PONDICHERRY UNIVERSITY
(A Central university)

SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
M.Sc. Integrated Computer Science

NAME	:	AANISHA ALMAAZ S
REG. NO.	:	
SEMESTER	:	VIII - Semester
SUBJECT	:	CSSC 424 – DATABASE SYSTEMS LAB

PONDICHERRY UNIVERSITY
(A Central university)

SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
M.Sc. Integrated Computer Science

PRACTICAL LAB RECORD

BONAFIDE CERTIFICATE

This is to certify that this is a Bonafide record of practical work done by **AANISHA ALMAAZ S**, having Reg. No.   semester - VIII from the month February 2024 to June 2024.

FACULTY IN-CHARGE

SUBMITTED FOR THE PRACTICAL EXAM HELD ON: _______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

EXPERIMENT 1- PRCATICE SQL

```
create database Ads_8;
use Ads_8;

create table salesman(salesman_id integer primary key, name text not null, city text , commision
decimal(2,2));
alter table salesman rename column commision to commission;

insert into salesman values(5001,'James hoog','New york',0.15),(5002,'nail knite','Paris',0.13),
(5005,'Pit Alex','London',0.11),(5006,'Mc Lyon','Paris',0.14),
(5003,'Lauson Ken','',0.12),(5007,'Paul Adam','Rome',0.13);

select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James hoog | New york |       0.15 |
|        5002 | nail knite | Paris    |       0.13 |
|        5003 | Lauson Ken |          |       0.12 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
6 rows in set (0.010 sec)



create table customer(customer_id integer primary key, customer_name text not null, city text ,
grade integer, salesman_id integer,
foreign key(salesman_id) references salesman(salesman_id));

insert into customer values(3002,'Nick Rink','New York',100,5001),(3005,'Graham
Bell','California',200,5002),(3001,'Brad Pitt','London',null,null),(3004,'Fabio
Carl','London',300,5006),
(3007,'James Bond','Minnesota',200,5001),(3008,'Joey Mann','London',200,5007),(3009,'Geff
Matt','Berlin',100,null);

select * from customer;
+-------------+---------------+------------------------------+-------+-------------+
| customer_id | customer_name | city                         | grade | salesman_id |
+-------------+---------------+------------------------------+-------+-------------+
|        3001 | Brad Pitt     | London                       |  NULL |        NULL |
|        3002 | Nick Rink     | New York                     |   100 |        5001 |
|        3004 | Fabio Carl    | London                       |   300 |        5006 |
|        3005 | Graham Bell   | California                   |   200 |        5002 |
|        3007 | James Bond    | Minnesota                    |   200 |        5001 |
|        3008 | Joey Mann     | London                       |   200 |        5007 |
|        3009 | Geff Matt     | Berlin                       |   100 |        NULL |
|        3011 | Micky Mouse   | currently delivery unavilable |   290 |        5005 |
+-------------+---------------+------------------------------+-------+-------------+
8 rows in set (0.017 sec)
```

```
create table orders (order_no integer primary key,purch_amt decimal(6,2) not null, order_date date not null, customer_id integer,salesman_id integer,
foreign key(customer_id) references customer(customer_id), foreign key(salesman_id) references salesman(salesman_id));

insert into orders values(7001,150.5,'2021-10-05',3005,5002),(7009,279.65,'2021-10-05',3001,null),(7002,65.26,'2021-11-01',3002,5001),(7004,110.5,'2021-11-03',3009,null),(7007,986.6,'2021-11-05',3005,5002),
    (7005,2400.8,'2021-11-10',3007,5001),(7008,5760,'2021-11-29',3002,5001),(7012,2480.7,'2021-12-12',3009,null);
update orders set salesman_id = 5007 where order_no = 7012;

select* from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|     7001 |    150.50 | 2021-10-05 |        3005 |        5002 |
|     7002 |     65.26 | 2021-11-01 |        3002 |        5001 |
|     7004 |    110.50 | 2021-11-03 |        3009 |        NULL |
|     7005 |   2400.80 | 2021-11-10 |        3007 |        5001 |
|     7007 |    986.60 | 2021-11-05 |        3005 |        5002 |
|     7008 |   5760.00 | 2021-11-29 |        3002 |        5001 |
|     7009 |    279.65 | 2021-10-05 |        3001 |        NULL |
|     7012 |   2480.70 | 2021-12-12 |        3009 |        5007 |
+----------+-----------+------------+-------------+-------------+
8 rows in set (0.001 sec)
```

```
-- query 1: Display name and commission of all the salesmen.
select name, commission from salesman;
+------------+------------+
| name       | commission |
+------------+------------+
| James hoog |       0.15 |
| nail knite |       0.13 |
| Lauson Ken |       0.12 |
| Pit Alex   |       0.11 |
| Mc Lyon    |       0.14 |
| Paul Adam  |       0.13 |
+------------+------------+
6 rows in set (0.000 sec)
```

```
-- query 2: Retrieve salesman id of all salesmen from orders table without any repeats.
select distinct salesman_id from orders;
+-------------+
| salesman_id |
+-------------+
|        NULL |
|        5001 |
|        5002 |
|        5007 |
+-------------+
```

```
-- query 3: Display names and city of salesman, who belongs to the city of Paris.
select name, city from salesman where city ='Paris';
+------------+-------+
| name       | city  |
+------------+-------+
| nail knite | Paris |
| Mc Lyon    | Paris |
+------------+-------+
2 rows in set (0.000 sec)
```

```
-- query 4: Display all the information for those customers with a grade of 200.
select * from customer where grade=200;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3005 | Graham Bell   | California |   200 |        5002 |
|        3007 | James Bond    | Minnesota  |   200 |        5001 |
|        3008 | Joey Mann     | London     |   200 |        5007 |
+-------------+---------------+------------+-------+-------------+
3 rows in set (0.000 sec)
```

```
-- query 5: Display the order number, order date and the purchase amount for order(s) which will be
delivered by the salesman with ID 5001.
select order_no,order_date,purch_amt from orders where salesman_id=5001;
+----------+------------+-----------+
| order_no | order_date | purch_amt |
+----------+------------+-----------+
|     7002 | 2021-11-01 |     65.26 |
|     7005 | 2021-11-10 |   2400.80 |
|     7008 | 2021-11-29 |   5760.00 |
+----------+------------+-----------+
3 rows in set (0.000 sec)
```

```
-- query 6: Display all the customers, who are either belongs to the city New York or not had a
grade above 100.
```

```
select * from customer where city= 'New York' or not grade > 100;
+-------------+---------------+----------+-------+-------------+
| customer_id | customer_name | city     | grade | salesman_id |
+-------------+---------------+----------+-------+-------------+
|        3002 | Nick Rink     | New York |   100 |        5001 |
|        3009 | Geff Matt     | Berlin   |   100 |        NULL |
+-------------+---------------+----------+-------+-------------+
2 rows in set (0.000 sec)


-- query 7: Find those salesmen with all information who gets the commission within a range of
0.12 and 0.14.
select * from salesman where commission >=0.12 and commission <=0.14;
-- can also use between clause
+-------------+------------+-------+------------+
| salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|        5002 | nail knite | Paris |       0.13 |
|        5003 | Lauson Ken |       |       0.12 |
|        5006 | Mc Lyon    | Paris |       0.14 |
|        5007 | Paul Adam  | Rome  |       0.13 |
+-------------+------------+-------+------------+
4 rows in set (0.001 sec)


-- query 8: Find all those customers with all information whose names are ending with the letter 'n'.
select * from customer where customer_name like '%n';
+-------------+---------------+--------+-------+-------------+
| customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3008 | Joey Mann     | London |   200 |        5007 |
+-------------+---------------+--------+-------+-------------+
1 row in set (0.000 sec)


-- query 9: Find those salesmen with all information whose name containing the 1st character is 'N'
and the 4th character is 'l' and rests may be any character.
select * from salesman where name like 'n_i%';
+-------------+------------+-------+------------+
| salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|        5002 | nail knite | Paris |       0.13 |
+-------------+------------+-------+------------+
1 row in set (0.000 sec)


-- query 10: Find that customer with all information who does not get any grade except NULL.
select * from customer where grade is null;
+-------------+---------------+--------+-------+-------------+
| customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3001 | Brad Pitt      | London |  NULL |        NULL |
+-------------+---------------+--------+-------+-------------+
1 row in set (0.000 sec)
```

```
-- query 11: Find the total purchase amount of all orders.
select sum(purch_amt) from orders;
+----------------+
| sum(purch_amt) |
+----------------+
|       12234.01 |
+----------------+
1 row in set (0.000 sec)
```

```
-- query 12: Find the number of salesman currently listing for all of their customers.
select count( distinct salesman_id) from orders;
+------------------------------+
| count( distinct salesman_id) |
+------------------------------+
|                            3 |
+------------------------------+
1 row in set (0.008 sec)
```

```
-- query 13:  Find the highest grade for each of the cities of the customers.
select city, max(grade) from customer group by city;
+------------------------------+------------+
| city                         | max(grade) |
+------------------------------+------------+
| Berlin                       |        100 |
| California                   |        200 |
| currently delivery unavilable |        290 |
| London                       |        300 |
| Minnesota                    |        200 |
| New York                     |        100 |
+------------------------------+------------+
6 rows in set (0.000 sec)
```

```
-- query 14: Find the highest grade for each of the cities of the customers.
select customer_id, max(purch_amt) from orders group by customer_id;
+-------------+----------------+
| customer_id | max(purch_amt) |
+-------------+----------------+
|        3001 |         279.65 |
|        3002 |        5760.00 |
|        3005 |         986.60 |
|        3007 |        2400.80 |
|        3009 |        2480.70 |
+-------------+----------------+
5 rows in set (0.001 sec)
```

-- query 15: Find the highest purchase amount ordered by the each customer on a particular date with their ID, order date and highest purchase amount.
select customer_id, order_date, max(purch_amt) from orders group by customer_id, order_date;

```
+-------------+------------+----------------+
| customer_id | order_date | max(purch_amt) |
+-------------+------------+----------------+
|        3001 | 2021-10-05 |         279.65 |
|        3002 | 2021-11-01 |          65.26 |
|        3002 | 2021-11-29 |        5760.00 |
|        3005 | 2021-10-05 |         150.50 |
|        3005 | 2021-11-05 |         986.60 |
|        3007 | 2021-11-10 |        2400.80 |
|        3009 | 2021-11-03 |         110.50 |
|        3009 | 2021-12-12 |        2480.70 |
+-------------+------------+----------------+
8 rows in set (0.001 sec)
```

-- query 16: Find the highest purchase amount on a date '2021-11-01' for each salesman with their ID
select salesman_id, purch_amt from orders where order_date ='2021-11-01' group by salesman_id;

```
+-------------+-----------+
| salesman_id | purch_amt |
+-------------+-----------+
|        5001 |     65.26 |
+-------------+-----------+
1 row in set (0.000 sec)
```

-- query 17: Find the highest purchase amount with their customer ID and order date, for only those customers who have the
--   highest purchase amount in a day is more than 2000.
select customer_id, order_date ,purch_amt from orders group by customer_id, order_date having max(purch_amt)>2000;

```
+-------------+------------+-----------+
| customer_id | order_date | purch_amt |
+-------------+------------+-----------+
|        3002 | 2021-11-29 |   5760.00 |
|        3007 | 2021-11-10 |   2400.80 |
|        3009 | 2021-12-12 |   2480.70 |
+-------------+------------+-----------+
3 rows in set (0.000 sec)
```

```
-- query 18:  Write a SQL statement that counts all orders for a date 2021-11-10.
select count(*) from orders where order_date ='2021-11-10';
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.000 sec)
```

EXPERIMENT 2- PROCEDURE,TRIGGERS AND CURSOR

```
-- EXPERIMENT 2


-- PROCEDURE !!!!
drop procedure query_db;
delimiter //
create procedure query_db (in o_date date , out val int)
begin
	select sum(purch_amt) into val from orders where order_date >= o_date;

end //
delimiter ;

call query_db('2021-11-05', @val);
select @val;
+-------+
| @val  |
+-------+
| 11628 |
+-------+
1 row in set (0.000 sec)



-- CURSORS!!!
delimiter //
declare @c_id integer;
declare @name text(128);
declare @city text(128);
declare @commission decimal(2,2);

-- declare cursors
declare cursor_test  cursor for select * from salesman where commission > 0.13;

-- open cursor
open cursor_test;
-- loop through a cursor
fetch next from  cursor_test into @s_id, @name, @city, @commission;
while @@fetchstatus =0
        begin
   print concat('id: ', @s_id, '/ name: ', @name, '/ city: ',@city, '/ commission: ',@commission);
        fetch next from  cursor_test into @s_id, @name, @city, @commission;
        end;

-- close the cursor
close cursor_test;
deallocate cursor_test;

delimiter ;


-- TRIGGER!!!!
-- drop trigger cityval;
delimiter //
create trigger cityval before insert on customer
for each row
begin
        if new.city ="Uganda" then set new.city = "currently delivery unavilable";
   end if;
end //
delimiter ;
insert into customer values(3010,'Micky me','Uganda',260,5003);
select * from customer;
+-------------+---------------+------------------------------+-------+-------------+
| customer_id | customer_name | city                         | grade | salesman_id |
+-------------+---------------+------------------------------+-------+-------------+
|        3001 | Brad Pitt     | London                       |  NULL |        NULL |
|        3002 | Nick Rink     | New York                     |   100 |        5001 |
|        3004 | Fabio Carl    | London                       |   300 |        5006 |
|        3005 | Graham Bell   | California                   |   200 |        5002 |
|        3007 | James Bond    | Minnesota                    |   200 |        5001 |
|        3008 | Joey Mann     | London                       |   200 |        5007 |
|        3009 | Geff Matt     | Berlin                       |   100 |        NULL |
|        3010 | Micky me      | currently delivery unavilable |   260 |        5003 |
|        3011 | Micky Mouse   | currently delivery unavilable |   290 |        5005 |
+-------------+---------------+------------------------------+-------+-------------+
9 rows in set (0.001 sec)
```

EXPERIMENT 3- ACCESSING THE DATABASE

```
-- create database Ads2_8;
-- use Ads2_8;


-- EXPERIMENT 3
-- 1. TABLE INSTRUCTOR
create table instructor(id integer primary key, name text, dept_name text, salary integer);
insert into instructor values(10101,"srinivasan","comp.sci",65000),(12121,"wu","finance",90000),
(15151,"mozarat","music",40000),
(22222,"einstein","physics",95000),(32343,"el said","history",60000),
(33456,"gold","physics",87000),(45565,"katz","comp.sci",75000);
insert into instructor values(58583,"califeri","history",62000),(76543,"singh","finance",80000),
(76766,"crick","biology",72000),
(83821,"brandt","comp.sci",92000),(98345,"kim","elec.eng",80000);


select* from instructor;
+-------+------------+-----------+--------+
| id    | name       | dept_name | salary |
+-------+------------+-----------+--------+
| 10101 | srinivasan | comp.sci  |  65000 |
| 10211 | smith      | biology   |  66000 |
| 10212 | tom        | biology   |   NULL |
| 12121 | wu         | finance   |  90000 |
| 15151 | mozarat    | music     |  40000 |
| 22222 | einstein   | physics   |  95000 |
| 32343 | el said    | history   |  60000 |
| 33456 | gold       | physics   |  87000 |
| 45565 | katz       | comp.sci  |  75000 |
| 58583 | califeri   | history   |  62000 |
| 76543 | singh      | finance   |  80000 |
| 76766 | crick      | biology   |  72000 |
| 83821 | brandt     | comp.sci  |  92000 |
| 98345 | kim        | elec.eng  |  80000 |
+-------+------------+-----------+--------+
14 rows in set (0.001 sec)
```

```
-- 2. TABLE TEACHES
create table teaches(id integer,course_id text not null, sec_id integer, semester text, year integer(4),
 foreign key(id) references instructor(id));
insert into teaches values(10101,"CS-101",1,"fall",2017),(10101,"CS-315",1,"spring",2018),
(10101,"CS-347",1,"fall",2017),
(12121,"FIN-201",1,"spring",2018),(15151,"MU-199",1,"spring",2018),(22222,"PHY-
101",1,"fall",2017),(10101,"CS-101",1,"spring",2018),
(32343,"HIS-351",1,"spring",2018),(45565,"CS-319",1,"spring",2018),(45565,"CS-
319",1,"spring",2017),(76766,"BIO-101",1,"summer",2018),
(76766,"BIO-301",1,"summer",2017),(83821,"CS-190",1,"spring",2017),(83821,"CS-
190",2,"spring",2017),(83821,"CS-319",2,"spring",2018),
(98345,"EE-181",1,"spring",2017);

select* from teaches;
+-------+-----------+--------+----------+------+
| id    | course_id | sec_id | semester | year |
+-------+-----------+--------+----------+------+
| 10101 | CS-101    |      1 | fall     | 2017 |
| 10101 | CS-315    |      1 | spring   | 2018 |
| 10101 | CS-347    |      1 | fall     | 2017 |
| 12121 | FIN-201   |      1 | spring   | 2018 |
| 15151 | MU-199    |      1 | spring   | 2018 |
| 22222 | PHY-101   |      1 | fall     | 2017 |
| 10101 | CS-101    |      1 | spring   | 2018 |
| 32343 | HIS-351   |      1 | spring   | 2018 |
| 45565 | CS-319    |      1 | spring   | 2018 |
| 45565 | CS-319    |      1 | spring   | 2017 |
| 76766 | BIO-101   |      1 | summer   | 2018 |
| 76766 | BIO-301   |      1 | summer   | 2017 |
| 83821 | CS-190    |      1 | spring   | 2017 |
| 83821 | CS-190    |      2 | spring   | 2017 |
| 83821 | CS-319    |      2 | spring   | 2018 |
| 98345 | EE-181    |      1 | spring   | 2017 |
+-------+-----------+--------+----------+------+
16 rows in set (0.000 sec)
```

```
-- 3. Insert following additional tuple in instructor ('10211', 'Smith', 'Biology', 66000)
insert into instructor values(10211,"smith","biology",66000);
Query OK, 1 row affected (0.001 sec)
```

```
-- 4. Delete this tuple from instructor ('10211', 'Smith', 'Biology', 66000)
delete from instructor where id =10211;
Query OK, 1 row affected (0.001 sec)
```

```
-- 5. Select tuples from instructor where dept_name = ‘History’
select* from instructor where dept_name="history";
+-------+----------+-----------+--------+
| id    | name     | dept_name | salary |
+-------+----------+-----------+--------+
| 32343 | el said  | history   |  60000 |
| 58583 | califeri | history   |  62000 |
+-------+----------+-----------+--------+
2 rows in set (0.000 sec)
```

-- 6. Find the Cartesian product instructor x teaches.

select * from instructor cross join teaches;

```
+-------+------------+-----------+--------+-------+-----------+--------+----------+------+
| id    | name       | dept_name | salary | id    | course_id | sec_id | semester | year |
+-------+------------+-----------+--------+-------+-----------+--------+----------+------+
| 10101 | srinivasan | comp.sci  |  65000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 10211 | smith      | biology   |  66000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 10212 | tom        | biology   |   NULL | 10101 | CS-101    |      1 | fall     | 2017 |
| 12121 | wu         | finance   |  90000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 15151 | mozarat    | music     |  40000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 22222 | einstein   | physics   |  95000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 32343 | el said    | history   |  60000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 33456 | gold       | physics   |  87000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 45565 | katz       | comp.sci  |  75000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 58583 | califeri   | history   |  62000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 76543 | singh      | finance   |  80000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 76766 | crick      | biology   |  72000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 83821 | brandt     | comp.sci  |  92000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 98345 | kim        | elec.eng  |  80000 | 10101 | CS-101    |      1 | fall     | 2017 |
| 10101 | srinivasan | comp.sci  |  65000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 10211 | smith      | biology   |  66000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 10212 | tom        | biology   |   NULL | 10101 | CS-315    |      1 | spring   | 2018 |
| 12121 | wu         | finance   |  90000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 15151 | mozarat    | music     |  40000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 22222 | einstein   | physics   |  95000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 32343 | el said    | history   |  60000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 33456 | gold       | physics   |  87000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 45565 | katz       | comp.sci  |  75000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 58583 | califeri   | history   |  62000 | 10101 | CS-315    |      1 | spring   | 2018 |
| 76543 | singh      | finance  |  80000 | 10101 | CS-315  |      1 | spring | 2018 |
| 76766 | crick      | biology  |  72000 | 10101 | CS-315  |      1 | spring | 2018 |
| 83821 | brandt     | comp.sci |  92000 | 10101 | CS-315  |      1 | spring | 2018 |
| 98345 | kim        | elec.eng |  80000 | 10101 | CS-315  |      1 | spring | 2018 |
| 10101 | srinivasan | comp.sci |  65000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 10211 | smith      | biology  |  66000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 10212 | tom        | biology  |   NULL | 10101 | CS-347  |      1 | fall   | 2017 |
| 12121 | wu         | finance  |  90000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 15151 | mozarat    | music    |  40000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 22222 | einstein   | physics  |  95000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 32343 | el said    | history  |  60000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 33456 | gold       | physics  |  87000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 45565 | katz       | comp.sci |  75000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 58583 | califeri   | history  |  62000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 76543 | singh      | finance  |  80000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 76766 | crick      | biology  |  72000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 83821 | brandt     | comp.sci |  92000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 98345 | kim        | elec.eng |  80000 | 10101 | CS-347  |      1 | fall   | 2017 |
| 10101 | srinivasan | comp.sci |  65000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 10211 | smith      | biology  |  66000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 10212 | tom        | biology  |   NULL | 12121 | FIN-201 |      1 | spring | 2018 |
| 12121 | wu         | finance  |  90000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 15151 | mozarat    | music    |  40000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 22222 | einstein   | physics  |  95000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 32343 | el said    | history  |  60000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 33456 | gold       | physics  |  87000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 45565 | katz       | comp.sci |  75000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 58583 | califeri   | history  |  62000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 76543 | singh      | finance  |  80000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 76766 | crick      | biology  |  72000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 83821 | brandt     | comp.sci |  92000 | 12121 | FIN-201 |      1 | spring | 2018 |
| 98345 | kim        | elec.eng   |  80000 | 12121 | FIN-201   |      1 | spring   | 2018 |
| 10101 | srinivasan | comp.sci   |  65000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 10211 | smith      | biology    |  66000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 10212 | tom        | biology    |   NULL | 15151 | MU-199    |      1 | spring   | 2018 |
| 12121 | wu         | finance    |  90000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 15151 | mozarat    | music      |  40000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 22222 | einstein   | physics    |  95000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 32343 | el said    | history    |  60000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 33456 | gold       | physics    |  87000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 45565 | katz       | comp.sci   |  75000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 58583 | califeri   | history    |  62000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 76543 | singh      | finance    |  80000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 76766 | crick      | biology    |  72000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 83821 | brandt     | comp.sci   |  92000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 98345 | kim        | elec.eng   |  80000 | 15151 | MU-199    |      1 | spring   | 2018 |
| 10101 | srinivasan | comp.sci   |  65000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 10211 | smith      | biology    |  66000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 10212 | tom        | biology    |   NULL | 22222 | PHY-101   |      1 | fall     | 2017 |
| 12121 | wu         | finance    |  90000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 15151 | mozarat    | music      |  40000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 22222 | einstein   | physics    |  95000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 32343 | el said    | history    |  60000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 33456 | gold       | physics    |  87000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 45565 | katz       | comp.sci   |  75000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 58583 | califeri   | history    |  62000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 76543 | singh      | finance    |  80000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 76766 | crick      | biology    |  72000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 83821 | brandt     | comp.sci   |  92000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 98345 | kim        | elec.eng   |  80000 | 22222 | PHY-101   |      1 | fall     | 2017 |
| 10101 | srinivasan | comp.sci   |  65000 | 10101 | CS-101    |      1 | spring   | 2018 |
| 10211 | smith      | biology    |  66000 | 10101 | CS-101    |      1 | spring   | 2018 |
| 10212 | tom        | biology    |  NULL | 10101 | CS-101   |      1 | spring   | 2018 |
| 12121 | wu         | finance    | 90000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 15151 | mozarat    | music      | 40000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 22222 | einstein   | physics    | 95000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 32343 | el said    | history    | 60000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 33456 | gold       | physics    | 87000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 45565 | katz       | comp.sci   | 75000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 58583 | califeri   | history    | 62000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 76543 | singh      | finance    | 80000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 76766 | crick      | biology    | 72000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 83821 | brandt     | comp.sci   | 92000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 98345 | kim        | elec.eng   | 80000 | 10101 | CS-101   |      1 | spring   | 2018 |
| 10101 | srinivasan | comp.sci   | 65000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 10211 | smith      | biology    | 66000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 10212 | tom        | biology    |  NULL | 32343 | HIS-351  |      1 | spring   | 2018 |
| 12121 | wu         | finance    | 90000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 15151 | mozarat    | music      | 40000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 22222 | einstein   | physics    | 95000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 32343 | el said    | history    | 60000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 33456 | gold       | physics    | 87000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 45565 | katz       | comp.sci   | 75000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 58583 | califeri   | history    | 62000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 76543 | singh      | finance    | 80000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 76766 | crick      | biology    | 72000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 83821 | brandt     | comp.sci   | 92000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 98345 | kim        | elec.eng   | 80000 | 32343 | HIS-351  |      1 | spring   | 2018 |
| 10101 | srinivasan | comp.sci   | 65000 | 45565 | CS-319   |      1 | spring   | 2018 |
| 10211 | smith      | biology    | 66000 | 45565 | CS-319   |      1 | spring   | 2018 |
| 10212 | tom        | biology    |  NULL | 45565 | CS-319   |      1 | spring   | 2018 |
| 12121 | wu         | finance    | 90000 | 45565 | CS-319   |      1 | spring   | 2018 |
| 15151 | mozarat    | music      | 40000 | 45565 | CS-319   |      1 | spring   | 2018 |
| 22222 | einstein   | physics    |  95000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 32343 | el said    | history    |  60000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 33456 | gold       | physics    |  87000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 45565 | katz       | comp.sci   |  75000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 58583 | califeri   | history    |  62000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 76543 | singh      | finance    |  80000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 76766 | crick      | biology    |  72000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 83821 | brandt     | comp.sci   |  92000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 98345 | kim        | elec.eng   |  80000 | 45565 | CS-319    |      1 | spring   | 2018 |
| 10101 | srinivasan | comp.sci   |  65000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 10211 | smith      | biology    |  66000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 10212 | tom        | biology    |   NULL | 45565 | CS-319    |      1 | spring   | 2017 |
| 12121 | wu         | finance    |  90000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 15151 | mozarat    | music      |  40000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 22222 | einstein   | physics    |  95000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 32343 | el said    | history    |  60000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 33456 | gold       | physics    |  87000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 45565 | katz       | comp.sci   |  75000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 58583 | califeri   | history    |  62000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 76543 | singh      | finance    |  80000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 76766 | crick      | biology    |  72000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 83821 | brandt     | comp.sci   |  92000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 98345 | kim        | elec.eng   |  80000 | 45565 | CS-319    |      1 | spring   | 2017 |
| 10101 | srinivasan | comp.sci   |  65000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 10211 | smith      | biology    |  66000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 10212 | tom        | biology    |   NULL | 76766 | BIO-101   |      1 | summer   | 2018 |
| 12121 | wu         | finance    |  90000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 15151 | mozarat    | music      |  40000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 22222 | einstein   | physics    |  95000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 32343 | el said    | history    |  60000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 33456 | gold       | physics    |  87000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 45565 | katz       | comp.sci  |  75000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 58583 | califeri   | history   |  62000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 76543 | singh      | finance   |  80000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 76766 | crick      | biology   |  72000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 83821 | brandt     | comp.sci  |  92000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 98345 | kim        | elec.eng  |  80000 | 76766 | BIO-101   |      1 | summer   | 2018 |
| 10101 | srinivasan | comp.sci  |  65000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 10211 | smith      | biology   |  66000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 10212 | tom        | biology   |   NULL | 76766 | BIO-301   |      1 | summer   | 2017 |
| 12121 | wu         | finance   |  90000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 15151 | mozarat    | music     |  40000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 22222 | einstein   | physics   |  95000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 32343 | el said    | history   |  60000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 33456 | gold       | physics   |  87000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 45565 | katz       | comp.sci  |  75000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 58583 | califeri   | history   |  62000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 76543 | singh      | finance   |  80000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 76766 | crick      | biology   |  72000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 83821 | brandt     | comp.sci  |  92000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 98345 | kim        | elec.eng  |  80000 | 76766 | BIO-301   |      1 | summer   | 2017 |
| 10101 | srinivasan | comp.sci  |  65000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 10211 | smith      | biology   |  66000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 10212 | tom        | biology   |   NULL | 83821 | CS-190    |      1 | spring   | 2017 |
| 12121 | wu         | finance   |  90000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 15151 | mozarat    | music     |  40000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 22222 | einstein   | physics   |  95000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 32343 | el said    | history   |  60000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 33456 | gold       | physics   |  87000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 45565 | katz       | comp.sci  |  75000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 58583 | califeri   | history   |  62000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 76543 | singh      | finance   |  80000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 76766 | crick      | biology  |  72000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 83821 | brandt     | comp.sci |  92000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 98345 | kim        | elec.eng |  80000 | 83821 | CS-190    |      1 | spring   | 2017 |
| 10101 | srinivasan | comp.sci |  65000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 10211 | smith      | biology  |  66000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 10212 | tom        | biology  |   NULL | 83821 | CS-190    |      2 | spring   | 2017 |
| 12121 | wu         | finance  |  90000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 15151 | mozarat    | music    |  40000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 22222 | einstein   | physics  |  95000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 32343 | el said    | history  |  60000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 33456 | gold       | physics  |  87000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 45565 | katz       | comp.sci |  75000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 58583 | califeri   | history  |  62000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 76543 | singh      | finance  |  80000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 76766 | crick      | biology  |  72000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 83821 | brandt     | comp.sci |  92000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 98345 | kim        | elec.eng |  80000 | 83821 | CS-190    |      2 | spring   | 2017 |
| 10101 | srinivasan | comp.sci |  65000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 10211 | smith      | biology  |  66000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 10212 | tom        | biology  |   NULL | 83821 | CS-319    |      2 | spring   | 2018 |
| 12121 | wu         | finance  |  90000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 15151 | mozarat    | music    |  40000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 22222 | einstein   | physics  |  95000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 32343 | el said    | history  |  60000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 33456 | gold       | physics  |  87000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 45565 | katz       | comp.sci |  75000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 58583 | califeri   | history  |  62000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 76543 | singh      | finance  |  80000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 76766 | crick      | biology  |  72000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 83821 | brandt     | comp.sci |  92000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 98345 | kim        | elec.eng |  80000 | 83821 | CS-319    |      2 | spring   | 2018 |
| 10101 | srinivasan | comp.sci  |  65000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 10211 | smith      | biology   |  66000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 10212 | tom        | biology   |   NULL | 98345 | EE-181    |      1 | spring   | 2017 |
| 12121 | wu         | finance   |  90000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 15151 | mozarat    | music     |  40000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 22222 | einstein   | physics   |  95000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 32343 | el said    | history   |  60000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 33456 | gold       | physics   |  87000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 45565 | katz       | comp.sci  |  75000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 58583 | califeri   | history   |  62000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 76543 | singh      | finance   |  80000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 76766 | crick      | biology   |  72000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 83821 | brandt     | comp.sci  |  92000 | 98345 | EE-181    |      1 | spring   | 2017 |
| 98345 | kim        | elec.eng  |  80000 | 98345 | EE-181    |      1 | spring   | 2017 |
+-------+------------+-----------+--------+-------+-----------+--------+----------+------+
224 rows in set (0.001 sec)
```

```
-- 7. Find the names of all instructors who have taught some course and the course_id
select distinct name,teaches.course_id from instructor join teaches on instructor.id = teaches.id;
+------------+-----------+
| name       | course_id |
+------------+-----------+
| srinivasan | CS-101    |
| srinivasan | CS-315    |
| srinivasan | CS-347    |
| wu         | FIN-201   |
| mozarat    | MU-199    |
| einstein   | PHY-101   |
| el said    | HIS-351   |
| katz       | CS-319    |
| crick      | BIO-101   |
| crick      | BIO-301   |
| brandt     | CS-190    |
| brandt     | CS-319    |
| kim        | EE-181    |
+------------+-----------+
13 rows in set (0.001 sec)
```

-- 8. Find the names of all instructors whose name includes the substring “dar”.

```
select name from instructor where name like "%at%";
+---------+
| name    |
+---------+
| mozarat |
| katz    |
+---------+
2 rows in set (0.000 sec)
```

-- 9. Find the names of all instructors with salary between 90,000 and 100,000 (that is, ≥ 90,000 and ≤ 100,000)

```
select name from instructor where salary between 90000 and 100000;
+----------+
| name     |
+----------+
| wu       |
| einstein |
| brandt   |
+----------+
3 rows in set (0.000 sec)
```

EXPERIMENT 4 -BASIC SQL

-- EXPERIMENT 4

-- 1. Order the tuples in the instructors relation as per their salary.

```
select * from instructor order by salary asc;
+-------+------------+-----------+--------+
| id    | name       | dept_name | salary |
+-------+------------+-----------+--------+
| 10212 | tom        | biology   |   NULL |
| 15151 | mozarat    | music     |  40000 |
| 32343 | el said    | history   |  60000 |
| 58583 | califeri   | history   |  62000 |
| 10101 | srinivasan | comp.sci  |  65000 |
| 10211 | smith      | biology   |  66000 |
| 76766 | crick      | biology   |  72000 |
| 45565 | katz       | comp.sci  |  75000 |
| 98345 | kim        | elec.eng  |  80000 |
| 76543 | singh      | finance   |  80000 |
| 33456 | gold       | physics   |  87000 |
| 12121 | wu         | finance   |  90000 |
| 83821 | brandt     | comp.sci  |  92000 |
| 22222 | einstein   | physics   |  95000 |
+-------+------------+-----------+--------+
14 rows in set (0.000 sec)
```

-- 2. Find courses that ran in Fall 2017 or in Spring 2018

```
select distinct course_id from teaches where (semester = "fall" and year =2017) or (semester = "spring" and year =2018);
```

```
+-----------+
| course_id |
+-----------+
| CS-101    |
| CS-315    |
| CS-347    |
| FIN-201   |
| MU-199    |
| PHY-101   |
| HIS-351   |
| CS-319    |
+-----------+
8 rows in set (0.000 sec)
```

-- 3. Find courses that ran in Fall 2017 and in Spring 2018

```
select course_id from teaches where  semester = ("fall" and year =2017) and (semester = "spring" and year =2018);
+-----------+
| course_id |
+-----------+
| CS-315    |
| FIN-201   |
| MU-199    |
| CS-101    |
| HIS-351   |
| CS-319    |
| CS-319    |
+-----------+
7 rows in set (0.000 sec)
```

-- 4. Find courses that ran in Fall 2017 but not in Spring 2018

```
select course_id from teaches where (semester = "fall" and year =2017) AND NOT (semester = "spring" and year =2018);

+-----------+
| course_id |
+-----------+
| CS-101    |
| CS-347    |
| PHY-101   |
+-----------+
3 rows in set (0.000 sec)
```

-- 5. Insert following additional tuples in instructor :('10211', 'Smith', 'Biology', 66000), ('10212', 'Tom', 'Biology', NULL)

```
insert into instructor values(10211,"smith","biology",66000),(10212,"tom","biology",null);
Query OK, 2 row affected (0.001 sec)
```

-- 6. Find all instructors whose salary is null.

```
select * from instructor where salary is null;
+-------+------+-----------+--------+
| id    | name | dept_name | salary |
+-------+------+-----------+--------+
| 10212 | tom  | biology   |   NULL |
+-------+------+-----------+--------+
1 row in set (0.000 sec)
```

-- 7. Find the average salary of instructors in the Computer Science department.

```
select avg(salary) as avg_salary from instructor where dept_name='Comp.Sci';
```

```
+------------+
| avg_salary |
+------------+
| 77333.3333 |
+------------+
1 row in set (0.000 sec)
```

EXPERIMENT 5 – INTERMEDIATE SQL

```
-- EXPERIMENT 5
-- 1. Find the total number of instructors who teach a course in the Spring 2018 semester.
select count(distinct id) from teaches where semester ="spring" and year = 2018;
+--------------------+
| count(distinct id) |
+--------------------+
|                  6 |
+--------------------+
1 row in set (0.000 sec)

-- 2. Find the number of tuples in the teaches relation
Select count(*) from teaches;
+----------+
| count(*) |
+----------+
|       16 |
+----------+
1 row in set (0.000 sec)


-- 3. Find the average salary of instructors in each department
select dept_name , avg(salary) from instructor group by dept_name;
+-----------+-------------+
| dept_name | avg(salary) |
+-----------+-------------+
| biology   |  69000.0000 |
| comp.sci  |  77333.3333 |
| elec.eng  |  80000.0000 |
| finance   |  85000.0000 |
| history   |  61000.0000 |
| music     |  40000.0000 |
| physics   |  91000.0000 |
+-----------+-------------+
7 rows in set (0.000 sec)
```

-- 4. Find the names and average salaries of all departments whose average salary is greater than 42000

```
select dept_name , avg(salary) from instructor group by dept_name having avg(salary)> 42000;
+-----------+-------------+
| dept_name | avg(salary) |
+-----------+-------------+
| biology   |  69000.0000 |
| comp.sci  |  77333.3333 |
| elec.eng  |  80000.0000 |
| finance   |  85000.0000 |
| history   |  61000.0000 |
| physics   |  91000.0000 |
+-----------+-------------+
6 rows in set (0.000 sec)
```

-- 5. Name all instructors whose name is neither “Mozart” nor Einstein”.

```
select * from instructor where name not in ("mozarat","einstein");
+-------+------------+-----------+--------+
| id    | name       | dept_name | salary |
+-------+------------+-----------+--------+
| 10101 | srinivasan | comp.sci  |  65000 |
| 10211 | smith      | biology   |  66000 |
| 10212 | tom        | biology   |   NULL |
| 12121 | wu         | finance   |  90000 |
| 32343 | el said    | history   |  60000 |
| 33456 | gold       | physics   |  87000 |
| 45565 | katz       | comp.sci  |  75000 |
| 58583 | califeri   | history   |  62000 |
| 76543 | singh      | finance   |  80000 |
| 76766 | crick      | biology   |  72000 |
| 83821 | brandt     | comp.sci  |  92000 |
| 98345 | kim        | elec.eng  |  80000 |
+-------+------------+-----------+--------+
12 rows in set (0.000 sec)
```

-- 6. Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.

```
select * from instructor where salary> any (select salary from instructor where dept_name='biology');
+-------+----------+-----------+--------+
| id    | name     | dept_name | salary |
+-------+----------+-----------+--------+
| 12121 | wu       | finance   |  90000 |
| 22222 | einstein | physics   |  95000 |
| 33456 | gold     | physics   |  87000 |
| 45565 | katz     | comp.sci  |  75000 |
| 76543 | singh    | finance   |  80000 |
| 76766 | crick    | biology   |  72000 |
| 83821 | brandt   | comp.sci  |  92000 |
| 98345 | kim      | elec.eng  |  80000 |
+-------+----------+-----------+--------+
8 rows in set (0.000 sec)
```

-- 7. Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.

```
select * from instructor where salary > all (select salary from instructor where dept_name='biology');

Empty set (0.000 sec)
```

-- 8. Find the average instructors’ salaries of those departments where the average salary is greater than 42,000.

```
select dept_name,avg(salary) from instructor group by dept_name having avg(salary)> 42000;
+-----------+-------------+
| dept_name | avg(salary) |
+-----------+-------------+
| biology   |  69000.0000 |
| comp.sci  |  77333.3333 |
| elec.eng  |  80000.0000 |
| finance   |  85000.0000 |
| history   |  61000.0000 |
| physics   |  91000.0000 |
+-----------+-------------+
6 rows in set (0.000 sec)
```

EXPERIMENT 6 – ADVANCED AND INTERMEDIATE SQL

```
-- EXPERIMENT 6


-- 1. Find all departments where the total salary is greater than the average of the total salary at all departments

select dept_name,sum(salary), avg(salary) from instructor group by dept_name having sum(salary) > avg(salary);

-- select avg(salary) from instructor;

--  where salary >= (select avg(salary) from instructor group by dept_name)

+-----------+-------------+-------------+
| dept_name | sum(salary) | avg(salary) |
+-----------+-------------+-------------+
| biology   |      138000 |  69000.0000 |
| comp.sci  |      232000 |  77333.3333 |
| finance   |      170000 |  85000.0000 |
| history   |      122000 |  61000.0000 |
| physics   |      182000 |  91000.0000 |
+-----------+-------------+-------------+
5 rows in set (0.000 sec)


-- 2. List the names of instructors along with the course ID of the courses that they taught.

select distinct name,course_id from instructor inner join teaches on instructor.id=teaches.id ;

+------------+-----------+
| name       | course_id |
+------------+-----------+
| srinivasan | CS-101    |
| srinivasan | CS-315    |
| srinivasan | CS-347    |
| wu         | FIN-201   |
| mozarat    | MU-199    |
| einstein   | PHY-101   |
| el said    | HIS-351   |
| katz       | CS-319    |
| crick      | BIO-101   |
| crick      | BIO-301   |
| brandt     | CS-190    |
| brandt     | CS-319    |
| kim        | EE-181    |
+------------+-----------+
13 rows in set (0.000 sec)
```

-- 3. List the names of instructors along with the course ID of the courses that they taught. In case, an instructor teaches no courses keep the course ID as null.

```
select distinct name,course_id from instructor left join teaches on instructor.id=teaches.id ;
```

```
+------------+-----------+
| name       | course_id |
+------------+-----------+
| srinivasan | CS-101    |
| srinivasan | CS-315    |
| srinivasan | CS-347    |
| smith      | NULL      |
| tom        | NULL      |
| wu         | FIN-201   |
| mozarat    | MU-199    |
| einstein   | PHY-101   |
| el said    | HIS-351   |
| gold       | NULL      |
| katz       | CS-319    |
| califeri   | NULL      |
| singh      | NULL      |
| crick      | BIO-101   |
| crick      | BIO-301   |
| brandt     | CS-190    |
| brandt     | CS-319    |
| kim        | EE-181    |
+------------+-----------+
18 rows in set (0.000 sec)
```

```
-- 4. Create a view of instructors without their salary called faculty
create view FACULTY as select id, name, dept_name from instructor;
select * from FACULTY;
```

```
+-------+------------+-----------+
| id    | name       | dept_name |
+-------+------------+-----------+
| 10101 | srinivasan | comp.sci  |
| 10211 | smith      | biology   |
| 10212 | tom        | biology   |
| 12121 | wu         | finance   |
| 15151 | mozarat    | music     |
| 22222 | einstein   | physics   |
| 32343 | el said    | history   |
| 33456 | gold       | physics   |
| 45565 | katz       | comp.sci  |
| 58583 | califeri   | history   |
| 76543 | singh      | finance   |
| 76766 | crick      | biology   |
| 83821 | brandt     | comp.sci  |
| 98345 | kim        | elec.eng  |
+-------+------------+-----------+
14 rows in set (0.000 sec)
```

```
-- 5. Give select privileges on the view faculty to the new user.
create user "new"@"localhost" identified by 'password';
grant select on Ads2_8.FACULTY TO "new"@"localhost";
```

EXPERIMENT 7- ADVANCED SQL

```
-- EXPERIMENT 7
-- 1. Create a view of instructors without their salary called faculty
create view FACULTY as select id, name, dept_name from instructor;

-- 2. Create a view of department salary totals
create view dept_salary as select dept_name,sum(salary) from instructor group by dept_name;
-- drop view dept_salary;
select * from dept_salary;

+-----------+-------------+
| dept_name | sum(salary) |
+-----------+-------------+
| biology   |      138000 |
| comp.sci  |      232000 |
| elec.eng  |       80000 |
| finance   |      170000 |
| history   |      122000 |
| music     |       40000 |
| physics   |      182000 |
+-----------+-------------+
7 rows in set (0.000 sec)



-- 3. CREATE A ROLE OF STUDENT
create role student;

-- 4. Give select privileges on the view faculty to the role student.
```

```
grant select on Ads2_8.FACULTY to student;

-- 5. Create a new user and assign her the role of student.
create user "student_user"@"localhost" identified by "root";
-- grant select on Ads2_8.* to  "student_user"@"localhost";
grant 'student' to "student_user"@"localhost";

-- 6. Login as this new user and find all instructors in the Biology department.
select name from instructor where dept_name ='biology';

-- 7. Revoke privileges of the new user
-- revoke select on Ads2_8.FACULTY from  "student_user"@"localhost";
+-------+
| name  |
+-------+
| smith |
| tom   |
| crick |
+-------+
3 rows in set (0.000 sec)


-- 8. Remove the role of student.
drop role student;

-- 9. Give select privileges on the view faculty to the new user.
grant select on Ads2_8.FACULTY to  "student_user"@"localhost";

-- 10. Login as this new user and find all instructors in the finance department.
select name from Ads2_8.FACULTY where dept_name ='finance';
```

-- 11. Login again as root user

mysql - u root -p

-- 12. Create table teaches2 with same columns as teaches.

create table teaches2 select * from teaches;

select * from teaches2;

```
+-------+-----------+--------+----------+------+
| id    | course_id | sec_id | semester | year |
+-------+-----------+--------+----------+------+
| 10101 | CS-101    |      1 | fall     | 2017 |
| 10101 | CS-315    |      1 | spring   | 2018 |
| 10101 | CS-347    |      1 | fall     | 2017 |
| 12121 | FIN-201   |      1 | spring   | 2018 |
| 15151 | MU-199    |      1 | spring   | 2018 |
| 22222 | PHY-101   |      1 | fall     | 2017 |
| 10101 | CS-101    |      1 | spring   | 2018 |
| 32343 | HIS-351   |      1 | spring   | 2018 |
| 45565 | CS-319    |      1 | spring   | 2018 |
| 45565 | CS-319    |      1 | spring   | 2017 |
| 76766 | BIO-101   |      1 | summer   | 2018 |
| 76766 | BIO-301   |      1 | summer   | 2017 |
| 83821 | CS-190    |      1 | spring   | 2017 |
| 83821 | CS-190    |      2 | spring   | 2017 |
| 83821 | CS-319    |      2 | spring   | 2018 |
| 98345 | EE-181    |      1 | spring   | 2017 |
+-------+-----------+--------+----------+------+
16 rows in set (0.000 sec)
```

-- 13. Create index ID column of teaches.

create index t_index on teaches2(id);

show index from teaches2;

```
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------
+------+------------+---------+---------------+---------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment | Index_comment | Ignored |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------
+------+------------+---------+---------------+---------+
| teaches2 |          1 | t_index  |            1 | id          | A         |          16 |     NULL | NULL   | YES  |
BTREE      |         |               | NO      |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------
+------+------------+---------+---------------+---------+
1 row in set (0.000 sec)
```

-- 14. Drop the index to free up the space.

alter table teaches2 drop index t_index ;

EXPERIMENT 8 – ACCESSING DATABASE THROUGH PYTHON

```
import mysql.connector


conn = mysql.connector.connect(user='root',
                    host='localhost',
                    passwd='root',
                    database='Ads2_8',
                    auth_plugin='mysql_native_password')


cur = conn.cursor()
dis_cur = conn.cursor()


# 1. Insert following additional tuple in instructor : ('10211', 'Smith', 'Biology', 66000)
stmt = 'create table instructor(id integer primary key, name text, dept_name text, salary integer);'
cur.execute(stmt)
insert_ele= [(10101,"srinivasan","comp.sci",65000),(12121,"wu","finance",90000),
(15151,"mozarat","music",40000),
(22222,"einstein","physics",95000),(32343,"el said","history",60000),
(33456,"gold","physics",87000),(45565,"katz","comp.sci",75000),
(58583,"califeri","history",62000),(76543,"singh","finance",80000),
(76766,"crick","biology",72000),
(83821,"brandt","comp.sci",92000),(98345,"kim","elec.eng",80000),]
stmt ="insert into instructor (id, name, dept_name, salary) values(%s, %s, %s, %s)"
# cur.executemany(stmt,insert_ele)



insert_ele =[('10211', 'Smith', 'Biology', 66000)]
cur.executemany(stmt,insert_ele)
```

```
# 2. Delete this tuple from instructor : ('10211', 'Smith', 'Biology', 66000)
stmt ="delete from instructor where id = 10211"
cur.execute(stmt)



# 3. Select tuples from instructor where dept_name = ‘History’
stmt="select * from instructor where dept_name='history' "
# dis_cur.execute(stmt)

(32343, 'el said', 'history', 60000)
(58583, 'califeri', 'history', 62000)

# 4. Find the Cartesian product instructor x teaches.
stmt="select * from instructor cross join teaches;"

(10101, 'srinivasan', 'comp.sci', 65000, 10101, 'CS-101', 1, 'fall', 2017)
(10211, 'smith', 'biology', 66000, 10101, 'CS-101', 1, 'fall', 2017)
(10212, 'tom', 'biology', None, 10101, 'CS-101', 1, 'fall', 2017)
(12121, 'wu', 'finance', 90000, 10101, 'CS-101', 1, 'fall', 2017)
(15151, 'mozarat', 'music', 40000, 10101, 'CS-101', 1, 'fall', 2017)
(22222, 'einstein', 'physics', 95000, 10101, 'CS-101', 1, 'fall', 2017)
(32343, 'el said', 'history', 60000, 10101, 'CS-101', 1, 'fall', 2017)
(33456, 'gold', 'physics', 87000, 10101, 'CS-101', 1, 'fall', 2017)
(45565, 'katz', 'comp.sci', 75000, 10101, 'CS-101', 1, 'fall', 2017)
(58583, 'califeri', 'history', 62000, 10101, 'CS-101', 1, 'fall', 2017)
(76543, 'singh', 'finance', 80000, 10101, 'CS-101', 1, 'fall', 2017)
(76766, 'crick', 'biology', 72000, 10101, 'CS-101', 1, 'fall', 2017)
(83821, 'brandt', 'comp.sci', 92000, 10101, 'CS-101', 1, 'fall', 2017)
(98345, 'kim', 'elec.eng', 80000, 10101, 'CS-101', 1, 'fall', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 10101, 'CS-315', 1, 'spring', 2018)
```

```
(10211, 'smith', 'biology', 66000, 10101, 'CS-315', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 10101, 'CS-315', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 10101, 'CS-315', 1, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 10101, 'CS-315', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 10101, 'CS-315', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 10101, 'CS-315', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 10101, 'CS-315', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 10101, 'CS-315', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 10101, 'CS-315', 1, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 10101, 'CS-315', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 10101, 'CS-315', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 10101, 'CS-315', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 10101, 'CS-315', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 10101, 'CS-347', 1, 'fall', 2017)
(10211, 'smith', 'biology', 66000, 10101, 'CS-347', 1, 'fall', 2017)
(10212, 'tom', 'biology', None, 10101, 'CS-347', 1, 'fall', 2017)
(12121, 'wu', 'finance', 90000, 10101, 'CS-347', 1, 'fall', 2017)
(15151, 'mozarat', 'music', 40000, 10101, 'CS-347', 1, 'fall', 2017)
(22222, 'einstein', 'physics', 95000, 10101, 'CS-347', 1, 'fall', 2017)
(32343, 'el said', 'history', 60000, 10101, 'CS-347', 1, 'fall', 2017)
(33456, 'gold', 'physics', 87000, 10101, 'CS-347', 1, 'fall', 2017)
(45565, 'katz', 'comp.sci', 75000, 10101, 'CS-347', 1, 'fall', 2017)
(58583, 'califeri', 'history', 62000, 10101, 'CS-347', 1, 'fall', 2017)
(76543, 'singh', 'finance', 80000, 10101, 'CS-347', 1, 'fall', 2017)
(76766, 'crick', 'biology', 72000, 10101, 'CS-347', 1, 'fall', 2017)
(83821, 'brandt', 'comp.sci', 92000, 10101, 'CS-347', 1, 'fall', 2017)
(98345, 'kim', 'elec.eng', 80000, 10101, 'CS-347', 1, 'fall', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 12121, 'FIN-201', 1, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 12121, 'FIN-201', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 12121, 'FIN-201', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 12121, 'FIN-201', 1, 'spring', 2018)
```

```
(15151, 'mozarat', 'music', 40000, 12121, 'FIN-201', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 12121, 'FIN-201', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 12121, 'FIN-201', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 12121, 'FIN-201', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 12121, 'FIN-201', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 12121, 'FIN-201', 1, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 12121, 'FIN-201', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 12121, 'FIN-201', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 12121, 'FIN-201', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 12121, 'FIN-201', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 15151, 'MU-199', 1, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 15151, 'MU-199', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 15151, 'MU-199', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 15151, 'MU-199', 1, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 15151, 'MU-199', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 15151, 'MU-199', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 15151, 'MU-199', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 15151, 'MU-199', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 15151, 'MU-199', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 15151, 'MU-199', 1, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 15151, 'MU-199', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 15151, 'MU-199', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 15151, 'MU-199', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 15151, 'MU-199', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 22222, 'PHY-101', 1, 'fall', 2017)
(10211, 'smith', 'biology', 66000, 22222, 'PHY-101', 1, 'fall', 2017)
(10212, 'tom', 'biology', None, 22222, 'PHY-101', 1, 'fall', 2017)
(12121, 'wu', 'finance', 90000, 22222, 'PHY-101', 1, 'fall', 2017)
(15151, 'mozarat', 'music', 40000, 22222, 'PHY-101', 1, 'fall', 2017)
(22222, 'einstein', 'physics', 95000, 22222, 'PHY-101', 1, 'fall', 2017)
(32343, 'el said', 'history', 60000, 22222, 'PHY-101', 1, 'fall', 2017)
```

```
(33456, 'gold', 'physics', 87000, 22222, 'PHY-101', 1, 'fall', 2017)
(45565, 'katz', 'comp.sci', 75000, 22222, 'PHY-101', 1, 'fall', 2017)
(58583, 'califeri', 'history', 62000, 22222, 'PHY-101', 1, 'fall', 2017)
(76543, 'singh', 'finance', 80000, 22222, 'PHY-101', 1, 'fall', 2017)
(76766, 'crick', 'biology', 72000, 22222, 'PHY-101', 1, 'fall', 2017)
(83821, 'brandt', 'comp.sci', 92000, 22222, 'PHY-101', 1, 'fall', 2017)
(98345, 'kim', 'elec.eng', 80000, 22222, 'PHY-101', 1, 'fall', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 10101, 'CS-101', 1, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 10101, 'CS-101', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 10101, 'CS-101', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 10101, 'CS-101', 1, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 10101, 'CS-101', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 10101, 'CS-101', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 10101, 'CS-101', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 10101, 'CS-101', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 10101, 'CS-101', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 10101, 'CS-101', 1, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 10101, 'CS-101', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 10101, 'CS-101', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 10101, 'CS-101', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 10101, 'CS-101', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 32343, 'HIS-351', 1, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 32343, 'HIS-351', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 32343, 'HIS-351', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 32343, 'HIS-351', 1, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 32343, 'HIS-351', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 32343, 'HIS-351', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 32343, 'HIS-351', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 32343, 'HIS-351', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 32343, 'HIS-351', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 32343, 'HIS-351', 1, 'spring', 2018)
```

```
(76543, 'singh', 'finance', 80000, 32343, 'HIS-351', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 32343, 'HIS-351', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 32343, 'HIS-351', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 32343, 'HIS-351', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 45565, 'CS-319', 1, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 45565, 'CS-319', 1, 'spring', 2018)
(10212, 'tom', 'biology', None, 45565, 'CS-319', 1, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 45565, 'CS-319', 1, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 45565, 'CS-319', 1, 'spring', 2018)
(22222, 'einstein', 'physics', 95000, 45565, 'CS-319', 1, 'spring', 2018)
(32343, 'el said', 'history', 60000, 45565, 'CS-319', 1, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 45565, 'CS-319', 1, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 45565, 'CS-319', 1, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 45565, 'CS-319', 1, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 45565, 'CS-319', 1, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 45565, 'CS-319', 1, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 45565, 'CS-319', 1, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 45565, 'CS-319', 1, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 45565, 'CS-319', 1, 'spring', 2017)
(10211, 'smith', 'biology', 66000, 45565, 'CS-319', 1, 'spring', 2017)
(10212, 'tom', 'biology', None, 45565, 'CS-319', 1, 'spring', 2017)
(12121, 'wu', 'finance', 90000, 45565, 'CS-319', 1, 'spring', 2017)
(15151, 'mozarat', 'music', 40000, 45565, 'CS-319', 1, 'spring', 2017)
(22222, 'einstein', 'physics', 95000, 45565, 'CS-319', 1, 'spring', 2017)
(32343, 'el said', 'history', 60000, 45565, 'CS-319', 1, 'spring', 2017)
(33456, 'gold', 'physics', 87000, 45565, 'CS-319', 1, 'spring', 2017)
(45565, 'katz', 'comp.sci', 75000, 45565, 'CS-319', 1, 'spring', 2017)
(58583, 'califeri', 'history', 62000, 45565, 'CS-319', 1, 'spring', 2017)
(76543, 'singh', 'finance', 80000, 45565, 'CS-319', 1, 'spring', 2017)
(76766, 'crick', 'biology', 72000, 45565, 'CS-319', 1, 'spring', 2017)
(83821, 'brandt', 'comp.sci', 92000, 45565, 'CS-319', 1, 'spring', 2017)
```

```
(98345, 'kim', 'elec.eng', 80000, 45565, 'CS-319', 1, 'spring', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 76766, 'BIO-101', 1, 'summer', 2018)
(10211, 'smith', 'biology', 66000, 76766, 'BIO-101', 1, 'summer', 2018)
(10212, 'tom', 'biology', None, 76766, 'BIO-101', 1, 'summer', 2018)
(12121, 'wu', 'finance', 90000, 76766, 'BIO-101', 1, 'summer', 2018)
(15151, 'mozarat', 'music', 40000, 76766, 'BIO-101', 1, 'summer', 2018)
(22222, 'einstein', 'physics', 95000, 76766, 'BIO-101', 1, 'summer', 2018)
(32343, 'el said', 'history', 60000, 76766, 'BIO-101', 1, 'summer', 2018)
(33456, 'gold', 'physics', 87000, 76766, 'BIO-101', 1, 'summer', 2018)
(45565, 'katz', 'comp.sci', 75000, 76766, 'BIO-101', 1, 'summer', 2018)
(58583, 'califeri', 'history', 62000, 76766, 'BIO-101', 1, 'summer', 2018)
(76543, 'singh', 'finance', 80000, 76766, 'BIO-101', 1, 'summer', 2018)
(76766, 'crick', 'biology', 72000, 76766, 'BIO-101', 1, 'summer', 2018)
(83821, 'brandt', 'comp.sci', 92000, 76766, 'BIO-101', 1, 'summer', 2018)
(98345, 'kim', 'elec.eng', 80000, 76766, 'BIO-101', 1, 'summer', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 76766, 'BIO-301', 1, 'summer', 2017)
(10211, 'smith', 'biology', 66000, 76766, 'BIO-301', 1, 'summer', 2017)
(10212, 'tom', 'biology', None, 76766, 'BIO-301', 1, 'summer', 2017)
(12121, 'wu', 'finance', 90000, 76766, 'BIO-301', 1, 'summer', 2017)
(15151, 'mozarat', 'music', 40000, 76766, 'BIO-301', 1, 'summer', 2017)
(22222, 'einstein', 'physics', 95000, 76766, 'BIO-301', 1, 'summer', 2017)
(32343, 'el said', 'history', 60000, 76766, 'BIO-301', 1, 'summer', 2017)
(33456, 'gold', 'physics', 87000, 76766, 'BIO-301', 1, 'summer', 2017)
(45565, 'katz', 'comp.sci', 75000, 76766, 'BIO-301', 1, 'summer', 2017)
(58583, 'califeri', 'history', 62000, 76766, 'BIO-301', 1, 'summer', 2017)
(76543, 'singh', 'finance', 80000, 76766, 'BIO-301', 1, 'summer', 2017)
(76766, 'crick', 'biology', 72000, 76766, 'BIO-301', 1, 'summer', 2017)
(83821, 'brandt', 'comp.sci', 92000, 76766, 'BIO-301', 1, 'summer', 2017)
(98345, 'kim', 'elec.eng', 80000, 76766, 'BIO-301', 1, 'summer', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 83821, 'CS-190', 1, 'spring', 2017)
(10211, 'smith', 'biology', 66000, 83821, 'CS-190', 1, 'spring', 2017)
```

```
(10212, 'tom', 'biology', None, 83821, 'CS-190', 1, 'spring', 2017)
(12121, 'wu', 'finance', 90000, 83821, 'CS-190', 1, 'spring', 2017)
(15151, 'mozarat', 'music', 40000, 83821, 'CS-190', 1, 'spring', 2017)
(22222, 'einstein', 'physics', 95000, 83821, 'CS-190', 1, 'spring', 2017)
(32343, 'el said', 'history', 60000, 83821, 'CS-190', 1, 'spring', 2017)
(33456, 'gold', 'physics', 87000, 83821, 'CS-190', 1, 'spring', 2017)
(45565, 'katz', 'comp.sci', 75000, 83821, 'CS-190', 1, 'spring', 2017)
(58583, 'califeri', 'history', 62000, 83821, 'CS-190', 1, 'spring', 2017)
(76543, 'singh', 'finance', 80000, 83821, 'CS-190', 1, 'spring', 2017)
(76766, 'crick', 'biology', 72000, 83821, 'CS-190', 1, 'spring', 2017)
(83821, 'brandt', 'comp.sci', 92000, 83821, 'CS-190', 1, 'spring', 2017)
(98345, 'kim', 'elec.eng', 80000, 83821, 'CS-190', 1, 'spring', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 83821, 'CS-190', 2, 'spring', 2017)
(10211, 'smith', 'biology', 66000, 83821, 'CS-190', 2, 'spring', 2017)
(10212, 'tom', 'biology', None, 83821, 'CS-190', 2, 'spring', 2017)
(12121, 'wu', 'finance', 90000, 83821, 'CS-190', 2, 'spring', 2017)
(15151, 'mozarat', 'music', 40000, 83821, 'CS-190', 2, 'spring', 2017)
(22222, 'einstein', 'physics', 95000, 83821, 'CS-190', 2, 'spring', 2017)
(32343, 'el said', 'history', 60000, 83821, 'CS-190', 2, 'spring', 2017)
(33456, 'gold', 'physics', 87000, 83821, 'CS-190', 2, 'spring', 2017)
(45565, 'katz', 'comp.sci', 75000, 83821, 'CS-190', 2, 'spring', 2017)
(58583, 'califeri', 'history', 62000, 83821, 'CS-190', 2, 'spring', 2017)
(76543, 'singh', 'finance', 80000, 83821, 'CS-190', 2, 'spring', 2017)
(76766, 'crick', 'biology', 72000, 83821, 'CS-190', 2, 'spring', 2017)
(83821, 'brandt', 'comp.sci', 92000, 83821, 'CS-190', 2, 'spring', 2017)
(98345, 'kim', 'elec.eng', 80000, 83821, 'CS-190', 2, 'spring', 2017)
(10101, 'srinivasan', 'comp.sci', 65000, 83821, 'CS-319', 2, 'spring', 2018)
(10211, 'smith', 'biology', 66000, 83821, 'CS-319', 2, 'spring', 2018)
(10212, 'tom', 'biology', None, 83821, 'CS-319', 2, 'spring', 2018)
(12121, 'wu', 'finance', 90000, 83821, 'CS-319', 2, 'spring', 2018)
(15151, 'mozarat', 'music', 40000, 83821, 'CS-319', 2, 'spring', 2018)
```

```
(22222, 'einstein', 'physics', 95000, 83821, 'CS-319', 2, 'spring', 2018)
(32343, 'el said', 'history', 60000, 83821, 'CS-319', 2, 'spring', 2018)
(33456, 'gold', 'physics', 87000, 83821, 'CS-319', 2, 'spring', 2018)
(45565, 'katz', 'comp.sci', 75000, 83821, 'CS-319', 2, 'spring', 2018)
(58583, 'califeri', 'history', 62000, 83821, 'CS-319', 2, 'spring', 2018)
(76543, 'singh', 'finance', 80000, 83821, 'CS-319', 2, 'spring', 2018)
(76766, 'crick', 'biology', 72000, 83821, 'CS-319', 2, 'spring', 2018)
(83821, 'brandt', 'comp.sci', 92000, 83821, 'CS-319', 2, 'spring', 2018)
(98345, 'kim', 'elec.eng', 80000, 83821, 'CS-319', 2, 'spring', 2018)
(10101, 'srinivasan', 'comp.sci', 65000, 98345, 'EE-181', 1, 'spring', 2017)
(10211, 'smith', 'biology', 66000, 98345, 'EE-181', 1, 'spring', 2017)
(10212, 'tom', 'biology', None, 98345, 'EE-181', 1, 'spring', 2017)
(12121, 'wu', 'finance', 90000, 98345, 'EE-181', 1, 'spring', 2017)
(15151, 'mozarat', 'music', 40000, 98345, 'EE-181', 1, 'spring', 2017)
(22222, 'einstein', 'physics', 95000, 98345, 'EE-181', 1, 'spring', 2017)
(32343, 'el said', 'history', 60000, 98345, 'EE-181', 1, 'spring', 2017)
(33456, 'gold', 'physics', 87000, 98345, 'EE-181', 1, 'spring', 2017)
(45565, 'katz', 'comp.sci', 75000, 98345, 'EE-181', 1, 'spring', 2017)
(58583, 'califeri', 'history', 62000, 98345, 'EE-181', 1, 'spring', 2017)
(76543, 'singh', 'finance', 80000, 98345, 'EE-181', 1, 'spring', 2017)
(76766, 'crick', 'biology', 72000, 98345, 'EE-181', 1, 'spring', 2017)
(83821, 'brandt', 'comp.sci', 92000, 98345, 'EE-181', 1, 'spring', 2017)
(98345, 'kim', 'elec.eng', 80000, 98345, 'EE-181', 1, 'spring', 2017)
```

```
# -- 5. Find the names of all instructors who have taught some course and the course_id
stmt="select distinct name,teaches.course_id from instructor join teaches on instructor.id = teaches.id;"
```

```
('srinivasan', 'CS-101')
('srinivasan', 'CS-315')
```

```
('srinivasan', 'CS-347')
('wu', 'FIN-201')
('mozarat', 'MU-199')
('einstein', 'PHY-101')
('el said', 'HIS-351')
('katz', 'CS-319')
('crick', 'BIO-101')
('crick', 'BIO-301')
('brandt', 'CS-190')
('brandt', 'CS-319')
('kim', 'EE-181')
```

```
# -- 6. Find the names of all instructors whose name includes the substring “dar”.
stmt ="select name from instructor where name like '%at%' ;"
```

```
('mozarat',)
('katz',)
```

```
# -- 7. Find the names of all instructors with salary between 90,000 and 100,000 (that is, ≥ 90,000
and ≤ 100,000)
stmt ="select name from instructor where salary between 90000 and 100000;"
dis_cur.execute(stmt)
```

```
('wu',)
('einstein',)
('brandt',)
```

```
rows = dis_cur.fetchall()
for row in rows:
    print(row)
```

```
conn.close()
```

EXPERIMENT 9 – ADVANCED QUERIES THROUGH PYTHON

```
import mysql.connector

conn = mysql.connector.connect(user='root',
                    host='localhost',
                    passwd='root',
                    database='Ads2_8',
                    auth_plugin='mysql_native_password')

cur = conn.cursor()

# -- 1. Order the tuples in the instructors relation as per their salary.
stmt ="select * from instructor order by salary asc;"
cur.execute(stmt)
```

```
(10212, 'tom', 'biology', None)
(15151, 'mozarat', 'music', 40000)
(32343, 'el said', 'history', 60000)
(58583, 'califeri', 'history', 62000)
(10101, 'srinivasan', 'comp.sci', 65000)
(10211, 'smith', 'biology', 66000)
(76766, 'crick', 'biology', 72000)
(45565, 'katz', 'comp.sci', 75000)
(98345, 'kim', 'elec.eng', 80000)
(76543, 'singh', 'finance', 80000)
(33456, 'gold', 'physics', 87000)
(12121, 'wu', 'finance', 90000)
```

```
(83821, 'brandt', 'comp.sci', 92000)
(22222, 'einstein', 'physics', 95000)
```

# -- 2. Find courses that ran in Fall 2017 or in Spring 2018

stmt="select distinct course_id from teaches where (semester = 'fall' and year =2017) or (semester = 'spring' and year =2018);"

```
('CS-101',)
('CS-315',)
('CS-347',)
('FIN-201',)
('MU-199',)
('PHY-101',)
('HIS-351',)
('CS-319',)
```

# -- 3. Find courses that ran in Fall 2017 and in Spring 2018

stmt="select course_id from teaches where semester = ('fall' and year =2017) and (semester = 'spring' and year =2018);"

```
('CS-315',)
('FIN-201',)
('MU-199',)
('CS-101',)
('HIS-351',)
('CS-319',)
('CS-319',)
```

# -- 4. Find courses that ran in Fall 2017 but not in Spring 2018

stmt="select course_id from teaches where (semester = 'fall' and year =2017) AND NOT (semester = 'spring' and year =2018);"

('CS-101',)

('CS-347',)

('PHY-101',)

# -- 5. Insert following additional tuples in instructor :('10211', 'Smith', 'Biology', 66000), ('10212', 'Tom', 'Biology', NULL)

stmt="insert into instructor values(10211,'smith','biology',66000),(10212,'tom','biology',null);"

# -- 6. Find all instructors whose salary is null.

stmt ="select * from instructor where salary is null;"

(10212, 'tom', 'biology', None)

# -- 7. Find the average salary of instructors in the Computer Science department.

stmt="select avg(salary) as avg_salary from instructor where dept_name='Comp.Sci';"

# (Decimal('77333.3333'),)

# -- 8 Find the total number of instructors who teach a course in the Spring 2018 semester.

stmt="select count(distinct id) from teaches where semester ='spring' and year = 2018;"

(6,)

# -- 9. Find the number of tuples in the teaches relation

stmt="Select count(*) from teaches;"

(16,)

# --10. Find the average salary of instructors in each department

stmt="select dept_name , avg(salary) from instructor group by dept_name;"

# ('biology', Decimal('69000.0000'))

# ('comp.sci', Decimal('77333.3333'))

# ('elec.eng', Decimal('80000.0000'))

# ('finance', Decimal('85000.0000'))

# ('history', Decimal('61000.0000'))

# ('music', Decimal('40000.0000'))

# ('physics', Decimal('91000.0000'))

# -- 11. Find the names and average salaries of all departments whose average salary is greater than 42000

stmt="select dept_name , avg(salary) from instructor group by dept_name having avg(salary)> 42000;"

# ('biology', Decimal('69000.0000'))

# ('comp.sci', Decimal('77333.3333'))

# ('elec.eng', Decimal('80000.0000'))

# ('finance', Decimal('85000.0000'))

# ('history', Decimal('61000.0000'))

# ('physics', Decimal('91000.0000'))

# -- 12. Name all instructors whose name is neither “Mozart” nor Einstein”.

stmt="select * from instructor where name not in ('mozarat','einstein');"

```
(10101, 'srinivasan', 'comp.sci', 65000)

(10211, 'smith', 'biology', 66000)

(10212, 'tom', 'biology', None)

(12121, 'wu', 'finance', 90000)

(32343, 'el said', 'history', 60000)

(33456, 'gold', 'physics', 87000)
```

```
(45565, 'katz', 'comp.sci', 75000)

(58583, 'califeri', 'history', 62000)

(76543, 'singh', 'finance', 80000)

(76766, 'crick', 'biology', 72000)

(83821, 'brandt', 'comp.sci', 92000)

(98345, 'kim', 'elec.eng', 80000)
```

# -- 13. Find names of instructors with salary greater than that of some (at least one) instructor in the Biology department.

stmt="select * from instructor where salary> any (select salary from instructor where dept_name='biology');"

```
(12121, 'wu', 'finance', 90000)

(22222, 'einstein', 'physics', 95000)

(33456, 'gold', 'physics', 87000)

(45565, 'katz', 'comp.sci', 75000)

(76543, 'singh', 'finance', 80000)

(76766, 'crick', 'biology', 72000)

(83821, 'brandt', 'comp.sci', 92000)

(98345, 'kim', 'elec.eng', 80000)
```

# -- 14. Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.

stmt="select * from instructor where salary > all (select salary from instructor where dept_name='biology');"

# -- 15. Find the average instructors’ salaries of those departments where the average salary is greater than 42,000.

stmt="select dept_name,avg(salary) from instructor group by dept_name having avg(salary)> 42000;"

```
# ('biology', Decimal('69000.0000'))
# ('comp.sci', Decimal('77333.3333'))
# ('elec.eng', Decimal('80000.0000'))
# ('finance', Decimal('85000.0000'))
# ('history', Decimal('61000.0000'))
# ('physics', Decimal('91000.0000'))
```

# -- 16. Find all departments where the total salary is greater than the average of the total salary at all departments

stmt="select dept_name,sum(salary), avg(salary) from instructor group by dept_name having sum(salary) > avg(salary);"

```
# ('biology', Decimal('138000'), Decimal('69000.0000'))
# ('comp.sci', Decimal('232000'), Decimal('77333.3333'))
# ('finance', Decimal('170000'), Decimal('85000.0000'))
# ('history', Decimal('122000'), Decimal('61000.0000'))
# ('physics', Decimal('182000'), Decimal('91000.0000'))
```

# -- 17. List the names of instructors along with the course ID of the courses that they taught.

stmt="select distinct name,course_id from instructor inner join teaches on instructor.id=teaches.id ;"

```
('srinivasan', 'CS-101')
('srinivasan', 'CS-315')
('srinivasan', 'CS-347')
('wu', 'FIN-201')
('mozarat', 'MU-199')
('einstein', 'PHY-101')
('el said', 'HIS-351')
```

```
('katz', 'CS-319')
('crick', 'BIO-101')
('crick', 'BIO-301')
('brandt', 'CS-190')
('brandt', 'CS-319')
('kim', 'EE-181')
```

# -- 18. List the names of instructors along with the course ID of the courses that they taught. In case, an instructor teaches no courses keep the course ID as null.

```
stmt="select distinct name,course_id from instructor left join teaches on instructor.id=teaches.id ;"
cur.execute(stmt)
```

```
('srinivasan', 'CS-101')
('srinivasan', 'CS-315')
('srinivasan', 'CS-347')
('smith', None)
('tom', None)
('wu', 'FIN-201')
('mozarat', 'MU-199')
('einstein', 'PHY-101')
('el said', 'HIS-351')
('gold', None)
('katz', 'CS-319')
('califeri', None)
('singh', None)
('crick', 'BIO-101')
('crick', 'BIO-301')
('brandt', 'CS-190')
('brandt', 'CS-319')
('kim', 'EE-181')
```

```
rows = cur.fetchall()
for row in rows:
    print(row)
conn.close()
```

EXPERIMENT 10 – OODBMS

```
-- query 1
CREATE TYPE addr_ty AS OBJECT
  2  (street    varchar2(60),
  3  city       varchar2(30),
  4  state      char(2),
  5  zip        varchar(9));
  6  /

Type created.

SQL> CREATE TYPE person_ty AS OBJECT
  2     (name   varchar2(25),
  3     address  addr_ty);
  4  /

Type created.

SQL> CREATE TYPE emp_ty AS OBJECT
  2     (empt_id        varchar2(9),
  3     person  person_ty);
  4
  5  /

Type created.

-- query 2
SQL> CREATE TABLE EMP_OO
```

```
  2     (full_emp emp_ty);
```

```
-- query 3
-- insert
insert into EMP_OO values( emp_ty('100', person_ty('ram', addr_ty('100
st','Patiala','up','605001'))));
insert into EMP_OO values( emp_ty('101', person_ty('sam', addr_ty('101
st','sire','Blore','105001'))));
```

```
-- query 4
-- select
select * from emp_oo;
```

```
FULL_EMP(EMPT_ID, PERSON(NAME, ADDRESS(STREET, CITY, STATE, ZIP)))
--------------------------------------------------------------------------------
EMP_TY('100', PERSON_TY('Raj', ADDR_TY('1000 st', 'Patiala', 'up', '605001')))
EMP_TY('101', PERSON_TY('sam', ADDR_TY('1001 st', 'sire', 'AP', '105001')))
```

```
select e.full_emp.empt_id ID,e.full_emp.person.name NAME, e.full_emp.person.address.city CITY
from emp_oo e;
```

```
ID        NAME                      CITY
--------- ------------------------- ------------------------------
100       Raj                       Patiala
101       sam                       sire
```

```
-- query 5
-- update
update emp_oo e set e.full_emp.person.name = 'Raj' where e.full_emp.empt_id = '1000';
```

```
-- query 6
-- create new obj with member function
create or replace type newemp_ty as object (firstname varchar2(25),
lastname Varchar2(25), birthdate Date, member function age (birthdate in date) return number);


-- query 7
create or replace type body newemp_ty as
        member function age(birthdate in date) return number is
        begin
                return round(sysdate - birthdate);
        end;
end;


-- query 8
create table new_emp_oo (employee newemp_ty);

-- query 9
insert into new_emp_oo values(newemp_ty('ram', 'lal','1976-12-12'));

-- query 10 how to call a member function
select e.employee.firstname, e.employee.age, e.employee.age(e.employee.birthdate) from
new_emp_oo e;

-- query 11 creation of object table
create table new_emp1 of emp_ty;
```

```
-- query 12
insert into new_emp1 values('102',person_ty('raul',addr_ty('100 TU', 'Pta','PB', '147002'))));

-- query 13
select * from new_emp1;

PERSON_TY('raul', ADDR_TY('100 TU', 'Pta', 'PB', '147002'))


-- query 14 references
select ref(p) from new_emp1 p;

REF(P)
--------------------------------------------------------------------------------
0000280209E44C561C843C4E90B9AB35A22AD3E8FBAFAB0D508DDF493C87F3A6F19DC68
04F0041DC
C90000

-- query 15 implementing the concept of fk
create type new_dept_oo as object (deptno number(3),dname varchar(10));

-- query 16
create table dept_table of new_dept_oo;

-- query 17
insert into dept_table values (10,'comp');
insert into dept_table values (20,'chem');
insert into dept_table values (30,'math');


-- query 18
```

```
create table emp_test_fk(empno number(3), name varchar2(10), dept ref new_dept_oo);

-- query 19
set desc depth 2
desc emp_test_fk

 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(3)
 NAME                                               VARCHAR2(10)
 DEPT                                               REF OF NEW_DEPT_OO
   DEPTNO                                           NUMBER(3)
   DNAME                                            VARCHAR2(10)



-- query 20
insert into emp_test_fk select 100, 'raj', ref(p) from dept_table p where deptno =10;
insert into emp_test_fk select 101, 'sam', ref(p) from dept_table p where deptno = 20;

-- query 21 accessing values
select empno, name, deref(e.dept) from emp_test_fk e;
     EMPNO NAME
---------- ----------
DEREF(E.DEPT)(DEPTNO, DNAME)
--------------------------------------------------------------------------------
       100 raj
NEW_DEPT_OO(10, 'comp')

       101 sam
NEW_DEPT_OO(20, 'chem')
```

```
select empno, name, deref(e.dept), deref(e.dept).deptno DEPTNO,deref(e.dept).dname DNAME
from emp_test_fk e;

     EMPNO NAME
---------- ----------
DEREF(E.DEPT)(DEPTNO, DNAME)
--------------------------------------------------------------------------------
    DEPTNO DNAME
---------- ----------
       100 raj
NEW_DEPT_OO(10, 'comp') 10 comp


       101 sam
NEW_DEPT_OO(20, 'chem') 20 chem


     EMPNO NAME
---------- ----------
DEREF(E.DEPT)(DEPTNO, DNAME)
--------------------------------------------------------------------------------
    DEPTNO DNAME
---------- ----------


-- query 22
 create table emp_table_fk (employee emp_ty, dept ref new_dept_oo);


set desc depth 2


-- query 23
insert into emp_table_fk values (emp_ty('100', person_ty('ram', addr_ty('100
st','Patiala','up','605001'))), (select ref(p) from dept_table p where deptno = 10));


-- query 24
```

```
select * from em_table_fk;

EMPLOYEE(EMPT_ID, PERSON(NAME, ADDRESS(STREET, CITY, STATE, ZIP)))

--------------------------------------------------------------------------------

DEPT

--------------------------------------------------------------------------------

EMP_TY('100', PERSON_TY('ram', ADDR_TY('100 st', 'Patiala', 'up', '605001')))

00002202088ECB5F5DB94A44CD901A1BACD0D508D64D9EE4FAD8EF4404B2D19B5A449B
8463


select e.employee.empt_id ID, e.employee.person.name NAME, deref(e.dept), deref(e.dept).deptno
DEPTNO,deref(e.dept).dname DNAME from emp_table_fk e;


ID        NAME

--------- -------------------------

DEREF(E.DEPT)(DEPTNO, DNAME)

--------------------------------------------------------------------------------

   DEPTNO DNAME

---------- ----------

100       ram

NEW_DEPT_OO(10, 'comp')

        10 comp
```