

Student DB

STUDENT DATABASE

FACULTY

FID	Fname	Lname	Dob	Doj	Gender	Address	City	Salary	Designation	Dno
-----	-------	-------	-----	-----	--------	---------	------	--------	-------------	-----

DEPARTMENT

Dno	Dname	Dloc	Hod_Fid	Budget
-----	-------	------	---------	--------

STUDENT

USN	Fname	Lname	Gender	Address	Sem	Cell_no	Bdate	Dno
-----	-------	-------	--------	---------	-----	---------	-------	-----

ENGAGES

Fid	Sub_code	Hours
-----	----------	-------

SUBJECT

Sub_code	Sname	Type	Credit	FID
----------	-------	------	--------	-----

SCORE

USN	Sub_code	Test_no	IA_Marks
-----	----------	---------	----------

TEST

Test_no	Date	Time
---------	------	------

ATTENDENCE

USN	Sub_Code	Date	Time	Status	Reason
-----	----------	------	------	--------	--------

PARENT

USN	Pname	Address	City	Pincode	Cell_no
-----	-------	---------	------	---------	---------

Queries Used to CREATE and INSERT

```
create table faculty(fid int,fname varchar (20) not null,lname varchar (20),dob date,doj date,gender
char (1),address varchar (20),city varchar (20),salary int,designation varchar (20),dno int,primary
key(fid));

create table department(dno int,dname varchar (20) not null,dloc varchar (20),hod_fid int,primary
key(dno),foreign key(hod_fid) references faculty(fid));

create table student(usn varchar (10),fname varchar (20) not null,lname varchar (20),gender char
(1),address varchar (10),sem int not null,mob_no int,bdate date,
dno int,primary key(usn),foreign
key(dno) references department(dno));

create table subject(sub_code varchar (8),sname varchar (20) not null,type varchar (10) not
null,credit int,fid int,primary key(sub_code)foreign key(fid) references faculty(fid));

create table engages (fid int,sub_code varchar (10),hours int,primary key (fid, sub_code),foreign
key(fid) references faculty(fid),foreign key(sub_code) references subject(sub_code));

create table test(test_no int,ia_date date, time int,primary key(test_no));

create table score(usn varchar (10),sub_code varchar (10),test_no int,ia_marks int,primary key (usn,
sub_code, test_no),foreign key(usn) references student(usn),foreign key(sub_code) references
subject(sub_code),foreign key(test_no) references test(test_no));
```

```

create table attendance(usn varchar (10),sub_code varchar (8),at_date date,at_time int,status char
(1),primary key (usn, sub_code, at_date),foreign key(usn) references student(usn),foreign key(sub_code)
references subject(sub_code));

create table parent(usn varchar (10),pname varchar (15),address varchar (15),city varchar (15),pincode
int,state varchar (15),primary key (usn, pname));

alter table faculty add foreign key(dno)references department(dno);

insert into Faculty values('N007','Priya','Rao','2002-08-09','2018-06-01','F','Yelahanka,3rd
Street','Banglore',45000,'Asst.Prof',NULL);

insert into Faculty values('S098','Rakesh','Prabhu','1967-08-09','2018-06-01','M','Hebbal,4th
Street','Banglore',55000,'Prof',NULL);

...
insert into Department values(7,'ISE','1st Floor','K69',50000);

...
insert into Student values('1NT20IS009','Adithya','Borkar','M','Sulya','5A',807445654,'2002-08-01',3);
insert into Student values('1NT20IS005','Abhishek','Maurya','M','UP','5A',123456789,'2002-05-14',3);

...

```

WEBSITE USED FOR MYSQL

Link : <http://sqlfiddle.com/>

Queries

- Display the details of Faculty who were joined during 2000's.

```

select* from Faculty where extract(year from DOB) = "2002";

```

FID	Fname	Lname	Dob	Doj	Gender	Address	City	Salary	Designation	Dno
4	SUSHANT	M	06-NOV-86	12-JAN-10	M	AYODHYANAGAR	ATHANI	65000	ASCO.PROF	30
6	AMRUTHA	D	24-FEB-94	02-SEP-16	F	CENNAMANAGAR	BELGAVI	50000	ASST.PROF	40

- Show the effect of giving of 20% raise in salary where faculty ID is 2.

```

select salary+(salary*0.2) from Faculty where FID=2 ;

```

salary*1.2
72000

- Retrieve all Parent details whose address pin code ends with 9.

```
select* from Parent where right(Pincode,1)=9;
```

USN	Pname	Address	City	Pincode	Cell_no
2GI18CS003	PRATHMESH	RAMNAGAR	HUBLI	645389	KARNATAKA
2GI18CS004	SUNITA	SHANTINAGAR	ATHANI	834589	KARNATAKA
2GI18CS005	SANDYA	SHREENAGAR	GOKAK	756389	KARNATAKA

4. Compute the total number of male and female students in each semester.

```
select gender,count(gender) as count
from Student
group by gender;
```

Gender	COUNT
M	2
F	3

5. List all the student details studying in fourth semester 'C' section.

```
Select* from Student where Sem='4C';
```

6. Compute the total number of male and female students in each semester and in each section.

```
select count(distinct Sem) as male_count
from Student where Gender='M'
union
select count(distinct Sem) as female_count
from Student where Gender='F';
```

SQL Trigger | Student Database

Trigger: A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

Syntax

```
create trigger [trigger_name]
[before | after]
{insert | update | delete}
on [table_name]
[for each row]
```

```
[trigger_body]
```

Explanation of syntax:

- **create trigger [trigger_name]:** Creates or replaces an existing trigger with the trigger_name.
- **[before | after]:** This specifies when the trigger will be executed.
- **{insert | update | delete}:** This specifies the DML operation.
- **on [table_name]:** This specifies the name of the table associated with the trigger.
- **[for each row]:** This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
- **[trigger_body]:** This provides the operation to be performed as trigger is fired

BEFORE and AFTER of Trigger:

BEFORE triggers run the trigger action before the triggering statement is run. **AFTER** triggers run the trigger action after the triggering statement is run.

For More on Triggers : <https://www.geeksforgeeks.org/sql-trigger-student-database/>

7. Calculate the finalIA of the student as soon as his details are updated to the database. (use Triggers).

```
Create trigger finalIA_Update
After update on Student
For each row
Set Student.FinalIA = Student.sub1+Student.sub2+Student.sub3/3;
```

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the CREATE VIEW statement.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

SQL CREATE VIEW Examples

The following SQL creates a view that shows all customers from Brazil:

```
CREATE VIEW [Brazil Customers] AS
```

```
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = 'Brazil';
```

We can query the view above as follows:

```
SELECT * FROM [Brazil Customers];
```

CustomerName	ContactName
Comércio Mineiro	Pedro Afonso
Familia Arquibaldo	Aria Cruz
Gourmet Lanchonetes	André Fonseca
Hanari Carnes	Mario Pontes
Que Delícia Bernardo	Batista
Queen Cozinha	Lúcia Carvalho
Ricardo Adocicados	Janete Limeira
Tradição Hipermercados	Anabela Domingues
Wellington Importadora	Paula Parente

SQL Dropping a View

A view is deleted with the DROP VIEW statement.

SQL DROP VIEW Syntax

```
DROP VIEW view_name;
```

The following SQL drops the "Brazil Customers" view:

```
DROP VIEW [Brazil Customers];
```

For More on Views : https://www.w3schools.com/sql/sql_view.asp

8. Create a view of Test1 marks of student USN '1HK18IS003' in all Courses.

```
create view test1_marks as
select testno,courses from test
where usn = '1hk18is003';
```