

**PONDICHERRY UNIVERSITY
(A Central university)**



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

NAME : SURENDIRAN B

REG. NO. :

SEMESTER : VIII- Semester

SUBJECT : CSSC424- 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
AANISHAALMAAZ 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

EX · No	DATE	TITLE	PAGE	SIGNATURE
1.	07- 03- 24	PRACTICESQL	5	
2.	20- 03- 24	PROCEDURE,TRIGGERS,CURSOR	12	
3.	27- 03- 24	ACCESSINGTHEDATABASE	15	
4.	10- 04- 24	BASICSQL	27	
5.	10- 04- 24	INTERMEDIATESQL	31	
6.	10- 04- 24	INTERMEDIATEANDADVANCEDSQL	35	
7.	17- 04- 24	ADVANCEDSQL	39	
8.	17- 04- 24	ACCESSINGDBTHROUGHPYTHON	43	
9.	17- 04- 24	ADVANCEDQUERIESTHROUGH PYTHON	54	
10.	15- 05- 24	OODBMS	62	

EXPERIMENT1- PRCATICESQL

```
createdatabaseAds_8; useAds_8;
```

```
createtablesalesman(salesman_idintegerprimarykey,nametextnotnull,citytext,commisi  
on decimal(2,2));  
altertablesalesmanrenamecolumncommisiantocommission;
```

```
insertintosalesmanvalues(5001,'Jameshoog','Newyork',0.15),(5002,'nailknite','Paris',0.  
13), (5005,'Pit Alex','London',0.11),(5006,'Mc Lyon','Paris',0.14),  
(5003,'LausonKen','',0.12),(5007,'PaulAdam','Rome',0.13);
```

```
select*fromsalesman;
```

```
+-----+-----+-----+  
|salesman_id|name|city|commission|  
+-----+-----+-----+  
| 5001|Jameshoog|Newyork| 0.15 |  
| 5002|nailknite|Paris | 0.13 |  
| 5003|Lauson Ken | | 0.12 |  
| 5005|PitAlex| London| 0.11|  
| 5006|McLyon |Paris | 0.14 |  
| 5007|PaulAdam|Rome | 0.13 |  
+-----+-----+-----+  
6rowsinset(0.010sec)
```

```
createtablecustomer(customer_idintegerprimarykey,customer_nametextnotnull,city  
text, grade integer, salesman_id integer,  
foreignkey(salesman_id)referencessalesman(salesman_id));
```

```
insertintocustomervvalues(3002,'NickRink','NewYork',100,5001),(3005,'Graham  
Bell','California',200,5002),(3001,'Brad Pitt','London',null,null),(3004,'Fabio  
Carl','London',300,5006),  
(3007,'JamesBond','Minnesota',200,5001),(3008,'JoeyMann','London',200,5007),(300  
9,'Geff Matt','Berlin',100,null);
```

```
select*fromcustomer;
```

```
+-----+-----+-----+-----+  
+  
|customer_id|customer_name|city|grade|salesman_id|  
+-----+-----+-----+-----+  
+  
| 3001|Brad Pitt |London|NULL| NULL|  
| 3002|Nick Rink |NewYork|100| 5001|  
| 3004|Fabio Carl|London|300| 5006|  
| 3005|GrahamBell|California|200| 5002|  
| 3007|JamesBond |Minnesota|200| 5001|
```

| 3008 |Joey Mann |London |200| 5007 |

```
| 3009|GeffMatt |Berlin |100| NULL|
| 3011|MickyMouse|currentlydeliveryunavailable|290| 5005 |
+-----+-----+-----+-----+-----+
+
8rowsinset(0.017sec)
```

```
createtableorders(order_nointegerprimarykey,purch_amtdecimal(6,2)notnull,order_
datedate not null, customer_id integer,salesman_id integer,
foreignkey(customer_id)referencescustomer(customer_id),foreignkey(salesman_id)
references salesman(salesman_id));
```

```
insertintoordersvalues(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);
updateorderssetsalesman_id=5007whereorder_no=7012;
```

```
select*fromorders;
```

```
+-----+-----+-----+-----+ +
|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 |
|
+-----+-----+-----+-----+ +
8rowsinset(0.001sec)
```

```
-- query1:Displaynameandcommissionofallthesalesmen.
```

```
select name, commission from salesman;
```

```
+-----+-----+
|name |commission |
+-----+-----+
|Jameshoog| 0.15 |
|nailknite| 0.13 |
|Lauson Ken | 0.12 |
|PitAlex| 0.11|
|McLyon | 0.14 |
|PaulAdam| 0.13 |
+-----+-----+
6rowsinset(0.000sec)
```

--

query2:Retrievesalesmanidofallsalesmenfromorderstablewithoutanyrepeats. select distinct salesman_id from orders;

```
+-----+
|salesman_id |
+-----+
|    NULL    |
|    5001    |
|    5002    |
|    5007    |
+-----+
```

--

query3:Displaynamesandcityofsalesman,whobelongtothecityofParis.
select name, city from salesman where city ='Paris';

```
+-----+ +
|name   |city|
+-----+ +
|nailknight|Paris|
|McLyon   |Paris|
+-----+ +
2rowsinset(0.000sec)
```

-- query4:Displayalltheinformationforthosecustomerswithagradeof200.
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   |
+-----+-----+-----+-----+ +
3rowsinset(0.000sec)
```

--

query5:Displaytheordernumber,orderdateandthepurchaseamountfororder(s)whichwill
be delivered by the salesman with ID 5001.
select order_no,order_date,purch_amtfrom orderswheresalesman_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 |
+-----+-----+ +
3rowsinset(0.000sec)
```

-- query6: Display all the customers, who are either belong to the city New York or not had a grade above 100.


```
select*fromcustomerwherecity='NewYork'ornotgrade>100;
+-----+-----+-----+-----+
|customer_id|customer_name|city|grade|salesman_id|
+-----+-----+-----+-----+
| 3002 |Nick Rink |NewYork|100| 5001 |
| 3009|GeffMatt |Berlin|100| NULL|
+-----+-----+-----+-----+
2rowsinset(0.000sec)
```

-- query7:Findthosesalesmenwithallinformationwhogetsthecommissionwithinarangeof 0.12 and 0.14.

```
select*fromsalesmanwherecommission>=0.12andcommission<=0.14;
```

-- can also use between clause

```
+-----+-----+-----+
|salesman_id|name|city|commission|
+-----+-----+-----+
| 5002|nailknite|Paris| 0.13 |
| 5003|Lauson Ken | | 0.12 |
| 5006|McLyon |Paris| 0.14 |
| 5007|PaulAdam|Rome| 0.13 |
+-----+-----+-----+
4rowsinset(0.001sec)
```

--

query8:Findallthosecustomerswithallinformationwhosenamesareendingwiththeletter 'n'. select * from customer where customer_name like '%n';

```
+-----+-----+-----+-----+
|customer_id|customer_name|city|grade|salesman_id|
+-----+-----+-----+-----+
| 3008 |Joey Mann |London|200| 5007 |
+-----+-----+-----+-----+
1rowinset(0.000 sec)
```

--

query9:Findthosesalesmenwithallinformationwhosenamecontainingthe1stcharacteris 'N' and the 4th character is 'l' and rests may be any character.

```
select*fromsalesmanwherenamelike'n_i%';
```

```
+-----+-----+-----+
|salesman_id|name|city|commission|
+-----+-----+-----+
| 5002|nailknite|Paris| 0.13 |
+-----+-----+-----+
1rowinset(0.000 sec)
```

-- query10:FindthatcustomerwithallinformationwhodoesnotgetanygradeexceptNULL.

```
select * from customer where grade is null;
```

```
+-----+-----+-----+-----+
|customer_id|customer_name|city|grade|salesman_id|
+-----+-----+-----+-----+
```

|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 salesmen 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 unavailable | 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 |
+-----+-----+
5rowsinset(0.001sec)
```

--

query15:Findthehighestpurchaseamountorderedbytheeachcustomeronaparticularda
te with their ID, order date and highest purchase amount.
selectcustomer_id,order_date,max(purch_amt)fromordersgroupbycustomer_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
|
+-----+-----+
8rowsinset(0.001sec)
```

-- query16:Findthehighestpurchaseamountonadate'2021- 11-
01'foreachsalesmanwiththeir ID
selectsalesman_id,purch_amtfromorderswhereorder_date='2021- 11-
01'groupbysalesman_id;

```
+-----+
|salesman_id|purch_amt|
+-----+
| 5001| 65.26 |
+-----+
1rowinset(0.000 sec)
```

--

query17:FindthehighestpurchaseamountwiththeircustomerIDandorderdate,foronlythos
e customers who have the
-- highestpurchaseamountinadayismorethan2000.
selectcustomer_id,order_date,purch_amtfromordersgroupbycustomer_id,order_dat
ehaving 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	
+	-----	+	-----	+
3rowsinset(0.000sec)				

```
-- query18:WriteaSQLstatementthatcountsallordersforadate2021- 11- 10. select count(*) from orders where order_date ='2021- 11- 10';
```

```
+-----+
```

```
|count(*) |
```

```
+-----+
```

```
|      1 |
```

```
+-----+
```

```
1rowinset(0.000 sec)
```

EXPERIMENT 2- PROCEDURE, TRIGGERS AND CURSOR

```
-- EXPERIMENT2
```

```
-- PROCEDURE!!!!
```

```
drop procedure query_
```

```
db; delimiter //
```

```
create procedure query_db(ino_date date, outval in
```

```
t) 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(12
```


8);

```

declare@citytext(128);
declare@commissiondecimal(2,2);

-- declarecursors
declarecursor_testcursorforselect*fromsalesmanwherecommission>0.13;

-- open
cursor
opencursor_t
est;
-- loop through acursor
fetchnextfromcursor_testinto@s_id,@name,@city,@commission;
while @@fetchstatus =0
    begin
        printconcat('id:',@s_id,'/name:',@name,'/city:',@city,'/commission:',@commission);
        fetch next fromcursor_test into @s_id, @name, @city, @commission;
    end;

-- close the cursor
close cursor_test;
deallocatecursor_t
est;

delimiter ;

-- TRIGGER!!!!
--
droptriggercityval;
delimiter //
createtriggercityvalbeforeinsertoncustom

```

er for each row

begin

```
    if new.city="Uganda" then set new.city="currently delivery unavailable"; end if;
```

```

end //
delimiter;
insertintocustomervalues(3010,'Mickyme','Uganda',260,
5003); select * from customer;
+-----+-----+-----+-----+
+
|customer_id|customer_name|city          |grade|salesman_id|
+-----+-----+-----+-----+
| 3001 |Brad Pitt |London          |NULL| NULL|
| 3002 |Nick Rink |NewYork         |100| 5001 |
| 3004 |Fabio Carl |London          |300| 5006 |
| 3005 |GrahamBell|California       |200| 5002 |
| 3007 |JamesBond |Minnesota       |200| 5001 |
| 3008 |Joey Mann |London          |200| 5007 |
| 3009 |GeffMatt  |Berlin          |100| NULL|
| 3010 |Micky me  |currentlydeliveryunavailable|260| 5003 |
| 3011 |MickyMouse|currentlydeliveryunavailable|290| 5005 |
+-----+-----+-----+-----+
+
9rowsinset(0.001sec)

```

EXPERIMENT3- ACCESSING THE DATABASE

```
-- createdatabaseAds2_8;
-- useAds2_8;

-- EXPERIMENT3
-- 1.TABLEINSTRUCTOR

createtableinstructor(idintegerprimarykey,name text,dept_name text,salaryinteger);
insertintoinstructorvalues(10101,"srinivasan","comp.sci",65000),(12121,"wu","finance",
90000), (15151,"mozarat","music",40000),
(22222,"einstein","physics",95000),(32343,"elsaid","history",60000),
(33456,"gold","physics",87000),(45565,"katz","comp.sci",75000);

insertintoinstructorvalues(58583,"califeri","history",62000),(76543,"singh","finance",80000),
(76766,"crick","biology",72000),
(83821,"brandt","comp.sci",92000),(98345,"kim","elec.eng",80000);

select*frominstructor;
```

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	elsaid	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 |
+-----+-----+-----+ +
14rowsinset(0.001sec)
```

-- 2.TABLETEACHES

```
createtableteaches(idinteger,course_idtextnotnull,sec_idinteger,semesterint,yearinteger(4), foreign key(id) references instructor(id));
insertintoteachesvalues(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*fromteaches;
```

```
+-----+-----+-----+-----+ +
|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

+-----+-----+-----+-----+ +

16rowsinset(0.000sec)

--

3.Insertfollowingadditionaltupleininstructor('10211','Smith','Biology',66000)
insert into instructor values(10211,"smith","biology",66000);
QueryOK,1rowaffected(0.001sec)

-- 4.Deletethistuplefrominstructor('10211','Smith','Biology',66000)
delete from instructor where id =10211;
QueryOK,1rowaffected(0.001sec)

--

5.Selecttuplesfrominstructorwherdept_name=' History'
select* from instructor where dept_name="history";

id	name	dept_name	salary
32343	elsaid	history	60000
58583	califeri	history	62000

+-----+-----+-----+ +
2rowsinset(0.000sec)

-- 6.FindtheCartesianproductinstructorxteaches.

```
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	elsaid	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	elsaid	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 elsaid	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 elsaid	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 elsaid	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 elsaid	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 elsaid	history 60000 10101 CS- 101	1 spring 2018
33456 gold	physics 87000 10101 CS- 101	1 spring 2018
45565 katz	comp.s 75000 10101 CS- 101	1 spring 2018
	ci	
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 elsaid	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	elsaid	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	elsaid	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	elsaid	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 elsaid	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 elsaid	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 elsaid	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 elsaid	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	elsaid	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

```
+-----+-----+-----+-----+-----+-----+-----+
-----+
```

224rowsinset(0.001sec)

-- 7.Findthenamesofallinstructorswhohavetaughtsomecourseandthecourse_id

```
selectdistinctname,teaches.course_idfrominstructorjointeachesoninstructor.id=teaches.id;
```

+-----+-----+
name course_id
+-----+-----+
srinivasan CS- 101
srinivasan CS- 315
srinivasan CS- 347
wu FIN- 201
mozarat MU- 199
einstein PHY- 101
elsaid HIS- 351


```
|crick  |BIO- 101 |
|crick  |BIO- 301 |
|brandt |CS- 190  |
|brandt |CS- 319  |
|kim    |EE- 181  |
+-----+
13rowsinset(0.001sec)
```

--

8. Find the names of all instructors whose name includes the substring " dar"

```
. select name from instructor where name like "% at% ";
```

```
+-----+
|name  |
+-----+
|mozarat|
|katz  |
+-----+
2rowsinset(0.000sec)
```

--

9. Find the names of all instructors with salary between 90,000 and 100,000 (that is, $\geq 90,000$ and

$\leq 100,000$)

```
select name from instructor where salary between 90000 and 100000;
```

```
+-----+
|name  |
+-----+
|wu    |
|einstein |
|brandt|
```

+-----+

3rowsinset(0.000sec)

EXPERIMENT4- BASICSQL

-- EXPERIMENT4

--

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 |elsaid |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 |
+-----+
8rowsinset(0.000sec)

```

-- 3.FindcoursesthatraninFall2017andinSpring2018

```

selectcourse_idfromteacheswheresemester=("fall"andyear=2017)and(semester="s
pring" and year =2018);

```

```

+-----+
|course_id|
+-----+
|CS- 315 |
|FIN- 201|
|MU- 199 |
|CS- 101 |
|HIS- 351|
|CS- 319 |
|CS- 319 |
+-----+
7rowsinset(0.000sec)

```

-- 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

t. select avg(salary) as avg_salary from instructor where
dept_name='Comp.Sci';

```
+-----+  
|avg_salary |  
+-----+  
|77333.3333 |  
+-----+  
1rowinset(0.000 sec)
```

EXPERIMENT5– INTERMEDIATE SQL

-- EXPERIMENT5

--

1.FindthetotalnumberofinstructorswhoteachacourseintheSpring2018semester.

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

+-----+

|count(distinctid)|

+-----+

| 6 |

+-----+

1rowinset(0.000 sec)

-- 2.Findthenumberoftuplesintheteachesrelation

Select count(*) from teaches;

+-----+

|count(*) |

+-----+

| 16 |

+-----+

1rowinset(0.000 sec)

-- 3.Findtheaveragesalaryofinstructorsineachdepartment

selectdept_name,avg(salary)frominstructorgroupbydept_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 |
+-----+-----+
7rowsinset(0.000sec)
```

--

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 |
+-----+-----+
6rowsinset(0.000sec)
```

--

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 |elsaid      |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 |
+-----+-----+-----+ +
12rowsinset(0.000sec)

```

--

6.Findnamesofinstructorswithsalarygreaterthanthatofsome(atleastone)instructorinthe
Biology department.

```

select*frominstructorwheresalary>any(selectsalaryfrominstructorwhere
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)

EXPERIMENT6– ADVANCEDAND INTERMEDIATE SQL

-- EXPERIMENT6

--

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|
|elsaid  |HIS- 351 |
|katz    |CS- 319  |
|crick   |BIO- 101  |
|crick   |BIO- 301  |
|brandt  |CS- 190  |
|brandt  |CS- 319  |
|kim     |EE- 181  |
+-----+----+
13rowsinset(0.000sec)

```

--

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|
|elsaid    |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 |
+-----+-----+
18rowsinset(0.000sec)

```

-- 4. Create a view of instructors without their salary called
faculty

```

createviewFACULTYasselectid,name,dept_namefrominstru
ctor; 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|elsaid   |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|
```

```
+-----+-----+      +
```

```
14rowsinset(0.000sec)
```

-- 5.Giveselectprivilegesontheviewfacultytothenewuser.

```
create user "new"@"localhost" identified by 'password';
```

```
grantselectonAds2_8.FACULTYTO "new"@"localhost";
```


EXPERIMENT7- ADVANCEDSQL

```
-- EXPERIMENT7
```

```
-- 1. Create a view of instructors without their salary called  
faculty
```

```
createviewFACULTYasselectid,name,dept_namefrominstru  
ctor;
```

```
-- 2.Createaviewofdepartmentsalarytotals
```

```
createviewdept_salaryasselectdept_name,sum(salary)frominstructorgroupbydept_na  
me;
```

```
-- drop view
```

```
dept_salary;
```

```
select*fromdept_salar  
y;
```

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

-- 3.CREATEAROLEOFSTUDENT

create role student;

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

```
grantselectonAds2_8.FACULTYtostudent;
```

-- 5. Create a new user and assign her the role of student.

```
createuser"student_user"@ "localhost"identifiedby"root";
```

--

```
grantselectonAds2_8.*to"student_user"@ "localhost";
```

```
grant 'student' to "student_user"@ "localhost";
```

-- 6.LoginasthisnewuserandfindallinstructorsintheBiologydepartment.

```
select name from instructor where dept_name ='biology';
```

-- 7.Revokeprivilegesofthenewuser

-- 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.

```
grantselectonAds2_8.FACULTYto"student_user"@ "localh
```

ost";

-- 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.Loginagainasrootuser

mysql - u root - p

--

12.Createtableteaches2withsamecolumnsasteaches.

create table teaches2 select * from teaches;

select*fromteaches2;

```
+-----+-----+-----+-----+ +
|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|
+-----+-----+-----+-----+ +
16rowsinset(0.000sec)
```

-- 13.CreateindexIDcolumnof teaches.

```
createindex 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 |
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+
+-----+-----+-----+-----+
|teaches      1 |t_index|      1 |id      |A      |      16 |  NULL|NULL|YES|
2 |
BTREE      |      |      |NO      |
+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+
+-----+-----+-----+-----+
1 row in set (0.000sec)
```

```
-- 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,"elsaid","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,'elsaid','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,'elsaid', '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,'elsaid', '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,'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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,'elsaid','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,'elsaid','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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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, 'elsaid', '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.i
d= teaches.id;"

```

('srinivasan', 'CS- 101')

('srinivasan', 'CS- 315')

```

('srinivasan', 'CS- 347')
('wu', 'FIN- 201')
('mozarat', 'MU- 199')
('einstein', 'PHY- 101')
('elsaid', '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, $\geq 90,000$ and $\leq 100,000$)

```
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()

EXPERIMENT9– 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, 'elsaid', '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="selectcourse_idfromteacheswhere(semester='fall'andyear=2017)ANDNOT(semester  
='spring'andyear=2018);"
```

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

```
('CS- 347',)
```

```
('PHY- 101',)
```

```
#- -
```

```
5.Insertfollowingadditionaltuplesininstructor:('10211','Smith','Biology',66000),('10212',  
'Tom', 'Biology', NULL )
```

```
stmt="insertintoinstructorvalues(10211,'smith','biology',66000),(10212,'tom','biology',null)  
;"
```

```
#- - 6.Findallinstructorswhosesalaryisnull.
```

```
stmt="select*frominstructorwheresalaryisnull;"
```

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

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

```
stmt="selectavg(salary)asavg_salaryfrominstructorwheredept_name='Com  
p.Sci';"
```

```
# (Decimal('77333.3333'),)
```

```
#- -
```

```
8FindthetotalnumberofinstructorswhoteachacourseintheSpring2018semester.  
r. stmt="select count(distinct id) from teaches where semester = 'spring' and  
year = 2018;" (6,)
```

```
#- - 9.Findthenumberoftuplesintheteachesrelation
```

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

(16,)

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

```
stmt="selectdept_name,avg(salary)frominstructorgroupbydept_name;"
```

```
# ('biology',
Decimal('69000.0000')) #
('comp.sci',
Decimal('77333.3333'))
#('elec.eng',Decimal('80000.0
000')) # ('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('80000.0
000')) # ('finance',
Decimal('85000.0000')) #
('history',
Decimal('61000.0000')) #
('physics',
Decimal('91000.0000'))
```

```
#- - 12.Nameallinstructorswhosenameisneither" Mozart" norEinstein" .  
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, 'elsaid', '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*frominstructorwheresalary>any(selectsalaryfrominstructorwhere  
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*frominstructorwheresalary>all(selectsalaryfrominstructorwhere  
dept_name='biology');"
```

#- -

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

```
stmt="selectdept_name,avg(salary)frominstructorgroupbydept_namehavingavg(sal
```

```
ary)> 42000;"
```

```
# ('biology',
Decimal('69000.0000')) #
('comp.sci',
Decimal('77333.3333'))
#('elec.eng',Decimal('80000.0
000')) # ('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=tea
```

ches.id;"

('srinivasan', 'CS- 101')

('srinivasan', 'CS- 315')

('srinivasan', 'CS- 347')

('wu', 'FIN- 201')

('mozarat', 'MU- 199')

('einstein', 'PHY- 101')

('elsaid', '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')
('elsaid', '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()
```

EXPERIMENT10– OODBMS

-- query 1

```
CREATETYPEaddr_tyASOBJECT
  2 (street varchar2(60),
  3 city   varchar2(30),
  4 state  char(2),
  5 zip    varchar(9));
  6 /
```

Typecreated.

```
SQL>CREATETYPEperson_tyASOBJECT
  2 (namevarchar2(25),
  3 addressaddr_ty);
  4 /
```

Typecreated.

```
SQL>CREATETYPEemp_tyASOBJECT
  2 (empt_id  varchar2(9),
  3 personperson_ty);
  4
  5 /
```

Typecreated.

-- query 2

```
SQL>CREATETABLEEMP_OO
```

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

```
-- query 3
```

```
-- insert
```

```
insertintoEMP_OOvalues(emp_ty('100',person_ty('ram',addr_ty('100
st','Patiala','up','605001'))));
```

```
insertintoEMP_OOvalues(emp_ty('101',person_ty('sam',addr_ty('101
st','sire','Blore','105001'))));
```

```
-- query 4
```

```
-- select
```

```
select*fromemp_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')))
```

```
selecte.full_emp.empt_idID,e.full_emp.person.nameNAME,e.full_emp.person.addres
s.cityCITY from emp_oo e;
```

ID	NAME	CITY
----	------	------

100	Raj	Patiala
-----	-----	---------

101	sam	sire
-----	-----	------

```
-- query 5
```

```
-- update
```

```
updateemp_ooesete.full_emp.person.name='Raj'wheree.full_emp.empt_id='1000';
```

```

-- query 6
-- createnewobjwithmemberfunction
createorreplacetypenewemp_tyasobject(firstnamevarchar2(25),
lastnameVarchar2(25),birthdateDate,memberfunctionage(birthdateindate)returnnumb
er);

-- query 7
createorreplacetypebodynewemp_tyas
    memberfunctionage(birthdateindate)returnnumbe
    ris begin
        returnround(sysdate- birthdate);
    end;

end;

-- query 8
createtablenew_emp_oo(employeenewemp_ty);

-- query 9
insertintonew_emp_oovalues(newemp_ty('ram','lal','1976- 12- 12'));

-- query10 howtocallamember function
selecte.employee.firstname,e.employee.age,e.employee.age(e.employee.birthdate)f
rom new_emp_oo e;

--
query11creationofobjecttable
createtablenew_emp1ofemp_

```

ty;

```

-- query 12
insertintonew_emp1values('102',person_ty('raul',addr_ty('100TU','Pta','PB','147002'))));

-- query 13
select*fromnew_emp1;

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

-- query 14 references
selectref(p)fromnew_emp1p;

REF(P)

0000280209E44C561C843C4E90B9AB35A22AD3E8FBAFAB0D508DDF493C87F3A6F1
9DC68
04F0041DC
C90000

-- query15implementingtheconceptoffk
createtypenew_dept_ooasobject(deptnnumber(3),dnamevarchar(10));

-- query 16
createtabledept_tableofnew_dept_oo;

-- query 17
insertintodept_tablevalues(10,'co
mp');
insertintodept_tablevalues(20,'ch
em'); insert into dept_table values
(30,'math');

```


-- query 18

```
createtableemp_test_fk(empnonumber(3),namevarchar2(10),deptrefnew_dept_oo);
```

```
-- query 19
```

```
set desc depth 2
```

```
descemp_test_f
```

```
k
```

Name	Null?	Type
EMPNO		NUMBER(3)
NAME		VARCHAR2(10)
DEPT		
REFOFNEW_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;
```

```
insertintoemp_test_fkselect101,'sam',ref(p)fromdept_tablepwheredeptno  
=20;
```

```
-- query 21 accessing values
```

```
selectempno,name,deref(e.dept)fromemp_test_  
fke; EMPNO NAME
```

```
DEREF(E.DEPT)(DEPTNO, DNAME)
```

```
100 raj
```

```
NEW_DEPT_OO(10,'com
```

p')

101 sam

NEW_DEPT_OO(20,'chem')

```
select empno, name, deref(e.dept), deref(e.dept).deptno DEPTNO, deref(e.dept).dname  
eDNAME 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*fromem_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')))
```

```
00002202088ECB5F5DB94A44CD901A1BACD0D508D64D9EE4FAD8EF4404B2D19B5
```

```
A449B
```

```
8463
```

```
selecte.employee.empt_idID,e.employee.person.nameNAME,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)
```

```
DEPTNODNAME
```

```
100      ram
```

```
NEW_DEPT_OO(10,'com
```

```
p')
```

```
10 comp
```