

Assignment 1

```
create table Course( Course_id varchar(10), Title varchar(30),Dept_Name varchar(20),Credits int);
INSERT INTO Course(Course_id,Title,Dept_name,Credits) values('BIO-101','Intro. to
Biology','Biology',4);
```

```
create table Department( Dept_name varchar(20),Building varchar(20), Budget int);
INSERT INTO Department(Dept_name,Building,Budget) values('Biology','Watson',90000);
```

```
create table Instructor(ID int, Name varchar(30), Dept_name varchar(20), salary int);
INSERT INTO Instructor (ID,Name,Dept_name,salary) values(10101,'Srinivasan','Comp.Sci.',65000);
```

```
create table Prereq( Course_id varchar(10),Prereq_id varchar(10));
INSERT INTO Prereq(Course_id,Prereq_id) values('BIO-301','BIO-101');
```

```
create table Teaches( ID int, course_id varchar(10),sec_id int,semester varchar(10), year int);
INSERT INTO Teaches(ID,course_id,sec_id,semester,year) values(10101,'CS-101',1,'FALL',2009);
```

Assignment 2

1. Write the SQL Expressions for the following SQL Queries and verify it in your PC by executing it.

a) Display the structure of all the tables you have created in assignment (1).

< IN Oracle SQL>

```
describe instructor;
describe Department;
describe Prereq;
describe Teaches;
describe course;
```

<In Microsoft SQL server>

```
exec sp_columns Instructor;
exec sp_columns Department;
exec sp_columns Prereq;
exec sp_columns Teaches;
exec sp_columns course;
```

b) Display the contents of all the tables you have created in assignment (1).

```
select * from instructor;
select * from Department;
select * from Prereq;
select * from Teaches;
select * from course;
```

c) Display the name and department of each instructor.

```
select name,dept_name from instructor;
```

d) Display the name and salary of Comp. Sci. Instructors.

```
select name,salary from instructor where dept_name='Comp.Sci.';
```

e) Display the records of instructor who belongs to Physics department and getting salary less than 90000.

```
select * from instructor where dept_name='Physics' and salary<90000;
```

f) Display the name of the instructors who do not belong to Comp. Sci. Department.

```
select * from instructor where dept_name!='Comp.Sci.';
```

g) Display the names of the different department distinctly available in Instructor table.

```
select distinct dept_name from instructor;
```

h) Display all Course_id's that are taught in Spring semester of 2009.

```
select course_id from Teaches where semester='Spring' and year=2009;
```

i) Display course titles that are taught in Comp. Sci. Department and do not have credit equals to 3.

```
select title from course where credits!=3;
```

j) Display all columns of course table sorted in descending order of department names.

```
select * from course order by dept_name DESC;
```

k) Add a date_of_join column to instructor table.

```
Alter table instructor add date_of_join DATE;
```

l) Insert date values to existing rows.

```
update Instructor set date_of_join='2022-05-16';
```

m) Change the name of dept_name to department in all the tables

<oracle sql command>

```
ALTER TABLE instructor RENAME COLUMN dept_name TO department;
```

<ms sql command>

```
sp_rename 'dbo.Instructor.Dept_name','department';
```

n) Change the name of "Prereq" table with new name "Prerequisite"

<ms sql command>

```
sp_rename 'dbo.Prereq','Prerequisite';
```

o) Change Course_id column name to Sub_code.

```
sp_rename 'dbo.Course.Course_id','Sub_code';
```

p) Change the data type of name to varchar (50).

```
ALTER TABLE instructor Alter COLUMN name varchar(50);
```

q) Change the name of Instructor table to Faculty_Info.

```
sp_rename 'dbo.Instructor','Faculty_Info';
```

r) Change the Column size of Course_id in Course table from 10 to 8.

```
ALTER TABLE course Alter COLUMN course_id varchar(8);
```

on testing

s) Delete the content of the table Prereq along with its description.

```
truncate prereq;
```

t) Change the column name 'Building' of Department table by new column name 'Builder'.

```
ALTER TABLE Department RENAME COLUMN Building TO Builder;
```

Assignment 3

a. Display the Course_ids, Titles and Credits of course that are offered in any of the departments namely: Physics, Music, Finance and Biology.

```
select course_id, title, credits from course where dept_name = ANY  
( 'Biology', 'Finance', 'Physics', 'Music' );
```

b. Display records of the instructors whose name starts with “K” and who get salary more than 65000.

```
select * from instructor where name Like 'K%' AND salary>65000;
```

c. Display name, department, gross salary and net salary of instructors with 105% DA, 20% HRA, 30% IT. (gross salary = salary + DA + HRA, net salary = gross salary – IT).

```
select name,  
       Department, (salary+1.05*salary+0.2*salary) as 'gross salary',  
       (salary+1.05*salary+0.2*salary)-0.3*salary as 'net salary'  
from Faculty_Info;
```

d. Display records of the Instructors with salary range 60000 to 80000.

```
select * from instructor where salary between 60000 and 80000;
```

e. Display the records of the instructors having the second letter in their name as ‘r’.

```
select * from instructor where name Like '_r%';
```

f. Display the names of the instructors of Comp.Sci. Department in the descending order of their salary.

```
select name from Faculty_Info where department='Comp.Sci.' order by salary desc;
```

g. Update all records of Instructor table with a salary hike of 15%.

```
update Faculty_Info set salary=salary+0.15*salary;
```

h. Update the records with a salary hike of 3% for Comp.Sci. Dept instructors having salary less than 70000.

```
update Faculty_Info set salary=salary+0.03*salary where department='Comp.Sci.' and salary<70000;
```

i. Display the annual salary of each instructor.

```
select salary*12 as 'annual salary' from Faculty_Info;
```

j. Update the title of the course having title 'Game Design' to 'Game Theory'

```
update Course set Title='Game Theory' where Title='Game Design';
```

k. Delete the instructor records of History department.

```
delete from Faculty_Info where department='History';
```

l. Delete the course records of the courses having course_id starting with 'BIO'

```
delete from Course where department like 'bio%';
```

update in instructor table of assignment 1

```
INSERT INTO Faculty_Info(ID,Name,department,salary,date_of_join)  
values(22222,'Einstein','Physics',95000,'2022-05-16');
```

--c) Display the ID, Name & Department of the instructor drawing the highest salary.

```
select id,name,Department,salary from Faculty_Info where salary=(select max(salary) from  
Faculty_Info);
```

--d) Display the number of instructors available in Comp. Sci. Department.
select count(department) from Faculty_Info where department='Comp.Sci.';

--e) Display the total credits of all courses offered in Comp.Sci. Department.
select sum(credits) from Course where department='Comp. Sci.';

--f) Display the number of instructors and total salary drawn by Physics and Comp.Sci.departments.
select count(name), sum(salary) from Faculty_Info where department in ('Physics', 'Comp.Sci.');

--g) Display the total credits of Comp.Sci. and Biology departments from course table.
select sum(credits) from Course where department in ('Comp. Sci.','Biology');

--h) Display building wise total budget values.
select building,sum(budget) as budget from Department group by Building;

3. Write the SQL Expressions for the following queries using suitable SQL scalar function.

=====on testing=====

a) Display your name with first letter being capital, where the entered name is in lower case.
mysql cmd-> Select CONCAT(UPPER(SUBSTRING(name,1,1)),LOWER(SUBSTRING(name,2))) AS Name
from table_name;

b) Display 2nd- 6th characters of your name.
mysql cmd-> Select CONCAT(SUBSTRING(name,2,1),LOWER(SUBSTRING(name,6,1))) from table_name;

c) Find length of your full university name.
select len(university_name) from Table_name

d) Display all the Instructor names with its first letter in upper case.
mysql cmd-> Select CONCAT(UPPER(SUBSTRING(name,1,1)),LOWER(SUBSTRING(name,2))) AS Name
from Instructor;

e) List the department name of each instructor as a three letter code.
-> mysql cmd-> Select SUBSTRING(name,1,3) AS Name from Instructor;

f) Display the month of the joining of each instructor.
-> SELECT EXTRACT(MONTH FROM date) from table_name

j) Display the date of the next Friday after today's date.
-> SELECT DATEADD(DAY, 13 - (@@DATEFIRST + (DATEPART(WEEKDAY,GETDATE()) %7)), GETDATE())

k) Display the day of joining of each instructor.
-> SELECT EXTRACT(Day FROM joining_date) from table_name

l) Display the date corresponding to 15 days after today's date.
-> declare @thirtydaysago datetime
declare @now datetime
set @now = getdate()
set @thirtydaysago = dateadd(day,-30,@now)

select @now, @thirtydaysago

m) Display the value 94204.27348 truncated up to 2 digits after decimal point.
DECLARE @i FLOAT = 6.677756

SELECT
ROUND(@i, 2)

```
, FORMAT(@i, 'N2')
, CAST(@i AS DECIMAL(18,2))
, SUBSTRING(PARSENAME(CAST(@i AS VARCHAR(10)), 1), PATINDEX('%.%', CAST(@i AS
VARCHAR(10))) - 1, 2)
, FLOOR((@i - FLOOR(@i)) * 100)
```

FORMAT (value, format [, culture])

<https://stackoverflow.com/questions/16412231/how-to-get-2-digits-after-decimal-point-in-tsql>

n) select 5+pow(8,9)

o) select sqrt(6464312);

Assignment 4

1a

```
create table customer (
cust_no varchar(5) primary key,check(cust_no like 'c%') ,
name varchar(30) not null,
phone_no varchar(10),
city varchar(30) not null) ;
```

exec sp_columns customer;

1b

```
create table Account(
account_no varchar(5) primary key,
check(account_no like 'A%'),
Type varchar(2),check(Type in('SB','FD','CA')),
balance int,check(balance<10000000),
branch_code varchar(5),
FOREIGN KEY (branch_code) REFERENCES Branch(branch_code));
```

1c

```
create table DEPOSITOR(
cust_no varchar(5),
account_no varchar(5),
FOREIGN KEY (cust_no) REFERENCES Customer(cust_no),
FOREIGN KEY (account_no) REFERENCES Account(account_no)
);
exec sp_columns DEPOSITOR;
```

1d

```
create table LOAN(
loan_no varchar(5) primary key,check(loan_no like 'l%'),
cust_no varchar(5),
amount int,check(amount>1000),
branch_code varchar(5),
FOREIGN KEY (cust_no) REFERENCES Customer(cust_no),
FOREIGN KEY (branch_code) REFERENCES Branch(branch_code)
);
exec sp_columns LOAN;
```

1e

```
create table branch(
branch_code varchar(5) primary key,
```

```
branch_name varchar(30) not null,
branch_city varchar(20),check(branch_city in ('Delhi','Mumbai','Kolkata','Chennai')));
exec sp_columns branch;
```

```
1f
create table installment(
inst_no int primary key,check(inst_no<=10),
loan_no varchar(5),
FOREIGN KEY (loan_no) REFERENCES Loan(loan_no),
inst_amount int not null
);
exec sp_columns installment;
```

Assignment 5

1a
select name,phone_no,cust_no from customer
where cust_no=(select cust_no from DEPOSITOR where account_no='A0004');

1b
select name from customer
where cust_no not in (select cust_no from LOAN);

1c
select branch_city from branch
where branch_code=(select branch_code from LOAN
where cust_no=(select cust_no from customer where name='ASLESHA TIWARI'));

--d) Find out the installment details of customer named "ANKITA SINGH"
select * from installment where loan_no=(
select loan_no from loan where cust_no=(
select cust_no from customer where name='ANKITA SINGH'
)
);

1e
--Find out the branch name and branch city, in which "ABHIJIT MISHRA" has an account.
select branch_name,branch_city from branch where
branch_code in(select branch_code from Account where
account_no in (select account_no from DEPOSITOR where
cust_no=(select cust_no from customer where name='ABHIJIT MISHRA'))
);

2
--a) Find out the Loan_nos where the loans are taken from
--any branch with branch_city=MUMBAI.
select loan.loan_no from loan inner join branch on branch.branch_code=loan.branch_code
where branch_city='mumbai';

--b) Find the Type of the accounts available in any branch with branch_city =DELHI.
select Account.Type from Account inner join branch on
branch.branch_code=Account.branch_code where branch_city='delhi';

--c) Find out Installment_no and Installment amount of customer with Name= RAJ ANAND SINGH.
select installment.inst_no, installment.inst_amount from installment

```
inner join loan on installment.loan_no=loan.loan_no
inner join customer on loan.cust_no=customer.cust_no
where customer.name='RAJ ANAND SINGH';
```

--d) Find out the Name of the customers who have paid installments of Amount 50000 against his/her loan

```
select customer.name from customer
inner join loan on customer.cust_no=loan.cust_no
inner join installment on loan.loan_no=installment.loan_no
where installment.inst_amount=50000;
```

--e) Find out the Ph_no of customers having account at branch with Branch_name equal to SALT LAKE.

```
select customer.phone_no from customer
inner join DEPOSITOR on customer.cust_no=DEPOSITOR.cust_no
inner join Account on Account.account_no=DEPOSITOR.account_no
inner join branch on branch.branch_code=Account.branch_code
where branch.branch_name='SALT LAKE branch';
```