DBMS ASSIGNMENT – 4

AGGREGATE FUNCTIONS

***Roll Number: U19CS012***

***Name: BHAGYA VINOD RANA***

**Q1)** Create a table **Employee** with fields

EmpID Number (6) Primary key

Name Character (25)

Department Character (30)

Manager ID Number (6)

JoiningDate Date

Salary Number (8)

Insert 15 Rows in the above created table.

**SQL-Code [**SQLite 3.29.0**]:**

BEGIN TRANSACTION;

CREATE TABLE EMPLOYEE(

    emp\_id integer PRIMARY KEY,

    emp\_name text,

    department text,

    manager\_id integer,

*-- YEAR MONTH DAY [Important Mistake!]*

    joining\_date DATE,

    salary integer

);

*-- Insert 10 Rows in the above created table.*

INSERT INTO EMPLOYEE VALUES(

    4123,

    'Ninja\_Hatori',

    'Production',

    1002,

    '2020-04-01',

    65000

);

INSERT INTO EMPLOYEE VALUES(

    4129,

    'Ajay',

    'Research',

    1027,

    '2018-04-02',

    45000

);

INSERT INTO EMPLOYEE VALUES(

    4230,

    'Mickey',

    'Marketing',

    1022,

    '2016-04-03',

    35000

);

INSERT INTO EMPLOYEE VALUES(

    4428,

    'Kiteretsu',

    'Accounting',

    1012,

    '2019-04-04',

    75000

);

INSERT INTO EMPLOYEE VALUES(

    4073,

    'Shizuka',

    'HR',

    1035,

    '2020-04-05',

    60000

);

INSERT INTO EMPLOYEE VALUES(

    4983,

    'Aditya',

    'HR',

    1035,

    '2017-04-06',

    100000

);

INSERT INTO EMPLOYEE VALUES(

    4009,

    'Nobita',

    'Research',

    1027,

    '2015-04-07',

    50000

);

INSERT INTO EMPLOYEE VALUES(

    4773,

    'Doraemon',

    'Marketing',

    1022,

    '2020-04-08',

    25000

);

INSERT INTO EMPLOYEE VALUES(

    4833,

    'Gian',

    'Accounting',

    1012,

    '2018-04-09',

    95000

);

INSERT INTO EMPLOYEE VALUES(

    4337,

    'Donald',

    'HR',

    1035,

    '2012-04-10',

    55000

);

INSERT INTO EMPLOYEE VALUES(

    4113,

    'Akash',

    'HR',

    1035,

    '2017-04-06',

    110000

);

INSERT INTO EMPLOYEE VALUES(

    4010,

    'Naruto',

    'HR',

    1027,

    '2013-05-09',

    52000

);

INSERT INTO EMPLOYEE VALUES(

    4768,

    'Dishant',

    'Marketing',

    1022,

    '2019-08-02',

    35000

);

INSERT INTO EMPLOYEE VALUES(

    4830,

    'Gopal',

    'Marketing',

    1012,

    '2020-08-01',

    83000

);

INSERT INTO EMPLOYEE VALUES(

    4331,

    'Deepak',

    'Marketing',

    1022,

    '2014-02-14',

    57000

);

*-- Saving the Work*

COMMIT;

*-- For Checking the Inserted Values*

*-- SELECT \* FROM EMPLOYEE*

*-- 1. Display Manager Id of employees whose name starts with ‘A’.*

SELECT manager\_id FROM EMPLOYEE WHERE emp\_name LIKE 'A%'

*-- For Cross Checking*

*-- SELECT emp\_name,manager\_id FROM EMPLOYEE WHERE emp\_name LIKE 'A%'*

*-- 2. Display employees Id and employee name, department wise.*

SELECT emp\_id,emp\_name FROM EMPLOYEE ORDER BY department

*-- For Cross Checking*

*-- SELECT department,emp\_id,emp\_name FROM EMPLOYEE ORDER BY department*

*-- 3. Display employee count department wise.*

SELECT department,COUNT(\*) FROM EMPLOYEE GROUP BY department

*-- 4. Display all columns of employees whose experience is more than 3 years.*

SELECT \* FROM EMPLOYEE WHERE (joining\_date)<(DATE('now','-3 year'))

*-- 5. Display departments with more than 4 employees.*

SELECT department FROM EMPLOYEE GROUP BY department HAVING COUNT(\*)>4

*-- For Cross Checking*

*-- SELECT department,COUNT(\*) FROM EMPLOYEE GROUP BY department HAVING COUNT(\*)>4*

*-- 6. Display employees Id and employee name whose salary is greater than 50000, department wise.*

SELECT emp\_id, emp\_name FROM EMPLOYEE GROUP BY department HAVING salary>50000

*-- For Checking*

*-- SELECT emp\_id, emp\_name, salary FROM EMPLOYEE GROUP BY department HAVING salary>50000*

*-- 7. Display department name and average salary of employees in department wise.*

SELECT department,AVG(salary) FROM EMPLOYEE GROUP BY department

*-- 8. Display Employee Id and Name of employee with highest salary.*

SELECT emp\_id, emp\_name, MAX(salary) FROM EMPLOYEE

*-- 9. Display employees Id and employee name with least salary*

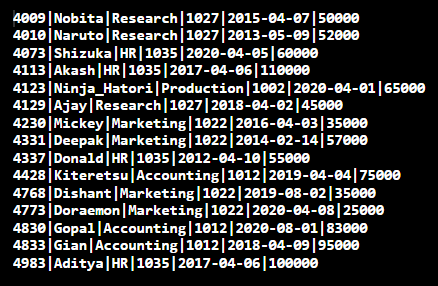
SELECT emp\_id, emp\_name, MIN(salary) FROM EMPLOYEE

*-- 10. Display employees Id and employee name with second highest salary.*

SELECT emp\_id, emp\_name, MAX(salary) FROM EMPLOYEE WHERE salary < (SELECT MAX(salary) FROM EMPLOYEE)

**Use Employee table from Assignment 3**

**Initial Table:**



1. Display Manager Id of employees whose name starts with ‘A’.

**Query:**

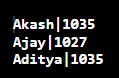
**Q1)**

SELECT manager\_id FROM EMPLOYEE WHERE emp\_name LIKE 'A%'

**Q1) For Checking**

SELECT emp\_name,manager\_id FROM EMPLOYEE WHERE emp\_name LIKE 'A%'

**Output:**



2. Display employees Id and employee name, department wise.

**Query:**

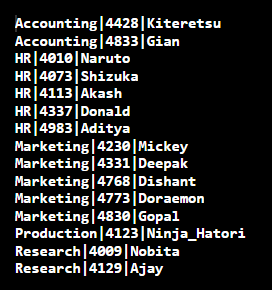
**Q2)**

SELECT emp\_id,emp\_name FROM EMPLOYEE ORDER BY department

**Q2) For Checking**

SELECT department,emp\_id,emp\_name FROM EMPLOYEE ORDER BY department

**Output:**

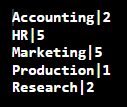


3. Display employee count department wise.

**Query:**

SELECT department,COUNT(\*) FROM EMPLOYEE GROUP BY department

**Output:**

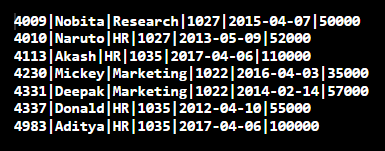


4. Display all columns of employees whose experience is more than 3 years.

**Query:**

SELECT \* FROM EMPLOYEE WHERE (joining\_date)<(DATE('now','-3 year'))

**Output:**



5. Display departments with more than 4 employees.

**Query:**

**Q5)**

SELECT department FROM EMPLOYEE GROUP BY department HAVING COUNT(\*)>4

**Q5) For Checking**

SELECT department,COUNT(\*) FROM EMPLOYEE GROUP BY department HAVING COUNT(\*)>4

**Output:**



6. Display employees Id and employee name whose salary is greater than 50000, Department wise.

**Query:**

**Q6)**

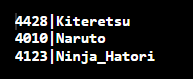
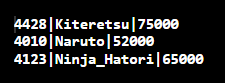
SELECT emp\_id, emp\_name FROM EMPLOYEE GROUP BY department HAVING salary>50000

**Q6) For Checking**

SELECT emp\_id, emp\_name, salary FROM EMPLOYEE

GROUP BY department HAVING salary>50000

**Output:**

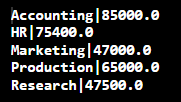
 

7. Display department name and average salary of employees in department wise.

**Query:**

SELECT department,AVG(salary) FROM EMPLOYEE GROUP BY department

**Output:**



8. Display Employee Id and Name of employee with highest salary.

**Query:**

SELECT emp\_id, emp\_name, MAX(salary) FROM EMPLOYEE

**Output:**



9. Display employees Id and employee name with least salary

**Query:**

SELECT emp\_id, emp\_name, MIN(salary) FROM EMPLOYEE

**Output:**



10. Display employees Id and employee name with second highest salary.

**Query:**

SELECT emp\_id, emp\_name, MAX(salary) FROM EMPLOYEE

WHERE salary < (SELECT MAX(salary) FROM EMPLOYEE)

**Output:**



**PART2: STUDENT TABLE**

Insert 15 Rows in the above created table.

**SQL-Code [**SQLite 3.29.0**]:**

BEGIN TRANSACTION;

CREATE TABLE STUDENT(

    roll\_no integer,

    stud\_name text,

    semester integer,

    dept\_name text,

*-- YEAR MONTH DAY [Important Mistake!]*

    date\_of\_birth DATE,

    admission\_date DATE,

    hostel\_room integer

*-- Null Values also Allowed in Hostel Room*

);

*-- Insert 15 Rows in the above created table.*

INSERT INTO STUDENT VALUES(1,'Alfred',1,'C.S.E.','2002-02-14','2021-01-15',93);

INSERT INTO STUDENT VALUES(5,'Bunty',6,'CHEMICAL','2000-03-18','2018-03-18',120);

INSERT INTO STUDENT VALUES(7,'Sunio',8,'E.C.E.','1999-01-04','2017-05-24',178);

INSERT INTO STUDENT VALUES(22,'Gian',3,'CIVIL','2001-08-07','2019-07-13',NULL);

INSERT INTO STUDENT VALUES(1,'Kitretsu',2,'E.C.E.','2002-07-19','2020-09-22',198);

INSERT INTO STUDENT VALUES(8,'Larry',5,'C.S.E.','2000-08-21','2018-02-19',NULL);

INSERT INTO STUDENT VALUES(1,'Newton',2,'CHEMICAL','2002-09-27','2020-04-10',207);

INSERT INTO STUDENT VALUES(43,'Moore',4,'E.C.E.','2001-08-17','2019-06-11',NULL);

INSERT INTO STUDENT VALUES(1,'Sachin',1,'MECHANICAL','2000-10-09','2020-08-17',234);

INSERT INTO STUDENT VALUES(10,'Abdul',3,'CIVIL','2001-11-16','2019-10-03',217);

INSERT INTO STUDENT VALUES(21,'Arham',4,'C.S.E.','2001-03-15','2019-12-06',314);

INSERT INTO STUDENT VALUES(59,'John',1,'MECHANICAL','2002-08-25','2020-11-09',NULL);

INSERT INTO STUDENT VALUES(72,'Anand',7,'MECHANICAL','1999-12-30','2017-03-27',404);

INSERT INTO STUDENT VALUES(1,'Jethalal',6,'CIVIL','2000-10-11','2018-06-30',102);

INSERT INTO STUDENT VALUES(17,'Shizuka',3,'C.S.E.','2001-12-06','2019-07-12',NULL);

*-- Saving the Work*

COMMIT;

*-- -----------------------------------------------------------------*

*-- For Output Formatting [Human Understandable Form] in SQLite*

.mode column

.headers on

.separator ROW "\n"

.nullvalue NULL

*-- For Checking the Inserted Values*

*-- SELECT \* FROM STUDENT*

*-- 1. Display semester of students whose name has the letter ‘a’.*

*-- [GROP = Case Sensistive]  [LIKE = Case In-sensititive (A & a Both Tuples)]*

SELECT semester FROM STUDENT WHERE stud\_name LIKE "A%";

*-- For Checking*

SELECT stud\_name,semester FROM STUDENT WHERE stud\_name LIKE "A%";

*-- 2. Display count of students semester wise.*

SELECT semester AS "SEMESTER",COUNT(\*) AS "NUMBER OF STUDENTS" FROM STUDENT GROUP BY semester;

*-- 3. Display students' names from every department whose roll number is 1.*

SELECT stud\_name,roll\_no,dept\_name FROM STUDENT WHERE roll\_no=1;

*-- 4. Display student name and semester of students who are not staying in the hostel.*

SELECT stud\_name,semester FROM STUDENT WHERE hostel\_room IS NULL;

*-- For Checking*

SELECT stud\_name,semester,hostel\_room FROM STUDENT WHERE hostel\_room IS NULL;

*-- 5. Display student count in each semester whose birth month is august.*

SELECT semester, COUNT(\*) AS "NUMBER OF STUDENTS" FROM STUDENT

WHERE strftime('%m',date\_of\_birth) == '08'

GROUP BY semester ;

*-- For Checking*

SELECT stud\_name,semester,date\_of\_birth FROM STUDENT WHERE strftime('%m',date\_of\_birth) == '08' ORDER BY semester;

*-- 6. Display roll number and name of the student who was the first one to get admission in the college.*

SELECT roll\_no,stud\_name,MIN(admission\_date) FROM STUDENT ;

*-- 7. Display the average count of students. ( In any semester)*

SELECT AVG(cnt) FROM (SELECT COUNT(\*) as cnt FROM STUDENT GROUP BY semester);

*-- 8. For every month (Jan-Dec) display the count of students who are having birthdays in that month.*

SELECT strftime("%m",date\_of\_birth) AS "MONTH [01-12]",COUNT(\*) AS "NUMBER OF STUDENTS"

FROM STUDENT

GROUP BY strftime("%m",date\_of\_birth);

*-- 9. Display count of students who have taken admission in the last six months.*

SELECT COUNT(\*) AS "Last 6 Months Students Admitted" FROM STUDENT WHERE admission\_date > (SELECT date('now','-6 month'));

*-- For Checking*

SELECT stud\_name,admission\_date FROM STUDENT WHERE admission\_date > (SELECT date('now','-6 month'));

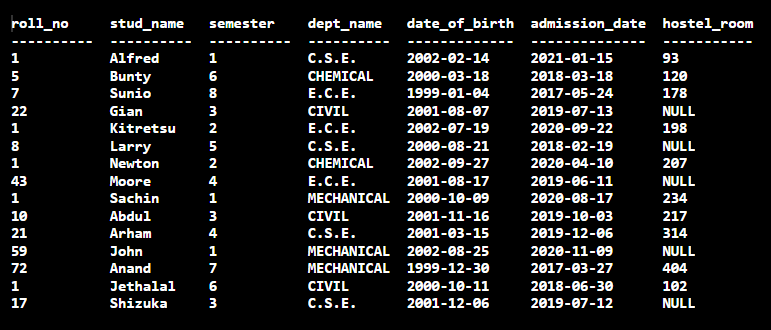
*-- 10. Display semester with least number of students.*

SELECT semester,COUNT(semester) FROM STUDENT

GROUP BY semester

HAVING COUNT(semester)==(SELECT MIN(semester) FROM (SELECT semester,COUNT(semester) FROM STUDENT GROUP BY semester));

**Initial Table:**



**Use Student table from Assignment 3**

1. Display semester of students whose name has the letter ‘A’.

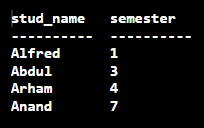
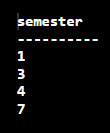
**Query:**

SELECT semester FROM STUDENT WHERE stud\_name LIKE "A%"

**For Checking**

SELECT stud\_name,semester FROM STUDENT WHERE stud\_name LIKE "A%"

**Output:**

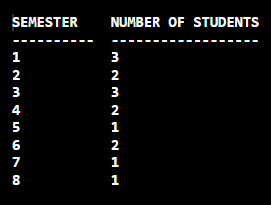


2. Display count of Student’s semester wise.

**Query:**

SELECT semester AS "SEMESTER",COUNT(\*) AS "NUMBER OF STUDENTS" FROM STUDENT GROUP BY semester

**Output:**

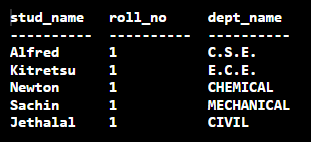


3. Display students' names from every department whose roll number is 1.

**Query:**

SELECT stud\_name,roll\_no,dept\_name FROM STUDENT WHERE roll\_no=1

**Output:**



4. Display student name and semester of students who are not staying in the hostel.

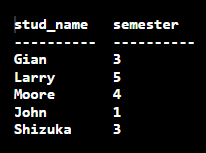
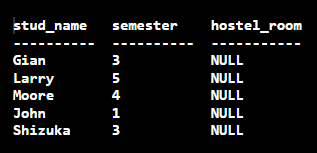
**Query:**

SELECT stud\_name,semester FROM STUDENT WHERE hostel\_room IS NULL

**For Checking**

SELECT stud\_name,semester,hostel\_room FROM STUDENT WHERE hostel\_room IS NULL

**Output:**

5. Display student count in each semester whose birth month is august.

**Query:**

SELECT semester, COUNT(\*) AS "NUMBER OF STUDENTS" FROM STUDENT

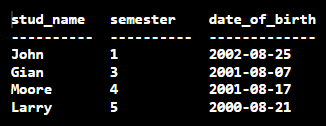
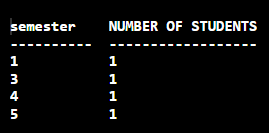
WHERE strftime('%m',date\_of\_birth) == '08'

GROUP BY semester

**For Checking**

SELECT stud\_name,semester,date\_of\_birth FROM STUDENT WHERE strftime('%m',date\_of\_birth) == '08' ORDER BY semester

**Output:**



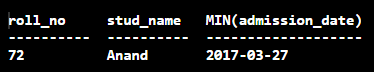
6. Display roll number and name of the student who was the first one to get admission in

the college.

**Query:**

SELECT roll\_no,stud\_name,MIN(admission\_date) FROM STUDENT

**Output:**

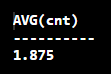


7. Display the average count of students. ( In any semester)

**Query:**

SELECT AVG(cnt) FROM (SELECT COUNT(\*) as cnt FROM STUDENT GROUP BY semester)

**Output:**



*Implies that on an Each Semester has an* ***Average*** *of 1.875* ***Students/Semester***

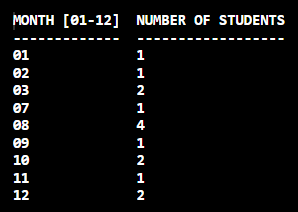
8. For every month (Jan-Dec) display the count of students who are having birthdays in that Month.

**Query:**

SELECT strftime("%m",date\_of\_birth) AS "MONTH [01-12]",COUNT(\*) AS "NUMBER OF STUDENTS"

FROM STUDENT

GROUP BY strftime("%m",date\_of\_birth);



9. Display count of students who have taken admission in the last six months.

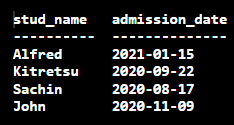
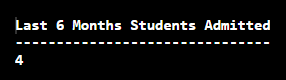
**Query:**

SELECT COUNT(\*) AS "Last 6 Months Students Admitted" FROM STUDENT WHERE admission\_date > (SELECT date('now','-6 month'));

**For Checking**

SELECT stud\_name,admission\_date FROM STUDENT WHERE admission\_date > (SELECT date('now','-6 month'));

**Output: [2020-08-08] = [‘2021-02-08’ – ‘6 Months’]**



10. Display semester with least number of students.

**Query:**

SELECT semester,COUNT(semester) FROM STUDENT

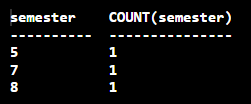
GROUP BY semester

HAVING COUNT(semester)==(SELECT MIN(semester) FROM (SELECT semester,COUNT(semester)

FROM STUDENT GROUP BY semester))

**For Checking [Refer Q2]**

**Output:**



**Submitted By:**

**BHAGYA VINOD RANA**

**U19CS012**