**WORKSHEET 3**

**Student Name:** Vivek Kumar **UID:** 21BCS8129

**DOMAIN CAMP:** 16-01-2023 to 28-01-2023 **Section/Group:** DWWC-77

**Subject Name:** Database Management System

**Using the following table schema** ord\_no purch\_amt ord\_date customer\_id salesman\_id

---------- ---------- ---------- ----------- ----------- 70001 150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007

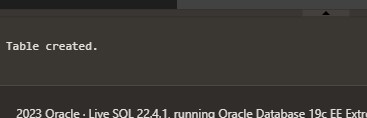
70013 3045.6 2012-04-25 3002 5001

**Ques 1 Write a SQL statement to find the total purchase amount of all orders.**

**Ques 2 Write a SQL statement to find the average purchase amount of all orders.**

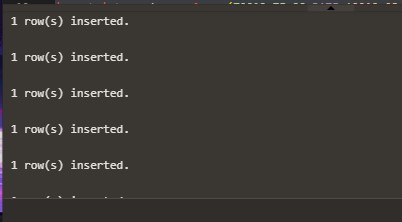
**Ques 3 Write a SQL statement to find the number of salesmen currently listing for all of their customers.**

**Solution:** create table orders(ord\_no number, purch\_amt number(6,2),ord\_date date, customer\_id number, salesman\_id number);



insert into orders values (70001,150.5,DATE '2012-10-05',3005,5002);

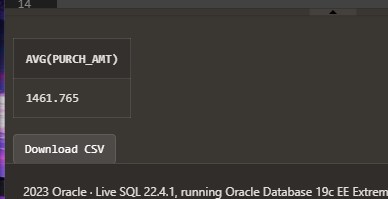
insert into orders values (70009,270.65,DATE '2012-09-10',3001,5005); insert into orders values (70002,65.26,DATE '2012-10-05',3002,5001); insert into orders values (70004,110.5,DATE '2012-08-17',3009,5003); insert into orders values (70007,948.5,DATE '2012-09-10',3005,5002); insert into orders values (70005,2400.6,DATE '2012-07-27',3007,5001); insert into orders values (70008,5760, DATE '2012-09-10',3002,5001); insert into orders values (70010,1983.43,DATE '2012-10-10',3004,5006); insert into orders values (70003,2480.4,DATE '2012-10-10',3009,5003); insert into orders values (70012,250.45,DATE '2012-06-27',3008,5002); insert into orders values (70012,75.29,DATE '2012-08-17',3003,5007); insert into orders values (70013,3045.6,DATE '2012-04-25',3002,5001);



Select sum(purch\_amt) from orders;



select avg(purch\_amt) from orders;



Select count(distinct salesman\_id) from orders;



**Using the criteria given below** customer\_id | cust\_name | city | grade | salesman\_id -------------+----------------+------------+-------+------------- 3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5005

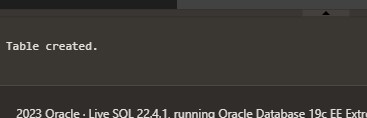
**Ques 4 Write a SQL statement to know how many customers have listed their names.**

**Ques 5 Write a SQL statement to find the number of customers who gets at least a gradation for his/her performance.**

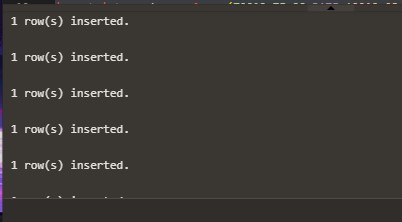
**Solution:**

Create table customers (customer\_id number, cust\_name varchar(20), city varchar(20), grade number, salesman\_id

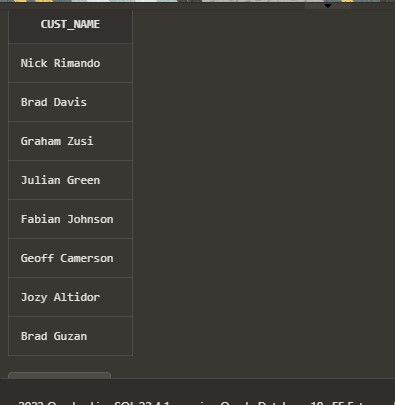
number);



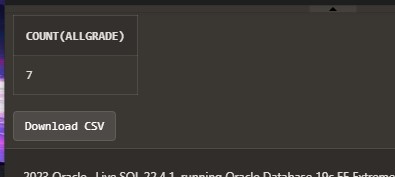
insert into customers values(3002,'Nick Rimando ', 'New York',100,5001); insert into customers values(3007,'Brad Davis ', 'New York',200,5001); insert into customers values(3005,'Graham Zusi ', 'California',200,5002); insert into customers values(3008,'Julian Green ', 'London',300,5002); insert into customers values(3004,'Fabian Johnson ', 'Paris',300,5006); insert into customers values(3009,'Geoff Camerson ', 'Berlin',100,5003); insert into customers values(3003,'Jozy Altidor', 'Moscow',200,5007); insert into customers values(3001,'Brad Guzan ', 'London', null,5005);



select cust\_name from customers;



select count(all grade) from customers;



**Use the following schema for solving questions**

ord\_no purch\_amt ord\_date customer\_id salesman\_id

---------- ---------- ---------- ----------- ----------- 70001 150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007

70013 3045.6 2012-04-25 3002 5001

**Ques 6 Write a SQL statement to get the maximum purchase amount of all the orders.**

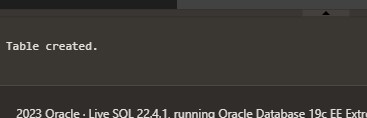
**Ques 7 Write a SQL statement to get the minimum purchase amount of all the orders.**

**Ques 8 Write a SQL statement to find the highest purchase amount ordered by each customer with their ID and highest purchase amount.**

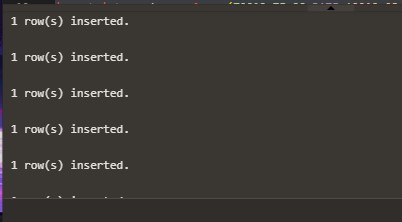
**Ques 9 Write a SQL statement to find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.**

**Ques 10 Write a SQL statement to find the highest purchase amount on a date '2012-0817' for each salesman with their ID.**

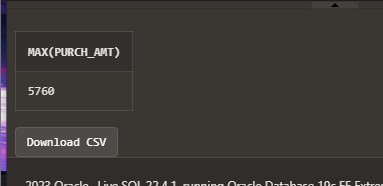
**Solution:** create table orders(ord\_no number, purch\_amt number(6,2),ord\_date date, customer\_id number, salesman\_id number);



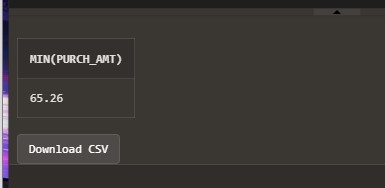
insert into orders values (70001,150.5,DATE '2012-10-05',3005,5002); insert into orders values (70009,270.65,DATE '2012-09-10',3001,5005); insert into orders values (70002,65.26,DATE '2012-10-05',3002,5001); insert into orders values (70004,110.5,DATE '2012-08-17',3009,5003); insert into orders values (70007,948.5,DATE '2012-09-10',3005,5002); insert into orders values (70005,2400.6,DATE '2012-07-27',3007,5001); insert into orders values (70008,5760, DATE '2012-09-10',3002,5001); insert into orders values (70010,1983.43,DATE '2012-10-10',3004,5006); insert into orders values (70003,2480.4,DATE '2012-10-10',3009,5003); insert into orders values (70012,250.45,DATE '2012-06-27',3008,5002); insert into orders values (70012,75.29,DATE '2012-08-17',3003,5007); insert into orders values (70013,3045.6,DATE '2012-04-25',3002,5001);



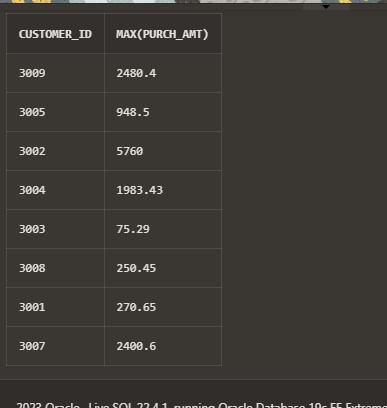
select max(purch\_amt) from orders;



select min(purch\_amt) from orders;



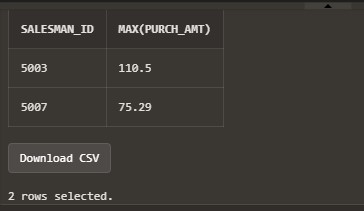
select customer\_id, max(purch\_amt) from orders group by customer\_id;



select customer\_id, ord\_date, max(purch\_amt) from orders group by customer\_id, ord\_date;



select salesman\_id, max(purch\_amt) from orders where ord\_date=DATE '2012-08-17' group by salesman\_id;



**Use the following schema** customer\_id | cust\_name | city | grade | salesman\_id -------------+----------------+------------+-------+------------- 3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

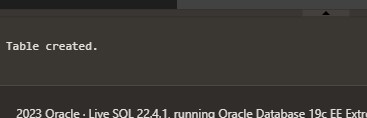
3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5005

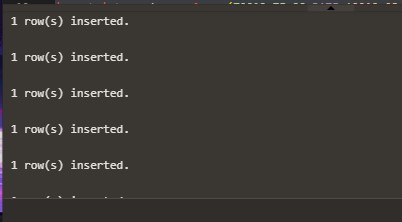
**Ques 11 Write a SQL statement which selects the highest grade for each of the cities of the customers.**

**Solution:**

Create table customers (customer\_id number, cust\_name varchar(20), city varchar(20), grade number, salesman\_id number);



insert into customers values(3002,'Nick Rimando ', 'New York',100,5001); insert into customers values(3007,'Brad Davis ', 'New York',200,5001); insert into customers values(3005,'Graham Zusi ', 'California',200,5002); insert into customers values(3008,'Julian Green ', 'London',300,5002); insert into customers values(3004,'Fabian Johnson ', 'Paris',300,5006); insert into customers values(3009,'Geoff Camerson ', 'Berlin',100,5003); insert into customers values(3003,'Jozy Altidor', 'Moscow',200,5007); insert into customers values(3001,'Brad Guzan ', 'London', null,5005);



select city, max(grade) from customers group by city;



**Use the following schema for following questions**

EMPLOYEE\_ID | FIRST\_NAME | LAST\_NAME | EMAIL | PHONE\_NUMBER |

HIRE\_DATE | JOB\_ID | SALARY | COMMISSION\_PCT | MANAGER\_ID |

DEPARTMENT\_ID |

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

| 100 | Steven | King | SKING | 515.123.4567 | 1987-06-17 | AD\_PRES |

24000.00 | 0.00 | 0 | 90 |

| 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 1987-06-18 |

AD\_VP | 17000.00 | 0.00 | 100 | 90 |

| 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 1987-06-19 | AD\_VP | 17000.00 | 0.00 | 100 | 90 |

| 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 1987-06-20 |

IT\_PROG | 9000.00 | 0.00 | 102 | 60 |

| 104 | Bruce | Ernst | BERNST | 590.423.4568 | 1987-06-21 |

IT\_PROG | 6000.00 | 0.00 | 103 | 60 |

| 105 | David | Austin | DAUSTIN | 590.423.4569 | 1987-06-22 |

IT\_PROG | 4800.00 | 0.00 | 103 | 60 |

| 106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 1987-06-23 |

IT\_PROG | 4800.00 | 0.00 | 103 | 60 |

| 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 1987-06-24 |

IT\_PROG | 4200.00 | 0.00 | 103 | 60 |

| 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 1987-06-25 |

FI\_MGR | 12000.00 | 0.00 | 101 | 100 |

| 109 | Daniel | Faviet | DFAVIET | 515.124.4169 | 1987-06-26 | FI\_ACCOUNT

| 9000.00 | 0.00 | 108 | 100 |

| 110 | John | Chen | JCHEN | 515.124.4269 | 1987-06-27 | FI\_ACCOUNT | 8200.00 | 0.00 | 108 | 100 |

| 111 | Ismael | Sciarra | ISCIARRA | 515.124.4369 | 1987-06-28 | FI\_ACCOUNT

| 7700.00 | 0.00 | 108 | 100 |

| 112 | Jose Manuel | Urman | JMURMAN | 515.124.4469 | 1987-06-29 |

FI\_ACCOUNT | 7800.00 | 0.00 | 108 | 100 |

| 113 | Luis | Popp | LPOPP | 515.124.4567 | 1987-06-30 | FI\_ACCOUNT | 6900.00 | 0.00 | 108 | 100 |

| 114 | Den | Raphaely | DRAPHEAL | 515.127.4561 | 1987-07-01 |

PU\_MAN | 11000.00 | 0.00 | 100 | 30 |

| 115 | Alexander | Khoo | AKHOO | 515.127.4562 | 1987-07-02 |

PU\_CLERK | 3100.00 | 0.00 | 114 | 30 |

| 116 | Shelli | Baida | SBAIDA | 515.127.4563 | 1987-07-03 |

PU\_CLERK | 2900.00 | 0.00 | 114 | 30 |

**Ques 12 Write a query to list the number of jobs available in the employees table.**

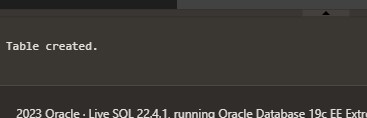
**Ques 13 Write a query to get the total salaries payable to employees**

**Ques 14 Write a query to get the minimum salary from the employees table.**

**Ques 15 Write a query to get the maximum salary of an employee working as a Programmer.**

**Solution:**

Create table employees (employee\_id number, first\_name varchar(30), last\_name varchar(30), email varchar(35), phone\_number varchar(30), hire\_date date, job\_id varchar(10), salary float, commission\_pct float, manager\_id number, department\_id number(10));

 insert into employees values(100,'Steven','King','SKING','515.123.4567', to\_date('17-06-1987', 'DD-MM-YYYY'),

'AD\_PRES',24000.00,0.00,0,90);

insert into employees values(101,'Neena','Kochhar','NKOCHHAR','515.123.4568', to\_date('18-06-1987', 'DD-MMYYYY'),'AD\_VP',17000.00,0.00,100,90);

insert into employees values(102,'Lex','De Haan','LDEHAAN','515.123.4569', to\_date('19-06-1987', 'DD-MMYYYY'),'AD\_VP',17000.00,0.00,100,90);

insert into employees values(103,'Alexander','Hunold','AHUNOLD','590.423.4567', to\_date('20-06-1987', 'DD-MMYYYY'),'IT\_PROG',9000.00,0.00,102,60);

insert into employees values(104,'Bruce','Ernst','BERNST','590.423.4568', to\_date('21-06-1987', 'DD-MMYYYY'),'IT\_PROG',6000.00,0.00,103,60);

insert into employees values(105,'David','Austin','DAUSTIN','590.423.4569', to\_date('22-06-1987', 'DD-MMYYYY'),'IT\_PROG',4800.00,0.00,103,60);

insert into employees values(106,'Valli','Pataballa','VPATABAL','590.423.4560', to\_date('23-06-1987', 'DD-MMYYYY'),'IT\_PROG',4800.00,0.00,103,60);

insert into employees values(107,'Diana','Lorentz','DLORENTZ','590.423.5567', to\_date('24-06-1987', 'DD-MMYYYY'),'IT\_PROG',4200.00,0.00,103,60);

insert into employees values(108,'Nancy','Greenberg','NGREENBE','515.124.4569', to\_date('25-06-1987', 'DD-MMYYYY'),'FI\_MGR',12000.00,0.00,101,100);

insert into employees values(109,'Daniel','Faviet','DFAVIET','515.124.4169', to\_date('26-06-1987', 'DD-MMYYYY'),'FI\_ACCOUNT',9000.00,0.00,108,100);

insert into employees values(110,'John','Chen','JCHEN','515.124.4269', to\_date('27-06-1987', 'DD-MMYYYY'),'FI\_ACCOUNT',8200.00,0.00,108,100);

insert into employees values(111,'Ismael','Sciarra','ISCIARRA','515.124.4369', to\_date('28-06-1987', 'DD-MMYYYY'),'FI\_ACCOUNT',7700.00,0.00,108,100);

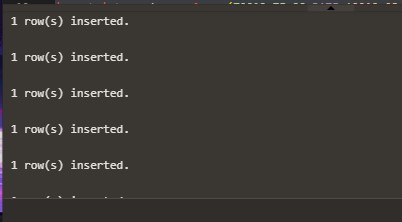
insert into employees values(112,'Jose Manuel','Urman','JMURMAN','515.124.4469', to\_date('29-06-1987', 'DD-MMYYYY'),'FI\_ACCOUNT',7800.00,0.00,108,100);

insert into employees values(113,'Luis','Popp','LPOPP','515.124.4567', to\_date('30-06-1987', 'DD-MMYYYY'),'FI\_ACCOUNT',6900.00,0.00,108,100);

insert into employees values(114,'Den','Raphaely','DRAPHEAL','515.127.4561', to\_date('01-07-1987', 'DD-MMYYYY'),'PU\_MAN',11000.00,0.00,100,30);

insert into employees values(115,'Alexander','Khoo','AKHOO','515.127.4562', to\_date('02-07-1987', 'DD-MMYYYY'),'PU\_CLERK',3100.00,0.00,114,30);

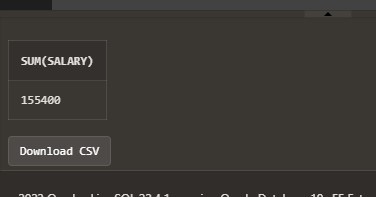
insert into employees values(116,'Shelli','Baida','SBAIDA','515.127.4563', to\_date('03-07-1987', 'DD-MMYYYY'),'PU\_CLERK',2900.00,0.00,114,30);



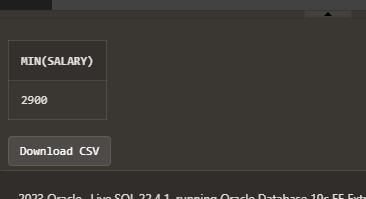
select job\_id, count(job\_id)from employees group by job\_id;



select sum(salary) from employees;



select min(salary) from employees;



select max(salary) from employees where job\_id='IT\_PROG';

