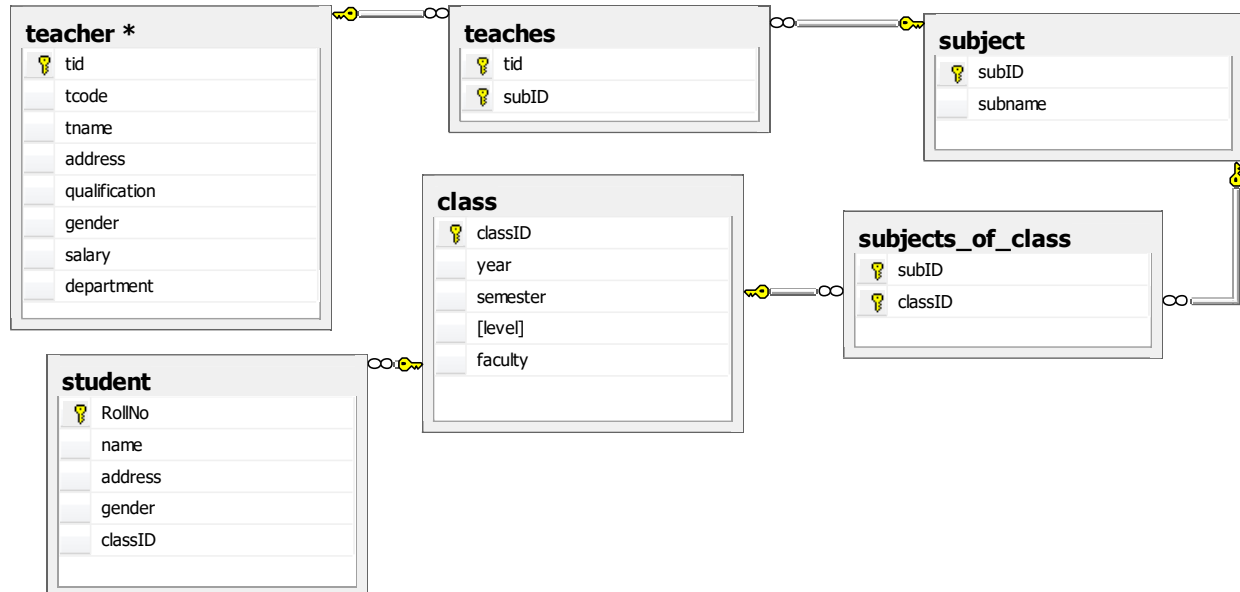


Assume database schema as following diagram. Based on the given schema, write SQL.



1. create a database named dblab
create database dblab;

2. Write a sql to choose/select a database from multiple databases
use dblab;

3. Write a SQL to create a table teacher with primary key
create table teacher(tid int primary key,tcode varchar(5), tname varchar(30),address varchar(20), qualification varchar(20),gender char(1), salary float,department varchar(10));

4. Write a sql to insert data into teacher table
insert into teacher values(100,'RPS','Ram Prasad Sharma','kathmandu','ME','M',20000,'Computer');

insert into teacher values(104,'RDB','Ram Dev Bhandari','kathmandu','PhD','M',10000,'Elx');

insert into teacher values(105,'MT','Manisha Thapa','Biratnagar','ME','F',40000,'Computer');

insert into teacher(tid,tcode,tname,address,qualification,gender,salary,department) values(101,'HPS','Hari Prasad Sharma','kathmandu','BE','M',5000,'Computer');

insert into teacher(tid,tcode,tname,address,gender,qualification,salary,department) values(102,'HT','Hari Thapa','kathmandu','M','PhD',5000,'Computer');

insert into teacher(tid,tname,address,gender,salary,department) values(103,'Sita Devi','Bhaktapur','F',54000,'Computer');

insert into teacher(tid,tcode,tname,address,gender,salary,department) values(107,'SG','Sital Guragain','Bhaktapur','F',6000,'Elx');

```
insert into teacher values(110,'IPS','Ishwor Prasad
Sharma','Pokhara','ME','M',5000,'Civil');
```

```
insert into teacher values(111,'RSS','Ramesh Sagar
Sharma','Birtnagar','BE','M',8000,'Mechanical');
```

```
insert into teacher values(112,'PD','Padam
Dhamala','Dhangadhi','PhD','M',9000,'Civil');
```

```
insert into teacher values(113,'RBC','Ram Bahadur
Chand','Baitadi','ME','M',15000,'Mechanical');
```

```
insert into teacher values(114,'MK','Manisha
Koirala','Biratnagar','ME','F',30000,'Civil');
```

5. Write a sql to drop table teacher and create again to test other constraints

drop table teacher

---to create teacher table again with not null constraint

```
create table teacher(tid int primary key,tcode varchar(5), tname varchar(30)
not null,address varchar(20), qualification varchar(20),gender char(1),salary
float,department varchar(10));
```

---to insert data again. you can use above queries also to insert more

```
insert into teacher(tid,tcode,address,qualification,gender,salary,department)
values(104,'PP','Biratnagar','BA','F',4000,'Computer');
```

---drop table teacher and to create teacher table again with not null and default constraint

```
create table teacher(tid int primary key,tcode varchar(5), tname varchar(30)
not null,address varchar(20) default 'KTM', qualification varchar(20),gender
char(1),salary float,department varchar(10));
```

---to insert data again. you can use above queries also to insert more

```
insert into teacher(tid,tcode,tname,gender,salary,department)
values(105,'CM','Chadani Mall','F',5000,'Computer');
```

---drop table teacher and to create teacher table again with not null,default and unique constraint

```
create table teacher(tid int primary key,tcode varchar(5), tname varchar(30)
not null,address varchar(20) default 'KTM', qualification varchar(20),gender
char(1),salary float,department varchar(10),constraint unq_teacher
unique(tcode));
```

---to insert data again. you can use above queries also to insert more

```
insert into teacher(tid,tcode,tname,gender,salary,department)
values(106,'CM','Chahana Mall','F',34000,'Computer');
```

---drop table teacher and to create teacher table again with not null,default, unique and check constraint

```
create table teacher(tid int primary key,tcode varchar(5), tname varchar(30)
not null,address varchar(20) default 'KTM', qualification varchar(20),gender
```

```
char(1),salary float,department varchar(10),constraint uniq_teacher
unique(tcode),constraint chk_gender check(gender in('M','F')))
```

```
---to insert data again. you can use above queries also to insert more
insert into teacher(tid,tcode,tname,gender,salary,department)
values(106,'CDM','Chahana Devi Mall','G',9000,'Computer');
```

6. Write a sql to create another table subject and insert data into it

```
create table subject(subID int primary key, subname varchar(40))
```

```
---to insert data into subject table
insert into subject values(1,'Programming in C');
insert into subject values(2,'Object Oriented Programming using C++');
insert into subject values(3,'Programming in Java');
insert into subject values(4,'Database Mangement system');
insert into subject values(5,'Operating System');
insert into subject values(6,'Computer Graphics');
```

7. Write a sql to create another table teaches with foreign key and insert data into it

```
create table teaches(tid int,subID int,constraint pk_teaches primary
key(tid,subID),constraint fk_tid_teaches foreign key(tid) references
teacher(tid),constraint fk_subID_teaches foreign key(subID) references
Subject(subID) )
```

```
---to insert data into teaches table
insert into teaches values(100,1);
insert into teaches values(101,4);
insert into teaches values(102,3);
insert into teaches values(103,2);
```

8. Write a sql to create another table class and insert data into it

```
create table class(classID int primary key, [year] int, semester
varchar(10),level varchar(10),faculty varchar(15));
```

```
---to insert data into class table
insert into class values(4000,4,'II','Bachelor','Computer')
insert into class values(4001,4,'II','Bachelor','Electronics')
insert into class values(4002,1,'II','Bachelor','Computer')
```

9. Write a sql to create another table subjects_of_class and insert data into it.

```
create table subjects_of_class(subID int references subject(subID),classID
int references class(classID), primary key(subID,classID))
```

```
---to insert data into subjects_of_class table
insert into subjects_of_class values(1,4002);
insert into subjects_of_class values(2,4002);
insert into subjects_of_class values(3,4000);
insert into subjects_of_class values(4,4001);
```

10. Write a sql to create another table student and insert data into it

```
create table student(RollNo varchar(20) primary key, [name] varchar(50),
address varchar(25),gender char(1),classID int references class(classID));
```

```
---to insert data into student table
```

```

insert into student values(431,'Ram Prasad Sharma','kathmandu','M',4002)
insert into student values(435,'Hari Bdr Rai','kathmandu','M',4001)
insert into student values(440,'Manisha Thapa','Bhaktapur','F',4002)
insert into student values(631,'Chaya Devi','Biratnagar','F',4000)

```

11. Write sql to display name, gender and address of all teachers

```

select tname,gender, address from teacher;

```

12. To display information (i.e. all columns) of all teachers

```

select * from teacher;
--or it can be written as
select tid,tcode,tname,address,qualification,gender from teacher;

```

13. To display information of all teacher on ascending order of name

```

select * from teacher order by tname asc;

```

14. To display information of all teacher on descending order of name

```

select * from teacher order by tname desc;

```

15. To display records of teacher on descending order of address and ascending order of name

```

--i.e. order by first address in descending order and then by name in
ascending order
select * from teacher order by address desc,tname asc;

```

16. To display records of all female teachers

```

select * from teacher where gender='F'

```

17. To display records of all male teacher with qualification PhD.

```

select * from teacher where gender='M' and qualification='PhD'

```

18. To display records of all teacher whose address is not kathmandu

```

select * from teacher where address<>'kathmandu'

```

19. To display records of teacher whose tid is less than 103

```

select * from teacher where tid<103

```

20. To display records of teacher whose tid is greater than or equal to 101 and less than or equal to 104

```

select * from teacher where tid between 101 and 104;
----or-----
select * from teacher where tid>=101 and tid<=104;

```

21. To display records of teacher whose address is biratnagar or bhktapur

```

select * from teacher where address='biratnagar' or address='bhaktapur'
--or----
select * from teacher where address in('biratnagar','bhaktapur');

```

22. Display teacher name, department, salary and salary after addition of 1000 in each teacher.

```

select tname,department,salary,salary+1000 as [Increased Salary] from
teacher;

```

23. To display records of teacher whose name starts with R

```

select * from teacher where tname like 'R%';

```

24. To display records of teacher whose code ends with PS

select * from teacher where tcode like '%PS';

25. To display records of teacher whose code is of length 2 character

select * from teacher where tcode like '__';

26. To display records of teacher whose name starts with character R and ends with a

select * from teacher where tname like 'R#a';

27. To display records of teacher whose name begins with 'R' and doesn't end with 'a'

select * from teacher where tname like 'R%[^a]';

28. To display records of teacher whose name contains character 'e' at any position.

select * from teacher where tname like '%e%';

29. To list out name of cities i.e. address of teacher

select distinct address from teacher;

30. To list out top five records from teacher table

select top 5 * from teacher;

31. How many number of teachers are there in table teacher? i.e. count the number of teacher

select count(*) from teacher;

32. To display average salary of teachers

select avg(salary) from teacher;

33. To display average salary of teachers working computer department

select avg(salary) from teacher where department='Computer';

34. To Alias column name

select avg(salary) as [Average Salary] from teacher where department='Computer';

35. To display minimum and maximum value of salary of teachers working at Elx department

select min(salary) as Minimum, max(salary) as Maximum from teacher where department = 'Elx';

36. To display records of teachers whose code is null

select * from teacher where tcode is null;

37. To count the teachers in each department

select department, count(*) as [No. of Employee] from teacher group by department;

38. To find average salary of each department

select department, avg(salary) as [Average Salary] from teacher group by department;

39. To find the average salary of teachers in those departments where the average salary is more than 100000

select department, avg(salary) as [Average Salary] from teacher group by department having avg(salary) > 100000;

40. To find the average salary of all teachers working in each department, if the department has at least 2 teachers;

```
select department,avg(salary)as [Average Salary] from teacher group by  
department having count(tid)>=3
```

41. To find maximum and minimum salary of teachers working in each department

```
select department, min(salary) as MinSal, Max(salary) as MaxSal from teacher  
group by department;
```

42. To find number of teachers for each gender (Male or female) in each department

```
select department,gender,count(*) from teacher group by department,gender  
order by department,gender
```

43. To find number of teachers for each gender (Male or female) in each department if there are teachers of specific gender working less than 2 in numbers in a department.

```
select department,gender,count(*) from teacher group by department,gender  
having count(*)<2
```

-----subqueries:A common use of subqueries is to perform tests for set membership, make set comparisons, and determine set cardinality, by nesting subqueries in the where clause

44. To find records of all teachers working in Computer, Civil and Elx department.

```
select * from teacher where department in('Computer','Civil','Elx');
```

-----or-----

```
select * from teacher where department ='Computer' or department='Civil' or  
department='Elx'
```

45. To find records of all teachers working in all departments except Computer.

```
select * from teacher where department not in('Computer');
```

46. To find all teachers who doesn't teach any subjects;

```
select * from teacher where tid not in(select distinct tid from teaches)
```

47. To find all teachers who teaches at least one subject.

```
select * from teacher where tid in(select distinct tid from teaches)
```

48. Find the names of all teachers whose salary is greater than at least one teacher in the Civil department

```
select T1.tname,T1.salary from teacher as T1, teacher as T2 where T1.salary  
> T2.salary and T2.department = 'Civil';
```

--or

```
select tname,salary from teacher where salary>some(select salary from  
teacher where department='Civil')
```

--or

```
select tname,salary from teacher where salary>(select min(salary) from  
teacher where department='Civil')
```

49. To find the names of all teachers whose salary is greater than that of each teacher in the Civil department

```
select tname,salary from teacher where salary>all(select salary from  
teacher where department='Civil')
```

-- or

```
select tname,salary from teacher where salary>(select max(salary) from  
teacher where department='Civil')
```

50. To find the departments that have the highest average salary

```
select department,avg(salary) from teacher group by department having  
avg(salary)>=all(select avg(salary) from teacher group by department)
```

51. To find records of teachers who teaches at least one subject.[Use exist keyword]

```
select * from teacher as T where exists(select * from teaches where  
T.tid=teaches.tid)
```

52. To find department which has most employee (i.e. teachers)

```
select department,count(*) as [no of teachers] from teacher group by  
department having count(*)>=all(select count(*) from teacher group by  
department);
```

53. To find all teachers that teaches at most one subject

```
select * from teacher where(select count(subID) from teaches where  
teaches.tid=teacher.tid)=1
```

54. To find all teachers that teaches more than one subjects

```
select * from teacher where(select count(subID) from teaches where  
teaches.tid=teacher.tid)>1
```

55. To find the average teachers' salaries of those departments where the average salary is greater than 10000 without having clause

```
select department,avg_salary from (select department,avg(salary) as  
avg_salary) from teacher group by department where avg_salary>10000
```

56. To find teacher's name and number of subjects taught by him without join

```
select tname,(select count(subID) from teaches where  
teaches.tid=teacher.tid)from teacher;
```

57. Write a query to copy all records from table 'teacher' to another table 'teacherCopy': suppose the teacherCopy is already created with same structure with teacher

```
insert into teacherCopy select * from teacher;
```

58. Write a SQL to increase salary of each teacher working in 'Civil' department by 5%

```
update teacher set salary=salary*1.05 where department='Civil'
```

59. Change name and tcode of a teacher of tid=101 such that new name and code will be 'Sita Devi Thapa' and 'SDT' respectively

```
update teacher set tname='Sita Devi Thapa' , tcode='SDT' where tid=101
```

60. Write sql to delete all teachers of department 'Mechanical'

-----data may not be deleted if teacher of department teaches some subjects. Delete first from teaches and then from teacher

```
delete from teacher where department='Mechanical'
```

61. Write sql to remove all teachers from the table teacherCopy

```
delete from teacherCopy;
```

62. Write sql to remove table teacherCopy

```
drop table teacherCopy;
```

63. To find name of teachers and students(if same name exist in both table, show once)

```
(select tname from teacher) union (select name from student)
```

64. To find name of teachers and students(if same name exist in both table, show all)

```
(select tname from teacher) union all (select name from student)
```

65. To find common name of teachers and students

```
(select tname from teacher) intersect (select name from student)
```

66. To find name of teachers that match with name of student;

```
(select tname from teacher) except (select name from student)
```

67. To find cartisian product of table 'teacher' and 'teaches'

```
select * from teacher,teaches;
```

68. To find teacher's name and his teaching subject's code

```
select tname as [teacher name],subID as [Subject Code] from teacher,teaches
where teacher.tid=teaches.tid;
```

--or

```
select tname as [teacher name],subID as [Subject Code] from teacher inner
join teaches on teacher.tid=teaches.tid;
```

69. Find teacher's name and his/her teaching subject's name

```
select tname as [teacher name], subName as [Subject Name] from teacher
inner join teaches on teacher.tid=teaches.tid inner join subject on
teaches.subID=subject.subID
```

--or

```
select tname as [teacher name], subName as [Subject Name] from
teacher,teaches,subject where teacher.tid=teaches.tid and
teaches.subID=subject.subID
```

70. To find student name and his/her teacher name

```
SELECT      teacher.tname, student.name
FROM        class INNER JOIN student ON class.classID = student.classID
INNER JOIN  subjects_of_class ON class.classID = subjects_of_class.classID
INNER JOIN  subject ON subjects_of_class.subID = subject.subID INNER JOIN
teaches ON subject.subID = teaches.subID INNER JOIN teacher ON teaches.tid =
teacher.tid
```

71. To find subject's name and class information in which it is taught

```
SELECT  class.[level], class.faculty, class.year, class.semester,
subject.subname FROM class INNER JOIN subjects_of_class ON class.classID =
subjects_of_class.classID INNER JOIN subject ON subjects_of_class.subID =
subject.subID
```

72. To find name of students studying in bachelor level, computer faculty, year=2069 and semester=II

```
SELECT      student.RollNo, student.name, student.address, student.gender
FROM        student INNER JOIN
            class ON student.classID = class.classID
WHERE       class.year = 4 AND class.semester = 'II' AND class.[level] =
'Bachelor' AND class.faculty = 'Computer'
```


73. To illustrate left outer join
select * from teacher left outer join teaches on teacher.tid=teaches.tid

74. To find teachers who doesn't teach any subjects
select teacher.* from teacher left outer join teaches on teacher.tid=teaches.tid where teaches.tid is null

75. To find subjects to which any teacher has not been assigned yet.
select subject.* from teaches right outer join subject on teaches.subID=subject.subID where teaches.subID is null

76. To illustrate full outer join
select * from teacher full outer join teaches on teacher.tid=teaches.tid full outer join subject on teaches.subID=subject.subID

77. To add extra column email with data type varchar(40) in teacher table
alter table teacher add email varchar(30);

78. To drop/delete column named email
alter table teacher drop column email;

79 To change size of column tname
alter table teacher alter column tname varchar(100)

80. To add constraint in salary field of teacher table such that salary must be greater than 3000
alter table teacher add constraint sal_check check(salary>3000)

81. To change constraint sal_check such that salary must be greater than 2000
alter table teacher drop sal_check

82. To create a view teacher_view which shows only teacher's name and salary
create view teacher_view as
select tname as [Teacher's Name], salary from teacher;

83. To display records from view named teacher_view
select * from teacher_view

84. create a trigger called teacher_update which is excuted when record in teacher table is updated. It copies data into another table say teacher_log
----create a table techer_log as
create table teacher_log(tid int, table_action varchar(20), action_date datetime);
----create a trigger
create trigger trg_teacher_update on teacher
for update
as
declare @id int
select @id=i.tid from inserted i
insert into teacher_log values (@id, 'Updated', getdate())
---update a record in teacher table
update teacher set salary=7000 where tid=101
---show effect of trigger in teacher_log table
select * from teacher_log