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
);
```

```sql
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 wi
se.
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 EM
PLOYEE)
```

Use Employee table from Assignment 3

Initial Table:

```
4009|Nobita|Research|1027|2015-04-07|50000
4010|Naruto|Research|1027|2013-05-09|52000
4073|Shizuka|HR|1035|2020-04-05|60000
4113|Akash|HR|1035|2017-04-06|110000
4123|Ninja_Hatori|Production|1002|2020-04-01|65000
4129|Ajay|Research|1027|2018-04-02|45000
4230|Mickey|Marketing|1022|2016-04-03|35000
4331|Deepak|Marketing|1022|2014-02-14|57000
4337|Donald|HR|1035|2012-04-10|55000
4428|Kiteretsu|Accounting|1012|2019-04-04|75000
4768|Dishant|Marketing|1022|2019-08-02|35000
4773|Doraemon|Marketing|1022|2020-04-08|25000
4830|Gopal|Accounting|1012|2020-08-01|83000
4833|Gian|Accounting|1012|2018-04-09|95000
4983|Aditya|HR|1035|2017-04-06|100000
```

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

Query:

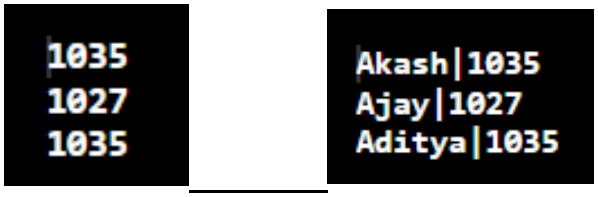
Q1)

```
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:

```
Accounting|2
HR|5
Marketing|5
Production|1
Research|2
```

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:

```
4009|Nobita|Research|1027|2015-04-07|50000
4010|Naruto|HR|1027|2013-05-09|52000
4113|Akash|HR|1035|2017-04-06|110000
4230|Mickey|Marketing|1022|2016-04-03|35000
4331|Deepak|Marketing|1022|2014-02-14|57000
4337|Donald|HR|1035|2012-04-10|55000
4983|Aditya|HR|1035|2017-04-06|100000
```

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:

```
HR
Marketing
```

```
HR|5
Marketing|5
```

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:

```
4428|Kiteretsu
4010|Naruto
4123|Ninja_Hatori
```

```
4428|Kiteretsu|75000
4010|Naruto|52000
4123|Ninja_Hatori|65000
```

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

Query:

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

Output:

```
Accounting|85000.0
HR|75400.0
Marketing|47000.0
Production|65000.0
Research|47500.0
```

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

Query:

```
SELECT emp_id, emp_name, MAX(salary) FROM EMPLOYEE
```

Output:

```
4113|Akash|110000
```

9. Display employees Id and employee name with least salary

Query:

```
SELECT emp_id, emp_name, MIN(salary) FROM EMPLOYEE
```

Output:

```
4773|Doraemon|25000
```

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:

```
4983|Aditya|100000
```

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
```

```sql
-- 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) == '0
8' ORDER BY semester;

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

```sql
SELECT COUNT(*) AS "Last 6 Months Students Admitted" FROM STUDENT WHERE admission_date > (SEL
ECT 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 STUD
ENT GROUP BY semester));
```

Initial Table:

roll_no	stud_name	semester	dept_name	date_of_birth	admission_date	hostel_room
1	Alfred	1	C.S.E.	2002-02-14	2021-01-15	93
5	Bunty	6	CHEMICAL	2000-03-18	2018-03-18	120
7	Sunio	8	E.C.E.	1999-01-04	2017-05-24	178
22	Gian	3	CIVIL	2001-08-07	2019-07-13	NULL
1	Kitretsu	2	E.C.E.	2002-07-19	2020-09-22	198
8	Larry	5	C.S.E.	2000-08-21	2018-02-19	NULL
1	Newton	2	CHEMICAL	2002-09-27	2020-04-10	207
43	Moore	4	E.C.E.	2001-08-17	2019-06-11	NULL
1	Sachin	1	MECHANICAL	2000-10-09	2020-08-17	234
10	Abdul	3	CIVIL	2001-11-16	2019-10-03	217
21	Arham	4	C.S.E.	2001-03-15	2019-12-06	314
59	John	1	MECHANICAL	2002-08-25	2020-11-09	NULL
72	Anand	7	MECHANICAL	1999-12-30	2017-03-27	404
1	Jethalal	6	CIVIL	2000-10-11	2018-06-30	102
17	Shizuka	3	C.S.E.	2001-12-06	2019-07-12	NULL

Use Student table from Assignment 3

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

Query:

```sql
SELECT semester FROM STUDENT WHERE stud_name LIKE "A%"
```

For Checking

```sql
SELECT stud_name,semester FROM STUDENT WHERE stud_name LIKE "A%"
```

Output:

semester
1
3
4
7

stud_name	semester
Alfred	1
Abdul	3
Arham	4
Anand	7

2. Display count of Student's semester wise.

Query:

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

Output:

SEMESTER	NUMBER OF STUDENTS
1	3
2	2
3	3
4	2
5	1
6	2
7	1
8	1

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:

stud_name	roll_no	dept_name
Alfred	1	C.S.E.
Kitretsu	1	E.C.E.
Newton	1	CHEMICAL
Sachin	1	MECHANICAL
Jethalal	1	CIVIL

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:

stud_name	semester
Gian	3
Larry	5
Moore	4
John	1
Shizuka	3

stud_name	semester	hostel_room
Gian	3	NULL
Larry	5	NULL
Moore	4	NULL
John	1	NULL
Shizuka	3	NULL

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) == '0
8' ORDER BY semester
```

Output:

semester	NUMBER OF STUDENTS
1	1
3	1
4	1
5	1

stud_name	semester	date_of_birth
John	1	2002-08-25
Gian	3	2001-08-07
Moore	4	2001-08-17
Larry	5	2000-08-21

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:

roll_no	stud_name	MIN(admission_date)
72	Anand	2017-03-27

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:

AVG(cnt)
1.875

*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);
```

MONTH [01-12]	NUMBER OF STUDENTS
01	1
02	1
03	2
07	1
08	4
09	1
10	2
11	1
12	2

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 > (SEL
ECT 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']

```
Last 6 Months Students Admitted
-------------------------------
4
```

stud_name	admission_date
Alfred	2021-01-15
Kitretsu	2020-09-22
Sachin	2020-08-17
John	2020-11-09

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:

semester	COUNT(semester)
5	1
7	1
8	1

Submitted By:

BHAGYA VINOD RANA

U19CS012