in(10,11);

	Student_name	Branch_id
▶	Andrew	10
	Angle	10
	Priya	10
	Analisa	11

select all Student_name from students where Age > 18;

	Student_name
▶	Angle
	Priya
	Analisa
	Anna

select * from students where Age > any (select Age from students) ;

	Roll_no	Student_name	Age	Branch_id
▶	2	Angle	19	10
	3	Priya	20	10
	4	Analisa	21	11
	5	Anna	21	12
*	NULL	NULL	NULL	NULL

PROBLEM STATEMENT 9-. Apply INNER JOIN and FULL JOIN on above both tables.

OBJECTIVE – Applying inner join and full join on student and branch

QUERY-

```
select * from students inner join branch on students.Branch_id= branch.Branch_id;
```

OUTPUT-

	Roll_no	Student_name	Age	Branch_id	Branch_id	Branch_name	HOD
▶	1	Andrew	18	10	10	CSE	Mr.abc
	2	Angle	19	10	10	CSE	Mr.abc
	3	Priya	20	10	10	CSE	Mr.abc
	4	Analisa	21	11	11	EC	Dr.xyz
	5	Anna	21	12	12	EX	Dr.pqr

PROBLEM STATEMENT 10 - Apply NATURAL JOIN on above both tables.

OBJECTIVE – Applying natural join on the tables.

QUERY-

```
select * from students natural join branch;
```

OUTPUT-

	Branch_id	Roll_no	Student_name	Age	Branch_name	HOD
►	10	1	Andrew	18	CSE	Mr.abc
	10	2	Angle	19	CSE	Mr.abc
	10	3	Priya	20	CSE	Mr.abc
	11	4	Analisa	21	EC	Dr.xyz
	12	5	Anna	21	EX	Dr.pqr

PROBLEM STATEMENT 11 - Apply OUTER JOIN and all its types on above both tables.

OBJECTIVE – Applying right outer ,left outer join in the tables.

QUERY-

select * from students right join branch on students.Branch_id=branch.Branch_id;

	Roll_no	Student_name	Age	Branch_id	Branch_id	Branch_name	HOD
►	3	Priya	20	10	10	CSE	Mr.abc
	2	Angle	19	10	10	CSE	Mr.abc
	1	Andrew	18	10	10	CSE	Mr.abc
	4	Analisa	21	11	11	EC	Dr.xyz
	5	Anna	21	12	12	EX	Dr.pqr

select * from students left join branch on students.Branch_id=branch.Branch_id;

	Roll_no	Student_name	Age	Branch_id	Branch_id	Branch_name	HOD
►	1	Andrew	18	10	10	CSE	Mr.abc
	2	Angle	19	10	10	CSE	Mr.abc
	3	Priya	20	10	10	CSE	Mr.abc
	4	Analisa	21	11	11	EC	Dr.xyz
	5	Anna	21	12	12	EX	Dr.pqr

PROBLEM STATEMENT 15- Apply group by statement on attribute Branch_id in table student and apply a condition(with having clause) that age>18

OBJECTIVE- implementing the above query using group by and having clause.

QUERY-

```
select * from students group by Branch_id having Age > 18;
```

OUTPUT-

Roll_no	Student_name	Age	Branch_id
4	Analisa	21	11
5	Anna	21	12

PROBLEM STATEMENT 16 - Create a VIEW from table student and select only Roll_no and name attributes with condition age>=19.

OBJECTIVE – creating the view from the table student as per the problem statement.

QUERY-

```
create view view_name as select Roll_no,Student_name from students where Age >=19;
```

```
select * from view_name;
```

OUTPUT-

	Roll_no	Student_name
►	2	Angle
	3	Priya
	4	Analisa
	5	Anna

PROBLEM STATEMENT 17 -Create these two tables and insert values as mentioned above.

OBJECTIVE – Creating the two tables and inserting the values inside it and displaying the tables

QUERY AND OUTPUT-

```
create table t1(empid int(2),name varchar(20),country varchar(20),age int(2),mob int(10));
```

```
insert into t1 values(1,'shubham','india',23,73847934);
```

```
insert into t1 values(2,'aman','australia',21,436789555);
```

```
insert into t1 values(3,'naveen','sri lanka',24,3473847);
```

```
insert into t1 values(5,'nishant','spain',22,73248679);
```

```
select * from t1;
```

	empid	name	country	age	mob
►	1	shubham	india	23	73847934
	2	aman	australia	21	436789555
	3	naveen	sri lanka	24	3473847
	4	aditya	austia	21	328440934
	5	nishant	spain	22	73248679

```
create table t2(empid int(2),name varchar(20),country varchar(20),age int(2),mob int(10));
```

```
insert into t2 values(1,'tommy','england',23,738985734);
```

```
insert into t2 values(2,'allen','france',21,43678055);
```

```
insert into t2 values(3,'nancy','india',24,34873847);
```

```
insert into t2 values(4,'adi','ireland',21,320254934);
```

```
insert into t2 values(5,'sandy','spain',22,70248679);
```

```
select * from t2;
```

	empid	name	country	age	mob
►	1	tommy	england	23	738985734
	2	allen	france	21	43678055
	3	nancy	india	24	34873847
	4	adi	ireland	21	320254934
	5	sandy	spain	22	70248679

Then apply all Set operations in these two tables i.e

a> UNION (On attribute Country)

b>UNION ALL (On attribute EmpId)

c>INTERSECT (On attribute Country)

d>MINUS (On attribute Mob)

QUERY-

select country from t1 union select country from t2;

	country
▶	india
	australia
	sri lanka
	austria
	spain
	england
	france
	ireland

select empid from t1 union all select empid from t2;

	empid
▶	1
	2
	3
	4
	5
	1
	2
	3
	4
	5

