



KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

DBMS LAB ASSIG 5

- Name : HITU RAJ
- Roll no. : 2005025
- Branch : CSE

2005025 Hitu raj

-- 1. Find the average salary of each department.

```
CREATE TABLE employee
(
emp_id INTEGER NOT NULL,
f_name VARCHAR(20),
l_name VARCHAR(20),
jobtype VARCHAR(20),
salary INTEGER,
commision INTEGER,
dept VARCHAR(20),
manager_id INTEGER,
PRIMARY KEY(emp_id),
doj VARCHAR(20)
```

```
);  
DROP TABLE Employee;  
ALTER TABLE Employee ADD COLUMN doj DATE;  
INSERT INTO  
employee(emp_id,f_name,l_name,jobtype,salary,dept,doj)  
values(1, 'arun' , 'khan' , 'manager' ,90000, 'production' ,  
'1998-01-04');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,doj) values(2 ,  
'barun','kumar','manager',80000,'marketing','1998-02-09');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,manager_id,doj) values(3 ,  
'chitra','kapoor','engineer',60000,'production',1,'1998-01-0  
8');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,manager_id,doj) values(4 ,  
2  
'dheeraj','mishra','manager',75000,'sales',2,'2001-12-27');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,manager_id,doj) values(5 ,  
'emma','dutt','engineer',55000,'production',1,'2002-03-20');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,doj) values(6 ,  
'floki','dutt','accountant',70000,'accounts','2000-07-16');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,manager_id,doj) values(7 ,  
'dheeraj','kumar','clerk',40000,'accounts',6,'2016-07-01');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,doj) values(8 ,  
'saul','good','engineer',60000,'r&d','2014-09-06');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,dept,manager_id,doj) values(9 ,  
'mou','bhatt','clerk',30000,'sales',4,'2018-03-08');  
INSERT INTO employee(emp_id ,  
f_name,l_name,jobtype,salary,commision,dept,manager_id,doj)  
values(10 , 'sunny', 'deol', 'salesman', 20000, 10000, 'marketing'  
, 2, '2001-03-31');
```

```

INSERT INTO employee(emp_id ,
f_name,l_name,jobtype,salary,dept,manager_id,dof) values(11,
'bobby','deol','engineer',35000,'r&d',8,'2017-10-17');
INSERT INTO employee(emp_id ,
f_name,l_name,jobtype,salary,commision,dept,manager_id,dof)
values(12,
'amir','khan','salesman',15000,5000,'marketing',2,'2013-01-1
1');
SELECT dept AS "Department",AVG(salary) AS "Average_Salary"
FROM Employee
GROUP BY dept;

-- 2
-- Find the average salary for each jobtype according
-- to each department.
SELECT jobtype,dept AS "Department",AVG(salary) AS "Average_Salary"
FROM Employee
GROUP BY jobtype, dept
ORDER BY jobtype;

-- 3.Find the department names and their corresponding
-- average salary where the average salary is greater
-- than 40000.
SELECT dept AS "Department",AVG(salary) AS "Average_Salary"
FROM Employee
GROUP BY dept
HAVING AVG(salary)>40000;

-- 4
-- Select the departments where the maximum salary
-- is more than 55000.
SELECT dept AS "Department"
FROM Employee
GROUP BY dept
HAVING MAX(salary)>55000;

-- 5. Find the department names and their average
-- salary where the maximum salary of the department is
-- higher than 55000.

```

```
SELECT dept AS "Department",AVG(salary) AS "Average_Salary"
FROM Employee
GROUP BY dept
HAVING AVG(salary)>55000;
```

-- 6. Display the jobtypes and the total monthly salary

-- --

-- for each jobtypes as "PAYROLL", where the total
-- payroll according to jobtypes exceeds 100000/month.

```
SELECT jobtype , SUM(salary)/12 AS "PAYROLL"
FROM Employee
GROUP BY jobtype
HAVING SUM(salary)/12>100000;
```

-- 7. Display the jobtypes and the total monthly salary

-- for each jobtypes as "PAYROLL", where the total
-- payroll according to jobtypes exceeds 100000/month
-- and jobtype is not engineer.

```
SELECT jobtype , SUM(salary)/12 AS "PAYROLL"
FROM Employee
GROUP BY jobtype
HAVING SUM(salary)/12>100000 AND jobtype<>'Engineer';
```

-- 8. Display the jobtypes and the total monthly salary

-- for each jobtypes as "PAYROLL", where the total
-- payroll according to jobtypes exceeds 60000/month
-- and jobtype is not engineer and sort the list in
-- ascending order of sum of salary.

```
SELECT jobtype , SUM(salary)/12 AS "PAYROLL"
FROM Employee
GROUP BY jobtype
HAVING SUM(salary)/12>60000 AND jobtype<>'Engineer'
ORDER BY SUM(salary) ASC;
```

-- 9. Display the jobtypes and the total monthly salary

-- for each jobtypes as "PAYROLL", where the total
-- payroll according to jobtypes exceeds 50000/month
-- and jobtype is not engineer and sort the list in
-- descending order of sum of salary.

```
SELECT jobtype , SUM(salary)/12 AS "PAYROLL"
```

```

FROM Employee GROUP BY jobtype
HAVING SUM(salary)/12>50000 AND jobtype<>'Engineer'
ORDER BY SUM(salary) DESC;
-- 10.
-- Find the maximum average salary according to
-- departments.
SELECT dept AS "Department", MAX(salary) AS "Maximum Average
Salary"
FROM (
SELECT dept,AVG(salary) AS salary
FROM Employee
GROUP BY dept
) AS E;
-- 11. Find the minimum average salary according to
-- jobtypes.
SELECT dept AS "Department", MIN(salary) AS "Maximum Average
Salary"
FROM (
SELECT dept,AVG(salary) AS salary
FROM Employee
GROUP BY dept
) AS E;
-- 12. Find the employee name and date of joining who
-- are working in delhi.
CREATE TABLE department
(
d_name VARCHAR(20),
d_loc VARCHAR(20),
hod_id INTEGER NOT NULL
);
INSERT INTO department VALUES('sales','Kolkata','4');
INSERT INTO department VALUES('accounts','New Delhi','6');
INSERT INTO department VALUES('production','Kolkata','1');
INSERT INTO department VALUES('marketing','Kolkata','2');
INSERT INTO department VALUES('r&d','New Delhi','8');
SELECT CONCAT(f_name," ",l_name) AS "Employee name", doj
FROM Employee

```

```

CROSS JOIN (
SELECT d_name,d_loc
FROM department
WHERE d_loc='New Delhi'
) AS E2
ON Employee.dept=E2.d_name;

-- 13. Create the table 'Emp_Address' for storing the
-- permanent address of the employees and insert the
-- values.

CREATE TABLE Emp_Address
(
emp_id INT NOT NULL,
city VARCHAR(30),
district VARCHAR(30),
state VARCHAR(30)
);

INSERT INTO Emp_Address VALUES(1,'Suri','Birbhum','WB');
INSERT INTO Emp_Address VALUES(3,'Kolkata','Kolkata','WB');
INSERT INTO Emp_Address
VALUES(4,'Bhubaneswar','Khurda','Odisha');
INSERT INTO Emp_Address VALUES(5,'Noida','GB Nagar','UP');
INSERT INTO Emp_Address
VALUES(6,'Secunderabad','Hydrabad','Telengana');
INSERT INTO Emp_Address
VALUES(7,'Derhadun','Derhadun','Uttarakhand');
INSERT INTO Emp_Address VALUES(8,'Asansol','Burdwan','WB');
INSERT INTO Emp_Address
VALUES(9,'Siliguri','Darjeeling','WB');
INSERT INTO Emp_Address VALUES(10,'Kolkata','Kolkata','WB');
INSERT INTO Emp_Address VALUES(11,'New Delhi','New
Delhi','Delhi');

-- 14. Display the employee name with their home city
-- and the city they work in.

SELECT CONCAT(f_name," ",l_name) AS "Employee name", city AS
"Home city",d_loc AS "Work city"
FROM Employee

```

```

CROSS JOIN (
SELECT d_name,d_loc
FROM department
) AS E2 ON Employee.dept=E2.d_name
CROSS JOIN
emp_address ON Employee.emp_id=emp_address.emp_id;
-- 15. Create the following Job_Grades table.
CREATE TABLE Job_Grades
(
grade CHAR(1),
lowest_sal INT NOT NULL,
highest_sal INT
);
INSERT INTO Job_Grades VALUES('A','10000','24999');
INSERT INTO Job_Grades VALUES('B','25000','49999');
INSERT INTO Job_Grades VALUES('C','50000','100000');
SELECT emp_id,max(l_name) AS "last name" FROM Employee UNION
SELECT emp_id,MIN(l_name) AS "last name" FROM Employee ;

-- 16. Display the employee names along with their
-- salary and job_grade.
SELECT CONCAT(f_name," ",l_name) AS "Employee name" ,
salary , grade AS "Job Grade"
FROM employee
CROSS JOIN
job_grades WHERE salary BETWEEN lowest_sal AND highest_sal;
-- 17. Display the employees name along with their
-- manager's name. (use SELF JOIN)
SELECT A.f_name AS "Employee name",B.f_name AS "Manager
name"
FROM Employee A,Employee B
WHERE A.dept=B.dept AND B.jobtype='Manager';
-- 18. Display emp_id, f_name, d_loc, and hod_id (using
-- natural join).
SELECT emp_id,f_name,l_name,d_loc,hod_id FROM Employee
NATURAL JOIN department

```

```

WHERE Employee.dept = department.d_name;

-- 19. Display the employees f_name, city and state in
-- which they live (using natural join).
SELECT f_name,city,state FROM Employee
NATURAL JOIN emp_address
WHERE Employee.emp_id = emp_address.emp_id;

-- 20. Display the employees emp_id, f_name, d_loc,
-- hod_id using inner join.
SELECT emp_id,f_name,d_loc,hod_id FROM Employee
INNER JOIN department
ON Employee.dept = department.d_name;

-- 21. Display the employees f_name, city and state in
-- which they live (using inner join).
SELECT f_name,city,state FROM Employee
INNER JOIN emp_address
ON Employee.emp_id = emp_address.emp_id;

-- 22. Display the employees f_name, city and state in
-- which they live (using join keyword).
SELECT f_name,city,state FROM Employee
JOIN emp_address
ON Employee.emp_id = emp_address.emp_id;

-- 23. Insert the following two rows in the employee
-- table without inserting any value in the department
-- field.
INSERT INTO Employee (emp_id,f_name, jobtype,
salary,commision, manager_id, doj) VALUES (20,'Alex',
'Engineer', 28000, 2000, 1, '2017/01/31');
INSERT INTO Employee (emp_id,f_name,l_name, jobtype,
salary,commision, manager_id, doj) VALUES
(21,'Priya','Patel', 'Clerk', 12000, 500, 1, '2017/04/01');

-- 24. Insert the following two rows into the department
-- table.
INSERT INTO department VALUES ('Training','Mumbai',1);

```



```

INSERT INTO department VALUES ('Placement','Mumbai',1);
-- 25.Display the employees f_name, city and state in
-- which they live after joining employee and
-- employee_address table using left outer join.
SELECT f_name,city,state FROM employee
LEFT OUTER JOIN emp_address

ON employee.emp_id = emp_address.emp_id;
-- 26. Display the employees f_name and their work
-- location after joining employee and department table
-- using left join.
SELECT f_name, d_loc FROM Employee
LEFT JOIN department ON Employee.dept = department.d_name;

-- 27. Display the employees f_name and their work
-- location after joining employee and department table
-- using right join.
SELECT f_name, d_loc FROM Employee
RIGHT JOIN department ON Employee.dept = department.d_name;

-- 28. Display the employees f_name and their work
-- location after joining employee and department table
-- using full join/full outer join.
SELECT f_name, d_loc FROM Employee
LEFT JOIN department ON Employee.dept = department.d_name
UNION
SELECT f_name, d_loc FROM Employee
RIGHT JOIN department ON Employee.dept = department.d_name;

-- 29. Find the employees who are working in their home
-- city.
SELECT Employee.*
FROM Employee
CROSS JOIN department
ON Employee.dept=department.d_name
CROSS JOIN emp_address
ON Employee.emp_id=emp_address.emp_id

```

```
WHERE emp_address.city=department.d_loc;

-- 30. Find the job type having the minimum average
-- salary according to jobtypes.

SELECT jobtype , MIN(salary) AS "Minimum Salary" FROM
Employee GROUP BY jobtype;

SELECT d_name,d_loc,hod_id FROM department;

SELECT emp_id,city,district,state FROM emp_address;

SELECT emp_id,city,district,state FROM emp_address;
```


OUTPUTS:-


1 SELECT * FROM department LIMIT 100;


Input To Search Data


Free


1































Cost: 22ms < 1 > Total 7




<input checked="" type="checkbox"/>		d_name varchar(20)	d_loc varchar(20)	* hod_id int
	1	sales	Kolkata	4
	2	accounts	New Delhi	6
	3	production	Kolkata	1
	4	marketing	Kolkata	2
	5	r&d	New Delhi	8
	6	Training	Mumbai	1
	7	Placement	Mumbai	1

```
1 SELECT * FROM emp_address LIMIT 100;
```

A screenshot of the AI chat interface. The top bar is dark with a lock icon on the left and a search bar containing the text "Input To Search Data". To the right of the search bar are several icons: a gear (settings), a red "Free" badge, a blue envelope icon with a green "1" badge, a GitHub icon, a blue plus icon, a yellow plus icon, a red trash icon, and a toggle switch. Below the search bar is a row of icons: a speech bubble, a document, an up arrow, a down arrow, a green play button, and the text "Cost: 11ms". To the right of this row are navigation controls: a left arrow, a blue "1", a right arrow, and the text "Total 10".

<input checked="" type="checkbox"/>		<div><div>* emp_id</div><div>int</div></div>	<div><div>city</div><div>varchar(30)</div></div>	<div><div>district</div><div>varchar(30)</div></div>	<div><div>state</div><div>varchar(30)</div></div>
	1	1	Suri	Birbhum	WB
	2	3	Kolkata	Kolkata	WB
	3	4	Bhubaneswar	Khurda	Odisha
	4	5	Noida	GB Nagar	UP
	5	6	Secunderabad	Hydrabad	Telengana
	6	7	Derhadun	Derhadun	Uttarakhand
	7	8	Asansol	Burdwan	WB
	8	9	Siliguri	Darjeeling	WB
	9	10	Kolkata	Kolkata	WB
	10	11	New Delhi	New Delhi	Delhi

```
1 SELECT * FROM employee LIMIT 100;
```

		 emp_id int	 f_name varchar(20)	 l_name varchar(20)	 jobtype varchar(20)	 salary int	
	2	2	barun	kumar	manager	80000	
	3	3	chitra	kapoor	engineer	60000	
	4	5	emma	dutt	engineer	55000	
	5	6	floki	dutt	accountant	70000	
	6	7	dheeraj	kumar 	clerk	40000	
	7	8	saul	good	engineer	60000	
	8	9	mou	bhatt	clerk	30000	
	9	10	sunny	deol	salesman	20000	
	10	11	bobby	deol	engineer	35000	
	11	12	amir	khan	salesman	15000	
	12	20	Alex	(NULL)	Engineer	28000	
	13	21	Priya	Patel	Clerk	12000	

employee X

```
1 SELECT jobtype , MIN(salary) AS "Minimum Salary" FROM
2 Employee GROUP BY jobtype
```

Input To Search Data

Cost: 3ms < 1 > Total 5

		jobtype varchar(20)	Minimum Salary
	2	engineer	28000
	3	accountant	70000
	4	clerk	12000
	5	salesman	15000

```
1 SELECT * FROM job_grades LIMIT 100;
```

Input To Search Data

Cost: 11ms < 1 > Total 3

		grade char(1)	* lowest_sal int	highest_sal int
	1	A	10000	24999
	2	B	25000	49999
	3	C	50000	100000

```
301 SELECT f_name, d_loc FROM Employee
```

Data X

```
1 SELECT Employee.*
2 FROM Employee
3 CROSS JOIN department
4 ON Employee.dept=department.d_name
5 CROSS JOIN emp_address
6 ON Employee.emp_id=emp_address.emp_id
7 WHERE emp_address.city=department.d_loc
```

Input To Search Data

Cost: 3ms < 1 > Total 3

		* emp_id int	f_name varchar(20)	l_name varchar(20)	jobtype varchar(20)	salary int	commision int	dept varchar(20)	manag int
	1	3	chitra	kapoor	engineer	60000	(NULL)	production	1

assig9.sql

assig3.sql

assig5.sql

Data

1

2

3

4

SELECT f_name,city,state FROM employee

LEFT OUTER JOIN emp_address

ON employee.emp_id = emp_address.emp_id

Input To Search Data

Free

1

Cost: 3ms

<

1

>

Total 13

✓

Q

f_name

varchar(20)

⇅

city

⇅

state

⇅

	1	arun	Suri	WB
	2	barun	(NULL)	(NULL)
	3	chitra	Kolkata	WB
	4	emma	Noida	UP
	5	floki	Secunderabad	Telengana
	6	dheeraj	Derhadun	Uttarakhand
	7	saul	Asansol	WB
	8	mou	Siliguri	WB
	9	sunny	Kolkata	WB
	10	bobby	New Delhi	Delhi
	11	amir	(NULL)	(NULL)

assig9.sql

assig3.sql

assig5.sql

Data

1

2

3

SELECT f_name,city,state FROM Employee

JOIN emp_address

ON Employee.emp_id = emp_address.emp_id

Input To Search Data

Free

1

Cost: 2ms

<

1

>

Total 9

✓

Q

f_name

varchar(20)

⇅

city

⇅

state

⇅

	1	arun	Suri	WB
	2	chitra	Kolkata	WB
	3	emma	Noida	UP
	4	floki	Secunderabad	Telengana
	5	dheeraj	Derhadun	Uttarakhand
	6	saul	Asansol	WB
	7	mou	Siliguri	WB
	8	sunny	Kolkata	WB
	9	bobby	New Delhi	Delhi

240 -- 19. Display the employees f_name, city and state in

Data

```
1 SELECT emp_id,f_name,d_loc,hod_id FROM Employee
2 INNER JOIN department
3 ON Employee.dept = department.d_name
```

Cost: 3ms < 1 > Total 11

console

		* emp_id int	f_name varchar(20)	d_loc	hod_id
	1	1	arun	Kolkata	1
	2	2	barun	Kolkata	2
	3	3	chitra	Kolkata	1
	4	5	emma	Kolkata	1
	5	6	floki	New Delhi	6
	6	7	dheeraj	New Delhi	6
	7	8	saul	New Delhi	8
	8	9	mou	Kolkata	4
	9	10	sunny	Kolkata	2

Data

```
1 SELECT CONCAT(f_name," ",l_name) AS "Employee name" ,
2 salary , grade AS "Job Grade"
3 FROM employee
4 CROSS JOIN
5 job_grades WHERE salary BETWEEN lowest_sal AND highest_sal
```

Cost: 3ms < 1 > Total 13

		Employee name	salary int	Job Grade
	1	arun khan	90000	C
	2	barun kumar	80000	C
	3	chitra kapoor	60000	C
	4	emma dutt	55000	C
	5	floki dutt	70000	C
	6	dheeraj kumar	40000	B
	7	saul good	60000	C
	8	mou bhatt	30000	B

