Please complete this assignment within 48 hours. And note, every query has some points to it. Hence you should try to write the maximum number of queries.

Please use postgresql/mysql to create following tables:

1) **users table: name(user name), active(boolean to check if user is active)**

```
CREATE SEQUENCE users_id_seq;
CREATE TABLE users (
  id INTEGER PRIMARY KEY NOT NULL DEFAULT NEXTVAL ('users_id_seq'),
  name VARCHAR(50) NOT NULL,
  active BOOLEAN NOT NULL DEFAULT true,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

2) **batches table: name(batch name), active(boolean to check if batch is active)**

```
CREATE SEQUENCE batch_id_seq;
CREATE TABLE batches (
  id INTEGER PRIMARY KEY NOT NULL DEFAULT NEXTVAL('batch_id_seq'),
  name VARCHAR(100) UNIQUE NOT NULL,
  active BOOLEAN NOT NULL DEFAULT true,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

3) **student_batch_maps table: this table is a mapping of the student and his batch. deactivated_at is the time when a student is made inactive in a batch.**

```
CREATE SEQUENCE student_batch_maps_id_seq;
CREATE TABLE student_batch_maps (
  id INTEGER PRIMARY KEY NOT NULL DEFAULT
NEXTVAL('student_batch_maps_id_seq'),
  user_id INTEGER NOT NULL REFERENCES users(id),
  batch_id INTEGER NOT NULL REFERENCES batches(id),
  active BOOLEAN NOT NULL DEFAULT true,
  deactivated_at TIMESTAMP NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

4) **instructor_batch_maps table: this table is a mapping of the instructor and the batch he has been assigned to take class/session.**

```
CREATE SEQUENCE instructor_batch_maps_id_seq;
CREATE TABLE instructor_batch_maps (
  id INTEGER PRIMARY KEY NOT NULL DEFAULT
NEXTVAL('instructor_batch_maps_id_seq'),
  user_id INTEGER REFERENCES users(id),
  batch_id INTEGER REFERENCES batches(id),
  active BOOLEAN NOT NULL DEFAULT true,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

5) **sessions table: Every day session happens where the teacher takes a session or class of students.**

```
CREATE SEQUENCE sessions_id_seq;
CREATE TABLE sessions (
  id INTEGER PRIMARY KEY NOT NULL DEFAULT NEXTVAL('sessions_id_seq'),
  conducted_by INTEGER NOT NULL REFERENCES users(id),
  batch_id INTEGER NOT NULL REFERENCES batches(id),
  start_time TIMESTAMP NOT NULL,
  end_time TIMESTAMP NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
);
```

6) **attendances table: After session or class happens between teacher and student, attendance is given by student. students provide ratings to the teacher.**

```
CREATE TABLE attendances (
  student_id INTEGER NOT NULL REFERENCES users(id),
  session_id INTEGER NOT NULL REFERENCES sessions(id),
  rating DOUBLE PRECISION NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (student_id, session_id)
);
```

7) **tests table: Test is created by instructor. total_mark is the maximum marks for the test.**

```
CREATE SEQUENCE tests_id_seq;
CREATE TABLE tests (
   id INTEGER PRIMARY KEY NOT NULL DEFAULT NEXTVAL('tests_id_seq'),
  batch_id INTEGER REFERENCES batches(id),
  created_by INTEGER REFERENCES users(id),
  total_mark INTEGER NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
);
```

8) **test_scores table: Marks scored by students are added in the test_scores table.**

```
CREATE TABLE test_scores (
  test_id INTEGER REFERENCES tests(id),
  user_id INTEGER REFERENCES users(id),
  score INTEGER NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY(test_id, user_id)
);
```

Using the above table schema, please write the following queries. To test your queries, you can use some dummy data.

1) Calculate the average rating given by students to each teacher for each session created. Also, provide the batch name for which session was conducted.
2) Find the attendance percentage for each session for each batch. Also mention the batch name and users name who has conduct that session
3) What is the average marks scored by each student in all the tests the student had appeared?
4) A student is passed when he scores 40 percent of total marks in a test. Find out how many students passed in each test. Also mention the batch name for that test.
5) A student can be transferred from one batch to another batch. If he is transferred from batch a to batch b. batch b's active=true and batch a's active=false in student_batch_maps.
   At a time, one student can be active in one batch only. One Student can not be transferred more than four times. Calculate each student's attendance percentage for all the sessions created for his past batch. Consider only those sessions for which he was active in that past batch.

Note - Data is not provided for these tables, you can insert some dummy data if required.